Tarea Ejercicios bandit 0-17

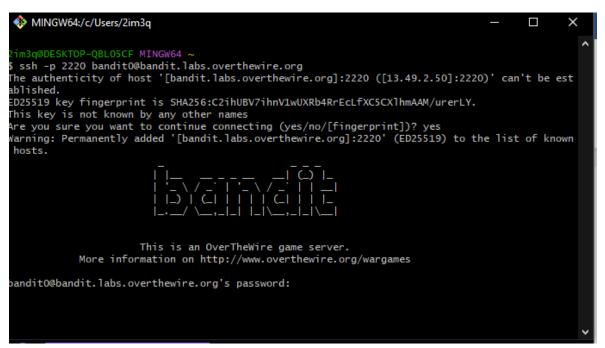
Alumno: Roberto Carlos Quintana Escamilla

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

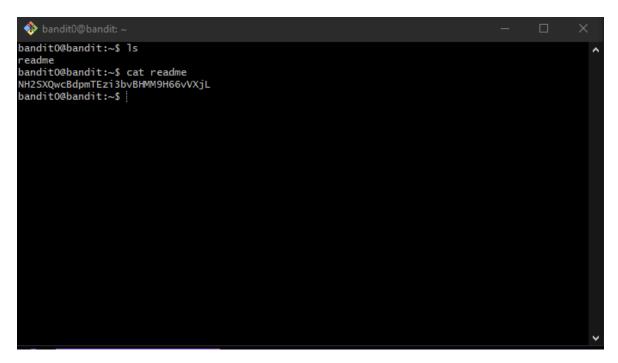
```
🚸 bandit0@bandit: ~
                                                                                                       -[ Tools ]--
 For your convenience we have installed a few useful tools which you can find
 in the following locations:
    * gef (https://github.com/hugsy/gef) in /opt/gef/
      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!
bandit0@bandit:~$
```

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.

```
🏶 bandit2@bandit: ~
                                                                                                        bandit1@bandit:~$ exit
Connection to bandit.labs.overthewire.org closed.
 im3q@DESKTOP-QBL05CF MINGW64 ~
$ ssh -p 2220 bandit2@bandit.labs.overthewire.org
                          This is an OverTheWire game server.
              More information on http://www.overthewire.org/wargames
bandit2@bandit.labs.overthewire.org's password:
 🚸 bandit2@bandit; ~
                                                                                                        --[ 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!
bandit2@bandit:~$
```

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.

```
🚸 bandit3@bandit: ~
                                                                                                                  bandit2@bandit:~$ exit
logout
Connection to bandit.labs.overthewire.org closed.
$ ssh -p 2220 bandit3@bandit.labs.overthewire.org
               This is an OverTheWire game server.
More information on http://www.overthewire.org/wargames
bandit3@bandit.labs.overthewire.org's password:
 🚸 bandit3@bandit: ~
                                                                                                                  -[ 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!
bandit3@bandit:~$
```

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.



```
🚸 bandit4@bandit: ~
                                                                                                      -[ 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!
bandit4@bandit:~$
```

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/*

```
bandit4@bandit:~/inhere$ cd ..
bandit4@bandit:~$ file inhere/*
inhere/-file00: data
inhere/-file01: data
inhere/-file03: data
inhere/-file04: data
inhere/-file05: data
inhere/-file06: data
inhere/-file08: data
inhere/-file07: ASCII text
inhere/-file08: data
inhere/-file09: data
```

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.



```
🚸 bandit5@bandit: ~
                                                                                              -[ 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!
bandit5@bandit:~$
```

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
                                                                                 bandit5@bandit:~$ ls
inhere
bandit5@bandit:~$ cd inhere/
bandit5@bandit:~/inhere$ ls -a
            maybehere02 maybehere10 maybehere14 maybehere18
            maybehere03
                       maybehere07
                                    maybehere11 maybehere15
                                                             maybehere19
                                                maybehere16
maybehere00 maybehere04 maybehere08
                                    maybehere12
maybehere01 maybehere05
                        maybehere09
                                    maybehere13
                                                maybehere17
bandit5@bandit:~/inhere$
```

```
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
           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

