Tarea Ejercicios bandit 0-17

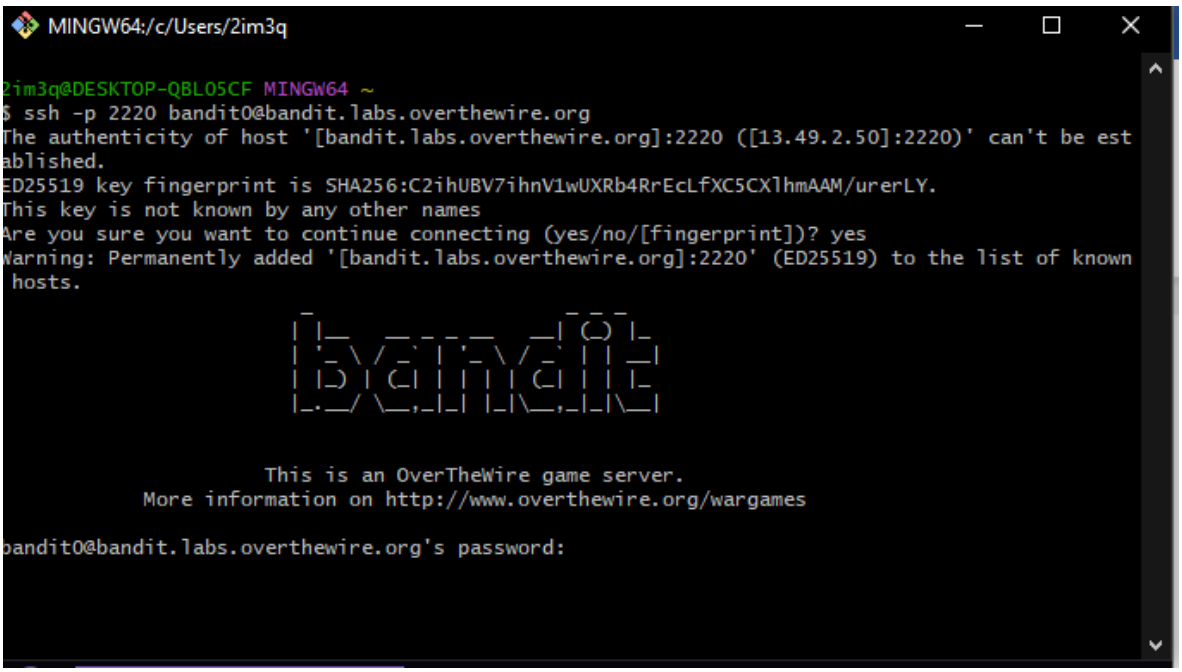
Alumno: Roberto Carlos Quintana Escamilla

Temas especializados en seguridad

bandit 0

Lo primero que se nos esenña es a conectarnos al servidor mediante el comando ssh. Utilizando en el puerto 2220 con el usuario bandit0 de la siguiente manera.

ssh -p 2220 bandit0@bandit.labs.overthewire.org



Ingresando la contraseña dada en las instrucciones del ejercisio

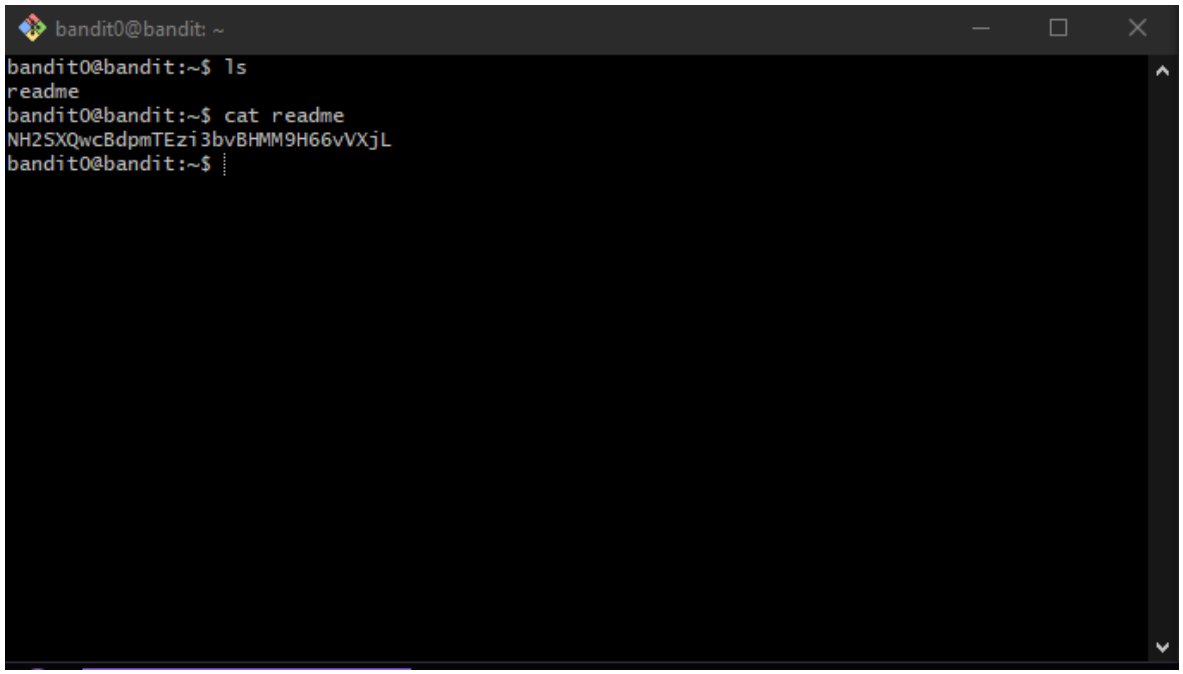
pass: bandit0

bandid 1

Para obtener la contraseña para el siguiente nivel debemos obtenerlo del archivo de texto readme el cual se puede obtener con el comando cat. Pero primero para verificar que el archivo se encuentra en el fichero podemos ver los archivos no ocultos con el comando ls.

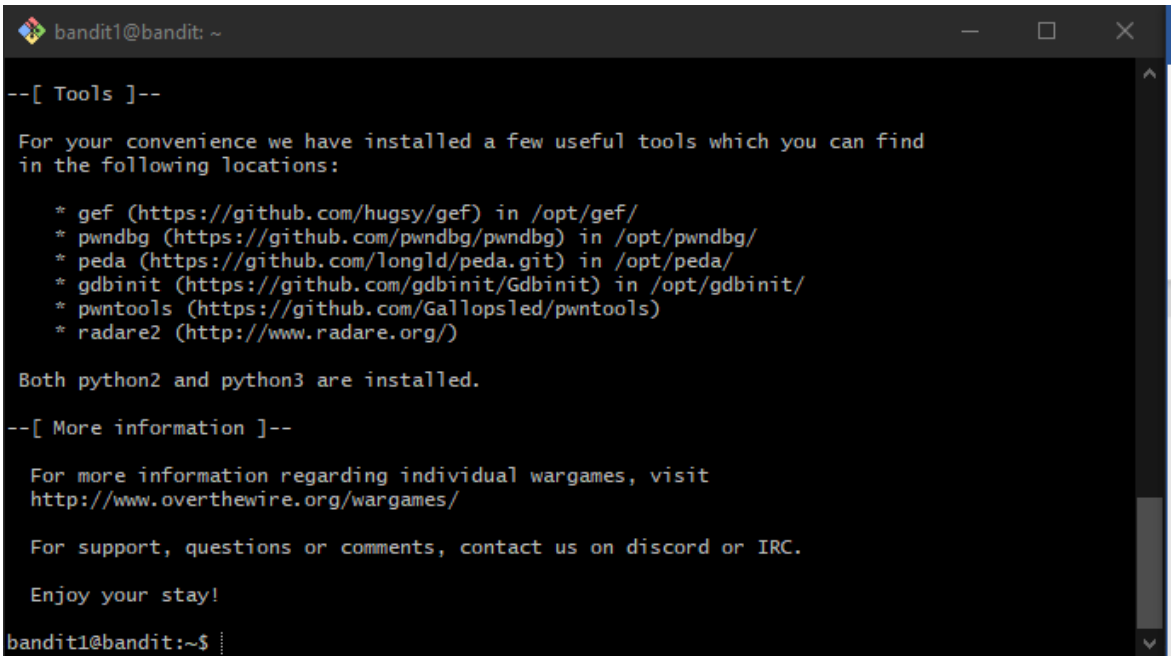
ls

cat readme



pass: NH2SXQwcBdpmTEzi3bvBHMM9H66vVXjL

Únicamente resta desloguearse y loguearse con el nuevo usuario y cotraseña.



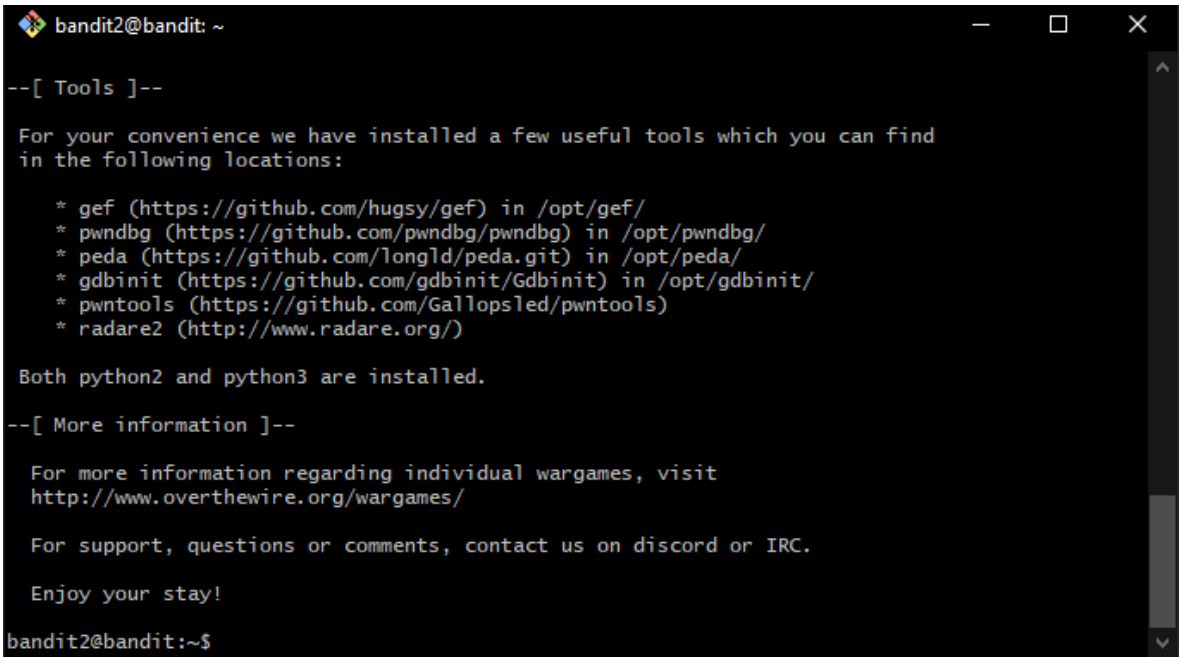
bandit 2

Para este ejercisio volveremos a usar el comando cat, sin embargo el archivo tiene un nombre que inicia con el carácter especial “–“ por lo cual tendemos que hacer una modificación a como escribimos el fichero.

cat ./-

pass: rRGizSaX8Mk1RTb1CNQoXTcYZWU6lgzi

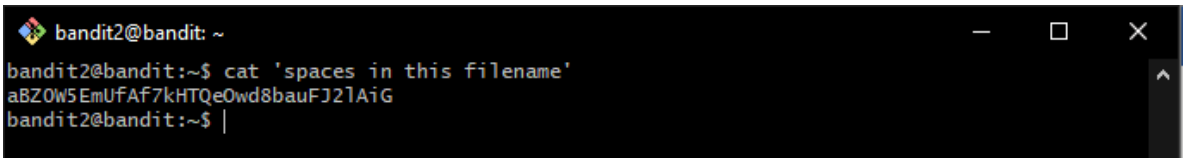
Únicamente resta desloguearse y loguearse con el nuevo usuario y cotraseña.



bandit 3

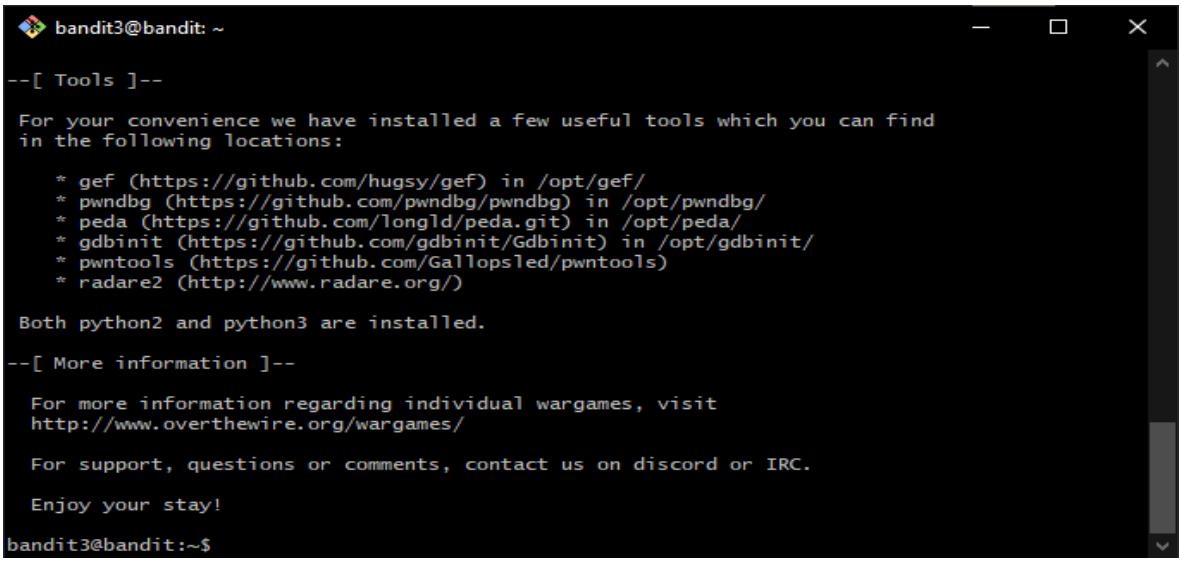
Para la siguente contraseña debemos acotar el nombre del archivo con comillas debido a que este contiene espacios.

cat 'spaces in this filename'



pass: aBZ0W5EmUfAf7kHTQeOwd8bauFJ2lAiG

Nos desloguearse y loguearse con el nuevo usuario y cotraseña.



bandit 4

Para este ejercisio tendremos que cambiar de directorio con el comando cd, luego con el comando ls con el modificador -a podremos verificar los archivos de manera detallada. Una vez confirmado podemos leer el archivo .hidden.

