**Lab: Reflected XSS into HTML context with nothing encoded**

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

Goal: perform a cross-site scripting attack that calls the alert function.

Solution:

1-

A screenshot of a computer

Description automatically generated

2-<script>alert(1)</script>

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Lab: Stored XSS into HTML context with nothing encoded**

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

Goal: submit a comment that calls the alert function when the blog post is viewed.

Solution:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

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

Problem: This lab contains a reflected cross-site scripting vulnerability in the search blog functionality where angle brackets are HTML-encoded.

Goal: perform a cross-site scripting attack that injects an attribute and calls the alert function.

Solution:

1-A computer screen with many colorful text

Description automatically generated with medium confidence

2-A screenshot of a search box

Description automatically generated

3-A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

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

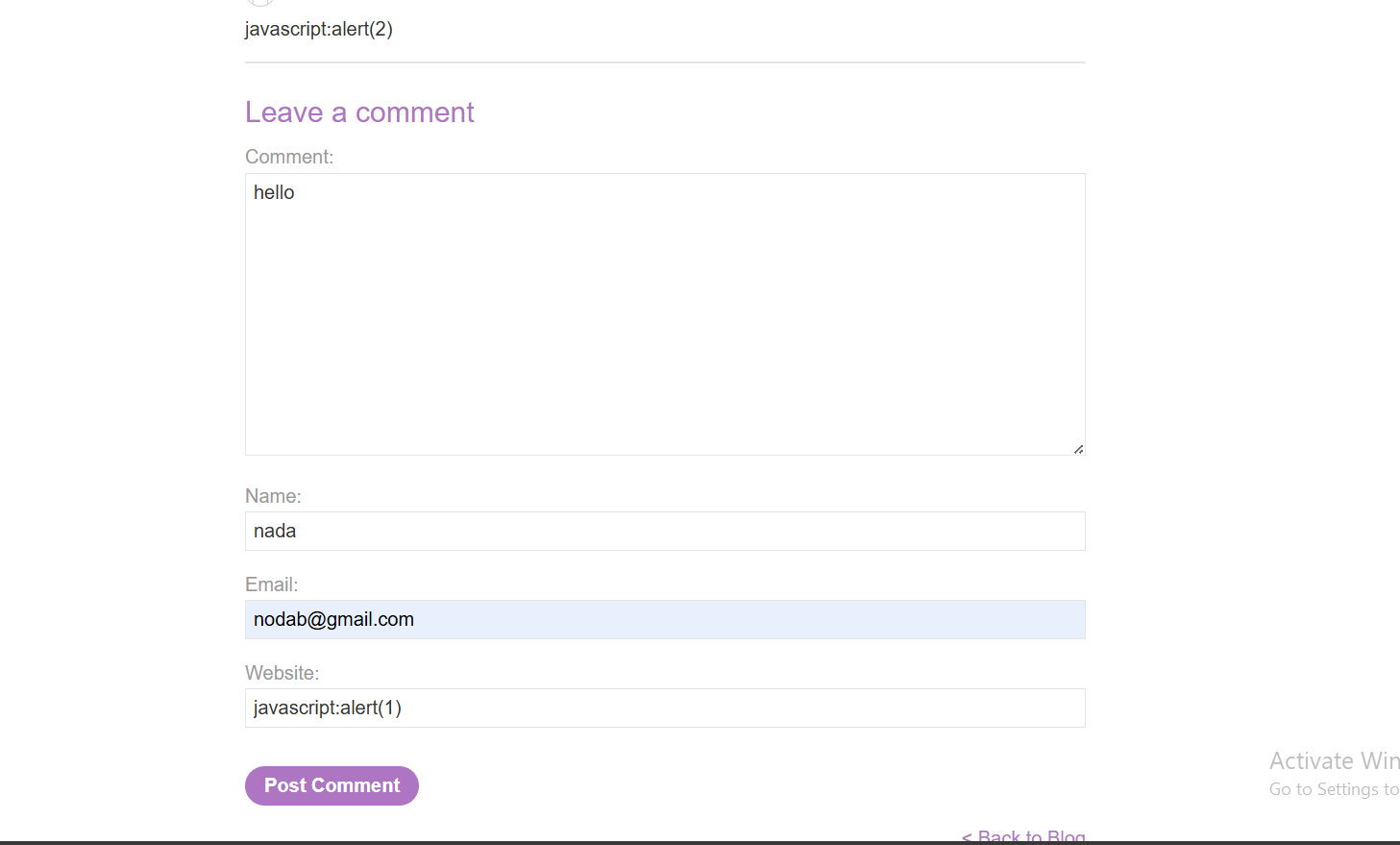
Problem: This lab contains a stored cross-site scripting vulnerability in the comment functionality.