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

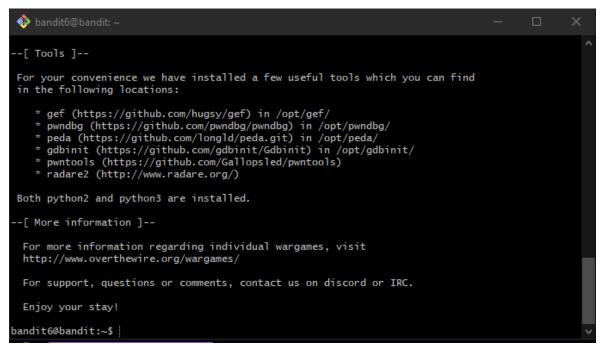
cat ./maybehere07/.file2



Pass P4L4vucdmLnm8I7VI7jG1ApGSfjYKqJU

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

Is -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

```
🏶 bandit6@bandit: ~
                                                                                                                           ×
bandit6@bandit:~$ find / -user bandit7 -group bandit6 -size 33c
find: '/dev/mqueue': Permission denied
find: '/dev/shm': Permission denied
find: '/var/spool/bandit24': Permission denied
find: '/var/spool/rsyslog': Permission denied
 ind: '/var/spool/cron/crontabs': Permission denied
 ind: '/var/crash': Permission denied
 find: '/var/snap/lxd/common/lxd': Permission denied
 Find: '/var/cache/ldconfig': Permission denied
Find: '/var/cache/apt/archives/partial': Permission denied
find: '/var/cache/apparmor/e10c1cf9.0': Permission denied
find: '/var/cache/apparmor/c47eabf7.0': Permission denied
find: '/var/cache/private': Permission denied
find: '/var/cache/pollinate': Permission denied
 ind: '/var/log': Permission denied
 find: '/var/lib/update-notifier/package-data-downloads/partial': Permission denied find: '/var/lib/polkit-1': Permission denied
find: '/var/lib/chrony': Permission denied
find: '/var/lib/snapd/void': Permission denied
find: '/var/lib/snapd/cookie': Permission denied
find: '/var/lib/amazon': Permission denied
/var/lib/dpkg/info/bandit7.password
 Find: '/var/lib/apt/lists/partial': Permission denied
Find: '/var/lib/private': Permission denied
find: '/var/tmp': Permission denied
```

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.

```
🚸 bandit7@bandit: ~
-[ 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!
bandit7@bandit:~$
```

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.

Vi data.tx

```
    bandit7@bandit: ~
    bandit7@bandit: ~
    ls
data.txt
bandit7@bandit:~$ vi data.txt |
```

```
bandit7@bandit: ~
                                                                                                 ×
aboding ErTQmlTafRb8szvTLpbV25MPOPEexBsH
                 f08zz1eLIJmv24fTys7e7zAWVYdnTbfg
locket's
melt
        HVLgPRIrjbzrbjwFZ5M8aQCavUuRdQtb
popular Rjy5b8oEjivOe4gX82ErCZ7BFZDgVkJP
odious 6JV4M56xFJkIUriwUcJzImGcs55THFQT
                OTXcsyXyXO8nCpuojmbChQf1RZIZj5nM
taxonomies
       3jvaD1qNXximI2EnBFaIO6HQqhylpucs
land
              bqmUOYYKbkoZyKlabxwjbNM6ZpB3y9eG
2EabBTby3LfWR5y9IHxdvSvqhUStUEeQ
elevator
vacationed
termed hKPxiEJFjOhPdoVvfq15am94F6Azholf
playgoer's 2AX7IxgtDuc1j0aYcIN2uCsCi8Rjx0WL
                  TESKZCOXvTetKOS9xNwm25STk5iWrBvP
millionth
fancies kuthgIL6KI7fpG88yLUmXiHNK6i8XHCg
effrontery
                s42Nm2Epse3L12rhttZHeZ13YuBTPwMd
heavy FZolxGLiO3cQebJBae4nJ4tnkpuRUC5y
psychiatrists Xi5sibqB4pLQR6bqWZ6EQMv7xy9tIFbT
piffle M12acskE21rV0RRS6v8yN9ZXwWqELzrt
misdo 7MyxRpqnsuVTz9IhYTSwNz0A1SMo4UYt
criteria
                  lRgIJGiE5tQ4emxurSktW9PWf9QCDTkh
                 WJmRA2YxAaamGHWZ1VXUT1s9hmRmp91x
bushwhacks
dish's lK2hpJkIw5LnmjRgcet6RPIOb0ERxbz8
shorting NqGPS1fPxoM1w2UCqPR5vvGy
facial 5BPb4JEuWoY5OnPYGb9b5rEPTgOdlCeY
                NqGPS1fPxoM1w2UCqPR5vvGy56eFHTvq
agglutinating blfMxTnd6XexL5MsQZmjorihwP8Blbzx
'data.txt" [readonly] 98567L, 4184396B
                                                                                      1,1
                                                                                                     Top
```

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.



```
bandit8@bandit: ~
                                                                                                      --[ 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!
bandit8@bandit:~$
```

