

Laboratorio 1

Iniciamos con el primer laboratorio en el que es un ataque de SQL injection y nuestro objetivo es obtener más datos a través de la consulta que se hace a la base de datos

Lab: SQL injection vulnerability in WHERE clause allowing retrieval of hidden data

APPRENTICE



LAB

Not solved

This lab contains a **SQL injection** vulnerability in the product category filter. When the user selects a category, the application carries out a SQL query like the following:

```
SELECT * FROM products WHERE category = 'Gifts' AND released = 1
```

Al hacer el ataque tenemos que modificar los parametros estan siendo enviados al servidor a través de la URL de forma que siempre se cumpla la condición.



SQL injection vulnerability in WHERE clause allowing retrieval of hidden data

LAB

Solved

[Back to lab description >>](#)

Congratulations, you solved the lab!

[Share your skills!](#)[Continue learning >>](#)[Home](#)

WE LIKE TO
SHOP 

' or 1=1--

Refine your search:

[All](#)[Accessories](#)[Clothing, shoes and accessories](#)[Food & Drink](#)[Tech gifts](#)

Laboratorio 2

Para el segundo laboratorio el objetivo es poder iniciar sesión como administrador haciendo igualmente un ataque de SQL injection

Lab: SQL injection vulnerability allowing login bypass

APPRENTICE



LAB

Not solved

This lab contains a **SQL injection** vulnerability in the login function.

To solve the lab, perform a SQL injection attack that logs in to the application as the `administrator` user.

Access the lab

La solución a este problema es escribir en el campo de username el usuario de administrator, esperando que esto exista en la base de datos y el campo de password solo tiene que tener información, no necesariamente necesita una palabra en especial.

Login

Username


administrator' --

Password

Log in

Y con eso hemos solucionado el lab

Congratulations, you solved the lab!

 Share your skills!

[Home](#) | [A](#)

My Account

Your username is: administrator

Email

Update email

Laboratorio 3

Procedemos con los ataques de Cross Site Scripting, lo cual es ejecutar código javascript en el buscador del usuario

Lab: Reflected XSS into HTML context with nothing encoded



APPRENTICE



LAB

Not solved

This lab contains a simple **reflected cross-site scripting** vulnerability in the search functionality.

To solve the lab, perform a cross-site scripting attack that calls the `alert` function.

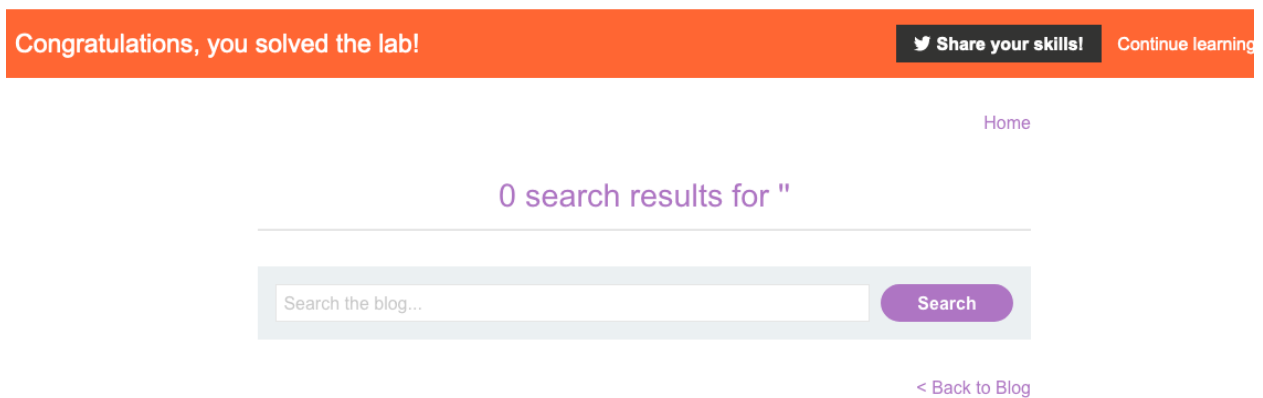
Access the lab

En este caso tenemos que hacer el llamado de alert function a través de la barra de búsquedas

Como queremos verificar si el sitio hace un manejo de etiquetas correctamente, buscamos que se ejecute cualquier sentencia, en este caso haremos uso de la función alert()



Con eso hemos resuelto el lab



Laboratorio 4

Para el siguiente lab tenemos que encontrar una vulnerabilidad en la sección de comentarios, de forma que como en el ejercicio anterior podamos ejecutar código javascript desde la sección de comentarios.

Lab: Stored XSS into HTML context with nothing encoded



APPRENTICE



LAB

Not solved

This lab contains a **stored cross-site scripting** vulnerability in the comment functionality.

To solve this lab, submit a comment that calls the `alert` function when the blog post is viewed.

Access the lab

Procedemos con el llenado del formulario y mandamos el comentario

Leave a comment

Comment:

```
<script>alert(1)</script>
```

Name:

werwerwer

Email:

erwe@ddsds.com

Website:

https://0aa400cc04d99c038161701d00c400bd.web-security-academy.net/post?postId=3

Post Comment

Y con eso hemos resuelto el laboratorio

Congratulations, you solved the lab!

 [Share your skills!](#)

[Continue learning >>](#)

[Home](#)

Thank you for your comment!

Your comment has been submitted.

[< Back to blog](#)