BÁO CÁO THỰC HÀNH

Môn học: Bảo mật web và ứng dụng

Lab 1: Tổng quan các lỗ hổng bảo mật web thường gặp

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]

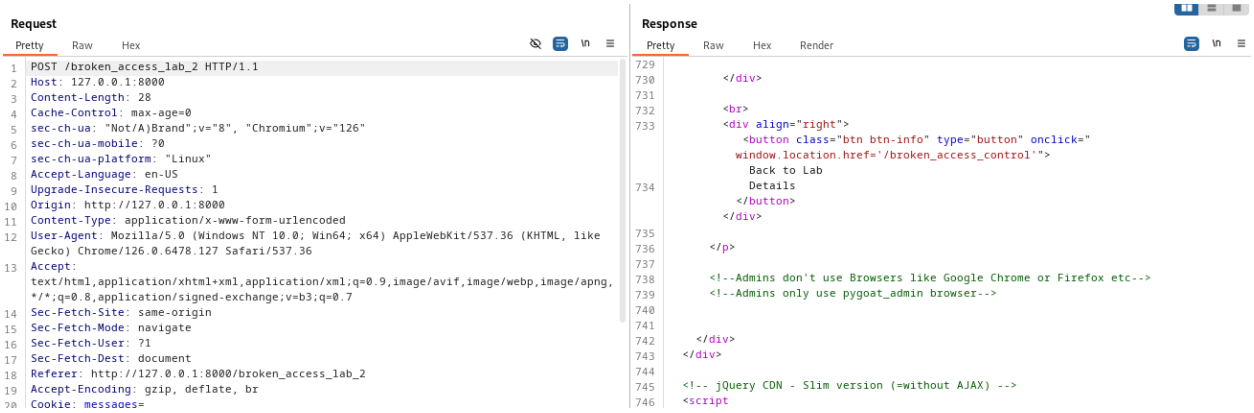
STT	Công việc	Kết quả tự đánh giá
1	Bài tập 1	100%
2	Bài tập 2	100%
3	Bài tập 3	100%
4	Bài tập 4	100%
5	Bài tập 5	100%
6	Bài tập 6	100%
7	Bài tập 7	100%
8	Bài tập 8	0%
9	Bài tập 9	0%
10	Bài tập 10	100%
11	Bài tập 11	100%
12	Bài tập 12	0%
13	Bài tập 13	0%

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

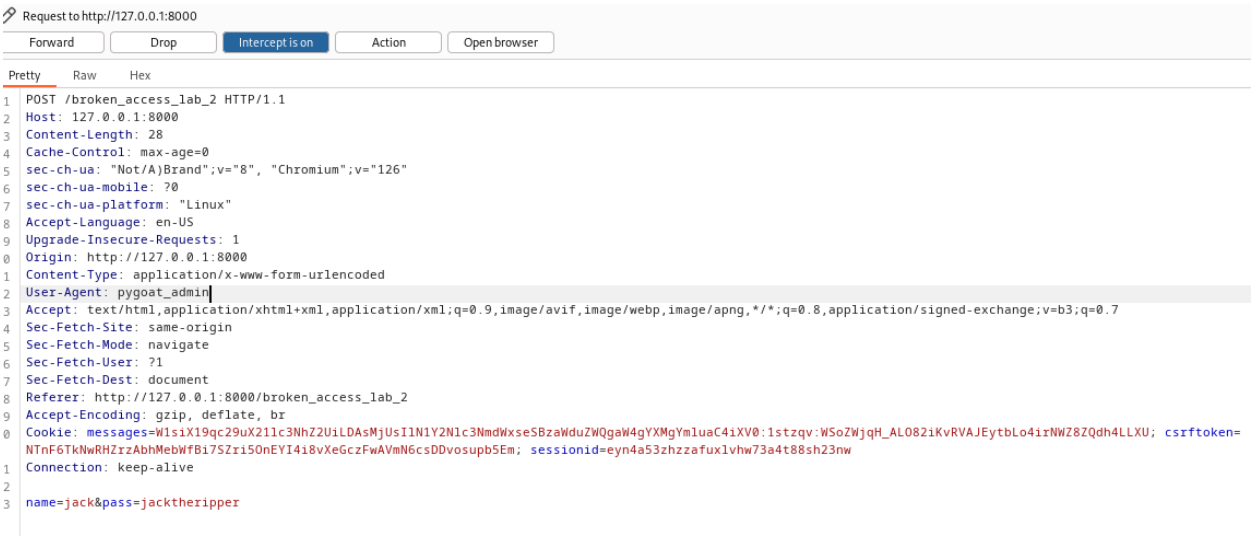
BÁO CÁO CHI TIẾT

1. http://localhost:8000/broken_access_lab_2

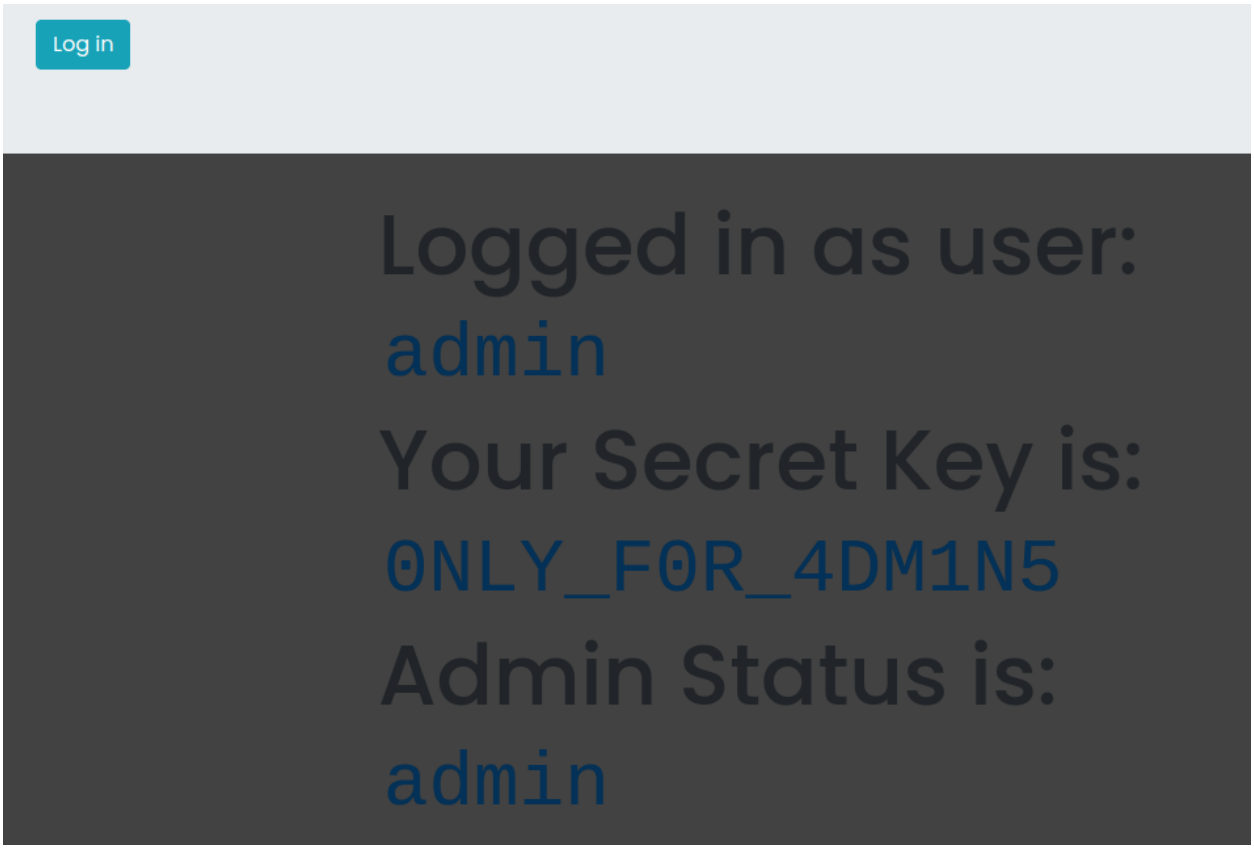
- Sau khi đăng nhập sử dụng tài khoản jack, từ response của server ta nhận thấy tài khoản admin chỉ sử dụng trình duyệt **pygoat_admin** :



- Sử dụng công cụ **intercept**, ta sửa user-agent(chứa thông tin về trình duyệt của người dùng) thành **pygoat_admin** và **forward** :



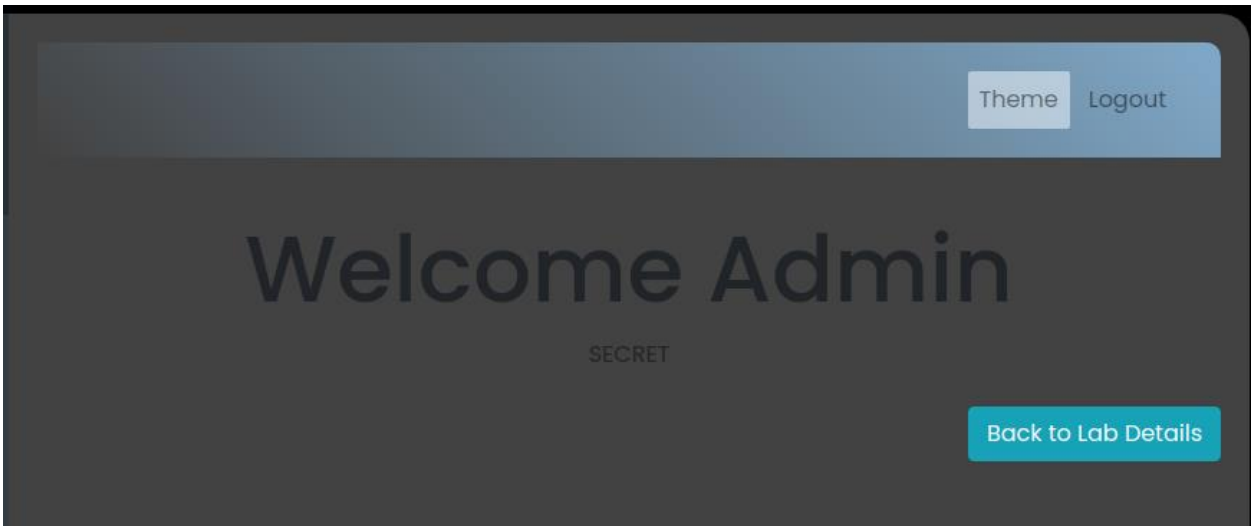
- Kết quả thu được:



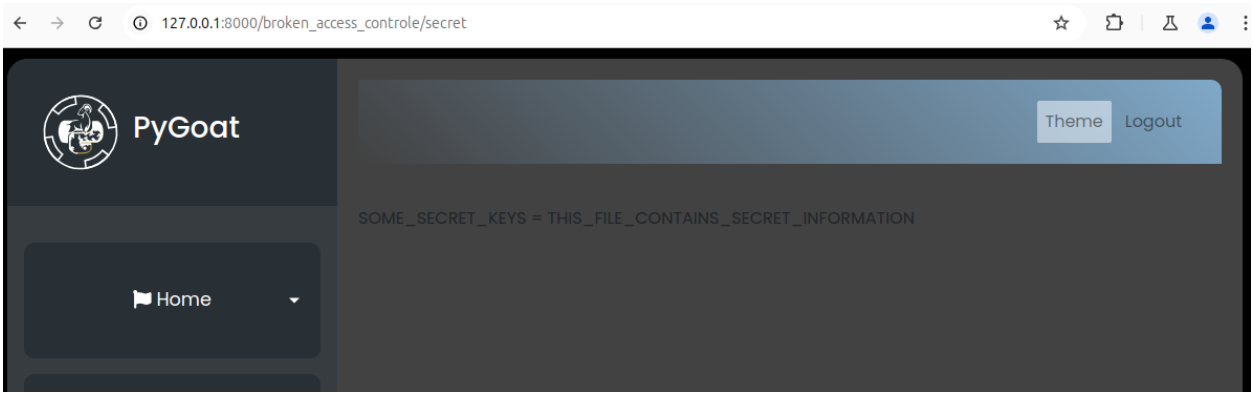
2. http://localhost:8000/broken_access_lab_3

- Đăng nhập bằng thông tin xác thực quản trị :

user : admin
password : admin_pass

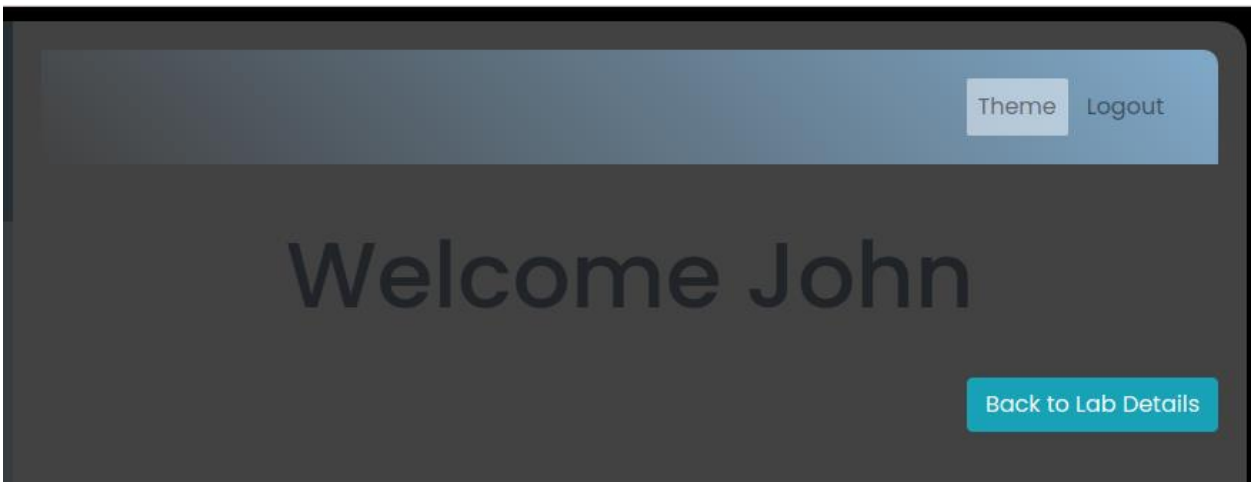


- Sau đó ta vào đường dẫn: **/broken_access_controle/secret** chứa thông tin bí mật :

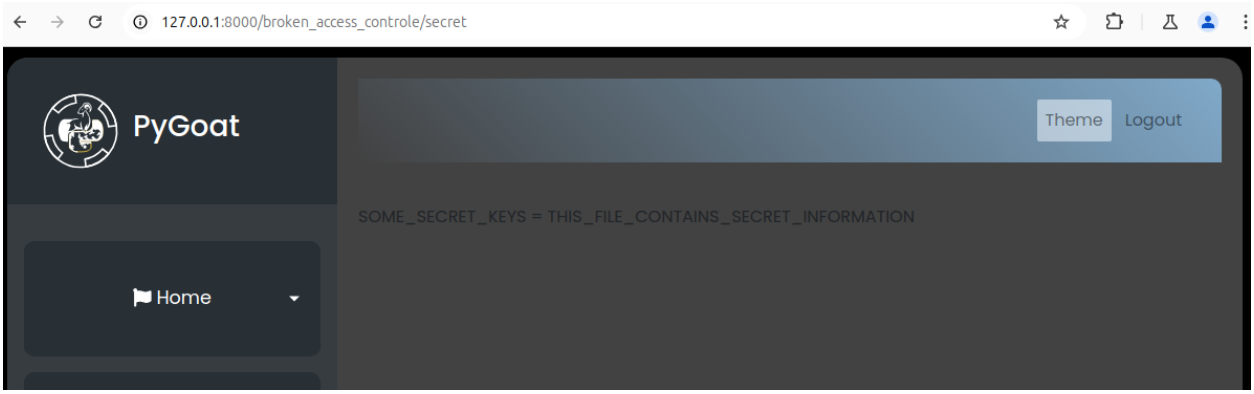


- Ta đăng xuất ra khỏi tài khoản quản trị và đăng nhập vào tài khoản người dùng, chúng ta không thể thấy tuỳ chọn cho thông tin bí mật nữa:

user : John
password : reaper



- Nếu chúng ta duyệt đến **/broken_access_controle/secret** thì chúng ta vẫn có thể truy cập trang đó vì không có kiểm tra xác thực nào được triển khai tại trang đó và đây là kết quả thu được:



- Trong trường hợp này cho thấy thực tế tin tặc không có thông tin xác thực quản trị nhưng vẫn có thể thấy được danh sách văn bản hiện có hoặc từ điển đường dẫn chung.

3. http://localhost:8000/cryptographic_failure/lab2

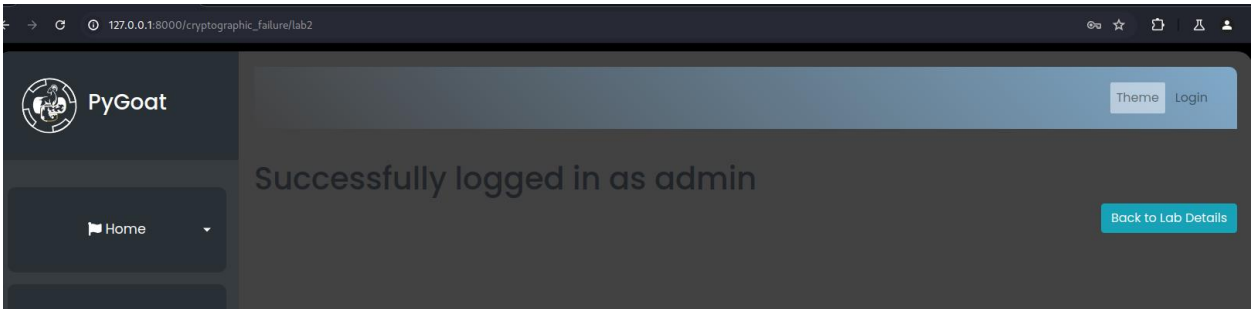
- Từ mã hash cho trước, ta sử dụng công cụ **hashid** để tìm thuật toán được sử dụng. Từ kết quả ta đoán mã hóa được sử dụng là **sha-256** hoặc **sha3-256** do tính phổ biến của chúng:

```
┌──(kali㉿kali)-[/usr/share/wordlists]
└─$ hashid -m d953b4a47ce307fcb8b1b85fc6a0d34aea5585b6ad9188beb94c1eea9bbb5c7a
Analyzing 'd953b4a47ce307fcb8b1b85fc6a0d34aea5585b6ad9188beb94c1eea9bbb5c7a'
[+] Snefru-256
[+] SHA-256 [Hashcat Mode: 1400]
[+] RIPEMD-256
[+] Haval-256
[+] GOST R 34.11-94 [Hashcat Mode: 6900]
[+] GOST CryptoPro S-Box
[+] SHA3-256 [Hashcat Mode: 5000]
[+] Skein-256
[+] Skein-512(256)
```

- Sau đó ta sử dụng công cụ **hashcat** với mode tương ứng của **sha-256** và **sha3-256**, cùng với **wordlist rockyou.txt**, thử lần lượt với mã hash gốc và mã hash gốc với thứ tự được đảo lại. Kết quả thu được là **password777** sử dụng **sha-256** với **mã hash bị đảo ngược** :

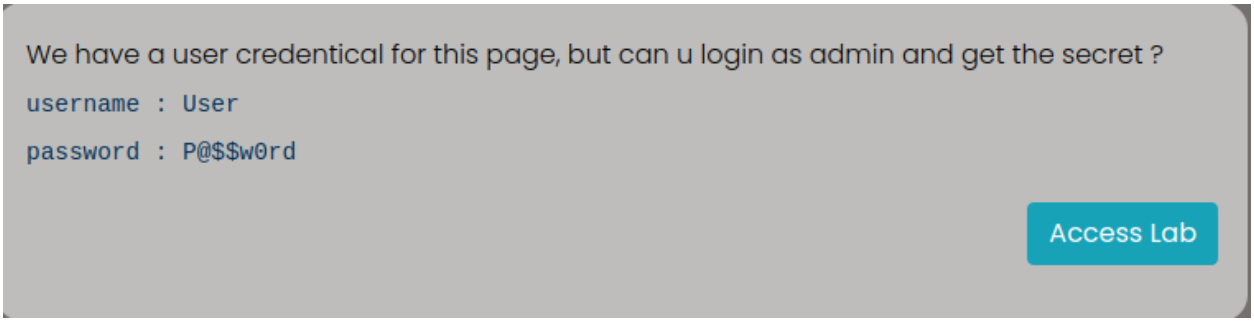
```
┌──(kali㉿kali)-[/usr/share/wordlists]
└─$ hashcat -m 1400 -a 0 admin-pass.txt  rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 3.1+debian  Linux, None+Asserts, RELOC, SPIR, LLVM 15.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
====================================================================================================================================
* Device #1: pthread-sandybridge-Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz, 2815/5694 MB (1024 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0×0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Hash
* Single-Salt
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1 MB

Dictionary cache hit:
* Filename..: rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

a7c5bbb9aee1c49beb8819da6b5855aea43d0a6cf58b1b8bcf703ec74a4b359d:password777

Session..........: hashcat
Status...........: Cracked
```

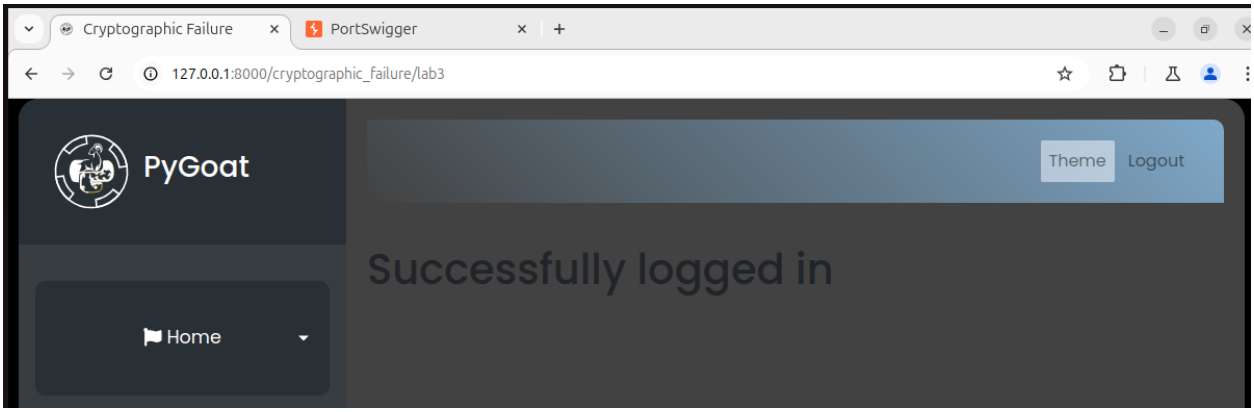
- Kết quả đăng nhập với admin-**password777:**



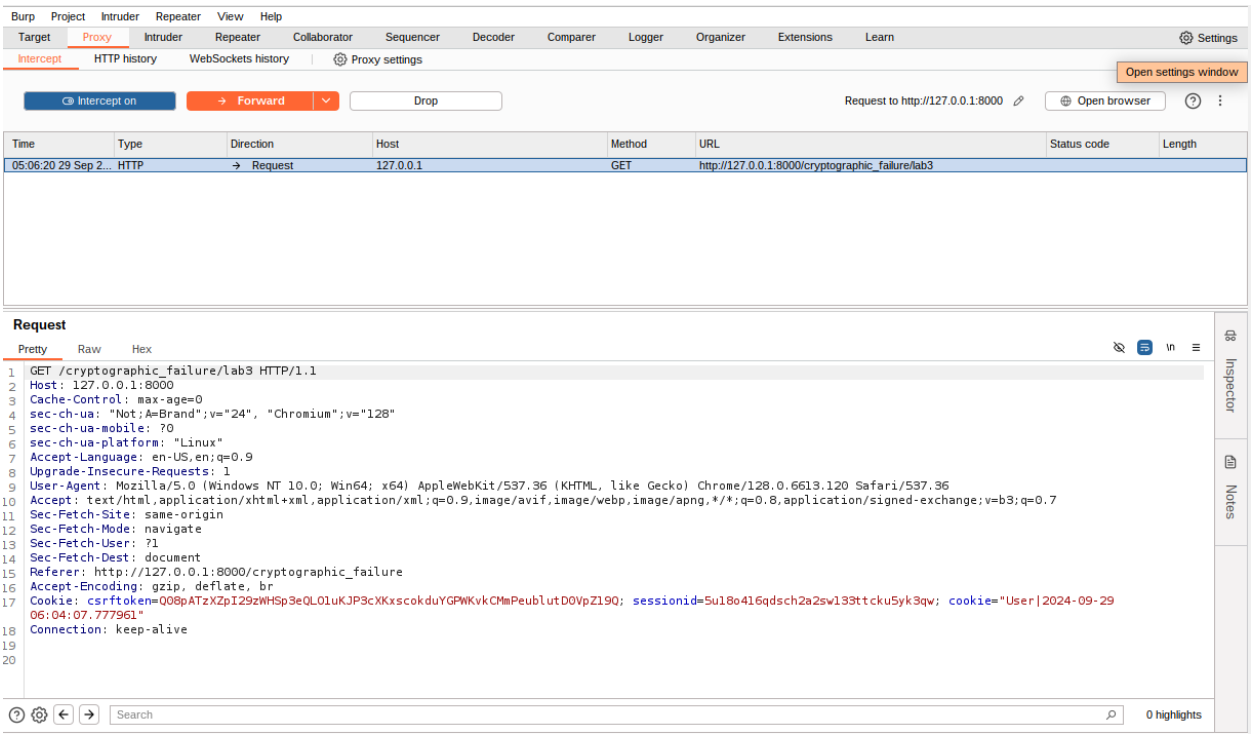
4. http://localhost:8000/cryptographic_failure/lab3

