Sri Lanka Institute of Information technology

2025

Web Security-IE2062

Practical

Year 2, Semester 2

IT23400368

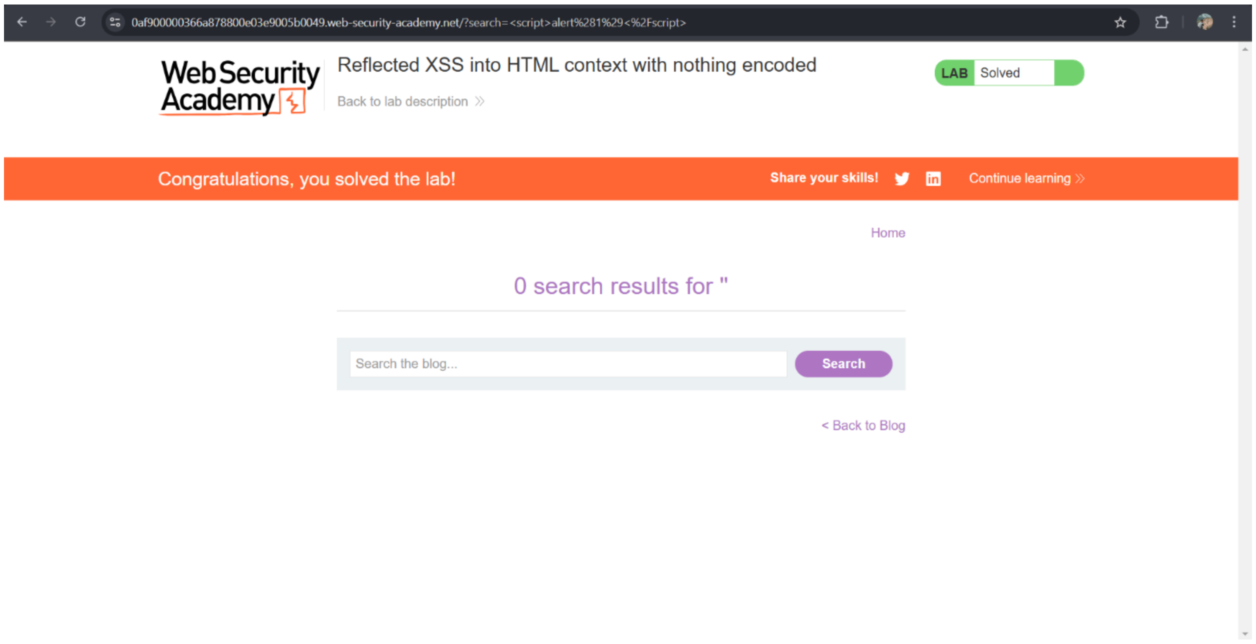
MMM ARKAM

Y2.S2.WD.CS.01.01

MALABE CAMPUS

Lab1:

This lab has a reflected XSS vulnerability in the search box. When **<script>alert(1)</script>** is entered and **"Search"** is clicked, the website shows the input without blocking the script. This causes the browser to run the script and display an alert box, proving the XSS works.



0af900000366a878800e03e9005b0049.web-security-academy.net/?search=<script>alert%281%29<%2Fscript>

Reflected XSS into HTML context with nothing encoded

Back to lab description

LAB Solved

Congratulations, you solved the lab!

Share your skills!

