

Bandit level 1 - 2

Una vez conectados, repetiremos el proceso para averiguar si está el archivo en el directorio.

En este caso, hay un archivo llamado “-” si intentamos leerlo con el comando cat, no nos devolverá nada y se quedaran en líneas vacías.

Por lo que hay que usar cat ./-

```
bandit1@bandit:~$ ls
-
bandit1@bandit:~$ cat ./-
rRGizSaX8Mk1RTb1CNQoXTcYZWU6lgzi
bandit1@bandit:~$
```

Nos dará la contraseña para conectarnos a la siguiente máquina.

Bandit level 2 - 3

Cuando usamos ls, vemos un archivo con espacios en el nombre.

Para poder usar cat en el archivo tenemos que ponerles “” para buscarlo.

```
bandit2@bandit:~$ ls
spaces in this filename
bandit2@bandit:~$ cat "spaces in this filename"
aBZ0W5EmUfAf7kHTQe0wd8bauFJ2lAiG
bandit2@bandit:~$
```

Bandit level 3 - 4

En el directorio actual hay un subdirectorio donde si vemos que hay dentro con ls, no sale nada. Por lo que usamos ls -a para ver los archivos ocultos donde nos sale el archivo .hidden con la contraseña del siguiente equipo.

```
bandit3@bandit:~$ ls
inhere
bandit3@bandit:~$ cd inhere/
bandit3@bandit:~/inhere$ ls
bandit3@bandit:~/inhere$ ls -a
.  ..  .hidden
bandit3@bandit:~/inhere$ cat .hidden
2EW7BBsr6aMMoJ2HjW067dm8EqX26xNe
bandit3@bandit:~/inhere$
```

Bandito level 4 – 5

En este ejercicio, en la carpeta de inhere hay 10 archivos, hay que averiguar cuál de todos tiene caracteres que nosotros podamos leer. Para eso usamos el comando `file ./*`, lo que hace es decirnos que tipo de contenido tiene el archivo por lo que no saldrá cual tiene texto en él.

```
bandit4@bandit:~/inhere$ file ./*
./-file00: data
./-file01: data
./-file02: data
./-file03: data
./-file04: data
./-file05: data
./-file06: data
./-file07: ASCII text
./-file08: data
./-file09: data
```

Luego solo tendremos que hacer un `cat` en él y ver la contraseña que está dentro del archivo.

```
bandit4@bandit:~/inhere$ cat ./-file07
lrIWWI6bB37kxfiCQZqUdOIYfr6eEqR
```

Bandito level 5 – 6

En este ejercicio nos encontramos con 19 subdirectorios dentro del inhere. En cada cual hay varios documentos y hay que encontrar el archivo con 3 características.

- Legible para el humano
- 1033 bytes

- Que no sea ejecutable

Para esto se usa el comando find con diferentes características

```
find . -type f -size 1033c ! -executable -exec file '{}' \; | grep ASCII
```

```
bandit5@bandit:~/inhere$ find . -type f -size 1033c ! -executable -exec file '{}' \; | grep ASCII
./maybehere07/.file2: ASCII text, with very long lines (1000)
```

Cuando te da el archivo lo único que tienes que hacer es usar cat en el para ver el contenido y ahí tienes la contraseña para el siguiente equipo.

```
bandit5@bandit:~/inhere$ cat ./maybehere07/.file2
P4L4vucdmLnM8I7Vl7jG1ApGSfjYKqJU
```

Bandito level 6 – 7

Es un nivel bastante parecido al anterior, lo que tenemos que hacer ahora es buscar el archivo solo que ahora tiene otras características.

