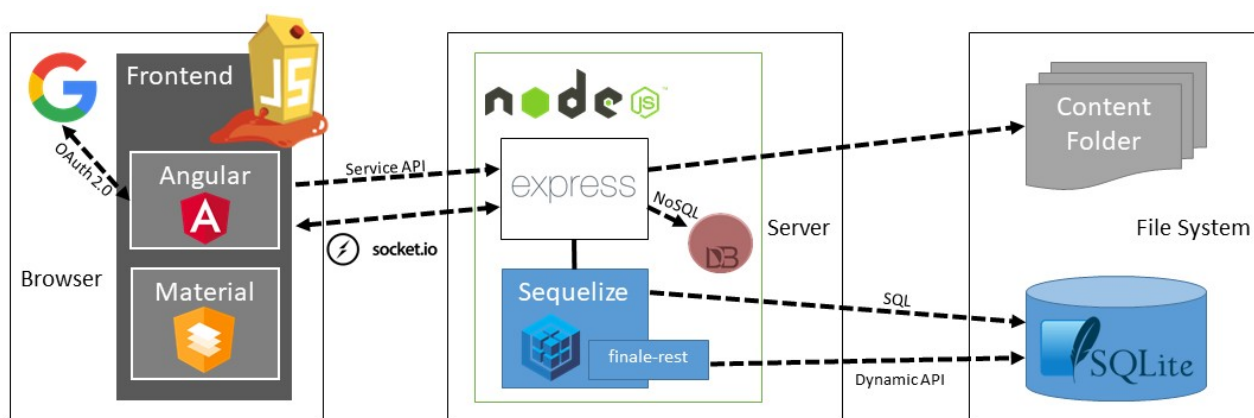


JUICE SHOP



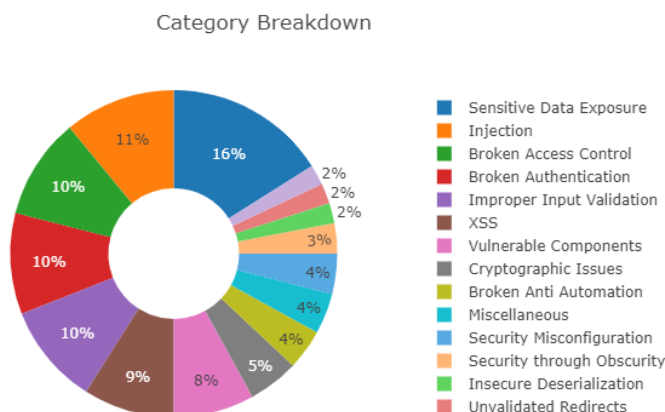
José Luis Berenguel Gómez – IES Zaidín-Vergeles

Sumario

1. Introducción.....	3
Descarga e instalación.....	3
Requisitos previos.....	5
Reconociendo la web.....	7
3. Caso práctico.....	8
Acceso a un documento confidencial.....	8
Descarga del fichero de copia de seguridad.....	9
Acceso como usuario administrador.....	10
Acceso a la página de administración.....	13
Acceso al carrito de la compra de otro usuario.....	14
Accediendo al tablero de puntuación.....	15
3. Ejercicios propuestos.....	16
4. Bibliografía.....	17

1. Introducción

OWASP Juice Shop es una de las aplicaciones web inseguras más modernas que se han desarrollado. Está escrita en Node.js, Express y Angular, contiene más de 100 retos de hacking web de diferentes categorías: Broken authentication, Broken Access Control, Injection, XSS, XEE, problemas criptográficos, deserialización insegura, etc.



Recurso

Web oficial de OWASP Juice Shop
<https://owasp.org/www-project-juice-shop/>

Además de Juice Shop, existen multitud de aplicaciones web vulnerables o VWA (*Vulnerable Web Applications*), entre las más conocidas están WebGoat, Multilidae, DVWA (Damn Vulnerable Web Application). Se puede consultar un listado completo en el directorio de aplicaciones web vulnerables del OWASP.

Recurso

Directorio del OWASP de VWA
<https://owasp.org/www-project-vulnerable-web-applications-directory/>

Descarga e instalación.

Juice Shop es un proyecto open source y el código fuente está disponible en GitHub. Se puede descargar e instalar directamente de este código fuente, desde una imagen Docker o a través de paquetes preparados para las diferentes plataformas (Windows, Linux, MAC).