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.



```
bandit9@bandit ~

--[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!
```

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 "="

```
🚸 bandit9@bandit: ~
                                                                                                    ×
bandit9@bandit:~$ strings data.txt | grep "="
           = the
I2=Z
K=y3>
!=j$u
         ==== password
h; ==
         == isT
 =XQ
[Qi#Z=c
i=|V
!/=j>:]zx
r>i"=
XZ>~=
n.E===
           ==== G7w8LIi6J3kTb8A7j9LgrywtEUlyyp6s
~UtFS=
eY4<={_
bandit9@bandit:~$ |
```

pass G7w8Lli6J3kTb8A7j9LgrywtEUlyyp6s

Comprobamos.



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.

```
bandit10@bandit:~$
bandit10@bandit:~$ ls
data.txt
bandit10@bandit:~$ cat data.txt
VGhlIHBhc3N3b33kIGlzIDZ6UGV6aUxkUjJSS05kTllGTmI2blZDS3pwaGxYSEJNCg==
bandit10@bandit:~$
```

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	Α	16	Q	32	g	48	W
1	В	17	R	33	h	49	x
2	С	18	S	34	i	50	у
3	D	19	Т	35	j	51	Z
4	Е	20	U	36	k	52	0
5	F	21	V	37	1	53	1
6	G	22	W	38	m	54	2
7	Н	23	X	39	n	55	3
8	I	24	Υ	40	0	56	4
9	J	25	Z	41	р	57	5
10	К	26	а	42	q	58	6
11	L	27	b	43	r	59	7
12	М	28	С	44	S	60	8
13	N	29	d	45	t	61	9
14	0	30	е	46	u	62	+
15	Р	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.

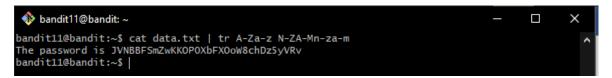
```
🚸 bandit11@bandit: ~
                                                                                                     ×
$ ssh -p 2220 bandit11@bandit.labs.overthewire.org
                         This is an OverTheWire game server.
             More information on http://www.overthewire.org/wargames
bandit11@bandit.labs.overthewire.org's password:
 🚸 bandit11@bandit: ~
                                                                                                     --[ 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!
bandit11@bandit:~$
```

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.

```
bandit11@bandit: ~
bandit11@bandit: ~$ cat data.txt
Gur cnffjbeq vf WIA00SFzMjXXBC0KoSKBbJ8puQmSlIEi
bandit11@bandit:~$ |
```

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



```
🏇 bandit12@bandit: ~
                                                                                       -[ 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!
bandit12@bandit:~$ |
```

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
```

```
bandit12@bandit:/tmp/roberto

bandit12@bandit:~\$ mkdir /tmp/roberto

cp data.txt /tmp/roberto

cd /tmp/roberto

bandit12@bandit:/tmp/roberto\$ ls

data.txt

bandit12@bandit:/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
```

```
bandit12@bandit:/tmp/roberto — — X

bandit12@bandit:/tmp/roberto$ xxd -r data.txt > archivo

ls
archivo data.txt
bandit12@bandit:/tmp/roberto$
```

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

```
bandit12@bandit:/tmp/roberto — — X

bandit12@bandit:/tmp/roberto$ file archivo
archivo: gzip compressed data, was "data2.bin", last modified: Wed Jan 11 19:18:38 2023, max c
ompression, from Unix, original size modulo 2^32 572
bandit12@bandit:/tmp/roberto$ mv archivo archivo.gz
bandit12@bandit:/tmp/roberto$ gzip -d archivo.gz
```

file archivo

mv archivo archivo.bz2

bzip2 -d archivo.bz2

```
bandit12@bandit:/tmp/roberto
bandit12@bandit:/tmp/roberto$ file archivo
archivo: bzip2 compressed data, block size = 900k
bandit12@bandit:/tmp/roberto$ mv archivo.bz2
bandit12@bandit:/tmp/roberto$ bzip2 -d archivo.bz2
```

file archivo

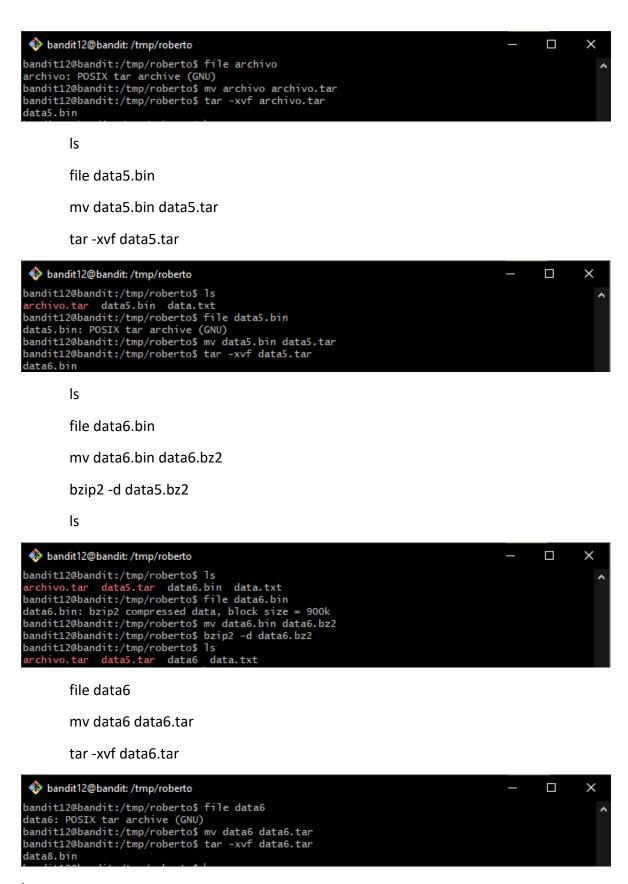
mv archivo archivo.gz

gzip -d archivo.gz

file archivo

my archivo archivo.tar

tar -xvf archivo.tar



file data8.bin mv data8.bin data8.gz gz -d data8.gz ls

file data8



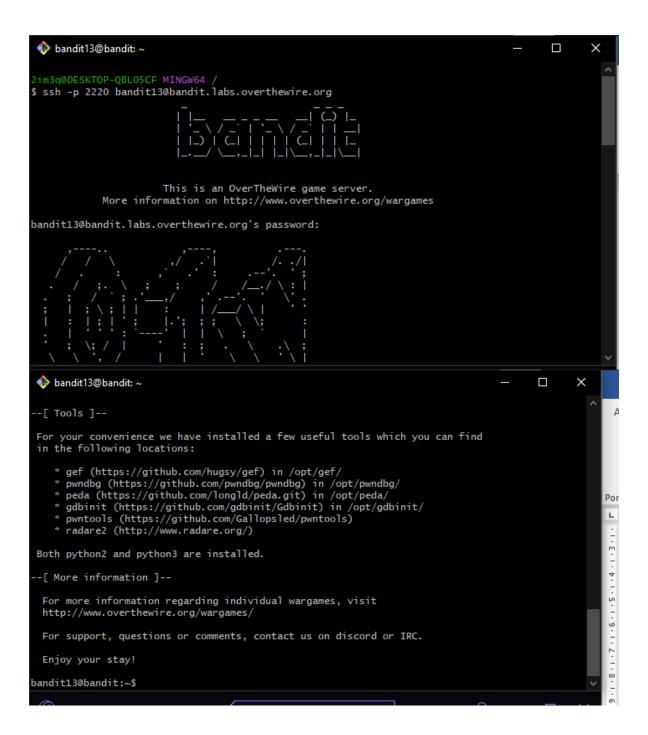
Llegamos a un archivo de texto por lo que podemos usar un cat para ver su contenido

Cat data8

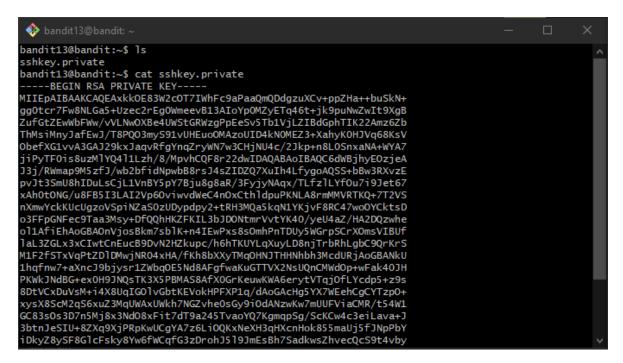


Pass wbWdlBxEir4CaE8LaPhauuOo6pwRmrDw

Comprobamos.



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

```
🚸 bandit14@bandit: ~
                                                                                              -[ 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!
bandit14@bandit:~$
```

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 fGrHPx402xGC7u7rXKDaxiWFT0iF0ENq

Comprobamos.