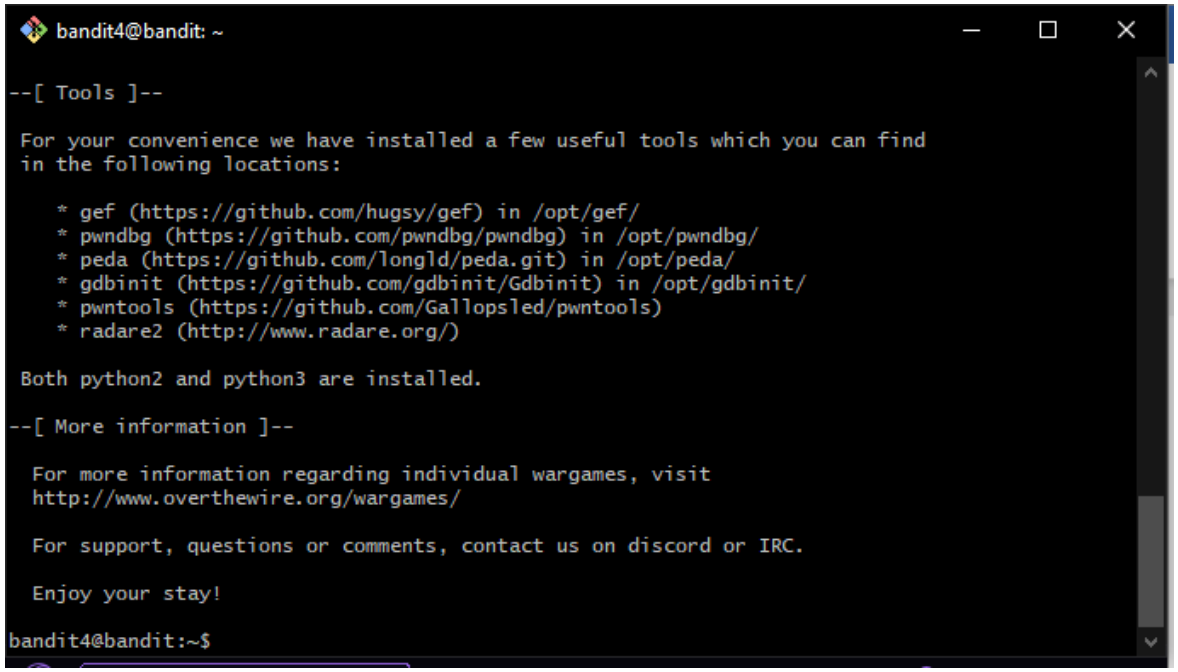
cd inhere

ls -a

cat .hiden

pass 2EW7BBsr6aMMoJ2HjW067dm8EgX26xNe

Nos desloguearse y loguearse con el nuevo usuario y cotraseña.



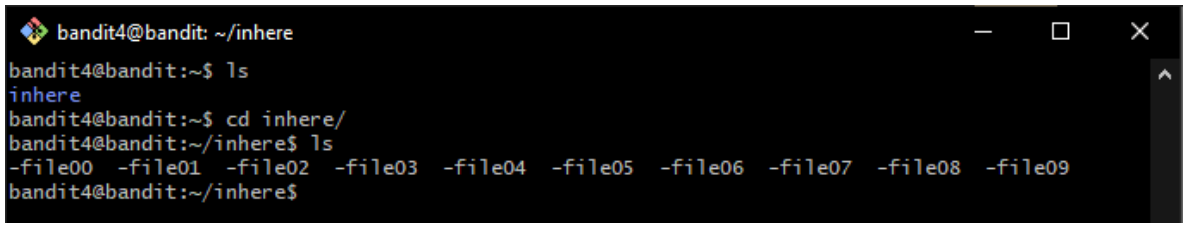
Bandit5

Para este ejercicio si confirmamos los ficheros en el directorio home con el comando ls notaremos el directorio inhere, si entramos con el comado cd y ejecutamos un ls notaremos que habrá varios archivos disponibles.

ls

cd inhere/

ls



En lugar de revisarlos uno por uno lo haremos todos de golpe regresando al directorio home y utilizando el comando file, para que revise todos los archivos agregaremos un /* al final del directorio.

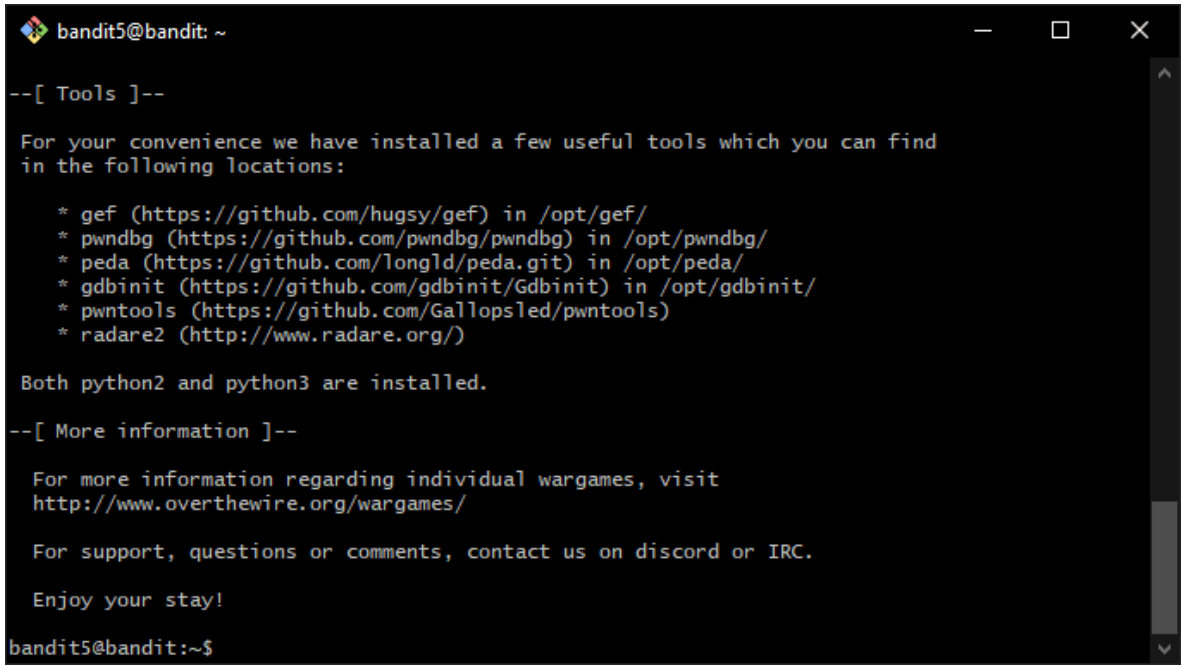
cd

file inhere/*

Observamos que el unico archivo text es el archivo -file07 por lo que al leerlo con un cat podremos obtener la contraseña.

Pass lrIWWI6bB37kxfiCQZqUdOIYfr6eEeqR

Comprobamos.



Bandit6

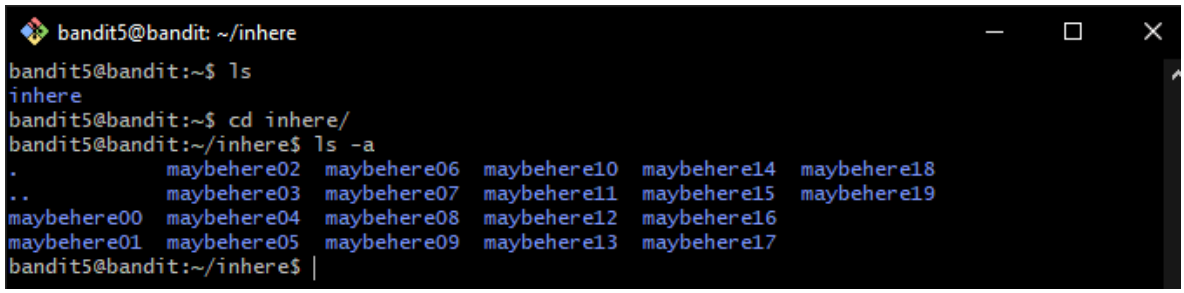
El ejercisio nos menciona que el archivo se encuentra el el directorio inhere y nos da ciertas caracteristicas del mismo por lo cual podremos ejecutar el comando find con algunos parametros de busqueda ya que si inspecionamos e directorio ingir notaremos varios supdirectorios por lo que revisarlos uno a uno tomaria demaciado tiempo.

ls

cd inhhere

ls -a

ls-h



```
bandit5@bandit:~/inhere$ ls -l
total 80
drwxr-x--- 2 root bandit5 4096 Jan 11 19:19 maybehere00
drwxr-x--- 2 root bandit5 4096 Jan 11 19:19 maybehere01
drwxr-x--- 2 root bandit5 4096 Jan 11 19:19 maybehere02
drwxr-x--- 2 root bandit5 4096 Jan 11 19:19 maybehere03
drwxr-x--- 2 root bandit5 4096 Jan 11 19:19 maybehere04
drwxr-x--- 2 root bandit5 4096 Jan 11 19:19 maybehere05
drwxr-x--- 2 root bandit5 4096 Jan 11 19:19 maybehere06
drwxr-x--- 2 root bandit5 4096 Jan 11 19:19 maybehere07
drwxr-x--- 2 root bandit5 4096 Jan 11 19:19 maybehere08
drwxr-x--- 2 root bandit5 4096 Jan 11 19:19 maybehere09
drwxr-x--- 2 root bandit5 4096 Jan 11 19:19 maybehere10
drwxr-x--- 2 root bandit5 4096 Jan 11 19:19 maybehere11
drwxr-x--- 2 root bandit5 4096 Jan 11 19:19 maybehere12
drwxr-x--- 2 root bandit5 4096 Jan 11 19:19 maybehere13
drwxr-x--- 2 root bandit5 4096 Jan 11 19:19 maybehere14
drwxr-x--- 2 root bandit5 4096 Jan 11 19:19 maybehere15
drwxr-x--- 2 root bandit5 4096 Jan 11 19:19 maybehere16
drwxr-x--- 2 root bandit5 4096 Jan 11 19:19 maybehere17
drwxr-x--- 2 root bandit5 4096 Jan 11 19:19 maybehere18
drwxr-x--- 2 root bandit5 4096 Jan 11 19:19 maybehere19
bandit5@bandit:~/inhere$
```

Para usar el comando find le indicaremos con “.” Que la busqueda iniciara desde el directorio actual, “-type f” para indicar que lo que queremos es tipo archivo, “-readable” que sea leible. “! -executable” que no sea ejecutable y por ultimo su tamaño con “-size 1033c”.

find . -type f -readable ! -executable -size 1033c

```
bandit5@bandit: ~/inhere
bandit5@bandit:~/inhere$ find . -type f -readable ! -executable -size 1033c
./maybehere07/.file2
bandit5@bandit:~/inhere$
```

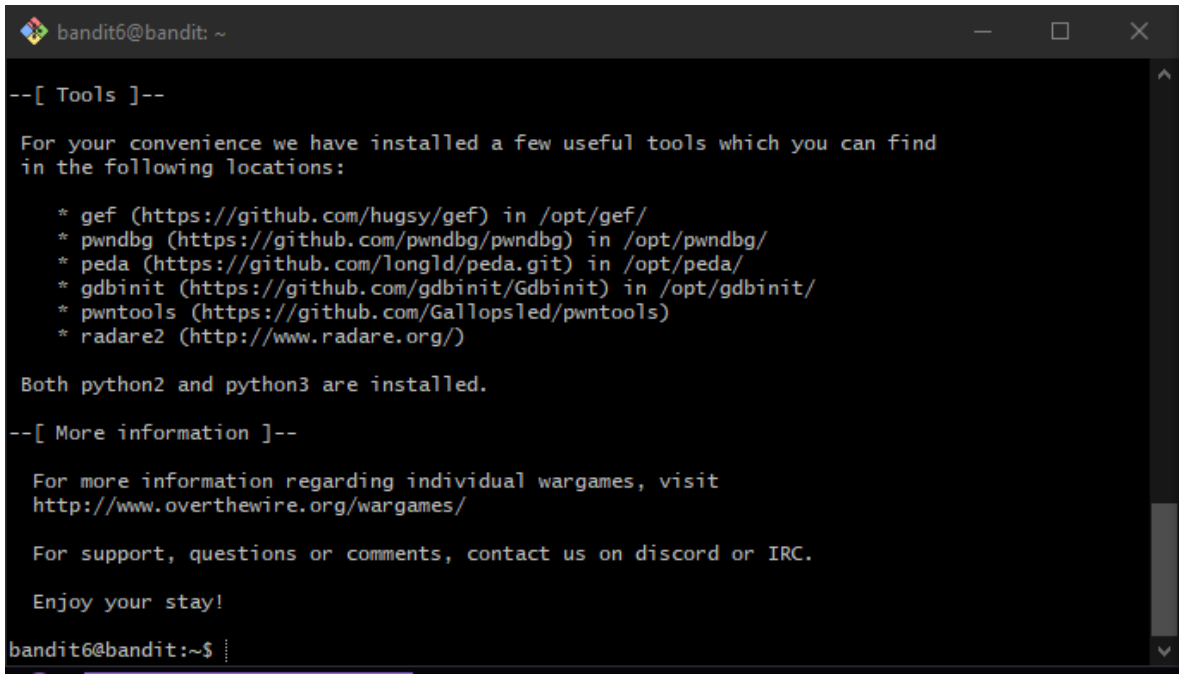
Ya que hay un unico archivo con las caracteristicas de la usqueda podemos realizar un cat para obtener la contraseña.

cat ./maybehere07/.file2

```
bandit5@bandit: ~/inhere
bandit5@bandit:~/inhere$ cat ./maybehere07/.file2
P4L4vucdmLnm8I7Vl7jG1ApGSfjYKqJU
bandit5@bandit:~/inhere$
```

Pass P4L4vucdmLnm8I7Vl7jG1ApGSfjYKqJU

Comprobamos



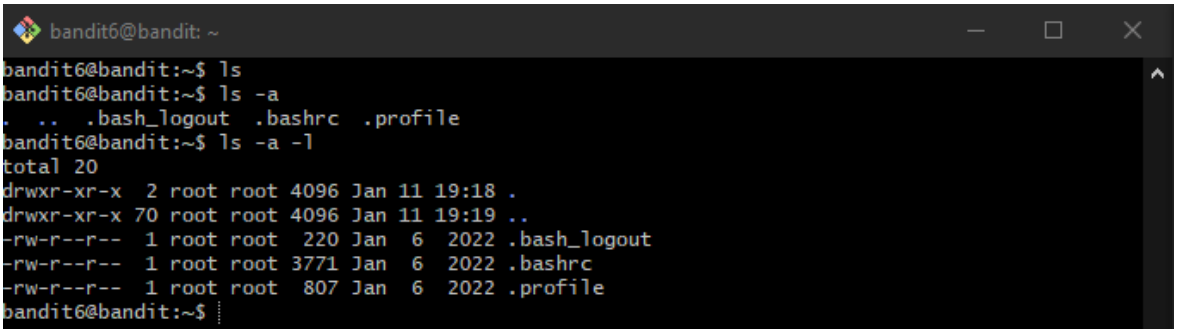
Bandit 7

Para este ejercicio nuevamente nos pide buscar la contraseña en algún lugar, esta vez del servidor, y nos da algunas características. Si por curiosidad ejecutamos un ls normal notaremos que no nos muestra ningún directorio solo hasta agregar el modificador -a encontraremos alguno.

