Ethical Hacking Internship

Sakshi Gawande

Institutional Affiliation : Internship Studio

-:Task-1:-

Task: “Solve any 5 XSS Labs on Portswigger ”

Due Date : August 24, 2022

Sakshi Gawande

Cross-site scripting

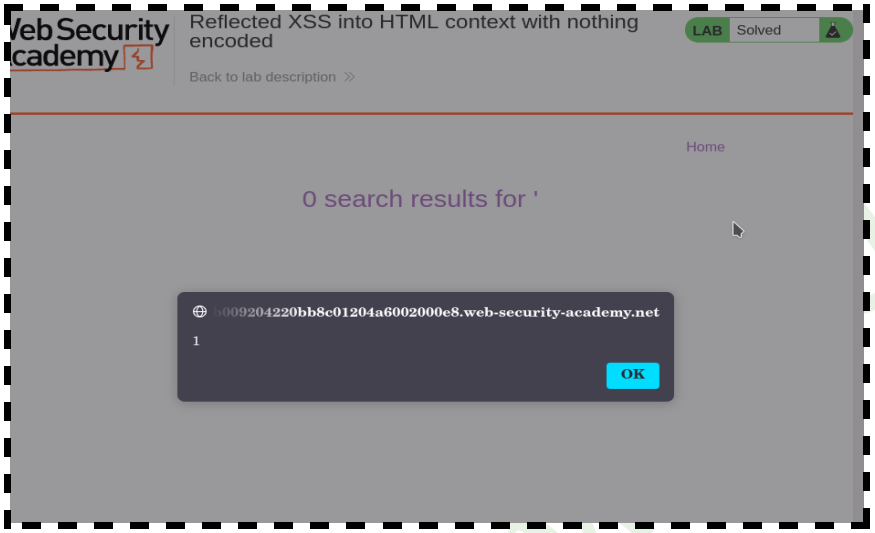
Lab 1 : Reflected XSS into HTML Context with nothing encoded

In this Lab, we will be performing a reflected XSS attack on the search function.

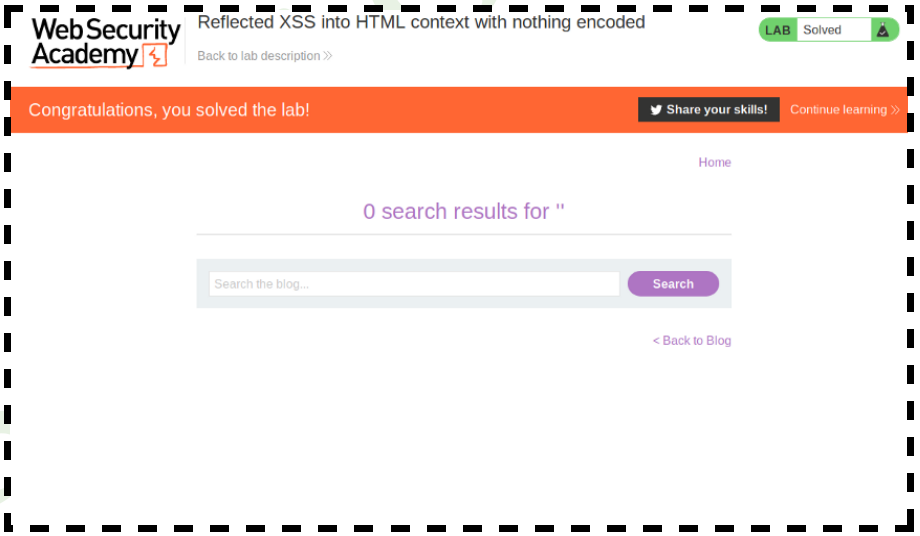
In the search bar we can input a simple alert function from javascript:

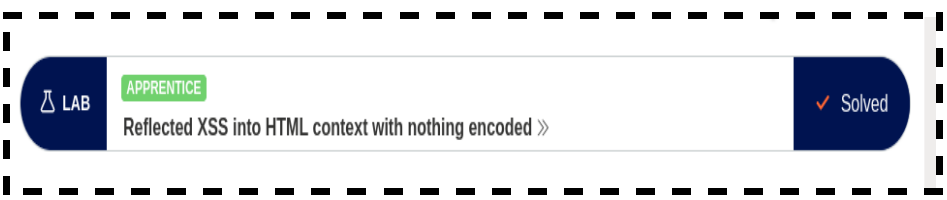
<script>alert(1)</script>

When you click on search, a pop-up alert will show your 1 message. This means your attack was successful.



