

Informe de Auditoría de Seguridad Web en Aplicación Damn Vulnerable Web Application (DVWA)

Autor(es): Ainhoa López

Fecha: 17/07/2025

2. Resumen Ejecutivo

Este informe detalla el proceso y los hallazgos de una auditoría de seguridad realizada sobre la aplicación web intencionalmente vulnerable, DVWA. El objetivo fue establecer un entorno de laboratorio, familiarizarse con herramientas de la industria y demostrar la identificación y explotación de vulnerabilidades web comunes.

La auditoría se centró en la aplicación DVWA en un entorno controlado de laboratorio y se encontraron vulnerabilidades como: **Cross-Site Scripting (XSS)**, junto con otras debilidades relacionadas con la configuración de cabeceras de seguridad, las cuales están explicadas más adelante.

Este proyecto culmina con la verificación práctica de diversas vulnerabilidades en una aplicación web, superando los desafíos técnicos inherentes al montaje de un laboratorio seguro. La experiencia obtenida con **Docker, ZAP y Burp Suite** es fundamental para comprender las etapas de una auditoría web y la importancia de una defensa en profundidad

3. Introducción

Una auditoría de seguridad web es un examen sistemático y exhaustivo de una aplicación web para identificar vulnerabilidades y debilidades que podrían ser explotadas por atacantes. Su objetivo principal es descubrir fallos de seguridad antes de que actores maliciosos los encuentren y los utilicen para comprometer la aplicación, robar datos o interrumpir servicios.

Una auditoría de seguridad web es un examen sistemático y exhaustivo de una aplicación web para identificar vulnerabilidades y debilidades que podrían ser explotadas por atacantes. Su objetivo principal es descubrir fallos de seguridad antes de que actores maliciosos los encuentren y los utilicen para comprometer la aplicación, robar datos o interrumpir servicios.

¿Qué implica una auditoría de seguridad web?

Generalmente, una auditoría de seguridad web abarca varios aspectos, incluyendo:

- **Análisis de vulnerabilidades (Vulnerability Scanning):** Se utilizan herramientas automatizadas para escanear la aplicación en busca de vulnerabilidades conocidas y comunes (por ejemplo, inyección SQL, Cross-Site Scripting - XSS, etc.).

- **Pruebas de penetración (Penetration Testing o Pentesting):** Un auditor de seguridad (simulando ser un atacante) intenta explotar activamente las vulnerabilidades encontradas o descubrir nuevas. Esto puede implicar:
 - **Pruebas de autenticación y autorización:** Verificar si los controles de acceso son robustos y si los usuarios tienen los permisos adecuados.
 - **Pruebas de inyección:** Intentar inyectar código malicioso en la aplicación (SQL, comandos, etc.).
 - **Pruebas de manejo de sesiones:** Evaluar la seguridad de las sesiones de usuario.
 - **Pruebas de configuración de seguridad:** Revisar la configuración del servidor, la base de datos y la aplicación en busca de puntos débiles.
 - **Análisis de código (Code Review):** En algunos casos, se revisa el código fuente de la aplicación para identificar posibles fallos de seguridad.
- **Revisión de la lógica de negocio:** Se busca identificar fallos en la lógica de la aplicación que podrían ser explotados, incluso si no son vulnerabilidades técnicas clásicas.
- **Evaluación de la infraestructura:** Se pueden evaluar los componentes subyacentes que soportan la aplicación web, como servidores, bases de datos y firewalls.
- **Generación de informes:** Tras la auditoría, se entrega un informe detallado que describe las vulnerabilidades encontradas, su nivel de riesgo y recomendaciones para remediarlas.

Importancia de una auditoría de seguridad web:

La importancia de una auditoría de seguridad web radica en su capacidad para identificar y mitigar riesgos antes de que se conviertan en incidentes de seguridad reales. Sus beneficios clave incluyen:

1. **Protección proactiva:** Permite descubrir debilidades antes de que los atacantes las exploten. Es un enfoque preventivo en lugar de reactivo.
2. **Reducción de riesgos:** Al identificar y corregir vulnerabilidades, se reduce significativamente la superficie de ataque y la probabilidad de sufrir un ciberataque exitoso.
3. **Prevención de pérdidas financieras:** Un ataque exitoso puede resultar en pérdidas económicas significativas debido a:
 - **Robo de datos confidenciales (información de clientes, datos financieros).**
 - **Pérdida de reputación y confianza de los clientes.**
 - **Tiempo de inactividad de la aplicación.**
 - **Costos de remediación y recuperación.**

- Posibles multas por incumplimiento de normativas de protección de datos (ej. GDPR).
- 4. **Cumplimiento normativo:** Muchas normativas y estándares de seguridad (como PCI DSS para procesamiento de tarjetas de crédito o GDPR para protección de datos) exigen auditorías de seguridad regulares. Realizar estas auditorías ayuda a garantizar el cumplimiento y evitar sanciones.
- 5. **Mantenimiento de la reputación:** Un ataque de seguridad puede dañar severamente la reputación de una empresa o marca, lo que puede ser difícil de recuperar. Las auditorías ayudan a proteger esa reputación.
- 6. **Mejora continua de la seguridad:** Las auditorías no solo identifican problemas, sino que también proporcionan información valiosa para mejorar los procesos de desarrollo seguro y las prácticas de seguridad a largo plazo. Ayudan a entender mejor dónde se necesita invertir en seguridad.
- 7. **Conciencia de seguridad:** La realización de auditorías y la corrección de las vulnerabilidades encontradas aumentan la conciencia sobre la seguridad dentro del equipo de desarrollo y la organización en general.

3.1. Objetivos del Proyecto:

- Configurar un laboratorio de auditoría web seguro y funcional.
- Familiarizarse con el uso de herramientas líderes de la industria (Docker, OWASP ZAP, Burp Suite).
- Identificar y, si es posible, explotar vulnerabilidades comunes en una aplicación web intencionalmente vulnerable (DVWA).
- Documentar el proceso y los hallazgos.

El alcance se limitó a la aplicación DVWA ejecutándose localmente en Kali Linux.
Este proyecto se realizó en un entorno de laboratorio controlado y autorizado.

4. Metodología y Configuración del Entorno

- **4.1. Plataforma y Herramientas Utilizadas:**
 - **Sistema Operativo:** Kali Linux
 - **Virtualización:** VMware Workstation
 - **Orquestación de Contenedores:** Docker
 - **Aplicación Vulnerable:** Damn Vulnerable Web Application (DVWA)
 - **Proxy de Interceptación y Escáner de Vulnerabilidades:** OWASP ZAP
 - **Proxy de Interceptación y Suite de Herramientas:** Burp Suite Community Edition
 - **Navegador Web:** Mozilla Firefox

4.2. Instalación y Configuración de Docker:

Docker es una herramienta que empaqueta una aplicación y todas sus dependencias en un entorno aislado conocido como **contenedor**. Esto permite que la aplicación se ejecute de manera consistente en cualquier máquina, independientemente de la configuración del sistema operativo.

Para el caso de **DVWA**, el uso de Docker es estratégico por dos motivos clave:

- **Aislamiento de seguridad:** Al ser una aplicación intencionalmente vulnerable, DVWA podría representar un riesgo para el sistema anfitrión. Al ejecutarla dentro de un contenedor Docker, cualquier vulnerabilidad explotada queda confinada, previniendo daños al sistema operativo subyacente o a otros servicios.
- **Facilidad y rapidez de despliegue:** Docker elimina la complejidad de configurar manualmente un servidor web, una base de datos y PHP. Con un solo comando, se puede levantar todo el entorno de DVWA, lo que simplifica enormemente el proceso de configuración y puesta en marcha del laboratorio de pruebas.

Procedimiento de Instalación de Docker

Los siguientes pasos detallan el proceso de instalación de Docker en un sistema operativo basado en Debian/Ubuntu.

Paso 1: Actualización del Sistema

Antes de instalar cualquier paquete, es fundamental actualizar el sistema para asegurar que se utilicen los repositorios y las dependencias más recientes.

```
sudo apt update && sudo apt upgrade -y
```

Paso 2: Instalación de Docker

Se procede a instalar el paquete `docker.io`, que incluye la plataforma Docker, desde los repositorios oficiales del sistema.

```
sudo apt install -y docker.io
```

Paso 3: Habilitación y Verificación del Servicio

Una vez instalado, es necesario habilitar el servicio de Docker para que se inicie automáticamente en cada arranque del sistema y verificar que esté en funcionamiento.

```
sudo systemctl enable docker --now
```

Paso 4: Configuración de Permisos de Usuario

Para evitar tener que usar `sudo` en cada comando de Docker, se recomienda añadir al usuario actual al grupo `docker`. Esto le otorga los permisos necesarios para gestionar contenedores de forma directa.

```
sudo usermod -aG docker $USER
```

Nota Importante: Para que los permisos del grupo `docker` se apliquen, es imprescindible cerrar la sesión actual y volver a iniciarla, o bien, reiniciar la máquina.

4.3. Despliegue de DVWA con Docker:

Para este proyecto, **DVWA (Damn Vulnerable Web Application)** se ha desplegado como un **contenedor Docker**. Esta aproximación es fundamental y ofrece ventajas significativas para un entorno de pruebas de seguridad:

- **Aislamiento del Entorno:** Al ejecutar DVWA dentro de un contenedor Docker, se crea un entorno completamente aislado del sistema operativo anfitrión. Esto significa que cualquier vulnerabilidad explotada dentro de DVWA (como inyecciones SQL, ejecución de comandos remotos, etc.) queda confinada al contenedor, impidiendo que afecte o comprometa la seguridad del sistema subyacente o de otras aplicaciones en la máquina. Este aislamiento es crítico para practicar ataques de forma segura y sin riesgos para la infraestructura principal.
- **Portabilidad y Facilidad de Despliegue:** Docker simplifica drásticamente el proceso de configuración. En lugar de instalar y configurar manualmente un servidor web (Apache), PHP y una base de datos (MySQL) con todas sus dependencias, un solo comando de Docker es suficiente para desplegar DVWA completamente funcional. Esto asegura que el entorno de pruebas sea consistente y reproducible en cualquier máquina que tenga Docker instalado.
- **Restauración Sencilla:** Si el contenedor de DVWA se corrompe o se ve comprometido durante las pruebas, puede ser fácilmente destruido y

recreado en cuestión de segundos, volviendo a un estado limpio y seguro para continuar con las prácticas.

Este método de despliegue establece una plataforma robusta y segura para realizar pruebas de penetración y aprender sobre vulnerabilidades web sin poner en riesgo ningún sistema real.

2. Pasos para el Despliegue y Configuración Inicial

Paso 1: Ejecución del Contenedor Docker de DVWA

Para iniciar el contenedor de DVWA y hacerlo accesible desde el navegador, se utiliza el siguiente comando en la terminal:

```
docker run --rm -it -p 80:80 vulnerables/web-dvwa
```

Paso 2: Acceso a DVWA y Configuración de la Base de Datos

Una vez que el contenedor está en ejecución, se puede acceder a DVWA abriendo un navegador web y navegando a la siguiente dirección:

```
http://localhost/
```

La primera vez que se accede, DVWA redirigirá automáticamente a la página de configuración de la base de datos (**setup.php**). En esta página, se debe hacer clic en el botón **"Create / Reset Database"** para inicializar la base de datos necesaria para el funcionamiento de la aplicación.

Paso 3: Inicio de Sesión

Tras la configuración exitosa de la base de datos, DVWA redirigirá a la página de inicio de sesión. Las credenciales predeterminadas para acceder son:

- **Usuario:** **admin**
- **Contraseña:** **password**

Paso 4: Importancia de Cambiar el Nivel de Seguridad

Una vez iniciada la sesión, es **crucial** familiarizarse con la sección **"DVWA Security"** (generalmente accesible desde el menú lateral). DVWA permite ajustar el nivel de seguridad de las vulnerabilidades presentes en la aplicación (por ejemplo, "low", "medium", "high", "impossible").

kali linux act - VMware Workstation 16 Player (Non-commercial use only)

Player

Archivo Acciones Editar Vista Ayuda

kali@kali: ~


```
[kali@kali]~$ docker run --rm -it -p 80:80 vulnerables/web-dvwa
[*] Starting mysql ...
[ok] Starting MariaDB database server: mysqld . . . .
[*] Starting apache
[....] Starting Apache httpd web server: apache2AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
. ok
=> /var/log/apache2/access.log <=
=> /var/log/apache2/error.log <=
[Sun Jul 13 16:33:13.911693 2025] [mpm_prefork:notice] [pid 352] AH00163: Apache/2.4.25 (Debian) configured -- resuming normal operations
[Sun Jul 13 16:33:13.911995 2025] [core:notice] [pid 352] AH00094: Command line: '/usr/sbin/apache2'
=> /var/log/apache2/other_vhosts_access.log <=
=> /var/log/apache2/access.log <=
172.17.0.1 - - [13/Jul/2025:16:33:59 +0000] "GET /login.php HTTP/1.1" 200 1192 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [13/Jul/2025:16:34:19 +0000] "POST /login.php HTTP/1.1" 302 337 "http://localhost/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [13/Jul/2025:16:34:19 +0000] "GET /setup.php HTTP/1.1" 200 2041 "http://localhost/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [13/Jul/2025:16:34:26 +0000] "POST /setup.php HTTP/1.1" 302 338 "http://localhost/setup.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [13/Jul/2025:16:34:26 +0000] "GET /setup.php HTTP/1.1" 200 2176 "http://localhost/setup.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
=> /var/log/apache2/error.log <=
[Sun Jul 13 16:34:26.412273 2025] [:error] [pid 361] [client 172.17.0.1:35480] PHP Notice: Constant DVWA_WEB_PAGE_TO_ROOT already defined in /var/www/html/dvwa/includes/DBMS/MySQL.php on line 9, referer: http://localhost/setup.php
=> /var/log/apache2/access.log <=
172.17.0.1 - - [13/Jul/2025:16:34:31 +0000] "GET /login.php HTTP/1.1" 200 1050 "http://localhost/setup.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [13/Jul/2025:16:34:34 +0000] "POST /login.php HTTP/1.1" 302 336 "http://localhost/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
```

kali linux act - VMware Workstation 16 Player (Non-commercial use only)

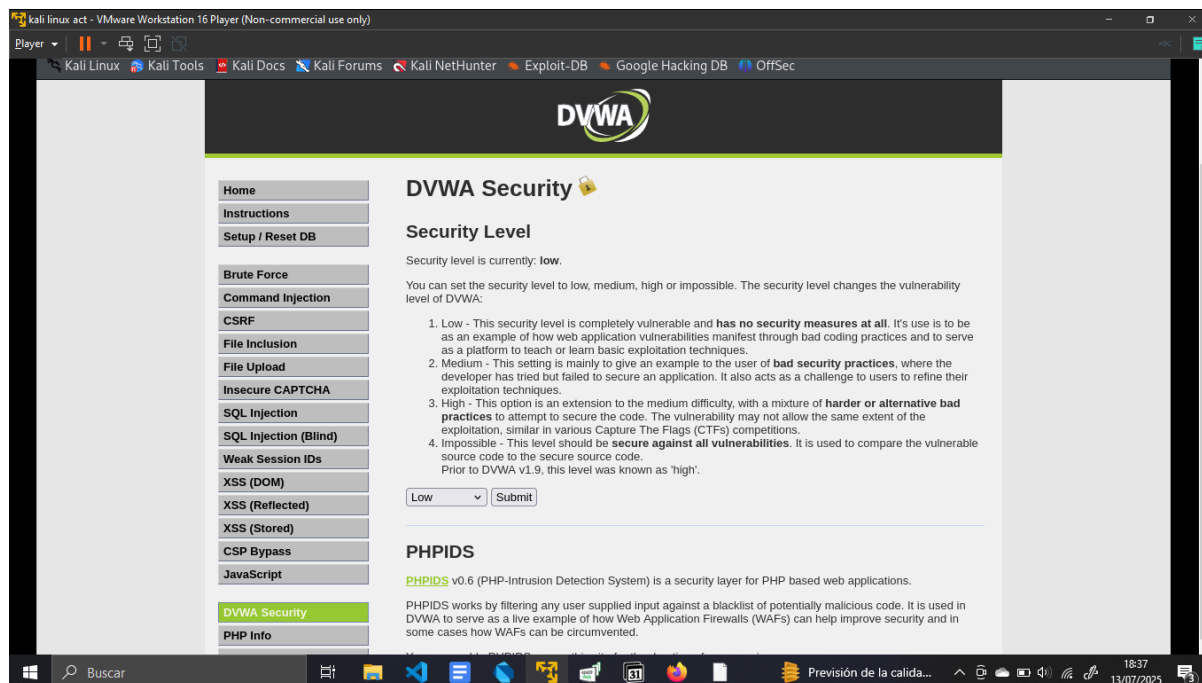
Player

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

PHP Version 7.0.30-0+deb9u1



| | |
|---|--|
| System | Linux 6134d6d3f061 6.12.33+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.33-1kali1 (2025-06-25) x86_64 |
| Build Date | Jun 14 2018 13:50:25 |
| Server API | Apache 2.0 Handler |
| Virtual Directory Support | disabled |
| Configuration File (php.ini) Path | /etc/php/7.0/apache2 |
| Loaded Configuration File | /etc/php/7.0/apache2/php.ini |
| Scan this dir for additional .ini files | /etc/php/7.0/apache2/conf.d |
| Additional .ini files parsed | /etc/php/7.0/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.0/apache2/conf.d/10-opcache.ini, /etc/php/7.0/apache2/conf.d/10-pdo.ini, /etc/php/7.0/apache2/conf.d/15-xml.ini, /etc/php/7.0/apache2/conf.d/20-calendar.ini, /etc/php/7.0/apache2/conf.d/20-ctype.ini, /etc/php/7.0/apache2/conf.d/20-dom.ini, /etc/php/7.0/apache2/conf.d/20-exif.ini, /etc/php/7.0/apache2/conf.d/20-fileinfo.ini, /etc/php/7.0/apache2/conf.d/20-ftp.ini, /etc/php/7.0/apache2/conf.d/20-gd.ini, /etc/php/7.0/apache2/conf.d/20-gettext.ini, /etc/php/7.0/apache2/conf.d/20-iconv.ini, /etc/php/7.0/apache2/conf.d/20-json.ini, /etc/php/7.0/apache2/conf.d/20-mysql.ini, /etc/php/7.0/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.0/apache2/conf.d/20-pdo_pgsql.ini, /etc/php/7.0/apache2/conf.d/20-pgsql.ini, /etc/php/7.0/apache2/conf.d/20-phar.ini, /etc/php/7.0/apache2/conf.d/20-posix.ini, /etc/php/7.0/apache2/conf.d/20-readline.ini, /etc/php/7.0/apache2/conf.d/20-shmop.ini, /etc/php/7.0/apache2/conf.d/20-simplexml.ini, /etc/php/7.0/apache2/conf.d/20-sockets.ini, /etc/php/7.0/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.0/apache2/conf.d/20-sysvshm.ini, /etc/php/7.0/apache2/conf.d/20-tokenizer.ini, /etc/php/7.0/apache2/conf.d/20-wddx.ini, /etc/php/7.0/apache2/conf.d/20-xmlreader.ini, /etc/php/7.0/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.0/apache2/conf.d/20-xsl.ini |
| PHP API | 20151012 |
| PHP Extension | 20151012 |
| Zend Extension | 320151012 |
| Zend Extension Build | API320151012.NTS |
| PHP Extension Build | API20151012.NTS |
| Debug Build | no |



4.4. Configuración de OWASP ZAP:

OWASP ZAP (Zed Attack Proxy) es una herramienta de código abierto ampliamente utilizada en el ámbito de la seguridad web. Su rol principal es actuar como un **proxy intermediario** y un **escáner de vulnerabilidades** para aplicaciones web.

Como **proxy intermediario**, ZAP se sitúa entre tu navegador web y la aplicación web que estás probando. Esto le permite interceptar, inspeccionar, modificar y reenviar todas las solicitudes y respuestas HTTP/HTTPS que fluyen entre ellos. Esta capacidad es crucial para entender cómo interactúa una aplicación, detectar parámetros ocultos, y manipular datos en tránsito.

Además de su función de proxy, ZAP incorpora un potente **escáner de vulnerabilidades**. Puede analizar automáticamente la aplicación web en busca de fallos de seguridad conocidos (como inyección SQL, Cross-Site Scripting - XSS, configuración incorrecta, etc.) tanto de forma pasiva (analizando el tráfico ya interceptado) como activa (enviando solicitudes maliciosas para provocar errores y revelar vulnerabilidades). En conjunto, estas funcionalidades hacen de ZAP una herramienta indispensable para auditar la seguridad de aplicaciones web.

A continuación, se detallan los pasos para instalar y configurar OWASP ZAP, incluyendo la configuración de tu navegador para trabajar con él.

Paso 1: Instalación de OWASP ZAP

Para instalar OWASP ZAP en tu sistema (en distribuciones basadas en Debian/Ubuntu), puedes usar el siguiente comando:

```
sudo apt install -y zaproxy
```

Este comando descargará e instalará la última versión estable de ZAP disponible en los repositorios de tu sistema.

Paso 2: Configuración del Proxy de ZAP

Una vez instalado, inicia OWASP ZAP. Por defecto, ZAP configura su proxy en **localhost** (o **127.0.0.1**) en el puerto **8080**. Es fundamental verificar esta configuración para asegurar que tu navegador pueda conectarse correctamente a ZAP.

1. Abre **OWASP ZAP**.
2. Ve a **"Tools"** (Herramientas) en la barra de menú superior.
3. Selecciona **"Options..."** (Opciones...).
4. En el árbol de la izquierda, busca y selecciona **"Local Proxies"** (Proxies locales).
5. Asegúrate de que la dirección del proxy sea **127.0.0.1** y el puerto sea **8080**. Si no es así, configúralo manualmente.

Paso 3: Configuración del Navegador Firefox para Usar ZAP como Proxy

Para que ZAP pueda interceptar el tráfico de tu navegador, debes configurar Firefox para que envíe todas sus solicitudes a través del proxy de ZAP.

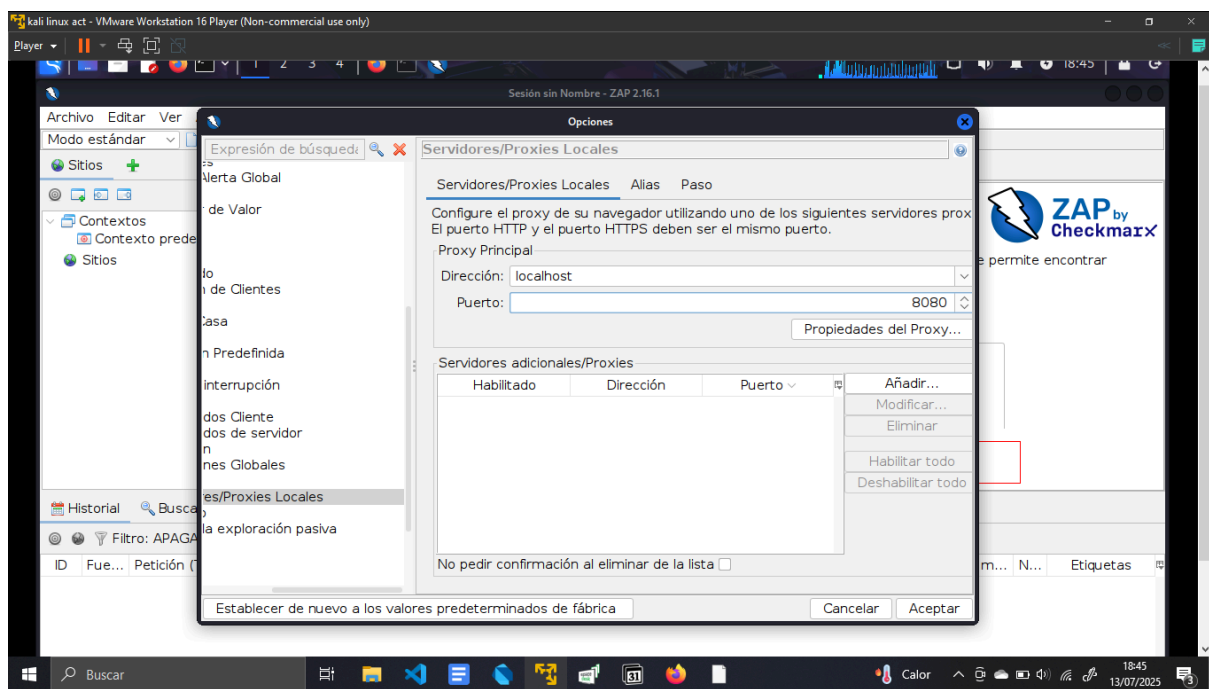
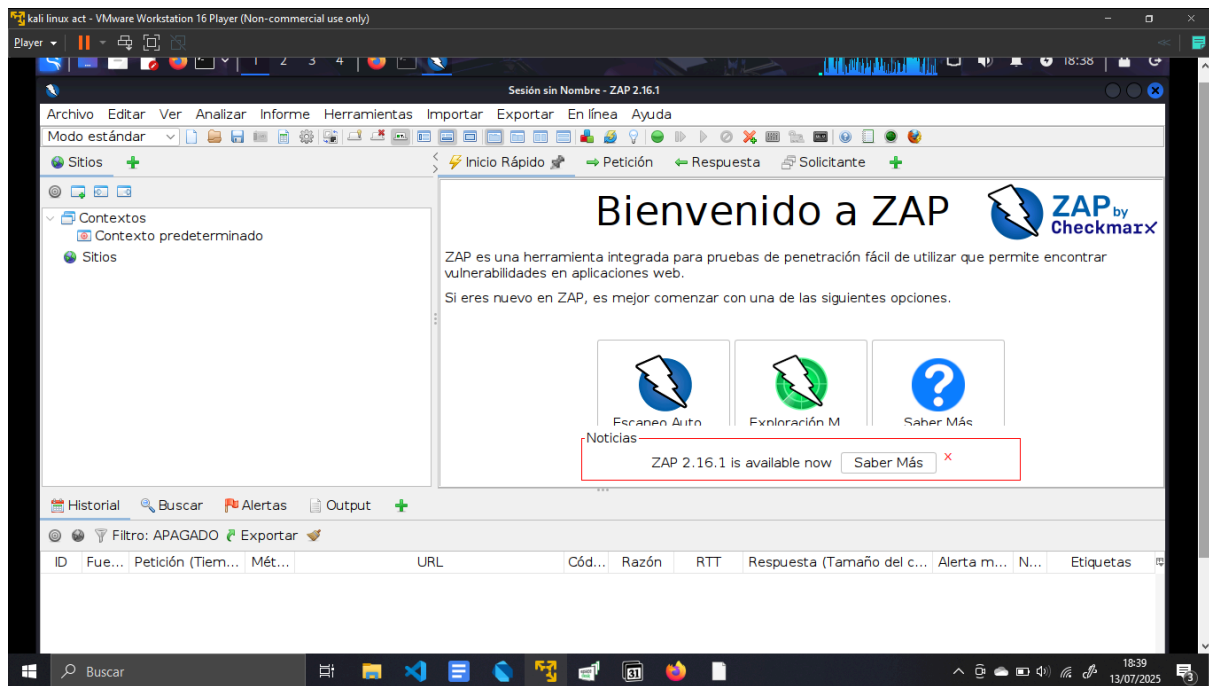
1. Abre **Firefox**.
2. Haz clic en el **icono del menú** (tres líneas horizontales) en la esquina superior derecha.
3. Selecciona **"Settings"** (Ajustes) o **"Opciones"**.
4. En la barra lateral izquierda, selecciona **"General"**.
5. Desplázate hacia abajo hasta la sección **"Network Settings"** (Ajustes de red) y haz clic en el botón **"Settings..."** (Configuración...).
6. En la ventana de configuración del proxy, selecciona la opción **"Manual proxy configuration"** (Configuración manual del proxy).
7. En los campos **"HTTP Proxy"** y **"SSL Proxy"**, introduce **127.0.0.1** y en el campo **"Port"** (Puerto), introduce **8080**.

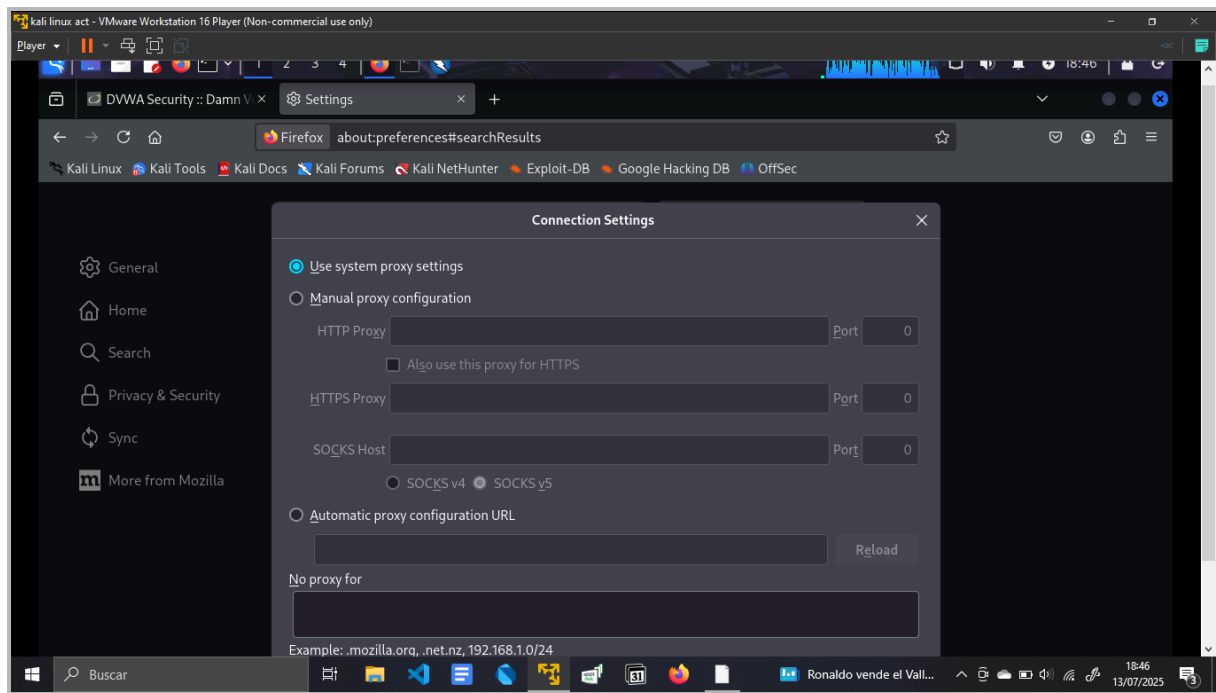
8. Asegúrate de marcar la opción **"Use this proxy server for all protocols"** (Usar este servidor proxy para todos los protocolos).
9. Haz clic en **"OK"** para guardar los cambios.

Paso 4: Importación e Instalación del Certificado SSL de ZAP en Firefox

Cuando ZAP intercepta tráfico HTTPS, genera sus propios certificados SSL/TLS para poder descifrarlo. Para evitar advertencias de seguridad en el navegador y permitir que ZAP intercepte correctamente el tráfico cifrado, debes importar el certificado raíz de ZAP en el almacén de confianza de Firefox.

1. En **OWASP ZAP**, ve a **"Tools"** (Herramientas) en la barra de menú superior.
2. Selecciona **"Options..."** (Opciones...).
3. En el árbol de la izquierda, busca y selecciona **"Dynamic SSL Certificates"** (Certificados SSL Dinámicos).
4. Haz clic en el botón **"Save..."** (Guardar...) y guarda el certificado (por ejemplo, **owasp_zap_root_ca.cer**) en una ubicación accesible en tu sistema (ej. tu carpeta de inicio o Escritorio).
5. En **Firefox**, ve al **icono del menú** (tres líneas horizontales) y selecciona **"Settings"** (Ajustes) o **"Opciones"**.
6. En la barra lateral izquierda, selecciona **"Privacy & Security"** (Privacidad y Seguridad).
7. Desplázate hacia abajo hasta la sección **"Certificates"** (Certificados) y haz clic en el botón **"View Certificates..."** (Ver certificados...).
8. En la pestaña **"Authorities"** (Autoridades), haz clic en **"Import..."** (Importar...).
9. Navega hasta la ubicación donde guardaste el certificado de ZAP (**owasp_zap_root_ca.cer**), selecciónalo y haz clic en **"Open"** (Abrir).
10. En la ventana emergente, asegúrate de marcar la opción **"Trust this CA to identify websites."** (Confiar en esta CA para identificar sitios web.).
11. Haz clic en **"OK"** para importar el certificado y luego **"OK"** nuevamente para cerrar la ventana de certificados.





4.5. Configuración de Burp Suite Community Edition:

Burp Suite Community Edition es una herramienta esencial en el arsenal de cualquier profesional de la seguridad web y un pilar fundamental en las pruebas de penetración. Al igual que OWASP ZAP, su rol principal es actuar como un **proxy interceptor** y un conjunto de herramientas para la **manipulación de solicitudes y respuestas web**.

Como **proxy**, Burp Suite se posiciona entre tu navegador web y la aplicación objetivo (en este caso, DVWA). Esto le permite capturar, visualizar, analizar y modificar todo el tráfico HTTP y HTTPS que fluye en ambas direcciones. Esta capacidad es invaluable para:

- **Inspección Detallada:** Examinar cada byte de las solicitudes enviadas por el navegador y las respuestas del servidor.
- **Manipulación de Datos:** Alterar parámetros, encabezados, cookies y otros datos en tiempo real antes de que lleguen al servidor o al navegador, lo que es crucial para probar vulnerabilidades como inyección SQL o XSS.
- **Identificación de Patrones:** Comprender cómo se comunica la aplicación, revelando rutas ocultas, funciones y posibles puntos de entrada para ataques.

Además de su potente proxy, Burp Suite Community Edition incluye herramientas básicas para el **escaneo pasivo de vulnerabilidades**, el **descubrimiento de contenido oculto** y la **repetición de solicitudes**, lo que lo convierte en una plataforma integral para la evaluación manual de la seguridad de aplicaciones web.

Paso 1: Proceso de Descarga e Instalación

Burp Suite Community Edition se descarga directamente desde el sitio web de PortSwigger (los desarrolladores de Burp Suite).

1. Descarga:

- Abre tu navegador y navega a la página de descarga de Burp Suite Community Edition en el sitio web de PortSwigger.
- Selecciona la versión adecuada para tu sistema operativo (generalmente un archivo `.sh` para sistemas Linux).
- Descarga el archivo a una ubicación de tu elección (ej., tu carpeta `Downloads`).

2. Permisos de Ejecución:

- Abre una terminal.
- Navega al directorio donde descargaste el archivo:

```
cd ~/Downloads
```

Concede permisos de ejecución al archivo `.sh`:

```
chmod +x burpsuite_community_linux_vXXXX_X.sh
```

- (Reemplaza `burpsuite_community_linux_vXXXX_X.sh` con el nombre exacto del archivo descargado).

3. Ejecución del Instalador:

- Ejecuta el script de instalación:

```
./burpsuite_community_linux_vXXXX_X.sh
```

- El instalador gráfico te guiará a través de los pasos de instalación. Simplemente sigue las instrucciones, aceptando los términos y condiciones, y eligiendo la ubicación de instalación deseada.

Paso 2: Configuración del Proxy de Burp Suite

Una vez instalado, inicia Burp Suite. Por defecto, Burp configura su proxy para escuchar en `localhost` (o `127.0.0.1`) en el puerto `8080`. Es vital verificar y confirmar esta configuración.

1. Abre **Burp Suite Community Edition**.
2. En la interfaz principal, ve a la pestaña **"Proxy"**.
3. Dentro de la pestaña "Proxy", selecciona la sub-pestaña **"Options"** (Opciones).

4. Bajo la sección "**Proxy Listeners**" (Escuchas del Proxy), deberías ver una entrada activa con la dirección IP **127.0.0.1** y el puerto **8080**. Asegúrate de que su casilla de verificación esté marcada (activa).

Paso 3: Gestión del Conflicto de Puertos (Burp Suite y ZAP)

Dado que tanto Burp Suite como OWASP ZAP utilizan por defecto el puerto **8080** para sus proxys, se debe gestionar este conflicto.

Alternancia del Proxy en Firefox

Este método es más sencillo y consiste en cambiar manualmente la configuración del proxy en Firefox cada vez que quieras usar una herramienta diferente.

1. **Burp Suite Activo:** Cuando quieras usar Burp Suite, configura Firefox para que su proxy HTTP y SSL apunte a **127.0.0.1:8080**.
2. **ZAP Activo:** Cuando quieras usar OWASP ZAP, configura Firefox para que su proxy HTTP y SSL apunte a **127.0.0.1:8080** (asegurándote de que Burp Suite no esté activo o haya cambiado su puerto).

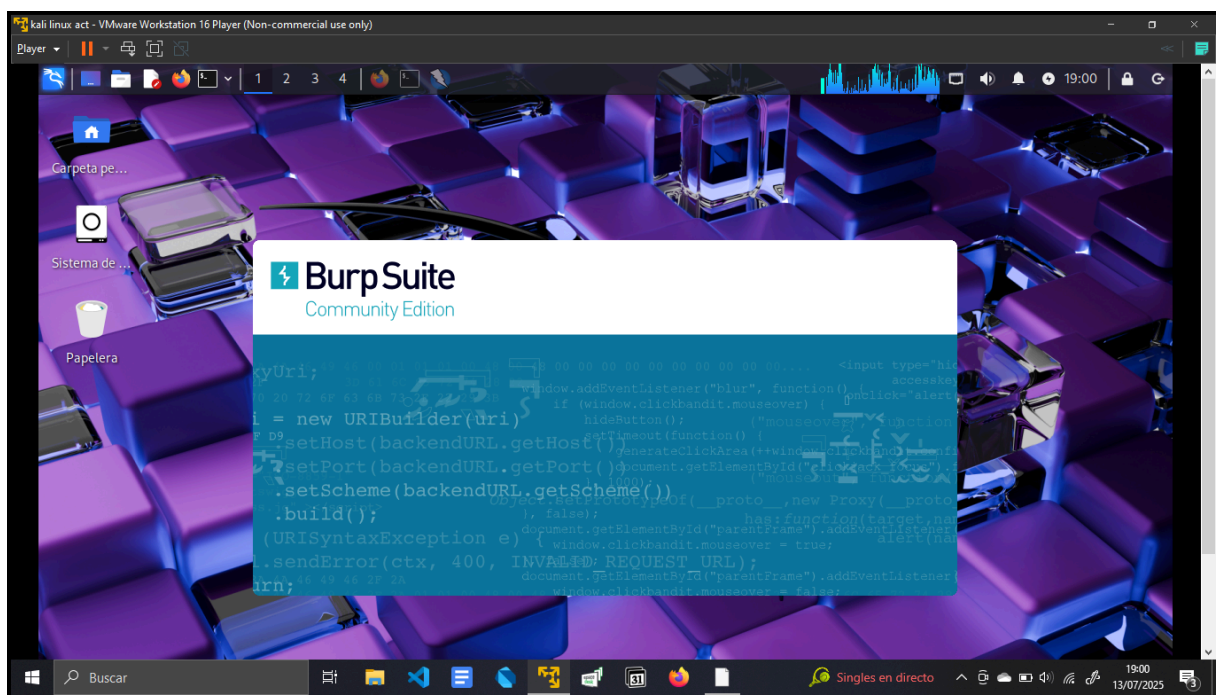
Este enfoque evita conflictos de puertos al asegurar que solo una herramienta esté escuchando en **8080** en un momento dado. Es menos automatizado pero más directo para entornos de prueba básicos. Para este proyecto, se optó por la **alternancia del proxy en Firefox**, configurando manualmente el proxy del navegador según la herramienta que se fuese a utilizar en cada momento, para simplificar la gestión del flujo de tráfico.

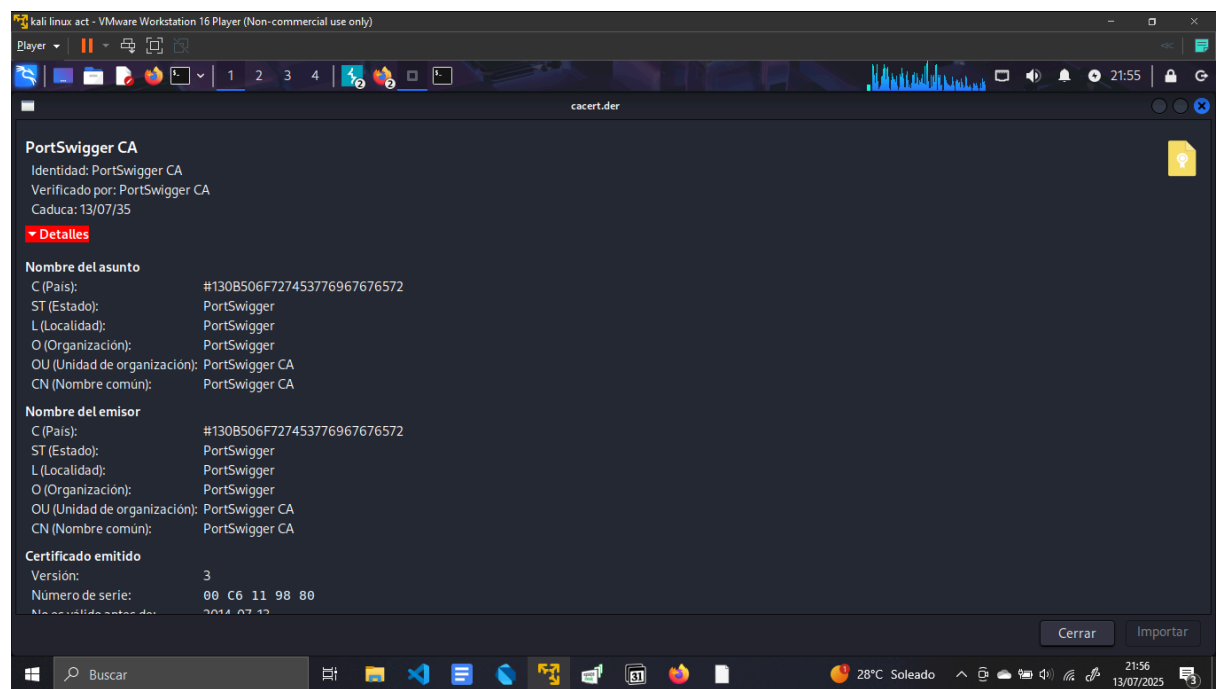
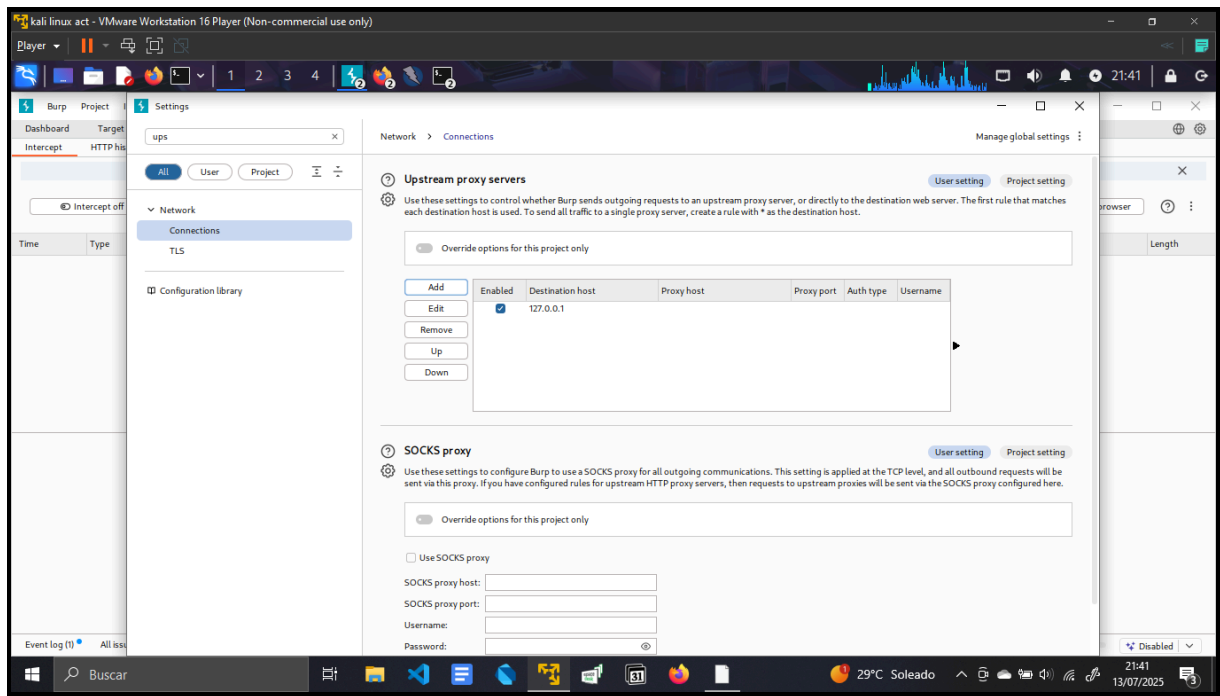
Paso 4: Descarga e Instalación del Certificado SSL de Burp Suite en Firefox

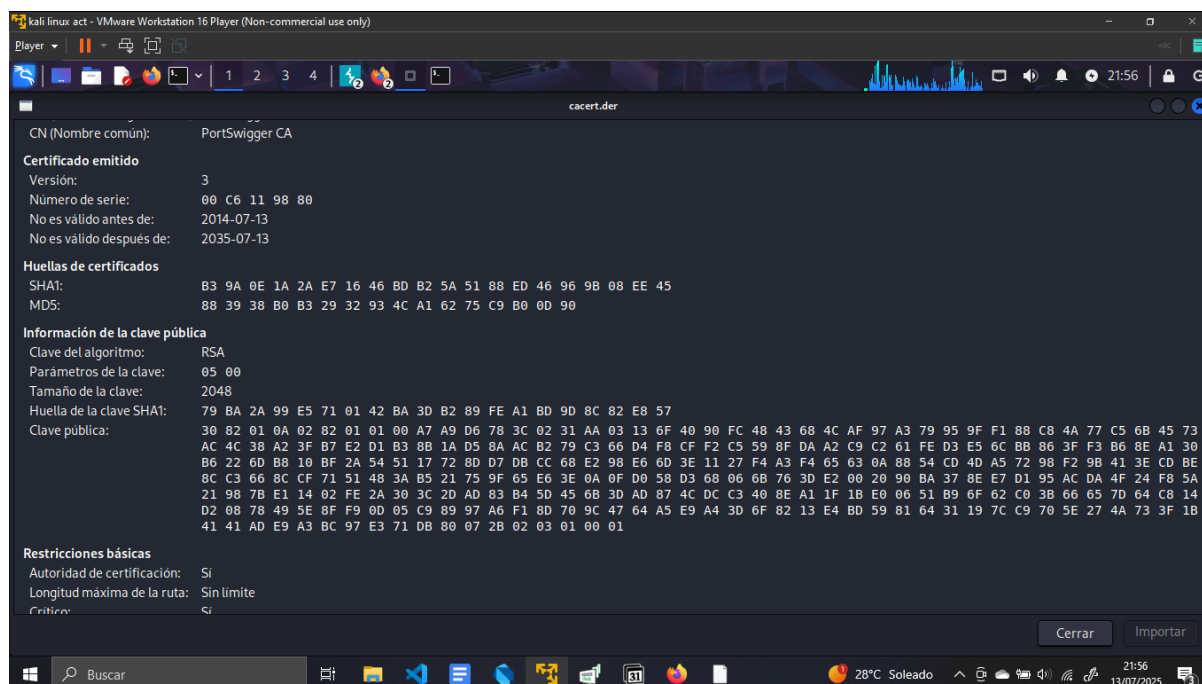
Para que Burp Suite pueda interceptar y descifrar el tráfico HTTPS sin generar advertencias de seguridad en el navegador, es necesario instalar su certificado raíz de confianza en Firefox.

1. **Acceso al Certificado:**
 - Con Burp Suite en ejecución y configurado como proxy en Firefox, abre tu navegador.
 - Navega a la siguiente URL: **http://burp/cert**
 - Esto te permitirá descargar el certificado de CA de Burp (normalmente **cacert.der**). Guarda el archivo en una ubicación fácil de recordar.
2. **Importación en Firefox:**
 - En **Firefox**, haz clic en el **icono del menú** (tres líneas horizontales) en la esquina superior derecha y selecciona **"Settings"** (Ajustes) o **"Opciones"**.

- En la barra lateral izquierda, selecciona **"Privacy & Security"** (Privacidad y Seguridad).
- Desplázate hacia abajo hasta la sección **"Certificates"** (Certificados) y haz clic en el botón **"View Certificates..."** (Ver certificados...).
- En la pestaña **"Authorities"** (Autoridades), haz clic en **"Import..."** (Importar...).
- Navega hasta la ubicación donde guardaste el archivo **cacert.der** de Burp, selecciónalo y haz clic en **"Open"** (Abrir).
- En la ventana emergente, marca la opción **"Trust this CA to identify websites."** (Confiar en esta CA para identificar sitios web.).
- Haz clic en **"OK"** para importar el certificado y luego **"OK"** nuevamente para cerrar las ventanas de configuración.







5. Hallazgos de la Auditoría y Pruebas de Concepto (PoC)

·Vulnerabilidad N° 1: Cabecera Content Security Policy (CSP) no configurada(3)

·URL: <http://localhost/>

·Riesgos: Medium

·Confianza: High

·CWE ID: 693

·WASC ID: 15

·Origen: Pasivo (10038-CSP no configurada)

·Referencia de Alerta: 10038-1

·Descripción:

La política de seguridad de Contenido (CSP) es una capa adicional de seguridad que ayuda a detectar y mitigar ciertos tipos de ataques, incluidos **Cross Site Scripting (XSS)** y **Ataques de Inyección de Datos**.

Estos ataques se utilizan para todo, desde el robo de datos hasta la desfiguración del sitio o la distribución de malware. **CSP** proporciona un conjunto de encabezados

HTTP estándar que permiten a los propietarios de sitios web declarar fuentes de contenido aprobadas que los navegadores deberían poder cargar en esa página, los tipos cubiertos son: **JavaScript, CSS, marcas HTML, fuentes, imágenes y objetos incrustados como applets de Java, Active X, archivos de audio y vídeo.**

·**Solución:**

Asegúrese que su servidor web o de aplicaciones, balanceado de carga...etc, esté configurado para establecer la cabecera CSP.

·**Etiquetas de Alerta:**

| Clave | Valor |
|----------------|---|
| CWE-693 | https://cwe.mitre.org/data/definitions/693.html |
| OWASP_2021_AO5 | https://owasp.org/Top10/A05_2021-Security_Misconfiguration/ |
| OWASP_2017_AO6 | https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration |

·**Vulnerabilidad N°2: Falta de Cabecera Anti-Clickjacking (2)**

·**URL:** http://localhost/

·**Riesgos:** Medium

·**Confianza:** Medium

·**Parámetro:** X-frame-options

·**CWE ID:** 1021

·**WASC ID:** 15

·**Origen:** Pasivo (10020-Cabecera Anti-Clickjacking)

·**Referencia de Alerta:** 10020-1

·**Descripción:**

Es una vulnerabilidad de seguridad web que ocurre cuando una aplicación no implementa mecanismos para prevenir ataques de **Clickjacking**.

El **Clickjacking**, también conocido como **"UI Redressing"** (rediseño de la interfaz de usuario), es una técnica maliciosa en la que un atacante engaña a un usuario para que haga clic en un elemento invisible o disfrazado en una página web. Este clic, sin que el usuario lo sepa, realiza una acción no deseada en un sitio web diferente o en una aplicación vulnerable.

·Solución:

La principal defensa contra el Clickjacking es la implementación de la cabecera HTTP **X-Frame-Options** en las respuestas del servidor. Esta cabecera le indica al navegador si la página web puede ser incrustada dentro de un `<iframe>`, `<frame>`, `<embed>` o `<object>`.

Los valores comunes para esta cabecera son:

·**X-Frame-Options: DENY**: La página no puede ser mostrada en un `<iframe>` o `<frame>` bajo ninguna circunstancia, ni siquiera si el sitio que la incrusta es del mismo dominio. Es la opción más segura.

·**X-Frame-Options: SAMEORIGIN**: La página solo puede ser mostrada en un `<iframe>` o `<frame>` si el sitio que la incrusta es del mismo dominio (mismo origen). Esta opción es útil si necesitas incrustar contenido de tu propio sitio.

·**X-Frame-Options: ALLOW-FROM uri**: Permite incrustar la página solo desde un URI específico. Esta opción ha sido deprecada en favor de la cabecera **Content-Security-Policy: frame-ancestors**.

·Etiquetas de Alerta:

| Clave | Valor |
|------------------|---|
| WSTG-v92-CLNT-04 | https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/07-Client_Side_Testing/04-Test_Client-side_URL_Redirect.html |
| OWASP_2021_AO5 | https://owasp.org/Top10/A05_2021-Security_Misconfiguration/ |
| OWASP_2017_AO6 | https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration |
| CWE-1021 | https://cwe.mitre.org/data/definitions/1021.html |

·Vulnerabilidad N°3: Cookie Sin Flag HttpOnly (2)

·URL: http://localhost/

·Riesgos: Low

·Confianza: Medium

·Parámetro: PHPSESSID

·CWE ID: 1004

·WASC ID: 13

·Origen: Pasivo (10010-Cookie Sin Flag HttpOnly)

·Referencia de Alerta: 10010-1

·Descripción:

Se ha establecido una cookie sin el flag **HttpOnly**, lo que significa que JavaScript puede acceder a la cookie. Si un script malicioso puede ser ejecutado en esta página, entonces la cookie será accesible y puede ser transmitida a otro sitio. Si se trata de una cookie de sesión, el secuestro de sesión puede ser posible.

·Solución:

Asegúrese de que la flag **HttpOnly** está establecida para todas las cookies.

·Etiquetas de Alerta:

| Clave | Valor |
|------------------|---|
| CWE-1004 | https://cwe.mitre.org/data/definitions/1004.html |
| OWASP_2021_AO5 | https://owasp.org/Top10/A05_2021-Security_Misconfiguration/ |
| OWASP_2017_AO6 | https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration |
| WSTG-v42-SESS-02 | https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/06-Session_Management_Testing/02-Test_Session_Fixation.html |

·Vulnerabilidad N°4: Cookie sin el atributo SameSite (2):

·**URL:** http://localhost/

·**Riesgos:** Low

·**Confianza:** Medium

·**Parámetro:** PHPSESSID

·**Evidencia:** Set-Cookie: PHPSESSID

·**CWE ID:** 1275

·**WASC ID:** 13

·**Origen:** Pasivo (10054-Cookie Sin el atributo SameSite)

·**Referencia de Alerta:** 10054-1

·**Descripción:**

Se ha establecido una cookie sin el atributo **SameSite**, lo que significa que la cookie puede ser enviada como resultado de una solicitud '**cross-site**'. El atributo **SameSite** es una medida eficaz para contrarrestar la falsificación de peticiones entre sitios, la inclusión de scripts entre sitios y los ataques de sincronización.

·**Solución:**

Asegúrese de que el atributo **SameSite** está establecido como '**lax**' o idealmente '**strict**' para todas las cookies.

·**Etiquetas de Alerta:**

| Clave | Valor |
|------------------|--|
| OWASP_2021_AO5 | https://owasp.org/Top10/A05_2021-Security_Misconfiguration/ |
| OWASP_2017_AO6 | https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration |
| WSTG-v42-SESS-02 | https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/06-Session_Man |

| | |
|-----------------|---|
| | agement_Testing/02-Test_Session_Fixation.html |
| CWE-1275 | https://cwe.mitre.org/data/definitions/1275.html |

·Vulnerabilidad N°5: El servidor filtra información de versión a través del campo ‘Server’ del encabezado HTTP

·**URL:** http://localhost/

·**Riesgos:** Low

·**Confianza:** High

·**Evidencia:** Apache/2.4.25 (Debian)

·**CWE ID:** 497

·**WASC ID:** 13

·**Origen:** Pasivo (10036-Cabecera de Respuesta del Servidor HTTP)

·**Referencia de Alerta:** 10036-1

·**Descripción:**

El servidor web/aplicación está filtrando información de versión a través de la cabecera de respuesta HTTP “Server”. El acceso a dicha información puede facilitar a los atacantes la identificación de otras vulnerabilidades a las que está sujeto su servidor web/aplicación.

·**Solución:**

Asegúrese de que su servidor web, servidor de aplicaciones, balanceador de carga,etc, está configurado para suprimir la cabecera ‘Server’ o proporcionar detalles genéricos.

·**Etiquetas de Alerta:**

| Clave | Valor |
|-----------------------|---|
| OWASP_2021_AO5 | https://owasp.org/Top10/A05_2021-Security_Misconfiguration/ |

| | |
|-------------------------|---|
| OWASP_2017_AO6 | https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration |
| WSTG-v42-INFO-02 | https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/01-Information_Gathering/02-Fingerprint_Web_Server.html |
| CWE-497 | https://cwe.mitre.org/data/definitions/497.html |

·Vulnerabilidad N°6: Falta encabezado X-Content-Type-Options (6)

·URL: <http://localhost/>

·Riesgos: Low

·Confianza: Medium

·Parámetro: x-content-type-options

·CWE ID: 693

·WASC ID: 15

·Origen: Pasivo (10021-Falta encabezado X-Content-Type-Options)

·Referencia de Alerta: 10021-1

·Descripción:

La cabecera ***Anti-MME-Sniffing X-Content-Type-Options*** no se ha establecido en 'nosniff'. Esto permite que las versiones anteriores de Internet Explorer y Chrome realicen ***MME-Sniffing*** en el cuerpo de la respuesta, lo que puede provocar que el cuerpo de la respuesta se interprete y se muestre como un tipo de contenido distinto al tipo de contenido declarado. Las versiones actuales y heredados de Firefox utilizarán el tipo de contenido declarado (si se establece uno), en lugar de realizar ***MME-Sniffing***.

·Solución:

Asegúrese de que la aplicación/servidor web establece el encabezado ***Content-Type*** adecuadamente, y que establece el encabezado ***X-Content-Type-Options*** a '***nosniff***' para todas las páginas web. Si es posible asegúrese de que el usuario final utilice un navegador web moderno y compatible

con los estándares que no realiza MME-sniffing en absoluto, o que puede ser dirigido por la aplicación web/servidor para que no realice MME-sniffing.

·**Etiquetas de Alerta:**

| Clave | Valor |
|----------------|---|
| CWE-693 | https://cwe.mitre.org/data/definitions/693.html |
| OWASP_2021_AO5 | https://owasp.org/Top10/A05_2021-Security_Misconfiguration/ |
| OWASP_2017_AO6 | https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration |

·**Vulnerabilidad N°7: Vulnerabilidad: Cross-Site Scripting (XSS) Reflejado**

·**Identificador de Alerta (ID):** 10017

·**Riesgo:** High

·**Confianza:** High

·**URL Afectada:** http://localhost/vulnerabilities/xss_r/

·**Parámetro Afectado:** name

·**Origen de la Alerta:** Passive Scan

·**Referencia de alerta:**

·**Descripción:**

El **Cross-Site Scripting (XSS) Reflejado** es una vulnerabilidad de inyección de código que ocurre cuando una aplicación web toma datos de una petición HTTP y los incluye de forma no segura en la respuesta HTML sin una adecuada sanitización o codificación. Esto permite a un atacante inyectar scripts maliciosos (generalmente JavaScript) que se ejecutan en el navegador de la víctima cuando ésta accede a la URL manipulada. El código inyectado se "refleja" de vuelta al usuario en la respuesta del servidor.

El impacto de un ataque de XSS es significativo y puede incluir: **Robo de Cookies de sesión, Defacement del Sitio, Redirecciones Maliciosas y Ejecución de código arbitrario.**

Pasos para Reproducir (Prueba de Concepto - PoC) en DVWA (Nivel de Seguridad: Low):

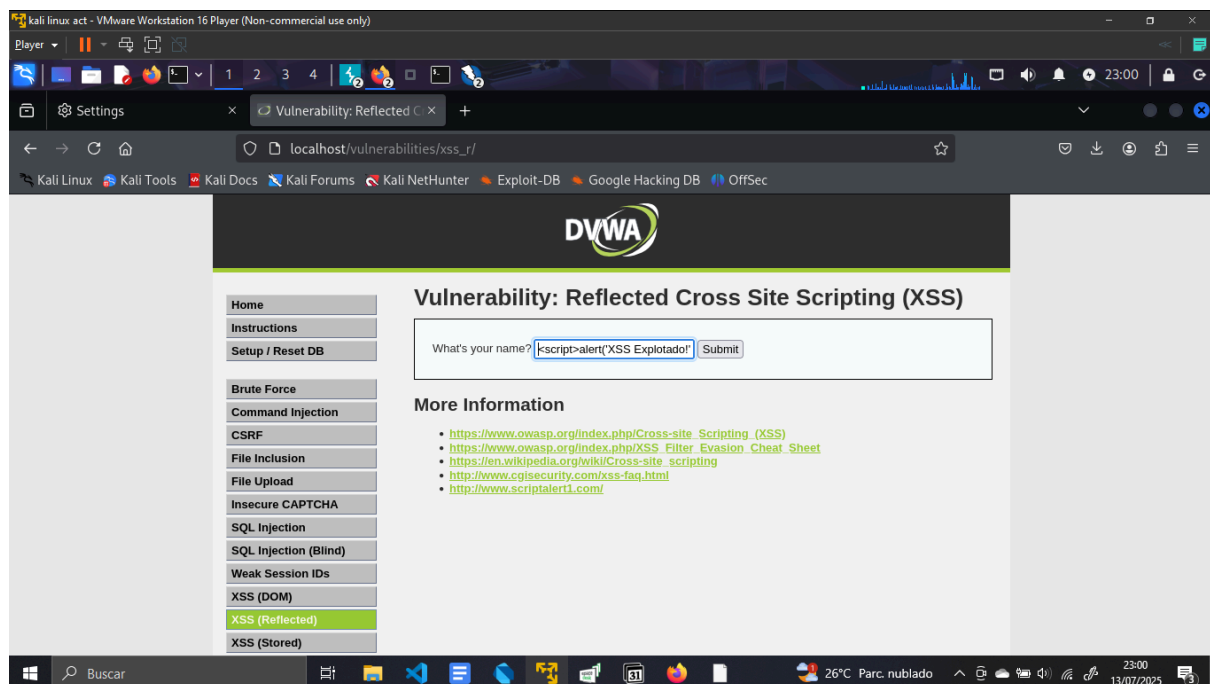
1. **Configuración de Seguridad:** Asegurarse de que el nivel de seguridad de DVWA esté configurado en "Low" (Bajo) a través de la sección "DVWA Security".
2. **Navegación:** Acceder a la sección **XSS (Reflected)** de DVWA en la URL: http://localhost/vulnerabilities/xss_r/.

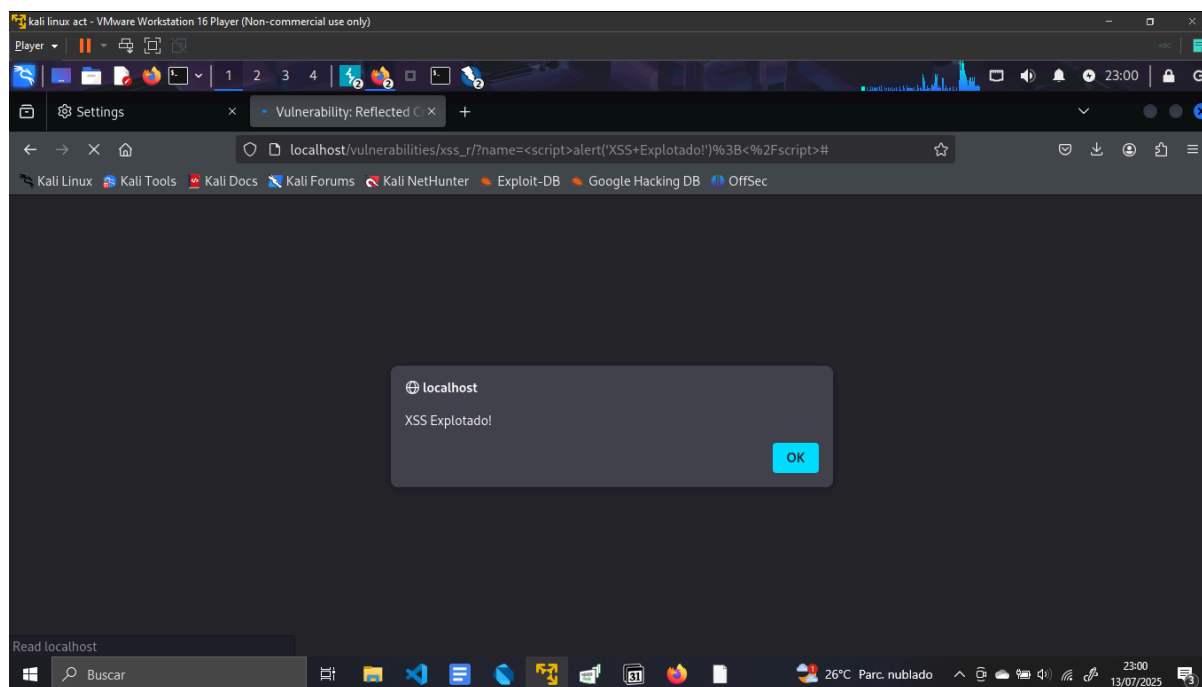
Inyección del Payload: En el campo de entrada etiquetado como "Enter your name:", introducir el siguiente payload de JavaScript:

HTML

```
<script>alert('XSS Explotado!');</script>
```

3. **Envío de la Petición:** Hacer clic en el botón "Submit" (Enviar).
4. **Observación de la Explotación:** Inmediatamente después de enviar la petición, una ventana emergente (un cuadro de diálogo **alert()** de JavaScript) aparecerá en el navegador con el mensaje "XSS Explotado!". Esto demuestra que el código inyectado se ejecutó con éxito en el contexto del navegador del usuario.





·Solución:

Para mitigar la vulnerabilidad de XSS Reflejado, se recomienda implementar las siguientes medidas:

- **Codificación de Salidas (Output Encoding):** Codificar adecuadamente todos los datos proporcionados por el usuario antes de mostrarlos en la página HTML. Esto convierte caracteres especiales (como `<`, `>`, `'`, `"`) en sus entidades HTML correspondientes, impidiendo que el navegador los interprete como código.
- **Validación de Entradas (Input Validation):** Validar y sanear rigurosamente todas las entradas del usuario en el lado del servidor para asegurar que solo se aceptan datos esperados y seguros.
- **Content Security Policy (CSP):** Implementar una cabecera HTTP **Content-Security-Policy** restrictiva para controlar qué recursos puede cargar y ejecutar el navegador, limitando la capacidad de un atacante para inyectar y ejecutar scripts.

·Etiquetas de Alerta:

| Clave | Valor |
|----------------|---|
| CWE-79 | https://cwe.mitre.org/data/definitions/79.html |
| OWASP_2021_AO5 | https://owasp.org/Top10/A05_2021-Sec |

| | |
|-----------------------|---|
| | urity_Misconfiguration/ |
| OWASP_2017_AO6 | https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration |

5.1. Hallazgos Informativos / Observaciones de OWASP ZAP

Además de las vulnerabilidades directas y explotables, OWASP ZAP también identificó una serie de observaciones y hallazgos informativos durante el escaneo. Estos elementos, aunque no representan una vulnerabilidad directa o explotable por sí mismos, proporcionan contexto sobre la aplicación o revelan detalles que podrían ser útiles en un proceso de auditoría más amplio.

·Observación: Petición de Autenticación Identificada

- URL:** http://localhost/login.php
- Riesgos:** Informational
- Confianza:** High
- Parámetro:** Login
- Evidencia:** password
- Origen:** Pasivo (10111-Petición de Autenticación Identificada)
- Referencia de Alerta:** 10111-1
- Definición:**

La petición en cuestión se ha identificado como una petición de autenticación. El campo "Otra información" contiene un conjunto de líneas key=value que identifican cualquier campo relevante. Si la solicitud está en un contexto que tiene un método de autenticación para que coincida con la petición identificada..

·**Otra información:**

·*userParam=Login*

·*userValue=Login*

·*passwordParam=password*

·**Solución:**

Se trata de una alerta informática y no de una vulnerabilidad por lo que no hay nada que corregir.

Observación: Respuesta de Gestión de Sesión Identificada

·**URL:** http://localhost/

·**Riesgos:** Informational

·**Confianza:** Medium

·**Parámetro:** PHPSESSID

·**Evidencia:** ti1uirhce54pp1gieg2373b70

·**Origen:** Pasivo (10112- Respuesta de Gestión de Sesión Identificada)

·**Referencia de Alerta:** 10112-1

·**Descripción:**

Se ha identificado que la respuesta dada contiene un token de gestión de sesión. El campo "Otra información" contiene un conjunto de tokens de cabecera que pueden utilizarse en el método Header Based Session Management. Si la petición se encuentra en un contexto que tiene un método Session Management establecido en "Auto-detect", esta regla cambiará la gestión de sesión para utilizar los tokens identificados.

·**Otra información:**

cookie PHPSESSID

·**Solución:**

Se trata de una alerta informática y no de una vulnerabilidad por lo que no hay nada que corregir.

6. Conclusiones y Lecciones Aprendidas

6.1. Logros del Proyecto

Este proyecto ha culminado con éxito **en la configuración de un entorno de auditoría de seguridad web robusto y funcional**, basado en la ejecución de DVWA dentro de un contenedor Docker. Se logró no solo establecer la infraestructura necesaria, sino también **identificar y explotar exitosamente varias vulnerabilidades clave** presentes en DVWA. Este proceso permitió una comprensión práctica y profunda de diversas técnicas de ataque y sus implicaciones, validando la efectividad del entorno de pruebas.

6.2. Aprendizajes Clave

La realización de este proyecto ha proporcionado una serie de aprendizajes fundamentales en el campo de la seguridad web:

- **La Indispensable Utilidad de Herramientas Profesionales (ZAP y Burp Suite):** Se confirmó la vital importancia de herramientas como OWASP ZAP y Burp Suite en una auditoría de seguridad. Su capacidad para interceptar, inspeccionar y manipular el tráfico **HTTP/HTTPS**, así como para realizar escaneos y fuzzing, son críticas para descubrir y explotar vulnerabilidades de manera eficiente. Aprender a navegar por sus interfaces, configurar sus proxies y utilizar sus funciones básicas es un pilar para cualquier auditor de seguridad web.
- **Comprensión Práctica de la Explotación de Vulnerabilidades y su Impacto:** A través de la interacción directa con DVWA, se logró entender cómo diversas vulnerabilidades, como la inyección SQL, Cross-Site Scripting (XSS) y la inyección de comandos, pueden ser explotadas en escenarios reales. Más allá de la teoría, se visualizó el impacto potencial de estas explotaciones, desde el acceso a bases de datos hasta la ejecución de comandos arbitrarios en el servidor, subrayando la gravedad de no proteger adecuadamente una aplicación.
- **La Relevancia de los Niveles de Seguridad en una Aplicación:** DVWA demostró ser una herramienta didáctica excepcional gracias a su funcionalidad de niveles de seguridad. Se comprendió cómo las aplicaciones pueden presentar diferentes grados de vulnerabilidad (desde "low" hasta "impossible") dependiendo de las validaciones, sanitizaciones y medidas de seguridad implementadas. Practicar en distintos niveles permitió observar cómo los atacantes deben adaptar sus técnicas a medida que las defensas

se fortalecen, y cómo un desarrollo seguro reduce drásticamente la superficie de ataque.

6.3. Desafíos Encontrados y Superación

A lo largo del proyecto, surgieron algunos desafíos técnicos que, al ser superados, reforzaron el aprendizaje:

- **Problemas con los Permisos de Docker y Burp Suite:** Inicialmente, hubo inconvenientes relacionados con los permisos de usuario al ejecutar comandos Docker y al guardar configuraciones de Burp Suite. Esto se resolvió añadiendo el usuario al grupo `docker` y asegurando los permisos correctos en las carpetas de trabajo, lo que enfatizó la importancia de una configuración de entorno adecuada.
 - **Conflicto de Puertos (ZAP y Burp Suite):** La coincidencia del puerto `8080` por defecto para los proxys de ZAP y Burp Suite fue un desafío inicial. Este se superó mediante la **alternancia manual de la configuración del proxy en Firefox**, activando la herramienta necesaria en cada momento. Esta decisión priorizó la simplicidad y la claridad del flujo de trabajo para las pruebas específicas del proyecto, permitiendo una comprensión más directa de cada herramienta individualmente.
-

6.4. Próximos Pasos y Mejoras (Opcional)

Si se dispusiera de más tiempo para extender este proyecto, las siguientes áreas serían priorizadas para una profundización y mejora adicionales:

- **Exploración de Vulnerabilidades Más Complejas:** Profundizar en la explotación de vulnerabilidades menos triviales de DVWA, como CSRF más allá de los ejemplos básicos, y técnicas avanzadas de inyección.
- **Integración de Otras Herramientas de Kali Linux:** Incorporar y experimentar con otras herramientas presentes en Kali Linux que complementen a ZAP y Burp Suite, como Nmap para escaneo de red, Nikto para escaneo de vulnerabilidades de servidor web, o Metasploit para la explotación de servicios.
- **Auditoría en Niveles de Seguridad Superiores:** Realizar una auditoría exhaustiva en los niveles "medium" y "high" de DVWA para entender cómo las defensas adicionales impactan las técnicas de ataque y cómo se requiere un enfoque más sofisticado para la explotación.

- **Automatización de Pruebas:** Explorar la posibilidad de automatizar ciertas pruebas de seguridad utilizando los scripts o APIs que ZAP y Burp Suite ofrecen, lo que sería útil para pruebas de regresión.

Este proyecto ha sentado una base sólida en el entendimiento de la seguridad de aplicaciones web y la operación de herramientas clave, preparando el terreno para un estudio y práctica más avanzados en el futuro.