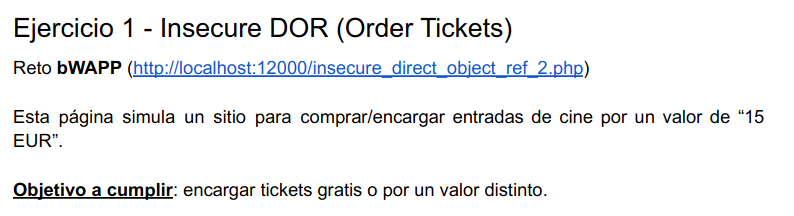
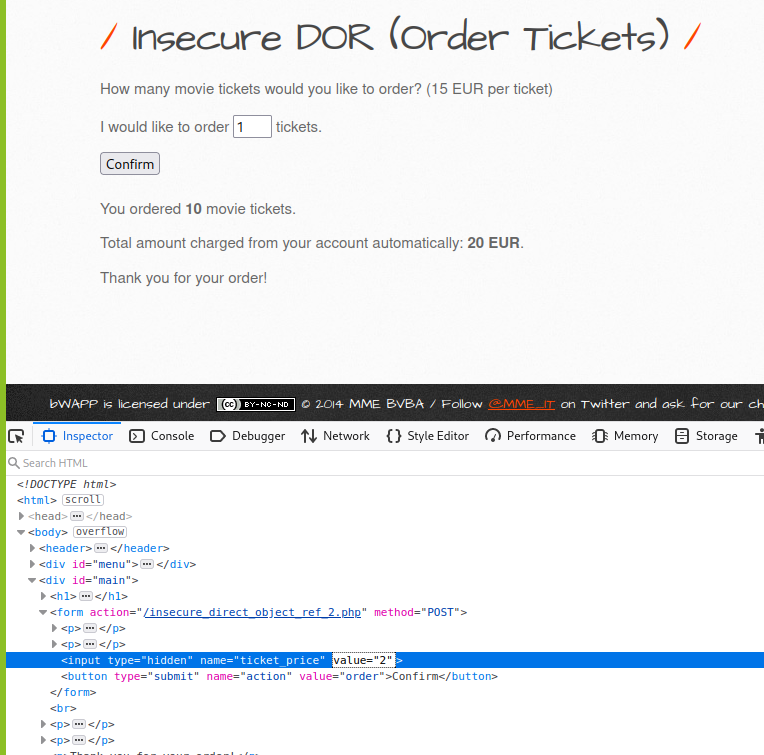
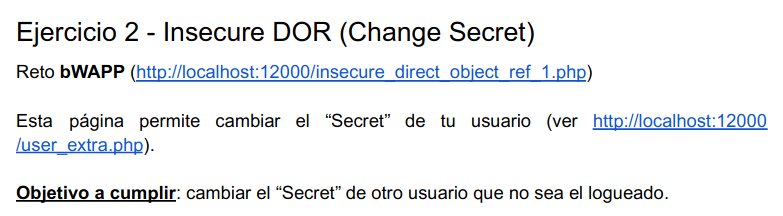
**DSA - Práctica 2**

Alejo Alfredo Santi y Fermín Moreno

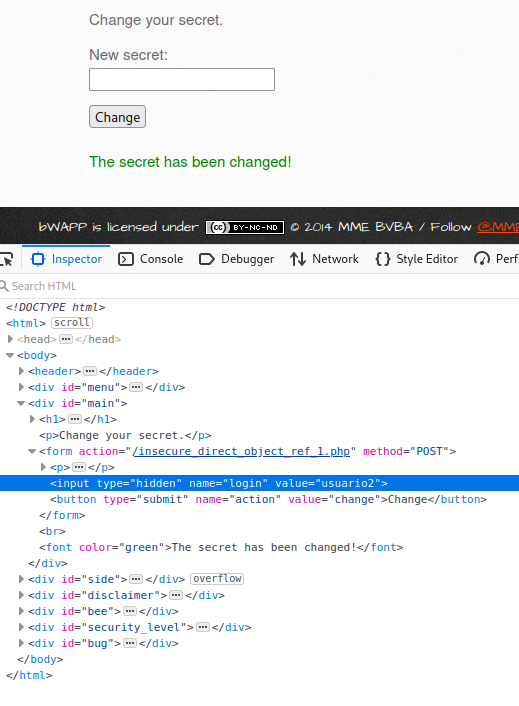
El precio de la entrada se guarda del lado del cliente como un input HTML, por lo que se puede cambiar editando el valor de la propiedad de la etiqueta o haciendo visible la etiqueta y cambiando el campo.  
  
