Sri Lanka Institute of Information Technology



B.Sc. (Hons) in Information Technology

Specializing in Cyber Security

Year 2, Semester 2

**IE2062: Web Security**

Week 1 Submission – PortSwigger Labs - XSS

IT23714120

A.M.M.G.K.P. Athawuda

Contents

[1 Introduction 3](#_Toc205756584)

[2 Lab 1: Reflected XSS into HTML context with nothing encoded 3](#_Toc205756585)

[3 Lab 2: Stored XSS into HTML context with nothing encoded 6](#_Toc205756586)

[4 Lab 3: DOM XSS in document.write sink using source location.search 9](#_Toc205756587)

[5 Lab 4: DOM XSS in innerHTML sink using source location.search 12](#_Toc205756588)

[6 Lab 5: DOM XSS in jQuery anchor href attribute sink using location.search source 14](#_Toc205756589)

[7 Lab 6: DOM XSS in jQuery selector sink using a hashchange event 15](#_Toc205756590)

[8 Lab 7: Reflected XSS into attribute with angle brackets HTML-encoded 17](#_Toc205756591)

[9 Lab 8: Stored XSS into anchor *href* attribute with double quotes HTML-encoded 18](#_Toc205756592)

[10 Lab 9: Reflected XSS into a JavaScript string with angle brackets HTML encoded 20](#_Toc205756593)

# Introduction

**XSS** is a Cross-Site Scripting, which is a client-side code injection attack where malicious JavaScript is injected into a web page viewed by other users.

Main types of XSS are:

1. Reflected XSS

Script is injected into a request and reflected immediately in the response.

1. Stored XSS

Script is stored on the server and served to multiple users.

1. DOM- based XSS

The injection occurs via client-side JavaScript without server-side modification.

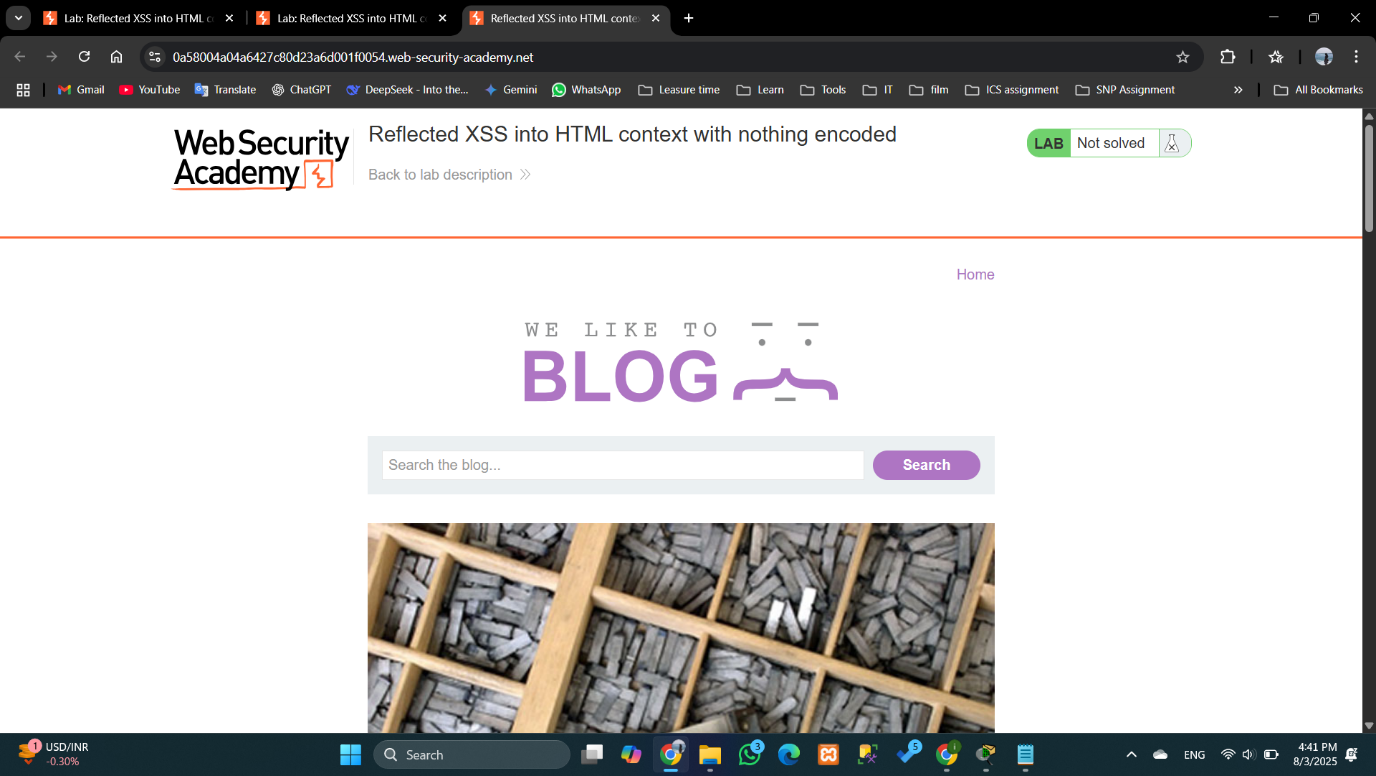
# Lab 1: Reflected XSS into HTML context with nothing encoded