La forma más sencilla de instalar Juice Shop y ponerla en marcha en nuestro equipo es a través del **contenedor docker**, si bien **algunos de los retos no estarán disponibles** en este entorno por motivos

de seguridad o por incompatibilidad técnica:

```
$ sudo docker pull bkimminich/juice-shop
Using default tag: latest
latest: Pulling from bkimminich/juice-shop
9b794450f7b6: Pull complete
9acbbc96f494: Pull complete
387d9933a547: Pull complete
2871dad46958: Pull complete
51e8d62119da: Pull complete
5e73d8e5d1a9: Pull complete
5db3f061e521: Pull complete
2628e6c54367: Pull complete
Digest: sha256:684ed5358233c77b7f4d5a9930d77434ca5cbbf7b4f84f05f935f97f59dba8d8
Status: Downloaded newer image for bkimminich/juice-shop:latest
docker.io/bkimminich/juice-shop:latest
```

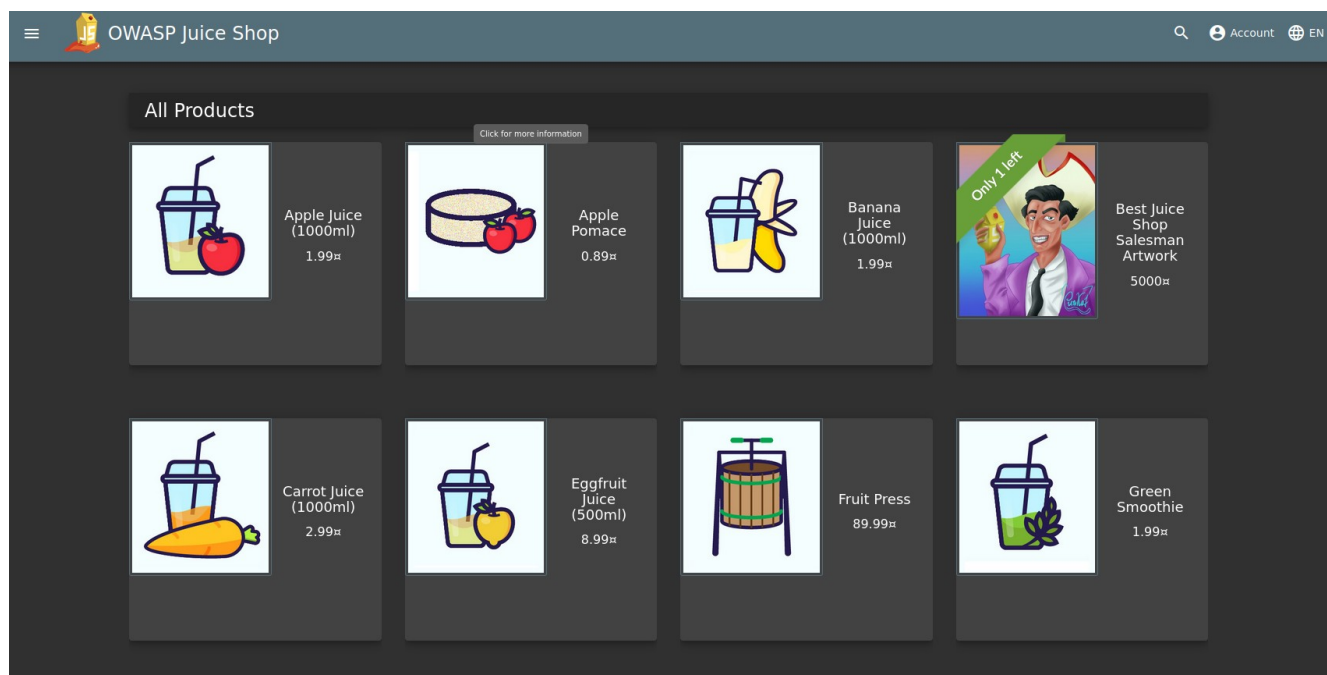
Y para arrancarlo:

```
$ sudo docker run --rm -p 3000:3000 bkimminich/juice-shop

> juice-shop@12.7.1 start /juice-shop
> node build/app

info: All dependencies in ./package.json are satisfied (OK)
info: Chatbot training data botDefaultTrainingData.json validated (OK)
info: Detected Node.js version v12.22.1 (OK)
info: Detected OS linux (OK)
info: Detected CPU x64 (OK)
info: Configuration default validated (OK)
info: Required file server.js is present (OK)
info: Required file index.html is present (OK)
info: Required file main-es2018.js is present (OK)
info: Required file tutorial-es2018.js is present (OK)
info: Required file styles.css is present (OK)
info: Required file polyfills-es2018.js is present (OK)
info: Required file runtime-es2018.js is present (OK)
info: Required file vendor-es2018.js is present (OK)
info: Required file main-es5.js is present (OK)
info: Required file tutorial-es5.js is present (OK)
info: Required file polyfills-es5.js is present (OK)
info: Required file runtime-es5.js is present (OK)
info: Required file vendor-es5.js is present (OK)
info: Port 3000 is available (OK)
info: Server listening on port 3000
```

Ya podemos abrir el navegador en la dirección <http://localhost:3000> para visitar la web de Juice Shop.



Requisitos previos.

Antes de comenzar el caso práctico es recomendable tener unos conocimientos mínimos del funcionamiento del protocolo HTTP. Para ello, puedes realizar el room WebFundamentals de TryHackMe y revisar los contenidos de la unidad, así como otros manuales sobre programación web para conocer el funcionamiento de las tecnologías de programación web de lado cliente y servidor.

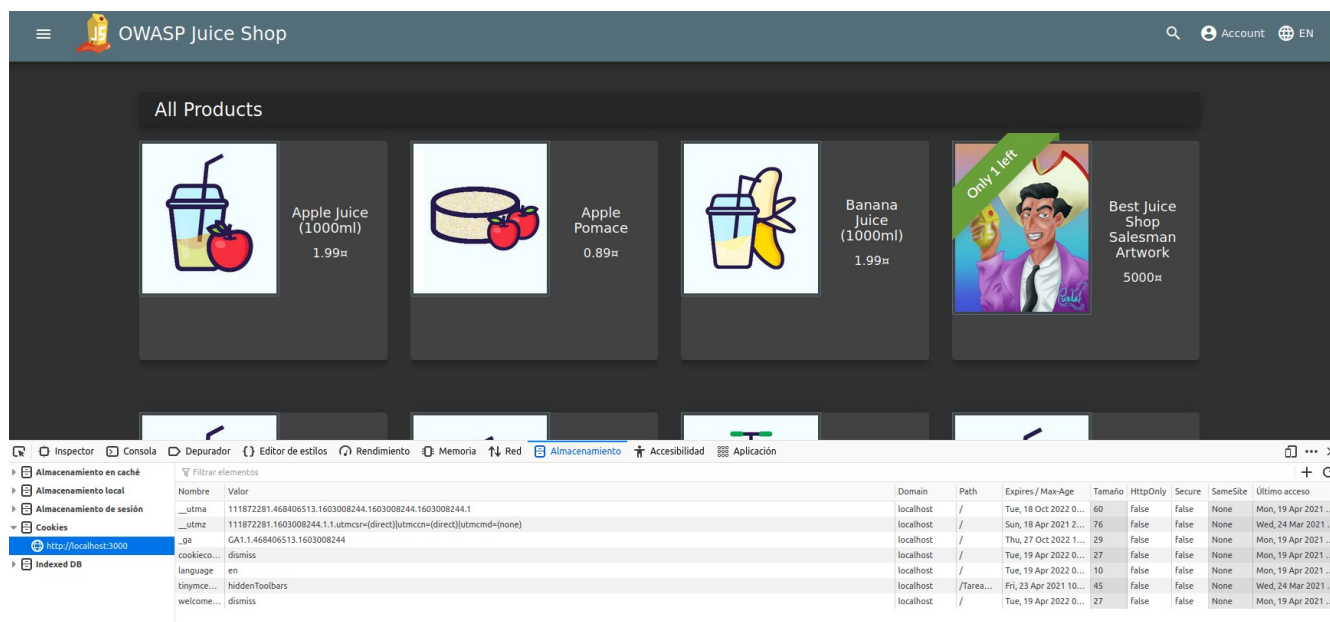
Actividad recomendada.

Realiza el room WebFundamentals de TryHackMe.

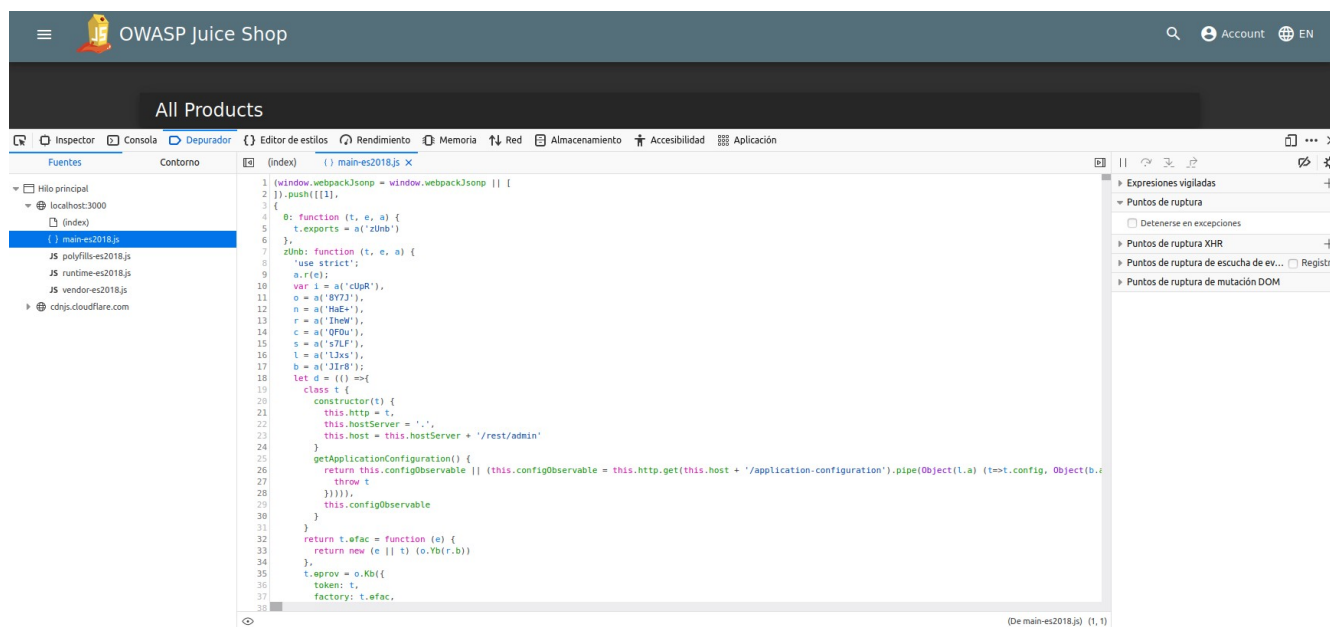
<https://tryhackme.com/room/webfundamentals>

Se recomienda emplear Mozilla Firefox o Google Chrome como navegador web y saber manejar las herramientas de desarrollo web de las que disponen. Durante la sesión de hacking, es altamente recomendable tener en todo momento abierta la **consola Javascript** ya que puede ofrecer información de depuración o de errores muy valiosa.

Otra herramienta importante es el **analizador de red** donde se puede analizar e incluso modificar (en Firefox) las peticiones realizadas, así como el almacenamiento local y de cookies empleado por la aplicación.



Por último, también es importante el análisis del código de la aplicación al que podemos acceder a través de la pestaña **Depurador**.



Se recomienda consultar el enlace siguiente para obtener más información sobre acciones que se pueden realizar con los navegadores, como editar cookies o editar el almacenamiento local, etc. Un resumen de las diferentes características disponibles en los diferentes navegadores se puede ver en el siguiente cuadro.

Function	Google Chrome	Mozilla Firefox	Edge/IE	Safari
Switching User Agents	✓	✓	✓	✓
Edit and Replay Requests	✗	✓	✗	✗
Editing Cookies	✓	✓	✓	✗
Editing Local Storage	✓	✓	✓	✗
Disable CSS	✓	✓	✓	✓
Disable Javascript	✓	✓	✗	✓
View Headers	✓	✓	✓	✓
Native screen-shot capture	✓	✓	✓	✗
Offline mode	✓	✓	✗	✗
Encode and Decode	✓	✓	✓	✓

Recurso

Web app security testing with browser.

<https://getmantra.com/web-app-security-testing-with-browsers/>

Por último, es recomendable conocer y manejar proxys como Burp Suite o ZAP para interceptar las peticiones que se realizan entre cliente y servidor, para poder analizarlas y/o modificarlas.

Actividad recomendada.

Realiza el seminario del INCIBE sobre el manejo de OWASP ZAP.

<https://www.incibe-cert.es/seminarios-web/uso-owasp-zap>

Reconociendo la web.

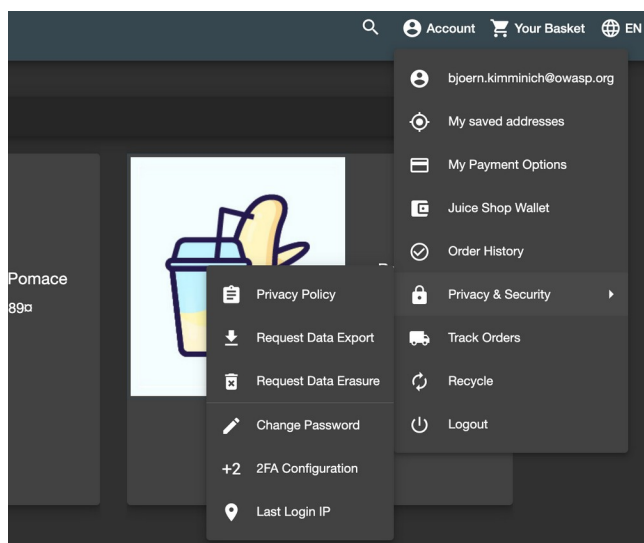
Antes de comenzar a realizar ataques sobre la web es necesario realizar un reconocimiento de la misma para comprender su funcionamiento y tratar de obtener pistas sobre posibles debilidades, fugas de información, vulnerabilidades, etc.

Juice Shop es una tienda online que dispone de todos los elementos de este tipo de aplicaciones (lista de productos, carrito de la compra, registro de usuarios, etc.). A continuación se detallan algunos de los elementos que se recomienda visitar y revisar:

- **Listado de productos.** Al hacer clic sobre los productos se muestra una descripción de los mismos y comentarios de revisión que han hecho los usuarios. Los usuarios autenticados pueden añadir/editar comentarios y votar los comentarios de otros usuarios. Además, se puede utilizar un búsqueda para filtrar productos, accesible a través del icono lupa.
- **Formulario de login.** Para realizar compras es necesario registrarse. Es posible hacerlo con la cuenta de Google o introduciendo nuestros datos.
- **Formulario de registro.** Si no tenemos cuenta, podemos registrarnos introduciendo los datos de usuario y contraseña. También es necesario introducir una frase de recordatorio de

contraseña.

- Recordatorio de contraseña. Para recuperar la contraseña se debe introducir nuestro correo electrónico y responder a la pregunta de seguridad que rellenamos en el formulario de registro.
- **Carro de la compra.** Permite realizar las acciones propias de este tipo de elemento.
- **Proceso de compra.** Se introduce la dirección de entrega, método de envío y opciones de pago. También es posible agregar códigos descuento.
- Seguimiento de compras.
- **Menú de usuario.** El menú de usuario dispone de numerosos elementos que es recomendable revisar y conocer.