- Chúng ta đăng nhập vào tài khoản xác thực người dùng:

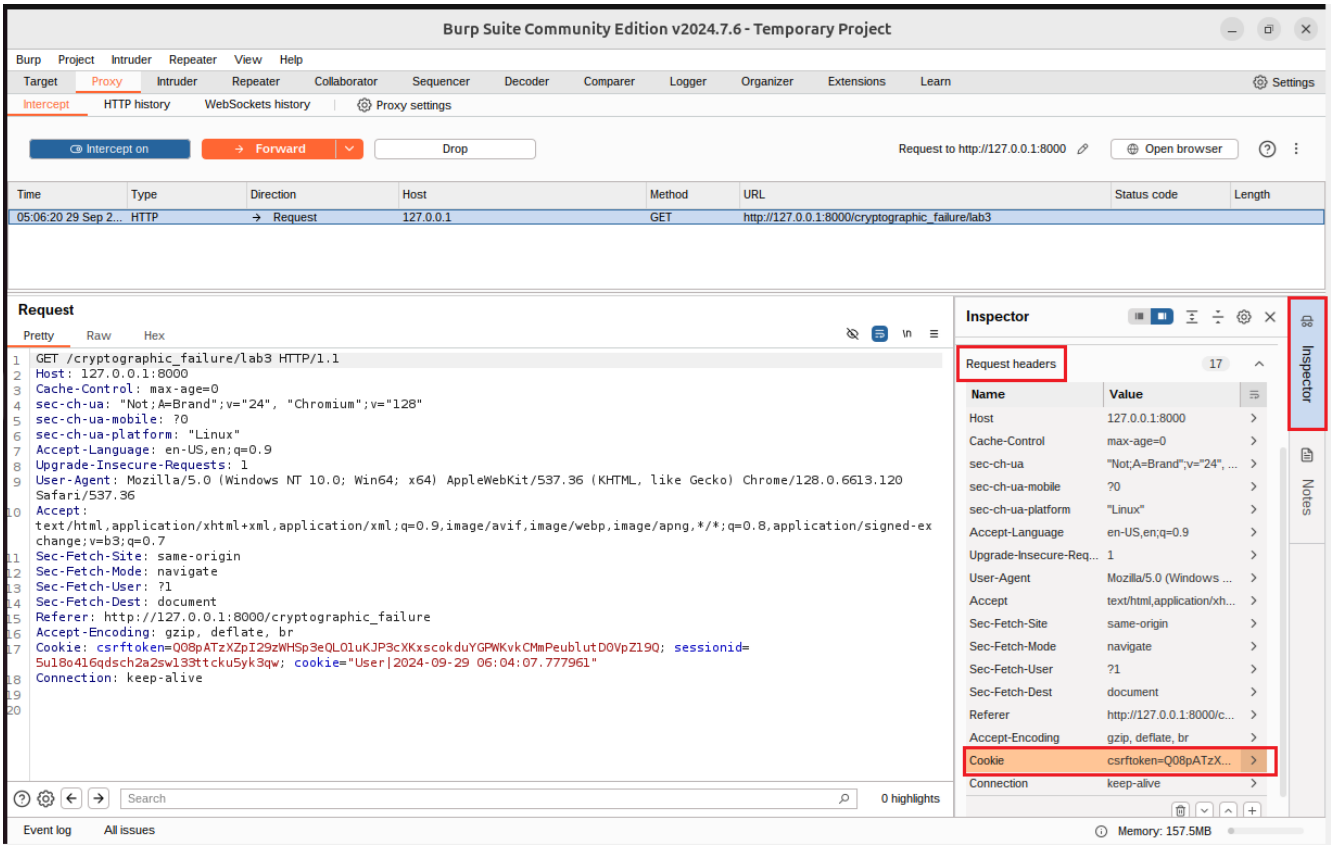




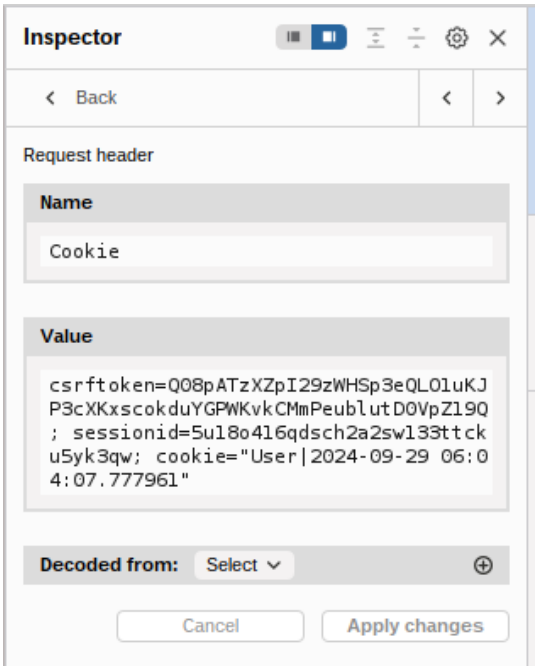
- Ta bật **Intercept is on** để chặn gói tin gửi lên server:



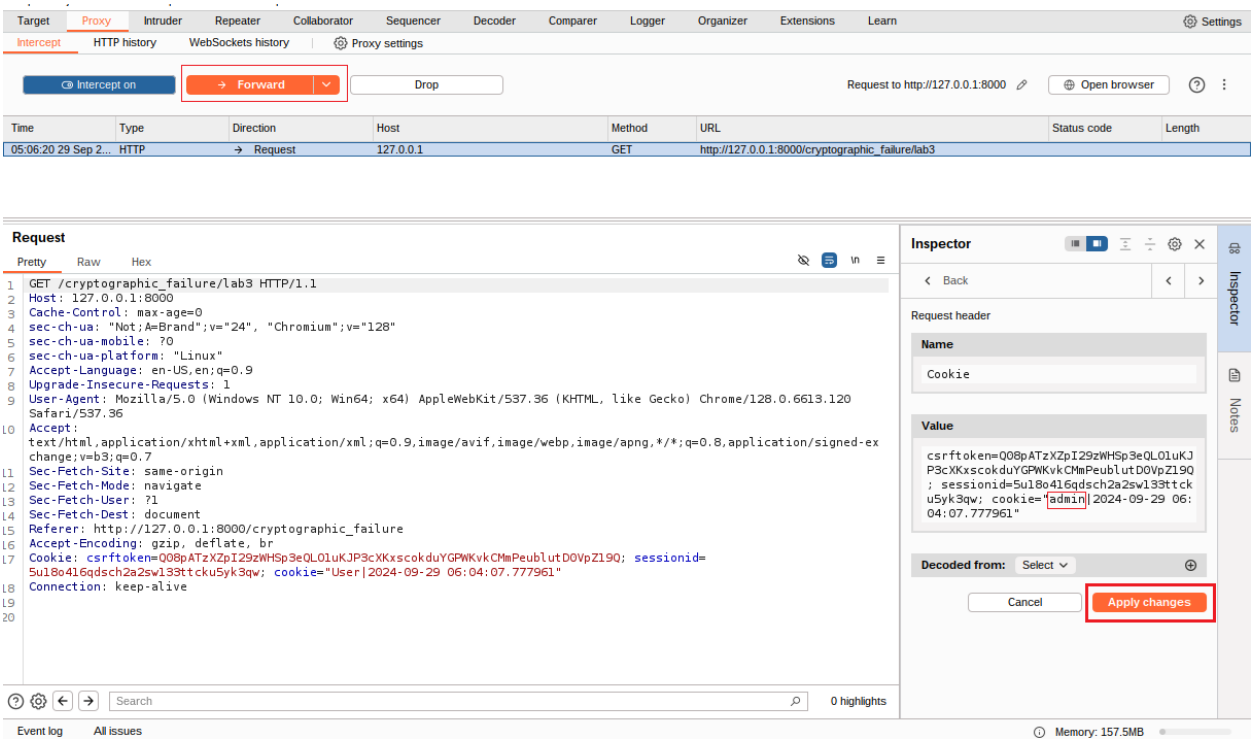
- Mở gói tin lên và sau đó ta chọn **Inspector > Request Hearders > Cookie** :