A screenshot of a computer

AI-generated content may be incorrect.

The instruction for the lab says that there is a vulnerability in search functionality. And to solve the lab it must perform a cross-site scripting attack that calls the ‘alert’ function.

**Solution:**

1. **Click Access the lab and go to the new site provided. **
2. In the search box or in the URL type something to how that text appeared.

A white and black striped object

AI-generated content may be incorrect.

A screenshot of a search results

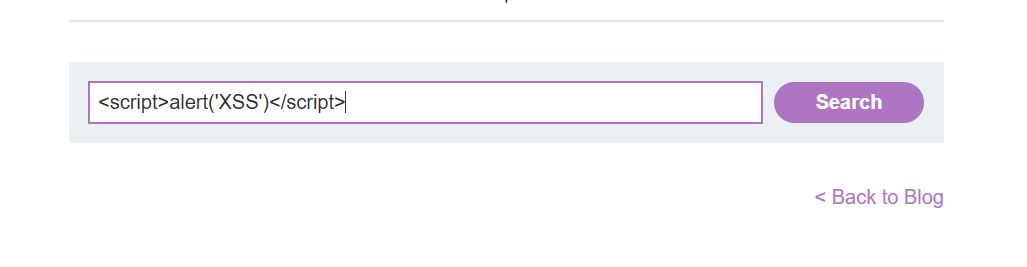
AI-generated content may be incorrect.

1. Type various kinds of inputs in the search box to see what happened. When type a html tag or js code it rendered the output.

A white and purple line

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. When JavaScript code entered an alert box popped up. It executes Js input without any encoding.

* It demonstrates that the website is vulnerable to reflected XSS attacks as it

A close-up of a computer screen

AI-generated content may be incorrect.

# Lab 2: Stored XSS into HTML context with nothing encoded

**Instruction**:

A screenshot of a web page

AI-generated content may be incorrect.

**Solution**:

1. There are some blogs on the site and to go to comment section by clicking ‘view post’.

A child holding a stuffed animal

AI-generated content may be incorrect.

A screenshot of a chat

AI-generated content may be incorrect.

1. Then post a comment in plain text and use HTML to see what happened.

A screenshot of a chat

AI-generated content may be incorrect.

A close-up of a white background

AI-generated content may be incorrect.

A close up of a date

AI-generated content may be incorrect.

Html:

A screenshot of a email

AI-generated content may be incorrect.