Ejercicio propuesto

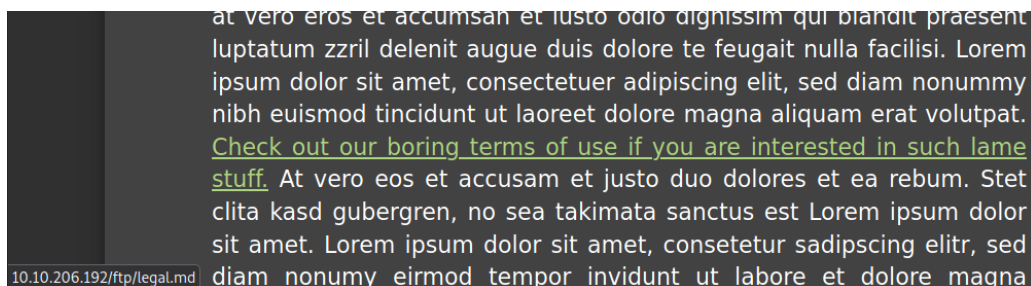
Realiza un reconocimiento de la web y recopila todas las direcciones de correo de los usuarios que te sean posibles. ¿Cuál es la dirección de correo del usuario administrador?

3. Caso práctico

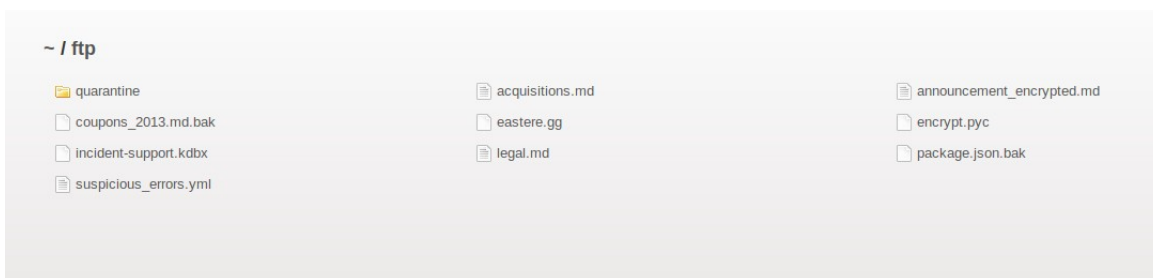
En este caso práctico explicaremos alguno de los retos contenidos en la aplicación Juice Shop.

Acceso a un documento confidencial.

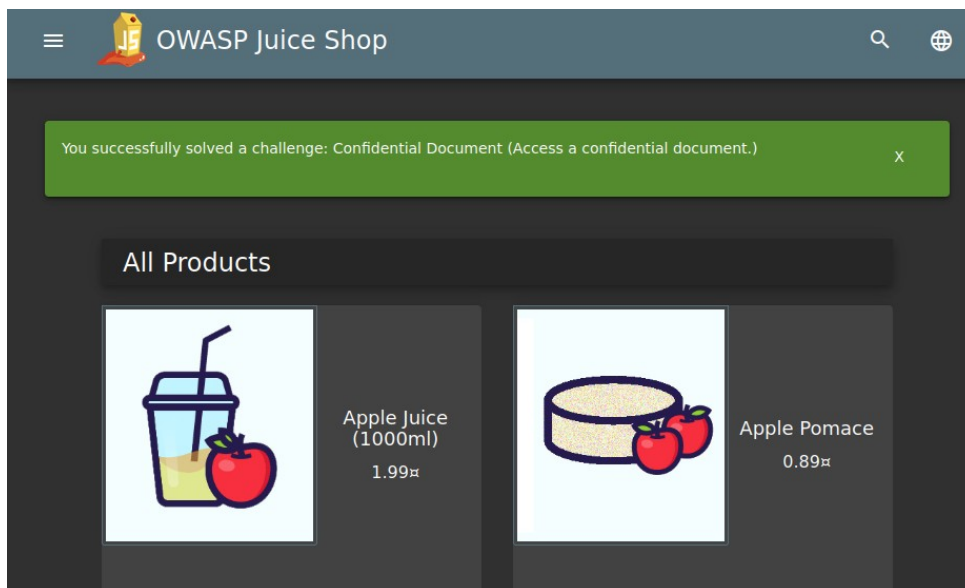
Si hemos hecho un buen reconocimiento de la web es posible que ya sepamos cómo resolver este reto. Navegamos a la **sección About Us** en el menú principal de la aplicación y leemos el texto que contiene. En él aparece un enlace a los términos de uso, no nos interesan estos, sino la URL que enlaza.



Observamos que el servidor tiene la **carpeta ftp** cuyo contenido es totalmente accesible.



Este directorio tiene numerosos ficheros que pueden ser relevantes para otros retos. Descargamos el fichero **acquisitions.md** y volvemos a la página principal donde se nos mostrará el aviso de nuestro primer reto resuelto.



Descarga del fichero de copia de seguridad.