We have solved this simple lab. Click ok and you will be at the result page.





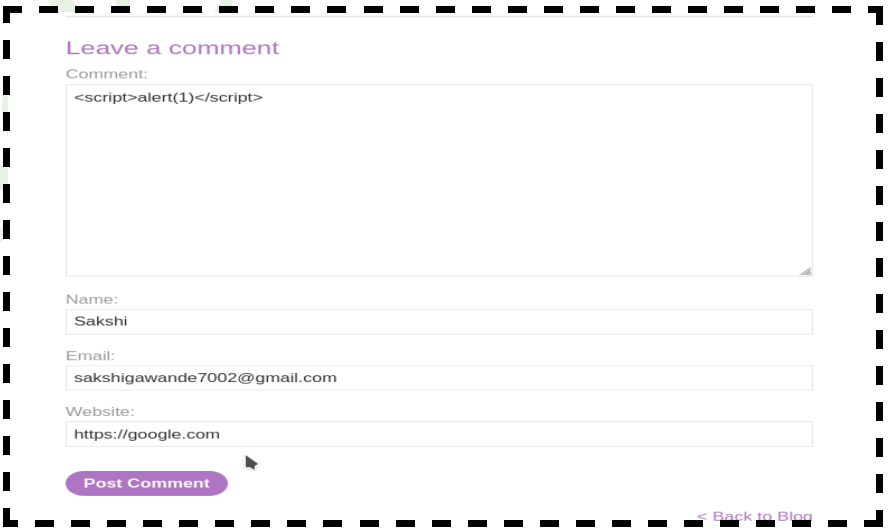
Lab 2: Stored XSS into HTML context with nothing encoded

In this lab, we will be inserting javascript code as a stored XSS attack.

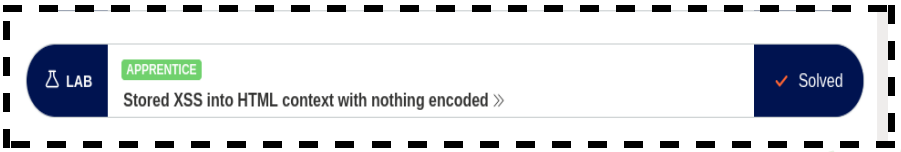
First, view one of the blog posts and scroll down to the comment section.

Here we need to post the comment with the malicious code so that it will be stored in their database. So the next time anyone that visits this page, their web browser will render this page while executing the malicious code.

Let's Post comment.

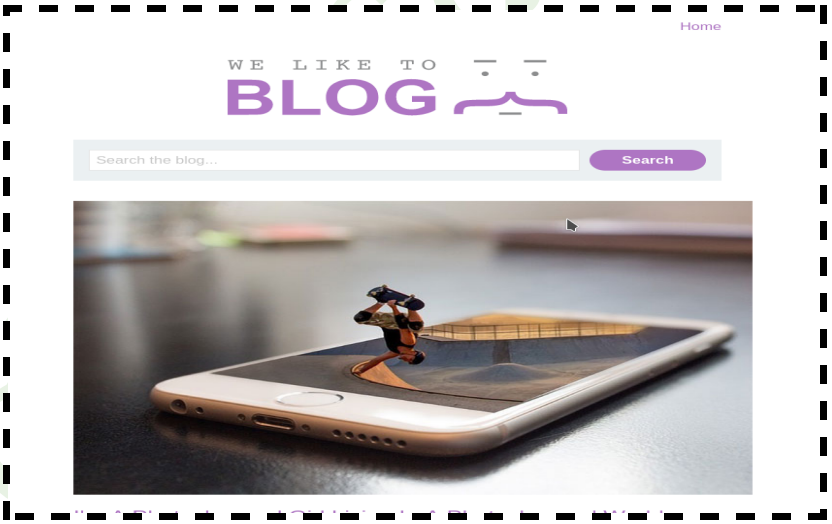


The pop-up alert with the 1 message appears. This means that you have successfully executed the attack.