- Usuario bandit7
- Grupo bandit6
- 33 bytes

Para lograr esto volveremos a usar el comando de find.

```
find / -type f -user bandit7 -group bandit6 -size 33c
```

Sin embargo, si ponemos el comando tal cual, nos daría error por los permisos (aunque intentemos el comando con SUDO) por lo que hay que agregar otra cosa al comando que es 2>/dev/null

Luego usaremos cat en el archivo y tendremos la contraseña.

```
bandit6@bandit:~$ find / -type f -user bandit7 -group bandit6 -size 33c 2>/dev/null
/var/lib/dpkg/info/bandit7.password
bandit6@bandit:~$ cat /var/lib/dpkg/info/bandit7.password
z7WtoNQU2XfjmMtWA8u5rN4vzqu4v99S
bandit6@bandit:~$
```

Bandito level 7 – 8

Te pide que encuentres la contraseña que está al lado de millionth.

Si vemos el archivo que tenemos en el equipo vemos que tiene un montón de líneas con contraseñas diferentes.

Para completar este ejercicio se pueden hacer de varias formas.

Una es sin comandos es usando nano para meterte en el archivo y con C^W buscamos la palabra y la contraseña está al lado.

La otra forma es con un comando para leer el archivo y separar la línea que contiene la contraseña

`cat data.txt | grep millionth`

```
bandit7@bandit:~$ nano data.txt
Unable to create directory /home/bandit7/.local/share/nano/: No such file or directory
It is required for saving/loading search history or cursor positions.

bandit7@bandit:~$ cat data.txt | grep millionth
millionth      TESKZC0XvTetK0S9xNwm25STk5iWrBvP
```

Bandito level 8 – 9

Veremos un documento con un montón de líneas de contraseñas, las cuales estarán todas repetidas menos la que necesitamos para avanzar.

Para averiguar cual no está repetida necesitamos un comando de Ubuntu.

`sort data.txt | uniq -u`

La funcionalidad de uniq es filtrar el contenido y con -u nos devuelve la línea que no se repiten, también se pueden contar las líneas repetidas con -c pero no nos hace falta ahora mismo

```
bandit8@bandit:~$ sort data.txt | uniq -u
EN632PlfYiZbn3PhVK3XOGSlNIInNE00t
```

Bandito level 9 – 10

Es algo parecido solo que se tiene que buscar de manera diferente, nos dan un archivo lleno de caracteres ilegibles para nosotros. Pero nos dicen que la contraseña esta después de varios ===.

En este caso buscamos una cadena de texto que este después de los signos === asique usaremos el comando strings sobre el archivo data.txt.

```
strings data.txt | grep ===
```

Así nos darán varias líneas que están después de los === pero siguiendo la lógica de las contraseñas anteriores sabremos cual es.

```
bandit9@bandit:~$ strings data.txt | grep =====  
x]T===== theG)"  
===== passwordk^  
===== is  
===== G7w8LIi6J3kTb8A7j9LgrywtEUlyyp6s  
bandit9@bandit:~$
```

Bandito level 10 – 11

Nos encontramos un archivo con los datos codificados en base 64. Si hacemos cat en el archivo nos saldrán muchos caracteres aleatorios.

Tendremos que usar un comando 'base64' con la función -d para decodificar el archivo.

```
bandit10@bandit:~$ base64 -d data.txt  
The password is 6zPezILdR2RKNdNYFNb6nVCKzphlXHBM  
bandit10@bandit:~$
```

Bandito level 11 - 12

En este ejercicio los caracteres del documento están movidos 13 letras.

Lo que hay que hacer es rotar los caracteres con el comando tr.

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

```
bandit11@bandit:~$ cat data.txt
Gur cnffjbeq vf WIA00SFzMjXXBC0KoSKBbJ8puQm5lIEi
bandit11@bandit:~$ cat data.txt | tr 'A-Za-z' 'N-ZA-Mn-za-m'
The password is JVNBBFSmZWKKOP0XbFX0oW8chDz5yVRv
bandit11@bandit:~$
```

Bandito level 12 – 13

En este ejercicio nos dan un documento comprimido, para que sea más fácil tendremos que crear una carpeta aparte y copiar el archivo ahí.

Cuando lo tenemos ahí se empieza un proceso de descomprimirlo con xxd, gzip y bzip2.

