Puesta en Producción Segura

Práctica 2.8

LocalFileInclusion

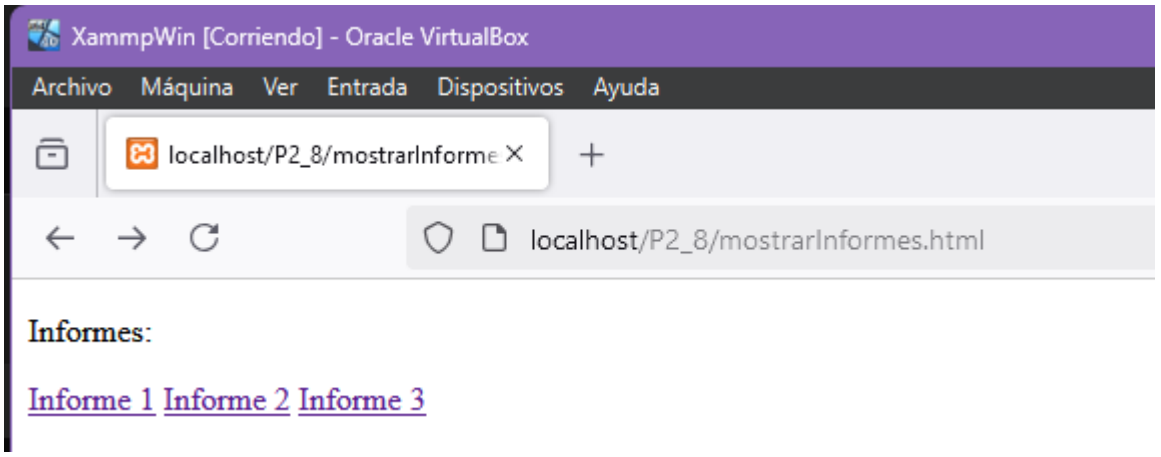
Jennifer Galván Bejarano

<Incluye en un documento Word capturas de pantalla de todo el proceso>

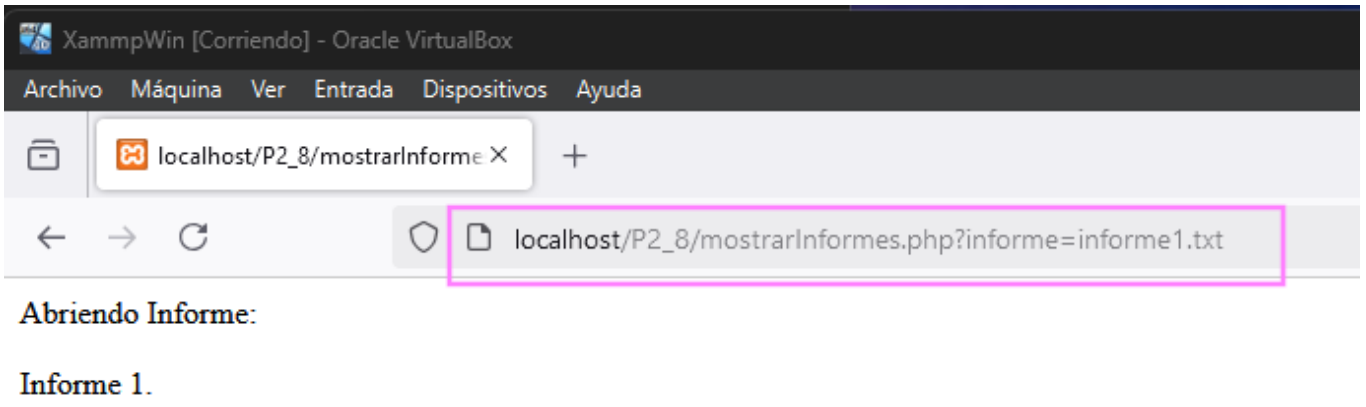
Preparación:
1) Despliega los ficheros mostrarinformes.html y mostrarinformes.php en el servidor.