Goal: submit a comment that calls the alert function when the comment author name is clicked.

Solution:

1-A screenshot of a computer

Description automatically generated

2- websitd🡪 javascript:alert(1)

A screenshot of a computer

Description automatically generated

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

Problem: This lab contains a reflected cross-site scripting vulnerability in the search query tracking functionality where angle brackets are encoded. The reflection occurs inside a JavaScript string.

Goal: perform a cross-site scripting attack that breaks out of the JavaScript string and calls the alert function.

Solution:

1-A black screen with many colorful lines

Description automatically generated

2-

A black background with many small colored lines

Description automatically generated with medium confidence

3-‘-alert(1)-‘

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

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

Problem: It uses the JavaScript document.write function, which writes data out to the page. The document.write function is called with data from location.search, which you can control using the website URL.

Goal: perform a cross-site scripting attack that calls the alert function.

Solution:

1-

A screenshot of a computer

Description automatically generated

2-

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

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

Problem: It uses an innerHTML assignment, which changes the HTML contents of a div element, using data from location.search.

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

Solution:

1-

A screen shot of a computer code

Description automatically generated

<img src=x onerror=alert(‘hi’)>

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

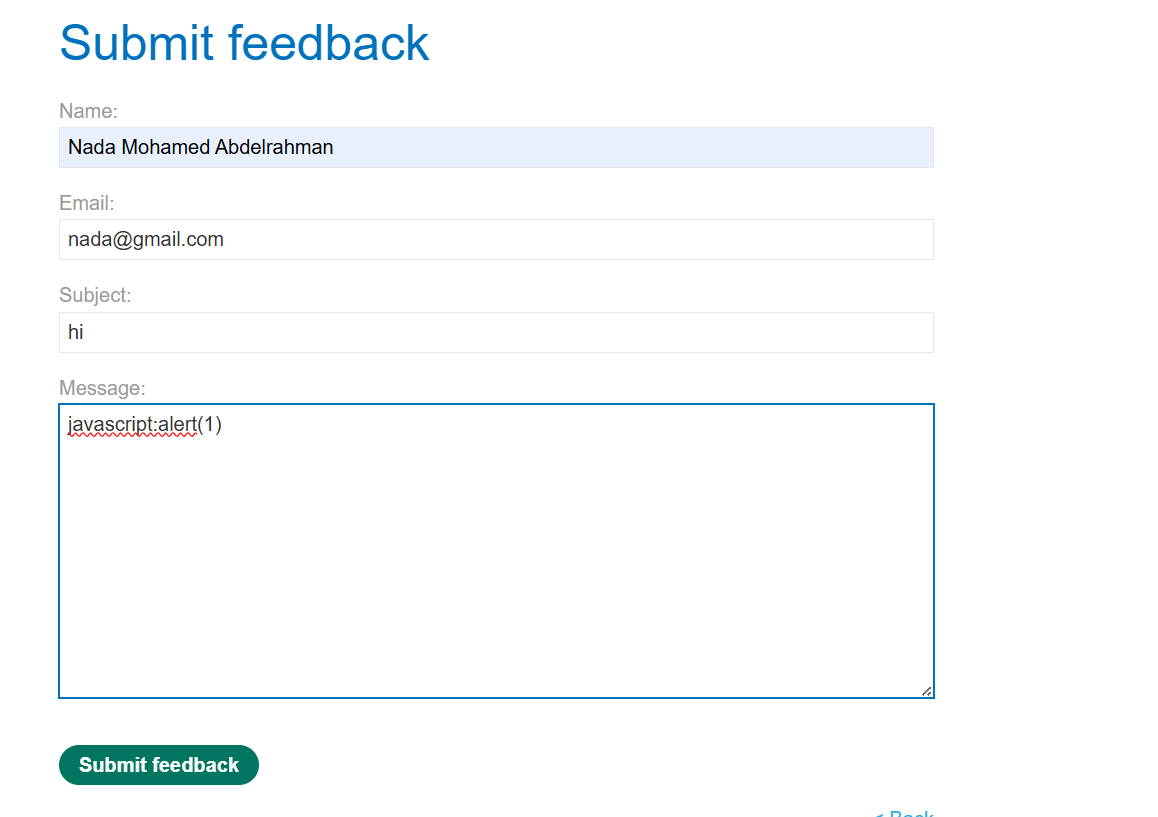
**Lab: DOM XSS in jQuery anchor**href**attribute sink using**location.search**source**

Problem: It uses the jQuery library's $ selector function to find an anchor element, and changes its href attribute using data from location.search.

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

Solution:

1-



2-A screenshot of a computer

Description automatically generated

3-

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

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

Problem:  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.

Goal: deliver an exploit to the victim that calls the print() function in their browser.

Solution:

1-

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated