Nombre Taller:lab 8

File Inclusion 0x03 [Challenge]

Detalles Prácticos:

Descripción: Explatación de una vulnerabilidad de tipo **local file inclusion (LFI)** para acceder archivos sensibles del sistema, como /etc/passwd, mediante modificación de párametros ocultos en solicitudes internas.
Nivel(Básico/Intermedio o avanzado)
Instructores:Rieradipe

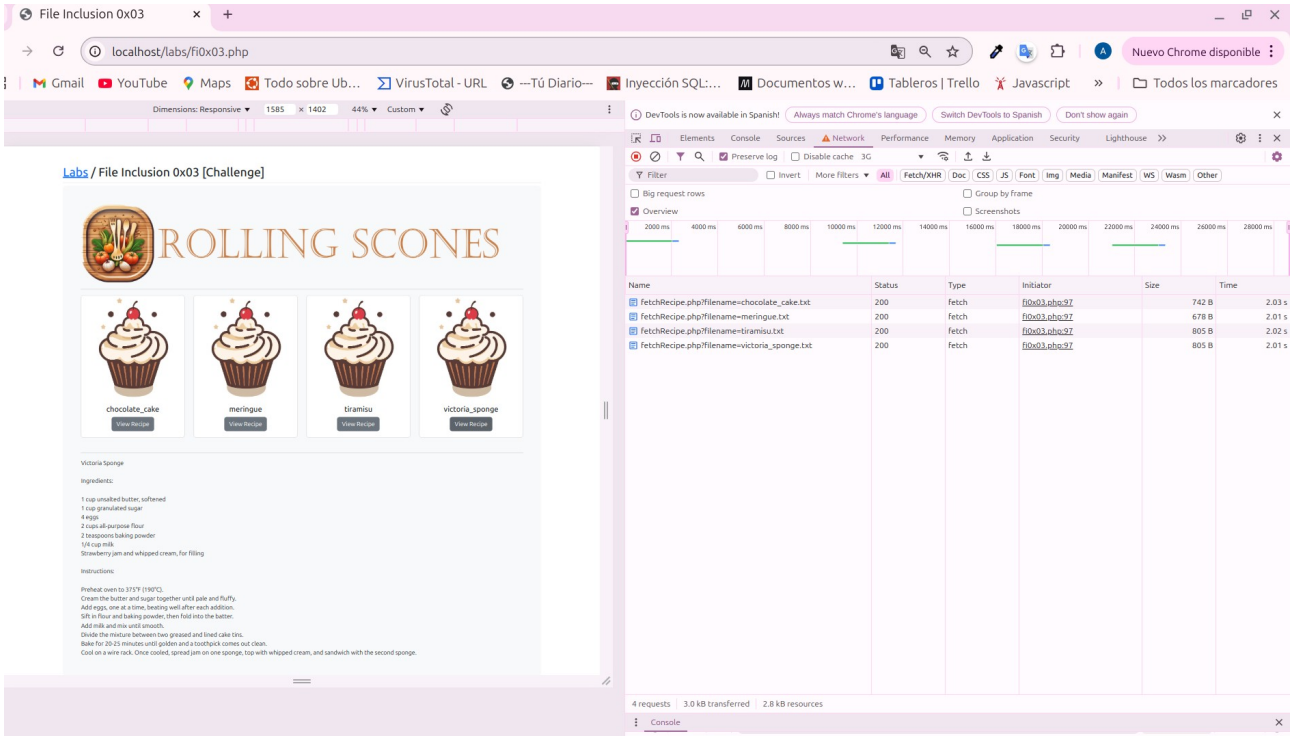
Tabla de contenidos:

🔍 Análisis inicial:

- La URL principal del lab es:

```
http://localhost/labs/fioX03.php
```

- A simple vista la Url no muestra ningún parámetro modificable.



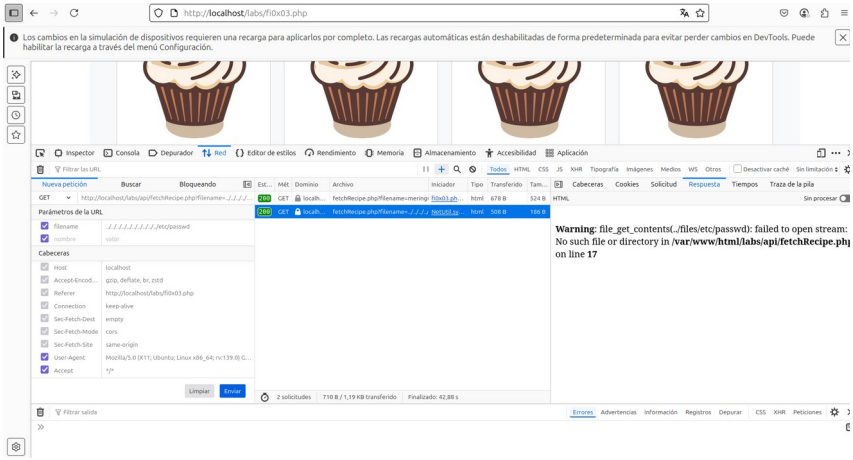
- Sin embargo, usando **DevTools,** pestaña de **Red**(network), se observa que al hacer click en las recetas se lanza una solicitud fetch:

fetchRecipe.php?filename=nombre_receta.txt

<u>Estrategia de Explotación:</u>

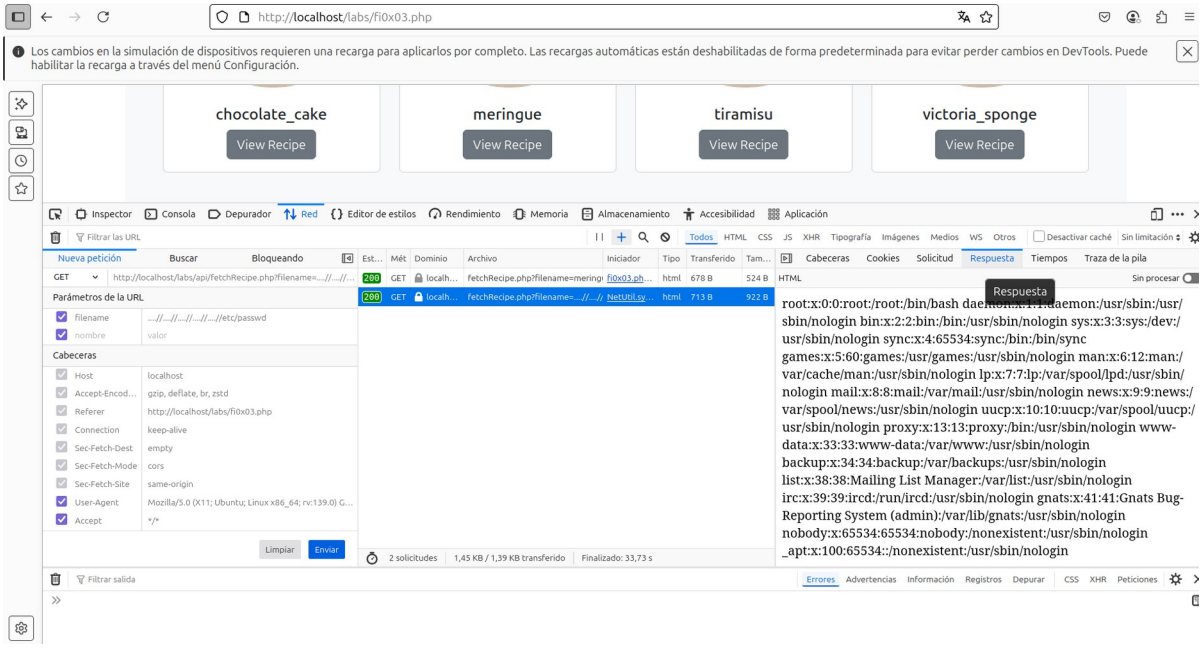
1. <u>Identifica la petición vulnerable</u>
   - Dentro de DevTools, seleccionamos la peticion fetchRecipe.php
   - Hacemos click derecho y vamos a Editar y Reenviar
   - Modificamos el parámetro filename= con una cadena de path transversal

../../../../../../etc/passwd



2. Ejecución del ataque:

- Tras enviar la petición modificada, el servidor responde con el contenido del archivo /etc/passwd, confirmando que la vulnerabilidad LFI es explotable.



Conclusion:

- La aplicación es vulnerable a LFI porque permite modificar de forma directa el nombre del archivo incluido sin una validación correcta.
- Usar herramientas como DevTools en combinacion con técnicas de path transeversal permite descubrir y explotar este tipo de fallos, incluso cuando no hay parámetros visibles en la URL

Recomendaciones de Seguridad:

Validacion estricta de entrada del usuario.	Solo deben permitirse nombres de archivos **predefinifos y controlados por el servidor**(ej: a traves de lista blanca, Ids internos)
	Núnca confiar en parámetros que vengas del cliente, incluso si están ocultos o gestionados por **fetch**

Uso de rutas absolutas **internas y controladas**	```php $archivosPermitidos = ["chocolate" => "/var/www/html/recetas/chocolate_cake.txt", "tiramisu" => "/var/www/html/recetas/tiramisu.txt"]; ``` El código no debe construir rutas dinámicas con strings que vengan de fuera. En su lugar, podemos usar rutas mapeadas dentro del código

Desactivar funciones peligrosas como *file_get_contents()* **sin validación**	Usar funciones más seguras que verifiquen que el archivo existe, pertenece a una carpeta permitida, y no contiene secuencias como ../.

Configurar adecuadamente el servidor web	Restringir el acceso a directorios sensibles (/etc, /var, /home) mediante reglas Ej: <Directory /etc> Deny from all </Directory>

Aplicar **control** de errores customizado	No mostrar trazas o errores detallados al usuario(warnings cuando falla la inclusion), ya que revelan rutas internas y lógica del sistema.

Monitoreo y logging	Registrar accesos inusualoes o rutas sospechosas don patrone ../ o llamadas a archivos fuera del contexto esperado