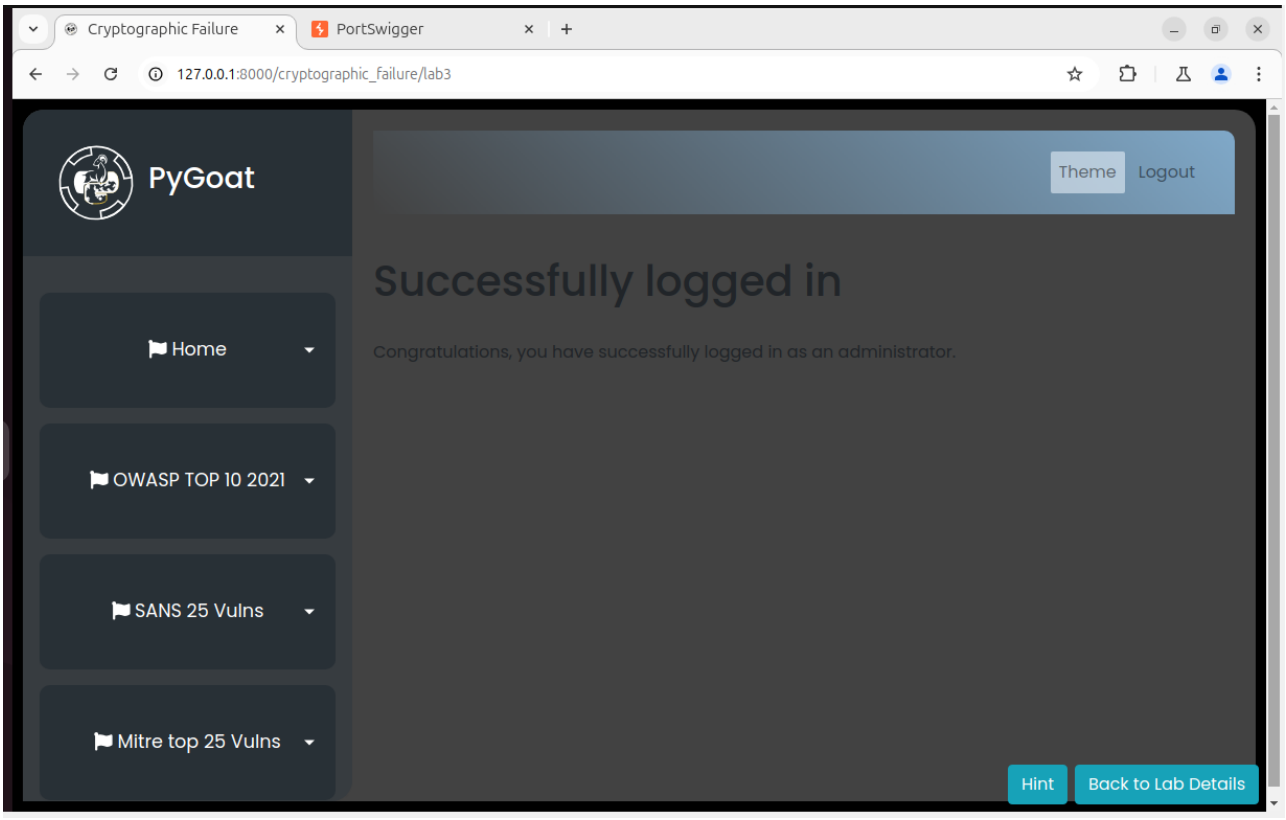
- Chọn **Cookie** hiển thị hình ảnh bên dưới:



- Sau đó ta tiến hành chỉnh sửa **User** thành **admin** rồi ta chọn **Apply changes** và **Forward** :



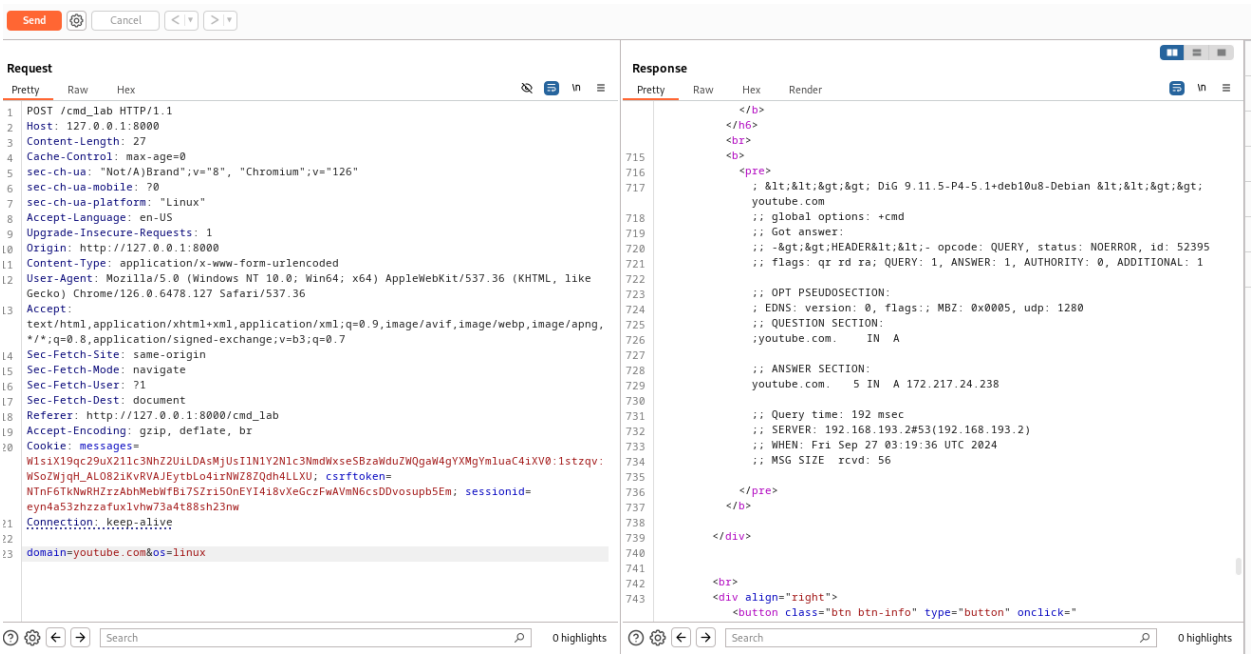
- Và đây là kết quả thu được:



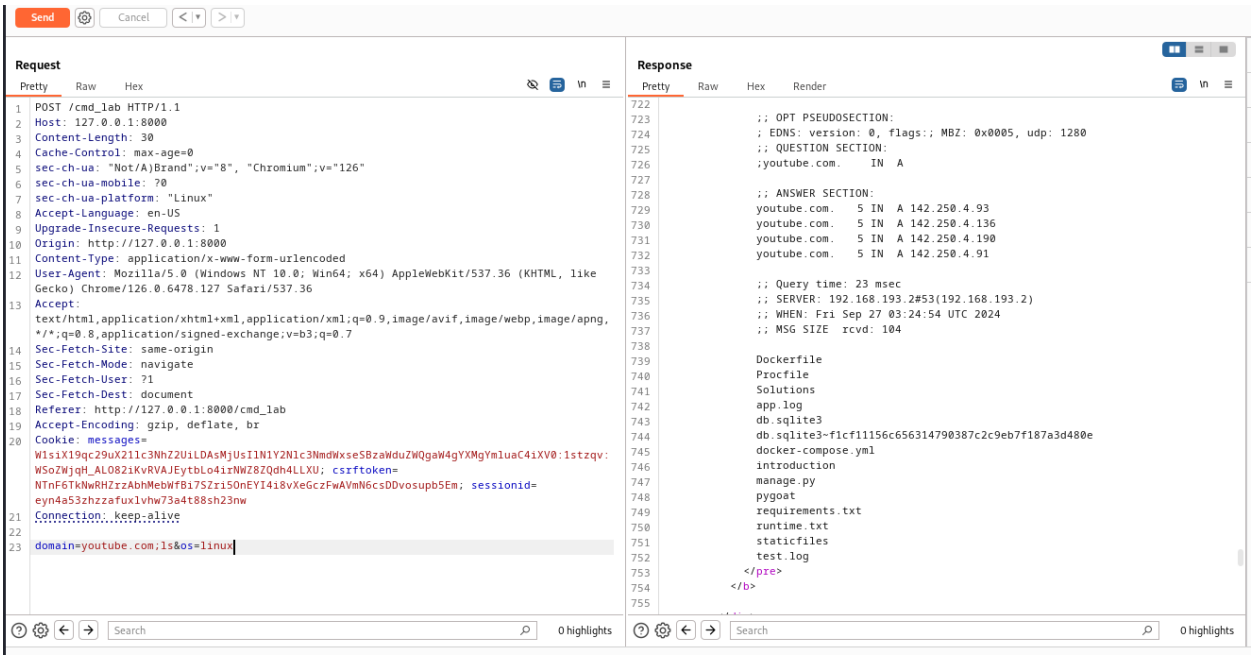
5. http://localhost:8000/cmd_lab

- Sử dụng công cụ **repeater** để gửi và kiểm tra kết quả phản hồi từ server. Ta nhận thấy có thể chèn kí tự **;** vào sau domain cần lookup để thực hiện lệnh khác.

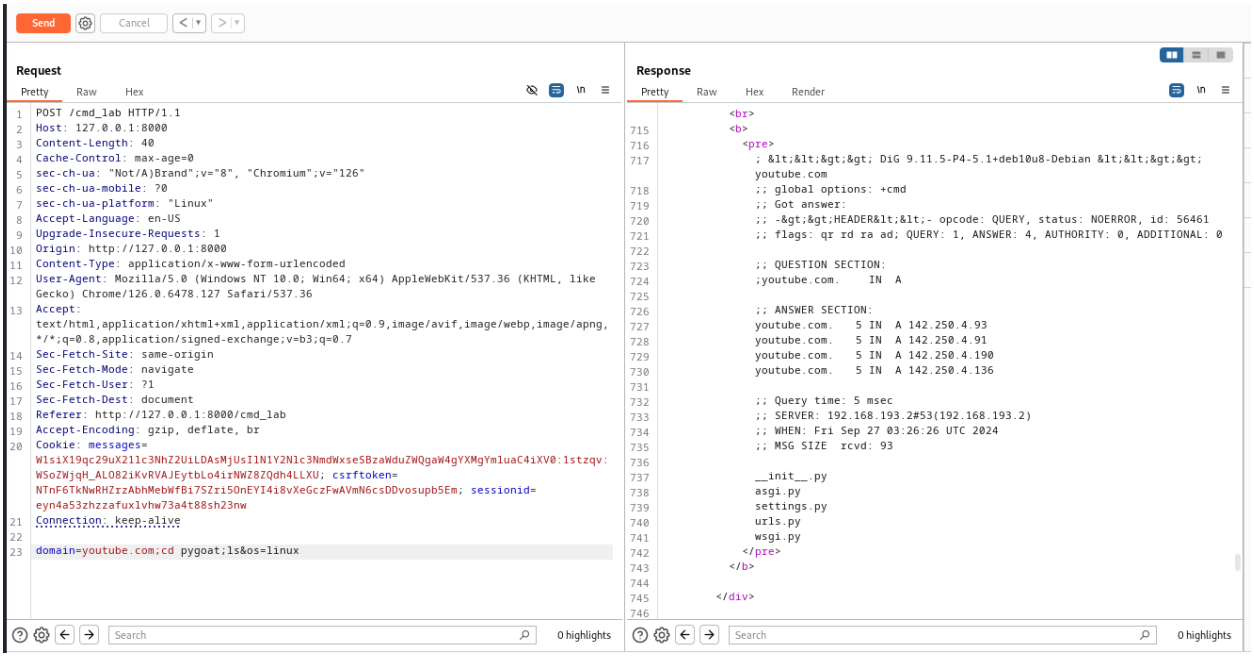
- Kết quả tìm kiếm domain **youtube.com** :



- Kết quả tìm kiếm domain **youtube.com;ls** :



- Kết quả tìm kiếm **youtube.com;cd pygoat;ls** :



6. http://localhost:8000/ssti/lab

- View code:

- Ta rút ra được dòng lệnh sau: {% load log %} {% get_admin_log 5 as log %} {% for e in log %} {{e.user.get_username}} : {{e.user.password}} {% endfor %}

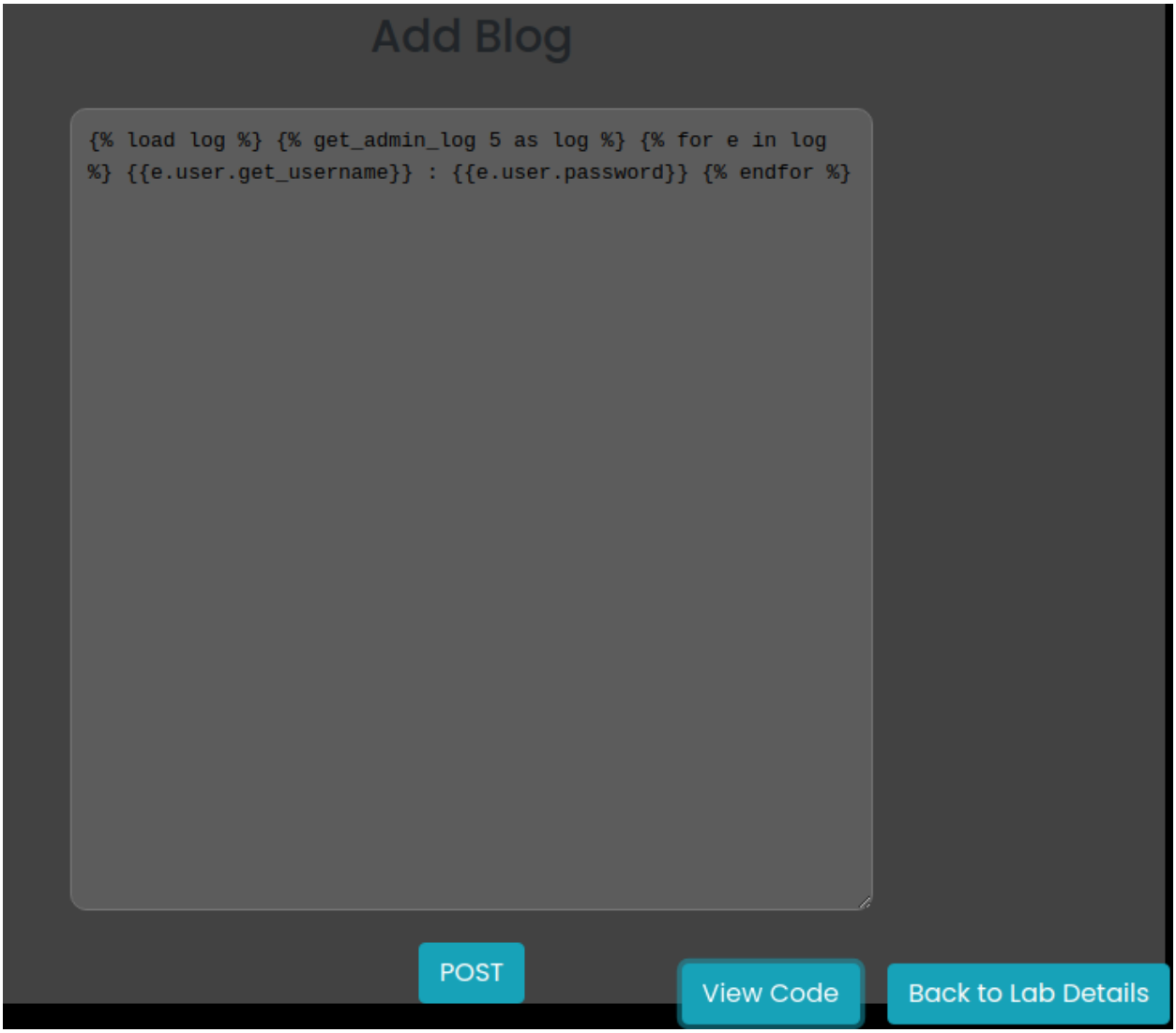
Trong đó:

+ {% load log %}: Nạp một module hoặc chức năng ghi log của hệ thống.