Adjunto una pequeña parte del proceso, ya que poner toda la consola sería muy largo.

```
bandit12@bandit:/tmp/tmp.wWB380hFoX$ bzip -d data6.bin
Command 'bzip' not found, but there are 20 similar ones.
bandit12@bandit:/tmp/tmp.wWB380hFoX$ bzip2 -d data6.bin
bzip2: Can't guess original name for data6.bin -- using data6.bin.out
bandit12@bandit:/tmp/tmp.wWB380hFoX$ ls
compressed_data.tar  data5.bin  data6.bin.out  hexdump_data
bandit12@bandit:/tmp/tmp.wWB380hFoX$ tar -xf data6.bin.out
bandit12@bandit:/tmp/tmp.wWB380hFoX$ ls
compressed_data.tar  data5.bin  data6.bin.out  data8.bin  hexdump_data
bandit12@bandit:/tmp/tmp.wWB380hFoX$ xxd data8.bin
00000000: 1f8b 0808 6855 1e65 0203 6461 7461 392e  ....hU.e..data9.
00000010: 6269 6e00 0bc9 4855 2848 2c2e 2ecf 2f4a  bin...HU(H,.../J
00000020: 51c8 2c56 284f 0a4f c971 aa70 cd2c 3271  Q.,V(0.0.q.p.,2q
00000030: 4e74 b5f0 490c c848 2c2d f5cf 372b 280f  Nt..I..H,-..7+(.
00000040: ca2d 7229 e702 00dc ec75 4731 0000 00    .-r).....UG1...
bandit12@bandit:/tmp/tmp.wWB380hFoX$ mv data8.bin data8.gz
bandit12@bandit:/tmp/tmp.wWB380hFoX$ ls
compressed_data.tar  data5.bin  data6.bin.out  data8.gz  hexdump_data
bandit12@bandit:/tmp/tmp.wWB380hFoX$ gzip -d data8.gz
bandit12@bandit:/tmp/tmp.wWB380hFoX$ ls
compressed_data.tar  data5.bin  data6.bin.out  data8  hexdump_data
bandit12@bandit:/tmp/tmp.wWB380hFoX$ cat data8
The password is wbWdLBxEir4CaE8LaPhauu0o6pwRmRDw
bandit12@bandit:/tmp/tmp.wWB380hFoX$
```

Bandito level 13 – 14

En este ejercicio no necesitamos una contraseña para pasar al siguiente, si no que necesitamos una llave para pasar al siguiente nivel.

Cuando accedemos al equipo de bandit13 nos encontramos la llave por lo que ahora necesitamos pasar la llave a nuestra máquina.

```
scp -P 2220 bandit13@bandit.labs.overthewire.org:sshkey.private.
```

```
bandit13@bandit:~$ ls  
sshkey.private  
bandit13@bandit:~$ exit  
logout  
Connection to bandit.labs.overthewire.org closed.  
dam1@dam1:~$ scp -P 2220 bandit13@bandit.labs.overthewire.org:sshkey.private .  
  
      [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ]  
    [ D ] [ G ] [ T ] [ T ] [ C ] [ L ] [ E ]  
      [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ] [ _ ]  
  
This is an OverTheWire game server.  
More information on http://www.overthewire.org/wargames  
  
bandit13@bandit.labs.overthewire.org's password:  
Permission denied, please try again.  
bandit13@bandit.labs.overthewire.org's password:  
sshkey.private          100% 1679       31.0KB/s   00:00
```

Bandito level 14 – 15

Antes de iniciar la siguiente maquina tenemos que cambiar los permisos de llave para que solo el propietario (nosotros) podamos usarla y así no de error, para eso usaremos el comando chmod

Chmod 700 sshkey.private

Luego iniciamos la máquina, pero con un cambio, como tenemos una llave el comando será ahora:

```
ssh -i sshkey.private bandit*(aquí ya es normal)
```

```
dam1@dam1:~$ chmod 700 sshkey.private
dam1@dam1:~$ ssh -i sshkey.private bandit14@bandit.labs.overthewire.org -p 2220

bandit14@bandit14:~$

This is an OverTheWire game server.
More information on http://www.overthewire.org/wargames
```