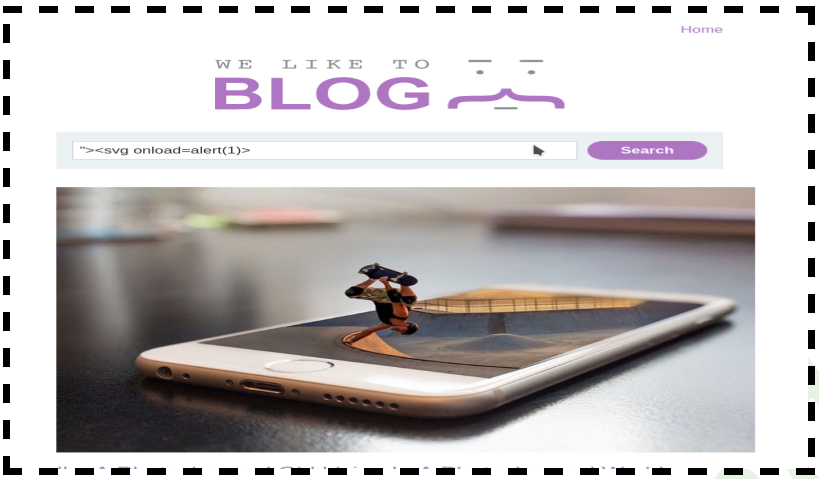
Lab 3: DOM XSS in document.write sink using source location.search

This lab contains a DOM-based cross-site scripting vulnerability in the search query tracking functionality.

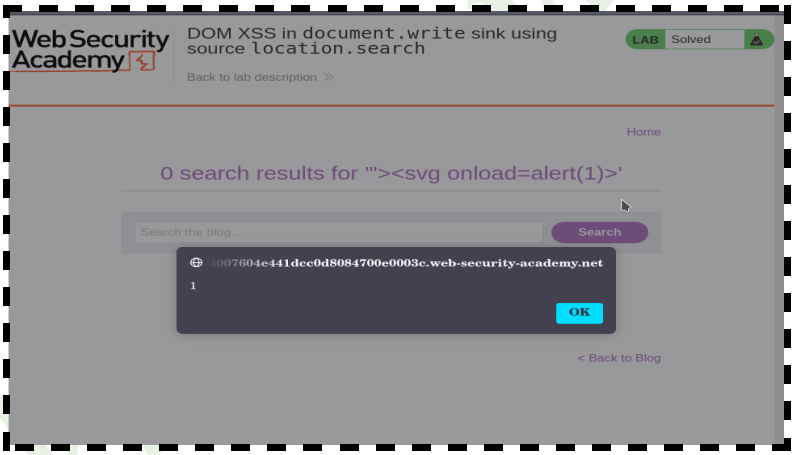


In the search bar we can input a alert function from javascript:

"><svg onload=alert(1)>



The pop-up alert with the 1 message appears. This means that you have successfully executed the attack.



Click ok and you will be at the result page.

We have solved this simple lab.

Lab 4: DOM XSS in innerHTML sink using source Location.search

This lab contains a DOM-based cross-site scripting vulnerability in the search blog functionality. It uses an innerHTML assignment, which changes the HTML contents of a div element, using data from location.search.

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

In the search bar we can input a alert function from javascript:

<img src=1 onerror=alert(1)>

Saksham

The pop-up alert with the 1 message appears. This means that you have successfully executed the attack.