En el directorio *ftp* se encuentra el fichero de copia de seguridad **package.json.bak**, pero si tratamos de descargarlo el servidor responde con un error 403 en el que solo los ficheros con extensión *.md* y *.pdf* pueden ser descargados.

OWASP Juice Shop (Express ^4.17.1)

403 Error: Only .md and .pdf files are allowed!

```
at verify (/juice-shop/routes/fileServer.js:30:12)
at /juice-shop/routes/fileServer.js:16:7
at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at trim_prefix (/juice-shop/node_modules/express/lib/router/index.js:317:13)
at /juice-shop/node_modules/express/lib/router/index.js:284:7
at param (/juice-shop/node_modules/express/lib/router/index.js:354:14)
at param (/juice-shop/node_modules/express/lib/router/index.js:365:14)
at Function.process_params (/juice-shop/node_modules/express/lib/router/index.js:410:3)
at next (/juice-shop/node_modules/express/lib/router/index.js:275:10)
at /juice-shop/node_modules/serve-index/index.js:145:39
at FSReqCallback.oncomplete (fs.js:172:5)
```

Para poder saltar esta restricción usaremos un ataque **Poison Null Byte**, de este modo, si la comprobación de seguridad no se ha realizado correctamente, haremos un bypass de la misma.

El null byte se considera %00 por lo que a partir de este carácter la cadena terminaría, pero según cómo se manejen los objetos que almacenan cadenas en los diferentes lenguajes de programación, es posible que el resto de la cadena sea tenida en cuenta. Por ejemplo: *archivo.txt%00.jpg* simularía un fichero con extensión *.jpg* (que su descarga está permitida), cuando en realidad el fichero tiene extensión *.txt*.

Recursos

Ataques de Null Byte.

<https://portswigger.net/blog/null-byte-attacks-are-alive-and-well>

<http://projects.webappsec.org/w/page/13246949/Null%20Byte%20Injection>

Si probamos con el enlace ***ftp/package.json.bak%00.md*** no obtendremos el fichero, pero sí un mensaje de error 400 (*BadRequestError*) con el que habremos resuelto otro reto. Para descargar el fichero correctamente es necesario codificar también el carácter % en la URL.

Ejercicio propuesto.

Averigua cómo codificar en la URL el carácter % para descargar el fichero y resolver este reto.

Acceso como usuario administrador.

Hay dos retos disponibles para acceder con la cuenta de administrador en Juice Shop.

El primero de ellos consiste en un ataque de **SQL Injection** en el panel de login. En la fase de reconocimiento hemos averiguado cuál era la cuenta de correo del administrador: ***admin@juice-sh.op***

Accedemos a la página de login y nos aseguramos que tenemos configurado nuestro proxy reverso para interceptar las peticiones que realice el navegador al servidor Juice Shop. Introducimos cualquier dato en el formulario y comprobamos la intercepción. Hacemos forward hasta que nos muestra los datos de la petición.