Actividades:

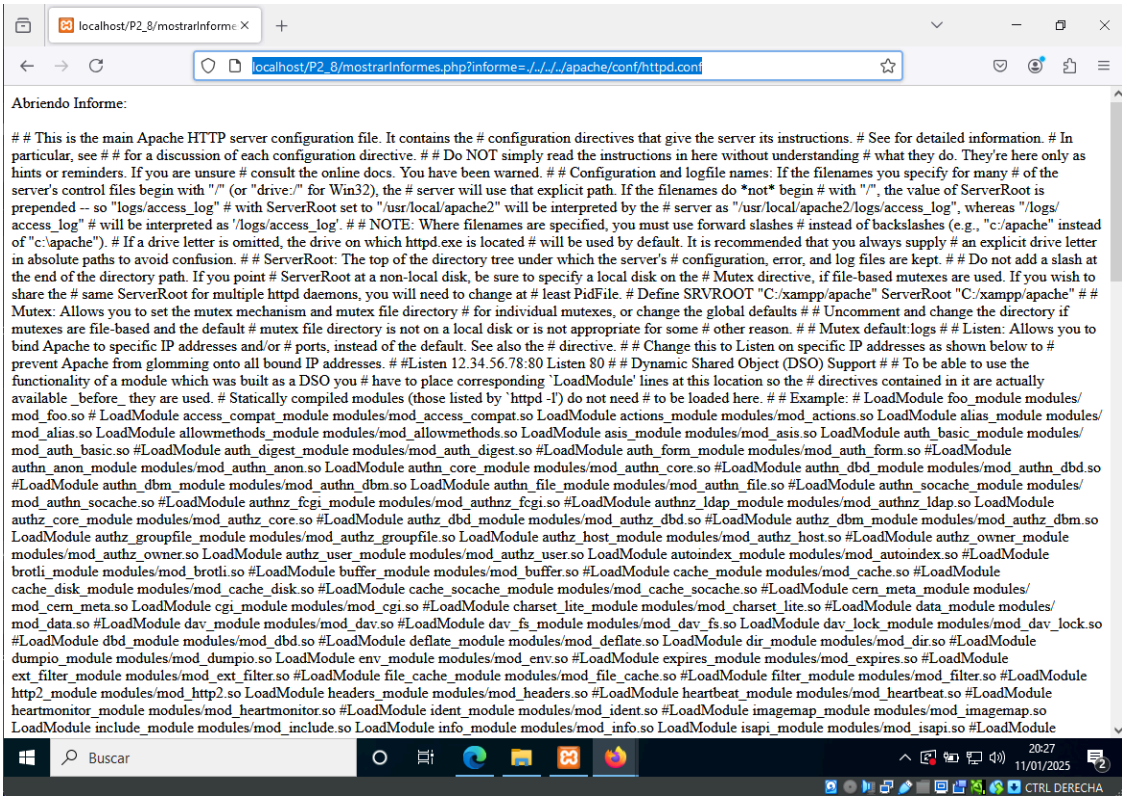
1) Realiza un ataque LocalFileInclusion que muestre el contenido del archivo de configuración del apache httpd.conf (contiene la configuración del servidor).
Para comprender este ataque primero entre en mi XAMMP primero entre a ver el html



Cuando le daba a uno de los informes en la ruta cambiaba por:
http://localhost/P2_8/mostrarInformes.php?informe=infomre1.txt



Teniendo esa información vi que es vulnerable y podría buscar el archivo de configuración del apache httpd.conf cambiando en lugar de informes1.txt podria cambiar de niveles del directorio hasta llegar a apache. de esta manera http://localhost/P2_8/mostrarInformes.php?informe=./../../../apache/conf/httpd.conf