Ls

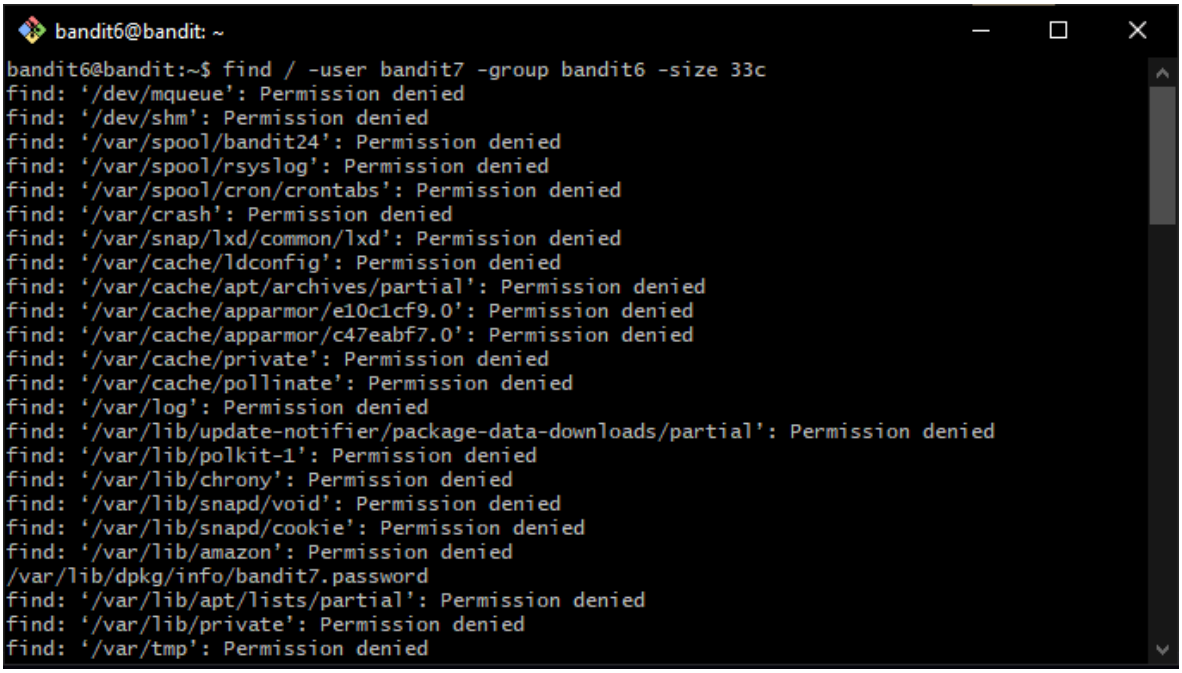
ls -a

ls -a -l

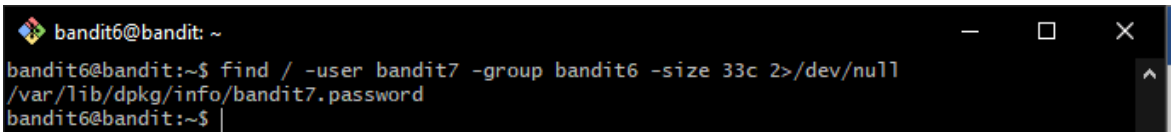


Así que esta vez probaremos realizar una busqueda en todo el servidor usado “/” en lugar de “.” Para buscar desde la raiz.

find / -user bandit7 -group bandit6 -size 33c

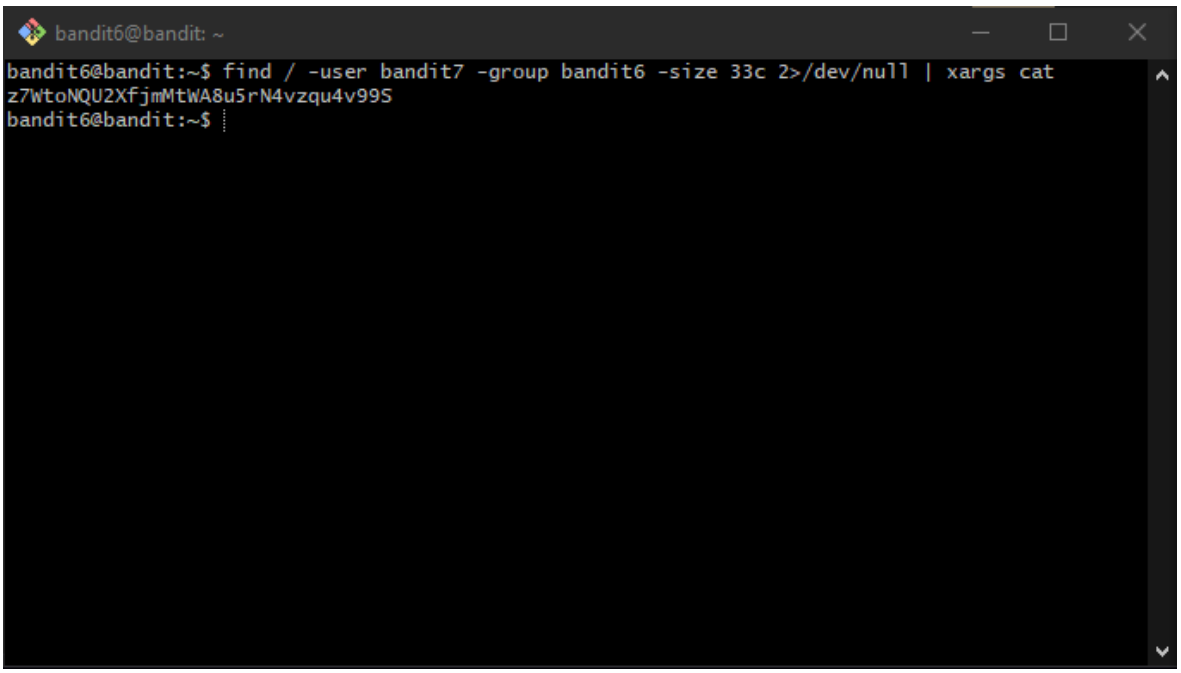


Notaremos varios resultados que nos dieron error por tener permisos denegadis y uno nombrado .password que podemos suponer es la contraseña, pero para fines mas instructivos puleremos mas este comando para tener un resultado mas legible en caso de necesitarlo. Para ello primero indicaremos que todos los errores (representados con un 2 en los tipos de salidas) sean dirigidos (con el carácter mayor que “>”)a vacio “null” añadiendo al comando “2>/dev/null”



Si quisieramos ser aun mas concretos podriamos indicar que esta salida sea la entrada de otro comando usando |, y utilizando xargs para entrada de entrada estándar en argumentos a un comando, en este caso el comando cat

find / -user bandit7 -group bandit6 -size 33c 2>/dev/null | xargs cat



Pass z7WtoNQU2XfjmMtWA8u5rN4vzqu4v99S

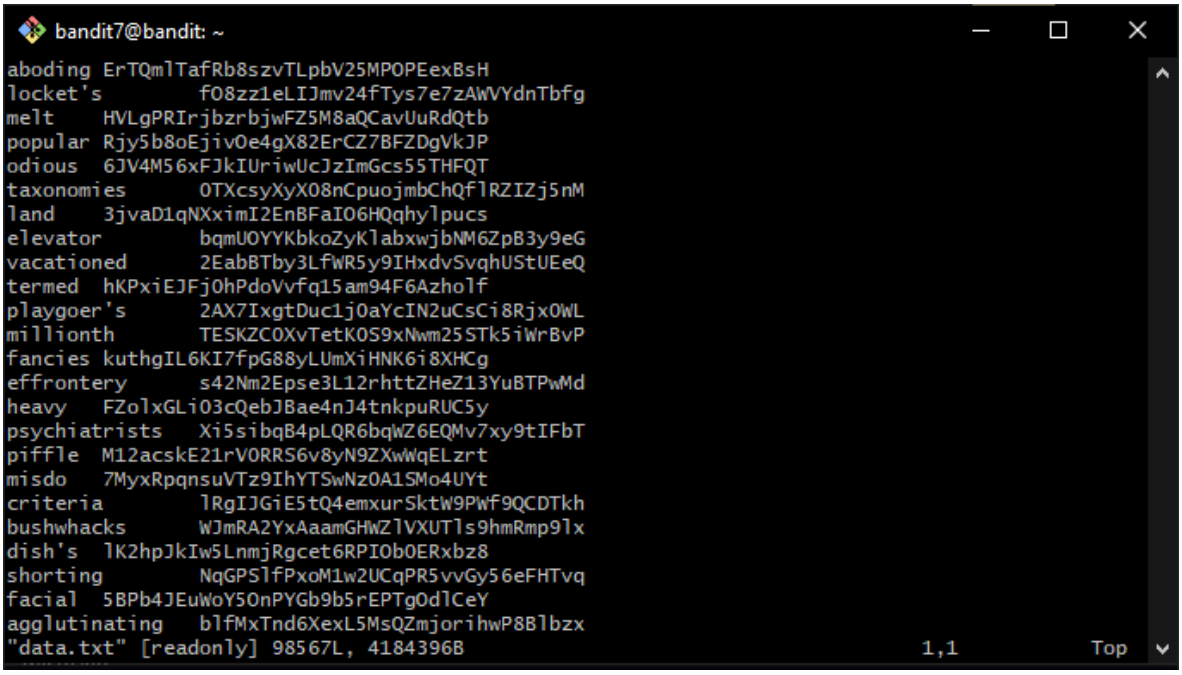
Comprobamos.

Bandit 8

En el siguiente ejercicio se nos indica que la contraseña esta en junto a la palabra “millionth” archivo data.txt , si en la raiz del server ejecutamos un ls veremos el archivo mencionado y al abrirlo con vi notaremos un archivo con muchas lineas de texto parecidas a una contraseña.

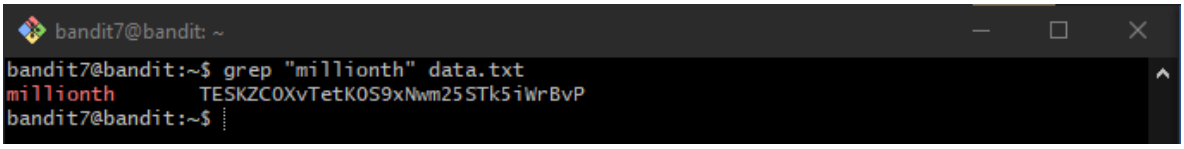
ls

Vi data.tx



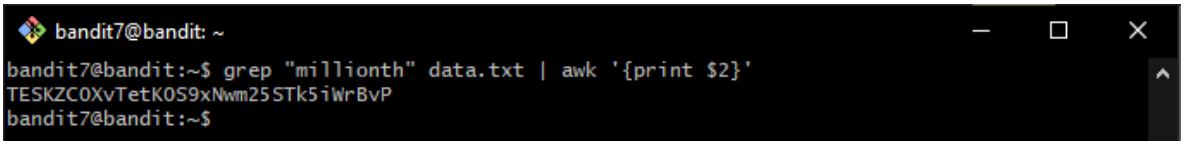
Utilizaremos el comando grep para buscar la linea que contenga la palabra “millionth” en el archivo

grep millionth



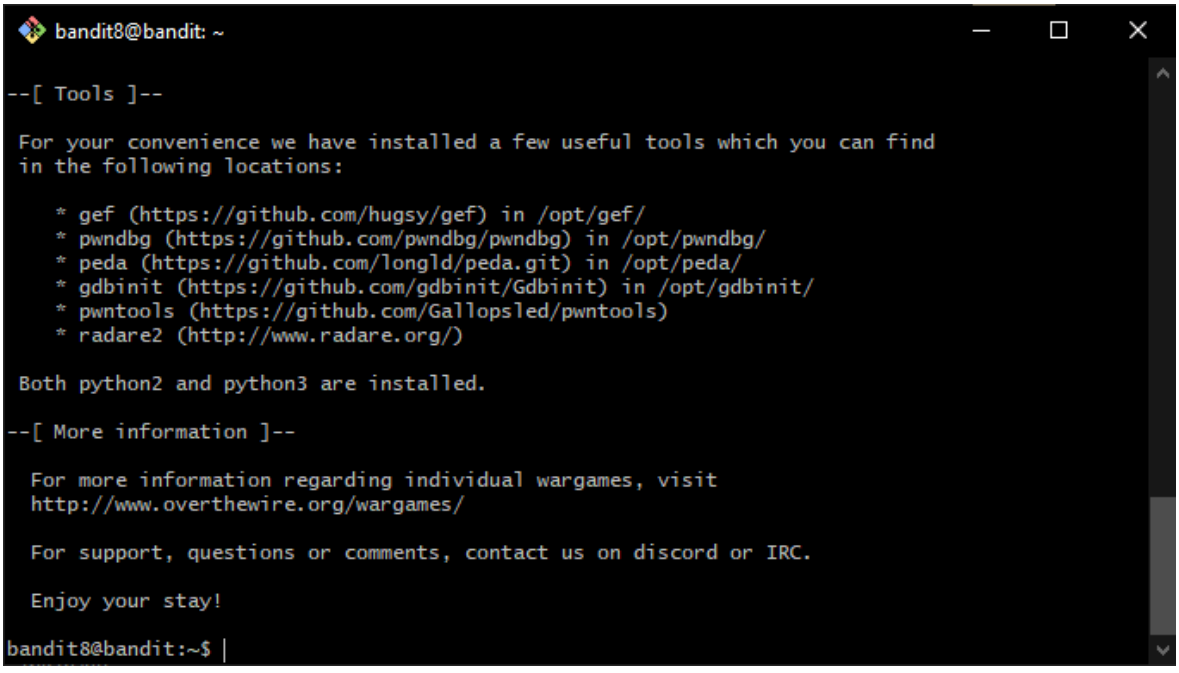
Para depurar más esta salida podemos utilisarla como entrada para el comando awk, indicando que nos devuelva unicamente el segundo argumento

grep "millionth" data.txt | awk '{print $2}'



Pass TESKZC0XvTetK0S9xNwm25STk5iWrBvP

Comprobamos.

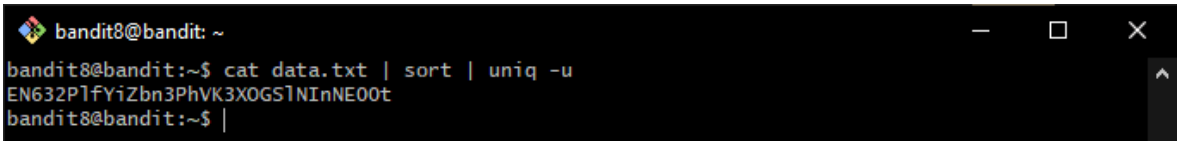


Badit9

En este ejercicio de nuevo se nos indica que la contraseña esta en un archivo y que es la unica que aparece solo una vez. Para descrubir cual es esta linea podemos utilizar el comando uniq con el modificador -u, sin embargo este comando nos pide una entrada ya

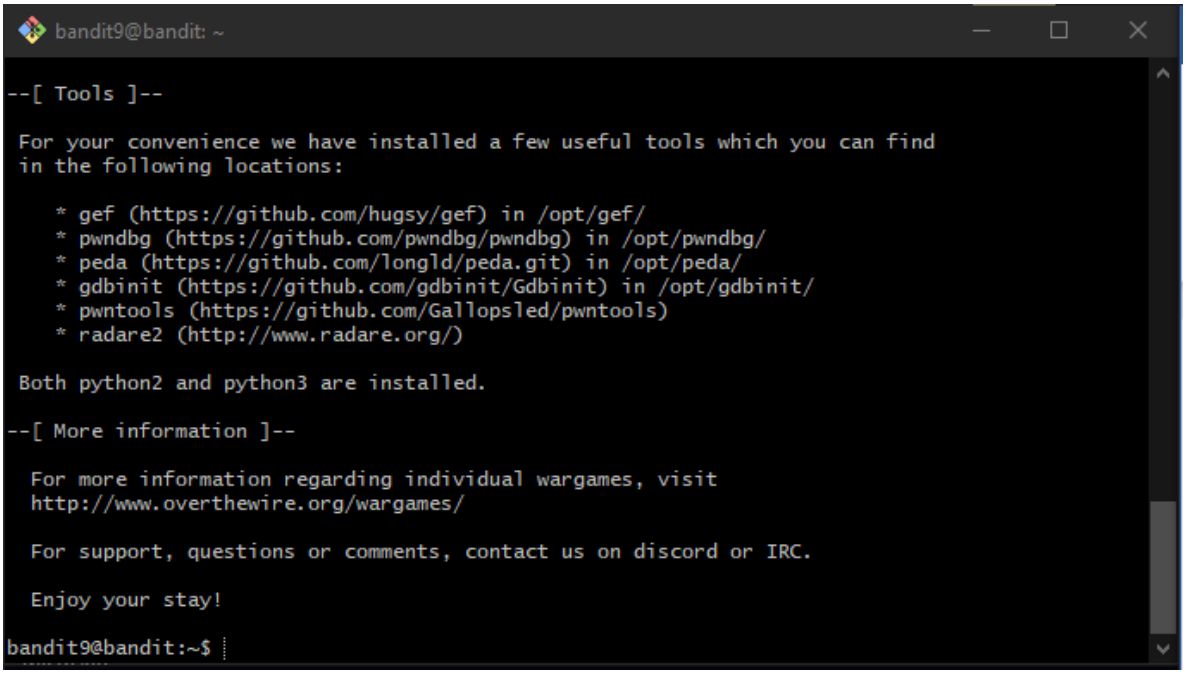
ordenada por lo cual primero debemos prosesar el archivo, en este caso con el comando sort que nos ordenara las lindeas del archivo por orden alfabetico.

cat data.txt | sort | uniq -u



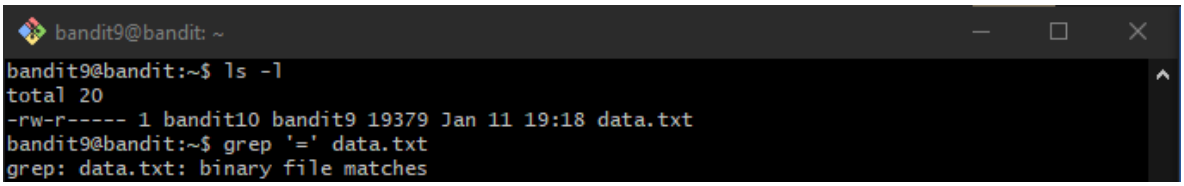
Pass EN632PlfYiZbn3PhVK3XOGSlNInNE00t

Comprobamos.