A screenshot of a calendar

AI-generated content may be incorrect.

1. Enter a JavaScript payload:

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Alert executes. It means the site is vulnerable to all users who view the comment, and the output is not encoded when displaying comments.



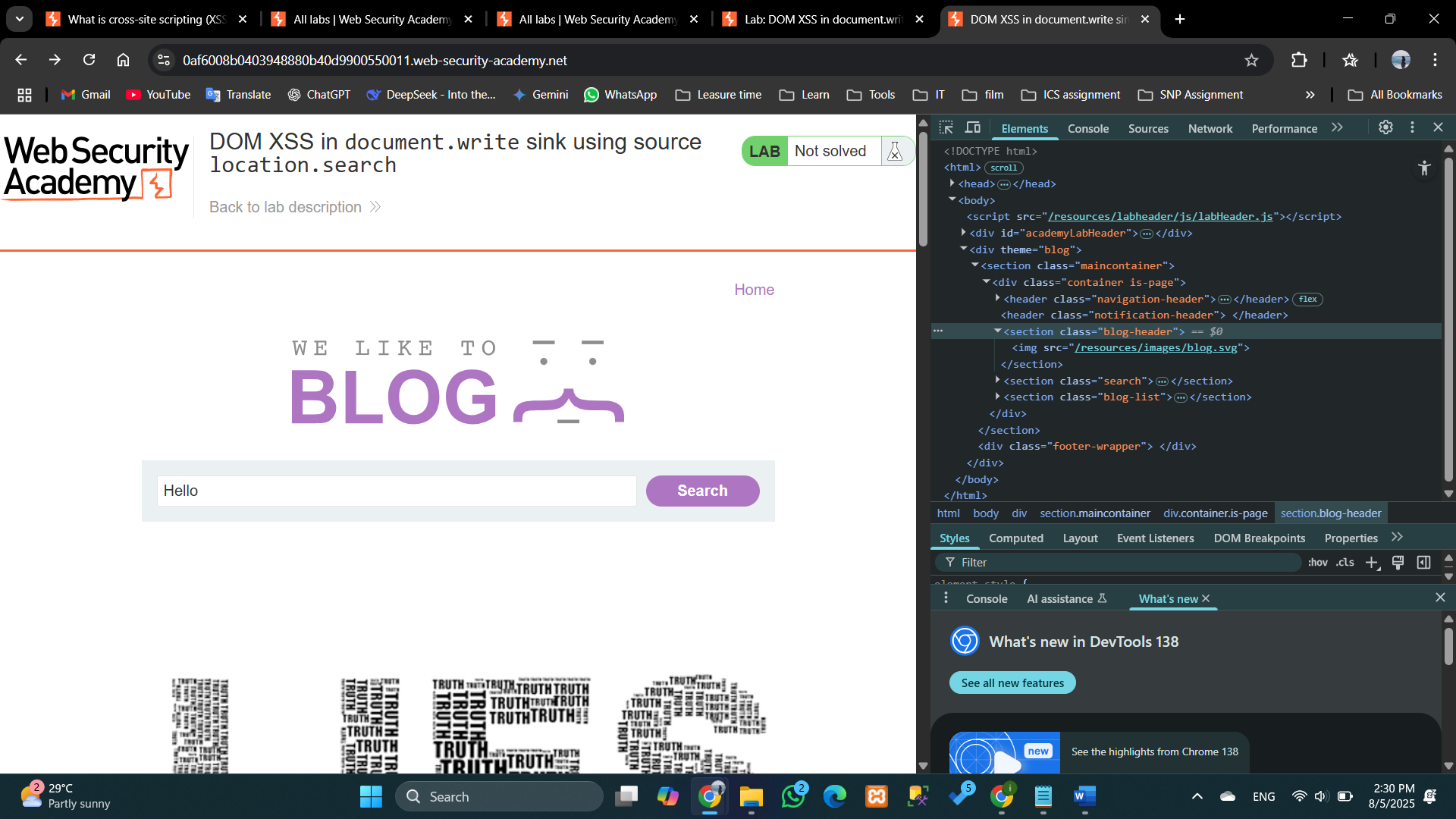
# Lab 3: DOM XSS in document.write sink using source location.search