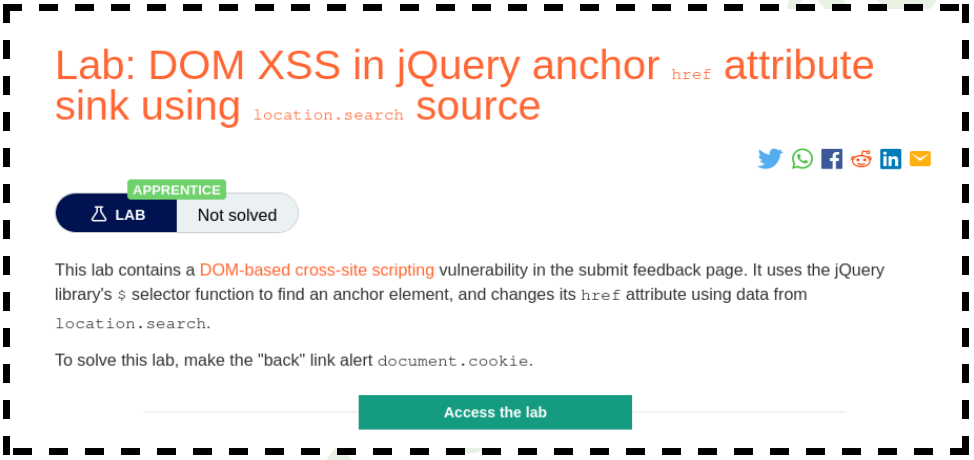
Click ok and you will be at the result page.

We have solved this simple lab.

Lab 5: DOM XSS in jQuery anchor href attribute sink using location.search source

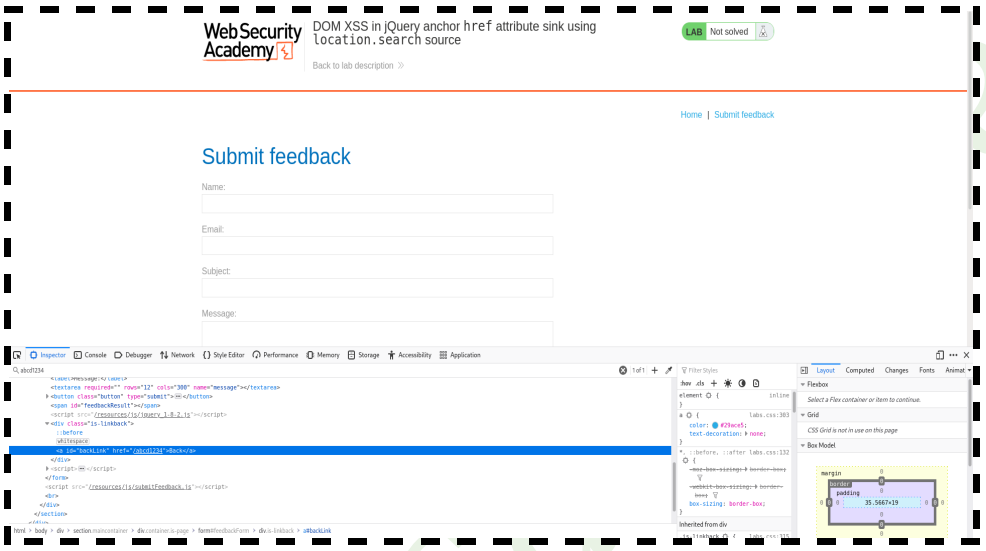
This lab contains a DOM-based cross-site scripting vulnerability in the submit feedback page. It uses the jQuery library's $ selector function to find an anchor element, and changes its href attribute using data from location.search.

To solve this lab, make the "back" link alert document.cookie.