+ {% get_admin_log 5 as log %}: Lấy 5 dòng log từ admin và lưu dưới biến log.

+ {% for e in log %}: Vòng lặp chạy qua từng mục trong biến log.

+ {{e.user.get_username}} và {{e.user.password}}: In ra tên đăng nhập và mật khẩu của người



- Và đây là kết quả sau khi ta **Post**:

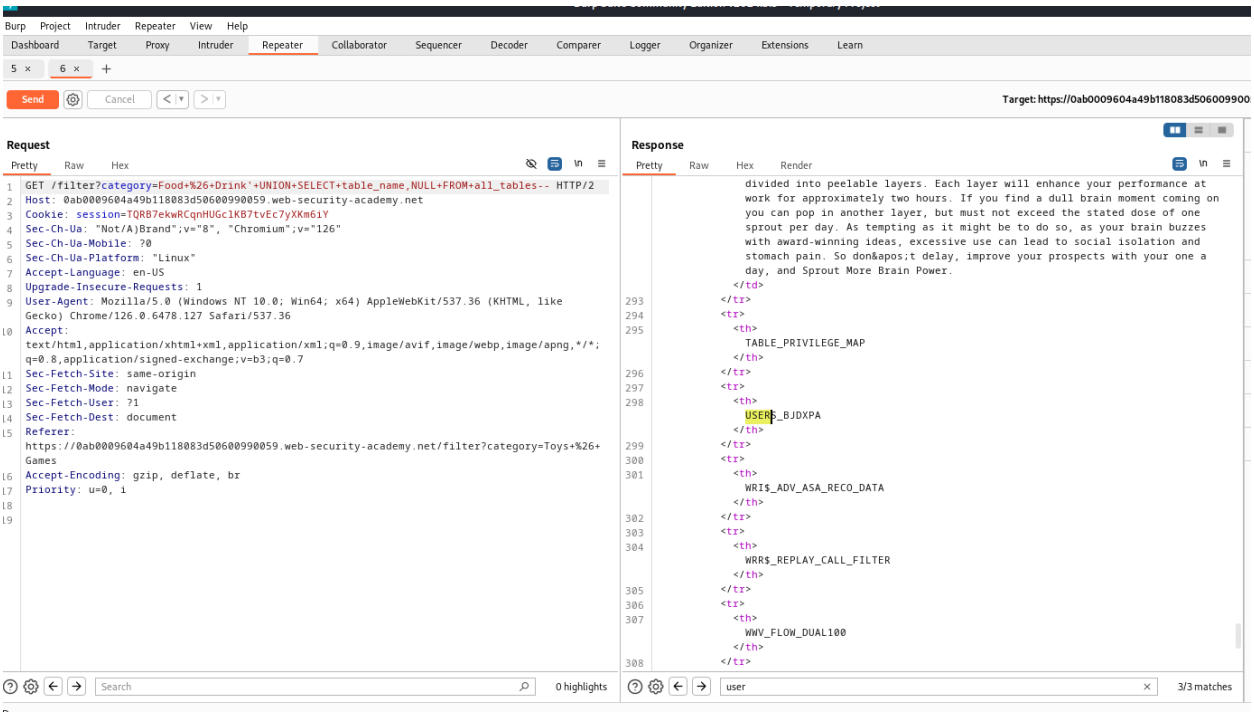
```
admin :
pbkdf2_sha256$320000$0DT0jMAKB1gI21vJlBGcuJ$QFldmvkyAgh/dYODxrevglfm16EGplTR7S9WmK30/VA=
admin :
pbkdf2_sha256$320000$0DT0jMAKB1gI21vJlBGcuJ$QFldmvkyAgh/dYODxrevglfm16EGplTR7S9WmK30/VA=
admin :
pbkdf2_sha256$320000$0DT0jMAKB1gI21vJlBGcuJ$QFldmvkyAgh/dYODxrevglfm16EGplTR7S9WmK30/VA=
admin :
pbkdf2_sha256$320000$0DT0jMAKB1gI21vJlBGcuJ$QFldmvkyAgh/dYODxrevglfm16EGplTR7S9WmK30/VA=
admin :
pbkdf2_sha256$320000$0DT0jMAKB1gI21vJlBGcuJ$QFldmvkyAgh/dYODxrevglfm16EGplTR7S9WmK30/VA=
```

7.

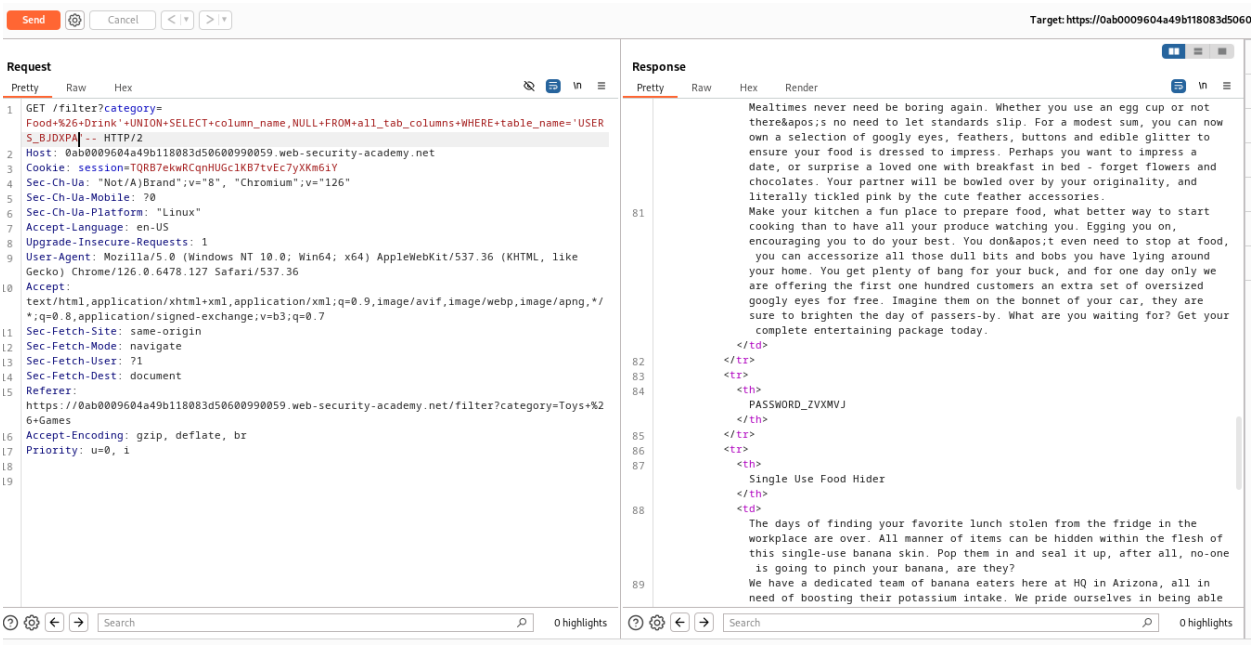
8.