**Instruction**:



**Solution**:

1. Access the lab and open the ‘inspect’ the page.
2. Enter a test string like ‘Hello’ and see what happens in the code.



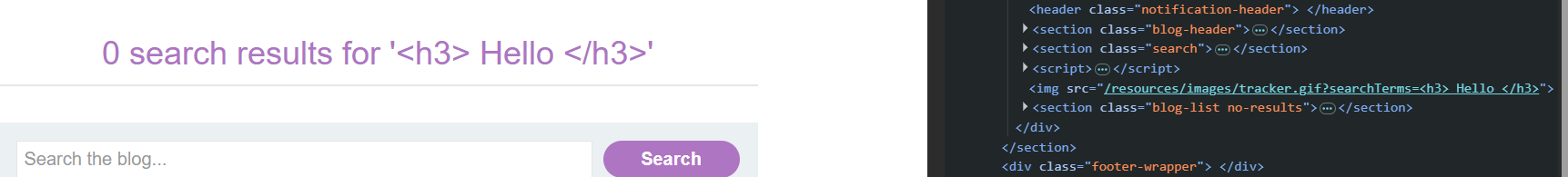
A screenshot of a computer screen

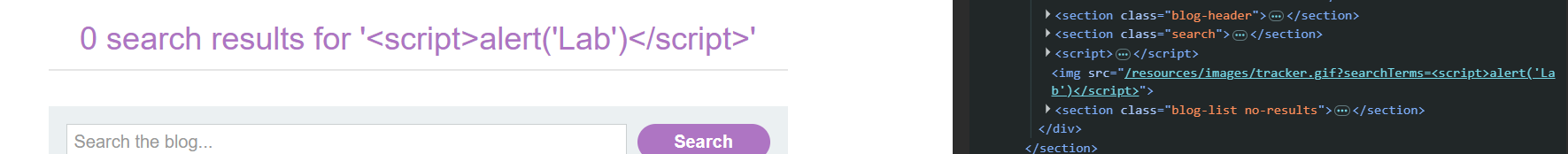
AI-generated content may be incorrect.

1. The string that entered appeared in ‘src’ attribute ‘<img>’ tag.

A screen shot of a computer

AI-generated content may be incorrect.





1. This indicates the application dynamically writes search terms into the DOM using ‘*document*.*write’*.
2. To break down that html attribute and enter JavaScript enter the below payload:

( “ closes the src attribute and > closes img tagged )

A purple line on a white background

AI-generated content may be incorrect.

A screenshot of a computer

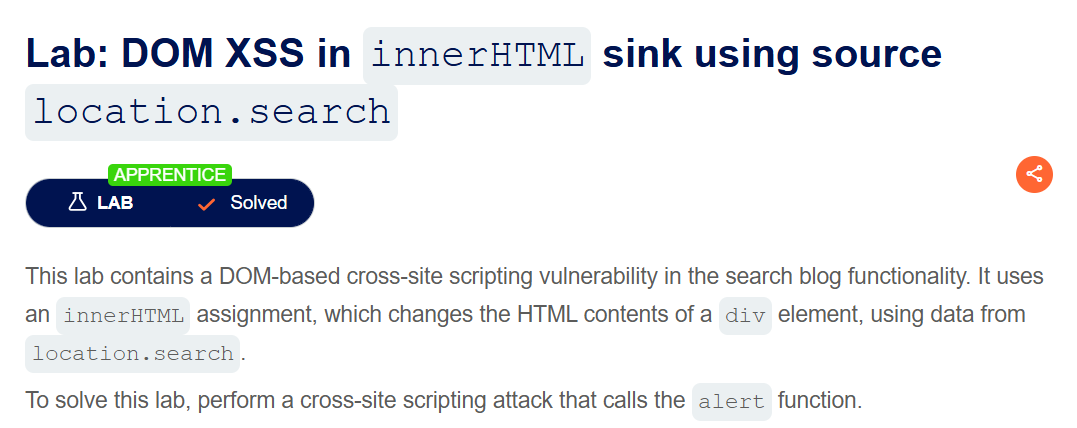
AI-generated content may be incorrect.