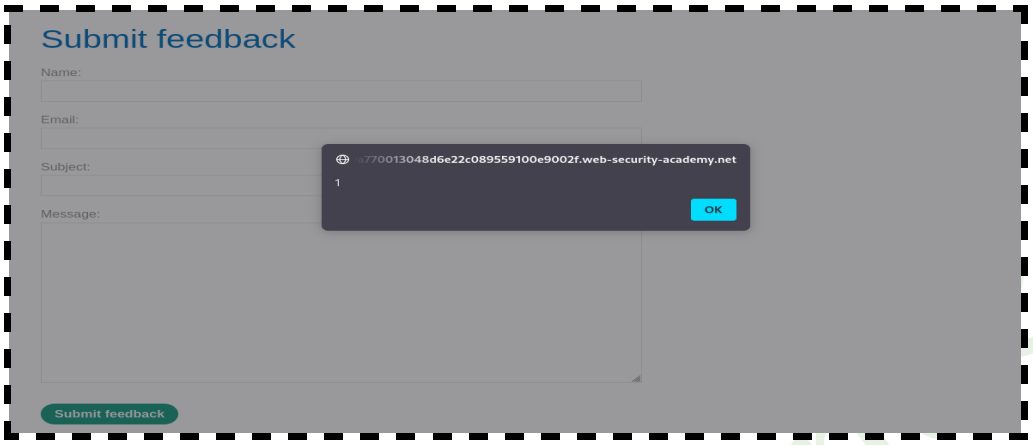
First Access the lab and then click on Submit feedbacks

Right-click and inspect the element, and observe that your random string has been placed inside an href attribute.

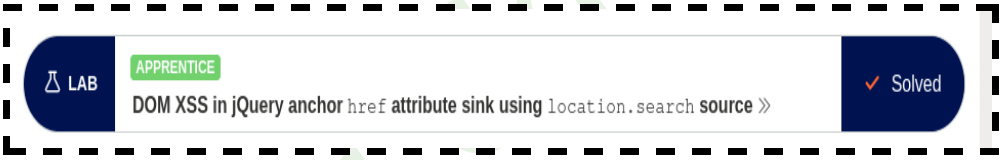


Change returnPath to: javascript:alert(1)

The pop-up alert with the 1 message appears. This means that you have successfully executed the attack.



Click ok and you will be at the result page



We have solved this simple lab.

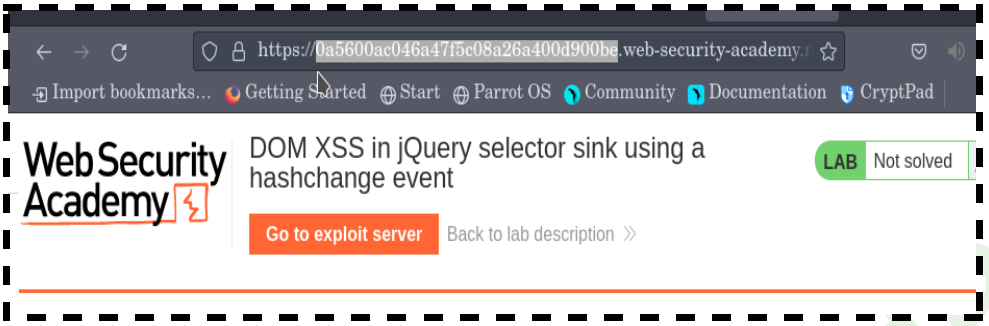
Sakshi Agnihotri

Lab 6: DOM XSS in jQuery selector sink using a hashchange event

This lab contains a DOM-based cross-site scripting vulnerability on the home page. It uses jQuery's $() selector function to auto-scroll to a given post, whose title is passed via the location.hash property.

To solve the lab, deliver an exploit to the victim that calls the print() function in their browser.

From the lab banner,open the exploit server.

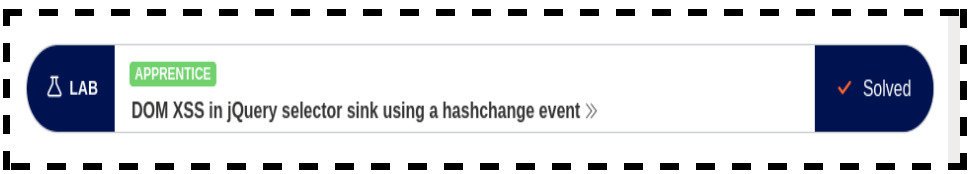


In the **Body** section, add the following malicious iframe:

<iframe src="https://YOUR-LAB-ID.web-security-academy.net/#" onload="this.src+='<img src=x onerror=print()>'"></iframe>

Store the exploit, then click **View exploit** to confirm that the print() function is called.

Go back to the exploit server and click **Deliver to the victim** to solve the lab.



We have solved this simple lab.

Solved Portswigger labs