En este caso cambiamos la propiedad “value” directamente a 2 EUR y compramos 10 tickets, por lo que salió 20 EUR la compra total.



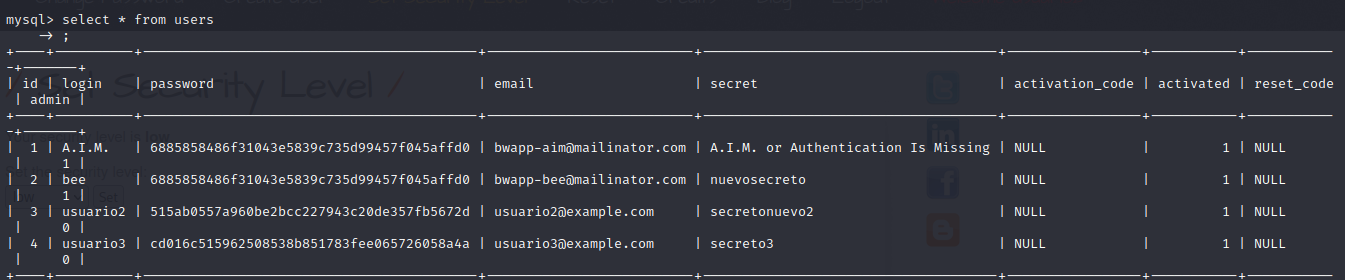
Para solucionar este inconveniente el precio de la entrada no debería guardarse del lado del cliente en el HTML, se lo debería guardar del lado del servidor, en una base de datos o, en su defecto, hardcodearlo.



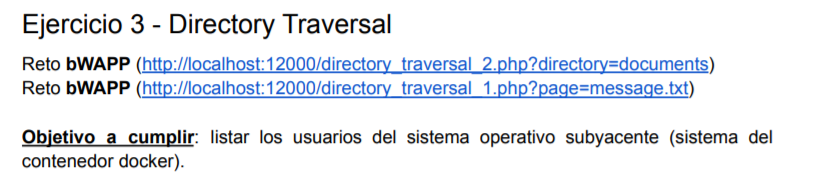
La vulnerabilidad que tiene la página es que guarda el usuario al que se le va a cambiar el secret en un input hidden, en la screen de abajo se ve cómo podemos cambiar el valor de la propiedad “value” y así cambiar al usuario al que se le modifica el secret, en este caso modificamos, desde el usuario “bee”, el secreto del “usuario2”.

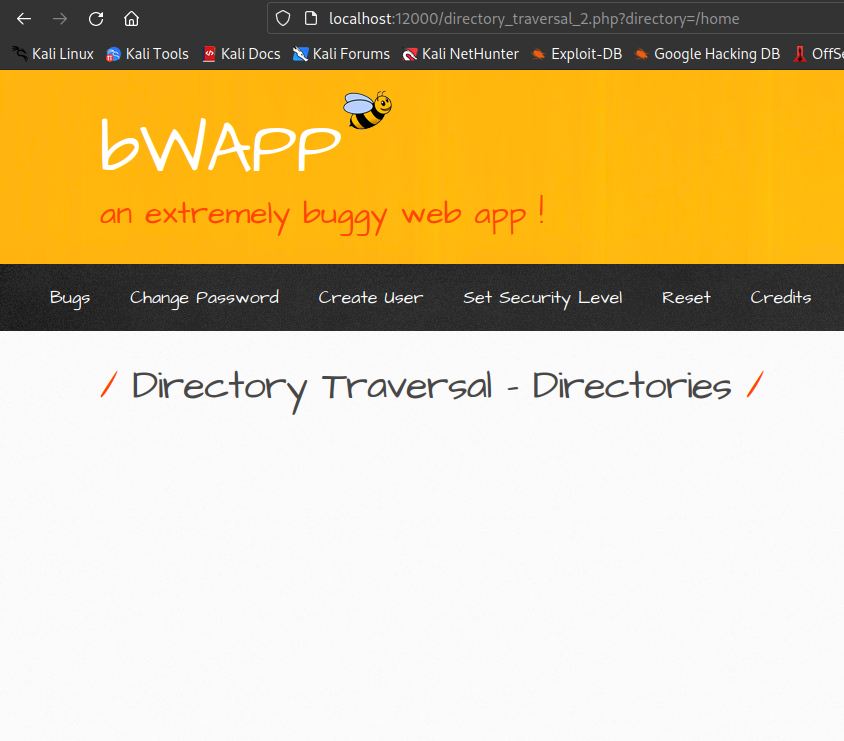


Acá en la base se ve como el secreto del “usuario2” ahora es “secretonuevo2”, cuando el que se configuró originalmente era “secreto2”:



Para solucionar esto se debería guardar, fuera del alcance del cliente, el usuario que está enviando la petición, por ejemplo con una cookie de sesión encriptada que le provea el servidor a cada usuario cuando se loguea.





En esta página se lista un directorio, el cual se pasa por parámetro en un pedido get. Si se edita el pedido get escribiendo otro directorio se puede acceder al mismo y listar los archivos/directorios que contenga, ya que evidentemente la manera de obtener los datos que utiliza la web es mediante la ejecución del comando ls en la bash tomando el directorio que se le pasa como parámetro. Para obtener los usuarios del sistema vamos a listar el contenido del directorio /home. En este caso está vacío, por lo que no hay otro usuario que no sea el root.



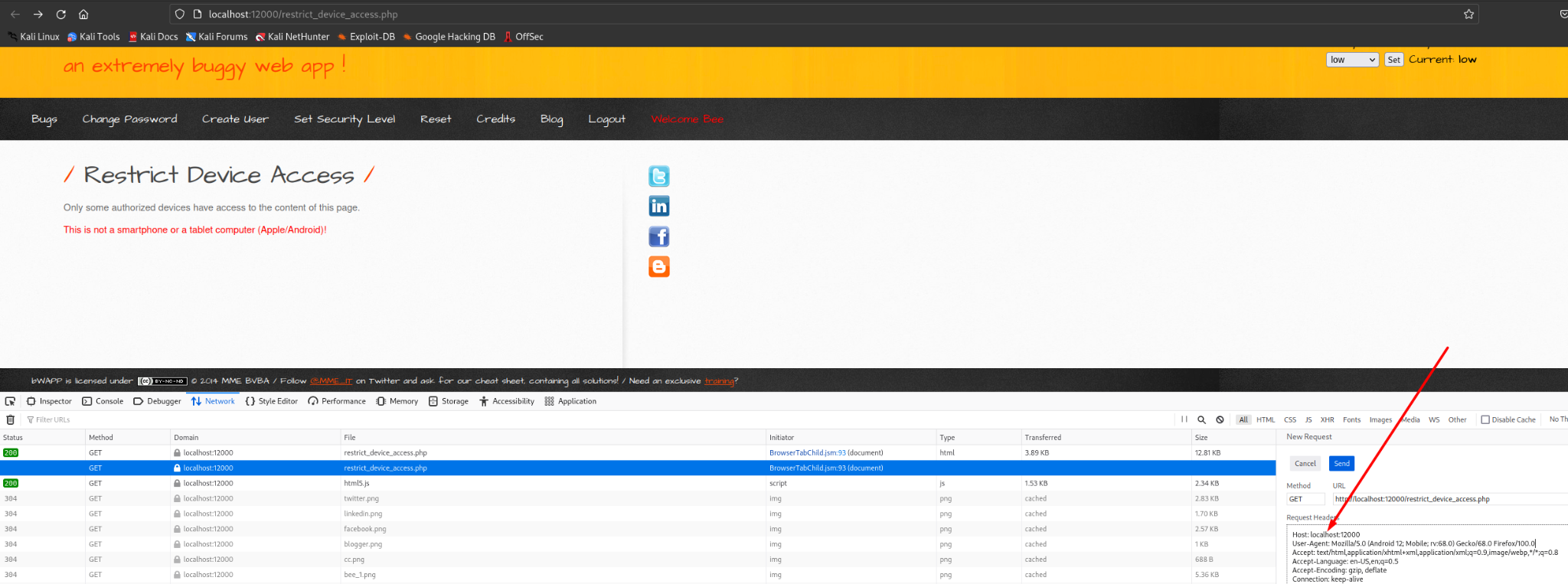
En este caso se muestra un mensaje, el cual se obtiene de un archivo que se le pasa como parámetro. Al cambiar el parámetro podemos obtener el contenido de cualquier otro archivo que deseemos, ya que la implementación que parece tener es utilizar el comando cat en una bash con el directorio pasado como parámetro. Para obtener los usuarios del sistema pasamos como parámetro /etc/passwd, y vemos como el único usuario que tiene el sistema es el root.

Para evitar esta situación, se debe guardar en una base de datos la información que se le quiere mostrar al usuario, haciendo una tabla correspondiente. En el primer caso el listado de libros, y en el segundo caso el mensaje.

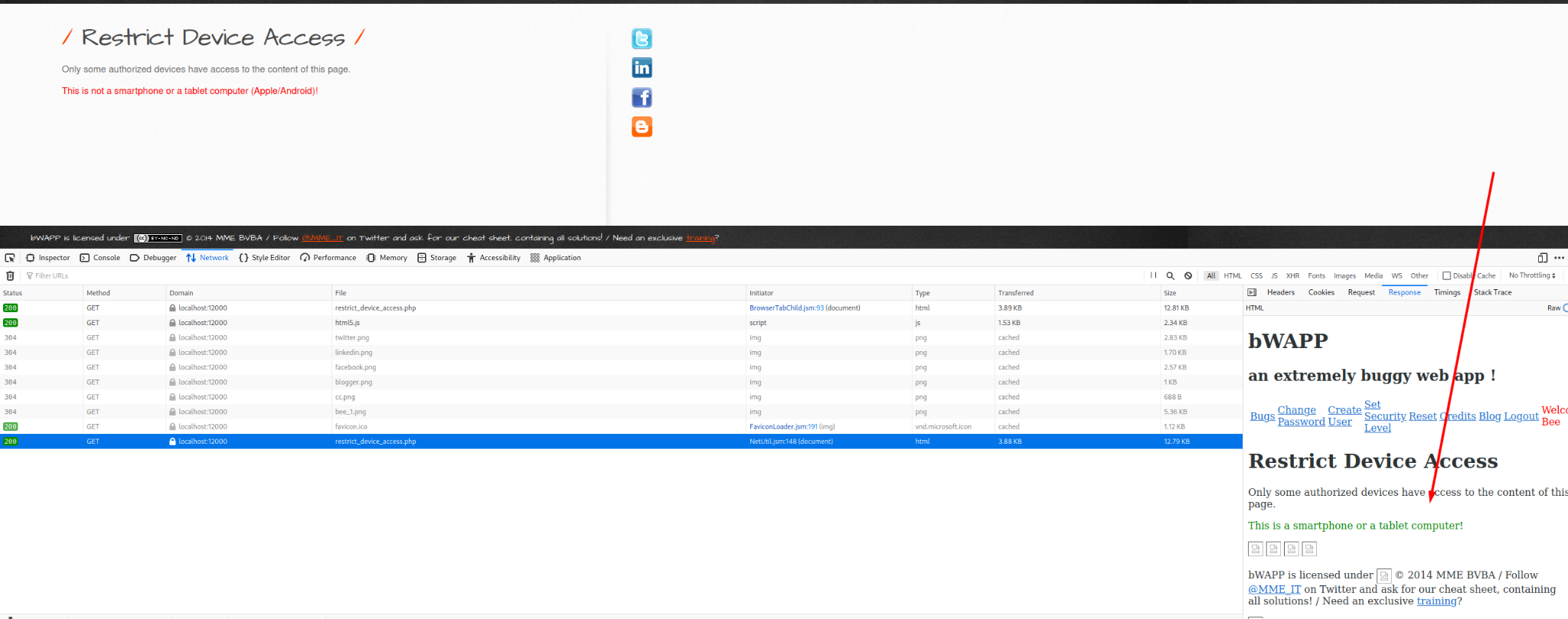


En este caso hay una vulnerabilidad porque la página, para darse cuenta si se entra con un dispositivo móvil, lo que hace es leer el campo user-agente del request, el cual se puede modificar.

Acá estamos editando el request y le estamos poniendo un user-agent correspondiente a un dispositivo móvil:

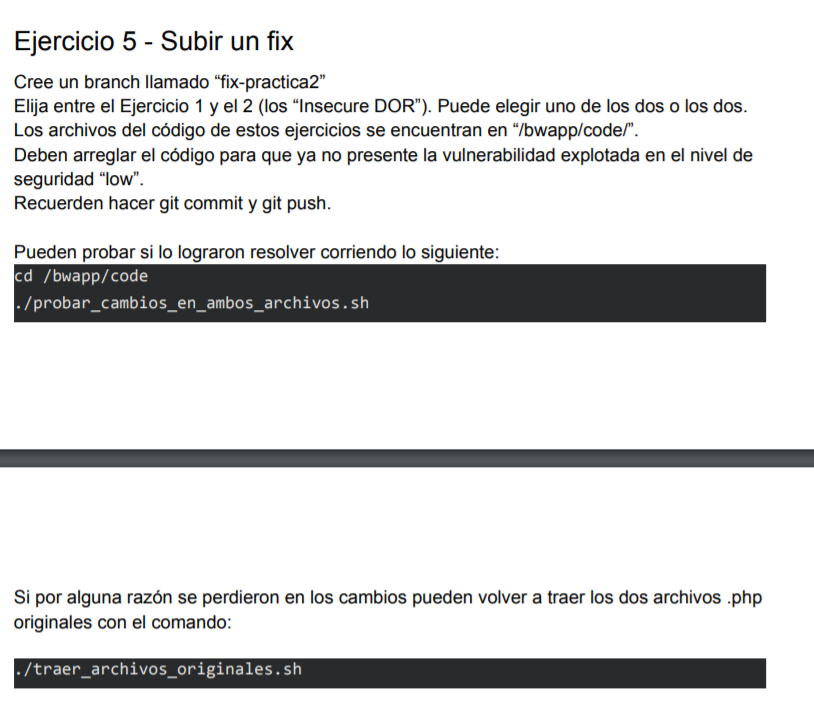
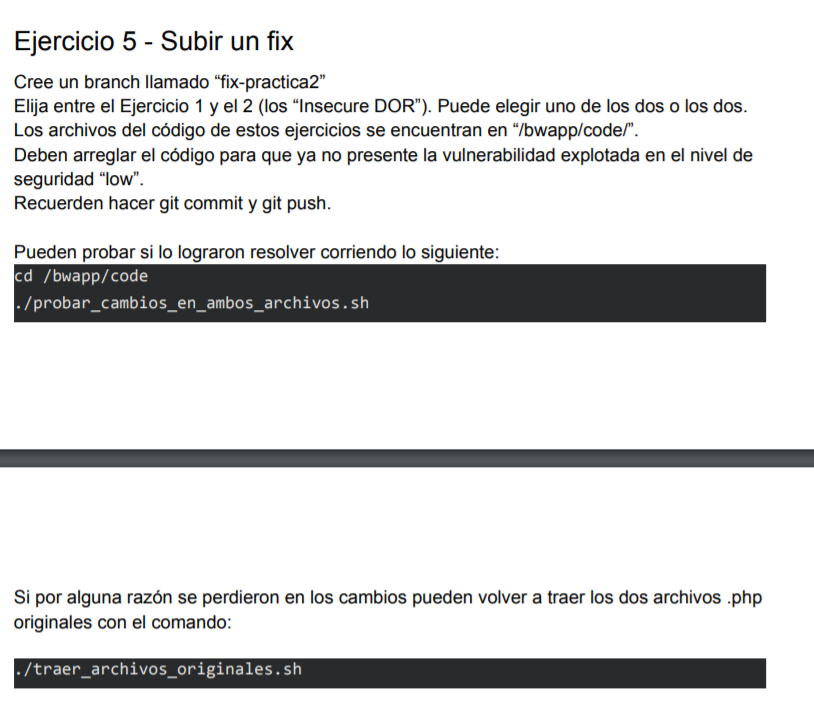


Se puede observar como en el response de la nueva consulta la página interpreta que estamos accediendo desde un dispositivo móvil.



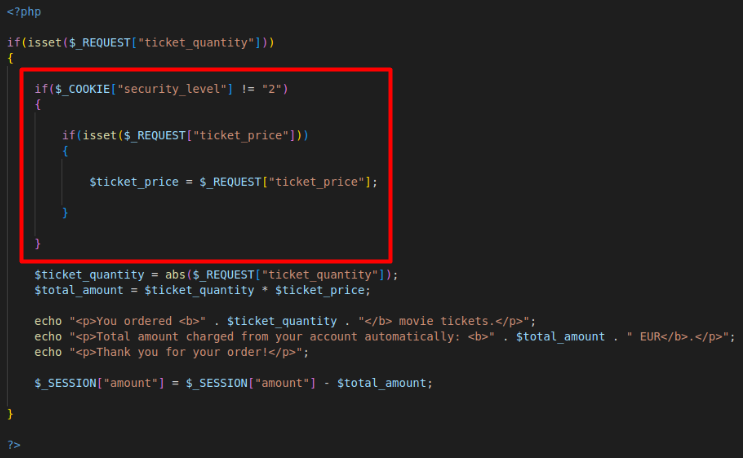
El error acá es hacer un control de acceso que dependa de un dato manipulable por el cliente, en este caso el user-agent.

Si se desea que los datos sólo sean accesibles desde un dispositivo móvil la solución óptima sería desarrollar una aplicación para celulares, la cual de todas maneras puede ser burlada con emuladores en una pc, por lo que hacer un control de acceso en base al dispositivo es un error de concepto, debería hacerse, por ejemplo, una solución tradicional de usuarios y roles para tener un sistema consistente.

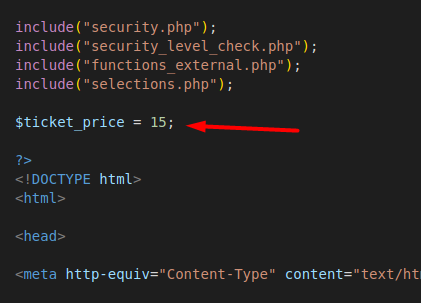


Vamos a elegir modificar el Ejercicio 1.

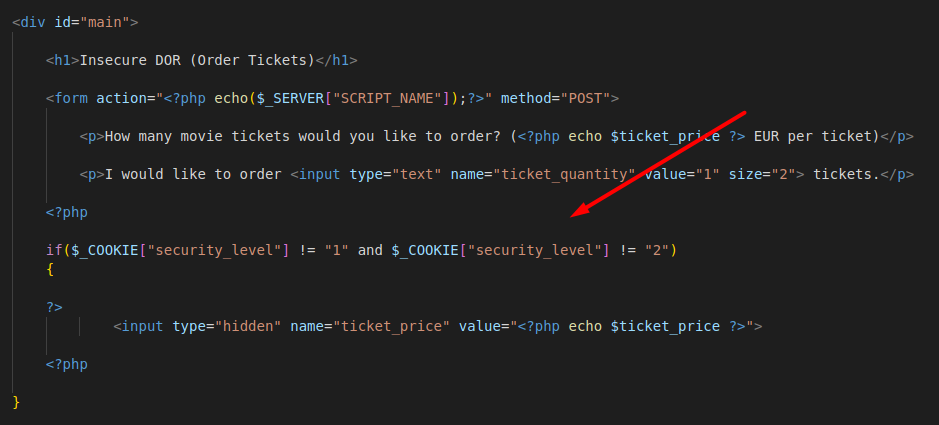
En este caso, el problema del ejercicio era que ponía a disposición del cliente el valor del ticket con un input escondido. Para solucionar el problema eliminamos el siguiente if:



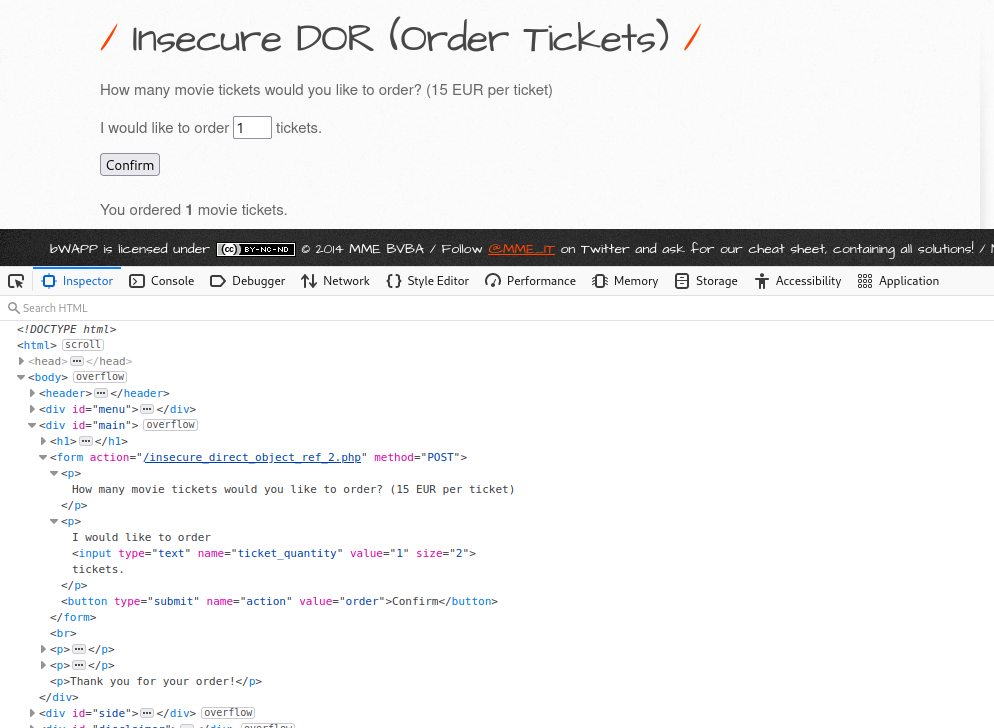
Cuando se realizaba el cálculo del valor de la compra, este $\_REQUEST[“ticket\_price”] buscaba el valor del formulario para setear la variable ticket\_price, la cual se usa para conocer el precio del ticket. Eliminando este if, el valor del ticket que se utilizará para calcular el valor de la compra es el que está seteado en el código, no el que se ingrese en el input del formulario:



El input que tomaba el valor del ticket quedó inutilizado, por lo que también lo eliminamos:



Al final de los cambios, desde el lado del cliente se ve esto:



No hay input que se pueda modificar, e incluso si lo hubiese el código tampoco lo tendría en cuenta.