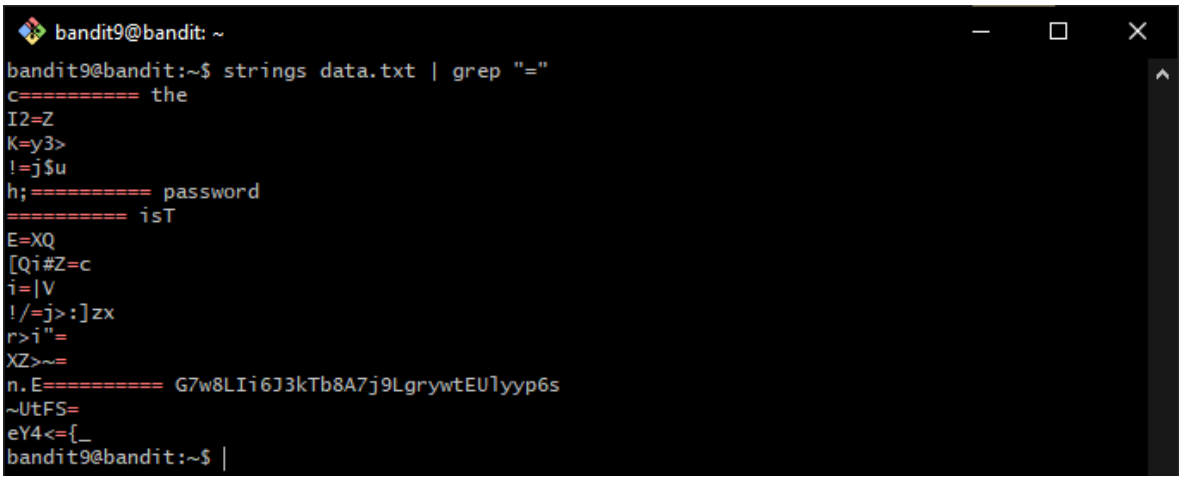
Bandit10

En este ejercisio se nos indica que la contraseña esta presedida por el carácter “=”. Sin embargo si utilizamos el comando grep como en el ejercio anterior, nos indicara que hay coincidencias en archivo bynario, por lo que no podremos verificar la contraseña.



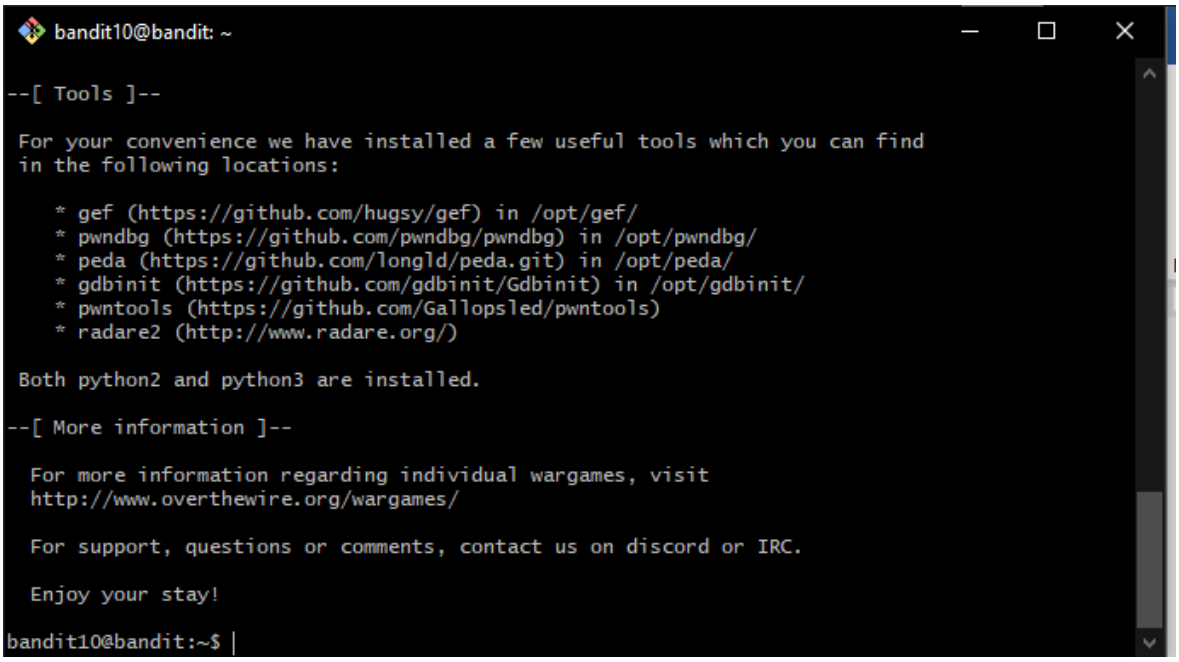
Para resolver esto utilzaremos el comando strings el cual extrae cadenas de caracteres imprimibles para poderlo usar como entrada de grep. Con lo que se nos muestra las lineas que ontengan el carácter pedido y obserbamos un mensaje que nos indica la contrase “the password ist”

strings data.txt | grep "="



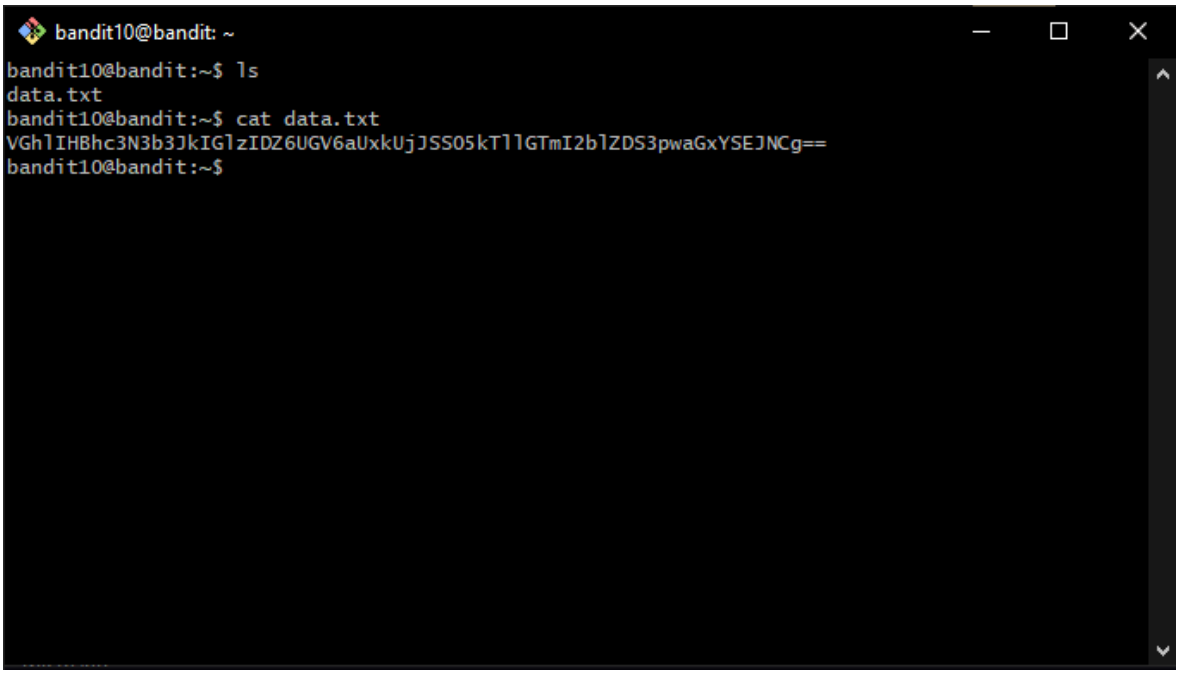
pass G7w8LIi6J3kTb8A7j9LgrywtEUlyyp6s

Comprobamos.



Bandit11

En este ejercisio se nos indica que el archivo esta codificado en base 64. Si ejecutamos un cat al unico archivo visible en el servidor obtendremos una linea que paresiera una contraseña sin embargo hemos visto que estas son mas cortas.

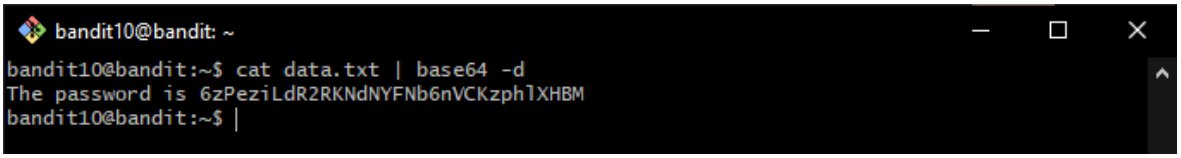


Podrimos decodificarlo manualmente con la tabla qe semuestra a continuacion (extraida de Wikipedia.org en el articulo pertinente). Tras obtener el valor en binario de cada carácter y posteriormente transformando el resultado en conjunto a codigo ascii.

Valor	Carácter	Valor	Carácter	Valor	Carácter	Valor	Carácter
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

O bien podemos usar el comando base64 de la consola de linux, brindandole como entrada el resultado de la lectura del archivo y añadiendo el modificador -d para que haga una decodificacion.

cat data.txt |base64 -d

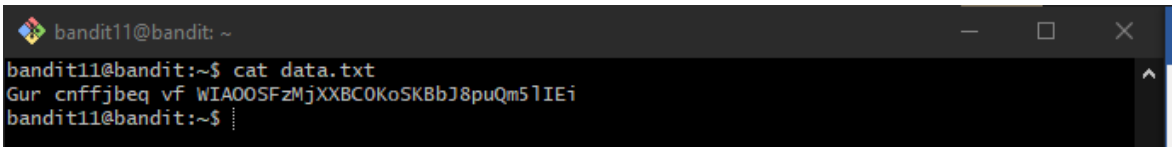


Pass 6zPeziLdR2RKNdNYFNb6nVCKzphlXHBM

Comprobamos.

Bandit12

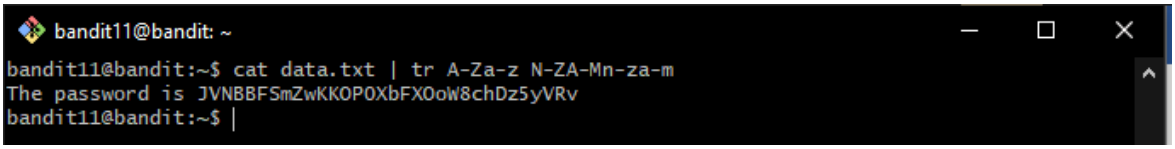
En este ejercicio nos indica que para obtener la contraseña debemos rotar todas las letras del abesedario en mayusculas y en minusculas en 13 posiciones. Si rebisamos el contenido del archivo data.txt con un cat notaremos una unica linea de caracteres.



Esta linea la podemos altrar con el comando tr, al cual le indicaremos que el conjunto de letras A-Z y a-z sean intercambiadas por su homologo trese posiciones despues, si contamos desde la posicion de la letra A la posicion 13+1 nos daria la letra N, por lo que el nuevo orden de letras seria de la N-Z seguido por la A-M. Por lo que nuestra linea de comando quedaria :

cat data.txt | tr A-Za-z N-ZA-Mn-za-m

pass JVNBBFSmZwKKOP0XbFXOoW8chDz5yVRv



Comprobamos

Bandit13

Para este ejercisio se nos recomienda primero crear un directorio donde poder trabajar con una copia del archivo data.txt-

mkdir /tmp/roberto

cp data.txt /tmp/roberto

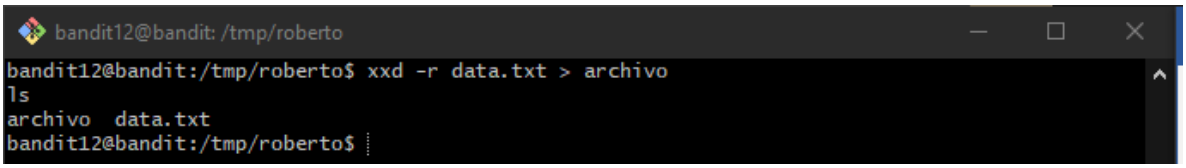
cd /tmp/roberto

Una vez echo esto, el ejerisio indica que el arcivo data.txt es un “hexdump” o volcado de datos en hexadesimal, por lo que lo siguiente que hay que hacer es revetir esta operación.

Para ello nos podemos valer del mismo comando con el ual se realiza “xxd” con el modificador -d. Tambien se nos indica que este volcado se hizo de un archivo comprimido por lo cual revertiremos este proseso hacia un archivo .gz

xxd -r data.txt > archivo

ls

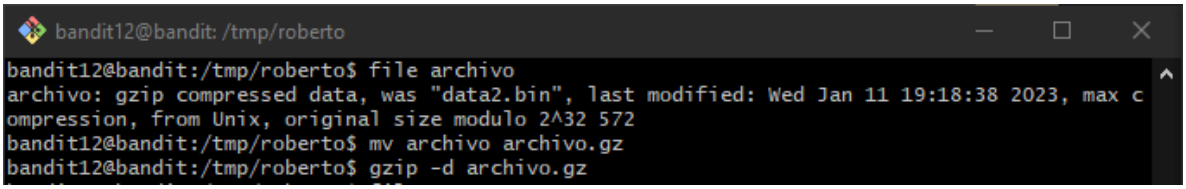


Resta seguir una serie de descompreciones hasta hallar el archivo que tenga la contraseña. Para saber cual comando usaremos para dicha descomprecion, primero debemos averiguar que tipo de rchivo enemos con el comando file, luego de ello le cambiaremos el nombre para agregar la extencion correspondiente y usar el comando adecuado (tar para .tar, gzip para .gz, bzip2 para .bz2).