Ahora que tenemos ya la maquina iniciada, para conectarnos al puerto 30000 necesitamos primero la contraseña de bandit14 (como hemos iniciado con la llave no la tenemos) para descubrirla tenemos que usar el cat /etc/bandit_pass/bandit14 el cual nos dará la contraseña para cuando queramos usar nc localhost 30000

```
bandit14@bandit:~$ cat /etc/bandit_pass/bandit14
fGrHPx402xGC7U7rXKDaxiWFT0iF0ENq
bandit14@bandit:~$ nc localhost 30000
fGrHPx402xGC7U7rXKDaxiWFT0iF0ENq
Correct!
jN2kgmIXJ6fShzhT2avhotn4Zcka6tnt
```


Bandito level 15 – 16

Nos pide que nos conectemos a puerto 30001 pero usando SSL.

Para esto solo necesitaremos usar el comando:

```
openssl s_client -connect localhost:30001
```

Luego tenemos que ingresar la contraseña de la maquina bandit 15 y ya nos darán la siguiente contraseña.

```
bandit15@bandit:~$ openssl s_client -connect localhost:30001
CONNECTED(00000003)
Can't use SSL_get_servername
depth=0 CN = localhost
verify error:num=18:self-signed certificate
verify return:1
depth=0 CN = localhost
```

```
read R BLOCK
jN2kgmIXJ6fShzhT2avhotn4Zcka6tnt
Correct!
JQttfApK4SeyHwDlI9SXGR50qcl0Ail1

closed
bandit15@bandit:~$
```

Bandito level 16 – 17

Pide que encontremos la contraseña entre los puertos 31000 – 32000 (en un servicio que este usando SSL), para hacerlo necesitamos usar el comando nmap

```
bandit16@bandit:~$ nmap -sV localhost -p 31000-32000
Starting Nmap 7.80 ( https://nmap.org ) at 2024-02-16 10:11 UTC
Stats: 0:01:33 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 80.00% done; ETC: 10:13 (0:00:23 remaining)
Stats: 0:01:38 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 100.00% done; ETC: 10:13 (0:00:00 remaining)
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00021s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE      VERSION
31046/tcp open  echo
31518/tcp open  ssl/echo
31691/tcp open  echo
31790/tcp open  ssl/unknown
```

Después de esto vemos que hay dos puertos que están usando SSL, pero uno de ellos está en echo asique nos conectaremos al otro con el comando

openssl s_client -connect localhost:31790

Aquí no nos dará una contraseña si no una llave de acceso

```
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAvm0kuiFmMg6HL2YPI0jon6iWfbp7c3jx34YkYwUH57SudyJ
imZzeyGC0gtZPGujUSxiJSWI/oTqexh+cAMTSMLOJf7+BrJ0bArnxd9Y7YT2bRPQ
Ja6Lzb558YW3FZL870Ri0+rW4LDCND2LuvLE/GL2GWyuKN0K5iCd5TbtJzEkQTu
DSt2mcNn4rHAL+JFr56o4T6z8WWAW18BR6yGrMq7Q/kALHYW30ekePQAZL0VUYbw
JGTi65CxbCnzc/w4+mqQyvmzpwTMAzJTzAzQxNbK2MBGySxDLrjg0LWN6sK7wNX
x0YVztz/zbIkPjfkU1jHS+9EbVNj+D1XF0JuaQIDAQABaoIBABagpxpM1aoLWfVd
KHcj10nqcoBc4oE11aFYQwik7xfW+24pRNUDE6SFth0ar69jp5RLLwD1NhPx3iBl
J9nOM80J0VToum43UOS8YxF8WwhXriYGnc1sskbwpXOUDc9uX4+UESzH22P29ovd
d8WErY0gPxun8pbJLmxkAtWNhpMvfe0050vk9TL5wqbu9AlbssgTcCXkMqnPw9nC
YNN6DDP2lbcBrvgT9YCNL6C+ZKufD52yOQ9qOkwFTEQpjTF4uNtJom+asvlpnS8A
vLY9r60wYsvmZhNqBURj7lyCtXMIu1kkd4w7F77k+DjHoAXyxcUp1DGL51sOmama
+TOWWgECgYEA8JtPxP0GRJ+IQkX262jM3dEIkza8ky5moIwUqYdsx0NxHgRRh0RT
8c8hAuRbB2G82so8vUHK/fur850Efc9TncnCY2cprpoqsghifKLxrLgtT+qDpfZnx
SatLdt8GfQ85yA7hnWJ2MxF3NaeSDm75Lsm+tBbAiyC9P2jGRntMSKcgYEAypHd
HCctNi/FwjulhttFx/rHYKhLidZDFYeIE/v45bN4yFm8x7R/b0iE7KaszX+Exdvt
SghaTdcG0Knyw1bpJVyusavPzpaJMjdJ6tcFhVAbAjm7enCIvGCSx+X3L5SiWg0A
R57hJglezIiVjv3aGwHwvLZvtszK6zV6oXFAu0ECgYAbjo46T4hyP5tJi93V5Hdi
Ttlek7xRVxUL+iu7rWkGAXFpMLFteQEsRr7PJ/lemmEY5eTDAFmLY9FL2m9oQWcg
R8VdwSk8r9FGLS+9aKcV5PI/WEKlwgXinB30hYimtIG2Cg5JCqIZFHxD6MjEG0iu
```

Ahora lo que tenemos que hacer es poner la llave en nuestro archivo, en mi caso es 'sshkey17.private', hay que acordarse de que en los permisos solo tenga el usuario

```
dam1@dam1:~$ ls -l
total 44
drwxr-xr-x 2 dam1 dam1 4096 févr. 14 12:54 Descargas
drwxr-xr-x 2 dam1 dam1 4096 févr. 14 12:54 Documentos
drwxr-xr-x 2 dam1 dam1 4096 févr. 15 14:09 Escritorio
drwxr-xr-x 2 dam1 dam1 4096 févr. 14 12:54 Imágenes
drwxr-xr-x 2 dam1 dam1 4096 févr. 14 12:54 Música
drwxr-xr-x 2 dam1 dam1 4096 févr. 14 12:54 Plantillas
drwxr-xr-x 2 dam1 dam1 4096 févr. 14 12:54 Público
drwx----- 3 dam1 dam1 4096 févr. 14 12:50 snap
-rwx----- 1 dam1 dam1 1613 févr. 16 11:32 sshkey17.private
-rwx----- 1 dam1 dam1 1679 févr. 15 13:49 sshkey.private
drwxr-xr-x 2 dam1 dam1 4096 févr. 14 12:54 Videos
dam1@dam1:~$
```

Bandito level 17 – 18

En el directorio tenemos dos archivos, tienen una linea cambiada entre uno y el otro. Para averiguar la contraseña utilizaremos diff

```
bandit17@bandit:~$ ls
passwords.new passwords.old
bandit17@bandit:~$ diff passwords.old passwords.new
42c42
< p6ggwdNHncnmCNxuAt0KtKVq185ZU7AW
---
> hga5tuuCLF6fFzUpnagiMN8ssu9LFrDg
bandit17@bandit:~$
```