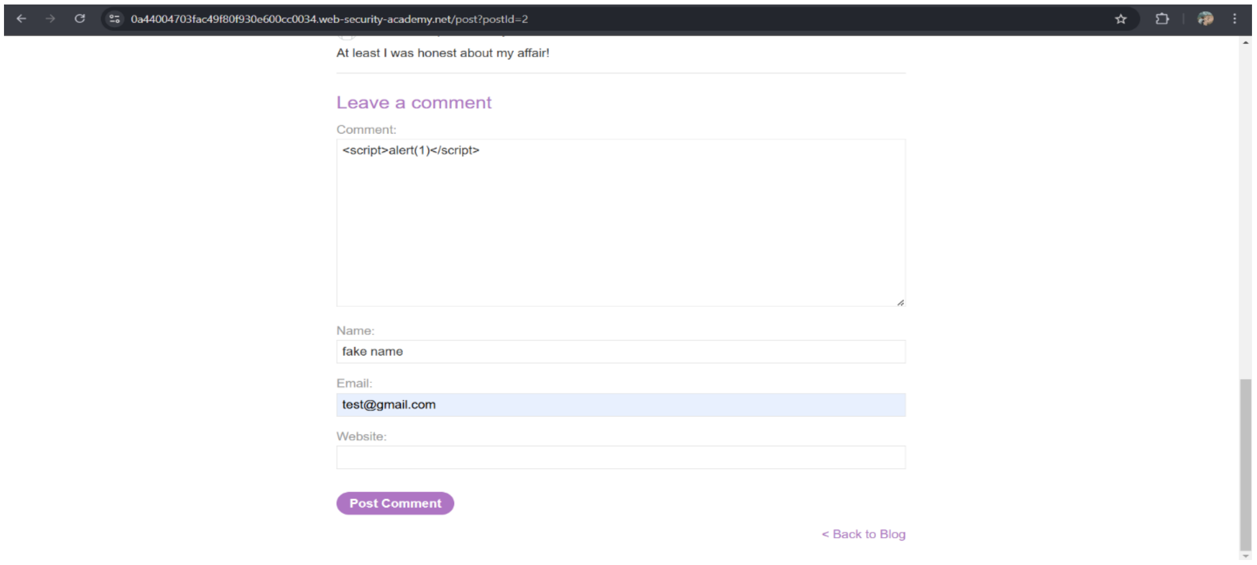
Continue learning

Home

0 search results for ''

Lab 2:

This lab has a stored XSS vulnerability in the comment section. When **<script>alert(1)</script>** is submitted as a comment, the website saves it without blocking the script. When someone views the blog post, the script runs, showing an alert box, proving the stored XSS works.



0a44004703fac49f80f930e600cc0034.web-security-academy.net/post?postId=2

At least I was honest about my affair!

Leave a comment

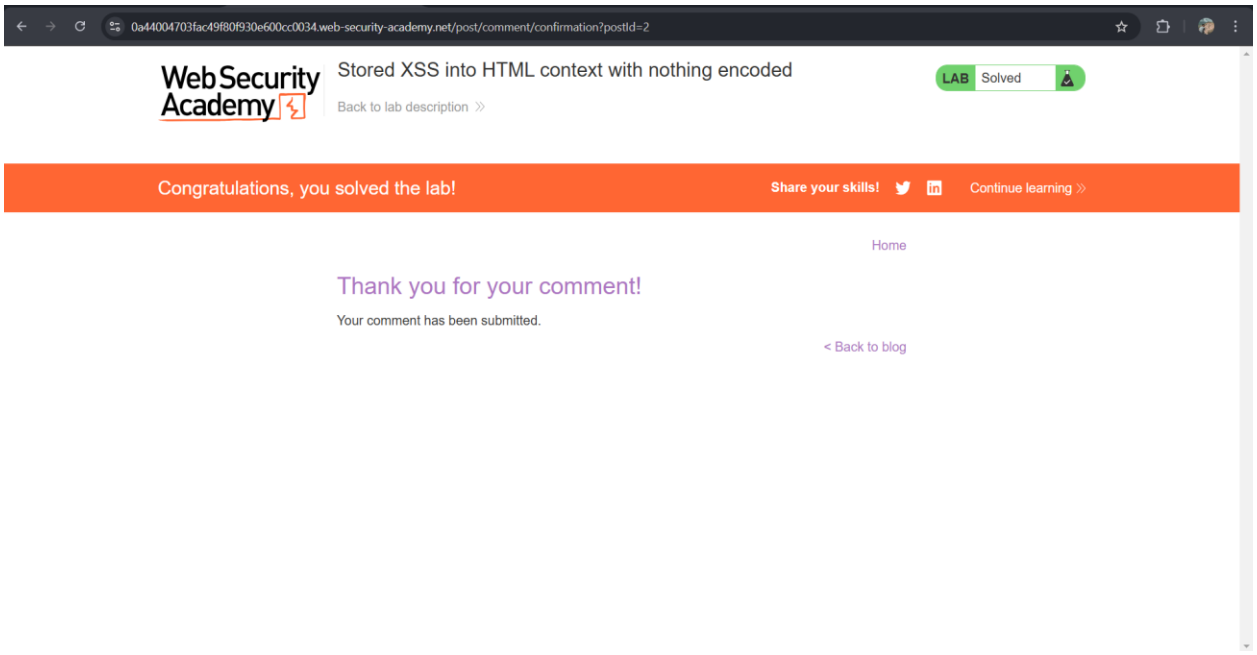
Comment:

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

Name: fake name

Email: test@gmail.com

Website:



0a44004703fac49f80f930e600cc0034.web-security-academy.net/post/comment/confirmation?postId=2

