

BÁO CÁO BÀI TẬP

Môn học: Bảo mật web và ứng dụng

Tên chủ đề: XSS và CSRF

GVHD: 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 Trần Đức Anh	20520392	20520392@gm.uit.edu.vn

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

BÁO CÁO CHI TIẾT

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 <code>document.write</code> sink using source <code>location.search</code> →	✓ Solved
LAB	APPRENTICE DOM XSS in <code>innerHTML</code> sink using source <code>location.search</code> →	✓ Solved
LAB	APPRENTICE DOM XSS in jQuery anchor <code>href</code> attribute sink using <code>location.search</code> source →	✓ Solved
LAB	APPRENTICE DOM XSS in jQuery selector sink using a hashchange event →	✓ Solved
LAB	APPRENTICE Reflected XSS into attribute with angle brackets HTML-encoded →	✓ Solved
LAB	APPRENTICE Stored XSS into anchor <code>href</code> attribute with double quotes HTML-encoded →	✓ Solved
LAB	APPRENTICE Reflected XSS into a JavaScript string with angle brackets HTML encoded →	✓ Solved

Cross-site request forgery (CSRF)

LAB

APPRENTICE

CSRF vulnerability with no defenses →

✓ Solved

Lab: Reflected XSS into HTML context with nothing encoded

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

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

Lỗi hỏng liên quan đến chức năng search -> nhập bất kỳ và search:

0 search results for 'abc'

Search

[< Back to Blog](#)

0 search results for '123'

Search

[< Back to Blog](#)

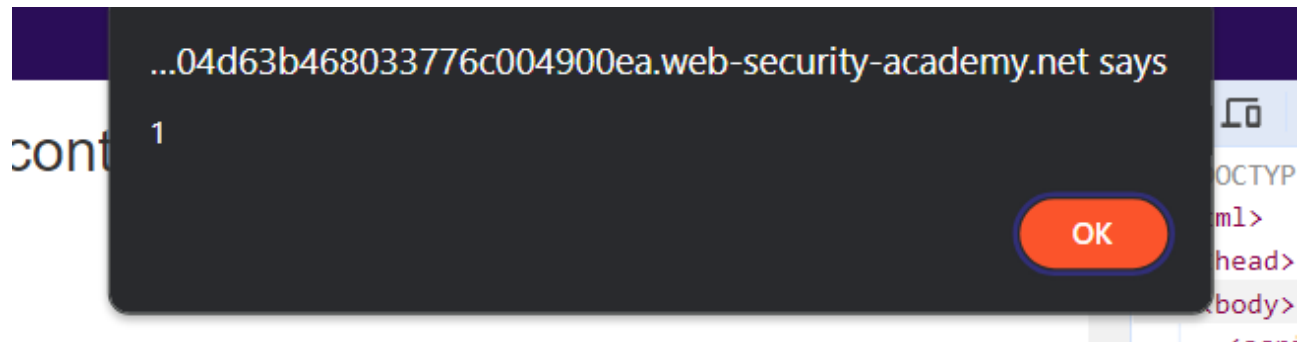
Ở đây có thể thấy nội dung nhập vào hiện trên thanh tìm kiếm. Xem page source thì có thể thấy nội dung nhập vào nằm trong thẻ `<h>` của HTML -> Có thể chèn script vào: `<script>alert(1)</script>`

0 search results for '123'

Search

[< Back to Blog](#)

Nội dung tìm kiếm trong thẻ <script> được thực thi:



Lab: Stored XSS into HTML context with nothing encoded

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

To solve this lab, submit a comment that calls the `alert` function when the blog post is viewed.

Vào bài blog bất kỳ, ở đây có phần bình luận:

Comments



Mick Mouse | 12 March 2024

Could you do a blog on hair loss? Asking for a friend. A very balding friend. The sooner the better really, please.

Leave a comment

Comment:

Name:

Email:

Website:

Post Comment

[< Back to Blog](#)

Thử bình luận gì đó bất kỳ:

Leave a comment

Comment:

abc

Name:

abc

Email:

abc@abc

Website:

<https://www.google.com>

Post Comment

Quay trở lại blog thì thấy tương tự lab trên, nội dung nhập vào được nằm trong thẻ <p> của HTML -> có thể chèn script vào. Khác ở chỗ nội dung được lưu trong CSDL nên tải lại trang vẫn còn:



abc | 28 March 2024

abc

Leave a comment

Comment:

```
> <section class="comment">... </section>
▼ <section class="comment">
  ▶ <p>... </p> == $0
    <p>abc</p>
    <p></p>
  </section>
<hr>
```

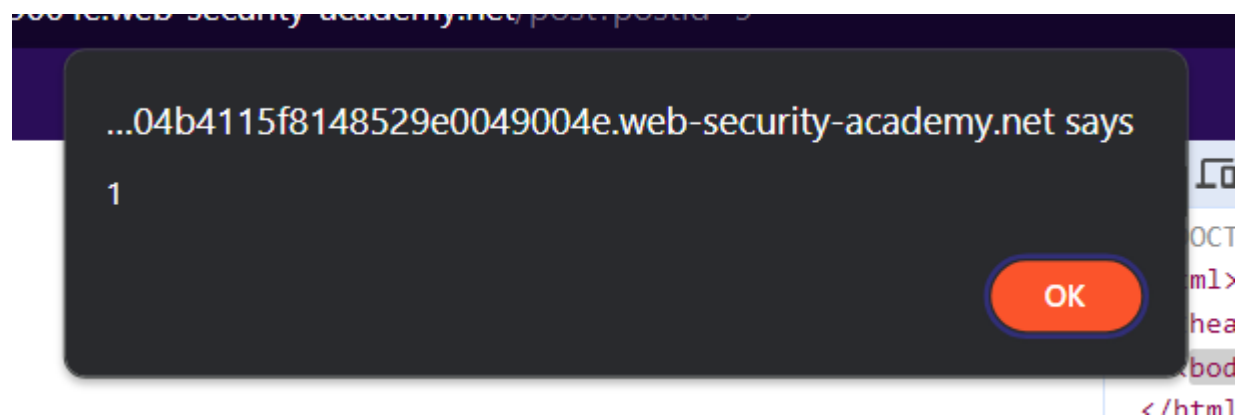
Chèn script vào bằng cách điền vào comment:

Leave a comment

Comment:

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

Script được thực thi mỗi khi vào trang blog:



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

This lab contains a DOM-based cross-site scripting vulnerability in the search query tracking functionality. 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.

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

Thử tìm kiếm như bình thường:

BLUG

Search

Nội dung được hiển thị nhưng không thực thi script:

0 search results for '<script>alert(1)</script>'

Search

Xem page source và thấy đoạn script xử lý nội dung tìm kiếm được thêm nội dung từ hàm `document.write`. Trong thẻ `` có chứa nội dung tìm kiếm:

```

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

```


Vậy cần nhập nội dung để đóng thẻ `` và thực hiện script alert:
"><script>alert(1)</script>

```


<script>alert(1)</script>
""> "
▶ <section class="blog-list no-results">... </section>

```

Kết quả là script sẽ được thực thi

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

This lab contains a DOM-based cross-site scripting vulnerability in the search blog functionality. It uses an `innerHTML` assignment, which changes the HTML contents of a `div` element, using data from `location.search`.

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

Tìm kiếm tương tự lab trên sau đó xem page source:

```

▼ <h1>
  <span>0 search results for '</span>
  <span id="searchMessage">abc</span>
  <span>'</span>
</h1>
▼ <script> == $0
    function doSearchQuery(query) {
        document.getElementById('searchMessage').innerHTML =
query;
    }
    var query = (new
URLSearchParams(window.location.search)).get('search');
    if(query) {
        doSearchQuery(query);
    }
</script>

```

Từ đây có thể thấy giá trị trong thẻ `abc` sẽ thay đổi thông qua DOM `innerHTML`.

Nhập thử `<script>alert(1)</script>` thấy không có gì xảy ra

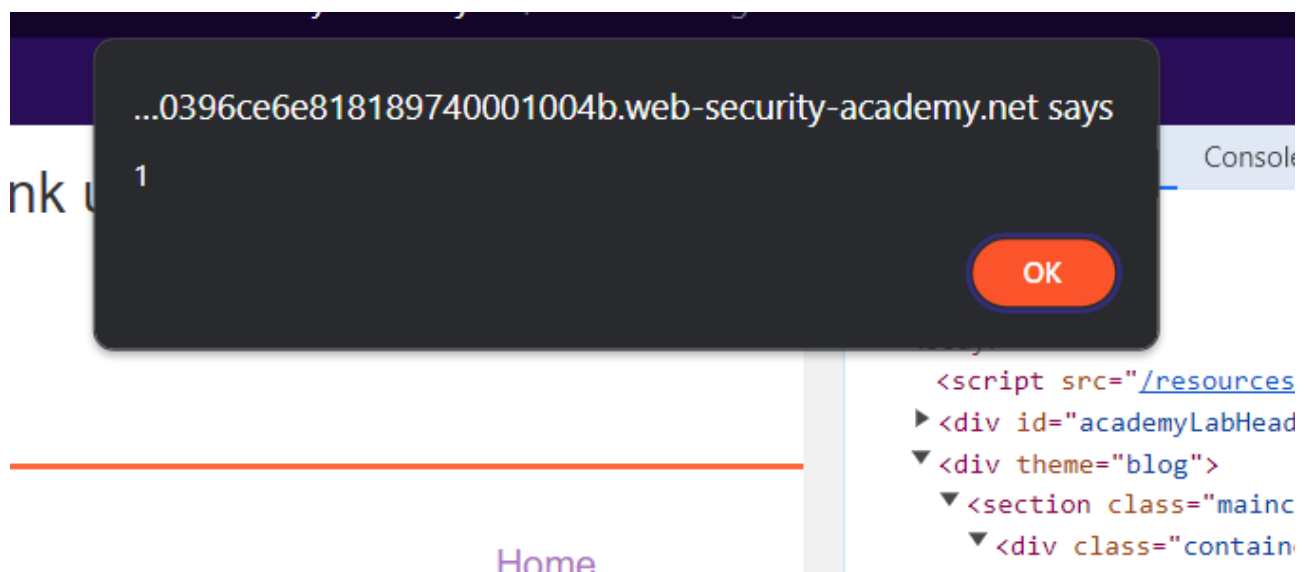
```
▼ <h1>
  <span>0 search results for '</span>
  ▼ <span id="searchMessage"> == $0
    <script>alert(1)</script>
  </span>
  <span>'</span>
</h1>
```

Có thể thẻ script trong thẻ span không thực thi =)) Sử dụng thẻ thử `` với thuộc tính onerror:

0 search results for "

[< Back to Blog](#)

Thành công rồi:

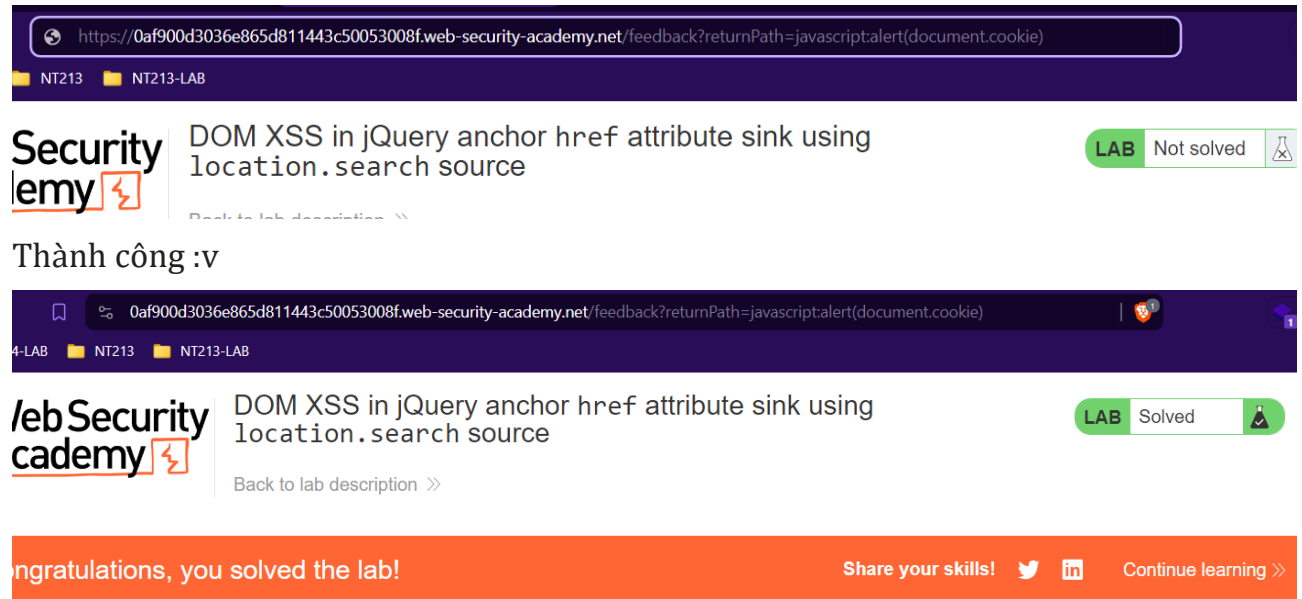


Lab: DOM XSS in jQuery anchor href attribute sink using location.search source

This lab contains a DOM-based cross-site scripting vulnerability in the submit feedback page. It uses the jQuery library's `$` selector function to find an anchor element, and changes its `href` attribute using data from `location.search`.

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

Theo đề bài, vào submit feedback page. Thay đổi giá trị trên URL thành `javascript:alert(document.cookie)`



Security Academy

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

LAB Not solved

Thành công :v

0af900d3036e865d811443c50053008f.web-security-academy.net/feedback?returnPath=javascript:alert(document.cookie)

LAB Solved

Congratulations, you solved the lab!

Share your skills! Continue learning >>

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

This lab contains a DOM-based cross-site scripting vulnerability 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.

To solve the lab, deliver an exploit to the victim that calls the `print()` function in their browser.

Xem page source thì thấy đoạn script này liên quan đến hashchange.

```
<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();
}); == $0
</script>
```

Window: hashchange event

The `hashchange` event is fired when the fragment identifier of the URL has changed (the part of the URL beginning with and following the `#` symbol).

Theo tài liệu thì đoạn script trên sẽ thực hiện cuộn đến tiêu đề của bài đăng nếu thêm #tiêu đề vào URL của web

Thử nhúng URL để gọi hàm `print()` từ server để gửi đến victim:

Body:

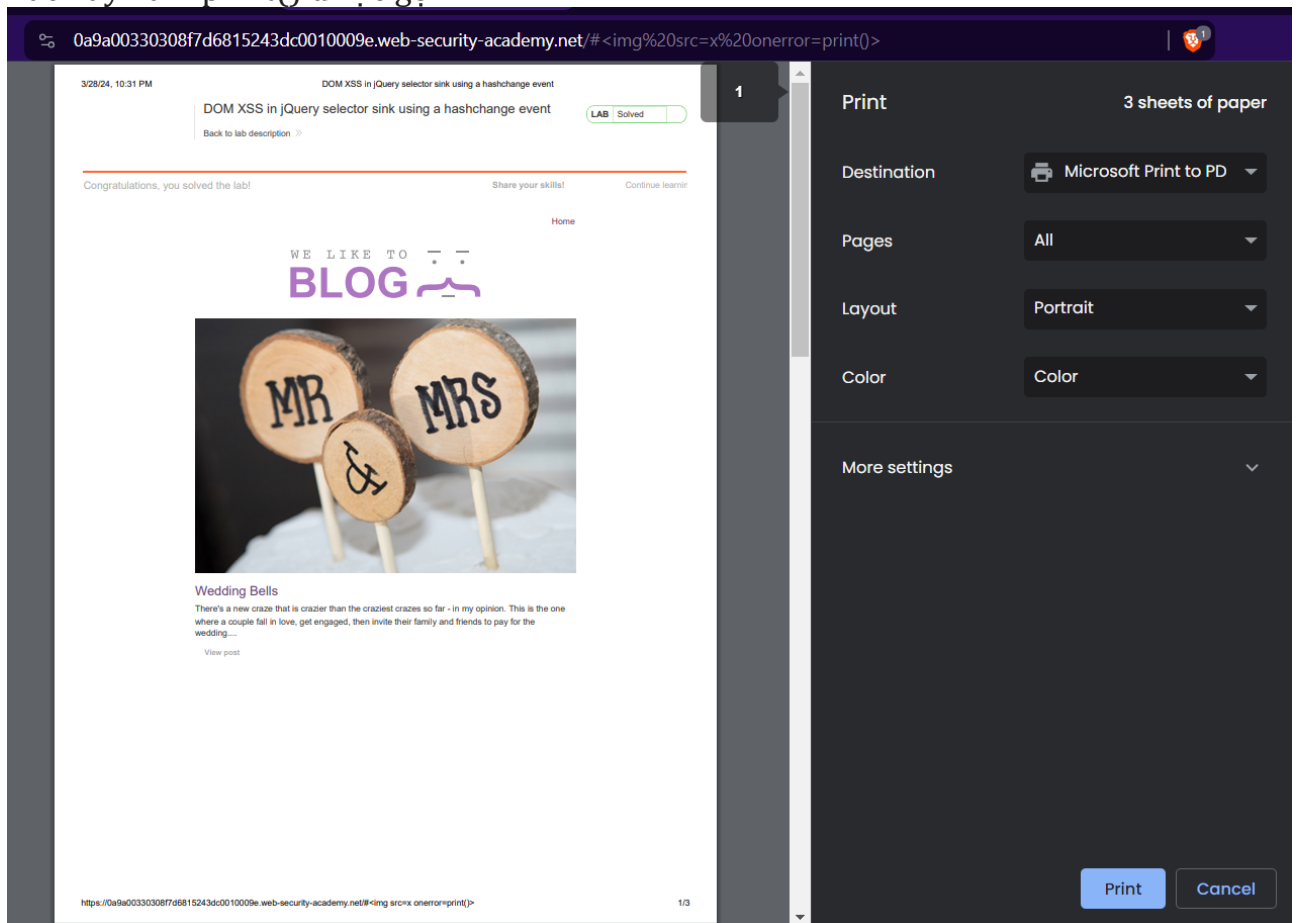
```
<iframe src="https://0a9a00330308f7d6815243dc0010009e-security-academy.net/#<img src=x onerror=print()>"></iframe>
```

Tuy nhiên không có gì xảy ra. Thử cách khác:

Body:

```
<iframe src="https://0a9a00330308f7d6815243dc0010009e.web-security-academy.net/#" onload="this.src+= '<img src=x onerror=print()>' "></iframe>
```

Lúc này hàm print() được gọi:



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

This lab contains a reflected cross-site scripting vulnerability in the search blog functionality where angle brackets are HTML-encoded. To solve this lab, perform a cross-site scripting attack that injects an attribute and calls the `alert` function.

Thử tìm kiếm bất kỳ, thấy ở thẻ input này có thể khai thác

```
</section>
<section class="search">
  <form action="/" method="GET"> flex == $0
    <input type="text" placeholder="Search the blog..." name="search" value="abc">
    <button type="submit" class="button">Search</button>
  </form>
</section>
<section class="blog-list no-results">...</section>
```

Tuy nhiên dung < > lại không được -> dùng thử thuộc tính onmouseover

0 search results for 'abc'

[< Back to Blog](#)

Khai thác đã thành công.

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

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. To solve this lab, perform a cross-site scripting attack that breaks out of the JavaScript string and calls the `alert` function.

Tìm kiếm bất kỳ và kiểm tra nguồn,

```
<script> == $0
    var searchTerms = 'abc';
    document.write('');
</script>
```

Tìm kiếm thử:

0 search results for 'abc'

Khai thác thành công:

