

# SAD - U4 Seguridad en redes

[Descargar estas actividades](#)

## Índice

### ▼ Actividades U4 Seguridad en redes

- [Actividad 1. Usos de nmap](#)
- [Actividad 2. Vulnerabilidades de inyección SQL en OWASP Juice Shop](#)
- [Actividad 3. Vulnerabilidades de XSS en OWASP Juice Shop](#)
- [Actividad 4. Squid, ejemplos de ACLs. Ejercicios de entrenamiento](#)

## Actividades U4 Seguridad en redes

### Actividad 1. Usos de nmap

Para cada uno de los siguientes ejercicios, realiza un escaneo de la red o del equipo indicado y muestra capturas de pantalla con los resultados obtenidos. Puedes guiarte de los ejemplos que se muestran en cada apartado.



#### Escaneos locales y remotos

Ten en cuenta que algunos escaneos se pueden hacer en local, ya que el descubrimiento de equipos y servicios se puede hacer en la propia máquina.

Otros, sin embargo, necesitan de una red local o de una red remota para poder realizar el escaneo y comprobar si nos podemos saltar las protecciones de seguridad de la red. En estos casos, un escaneo básico no devolverá información relevante, pero con los parámetros adecuados podemos obtener más información del sistema investigado.

#### Ejercicio 1.1. Escaneo básico sobre equipo o red.

- Equipo: `# nmap 172.16.0.132`
- Red: `# nmap 172.16.0.0/24`

#### Ejercicio 1.2. Filtrado de puertos.

Por defecto, Nmap escanea los 1000 puertos más usados: 21 (ftp), 22 (ssh), 80 (http), 53 (DNS), ... Se puede seleccionar puertos y rangos de los mismos.

- Puertos concretos `# nmap -p 21,22,80 172.16.0.132`
- Rango de puertos `# nmap -p 20-100 172.16.0.132`
- Escaneos UDP `# nmap -p 53,123 -sU 172.16.0.132`

#### Ejercicio 3. Descubrimiento de servicios.

Conocer qué servicio escucha detrás de un puerto.

- Versión del servicio usando los banners de respuesta `#nmap -sV 172.16.0.132`
- Intensidad del escaneo `#nmap -version-intensity 9 172.16.0.132` Mayor intensidad → más pruebas → pero más visibles

- Sistemas Operativos `#nmap -O 172.16.0.132`

#### Ejercicio 4. Escaneo TCP SYN (-sS) al puerto 443.

El método más común de escaneo TCP, es el escaneo SYN. Esto implica crear una conexión parcial al host en el puerto de destino por medio de un paquete SYN y luego evaluando la respuesta del host.

Si el paquete de solicitud no es filtrado o bloqueado por un firewall, entonces el host responderá enviando un paquete SYN/ACK si el puerto está abierto, de lo contrario enviará un paquete RST.

- `#nmap -sS -p 443 172.16.0.1`

Con la línea de comando anterior, solo se escaneará el puerto 443. Para escanear todos los puertos de la máquina, use el indicador -p seguido de un rango de puertos, como se muestra a continuación:

- `#nmap -sS 192.168.1.100 -p1-65535`

#### Ejercicio 5. Escanear una gran cantidad de máquinas, usando rangos y comodines.

- `#nmap -sA 192.168.*. 1-10,250-254`

Lo anterior escaneará todo lo que comience con 192.168 y termine con 1–10 o 250–254. También se puede utilizar la notación CIDR menos flexible. A continuación, se muestra un ejemplo sobre cómo realizar un escaneo UDP en una subred de clase C:

- `#nmap -sU 192.168.0.0/24`

Si necesitamos **descubrir equipos vivos** (es decir, que nos digan "oye, aquí estoy") en toda la red **pero no queremos saber mucho sobre ellos**, podemos lanzar un escaneo como el que sigue: `nmap -sP <ip/máscara de red>`

- `nmap -sP 192.168.1.0/24`

Con el siguiente comando podremos analizar toda una red o rango en busca de hosts. Se nos mostrarán los datos del ejemplo anterior y además el estado de algunos de sus puertos (los más comunes).

```
nmap -F 192.168.1.0/24
```

#### Ejercicio 6. Escaneo de red completa sigiloso con detección de SO.

- Ejemplo: `nmap -sS -O 192.168.1.0/24`

Este tipo de escaneo `nmap -sS -O <IP/máscara>` se diferencia del anterior en que añade algunos datos adicionales, como son:

- Tipo de dispositivo (device type): normalmente aparecerá «general purpose» o propósito general en ambientes domésticos.
- Sistema operativo: intentará reconocer el sistema o kernel (en versiones Linux).
- Distancia de red (network distance): se lanzará además una traza de red que nos indicará cuantos saltos nos separan del dispositivo/red analizado.