file archivo

mv archivo archivo.gz

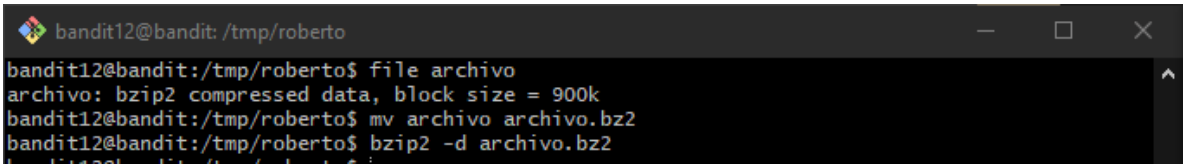
gzip -d archivo.gz



file archivo

mv archivo archivo.bz2

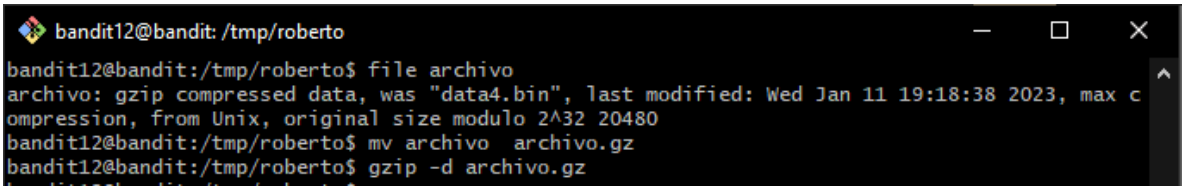
bzip2 -d archivo.bz2



file archivo

mv archivo archivo.gz

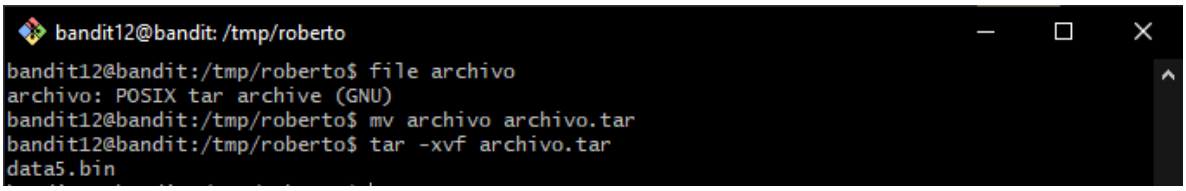
gzip -d archivo.gz



file archivo

mv archivo archivo.tar

tar -xvf archivo.tar

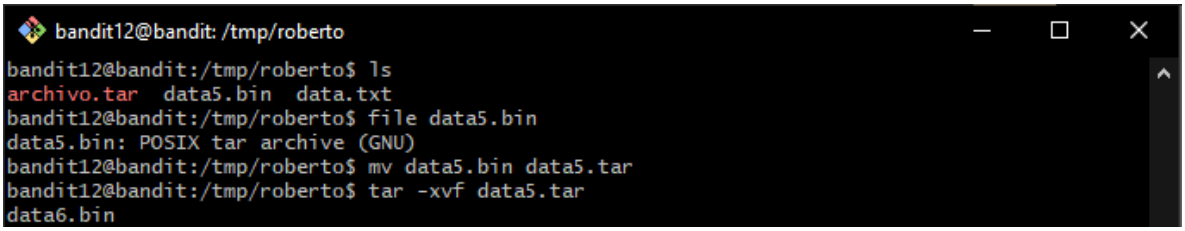


ls

file data5.bin

mv data5.bin data5.tar

tar -xvf data5.tar



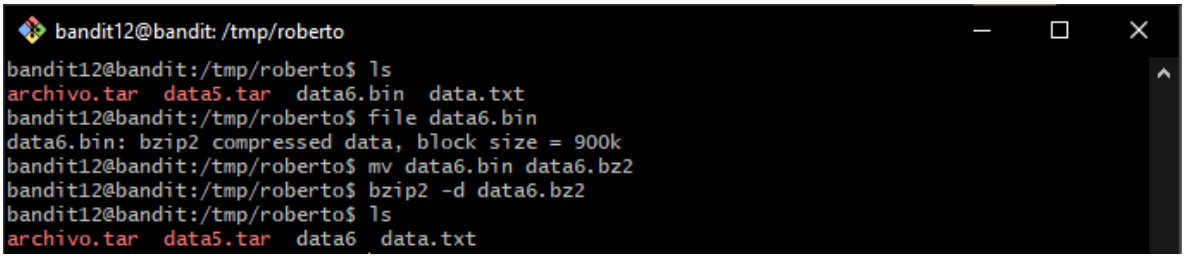
ls

file data6.bin

mv data6.bin data6.bz2

bzip2 -d data5.bz2

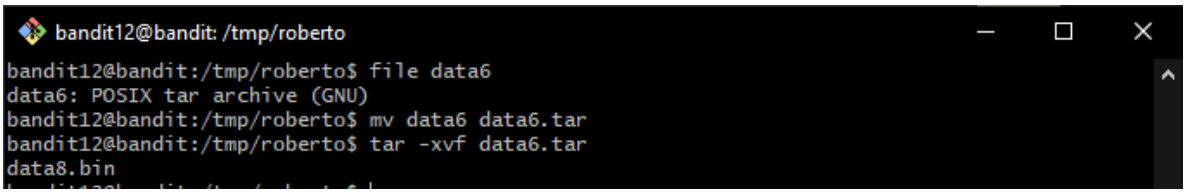
ls



file data6

mv data6 data6.tar

tar -xvf data6.tar



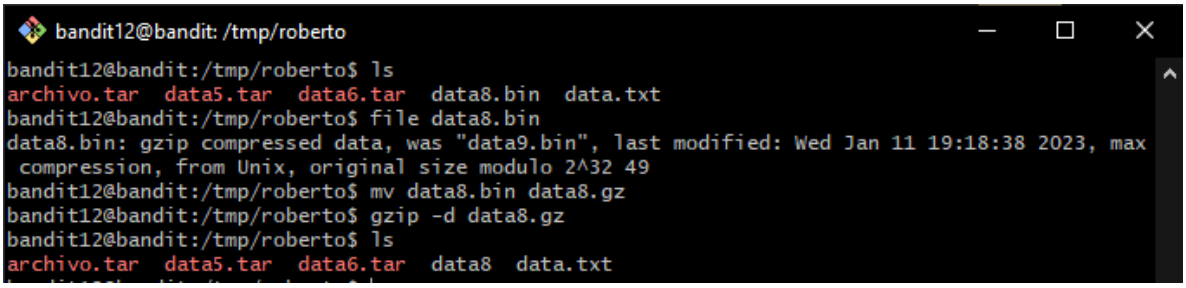
ls

file data8.bin

mv data8.bin data8.gz

gz -d data8.gz

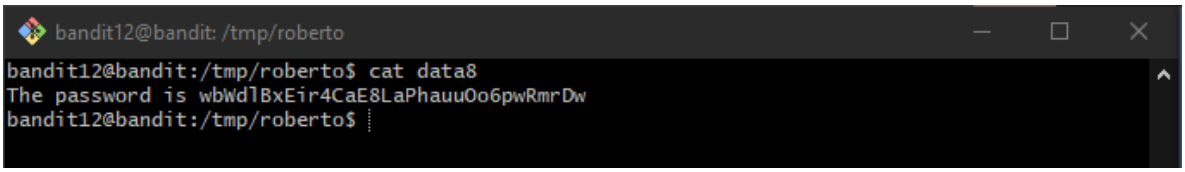
ls



file data8

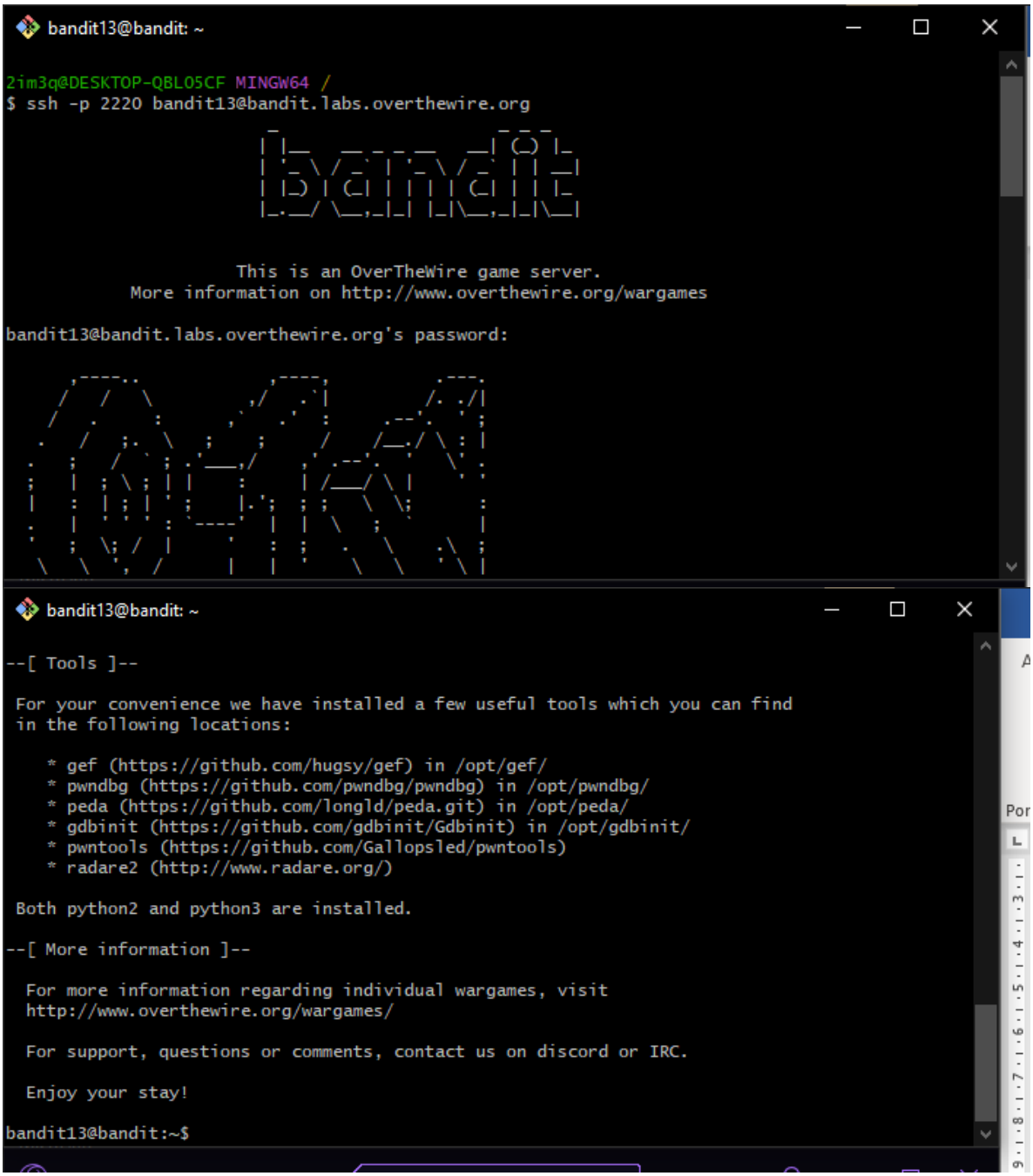
Llegamos a un archivo de texto por lo que podemos usar un cat para ver su contenido

Cat data8



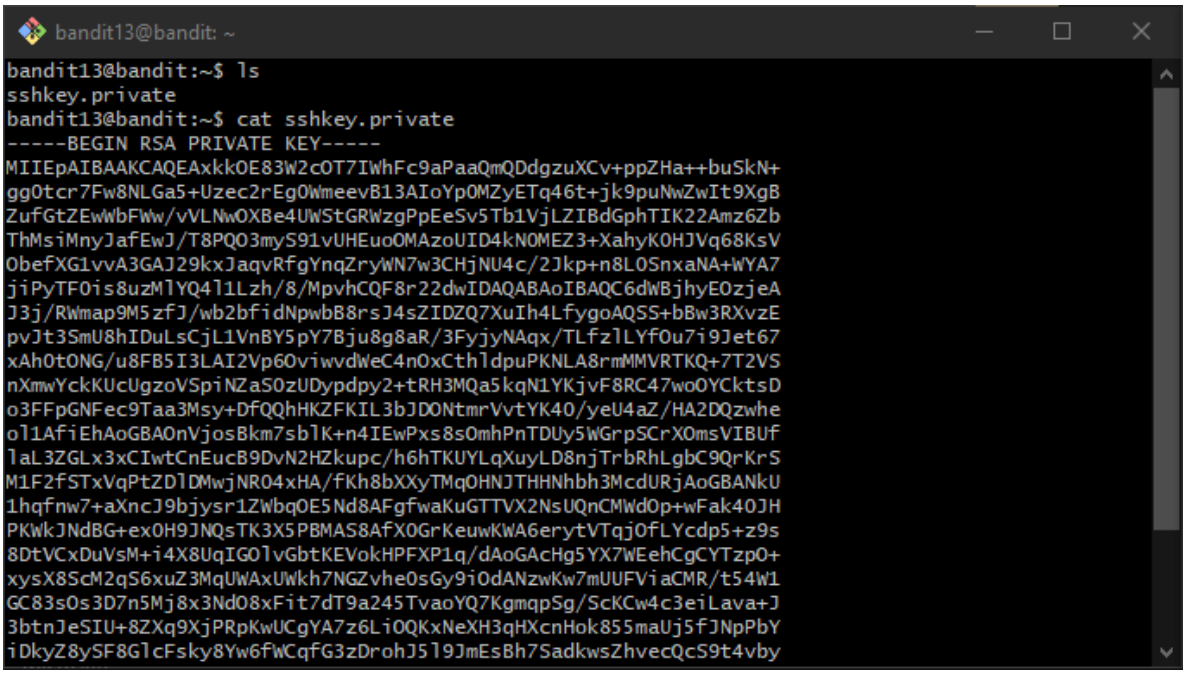
Pass wbWdlBxEir4CaE8LaPhauuOo6pwRmrDw

Comprobamos.



Badit14

Para el siguiente ejercisio se nos otorga una llave privada con la cual podemos accsesar al como el usuario bandid14 puesto a que este es el unico que puede leer el archivo que contiene la contraseña del nivel.



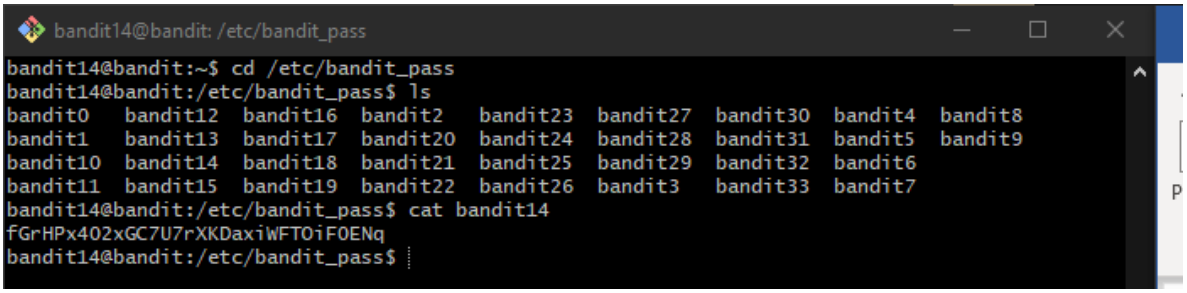
Para usarla nos podemos valer del comando ssh con el modificador -i

ssh -i sshkey.private bandit14@localhost -p 2220

En esta ocacion se uso localhoust en lugar de bandit para el servidor, debido a que ya estabamos conectados al mismo.

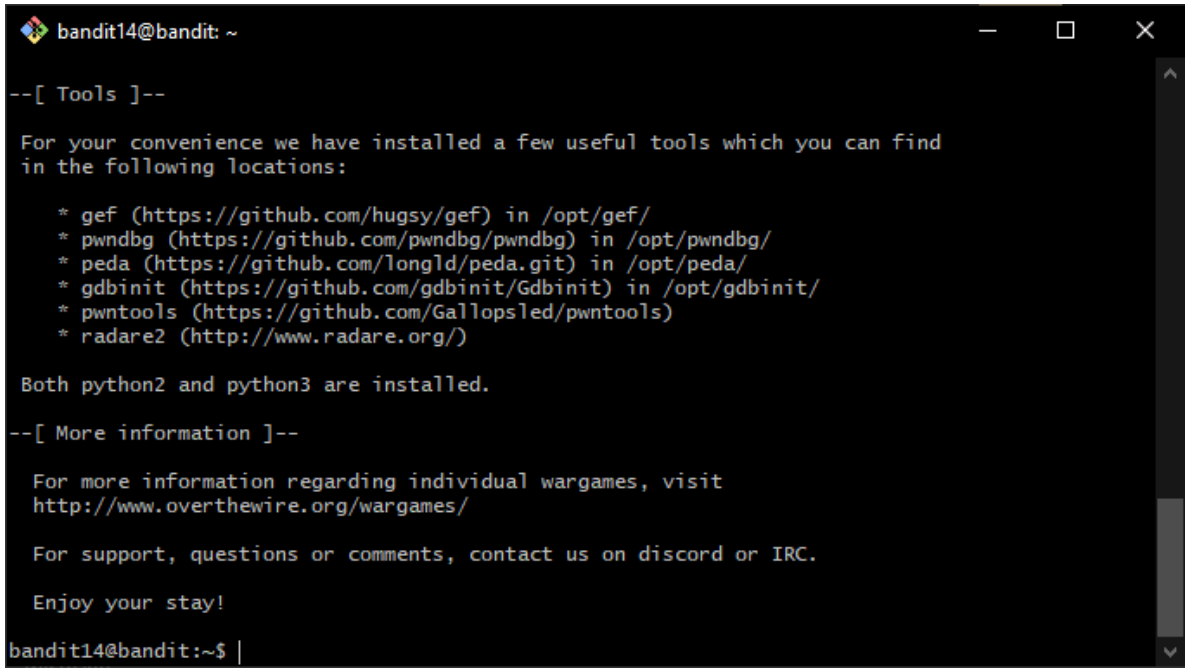
Una vez conectados como el usuario bandid 14 procedemos a dirigirnos al directorio indicado por el ejercisio, para podeer leer el archivo que contiene la contraseña.

cd /etc/bandit_pass/



Pass fGrHPx402xGC7U7rXKDaxiWFTOiF0ENq

Comprobamos.



Bandit15

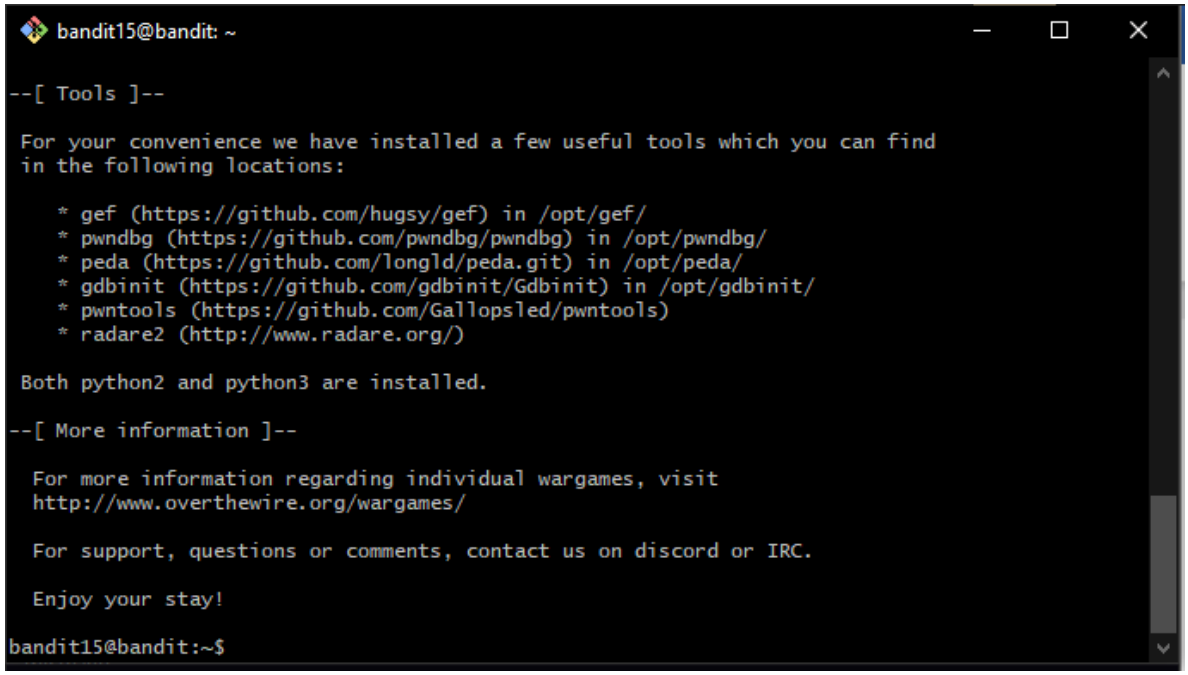
Para este ejercisio nos pide que la contraseña del ejercisio anterior se la enviemos por el puerto 30000 al servidor local. Por lo que primero nos moveremos al fichero indicado en el ejercisio anterior. Luego para enviar la informacion al puerto 30000 podemos usar el comando nc y redirecionando el arcivo bandid14.

cd /etc/bandit_pass/

nc localhost 30000 < bandit14

Pass jN2kgmIXJ6fShzhT2avhotn4Zcka6tnt

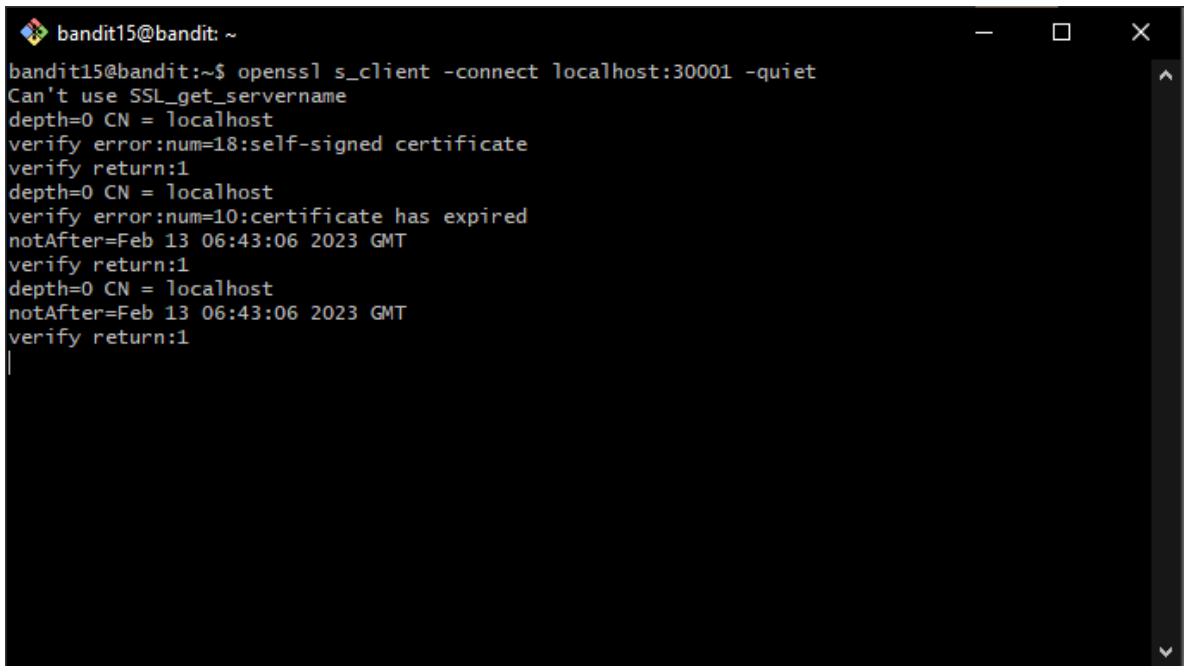
Comprobamos



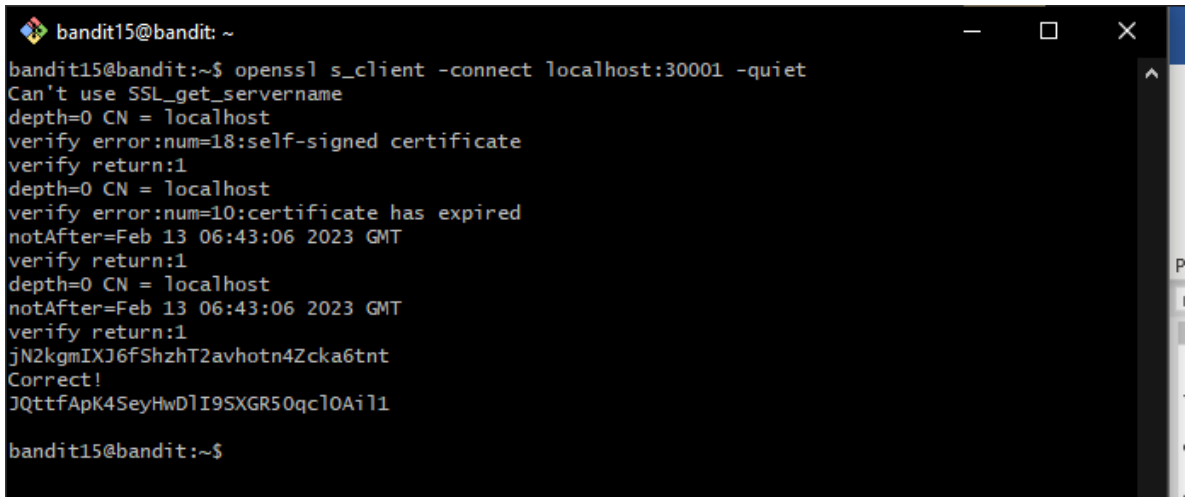
Bandit16

En este ejercisio se nos pide enviar el pasword de este nivel, encriptado en el protocolo ssl, el puerto 30000 Para ello usaremos el comando openssl con la herramienta s_client, para hacer conecciones usaremos la opcion -connect para espesificar el servidor y el puerto (30000 + 1) , y añadimos la opcion -quiet para tener una mejor lectura del resultado.

openssl s_client -connect localhost:30001 -quiet

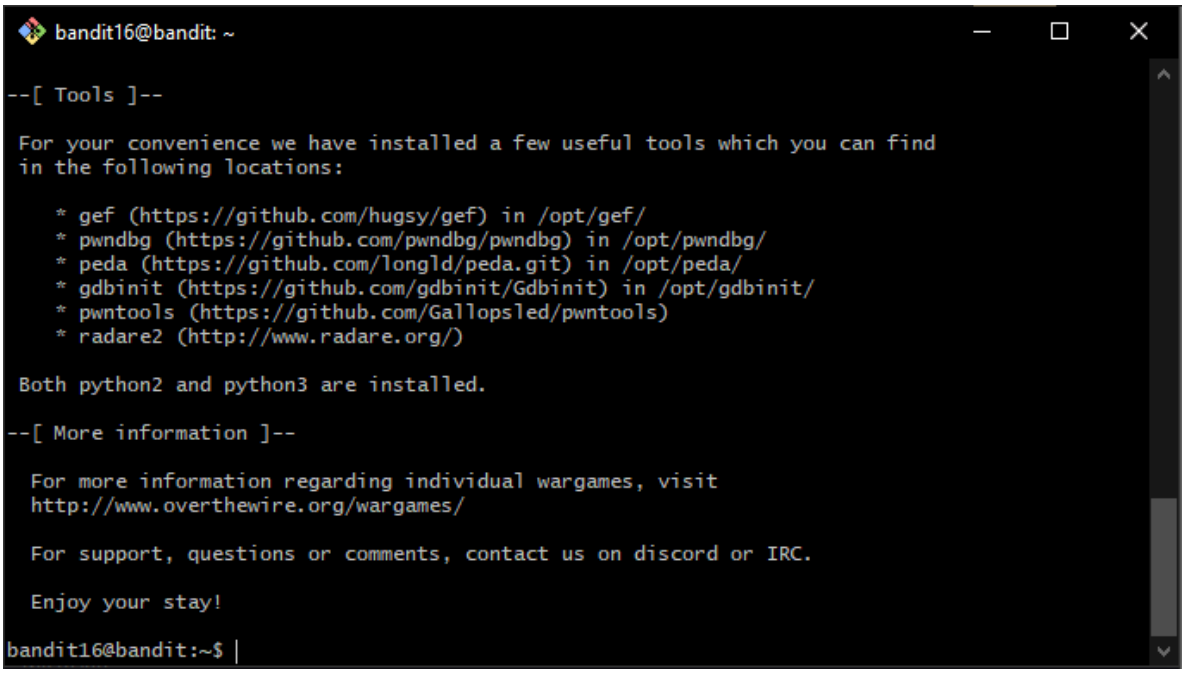


Aquí ingresaremos la contraseña del ejercisio



Pass `JQttfApK4SeyHwDlI9SXGR50qclOAil1`

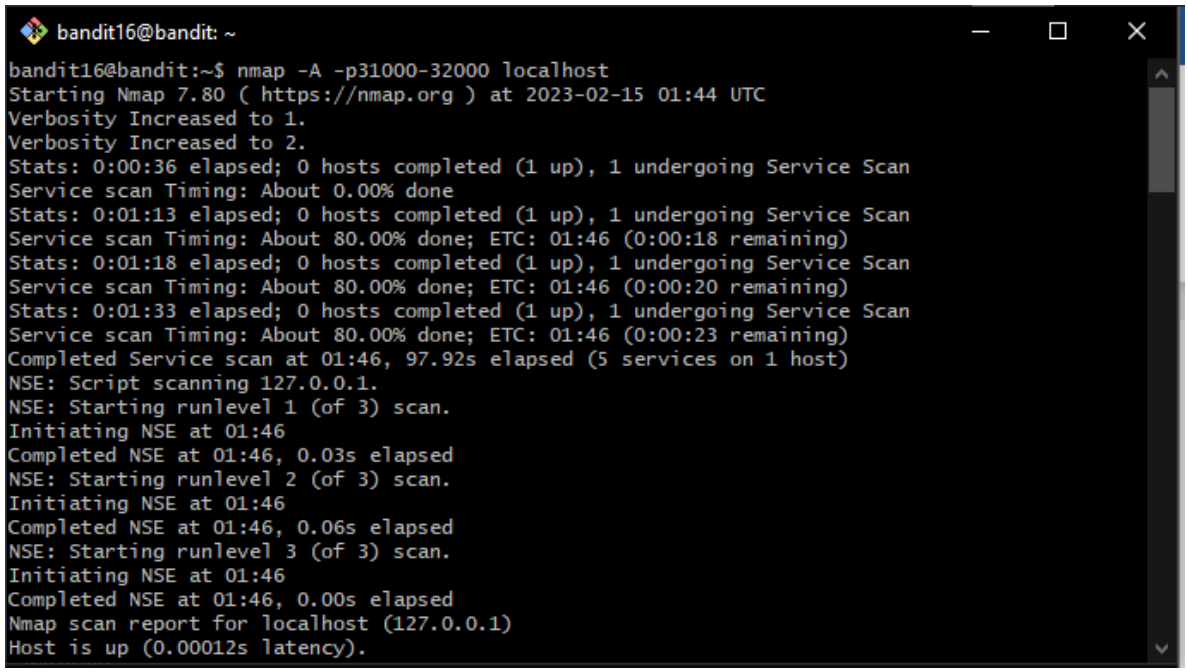
Comprobamos



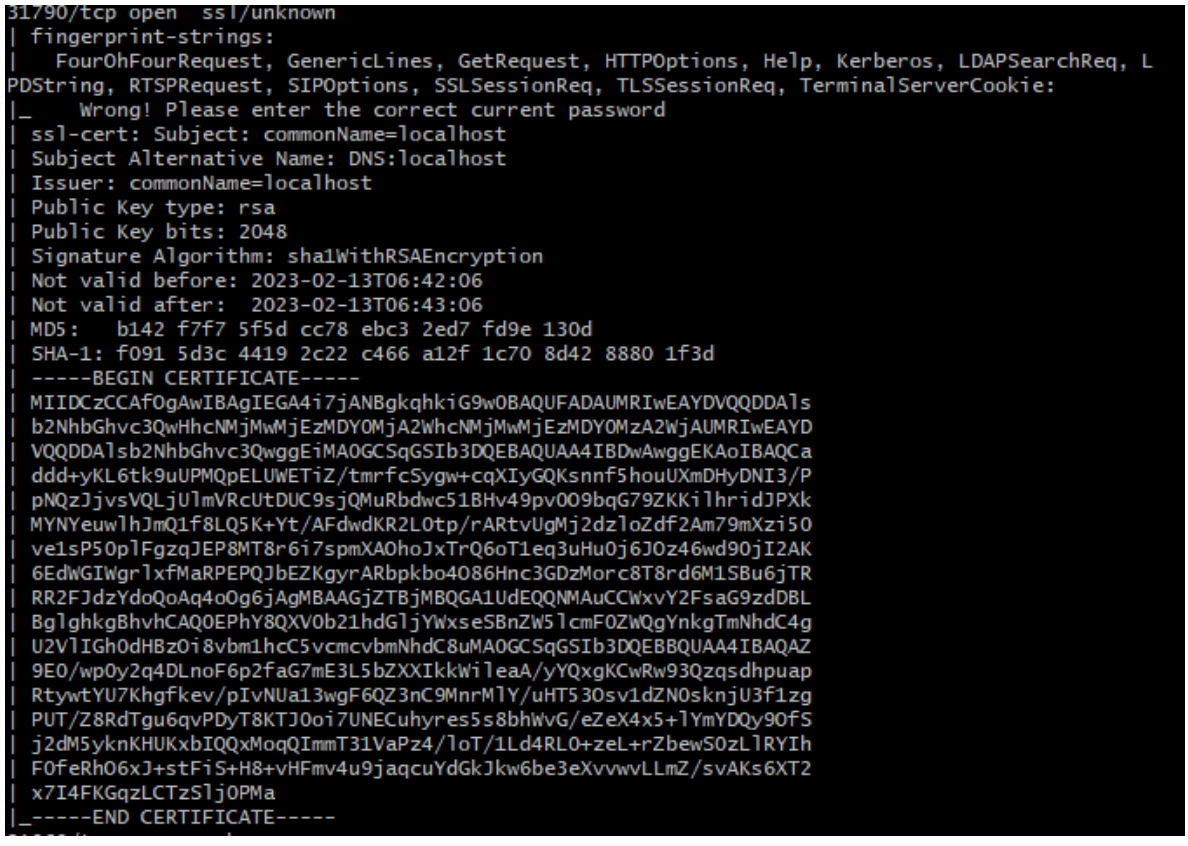
Bandit17

Este ejercisio se resuelve de manera similar al anterior, solo que esta vez no nos indican el puerto por el cual podemos enviar el mensaje, si no que nos dan un rango de puertos donde solo uno puede aceptar el mensaje en ssl. Para averiguar que puerto es nos valdremos de la herramienta nmap, la usaremos con la opcion -A para ejecutar un escaneo de fuerza bruta y posteriormente definiremos el rango de accion (para ver el porsentaje completado del escaneo podemos oprimer enter)

nmap -A -p31000 -p32000 localhost



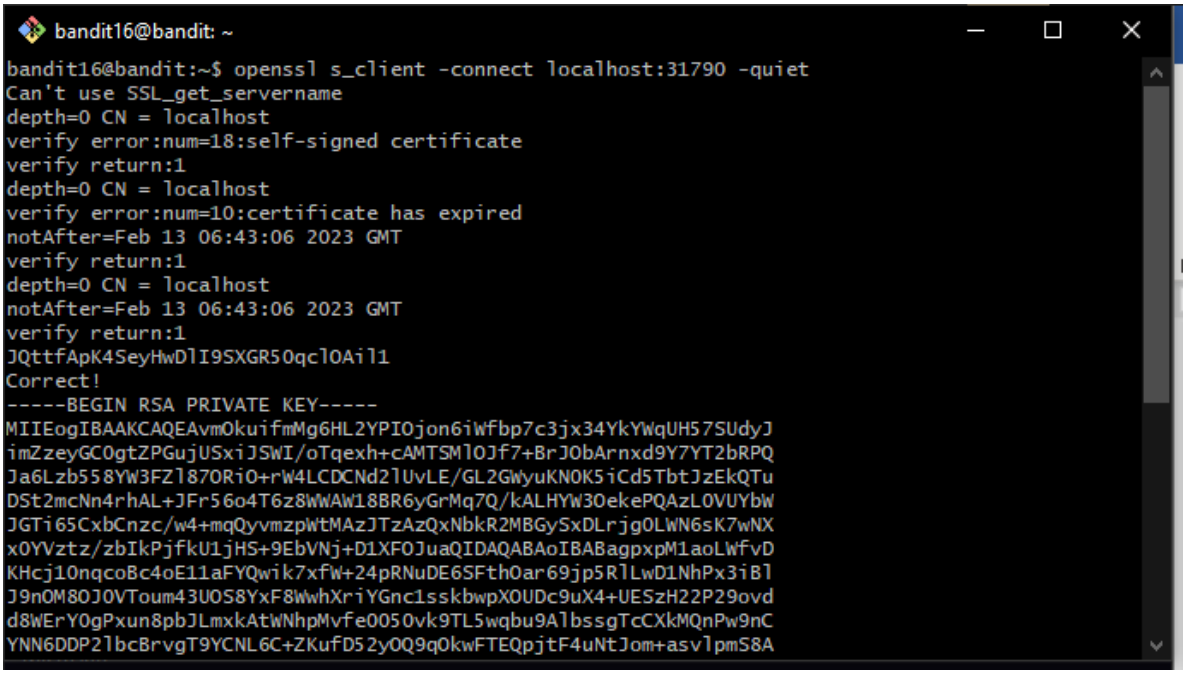
Si nos fijamos en el informe del puerto 31790 el cual es capas de leer ssl, nos fijaremos en un mensaje de error el cual nos pide qe ingresemos una contraseña



Probamos ingresar la contraseña a este puerto

openssl s_client -connect localhost:31790 -quie

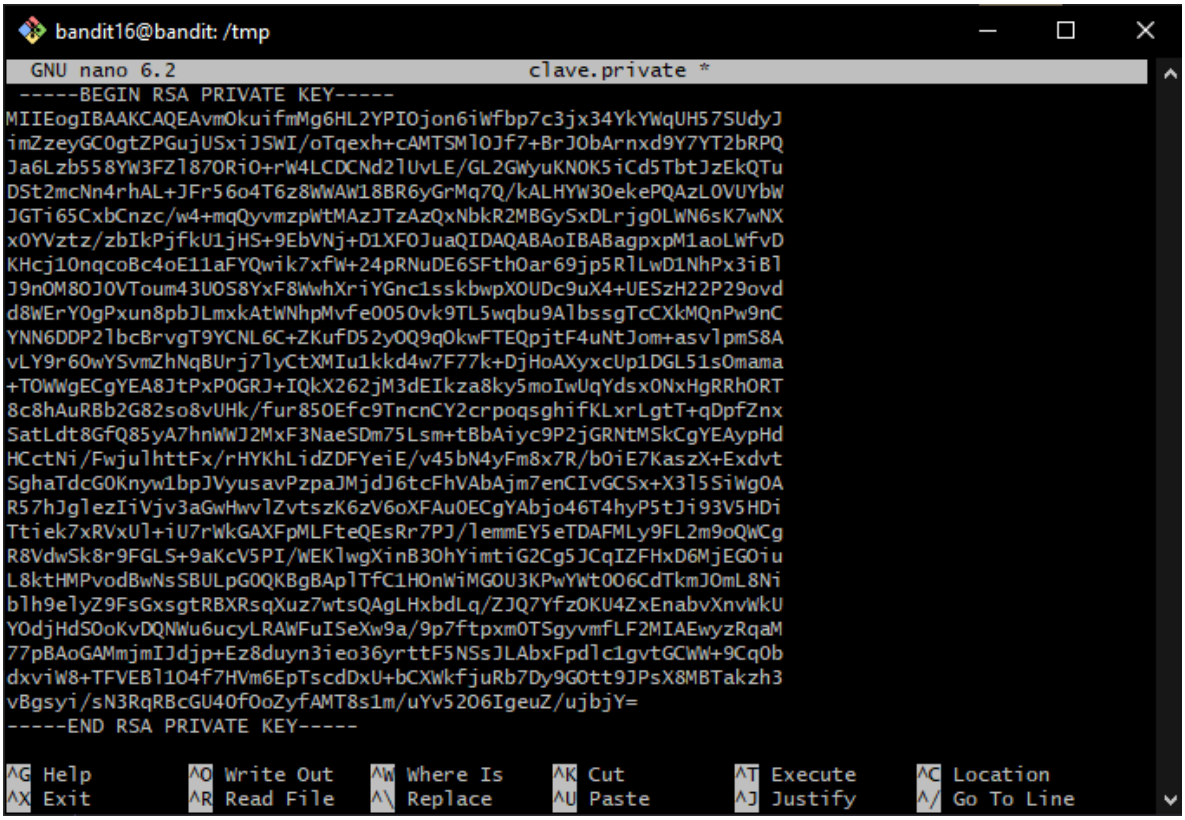
JQttfApK4SeyHwDlI9SXGR50qclOAil1

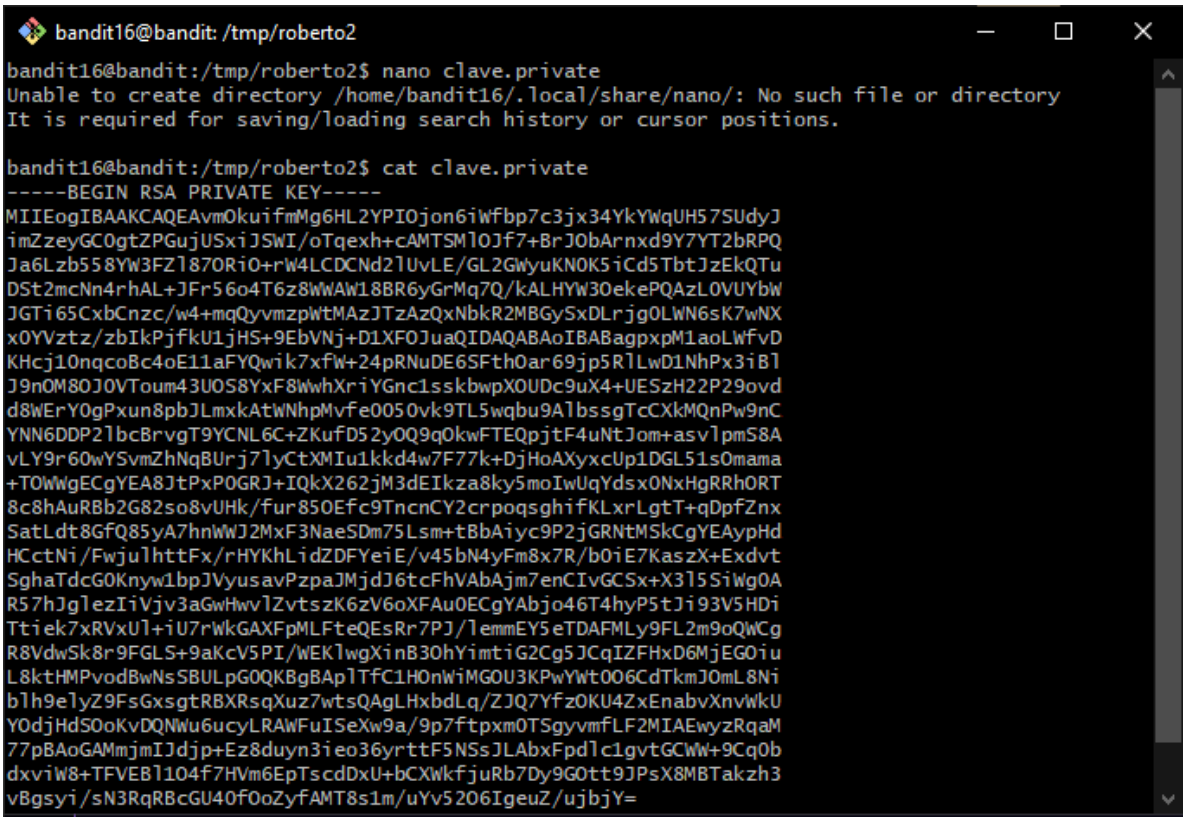



```
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAvmOkuifmMg6HL2YPIOjon6iWfbp7c3jx34YkYWqUH57SUdyJ
imZzeyGC0gtZPGujUSxiJSWI/oTqexh+cAMTSMlOJf7+BrJObArnxd9Y7YT2bRPQ
Ja6Lzb558YW3FZl87ORiO+rW4LCDCNd2lUvLE/GL2GWyuKN0K5iCd5TbtJzEkQTu
DSt2mcNn4rhAL+JFr56o4T6z8WWAW18BR6yGrMq7Q/kALHYW3OekePQAzLOVUYbW
JGTi65CxbCnzc/w4+mqQyvmzpWtMAzJTzAzQxNbkR2MBGySxDLrjgOLWN6sK7wNX
x0YVztz/zbIkPjfkU1jHS+9EbVNj+D1XFOJuaQIDAQABAoIBABagpxpM1aoLWfvD
```

```
KHcj10nqcoBc4oE11aFYQwik7xfW+24pRNuDE6SFthOar69jp5RlLwD1NhPx3iBl
J9nOM8OJOVToum43UOS8YxF8WwhXriYGnc1sskbwpXOUDc9uX4+UESzH22P29ovd
d8WErYOgPxun8pbJLmxkAtWNhpMvfeOO5Ovk9TL5wqbu9AlbssgTcCXkMQnPw9nC
YNN6DDP2lbcBrvgT9YCNL6C+ZKufD52yOQ9qOkwFTEQpjtF4uNtJom+asvlpmS8A
vLY9r6OwYSvmZhNqBUrj7lyCtXMIu1kkd4w7F77k+DjHoAXyxcUp1DGL51sOmama
+TOWWgECgYEA8JtPxPOGRJ+IQkX262jM3dEIkza8ky5moIwUqYdsxONxHgRRhORT
8c8hAuRBb2G82so8vUHk/fur85OEfc9TncnCY2crpoqsghifKLxrLgtT+qDpfZnx
SatLdt8GfQ85yA7hnWWJ2MxF3NaeSDm75Lsm+tBbAiyc9P2jGRNtMSkCgYEAypHd
HCctNi/FwjulhttFx/rHYKhLidZDFYeiE/v45bN4yFm8x7R/bOiE7KaszX+Exdvt
SghaTdcGOKnyw1bpJVyusavPzpaJMjdJ6tcFhVAbAjm7enCIvGCSx+X3l5SiWgOA
R57hJglezIiVjv3aGwHwvlZvtszK6zV6oXFAuOECgYAbjo46T4hyP5tJi93v5HDi
Ttiek7xRVxUl+iU7rWkGAXFpMLFteQEsRr7PJ/lemmEY5eTDAFMLy9FL2m9oQWCg
R8VdwSk8r9FGLS+9aKcV5PI/WEKlwgXinB3OhYimtiG2Cg5JCqIZFHxD6MjEGOiu
L8ktHMPvodBwNsSBULpGOQKBgBAplTfC1HOnWiMGOU3KPwYWtOO6CdTkmJOmL8Ni
blh9elyZ9FsGxsgtRBXRsqXuz7wtsQAgLHxbdLq/ZJQ7YfzOKU4ZxEnabvXnvWkU
YOdjHdSOoKvDQNWu6ucyLRAWFuISeXw9a/9p7ftpxm0TSgyvmfLF2MIAEwyzRqaM
77pBAoGAMmjmIJdjp+Ez8duyn3ieo36yrttF5NSsJLAbxFpdlc1gvtGCWW+9Cq0b
dxviW8+TFVEBl1O4f7HVm6EpTscdDxU+bCXWkfjuRb7Dy9GOtt9JPsX8MBTakzh3
vBgsyi/sN3RqRBcGU4OfOoZyfAMT8s1m/uYv52O6IgeuZ/ujbjY=
-----END RSA PRIVATE KEY-----
```

Observamos que nos otora una llave privada. Para usarla devemos crear un archivo tipo private, pero como no tenemos permisos de escritura en este directorio tendemos que hacerlo en el directorio tmp.

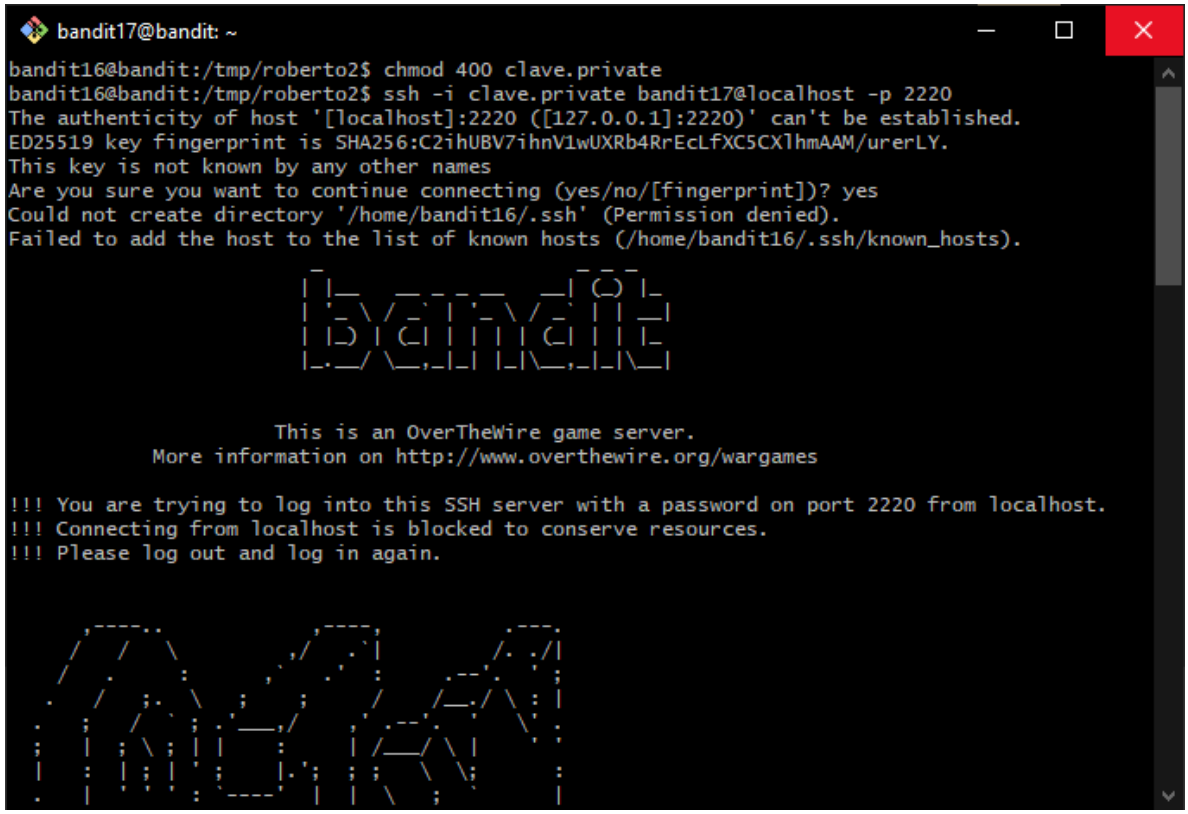


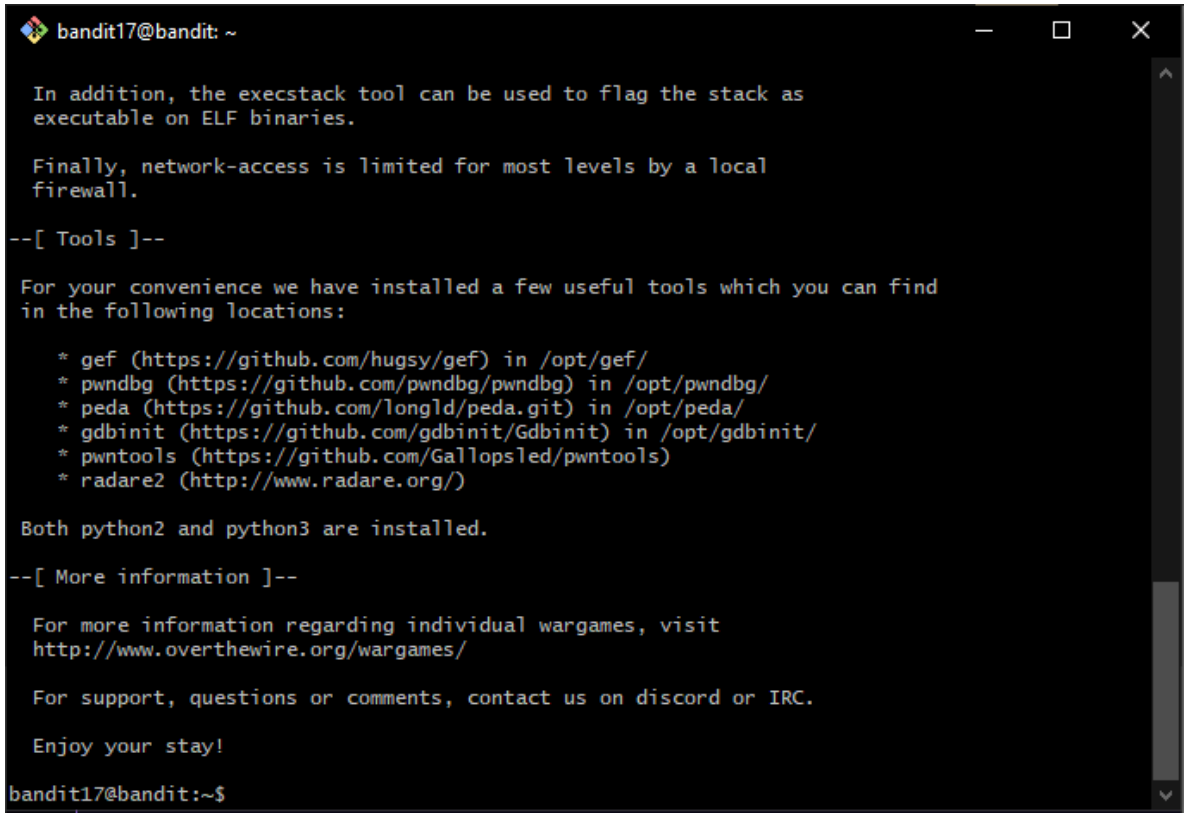


Echo esto para poder enviar este archivo requerimos que el archivo tenga los permisos necesarios de escritura y lectura para el usuario. Para ello utilizaremos el comando chmod con la configuracion 400, luego de esto lo compribamos.

chmod 400 clave.private

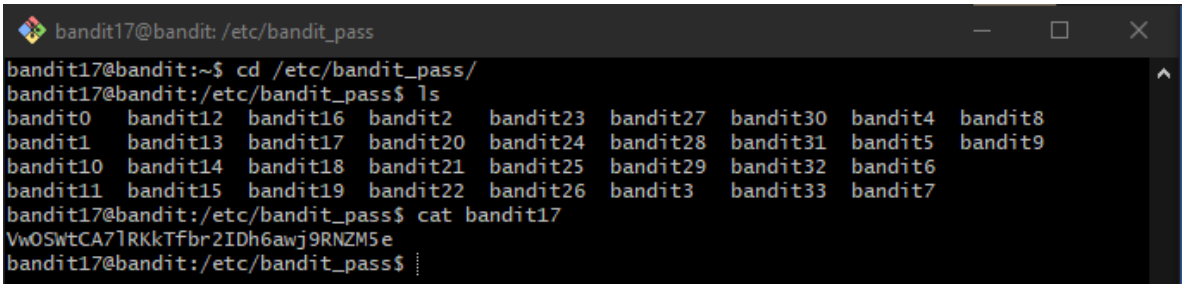
ssh -i clave.private bandit17@localhost -p 2220


bandit17@bandit: ~
bandit16@bandit:/tmp/roberto2$ chmod 400 clave.private
bandit16@bandit:/tmp/roberto2$ ssh -i clave.private bandit17@localhost -p 2220
The authenticity of host '[localhost]:2220 ([127.0.0.1]:2220)' can't be established.
ED25519 key fingerprint is SHA256:C2ihUBV7ihnV1wUXRb4RrEcLfXC5CXlhmAAM/urerLY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Could not create directory '/home/bandit16/.ssh' (Permission denied).
Failed to add the host to the list of known hosts (/home/bandit16/.ssh/known_hosts).
This is an OverTheWire game server.
More information on http://www.overthewire.org/wargames
!!! You are trying to log into this SSH server with a password on port 2220 from localhost.
!!! Connecting from localhost is blocked to conserve resources.
!!! Please log out and log in again.


bandit17@bandit: ~
In addition, the execstack tool can be used to flag the stack as
executable on ELF binaries.
Finally, network-access is limited for most levels by a local
firewall.
--[Tools]--
For your convenience we have installed a few useful tools which you can find
in the following locations:
* gef (https://github.com/hugsy/gef) in /opt/gef/
* pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/
* peda (https://github.com/longld/peda.git) in /opt/peda/
* gdbinit (https://github.com/gdbinit/Gdbinit) in /opt/gdbinit/
* pwntools (https://github.com/Gallopsled/pwntools)
* radare2 (http://www.radare.org/)
Both python2 and python3 are installed.
--[More information]--
For more information regarding individual wargames, visit
http://www.overthewire.org/wargames/
For support, questions or comments, contact us on discord or IRC.
Enjoy your stay!
bandit17@bandit:~$

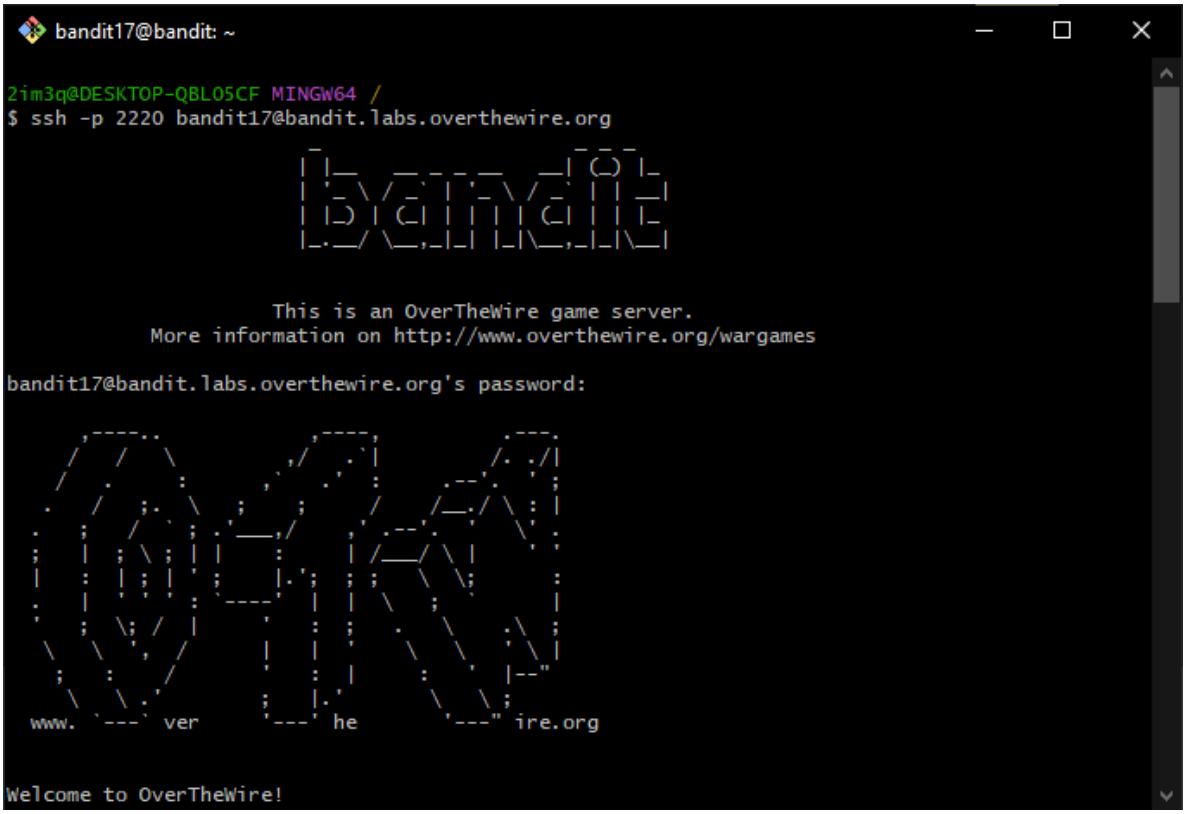
Por ultimo buscamos nuestra contraseña como en el ejercisio 15.

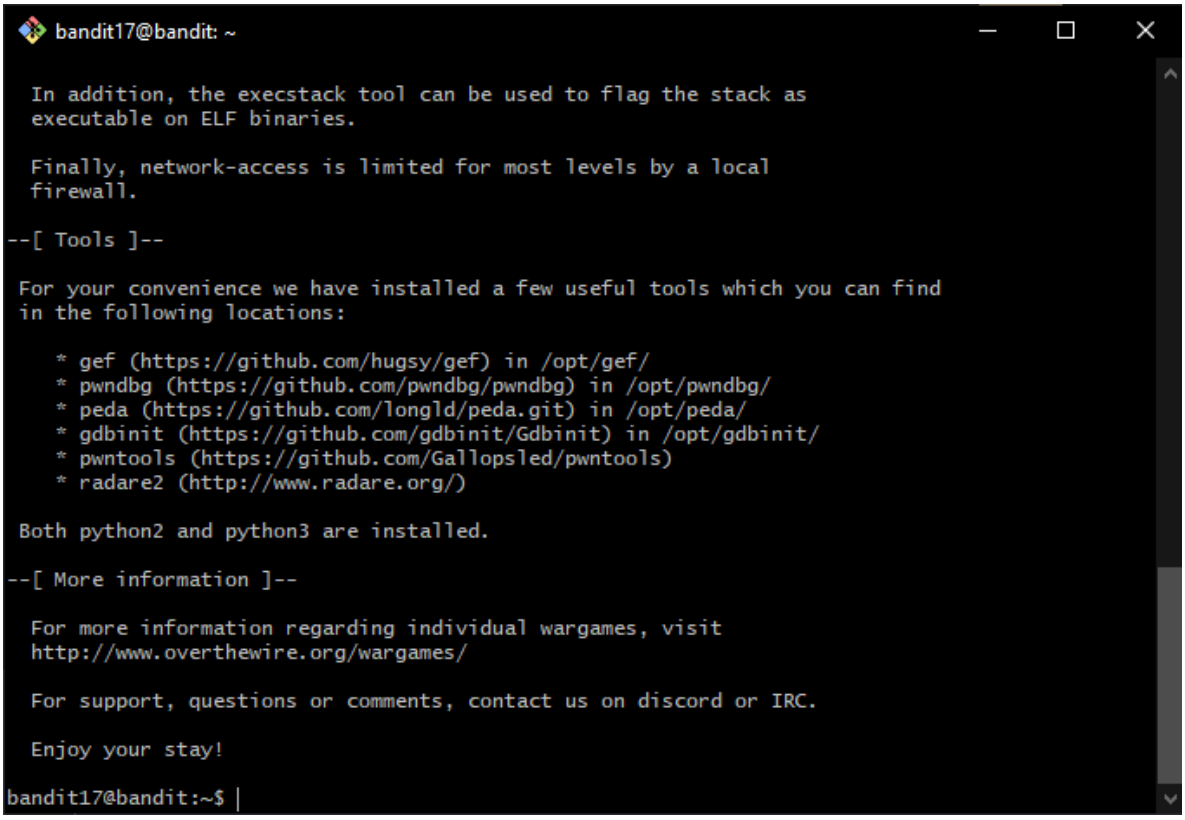
cd /etc/bandit_pass/



Pass VwOSWtCA7lRKkTfbr2IDh6awj9RNZM5e

Comprobamos.





Referencias

- https://es.wikipedia.org/wiki/Base64