

Reporte de Vulnerabilidades - juice-shop.herokuapp.com

Equipo: Chilaquiles

Integrantes:

- Chávez Sánchez Juan Daniel
- Nabor Lira Iván Damián
- Piedras Cruz Felipe de Jesus
- Venegas Barrita Edgar

Fecha: 28 de agosto de 2023

Vulnerabilidades encontradas:

SQL Injection

Descripción:

Un ataque de inyección SQL consiste en la inserción o “inyección” de una consulta SQL a través de los datos de entrada del cliente a la aplicación. Un exploit de inyección SQL exitoso puede leer datos confidenciales de la base de datos, modificar datos de la base de datos (Insertar/Actualizar/Eliminar), ejecutar operaciones de administración en la base de datos (como apagar el DBMS), recuperar el contenido de un archivo determinado presente en el archivo DBMS. sistema y en algunos casos emitir comandos al sistema operativo. Los ataques de inyección SQL son un tipo de ataque de inyección, en el que se inyectan comandos SQL en la entrada del plano de datos para afectar la ejecución de comandos SQL predefinidos.

El ataque de inyección SQL ocurre cuando:

- Un dato no deseado ingresa a un programa desde una fuente no confiable.
- Los datos se utilizan para construir dinámicamente una consulta SQL.

Los ataques de inyección SQL permiten a los atacantes falsificar la identidad, alterar los datos existentes, causar problemas de repudio como anular transacciones o cambiar saldos, permitir la divulgación completa de todos los datos en el sistema, destruir los datos o hacerlos no disponibles de otro modo y convertirse en administradores del sistema. servidor de base de datos.

La inyección SQL es muy común con aplicaciones PHP y ASP debido a la prevalencia de interfaces funcionales más antiguas. Debido a la naturaleza de las interfaces disponibles, es

menos probable que las aplicaciones J2EE y ASP.NET tengan inyecciones SQL fácilmente explotables.

La gravedad de los ataques de inyección SQL está limitada por la habilidad y la imaginación del atacante y, en menor medida, por las contramedidas de defensa en profundidad, como conexiones de bajo privilegio al servidor de la base de datos, etc. En general, considere la inyección SQL como una gravedad de alto impacto.

Las principales consecuencias son:

1. Confidencialidad: dado que las bases de datos SQL generalmente contienen datos confidenciales, la pérdida de confidencialidad es un problema frecuente con las vulnerabilidades de inyección SQL.
2. Autenticación: si se utilizan comandos SQL deficientes para verificar nombres de usuario y contraseñas, es posible conectarse a un sistema como otro usuario sin conocimiento previo de la contraseña.
3. Autorización: si la información de autorización se mantiene en una base de datos SQL, es posible cambiar esta información mediante la explotación exitosa de una vulnerabilidad de inyección SQL.
4. Integridad: Así como es posible leer información confidencial, también es posible realizar cambios o incluso eliminar esta información con un ataque de inyección SQL.

Evidencia:

Se encontró la vulnerabilidad al probar con la instrucción 'OR 1=1— y en el campo de contraseña usamos cualquier valor, lo cual nos permite acceder a la cuenta de admin@juice-sh.op.

Username: ' OR 1=1—

Password:

x

Login


Email *

'OR 1=1--

Password *


x

[Forgot your password?](#)



 Log in

☐ Remember me

or

 Log in with Google

User Profile



Email:
admin@juice-sh.op

Username:
e.g. SuperUser

Set Username

File Upload:

Elegir archivo No se ha seleccionado ningún archivo

Upload Picture

or

Image URL:
e.g. https://www.gravatar.com/avatar/526703ac2bd7c1

Link Image

Solución:

- No confíe en las entradas del lado del cliente, incluso si existe una validación del lado del cliente. En general, verifique todos los datos en el lado del servidor.
- Utilice procedimientos almacenados de la base de datos.
- No concatenar cadenas en consultas en el procedimiento almacenado, ni utilice funciones 'exec', 'exec inmediata' o equivalente.
- No crear consultas SQL dinámicas utilizando una concatenación de cadenas simple.
- Establezca una lista de caracteres permitidos o una lista denegada de caracteres no permitidos en la entrada del usuario.
- Aplique el principio de privilegio mínimo utilizando el usuario de base de datos con el menor privilegio posible.
- Otorgue el acceso mínimo a la base de datos que sea necesario para la aplicación.

Improper Input Validation

Descripción:

La validación de la entrada se realiza para garantizar que sólo los datos correctamente formados entran en el flujo de trabajo de un sistema de información, evitando que los datos malformados permanezcan en la base de datos y provoquen el mal funcionamiento de varios componentes posteriores. La validación de la entrada debe realizarse lo antes posible en el flujo de datos, preferiblemente en cuanto se reciben los datos de la parte externa.

Los datos de todas las fuentes potencialmente no fiables deben someterse a la validación de entrada, incluidos no sólo los clientes web orientados a Internet, sino también las fuentes backend a través de extranets, de proveedores, socios, vendedores o reguladores, cada uno de los cuales puede verse comprometido por sí mismo y empezar a enviar datos malformados.

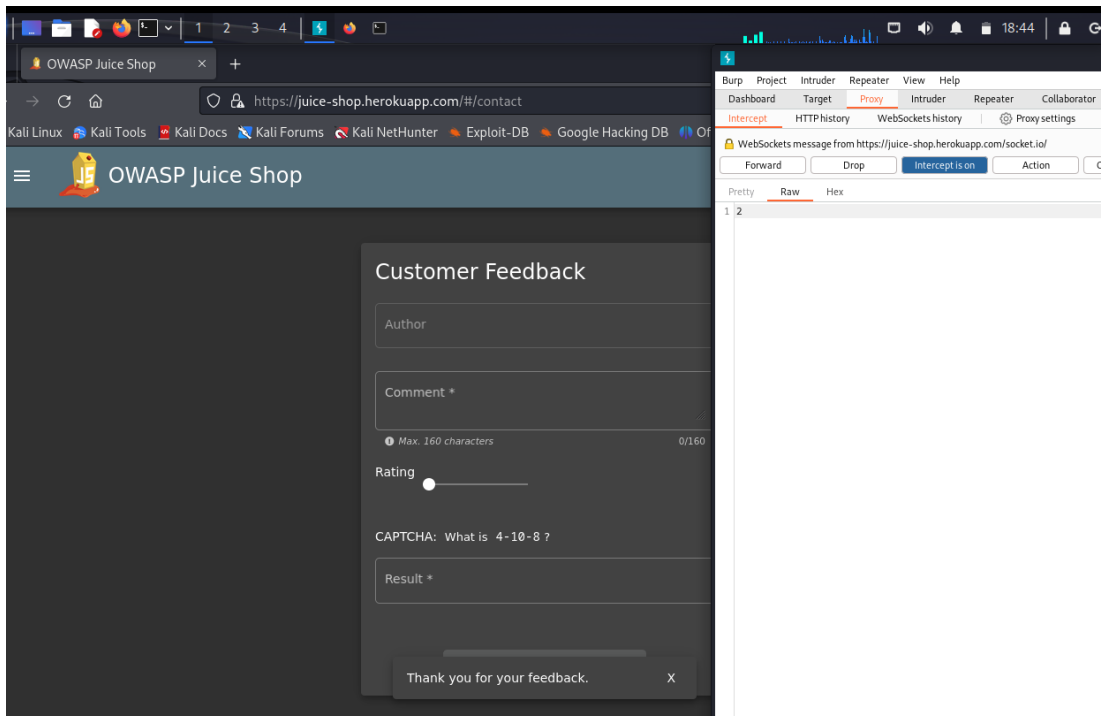
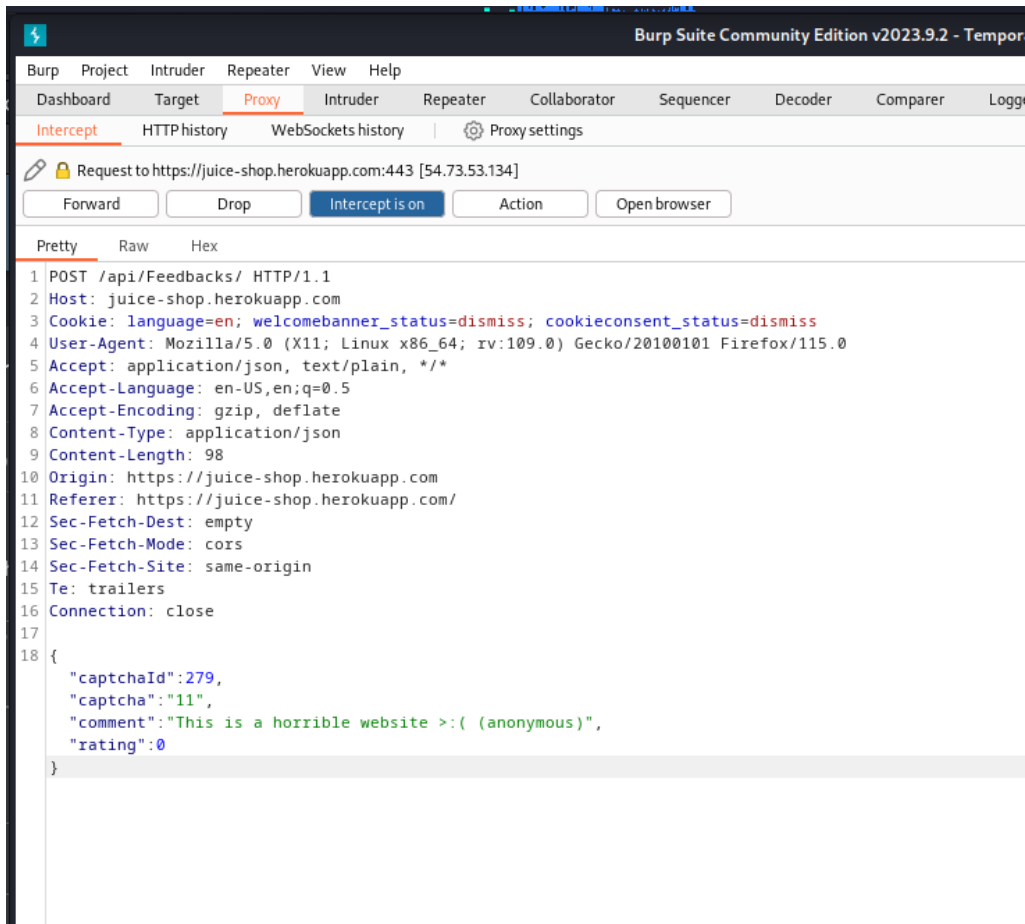
La validación de entrada no debe utilizarse como método principal de prevención de XSS, SQL Injection y otros ataques que se tratan en las respectivas hojas de trucos, pero puede contribuir significativamente a reducir su impacto si se aplica correctamente.

Interceptando la petición, es cómo podemos cambiar la información del encabezado rating a 0:

Evidencia:

The screenshot displays a web application on the left and a Burp Suite proxy tool on the right. The web application shows a 'Customer Feedback' form with fields for 'Author' (anonymous), 'Comment' (I'm about to zero rate this horrible website >:((anonymous)), 'Rating' (set to 1), and a CAPTCHA (What is 4 - 14). The Burp Suite tool is intercepting a POST request to https://juice-shop.herokuapp.com:443. The request details are shown in the 'Raw' tab, including headers like Host, Cookie, User-Agent, Accept, and Content-Type. The request body is a JSON object with 'captchaId', 'captcha', 'comment', and 'rating' fields.

```
1 POST /api/Feedbacks/ HTTP/1.1
2 Host: juice-shop.herokuapp.com
3 Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
5 Accept: application/json, text/plain, */*
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Content-Type: application/json
9 Content-Length: 117
10 Origin: https://juice-shop.herokuapp.com
11 Referer: https://juice-shop.herokuapp.com/
12 Sec-Fetch-Dest: empty
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Site: same-origin
15 Te: trailers
16 Connection: close
17
18 {
19   "captchaId":272,
20   "captcha":"-14",
21   "comment":"I'm about to zero rate this horrible website >:( (anonymous)",
22   "rating":1
23 }
```



Directory Listing (CWE-548)

Descripción:

Los servidores web pueden configurarse para enlistar automáticamente el contenido de los directorios que no tienen una página de índice presente. Esto puede ayudar a un atacante permitiéndole identificar rápidamente los recursos en una ruta dada, y proceder directamente a analizar y atacar esos recursos. En particular, aumenta la exposición de archivos sensibles dentro del directorio que no están destinados a ser accesibles a los usuarios, tales como archivos temporales y volcados de memoria.

Los listados de directorios en sí mismos no constituyen necesariamente una vulnerabilidad de seguridad. En cualquier caso, cualquier recurso sensible dentro de la raíz de la web debe tener un control de acceso adecuado y no debe ser accesible por una parte no autorizada que conozca o adivine la URL. Incluso cuando los listados de directorios están desactivados, un atacante puede adivinar la ubicación de archivos sensibles utilizando herramientas automatizadas.

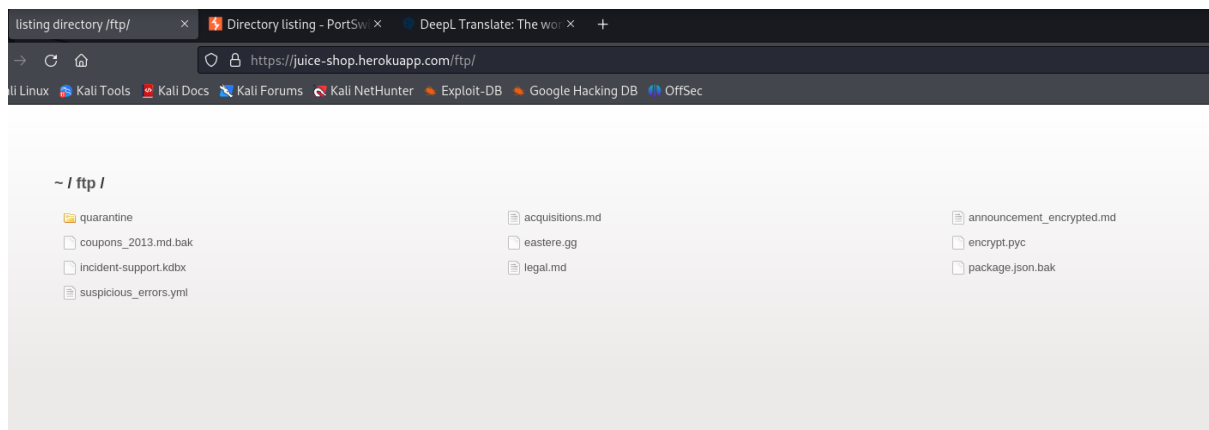
A través del descubrimiento y lectura del archivo de avisos de privacidad y legales, es como se sabe de la existencia del uso del protocolo FTP en uso dentro del servidor, mediante la manipulación de la URL: <https://juice-shop.herokuapp.com/ftp/> agregando /ftp/ al final de la liga, es así como se obtiene acceso a la carpeta raíz.

Remediación:

Normalmente no hay ninguna buena razón para proporcionar listados de directorios, y desactivarlos puede poner obstáculos adicionales en el camino de un atacante. Esto puede conseguirse normalmente de dos maneras:

- Configure su servidor web para evitar listados de directorios para todas las rutas por debajo de la raíz web;
- Colocar en cada directorio un archivo por defecto (como index.htm) que el servidor web mostrará en lugar de devolver un listado de directorios.

Evidencia:



Information Disclosure

La divulgación de información, también conocida como fuga de información, se produce cuando un sitio web revela involuntariamente información sensible a sus usuarios. Dependiendo del contexto, los sitios web pueden filtrar todo tipo de información a un atacante potencial, incluyendo:

- Datos sobre otros usuarios, como nombres de usuario o información financiera.
- Datos comerciales o empresariales confidenciales.
- Detalles técnicos sobre el sitio web y su infraestructura.

Los peligros de filtrar datos comerciales o de usuarios sensibles son bastante obvios, pero revelar información técnica a veces puede ser igual de grave. Aunque parte de esta información tendrá un uso limitado, puede ser potencialmente un punto de partida para exponer una superficie de ataque adicional, que puede contener otras vulnerabilidades interesantes. Los conocimientos que se consigan reunir podrían incluso proporcionar la pieza que falta en el rompecabezas cuando se trate de construir ataques complejos de gran gravedad.

Ocasionalmente, la información sensible puede filtrarse por descuido a usuarios que simplemente están navegando por el sitio web de forma normal. Sin embargo, lo más habitual es que un atacante tenga que provocar la revelación de información interactuando con el sitio web de forma inesperada o maliciosa.

Cómo prevenir las vulnerabilidades de divulgación de información.

Evitar por completo la divulgación de información es complicado debido a la enorme variedad de formas en que puede producirse. Sin embargo, hay algunas buenas prácticas generales que puedes seguir para minimizar el riesgo de que este tipo de vulnerabilidades se cuelen en tus propios sitios web.

Asegúrese de que todas las personas implicadas en la producción del sitio web son plenamente conscientes de qué información se considera sensible. A veces, información aparentemente inofensiva puede ser mucho más útil para un atacante de lo que la gente cree. Destacar estos peligros puede ayudar a garantizar que la información sensible se maneja de forma más segura en general en su organización.

Audite cualquier código para detectar posibles revelaciones de información como parte de sus procesos de control de calidad o compilación. Debería ser relativamente fácil automatizar algunas de las tareas asociadas, como eliminar los comentarios de los desarrolladores.

Utilice mensajes de error genéricos en la medida de lo posible. No proporcione innecesariamente a los atacantes pistas sobre el comportamiento de la aplicación.

Compruebe que todas las funciones de depuración o diagnóstico están desactivadas en el entorno de producción.

Asegúrese de que entiende completamente los ajustes de configuración y las implicaciones de seguridad de cualquier tecnología de terceros que implemente. Tómese su tiempo para investigar y desactivar cualquier función o configuración que no necesite realmente.

Descripción:

Se accede a los términos y condiciones de la página, mientras se sigue el historial en http en Burp Suite.

The screenshot displays the Burp Suite interface. The top menu bar includes options like Burp, Project, Intruder, Repeater, View, and Help. Below the menu, there's a toolbar with various tools such as Dashboard, Target, Proxy, Intruder, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, Extensions, and Settings. The main panel shows a list of intercepted HTTP requests. The request at index 224 is highlighted, showing a GET request to `https://juice-shop.herokuapp.com/ftp/legal.md` with a status code of 200. Below this list, the 'Request' and 'Response' tabs are visible. The 'Request' tab shows the raw HTTP request details, including headers like `Host: juice-shop.herokuapp.com`, `User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/2010101 Firefox/115.0`, and `Referer: https://juice-shop.herokuapp.com/`. The 'Response' tab shows the raw HTTP response details, including headers like `Server: Cowboy`, `Content-Type: text/markdown; charset=UTF-8`, and `Content-Length: 3047`. On the right side, the 'Inspector' panel shows the request attributes, cookies, headers, and response headers.

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title
217	https://juice-shop.herokuapp.com	GET	/socket.io/?EIO=4&transport=polling&t...	✓		400	209	JSON	io/	
218	https://juice-shop.herokuapp.com	POST	/socket.io/?EIO=4&transport=polling&t...	✓		400	209	JSON	io/	
219	https://juice-shop.herokuapp.com	GET	/socket.io/?EIO=4&transport=polling&t...	✓		200	264	JSON	io/	
220	https://juice-shop.herokuapp.com	POST	/socket.io/?EIO=4&transport=polling&t...	✓		200	153	text	io/	
221	https://juice-shop.herokuapp.com	GET	/socket.io/?EIO=4&transport=polling&t...	✓		200	200	JSON	io/	
222	https://juice-shop.herokuapp.com	GET	/socket.io/?EIO=4&transport=websocket...	✓		101	145	io/	io/	
223	https://juice-shop.herokuapp.com	GET	/socket.io/?EIO=4&transport=polling&t...	✓		200	168	text	io/	
224	https://juice-shop.herokuapp.com	GET	/ftp/legal.md			200	3521	text	md	
225	https://juice-shop.herokuapp.com	GET	/socket.io/?EIO=4&transport=polling&t...	✓				io/	io/	
226	https://juice-shop.herokuapp.com	GET	/socket.io/?EIO=4&transport=polling&t...	✓				io/	io/	
227	https://juice-shop.herokuapp.com	GET	/socket.io/?EIO=4&transport=polling&t...	✓				io/	io/	
228	https://juice-shop.herokuapp.com	GET	/socket.io/?EIO=4&transport=polling&t...	✓				io/	io/	

Request

1 GET /ftp/legal.md HTTP/1.1
2 Host: juice-shop.herokuapp.com
3 Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/2010101 Firefox/115.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Referer: https://juice-shop.herokuapp.com/
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-User: ?1
14 Te: xxxxxx

Response

1 HTTP/1.1 200 OK
2 Server: Cowboy
3 Connection: close
4 Access-Control-Allow-Origin: *
5 X-Content-Type-Options: nosniff
6 X-Frame-Options: SAMEORIGIN
7 Feature-Policy: payment 'self'
8 X-Recruiting: /#/jobs
9 Accept-Ranges: bytes
10 Cache-Control: public, max-age=0
11 Last-Modified: Mon, 28 Aug 2023 15:56:31 GMT
12 Etag: W/"be7-18a3cdc63b5"
13 Content-Type: text/markdown; charset=UTF-8
14 Vary: Accept-Encoding
15 Date: Mon, 28 Aug 2023 22:52:28 GMT
16 Via: 1.1 vegur
17 Content-Length: 3047
18

Inspector

Request attributes: 2
Request cookies: 3
Request headers: 14
Response headers: 16

Se encuentra un archivo legal.md, para la descarga de este se accede al servidor FTP.

1 x +

SendCancel<>

Target: https://juice-shop.herokuapp.com HTTP/1

Request

PrettyRawHex

1 GET /ftp HTTP/1.1
2 Host: juice-shop.herokuapp.com
3 Cookie: language=en; welcomebanner_status=dismiss;
4 cookieconsent_status=dismiss
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0)
6 Gecko/20100101 Firefox/115.0
7 Accept:
8 text/html,application/xhtml+xml,application/xml;q=0.9,i
9 mage/avif,image/webp,*/*;q=0.8
10 Accept-Language: en-US,en;q=0.5
11 Accept-Encoding: gzip, deflate
12 Referer: https://juice-shop.herokuapp.com/
13 Upgrade-Insecure-Requests: 1
14 Sec-Fetch-Dest: document
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-Site: same-origin
17 Te: trailers
18 Connection: close

Response

PrettyRawHexRender

359
</h1>
<ul id="files" class="view-tiles">

<a href="ftp/quarantine" class="icon
icon-directory" title="quarantine">

quarantine

5/19/2023 10:53:26 PM

360
<a href="ftp/acquisitions.md" class="icon
icon icon-md icon-text" title="
acquisitions.md">

acquisitions.md

909

5/19/2023 10:53:26 PM

361
<a href="ftp/announcement_encrypted.md"
class="icon icon-md icon-text" title=
"announcement_encrypted.md">

Inspector

Request attributes2
Request query parameters0
Request body parameters0
Request cookies3
Request headers14
Response headers12

Search...0 highlights

acqui3 matches

Done

11,413 bytes | 583 millis

Send

Cancel

<|>

Target: https/

Request

PrettyRawHex

1 GET /ftp/legal.md HTTP/1.1

2 Host: juice-shop.herokuapp.com

3 Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss

4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0

5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8

6 Accept-Language: en-US,en;q=0.5

7 Accept-Encoding: gzip, deflate

8 Referer: https://juice-shop.herokuapp.com/

9 Upgrade-Insecure-Requests: 1

10 Sec-Fetch-Dest: document

11 Sec-Fetch-Mode: navigate

12 Sec-Fetch-Site: same-origin

13 Sec-Fetch-User: ?1

14 Te: trailers

15 Connection: close

16

17

Response

PrettyRawHexRender

1 HTTP/1.1 200 OK

2 Server: Cowboy

3 Connection: close

4 Access-Control-Allow-Origin: *

5 X-Content-Type-Options: nosniff

6 X-Frame-Options: SAMEORIGIN

7 Feature-Policy: payment 'self'

8 X-Recruiting: /#/jobs

9 Accept-Ranges: bytes

10 Cache-Control: public, max-age=0

11 Last-Modified: Mon, 28 Aug 2023 15:56:31 GMT

12 Etag: W/"be7-18a3cdc63b5"

13 Content-Type: text/markdown; charset=UTF-8

14 Vary: Accept-Encoding

15 Date: Mon, 28 Aug 2023 23:01:25 GMT

16 Via: 1.1 vegur

17 Content-Length: 3047

18

19 # Legal Information

20

21 Lorem ipsum dolor sit amet, consetetur

22 sadipscing elitr, sed diam nonumy

23 eirmod tempor invidunt ut labore et dolore magna

24 aliquyam erat, sed diam

25 voluptua. At vero eos et accusam et justo duo

26 dolores et ea rebum. Stet

27 clita kasd gubergren, no sea takimata sanctus

28 est Lorem ipsum dolor sit

29 amet. Lorem ipsum dolor sit amet, consetetur

30

0 highlights

Search...

0 highlights

acqui

En el ftp se encuentra un archivo con nombre acquisitions.md

Request

PrettyRawHex

1 GET /ftp/acquisitions.md HTTP/1.1

2 Host: juice-shop.herokuapp.com

3 Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss

4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0

5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8

6 Accept-Language: en-US,en;q=0.5

7 Accept-Encoding: gzip, deflate

8 Referer: https://juice-shop.herokuapp.com/

9 Upgrade-Insecure-Requests: 1

10 Sec-Fetch-Dest: document

11 Sec-Fetch-Mode: navigate

12 Sec-Fetch-Site: same-origin

13 Sec-Fetch-User: ?1

14 Te: trailers

15 Connection: close

16

17

Response

PrettyRawHexRender

16 Date: Mon, 28 Aug 2023 22:59:27 GMT

17 Via: 1.1 vegur

18

19 # Planned Acquisitions

20

21 > This document is confidential! Do not

22 distribute!

23

24 Our company plans to acquire several competitors

25 within the next year.

26 This will have a significant stock market impact

27 as we will elaborate in

28 detail in the following paragraph:

29

30 Lorem ipsum dolor sit amet, consetetur

31 sadipscing elitr, sed diam nonumy

32 eirmod tempor invidunt ut labore et dolore magna

33 aliquyam erat, sed diam

34 voluptua. At vero eos et accusam et justo duo

35 dolores et ea rebum. Stet

36 clita kasd gubergren, no sea takimata sanctus

37 est Lorem ipsum dolor sit

38 amet. Lorem ipsum dolor sit amet, consetetur

39 sadipscing elitr, sed diam

40 nonumy eirmod tempor invidunt ut labore et

41 dolore magna aliquyam erat,

42 sed diam voluptua. At vero eos et accusam et

43 justo duo dolores et ea

44 rebum. Stet clita kasd gubergren, no sea

45

Inspector

Request attributes2

Request query parameters0

Request body parameters0

Request cookies3

Request headers14

Response headers16

0 highlights

Search...

0 highlights

acqui

Done

1,382 bytes | 2,459 millis

XSS

¿Qué es el cross-site scripting (XSS)?

Cross-site scripting (también conocido como XSS) es una vulnerabilidad de seguridad web que permite a un atacante poner en peligro las interacciones que los usuarios tienen con una aplicación vulnerable. Permite a un atacante eludir la misma política de origen, que está diseñada para segregar diferentes sitios web entre sí. Las vulnerabilidades de secuencias de comandos entre sitios normalmente permiten a un atacante hacerse pasar por un usuario víctima, llevar a cabo cualquier acción que el usuario pueda realizar y acceder a cualquiera de los datos del usuario. Si el usuario víctima tiene acceso privilegiado dentro de la aplicación, entonces el atacante podría obtener el control total sobre toda la funcionalidad y los datos de la aplicación.

¿Cómo funciona XSS?

Las secuencias de comandos entre sitios funcionan manipulando un sitio web vulnerable para que devuelva JavaScript malicioso a los usuarios. Cuando el código malicioso se ejecuta dentro del navegador de la víctima, el atacante puede comprometer completamente su interacción con la aplicación.

Prueba de concepto XSS

Puede confirmar la mayoría de los tipos de vulnerabilidad XSS inyectando una carga útil que hace que su propio navegador ejecute JavaScript arbitrario. Durante mucho tiempo ha sido una práctica común usar la función para este propósito porque es corta, inofensiva y bastante difícil de pasar por alto cuando se llama con éxito. De hecho, usted resuelve la mayoría de nuestros laboratorios XSS invocando en el navegador de una víctima simulada.

```
alert()alert()
```

Desafortunadamente, hay un ligero problema si usas Chrome. A partir de la versión 92 en adelante (20 de julio de 2021), se impide que los iframes de origen cruzado llamen a . Como

estos se utilizan para construir algunos de los ataques XSS más avanzados, a veces necesitará usar una carga útil PoC alternativa. En este escenario, recomendamos la función. Si está interesado en obtener más información sobre este cambio y por qué nos gusta, [consulte nuestra publicación de blog](#) sobre el tema. `alert()print()print()`

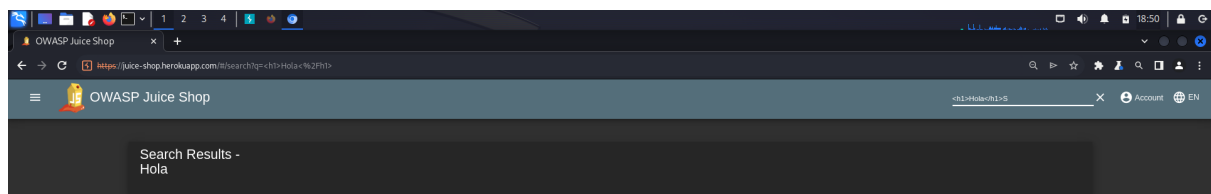
Como la víctima simulada en nuestros laboratorios usa Chrome, hemos modificado los laboratorios afectados para que también se puedan resolver con . Hemos indicado esto en las instrucciones siempre que sea relevante. `print()`

¿Cuáles son los tipos de ataques XSS?

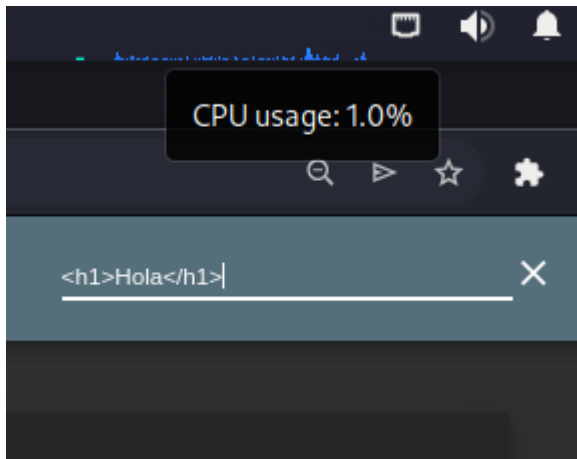
Hay tres tipos principales de ataques XSS. Estos son:

- [XSS reflejado](#), donde el script malicioso proviene de la solicitud HTTP actual.
- [XSS almacenado](#), donde el script malicioso proviene de la base de datos del sitio web.
- [XSS basado en DOM](#), donde la vulnerabilidad existe en el código del lado cliente en lugar del código del lado servidor.

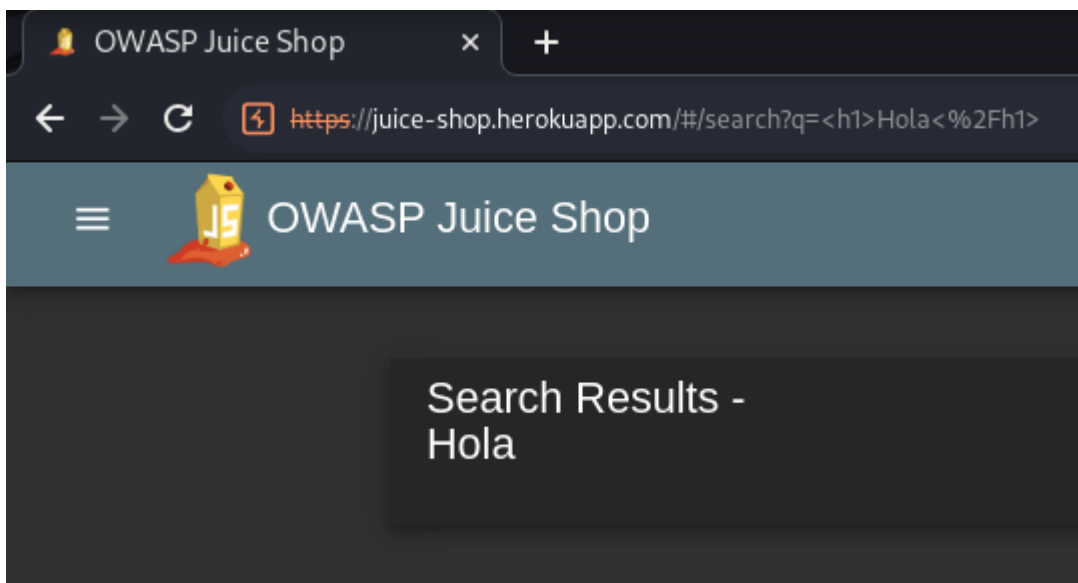
Primero testeamos el cuadro de búsqueda para ver que devuelve la aplicación



Usamos una etiqueta html para observar la respuesta del cuadro de búsqueda



Así comprobamos que podemos inyectar código que puede ser malicioso y detectamos la vulnerabilidad XSS



Cómo prevenir ataques XSS

Evitar el scripting entre sitios es trivial en algunos casos, pero puede ser mucho más difícil dependiendo de la complejidad de la aplicación y las formas en que maneja los datos controlables por el usuario.

En general, es probable que la prevención efectiva de las vulnerabilidades XSS implique una combinación de las siguientes medidas:

- **Filtrar la entrada a la llegada.** En el punto donde se recibe la entrada del usuario, filtre lo más estrictamente posible en función de lo que se espera o es una entrada válida.

- **Codificar datos en la salida.** En el punto donde los datos controlables por el usuario se generan en respuestas HTTP, codifique la salida para evitar que se interprete como contenido activo. Dependiendo del contexto de salida, esto podría requerir la aplicación de combinaciones de codificación HTML, URL, JavaScript y CSS.
- **Utilice los encabezados de respuesta adecuados.** Para evitar XSS en respuestas HTTP que no están pensadas para contener HTML o JavaScript, puede usar los encabezados y para asegurarse de que los exploradores interpreten las respuestas de la manera que desea. [Content-TypeX-Content-Type-Options](#)
- **Política de seguridad de contenido.** Como última línea de defensa, puede usar la directiva de seguridad de contenido (CSP) para reducir la gravedad de cualquier vulnerabilidad XSS que aún se produzca.