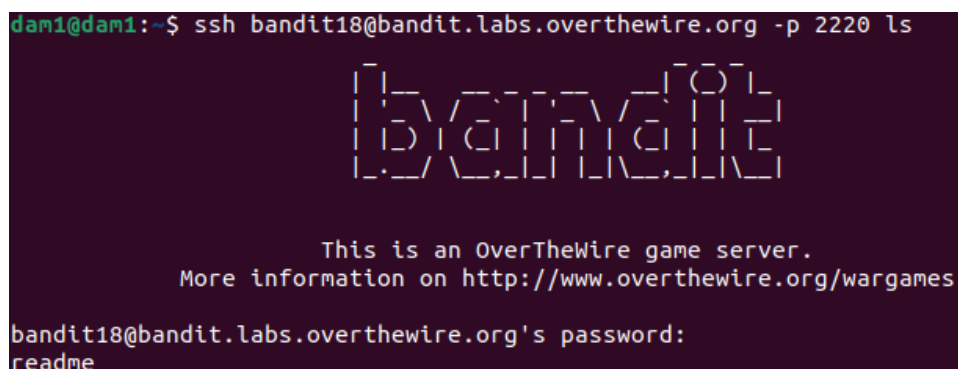
Bandito level 18 – 19

La contraseña esta en un readme en el directorio principal pero nos prohíben el paso, haciendo que nos desconectemos apenas nos conectamos.

Por lo que en vez de conectarte por ssh lo que hay que hacer es usar los comandos y asi conseguir la siguiente contraseña.

ssh bandit18@bandit.labs.overthewire.org -p 2220 ls

```
dam1@dam1:~$ ssh bandit18@bandit.labs.overthewire.org -p 2220 ls
bandit18@bandit.labs.overthewire.org:~$ ls
bandit18@bandit.labs.overthewire.org's password:
readme
```



ssh bandit18@bandit.labs.overthewire.org -p 2220 cat readme

```
dam1@dam1:~$ ssh bandit18@bandit.labs.overthewire.org -p 2220 cat readme
bandit18@bandit.labs.overthewire.org:~$ cat readme
bandit18@bandit.labs.overthewire.org's password:
awhqfNnAbc1naukrpqDYcF95h7HoMTrC
```



Ahora tenemos la contraseña y sin conectarnos directamente a la maquina.

Bandito level 19 – 20

En esta maquina nos encontramos que la contraseña solo puede ser leida por el usuario bandit20, pero por el grupo de bandit19.

Siendo nosotros el usuario bandit19 tenemos que usar cat fingiendo ser otro usuario , para esto usaremos el comando :

./bandit20-do cat /etc/bandit-pass/bandit20

```
bandit19@bandit:~$ ls -la
total 36
drwxr-xr-x  2 root    root    4096 Oct  5 06:19 .
drwxr-xr-x 70 root    root    4096 Oct  5 06:20 ..
-rwsr-x---  1 bandit20 bandit19 14876 Oct  5 06:19 bandit20-do
-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--r--  1 root    root     807 Jan  6 2022 .profile
bandit19@bandit:~$ ./bandit20-do cat /etc/bandit_pass/bandit20
VxCazJaVyki6W36BkBU0mJTCM8rR95XT
bandit19@bandit:~$
```

Bandito level 20 – 21

Nos pide que hagamos una conexión con algún puerto para poder comparar la contraseña de bandit20 y que nos pueda dar la siguiente contraseña.

Utilizaremos primero el comando

```
echo -n 'VxCazJaVyki6W36BkBU0mJTCM8rR95XT' | nc -l -p 1234 &
```

```
bandit20@bandit:~$ echo -n 'VxCazJaVyki6W36BkBU0mJTCM8rR95XT' | nc -l -p 1234 &
[2] 292037
```

Esto hará la conexión y para que nos de la siguiente contraseña es con el comando

```
./suconnect 1234
```

```
bandit20@bandit:~$ ./suconnect 1234
Read: VxCazJaVyki6W36BkBU0mJTCM8rR95XT
Password matches, sending next password
NvEJF7oVjkdltPSrdKEF0llh9V1IBcq
[1]-  Done                  echo -n 'VxCazJaVyki6W36BkBU0mJTCM8rR95XT' | nc -l
    -p 1234
bandit20@bandit:~$
```

Aquí verificará la contraseña y nos dará la siguiente si ponemos la correcta

Bandito level 21 – 22

Nos dice que nos fijemos en la carpeta /etc/cron.d/ para que nos fijemos en qué proceso está activo todo el rato.