#### Ejercicio 7. Escanear un servidor remoto.

- Ejemplo: `nmap -A -T4 scanme.nmap.org`.

#### Ejercicio 8. Escaneo de puertos TCP o UDP.

Para escanear los puertos TCP que están abiertos en el host, usa `-sT` como se muestra:

- `nmap -sT 192.168.1.103`

### Ejercicio 9. Guarde los resultados del análisis en un archivo.

Una vez que haya completado su escaneo, puede guardar los resultados en un archivo de texto usando el indicador `-oN` y especificando el archivo de salida como se muestra a continuación: `nmap -oN scan.txt 192.168.43.103`

### Ejercicio 10. Escanee con un conjunto de scripts de Nmap.

Nmap viene con numerosos y poderosos scripts que se utilizan para escanear vulnerabilidades y, por lo tanto, señalan las debilidades de un sistema. Para obtener la ubicación de los scripts de NSE, simplemente ejecute el comando: `locate *nse` (en sistemas Linux) o busque en la carpeta de instalación de Nmap en Windows.

- `nmap --script mysql-empty-password <dirección IP>`



**Prueba algún script contra alguno de los servicios o servidores que tengáis instalados de otros módulos.**

## Actividad 2. Vulnerabilidades de inyección SQL en OWASP Juice Shop

Vamos a probar varias vulnerabilidades en la aplicación web OWASP Juice Shop. Para ello, necesitamos tener la aplicación instalada y funcionando. Puedes usar la versión online o instalarla en tu equipo.

Si queremos usar la versión online de demo podemos acceder a <https://demo.owasp-juice.shop/#/search>

Por contra, si queremos instalarla en un servidor local, en una máquina Linux, ejecutamos los siguientes comandos:

```
sudo apt-get update
sudo apt-get install juice-shop
```

Además, necesitamos instalar algún proxy para poder interceptar las peticiones y modificarlas. Lo más recomendable es usar Burp Suite.

Vamos a realizar algunos de los retos propuestos en la aplicación.

### Ejercicio2: Accede como usuario administrador

¿Qué sería una aplicación web vulnerable sin una cuenta de usuario administrador cuyos derechos de acceso privilegiados (supuestamente) un hacker exitoso puede abusar?

La descripción del reto probablemente te haya dado una pista sobre la forma en que deberías atacar.

Si ya conoces la dirección de correo electrónico del administrador, puedes lanzar un ataque dirigido.

Podrías tener suerte con un patrón de ataque dedicado incluso si no tienes ni idea de la dirección de correo electrónico del administrador.

Si has obtenido el hash de la contraseña del administrador, puedes intentar atacar eso en lugar de usar la Inyección SQL.

Lo mejor en este caso es resolver este desafío como un combo tratando de iniciar sesión con las credenciales de usuario del administrador sin cambiarlas previamente y aplicar el desafío de Inyección SQL.

### Ejercicio3: Accede con la cuenta del usuario Bender

Bender es un cliente habitual, pero se hace notar por poner reviews en la Juice Shop para trolearla por su falta de bebidas alcohólicas.

La descripción del ejercicio debe darte una pista sobre la forma en que deberías atacar.

Necesitas saber (o adivinar) la dirección de correo electrónico de Bender para poder lanzar un ataque dirigido.

En caso de que intentes algún otro enfoque que no sea la Inyección SQL, notarás que el hash de la contraseña de Bender no es muy útil.

#### **Ejercicio4: Accede con la cuenta del usuario Jim**

Jim es un cliente habitual. Prefiere el zumo de frutas que ningún hombre ha probado antes.

La descripción del reto probablemente te haya dado una pista sobre la forma en que deberías atacar.

Necesitas saber (o adivinar) la dirección de correo electrónico de Jim para poder lanzar un ataque dirigido.

Si has obtenido el hash de la contraseña de Jim, puedes intentar atacar eso en lugar de usar la Inyección SQL.

#### **Ejercicio5: Accede con la cuenta del usuario contable**

En este desafío, debes iniciar sesión con un usuario de contabilidad , pero que no existe realmente. El usuario literalmente necesita ser efímero, así que debe existir un corto período de tiempo.

Registrarse normalmente con la dirección de correo electrónico del usuario obviamente no resolverá este desafío. Juice Shop ni siquiera te permitirá registrarte como acc0unt4nt@juice-sh.op, ya que esto haría que el desafío fuera irresoluble para ti.

Introducir el usuario en la base de datos de alguna otra manera también fallará para resolver este desafío. En caso de que de alguna manera hayas logrado hacerlo, debes reiniciar la aplicación Juice Shop para borrar la base de datos y hacer que el desafío sea resoluble nuevamente.

El hecho de que este desafío esté en la categoría de Inyección ya debería revelar el enfoque previsto.

#### **Ejercicio6: Obtener el SCHEMA de la base de datos**

Un atacante intentaría explotar la Inyección SQL para averiguar tanto como sea posible sobre el esquema de una base de datos. Esto permite SQL Injections mucho más dirigidos, sigilosos y devastadores, como recuperar una lista de todas las credenciales de usuario a través de la Inyección SQL.

Descubre qué sistema de base de datos se está utilizando y dónde suele almacenar sus definiciones de esquema.

Crea una cadena de ataque UNION SELECT para unir los datos relevantes de cualquier tabla del sistema identificada en el resultado original.

Cuando intentamos hacer la inyección de código SQL, obtenemos la consulta que está dando error por haber introducido una ' y a partir de ahí podemos inferir la siguiente cadena de ataque para obtener el esquema de la base de datos:

```
'))-- Para hacer que la consulta no de error
```

y a partir de ahí, podemos ir añadiendo más código SQL para obtener la información que necesitamos:

```
')) UNION SELECT * FROM sqlite_master--  
')) UNION SELECT 1 FROM sqlite_master WHERE type='table'--
```

```
')) UNION SELECT 1,2 FROM sqlite_master WHERE type='table'--  
')) UNION SELECT 1,2,3 FROM sqlite_master WHERE type='table'--
```

Así hasta que el UNION SELECT no de error y obtengamos la información que necesitamos.

Es posible que debas abordar algunos problemas de sintaxis de consulta paso a paso, básicamente saltando de un error al siguiente.

### ? Actividad entregable

Entrega las capturas de pantalla de los pasos realizados en la aplicación OWASP Juice Shop y Burp Suite.

#### Ejercicio7 : Obtener la lista de usuarios

En este caso, partiendo de la información obtenida del `schema` de la BD y usando el mismo método que en el ejercicio anterior, vamos a obtener la lista de usuarios de la base de datos.

Obtén el listado de id, email y password de todos los usuarios de la base de datos.

Recuerda que debes capturar las peticiones y modificarlas con Burp Suite.

## Actividad 3. Vulnerabilidades de XSS en OWASP Juice Shop

Para realizar esta actividad, necesitamos tener la aplicación OWASP Juice Shop instalada y funcionando. Puedes usar la versión online o instalarla en tu equipo.

Versión online: <https://demo.owasp-juice.shop/#/search>

En estas actividades vamos a probar varias vulnerabilidades de tipo XSS en la aplicación web OWASP Juice Shop.

Lo primero que debemos fijarnos es que cuando introducimos un valor en el campo de búsqueda, este valor aparece como un título en la página de resultados. Si introducimos un valor con código HTML, este se ejecuta en la página.

Challenge 1 : DOM XSS

#### Ejercicio1: Realiza un ataque de tipo DOM XSS

Para realizar un ataque de tipo DOM XSS, debemos introducir un valor en el campo de búsqueda que contenga código HTML que se ejecute en la página.

Por ejemplo, si introducimos el siguiente valor en el campo de búsqueda:

```
<script>alert('XSS')</script>
```

Al pulsar en buscar, se ejecutará el código JavaScript y se mostrará una alerta en la página.

Si no se muestra, es posible que tengamos que modificar la petición usando un iframe y un src con el código JavaScript.

```
<iframe src="javascript:alert(`xss`)">
```

#### Ejercicio2 : Bonus Payload

Usa un iframe que cargue/ejecute un contenido multimedia.

```
<iframe width="100%" height="166" scrolling="no" frameborder="no" allow="autoplay" src="https://w.soundcloud.com/play
```

Con este tipo de ataques, podemos hacer que se ejecute código JavaScript en la página, lo que puede ser peligroso si el atacante consigue que un usuario haga clic en un enlace malicioso.

### Ejercicio3: Realiza un ataque de tipo Reflected XSS

Perform a reflected XSS attack with

```
<iframe src="javascript:alert(`xss`)">
```

Para completar este desafío, primero debemos realizar un pedido o acceder a uno de los pedidos realizados por un usuario. A continuación iremos al Historial de pedidos y seleccionamos la opción de seguimiento que se mostrará con una URL similar a esta:

```
https://demo.owasp-juice.shop/#/track-result?id=5267-c5e1891a3b3
```

El parámetro `id` es el que vamos a modificar para realizar el ataque XSS, inyectando el código JavaScript en la URL.

#### ? Actividad entregable

Entrega las capturas de pantalla de los pasos realizados en la aplicación OWASP Juice Shop.

#### Ejercicio4: Realiza un ataque de tipo XSS del lado del servidor

Realiza un ataque de tipo XSS Persistente con `<iframe src="javascript:alert( xss )">` evitando un mecanismo de seguridad del lado del servidor.

Para este desafío, primero intentamos encontrar el campo de entrada que es vulnerable o inyectable y luego intentamos inyectar el payload.

Usaremos la sección de `Comentarios del Cliente (Customer Feedback)` donde tenemos un cuadro de comentarios que se puede usar para inyectar el payload.

Ahora iniciaremos sesión en la cuenta de administrador y accederemos a la página de administración <https://demo.owasp-juice.shop/#/administration> y debería saltar la alerta.

Si el Payload no funciona, usaremos otro Payload o modificaremos el Payload. Podemos verificar la versión del paquete Sanitize aquí, que usa 1.4.2 y, según eso, usaremos nuestro Payload modificado, es decir,

```
<<iframe src="javascript:evil"/>iframe src="javascript:alert(`xss`)">
```

#### ⚡ Ataque efectivo a todos los visitantes de la página

Este último tipo de ataque es el más peligroso, ya que puede afectar a todos los usuarios que visiten la página, no solo al usuario que ha introducido el código malicioso.

## Actividad 4. Squid, ejemplos de ACLs. Ejercicios de entrenamiento

- Sabiendo que Squid cuenta, entre otros, con los siguientes elementos de ACL:
  - src: source (client) IP addresses
  - dst: destination (server) IP addresses
  - dstdomain: destination (server) domain name
  - url\_regex: URL regular expression pattern matching
  - time: time of day, and day of week

Explique las siguientes configuraciones:

### ¿Quién podrá acceder a la web?

```
acl network172 src 172.16.5.0/24
http_access allow network172
```

### ¿Qué está permitido y qué está denegado en la siguiente configuración?

```
acl Cooking1 url_regex cooking
acl Recipe1 url_regex recipe
acl myclients src 172.16.5.0/24

http_access deny Cooking1
http_access deny Recipe1

http_access allow myclients
http_access deny all
```

### ¿Qué está permitido y qué está denegado en la siguiente configuración?

```
acl Cooking2 dstdomain www.gourmet-chef.com

http_access deny Cooking2
http_access allow all
```

### ¿Qué está permitido y qué está denegado en la siguiente configuración?

```
acl ME src 10.0.0.1
acl YOU src 10.0.0.2

http_access allow ME YOU
http_access deny all
```

### ¿Qué está permitido y qué está denegado en la siguiente configuración?

```
acl ME src 10.0.0.1
acl YOU src 10.0.0.2

http_access allow ME
http_access allow YOU
http_access deny all
```

¿Qué está permitido y qué está denegado en la siguiente configuración?

```
acl US src 10.0.0.1 10.0.0.2

http_access allow !US
http_access deny all
```

¿Qué está permitido y qué está denegado en la siguiente configuración?

```
acl GOOD dst 10.0.0.1

http_access allow GOOD
http_access deny all
```

¿Qué está permitido y qué está denegado en la siguiente configuración?

```
acl US src 10.0.0.1 10.0.0.2
acl GOOD dst 10.0.0.5

http_access allow GOOD US
http_access deny all
```

¿Qué está permitido y qué está denegado en la siguiente configuración?

```
acl PornSites url_regex "/usr/local/squid/etc/pornlist"

http_access deny !PornSites
http_access deny all
```

¿Qué está permitido y qué está denegado en la siguiente configuración?

```
acl USER1 src 192.168.100.0/24
acl USER2 src 192.168.200.0/24 192.168.201.0/24
acl DAY time 06:00-18:00

http_access allow USER1 DAY
http_access deny USER1
http_access allow USER2 !DAY
http_access deny USER2
http_access deny all
```