A orange rectangle with white text

AI-generated content may be incorrect.

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

**Instruction:**

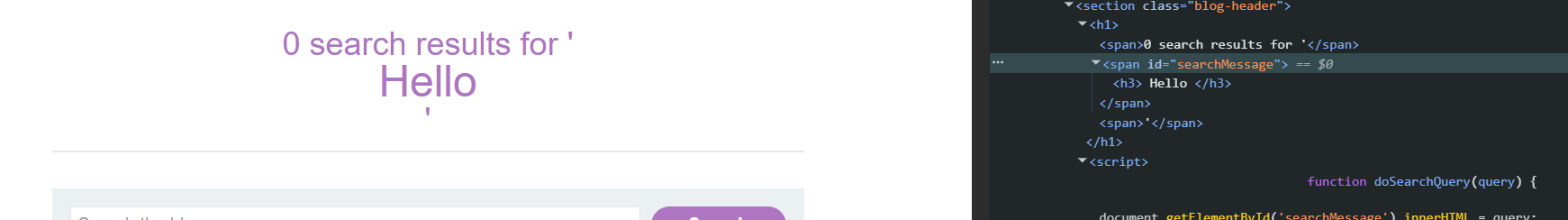


**Solution**:

A screenshot of a computer screen

AI-generated content may be incorrect.

Search <h3> Hello </h3>



<script>alert('Lab’) </script>

A purple and white button

AI-generated content may be incorrect.

Enter the payload: A white rectangular object with purple lines

AI-generated content may be incorrect.

[to force an error: - src(1); invalid source , an onerror event handler containing the malicious JavaScript]

A screenshot of a computer

AI-generated content may be incorrect.

Popup is appearing. The page is vulnerable.



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

**Instruction:**

**A screenshot of a web browser

AI-generated content may be incorrect.**

**Solution**:

1. Navigate to submit feedback page.
2. The URL:



Inspect the page:

A screenshot of a computer

AI-generated content may be incorrect.

1. Modify the URL to:



A screenshot of a computer

AI-generated content may be incorrect.

1. ‘Return path’ parameter controls the back links destination, and it directly updates the href tag.
2. Since href accepts JavaScript; enter the payload:



A screen shot of a computer

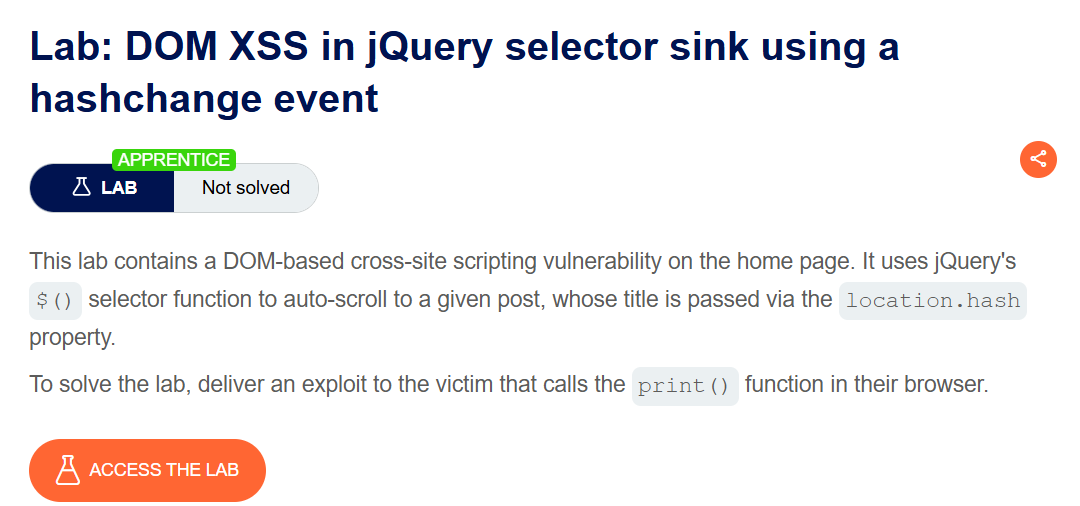
AI-generated content may be incorrect.