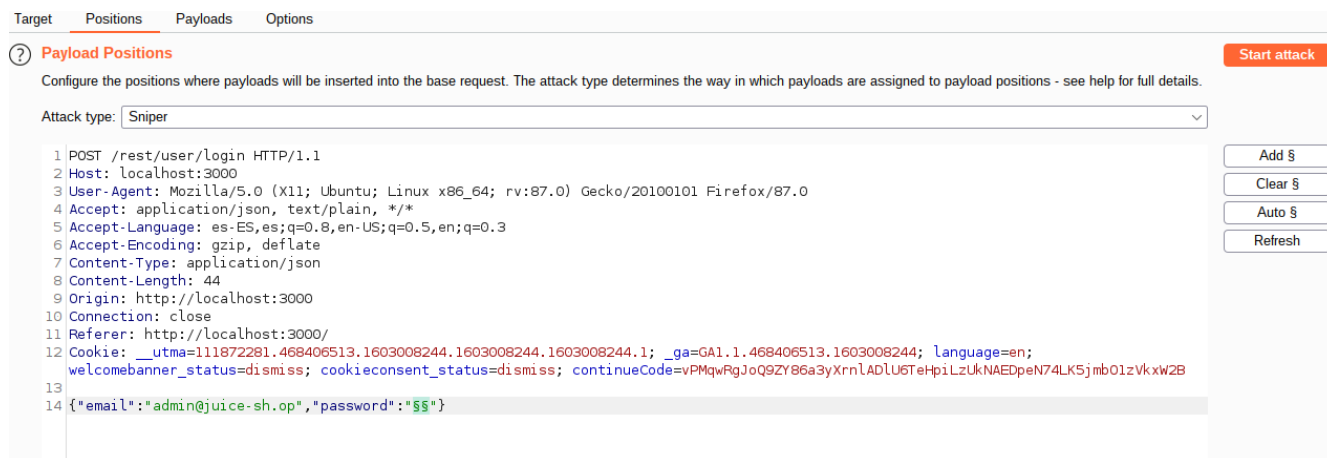


Modificamos el campo *email* por la siguiente **inyección SQL**: `' or 1=1--`

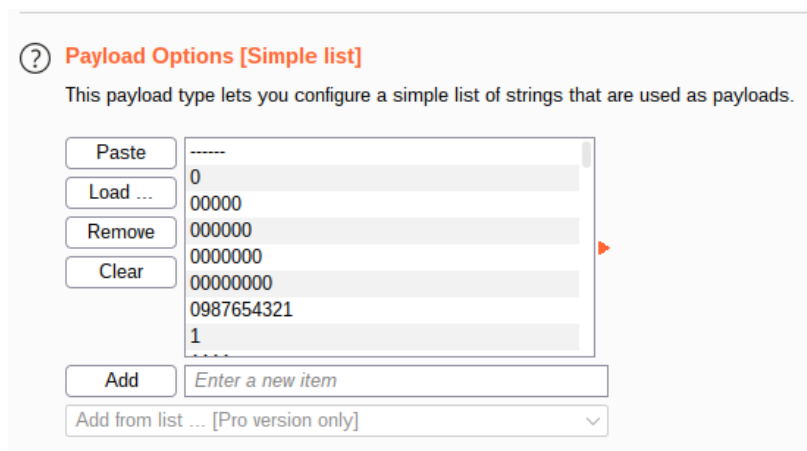
Al pulsar en forward se mostrará el mensaje de reto solucionado y veremos que tenemos acceso a los datos de la cuenta de administrador.

El segundo reto para acceder como administrador es intentando un ataque de fuerza bruta. Nos deslogueamos si estamos logueados y volvemos al formulario de login. Introducimos los datos en el formulario, y volvemos a interceptar la petición. Ahora en lugar de hacer forward, iremos a **Action – Send to intruder**.

Accedemos a la ventana **Intruder – Positions** para indicar las posiciones del formulario que serán utilizadas en el ataque de fuerza bruta. Las posiciones se marcan entre dos caracteres §. A la derecha, pulsamos el botón **Clear §** y a continuación lo añadimos en el campo password con el botón **Add §**.



A continuación accedemos a la pestaña **Payloads** para indicar el fichero de diccionario que usaremos en el ataque. Será el fichero **best1050.txt** de Seclists que debemos cargar en la sección **Payload Options [Simple list]**.



? **Payload Options [Simple list]**

This payload type lets you configure a simple list of strings that are used as payloads.

Paste Load ... Remove Clear

0
00000
000000
0000000
00000000
0987654321
1
.....

Add Enter a new item

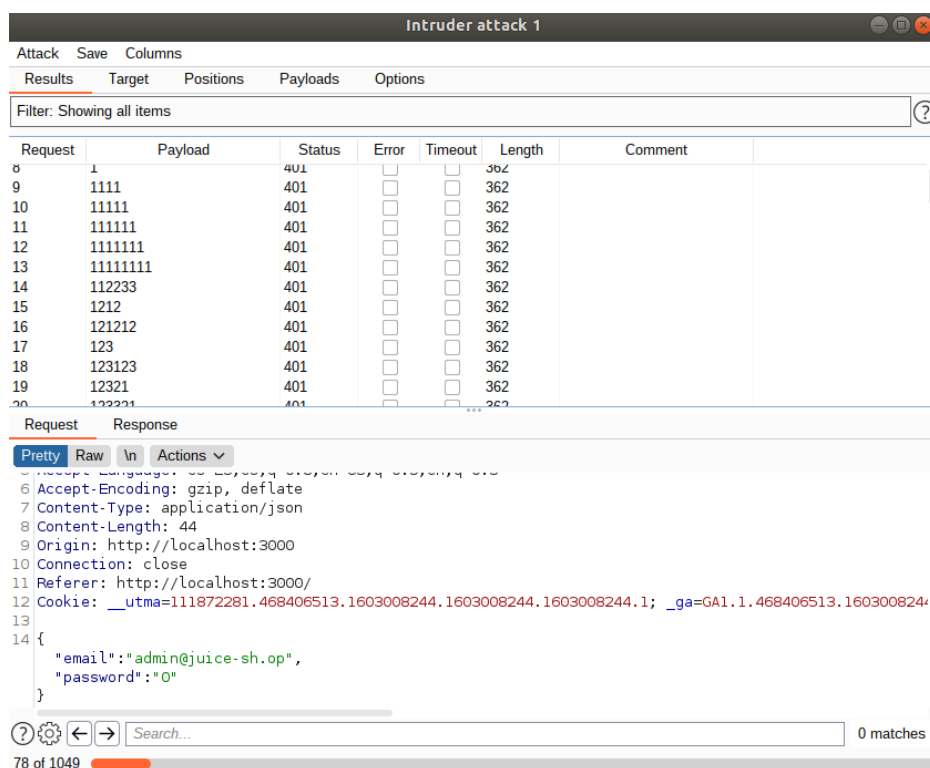
Add from list ... [Pro version only]

Recurso

Fichero de contraseñas para ataque de diccionario.