Stored XSS into HTML context with nothing encoded

LAB Solved

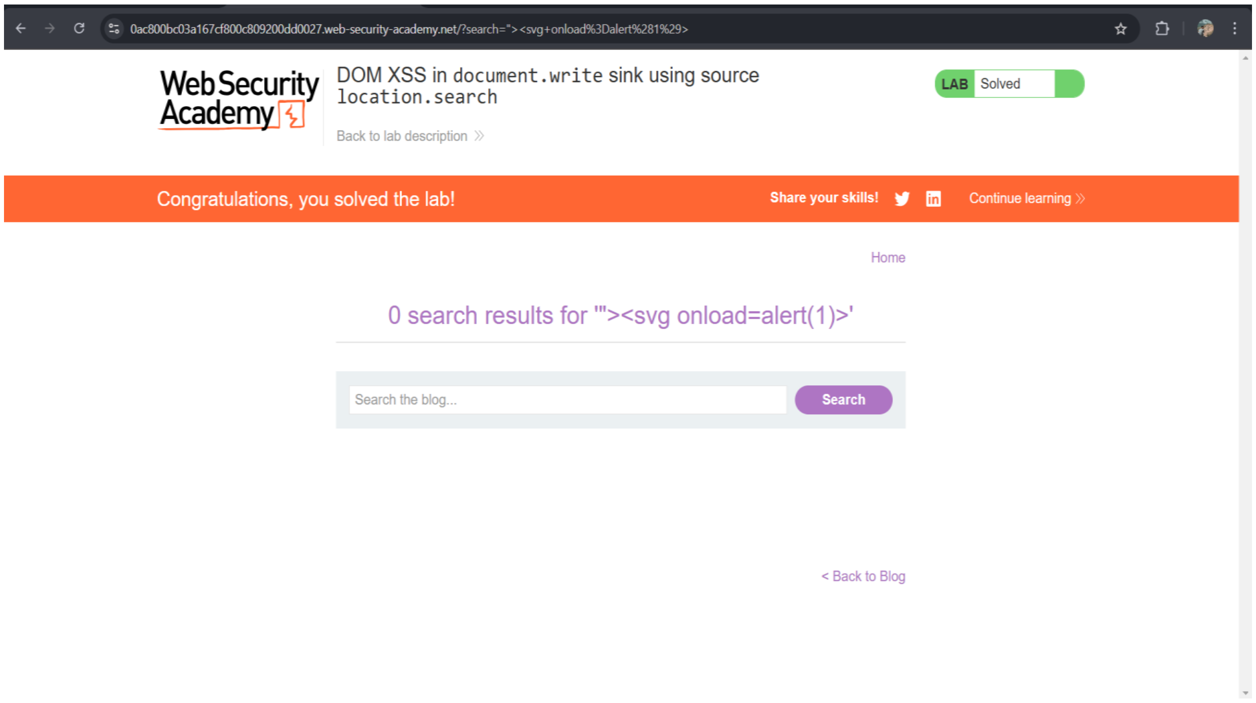
Congratulations, you solved the lab!

Thank you for your comment!

Your comment has been submitted.

Lab 3:

This lab has a DOM-based XSS vulnerability in the search function. The website takes input from the URL and inserts it into an **img** tag using **document.write**. By searching for **"><svg onload=alert(1)>,** the **img** attribute is broken, and a new <**svg**> element with an onload event is injected. This makes the browser execute **alert(1)**, proving the XSS works.



0ac800bc03a167cf800c809200dd0027.web-security-academy.net/?search="><svg+onload%3Dalert%281%29>

DOM XSS in `document.write` sink using source `location.search`

Back to lab description »

LAB Solved

Congratulations, you solved the lab!

Share your skills! Continue learning »

Home

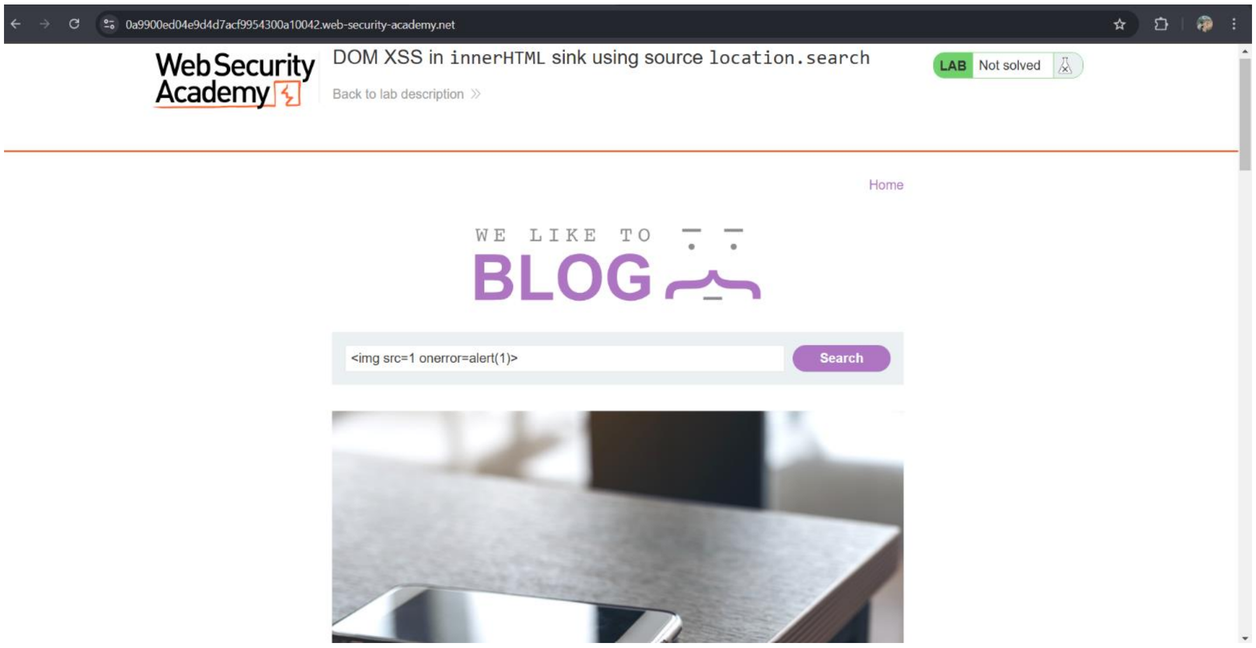
0 search results for '"><svg onload=alert(1)>'