A orange rectangle with white text

AI-generated content may be incorrect.

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

**Instruction:**

****

**Solution**:

1. Access the lab and see where the jQuery selector exists.

A screen shot of a computer code

AI-generated content may be incorrect.

1. go to the exploit server and paste the payload.

A screenshot of a computer

AI-generated content may be incorrect.

Click ‘view exploit’.



And then click ‘Deliver exploit to victim’. Then the lab will be solved.

A close up of a sign

AI-generated content may be incorrect.

# Lab 7: Reflected XSS into attribute with angle brackets HTML-encoded

**Instruction:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Solution**:

1. To see is the angle brackets are encoded in the search functionality search something in the search bar.

A white and black striped background

AI-generated content may be incorrect.

1. In the source code the brackets are encoded with < = &it and > = &gt.

A black screen with white text

AI-generated content may be incorrect.

1. Insert a malicious payload:

A purple line on a white background

AI-generated content may be incorrect.

1. When we move the mouse over the search bar. it pops up an alert.

A screenshot of a computer

AI-generated content may be incorrect.

A orange rectangle with white text

AI-generated content may be incorrect.

# Lab 8: Stored XSS into anchor *href* attribute with double quotes HTML-encoded

**Instruction:**

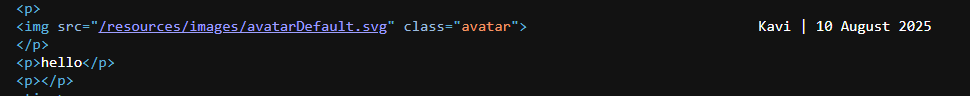
**A close-up of a computer screen

AI-generated content may be incorrect.**

**Solution**:

A screenshot of a chat

AI-generated content may be incorrect.



A screenshot of a computer

AI-generated content may be incorrect.

A close-up of a message

AI-generated content may be incorrect.

# Lab 9: Reflected XSS into a JavaScript string with angle brackets HTML encoded

**Instruction:**

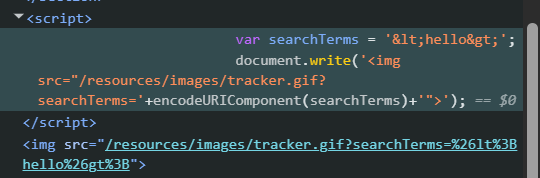
****

**Solution**:

In this lab angle brackets (>, <) are html encoded as &gt and &lt. but the other symbols (/, “) are not encoded.

A white rectangular object with purple lines

AI-generated content may be incorrect.



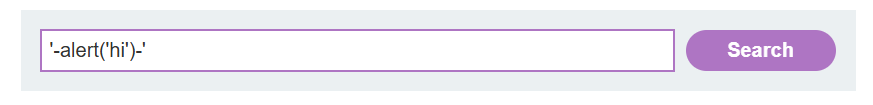
A purple line on a white surface

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

So we can use a single command to inject malicious code.



A screenshot of a computer

AI-generated content may be incorrect.

Pop up appeared.

A orange rectangle with white text

AI-generated content may be incorrect.