```
🚸 bandit14@bandit: ~
                                                                                                       -[ 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!
bandit14@bandit:~$
```

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 no y redirecionando el arcivo bandid14.

cd /etc/bandit_pass/

nc localhost 30000 < bandit14

Pass jN2kgmIXJ6fShzhT2avhotn4Zcka6tnt

Comprobamos



```
🚸 bandit15@bandit: ~
                                                                                              ×
 -[ 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!
bandit15@bandit:~$
```

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.

openssls client -connect localhost:30001 -quiet

Aquí ingresaremos la contraseña del ejercisio

```
🚸 bandit15@bandit: ~
                                                                                                  ×
bandit15@bandit:~$ openssl s_client -connect localhost:30001 -quiet
Can't use SSL_get_servername
depth=0 CN = localhost
verify error:num=18:self-signed certificate
verify return:1
depth=0 CN = localhost
verify error:num=10:certificate has expired
notAfter=Feb 13 06:43:06 2023 GMT
verify return:1
depth=0 CN = localhost
notAfter=Feb 13 06:43:06 2023 GMT
verify return:1
jN2kgmIXJ6fShzhT2avhotn4Zcka6tnt
Correct!
JQttfApK4SeyHwDlI9SXGR50qcl0Ail1
bandit15@bandit:~$
```

Pass JQttfApK4SeyHwDlI9SXGR50qclOAil1

Comprobamos



```
🚸 bandit16@bandit: ~
                                                                                                        -[ 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!
bandit16@bandit:~$
```

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

```
bandit16@bandit: ~
                                                                                               ×
bandit16@bandit:~$ nmap -A -p31000-32000 localhost
Starting Nmap 7.80 ( https://nmap.org ) at 2023-02-15 01:44 UTC
Verbosity Increased to 1.
Verbosity Increased to 2.
Stats: 0:00:36 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 0.00% done
Stats: 0:01:13 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 80.00% done; ETC: 01:46 (0:00:18 remaining)
Stats: 0:01:18 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 80.00% done; ETC: 01:46 (0:00:20 remaining)
Stats: 0:01:33 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 80.00% done; ETC: 01:46 (0:00:23 remaining)
Completed Service scan at 01:46, 97.92s elapsed (5 services on 1 host)
NSE: Script scanning 127.0.0.1.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 01:46
Completed NSE at 01:46, 0.03s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 01:46
Completed NSE at 01:46, 0.06s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 01:46
Completed NSE at 01:46, 0.00s elapsed
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00012s latency).
```

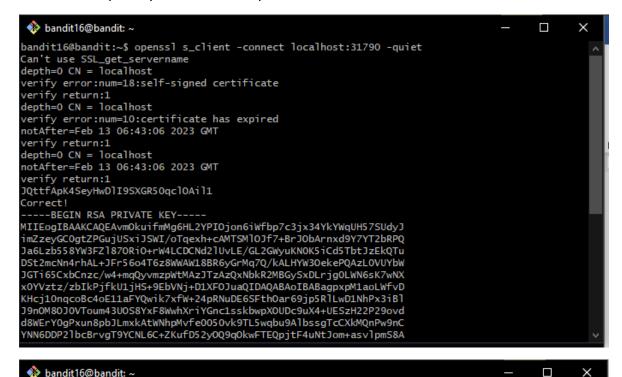
Si nos fijamos en el informe del puerto 31790 el cual es capas de leer ssl, nos fijaremos en un mensaje de error el cual nos pide qe ingresemos una contraseña

