Ejercicios Seguridad

Instalación WebGoat

docker pull webgoat/goatandwolf

docker run -d -p 80:8888 -p 8080:8080 -p 9090:9090 -e TZ=Europe/Amsterdam webgoat/goatandwolf

http://localhost/

WebGoat URL http://127.0.0.1:8080/WebGoat
WebWolf URL http://127.0.0.1:9090/WebWolf

Injección

A1) Injection

SQL Injection (intro)

Ejercicio 9
Ejercicio 10
Ejercicio 11

SQL Injection (mitigation)

Ejercicio 11

Cross Site Scripting

```
<script>alert(document.cookie)</script>
```

Cross Site Scripting, Lesson 7 Exercise

Después de usar las entradas de datos predeterminadas, vemos que el número de la tarjeta de crédito completada por el usuario se refleja en el mensaje de agradecimiento de la página web. Podemos reemplazar el número de la tarjeta de crédito con el siguiente JavaScript:

<script>alert('This is an example of reflected XSS')</script>

Cross Site Scripting, Lesson 10 Exercise

Clicking the Submit button without input (or with an incorrect solution) prompts the suggestion to look at the GoatRouter.js file. Within this file, we see a few uses of the word "test" and can discern that the test path is: start.mvc#test/

JSON Web Token

JSON Web Token (JWT) es un estándar abierto (RFC 7519) que define un compacto y forma autónoma para transmitir de forma segura información entre las partes como un objeto JSON. Esta información puede ser verificado y confiable porque está firmado digitalmente. Los JWT se pueden firmar usando un secreto (con el algoritmo HMAC) o un par de claves pública / privada mediante RSA.

JSON Web Token

JSON Web Token se utiliza para transportar información relacionada con la identidad y características (reclamos) de un cliente. Este "contenedor" está firmado por el servidor para evitar que un cliente lo manipule para cambiar, por ejemplo, la identidad o cualquier característica (ejemplo: cambiar el rol de simple usuario a administrador o cambiar el inicio de sesión del cliente). Este token se crea durante autenticación (se proporciona en caso de autenticación exitosa) y es verificado por el servidor antes de cualquier procesamiento. Es utilizado por una aplicación para permitir que un cliente presente un token que represente su "tarjeta de identidad" (contenedor con toda la información del usuario sobre él) al servidor y permitir que el servidor verifique la validez e integridad del token de forma segura, todo esto de forma apátrida y enfoque portátil (portátil de la manera en que las tecnologías de cliente y servidor pueden ser diferente, incluido también el canal de transporte, incluso si HTTP es el más utilizado)