<https://github.com/danielmiessler/SecLists/blob/master/Passwords/Common-Credentials/best1050.txt>

Una vez cargado el fichero, podemos comenzar el ataque, pulsando el botón *Start attack*. Las peticiones fallidas responden con el código 401 (Unauthorized) mientras que una petición exitosa devolverá el código 200 (OK). Podemos utilizarlo para filtrar el resultado y obtener la credencial correcta. Cuando lo hayamos hecho, podemos loguearnos en la cuenta de administrador para superar este reto.



Acceso a la página de administración.

Si hemos sido curiosos y hemos investigado la cuenta de administrador, habremos observado que en el menú del usuario no se encuentra ningún acceso a la página de administración donde se puedan crear o modificar usuario, etc.

Accede a las herramientas de desarrollador web (F12), a la pestaña **Depurador** y visualiza el código del fichero **main-es2018.js**. Busca en el código (**ctrl+f**) la palabra **admin**. Buscamos el enlace de la página de administración que debe ser similar a “**path:administration**”. Esto genera el enlace que estamos buscando a **##administration**.



Ejercicio propuesto.

Modifica el identificador de la petición para visualizar el carrito de compra de otro usuario.

Accediendo al tablero de puntuación.

La aplicación Juice Shop tiene un tablero de puntuación (scoreboard) que contiene todos los retos disponibles en la aplicación, sin embargo, no hay ningún enlace disponible desde los menús de la aplicación para poder acceder a él. Por tanto, debemos averiguarlo, o bien tratando de adivinar la url, o bien por fuerza bruta, o bien buscando en otros lugares no visibles como el código de la aplicación.

El tablero de puntuación se puede localizar de forma similar a la página de administración.

Ejercicio propuesto.

Sigue los pasos descritos anteriormente para hallar la URL del tablero de puntuación.

El tablero de puntuación es una herramienta útil para seguir tu progreso de hacking en Juice Shop. En él puedes consultar todos los retos clasificados por dificultad (de 1 a 6 estrellas), y también por categoría. Puedes obtener pistas e incluso revisar ejemplos de código que son ejemplos de cómo se producen dichas vulnerabilidades.

The screenshot shows the OWASP Juice Shop Score Board. At the top, there's a notification: "You successfully solved a challenge: Score Board (Find the carefully hidden 'Score Board' page.)". Below this, the "Score Board" section shows a progress bar at 5%. The interface includes a navigation bar with difficulty levels (1-6 stars) and filters for "Show all", "Show solved", "Show tutorials only", and "Show unavailable". A category bar lists various security issues like "Broken Access Control", "Broken Anti Automation", etc. The main table lists challenges with columns for Name, Difficulty, Description, Category, Tags, and Status.

Name	Difficulty	Description	Category	Tags	Status
Admin Section	★★	Access the administration section of the store.	Broken Access Control	Good for Demos	unsolved
Bonus Payload	★	Use the bonus payload <iframe width="100%" height="166" scrolling="no" frameborder="no" allow="autoplay" src="https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com/tracks/771984076&color=%23ff5500&auto_play=true&hide_related=false&show_comments=true&show_user=true&show_reposts=false&show_teaser=true"></iframe> in the DOM XSS challenge.	XSS	Shenanigans Tutorial	unsolved
Bully Chatbot	★	Receive a coupon code from the support chatbot.	Miscellaneous	Brute Force Shenanigans	unsolved

¡Ahora ya estás preparado/a para continuar con el resto de retos que tienes a tu disposición!

3. Ejercicios propuestos

En el caso práctico se han analizado algunos de los retos presentes en la aplicación web vulnerable Juice Shop. Para seguir avanzando en tu aprendizaje se propone que realices los siguientes retos.

#	Nombre	Descripción	Categoría
1	Login Bender	Logueate en la cuenta del usuario Bender	Injection
2	User Credentials	Obtén la lista de todas las credenciales de usuario via SQLi	Injection
3	CAPTCHA bypass	Envía 10 o más reseñas de clientes en 10 segundos	Broken Anti Automation
4	Two Factor Authentication	Resuelve el doble factor de autenticación (2FA) del usuario "wurstbrot"	Broken Authentication
5	Upload Type	Sube un fichero que no tiene extensión .zip o .pdf	Improper Input Validation
6	CSRF	Cambia el nombre de un usuario mediante Cross-Site Request Forgery desde otro origen.	Broken Access Control
7	View Basket	Inspecciona el carrito de compra de otro usuario	Broken Access Control
8	Allowlist Bypass	Forzar una redirección a una página para la que no tenemos permisos.	Unvalidated Redirects
9	Unsigned JWT	Forja un token JWT no firmado que impersonalice al usuario (no existente) jwt3d@juice-sh.op.	Vulnerable Components
10	Weird Crypto	Informa a la tienda de un algoritmo o librería que no deberían usar del modo en que lo hacen.	Cryptographic Issues

4. Bibliografía

Recursos y enlaces utilizados para elaborar este documento.

- Room OWASP Juice Shop en TryHackMe. <https://tryhackme.com/room/owaspjuiceshop>
- Guía oficial completa para explotar Juice Shop. <https://pwning.owasp-juice.shop/>
- Uso de las herramientas del navegador para testeo. <https://getmantra.com/web-app-security-testing-with-browsers/>