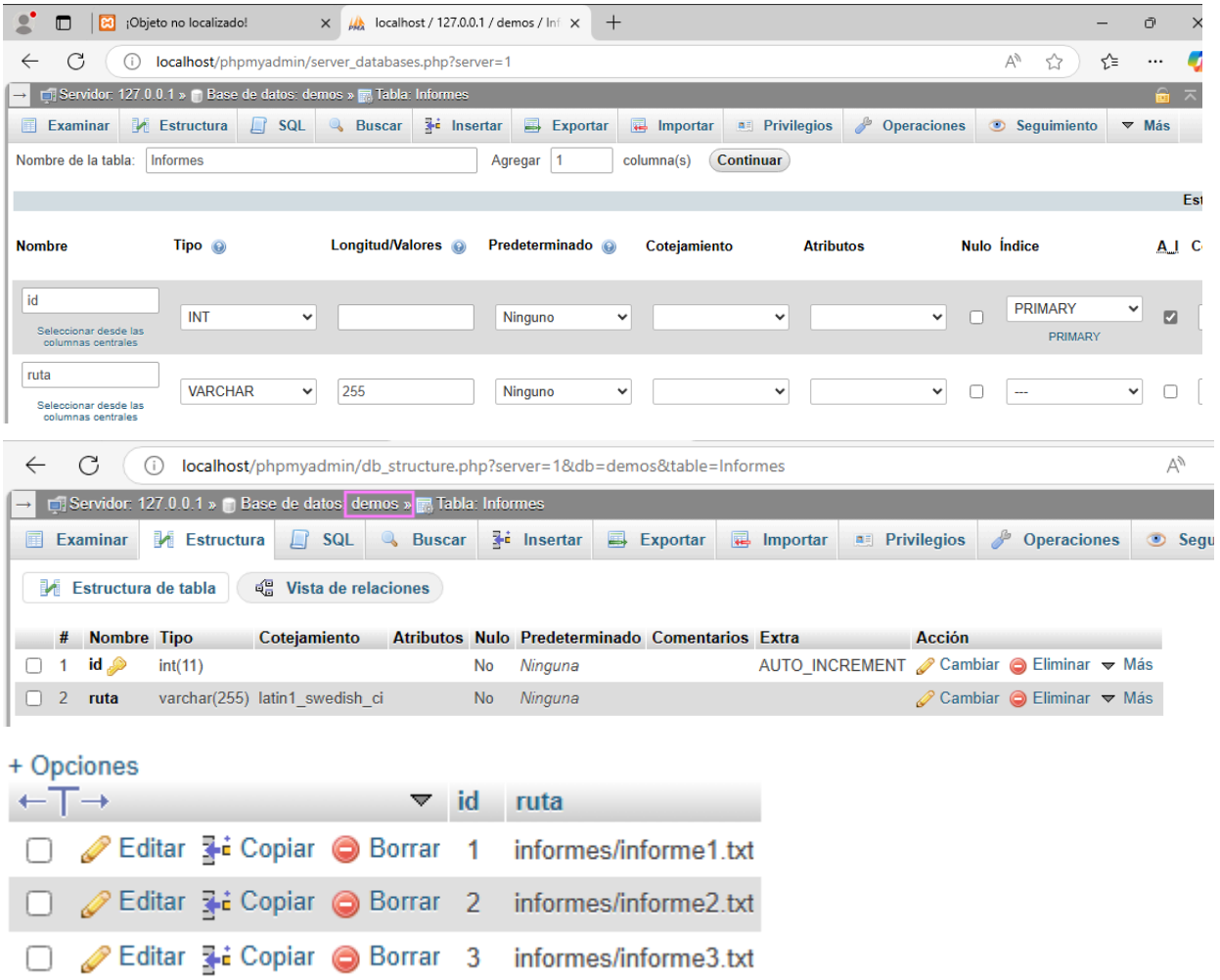
2) Mejora la visualización de archivos de esta práctica para evitar el ataque LocalFileInclusion.

Para ello:

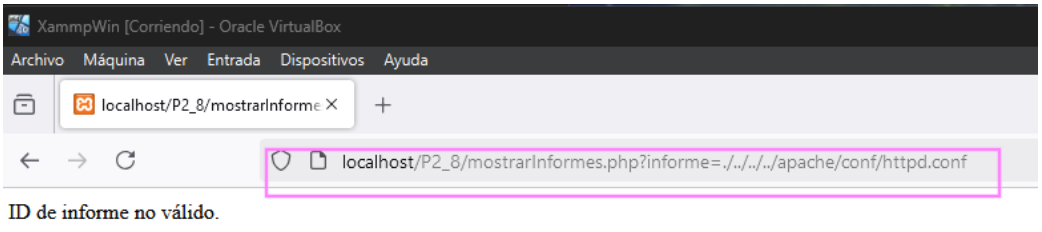
- En la base de datos “demos”, crea una tabla llamada Informes que contenga dos campos:

  • id: de tipo entero, identifica al fichero. Ejemplo: id=1 → fichero1.txt

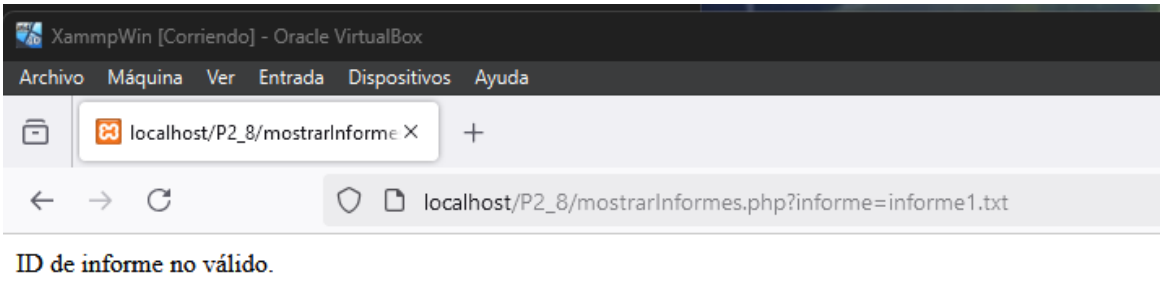
  • ruta: de tipo varchar(255), contiene la cadena de caracteres con la ruta relativa del fichero. Ejemplo: ficheros/fichero1.txt



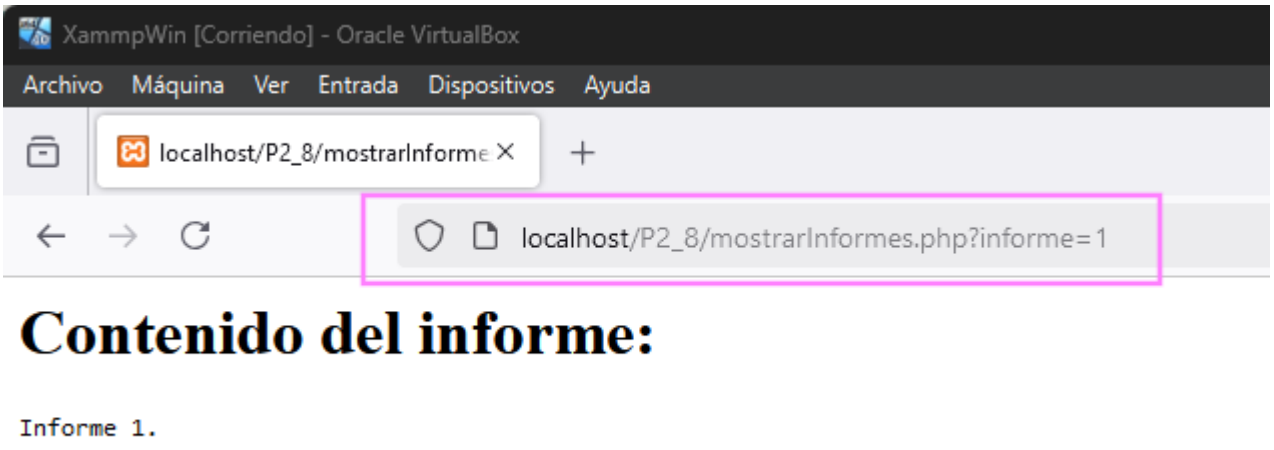
- Modifica mostrarinformes.php para acceder a los ficheros de la forma siguiente, donde se pase en la URL el id del fichero y se muestre por pantalla.
Ejemplo:
http://localhost/ciber/demos/mostrarInformes2.php?informe=3
Después de modificar el php intento a entrar al apache como lo indica en el apartado 1 mostrandome este resultado, esto es porque la consulta no encuentra ningún ID relacionado



Si intento entrar como anteriormente tampoco sale, esto es porque la consulta no encuentra ningún ID relacionado



Si entro al html y seleccione el informe, podemos ver que me especifica inofmres=1 que es llamar la id del informe que lo busca primero en la base de datos.



3) ¿Crees que podríamos evitar esta vulnerabilidad si pasáramos los parámetros por POST en lugar de GET?

La vulnerabilidad de Local File Inclusion (LFI) no depende del método de envío de los parámetros, sino de cómo el servidor utiliza esos parámetros para incluir archivos.

Con el método Post no se elimina completamente la vulnerabilidad, aunque sí puede reducir el riesgo de explotación casual. ya que los parámetros no son visibles en la URL, lo que puede dificultar su explotación directa por usuarios no técnicos o casuales.

En **GET**, los parámetros son visibles en la barra de direcciones del navegador, lo que facilita la manipulación directa por parte de un atacante (por ejemplo, cambiando ?informe=../../etc/passwd).

Así que para mayor seguridad estaría bien cambiar el método de GET a POST