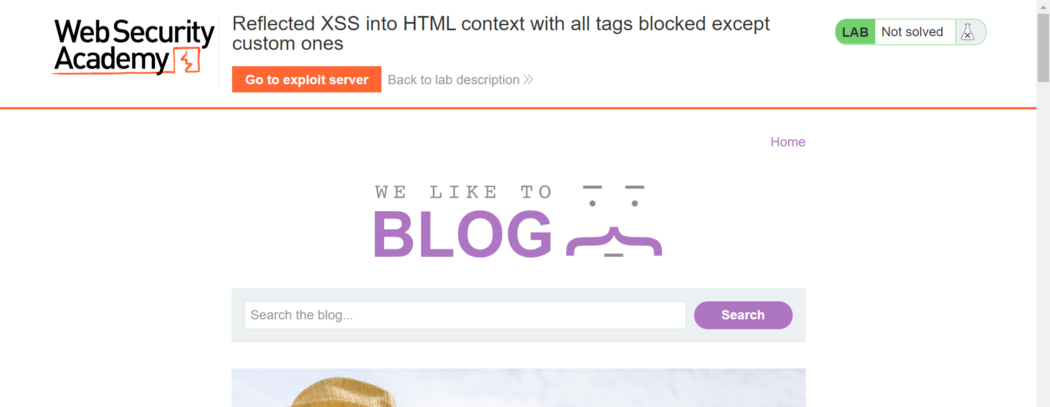
# **Reflected XSS into HTML context with all tags blocked except custom ones**

For this lab, we have to create a custom html tag since all other tags have been blocked. Below the code is given from the portswigger lab. You can solve the lab very easily by following the solution, just copy & paste the code they gave to solve it.

But for me, I wouldn’t properly understand what I just did. To really understand, we have to go through this lab bit by bit.

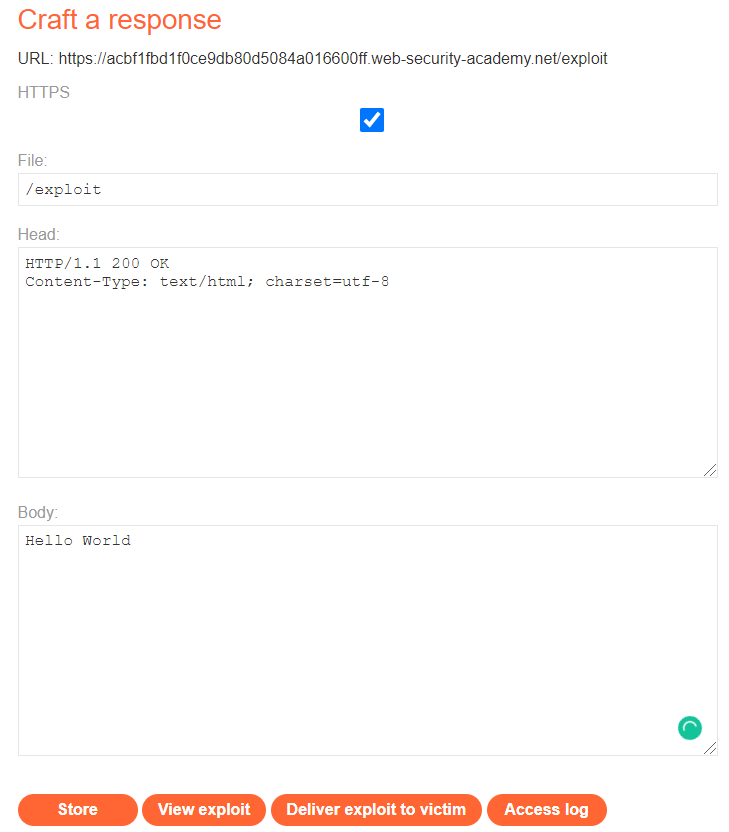
When you access the lab, you are greeted by a blog page.



Go to your exploit server by clicking the orange button on the top left of the page.

You will be redirected to an exploit server to craft your response.

Basically what this exploit server do is to craft a html file named ‘exploit ’, and when the victim click the URL link below: [https://acbf1fbd1f0ce9db80d5084a016600ff.web-security-academy.net/exploit](https://acbf1fbd1f0ce9db80d5084a016600ff.web-security-academy.net/exploit" \t "https://yikai0505.medium.com/_blank) , it will execute the exploit html file.



We can test this by clicking the view exploit button on the bottom or just copy and paste the exploit URL on your browser.

You should be re-directed to another web page showing ‘Hello World’.



Now, below is the code given in the solution page.

<script>  
location = 'https://your-lab-id.web-security-academy.net/?search=%3Cxss+id%3Dx+onfocus%3Dalert%28document.cookie%29%20tabindex=1%3E#x';  
</script>

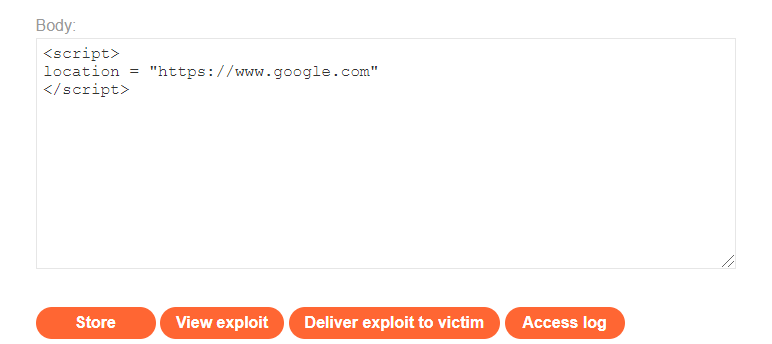
Let us understand this code better to know how this XSS attack works.

Thelocation variable inside the script tag, basically re-directs you to the website it is assigned with.

To show this, we can just insert the code:

<script>  
location = “[https://www.google.com](https://www.google.com/" \t "https://yikai0505.medium.com/_blank)"  
</script>

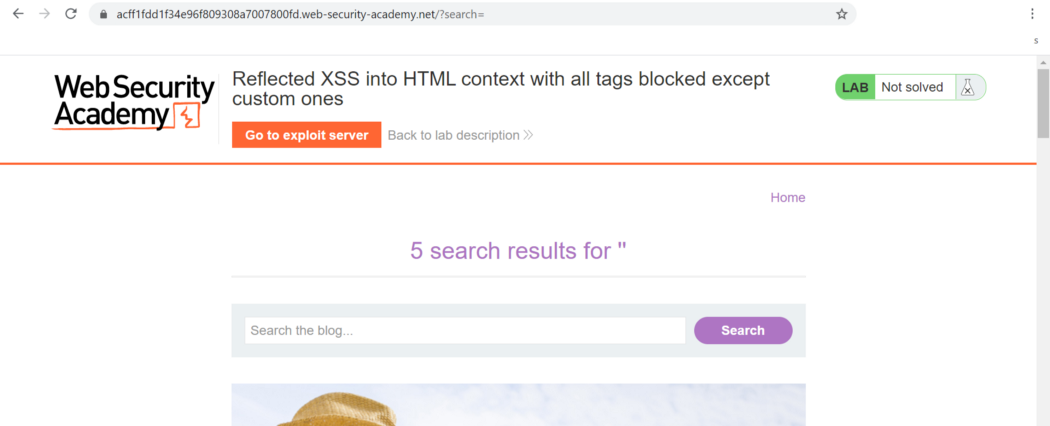
and when you click the view exploit button, it should re-direct you to the google page.



Since we know what the location variable does let’s move on to the next part of the code.

'https://your-lab-id.web-security-academy.net/?search=

When you click the search button on the blog page, you should be re-directed to a result page as shown below. Notice the url in the url bar looks similar to the code above.



This is to re-direct the victim to the results page when they click on the exploit link.

Use a [URL decoder](https://www.urldecoder.org/" \t "https://yikai0505.medium.com/_blank) to make it easier to read the statement after the ?search=.

Before URL decode:

%3Cxss+id%3Dx+onfocus%3Dalert%28document.cookie%29%20tabindex=1%3E#x

After decode:

<xss+id=x+onfocus=alert(document.cookie) tabindex=1>#x

Let’s break down this code bit by bit. You can treat the ‘+’ sign as space.

<xss This part of the code basically is creating a customized tag named ‘xss’. In this case, it is a self-closing tag.

As for this lab, most of the tags have been blocked except customized ones.

id=x This is to add an id attribute named ‘x’ to the customized <xss> tag.

onfocus=alert(document.cookie) [onfocus](https://www.w3schools.com/tags/ev_onfocus.asp" \t "https://yikai0505.medium.com/_blank) is an event attribute where it will execute the javascript code when the HTML element that it is assigned to gets focus. The javascript code:alert(document.cookie) , will be executed once the event happened.

alert(document.cookie) This is just to pop up an alert showing the cookie on the particular website.

tabindex=1 [Tabindex](https://www.w3schools.com/tags/att_global_tabindex.asp" \t "https://yikai0505.medium.com/_blank) attribute basically just specify the tab order of the HTML element when you use the ‘tab’ button to navigate.

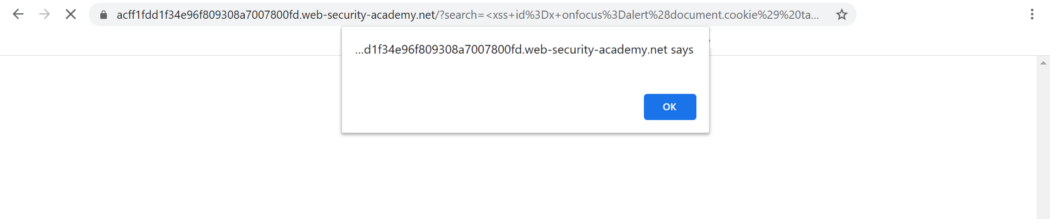
#x When you enter this code in the browser bar, it will focus on the html element which has the id of x, in this case, the cutomized xss was assigned the id=x.

In short, when the malicious link below is run:

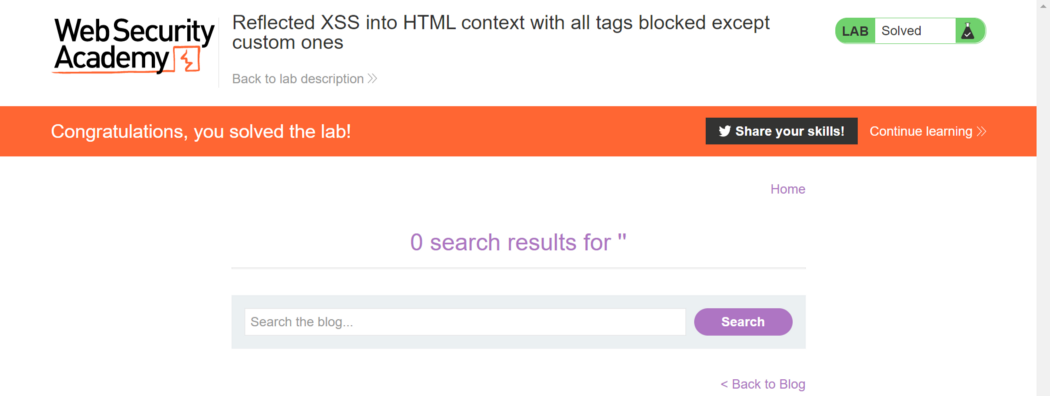
[https://your-lab-id.web-security-academy.net/?search=%3Cxss+id%3Dx+onfocus%3Dalert%28document.cookie%29%20tabindex=1%3E#x](https://your-lab-id.web-security-academy.net/?search=<xss+id=x+onfocus=alert(document.cookie) tabindex=1>" \l "x" \t "https://yikai0505.medium.com/_blank)

The code #x will cause the website to focus on the id ‘x’ which will trigger the event onfocus=alert(document.cookie) . This will execute the javascript alert(document.cookie) and pop-up a window showing the cookie of the webpage. Whenever you click on the URL bar and hit the tab button on your keyboard, the tabindex=1 will cause you to focus on the xss element again, thus triggering the onfocus event and execute the javascript again.

So after understanding how the code worked, you can click store and deliver exploit to solve this lab. By click on the view exploit, you should see an alert box as shown.



Once you clicked the ok button, it should redirect you to the results page.



Just to illustrate how some of the attributes works, I have created a simple webpage.

Below is a zoom-out view of the webpage, with 2 sections or 2 [divs](https://www.w3schools.com/tags/tag_div.ASP" \l ":~:text=The tag defines a,the class or id attribute." \t "https://yikai0505.medium.com/_blank). The first div is outlined in red, and the second div is outlined in blue. The first div also contains an input box.

Div1 is assigned with id=div1 , Div2 is assigned with id=div2 .



First, I want to demonstrate how to navigate to a particular section or div of a webpage.

By zooming back to 100%, we can only view the first section of the webpage.



The URL currently will look something like: http://<someAddress>/index.html

But by adding #div2 at the end of the URL and hitting enter: http://<someAddress>/index.html#div2

I am brought to the 2nd section.



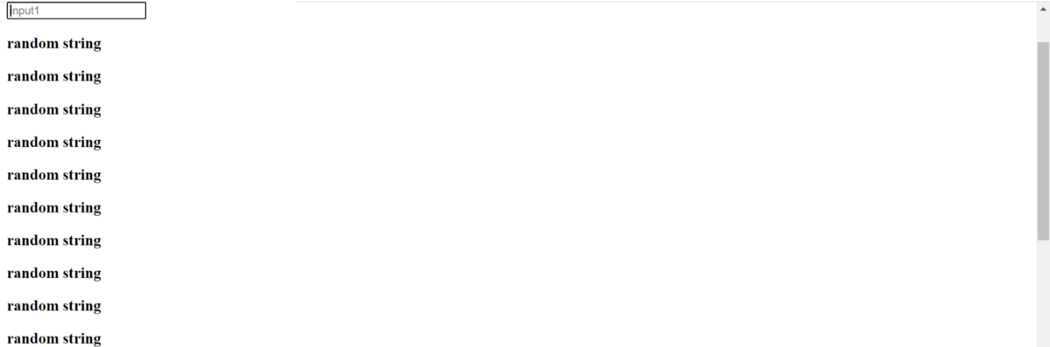
This is just to demonstrate one use of the id attribute.

Input 1 has an id assigned to it.

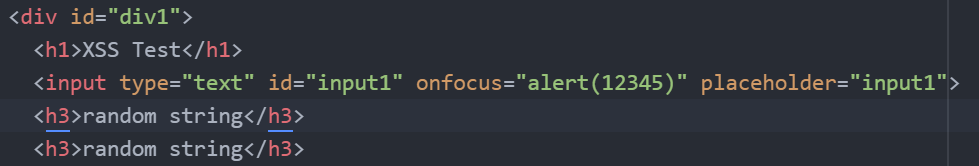
input 1 →id = input1

so when I enter the URL: <someAddress>/index.html#input1

I got brought to the input1 box, noticed that that is a black vertical line in input 1 box. This means it has been focused on that box.

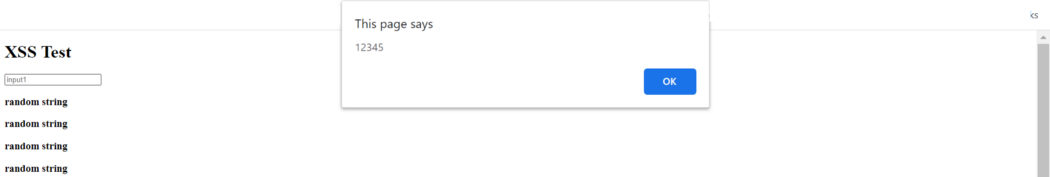


Now, let’s see how we can use onfocus attribute to pop up an alert box when the input 1 box gets focused.



Inside the input element, I added on a code: onfocus="alert(12345)"

So this should pop up something when I click on the input1 box on my own, or when I enter the URL: <someAddress>/index.html#input1 or we can use the tab button to navigate till it focuses on input1 box.



Now, if I click anywhere on the webpage and enter the tab button, I will get the alert again.