Search the blog... Search

< Back to Blog

Lab 4:

This lab has a DOM-based XSS vulnerability in the search function. The website takes input from the URL and inserts it into a div using **innerHTML**. By searching for
<img src=1 onerror=alert(1)>,
an invalid image **(src=1)** is injected, causing an error. The onerror event is triggered, executing **alert(1),** proving the XSS works.



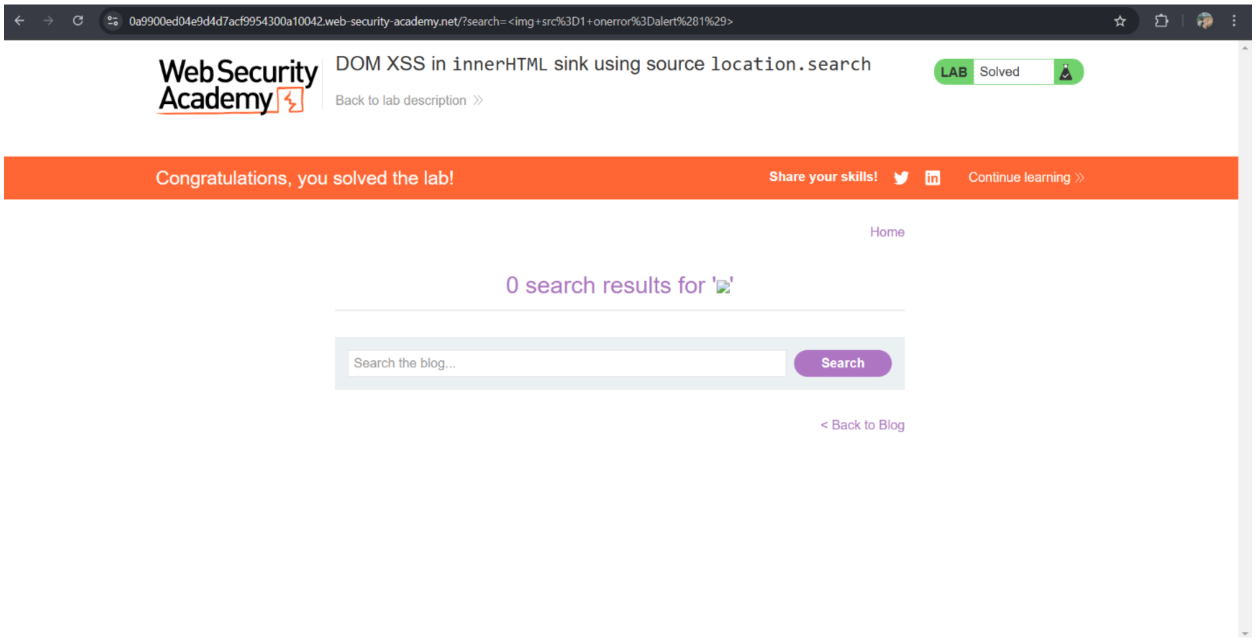
0a9900ed04e9d4d7acf9954300a10042.web-security-academy.net

DOM XSS in innerHTML sink using source location.search

Back to lab description

LAB Not solved

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



0a9900ed04e9d4d7acf9954300a10042.web-security-academy.net/?search=<img+src%3D1+onerror%3Dalert%281%29>

DOM XSS in innerHTML sink using source location.search

Back to lab description

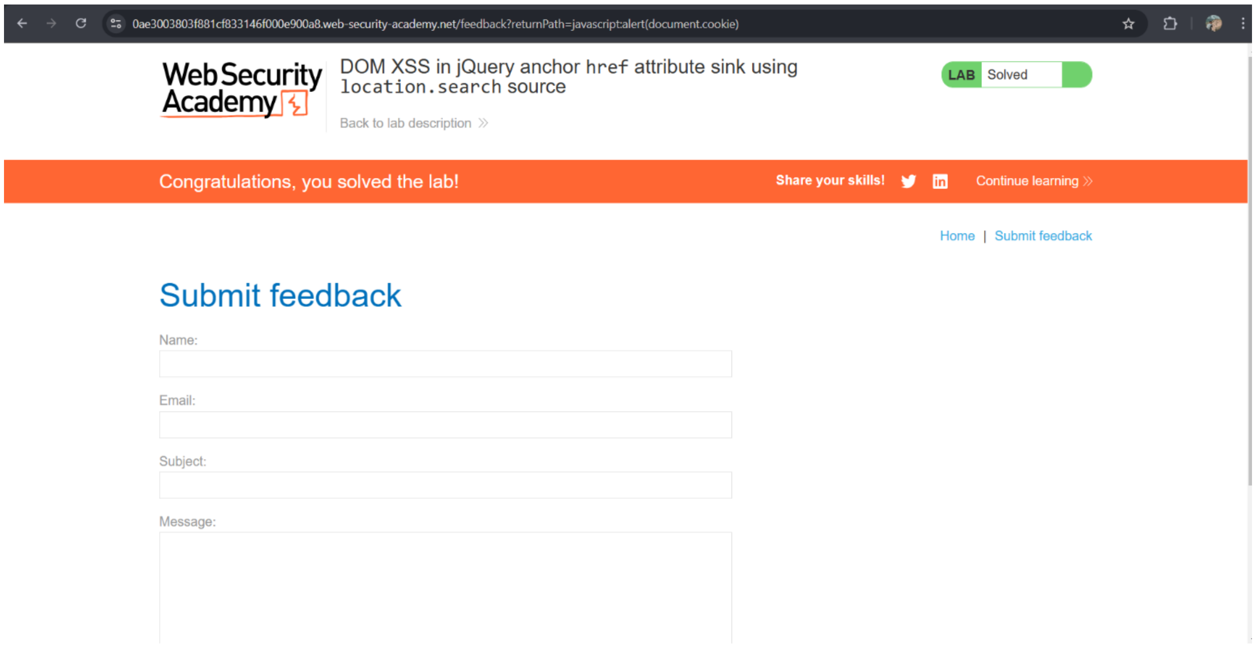
LAB Solved

Congratulations, you solved the lab!

0 search results for ''

Lab 5:

This lab has a DOM-based XSS vulnerability in the "Submit feedback" page. The website uses jQuery to update the **"back"** link's href using input from the URL. By changing **returnPath** to **javascript:alert(document.cookie)**, the **"back"** link becomes a JavaScript URL. When clicked, the browser runs **alert(document.cookie).**



0ae3003803f881cf833146f000e900a8.web-security-academy.net/feedback?returnPath=javascript:alert(document.cookie)

DOM XSS in jQuery anchor `href` attribute sink using `location.search` source

Back to lab description »

LAB Solved

Congratulations, you solved the lab!

Share your skills! Continue learning »

Home | Submit feedback

Submit feedback

Name:

Email:

Subject:

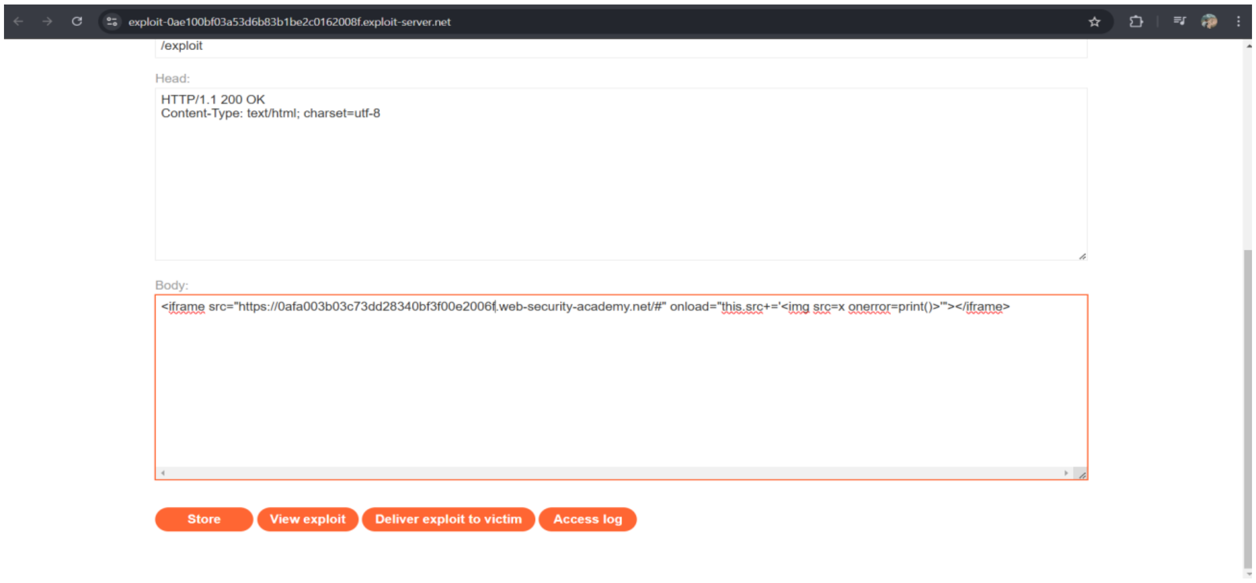
Message:

Lab 6:

This lab has a DOM-based XSS vulnerability on the home page, where the website uses **location.hash** to scroll to a post. A malicious **iframe** is created with the following code:

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

The onload event triggers an error in the image, calling the **print()** function in the victim's browser. After delivering the exploit, the **print()** function is executed, solving the lab.



exploit-0ae100bf03a53d6b83b1be2c0162008f.exploit-server.net

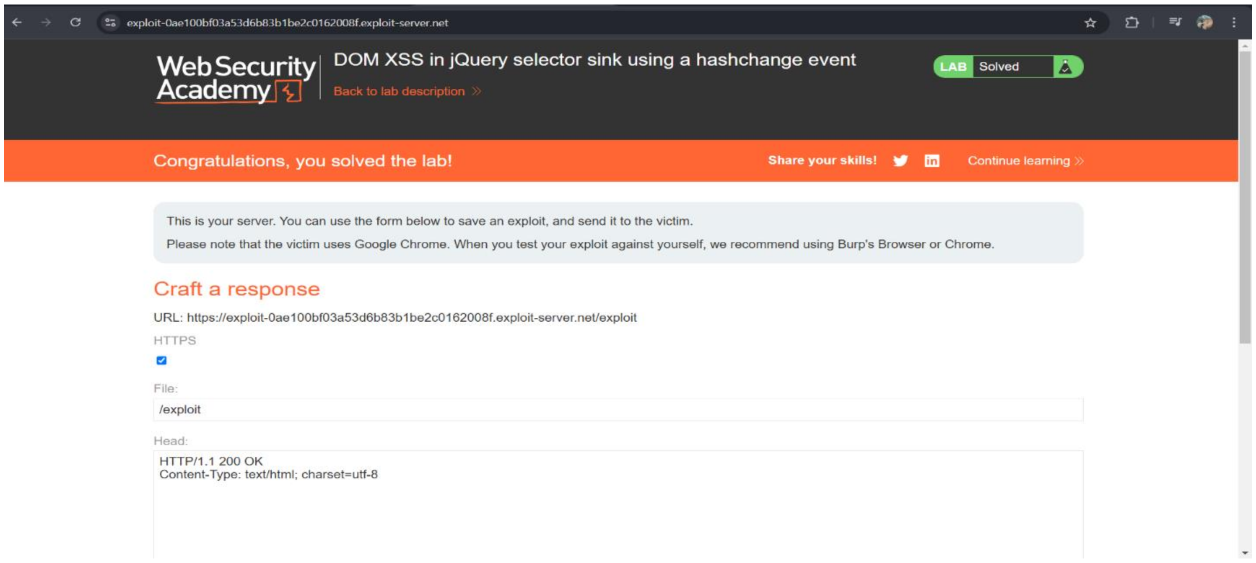
/exploit

Head:

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
```

Body:

```
<iframe src="https://0afa003b03c73dd28340bf3f00e2006f.web-security-academy.net/#" onload="this.src+='<img src=x onerror=print()>'"></iframe>
```



exploit-0ae100bf03a53d6b83b1be2c0162008f.exploit-server.net

WebSecurity Academy
DOM XSS in jQuery selector sink using a hashchange event
Back to lab description
LAB Solved
Congratulations, you solved the lab!
Share your skills!
Continue learning

This is your server. You can use the form below to save an exploit, and send it to the victim.

Please note that the victim uses Google Chrome. When you test your exploit against yourself, we recommend using Burp's Browser or Chrome.

Craft a response

URL: https://exploit-0ae100bf03a53d6b83b1be2c0162008f.exploit-server.net/exploit

HTTPS ☑

File:

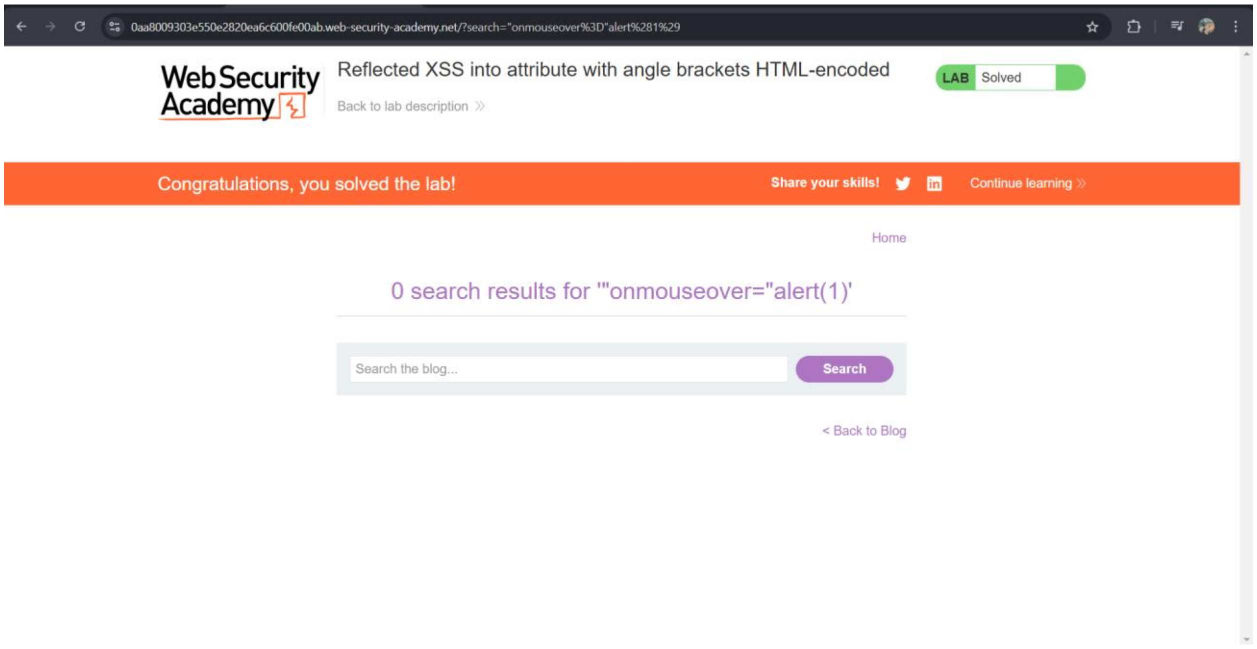
/exploit

Head:

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
```

Lab 7:

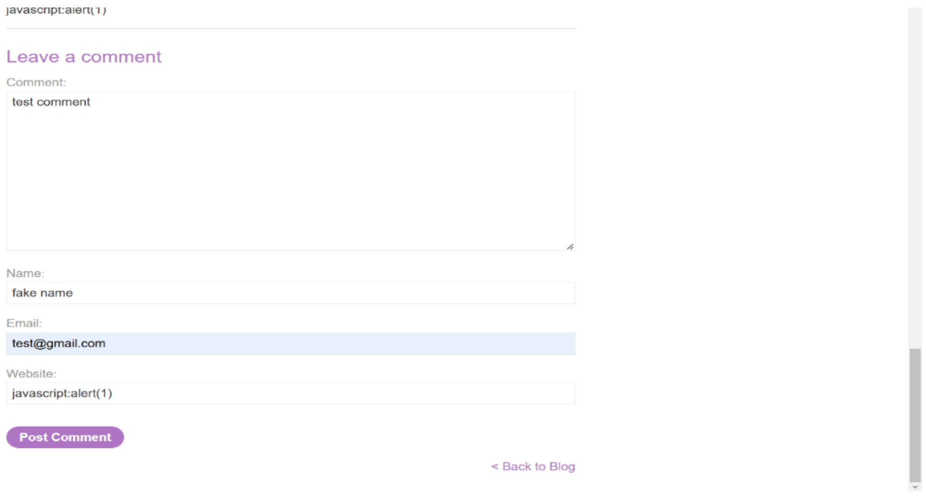
This lab has a reflected XSS vulnerability in the search function, where input is HTML-encoded. A random string is submitted in the search box, and the string is reflected inside a quoted attribute. To inject the alert, the input is replaced with:
"onmouseover="alert(1).
This breaks out of the attribute and injects an **onmouseover** event handler. Hovering over the element triggers an alert, proving the XSS works.


0aa8009303e550e2820ea6c600fe00ab.web-security-academy.net/?search="onmouseover%3D"alert%281%29
Web Security Academy
Reflected XSS into attribute with angle brackets HTML-encoded
Back to lab description
LAB Solved
Congratulations, you solved the lab!
Share your skills!
Continue learning
Home
0 search results for '"onmouseover="alert(1)'
Search the blog...
Search
< Back to Blog

Lab 8:

This lab has a stored XSS vulnerability in the comment section. To exploit it, a comment is posted with a random string in the **"Website"** input. When viewing the post, the string gets reflected inside an anchor **href** attribute.

To trigger the alert, the input is replaced with the following payload.
javascript:alert(1).
This injects a JavaScript URL that calls **alert(1).** By clicking the name above the comment, the alert is triggered, proving the XSS works.



javascript:alert(1)

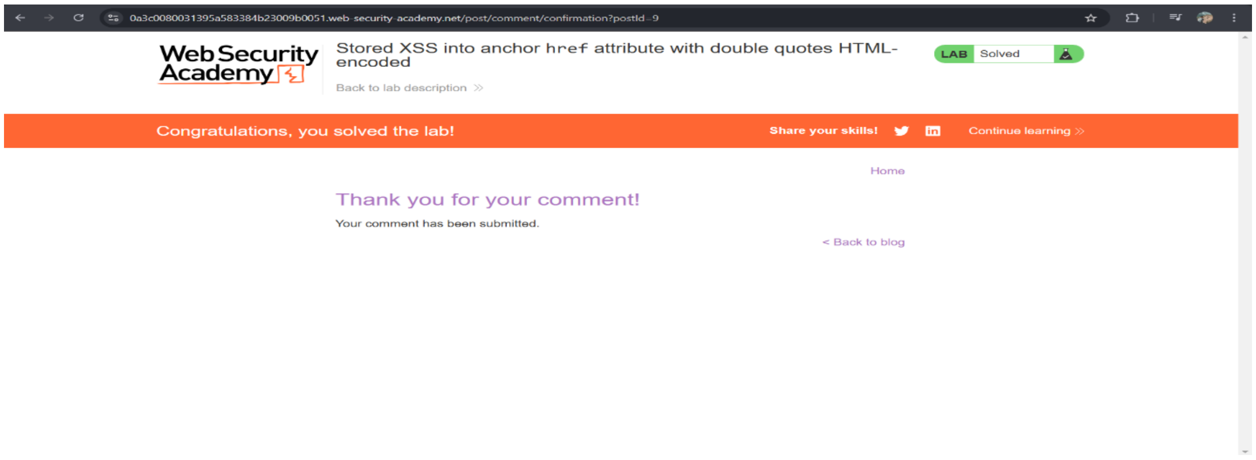
Leave a comment

Comment:
test comment

Name:
fake name

Email:
test@gmail.com

Website:
javascript:alert(1)



0a3c0080031395a583384b23009b0051.web-security-academy.net/post/comment/confirmation?postId=9

Stored XSS into anchor href attribute with double quotes HTML-encoded

LAB Solved

Congratulations, you solved the lab!

Thank you for your comment!

Your comment has been submitted.

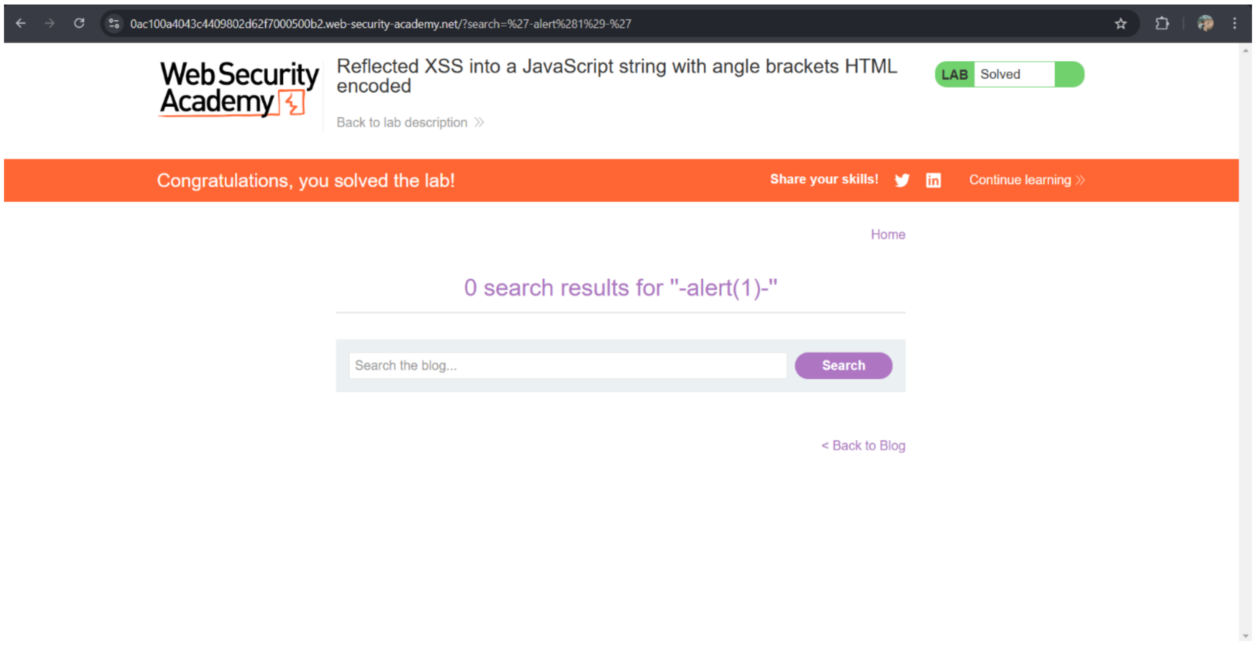
Lab 9:

This lab has a reflected XSS vulnerability where the search query is reflected inside a JavaScript string. To exploit this, a random string is submitted in the search box, and it is reflected inside the JavaScript code.

To break out of the JavaScript string and trigger an alert, the input is replaced with:

'-alert(1)-'

This breaks the string and injects the **alert(1)** function. When the page is loaded, it triggers the alert, proving the XSS works.



0ac100a4043c4409802d62f7000500b2.web-security-academy.net/?search=%27-alert%281%29-%27

Reflected XSS into a JavaScript string with angle brackets HTML encoded

Back to lab description

LAB Solved

Congratulations, you solved the lab!

0 search results for "-alert(1)-"