Nos fijamos en el sub directorio de cronjob_bandit22 (ya que por numeración es el siguiente nivel)

```

drwxr-xr-x  2 root root  4096 Oct  5 06:20 .
drwxr-xr-x 106 root root 12288 Oct  5 06:20 ..
-rw-r--r--  1 root root   62 Oct  5 06:19 cronjob_bandit15_root
-rw-r--r--  1 root root   62 Oct  5 06:19 cronjob_bandit17_root
-rw-r--r--  1 root root  120 Oct  5 06:19 cronjob_bandit22
-rw-r--r--  1 root root  122 Oct  5 06:19 cronjob_bandit23
-rw-r--r--  1 root root  120 Oct  5 06:19 cronjob_bandit24
-rw-r--r--  1 root root   62 Oct  5 06:19 cronjob_bandit25_root
-rw-r--r--  1 root root  201 Jan  8 2022 e2scrub_all
-rwx-----  1 root root   52 Oct  5 06:20 otw-tmp-dir
-rw-r--r--  1 root root  102 Mar 23 2022 .placeholder
-rw-r--r--  1 root root  396 Feb  2 2021 sysstat
bandit21@bandit:~$ cat /etc/cron.d/cronjob_bandit22
@reboot bandit22 /usr/bin/cronjob_bandit22.sh &> /dev/null
* * * * * bandit22 /usr/bin/cronjob bandit22.sh &> /dev/null

```

Con los 5 asteriscos nos damos cuenta que esta en proceso todo el rato, asique ahora iremos a mirar en su archivo bash para conocer que esta haciendo.

Aqui nos muestra que lo que hace es crear un directorio en el tmp asique ahora nos iremos a fijar en esa carpeta. Cuando le hagamos cat al archivo nos daran la nueva contraseña.

```

bandit21@bandit:~$ cat /usr/bin/cronjob_bandit22.sh
#!/bin/bash
chmod 644 /tmp/t706lds9S0RqQh9aMcz6ShpAoZKF7fgv
cat /etc/bandit_pass/bandit22 > /tmp/t706lds9S0RqQh9aMcz6ShpAoZKF7fgv
bandit21@bandit:~$ cat /tmp/t706lds9S0RqQh9aMcz6ShpAoZKF7fgv
WdDozAdTMz9DiFEQ2mGLwngMfj4EZff
bandit21@bandit:~$

```

Bandito level 22 – 23

Hay que seguir los mismos pasos hasta que leemos el archivo bash, aqui nos damos cuenta que nos dan un comando

Echo I am user \$myname | md5sum | cut -d ' ' -f 1

Cuando ejecutamos esto nos dan una serie de digitos , que es un subdirectorio de tmp donde se encuentra la siguiente contraseña

```

bandit22@bandit:~$ cat /etc/cron.d/cronjob_bandit23
@reboot bandit23 /usr/bin/cronjob_bandit23.sh &> /dev/null
* * * * * bandit23 /usr/bin/cronjob_bandit23.sh &> /dev/null
bandit22@bandit:~$ cat /usr/bin/cronjob_bandit23.sh
cat: /usr/bin/cronjob_bandit23.sh: No such file or directory
bandit22@bandit:~$ cat /usr/bin/cronjob_bandit23.sh
#!/bin/bash

myname=$(whoami)
mytarget=$(echo I am user $myname | md5sum | cut -d ' ' -f 1)

echo "Copying passwordfile /etc/bandit_pass/$myname to /tmp/$mytarget"

cat /etc/bandit_pass/$myname > /tmp/$mytarget
bandit22@bandit:~$ echo I am user bandit23 | md5sum | cut -d ' ' -f 1
8ca319486bfbbc3663ea0fbc81326349
bandit22@bandit:~$ cat /tmp/8ca319486bfbbc3663ea0fbc81326349
QYw0Y2aiA672PsMmh9puTQuhoz8SyR2G
bandit22@bandit:~$

```

Bandito level 23 – 24

Lo siento profe ya no encontraba la solucion a esto, siempre me daban errores.

Les pregunte a algunos que ya lo habian pasado probe sus soluciones y tampoco asique me canse y pase a programacion