```
fingerprint-strings:
   FourOhFourRequest, GenericLines, GetRequest, HTTPOptions, Help, Kerberos, LDAPSearchReq, L
PDString, RTSPRequest, SIPOptions, SSLSessionReq, TLSSessionReq, TerminalServerCookie:
      Wrong! Please enter the correct current password
 ssl-cert: Subject: commonName=localhost
 Subject Alternative Name: DNS:localhost
 Issuer: commonName=localhost
 Public Key type: rsa
Public Key bits: 2048
 Signature Algorithm: sha1WithRSAEncryption
 Not valid before: 2023-02-13T06:42:06
 Not valid after: 2023-02-13T06:43:06
MD5: b142 f7f7 5f5d cc78 ebc3 2ed7 fd9e 130d
 SHA-1: f091 5d3c 4419 2c22 c466 a12f 1c70 8d42 8880 1f3d
   ----BEGIN CERTIFICATE--
 MIIDCzCCAfOgAwIBAgIEGA4i7jANBgkqhkiG9w0BAQUFADAUMRIwEAYDVQQDDAls
 b2NhbGhvc3QwHhcNMjMwMjEzMDY0MjA2WhcNMjMwMjEzMDY0MzA2WjAUMRIwEAYD
 VQQDDA1sb2NhbGhvc3QwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCa
 ddd+yKL6tk9uUPMQpELUWETiZ/tmrfcSygw+cqXIyGQKsnnf5houUXmDHyDNI3/PpNQzJjvsVQLjUlmVRcUtDUC9sjQMuRbdwc51BHv49pv009bqG79ZKKi1hridJPXk
 MYNYeuwlhJmQ1f8LQ5K+Yt/AFdwdKR2LOtp/rARtvUgMj2dzloZdf2Am79mXzi50
 ve1sP50plFgzqJEP8MT8r6i7spmXAOhoJxTrQ6oT1eq3uHu0j6J0z46wd90jI2AK
 6EdWGIWgrlxfMaRPEPQJbEZKgyrARbpkbo4086Hnc3GDzMorc8T8rd6M1SBu6jTR
 RR2FJdzYdoQoAq4oOg6jAgMBAAGjZTBjMBQGA1UdEQQNMAuCCWxvY2FsaG9zdDBL
 BglghkgBhvhCAQ0EPhY8QXV0b21hdGljYWxseSBnZW5lcmF0ZWQgYnkgTmNhdC4g
 U2V1IGh0dHBz0i8vbm1hcC5vcmcvbmNhdC8uMA0GCSqGSIb3DQEBBQUAA4IBAQAZ
 9E0/wp0y2q4DLnoF6p2faG7mE3L5bZXXIkkWileaA/yYQxgKCwRw93Qzqsdhpuap
 RtywtYU7Khgfkev/pIvNUa13wgF6QZ3nC9MnrMlY/uHT53Osv1dZNOsknjU3f1zg
 PUT/Z8RdTgu6qvPDyT8KTJ0oi7UNECuhyres5s8bhWvG/eZeX4x5+1YmYDQy90fS
 j2dM5yknKHUKxbIQQxMoqQImmT31VaPz4/loT/1Ld4RL0+zeL+rZbewS0zLlRYIh
 F0feRh06xJ+stFiS+H8+vHFmv4u9jaqcuYdGkJkw6be3eXvvwvLLmZ/svAKs6XT2
 x7I4FKGqzLCTzSlj0PMa
     --END CERTIFICATE----
```

Probamos ingresar la contraseña a este puerto

openssl s_client -connect localhost:31790 -quie

JQttfApK4SeyHwDll9SXGR50qclOAil1



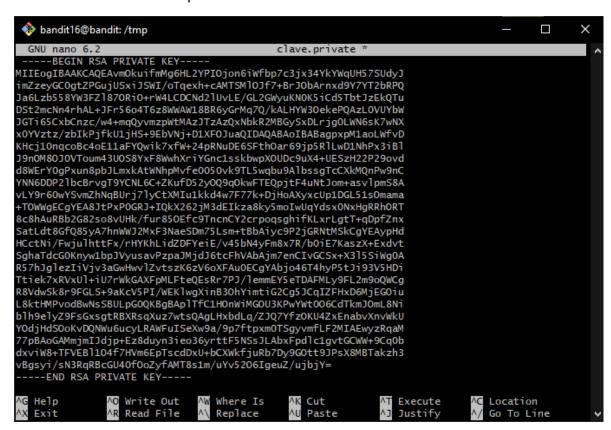
bandit16@bandit: ~ DSt2mcNn4rhAL+JFr56o4T6z8WWAW18BR6yGrMq7Q/kALHYW3OekePQAzLOVUYbW JGTi65CxbCnzc/w4+mqQyvmzpWtMAzJTzAzQxNbkR2MBGySxDLrjg0LWN6sK7wNX xOYVztz/zbIkPjfkU1jHS+9EbVNj+D1XFOJuaQIDAQABAoIBABagpxpM1aoLWfvD KHcj1OnqcoBc4oE11aFYQwik7xfW+24pRNuDE6SFthOar69jp5R1LwD1NhPx3iB1 J9nOM80J0VToum43U0S8YxF8WwhXriYGnc1sskbwpX0UDc9uX4+UESzH22P29ovd d8WErY0gPxun8pbJLmxkAtWNhpMvfe0050vk9TL5wqbu9A1bssgTcCXkMQnPw9nC YNN6DDP21bcBrvgT9YCNL6C+ZKufD52yOQ9qOkwFTEQpjtF4uNtJom+asv1pmS8A vLY9r60wYSvmZhNqBUrj7lyCtXMIu1kkd4w7F77k+DjHoAXyxcUp1DGL51sOmama +TOWWgECgYEA8JtPxPOGRJ+IQkX262jM3dEIkza8ky5moIwUqYdsx0NxHgRRhORT 8c8hAuRBb2G82so8vUHk/fur850Efc9TncnCY2crpoqsghifKLxrLgtT+qDpfZnx SatLdt8GfQ85yA7hnWWJ2MxF3NaeSDm75Lsm+tBbAiyc9P2jGRNtMSkCgYEAypHd HCctNi/FwjulhttFx/rHYKhLidZDFYeiE/v45bN4yFm8x7R/b0iE7KaszX+Exdvt SghaTdcG0Knyw1bpJVyusavPzpaJMjdJ6tcFhVAbAjm7enCIvGCSx+X315SiWg0A R57hJglezIiVjv3aGwHwvlZvtszK6zV6oXFAu0ECgYAbjo46T4hyP5tJi93V5HDi Ttiek7xRVxU1+iU7rWkGAXFpMLFteQEsRr7PJ/lemmEY5eTDAFMLy9FL2m9oQWCq R8VdwSk8r9FGLS+9aKcV5PI/WEKlwgXinB3OhYimtiG2Cg5JCqIZFHxD6MjEGOiu L8ktHMPvodBwNsSBULpGOQKBgBAplTfC1HOnWiMGOU3KPwYWt0O6CdTkmJOmL8Ni blh9elyZ9FsGxsgtRBXRsqXuz7wtsQAgLHxbdLq/ZJQ7YfzOKU4ZxEnabvXnvWkU YOdjHdSOoKvDQNWu6ucyLRAWFuISeXw9a/9p7ftpxm0TSgyvmfLF2MIAEwyzRqaM 77pBAoGAMmjmlJdjp+Ez8duyn3ieo36yrttF5NSsJLAbxFpdlc1gvtGCWW+9Cq0b dxviW8+TFVEB1104f7HVm6EpTscdDxU+bCXWkfjuRb7Dy9G0tt9JPsX8MBTakzh3 vBgsyi/sN3RqRBcGU40f0oZyfAMT8s1m/uYv5206IgeuZ/ujbjY= --END RSA PRIVATE KEY---bandit16@bandit:~\$

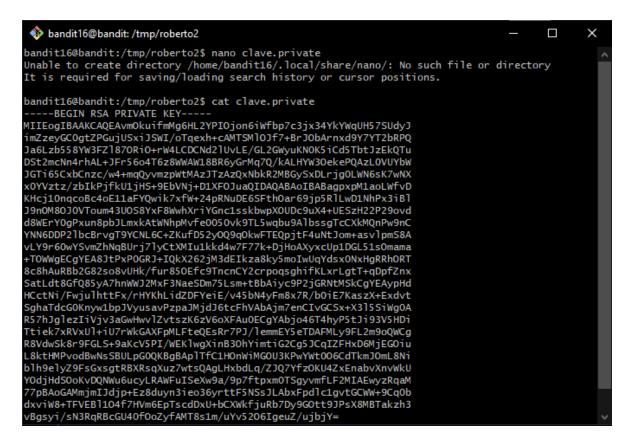
----BEGIN RSA PRIVATE KEY----

MIIEogIBAAKCAQEAvmOkuifmMg6HL2YPIOjon6iWfbp7c3jx34YkYWqUH57SUdyJimZzeyGCOgtZPGujUSxiJSWI/oTqexh+cAMTSMlOJf7+BrJObArnxd9Y7YT2bRPQJa6Lzb558YW3FZ187ORiO+rW4LCDCNd2lUvLE/GL2GWyUKNOK5iCd5TbtJzEkQTuDSt2mcNn4rhAL+JFr56o4T6z8wWaW18BR6yGrMq7Q/kALHYW3OekePQAzLOVUYbWJGTi65CxbCnzc/w4+mqQyvmzpWtMAzJTzAzQxNbkR2MBGySxDLrjg0LWN6sK7WNXx0YVztz/zbIkPjfkU1jHS+9EbVNj+D1XFOJuaQIDAQABAoIBABagpxpM1aoLWfvD

KHcj10nqcoBc4oE11aFYQwik7xfW+24pRNuDE6SFthOar69jp5RlLwD1NhPx3iBl J9nOM8OJOVToum43UOS8YxF8wwhXriYGnc1sskbwpXOUDc9uX4+UESzH22P29ovd d8WErYOgPxun8pbJLmxkAtWNhpMvfe0050vk9TL5wqbu9AlbssgTcCXkMQnPw9nC YNN6DDP2lbcBrvgT9YCNL6C+ZKufD52yOQ9qOkwFTEQpjtF4UNtJom+asvlpm58A vLY9r60wYsvmZhNqBUrj7lyCtXMIu1kkd4w7F77k+DjHoAXyxcUp1DGL51sOmama +TOWWgECgYEA8JtPxPOGRJ+IQkX262jM3dEIkza8ky5moIwUqYdsxONxHgRRhORT 8c8hAuRBb2G82so8vUHk/fur850Efc9TncnCY2crpoqsghifKLxrLgtT+qDpfZnx SatLdt8GfQ85yA7hnWwJ2MxF3NaeSDm75Lsm+tBbAiyc9P2jGRNtMSkCgYEAypHd HCctni/FwjulhttFx/rHYKhLidZDFYeiE/v45bN4yFm8x7R/b0iE7KaszX+Exdvt SghaTdcGOKnyw1bpJVyusavPzpaJMjdJ6tcFhVAbAjm7encIvGCSx+X315siwg0A R57hJglezIiVjv3aGwHwv1ZvtszK6zV6oXFAu0ECgYAbjo46T4hyP5tJi93V5HDiTtiek7xRVxUl+iU7rWkGAXFpMLFteQEsRr7PJ/lemmEY5eTDAFMLy9FL2m9oQWCg R8VdwSk8r9FGLS+9aKcV5PI/WEKlwgXinB3OhYimtiG2Cg5JCqIZFHxD6MjEGOiuL8ktHMPvodBwNsSBULpGOQKBgBAplTfC1HOnwiMGOU3KPwYwt0O6CdTkmJOmL8Nib1h9elyZ9FsGxsgtrBXRsqXuz7wtsQAgLHxbdLq/ZJQ7YfzOKU4ZxEnabvXnvwkUYOdjHdSOOKvDQNwu6ucyLRAWFuISexw9a/9p7ftpxm0TSgyvmfLF2MIAEwyzRqaM77pBAoGAMmjmIJdjp+Ez8duyn3ieo36yrttF5NSsJLAbxFpdlc1gvtGCwW+9Cq0bdxviw8+TFVEBl104f7HVm6EpTscdDxU+bCXwkfjuRb7Dy9GOtt9JPsx8MBTakzh3vBgsyi/sN3RqRBcGU40fOoZyfAMT8s1m/uYv5206IgeuZ/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.

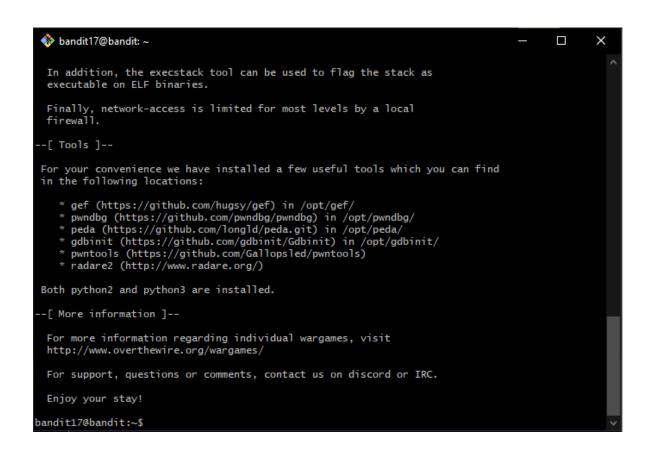




Echo esto para poder enviar este archivo requerimos que el archivo tenga los permisos necesarios de escritura y lectura para el usuario. Para ello utilizaremos el comando chmod con la configuracion 400, luego de esto lo compribamos.

chmod 400 clave.private

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



Por ultimo buscamos nuestra contraseña como en el ejercisio 15.

cd /etc/bandit_pass/

```
bandit17@bandit: /etc/bandit_pass
bandit17@bandit:~$ cd /etc/bandit_pass/
bandit17@bandit:/etc/bandit_pass$ ls
         bandit12 bandit16 bandit2
bandit0
                                       bandit23
                                                 bandit27
                                                           bandit30
                                                                     bandit4
                                                                              bandit8
bandit1
         bandit13
                   bandit17
                             bandit20
                                       bandit24
                                                 bandit28
                                                           bandit31
                                                                     bandit5
                                                                              bandit9
bandit10 bandit14 bandit18 bandit21
                                       bandit25
                                                 bandit29
                                                           bandit32
                                                                     bandit6
bandit11 bandit15 bandit19 bandit22 bandit26
                                                 bandit3
                                                           bandit33
                                                                     bandit7
bandit17@bandit:/etc/bandit_pass$ cat bandit17
VwOSWtCA71RKkTfbr2IDh6awj9RNZM5e
bandit17@bandit:/etc/bandit_pass$
```

Pass VwOSWtCA7lRKkTfbr2IDh6awj9RNZM5e

Comprobamos.



```
🚸 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)
# padare2 (http://www.padare.org/)
        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:~$
```

Referencias

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