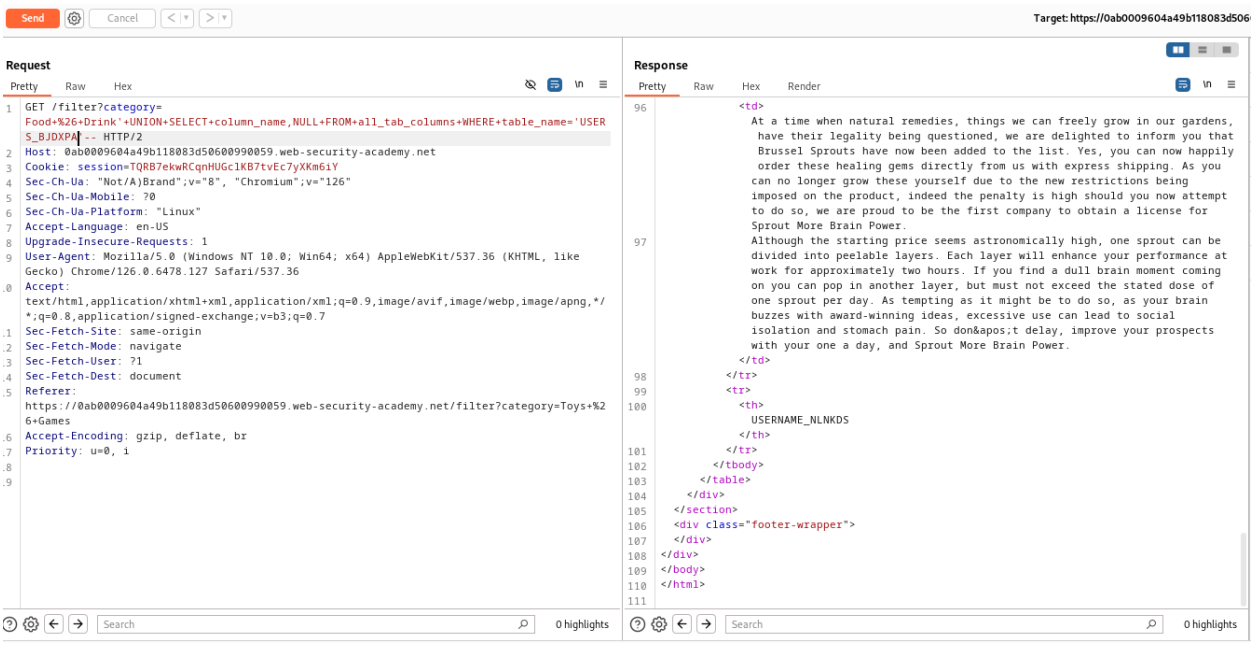
9. https://portswigger.net/web-security/sql-injection/examining-thedatabase/lab-listing-database-contents-oracle

- Truy cập vào trang web và chọn một mục bất kỳ trong khi sử dụng burpsuite để proxy. Sau khi tìm thấy thông điệp request chứa yêu cầu có tương tác với cơ sở dữ liệu, ta chuyển yêu cầu đó qua repeater và chèn thêm **'+UNION+SELECT+table_name,NULL+FROM+all_tables—**. Lệnh khiến trang web trả về tất cả tên bảng có trong cơ sở dữ liệu :

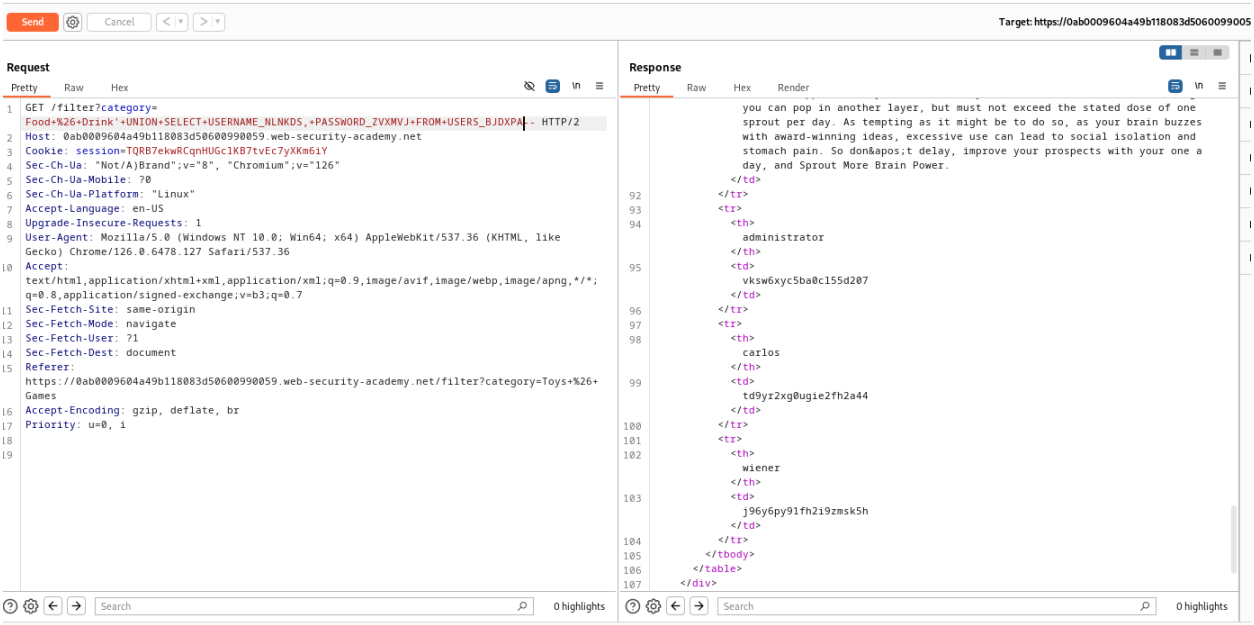


- Sau khi tìm được tên bảng chứa thông tin về dữ liệu người dùng, ta chèn lệnh **'+UNION+SELECT+column_name,NULL+FROM+all_tab_columns+WHERE+table_name='USERS_BJDXPA'—**để lấy tên các cột có trong bảng.





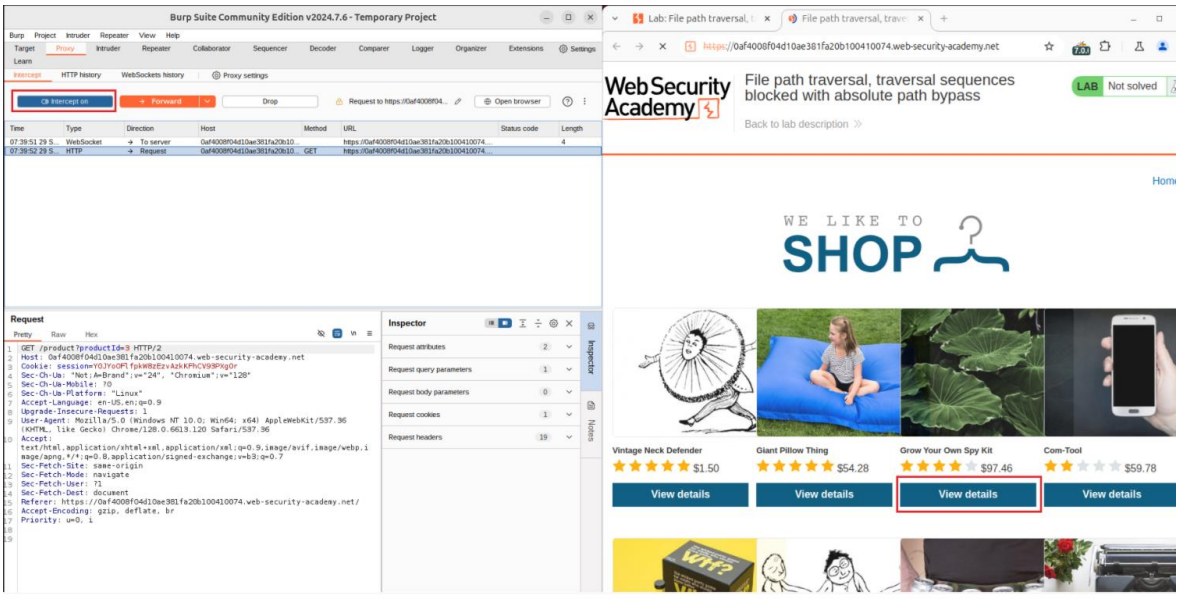
- Sau khi đã có tên các cột trong bảng. Sử dụng lệnh **'+UNION+SELECT+USERNAME_NLNKDS,+PASSWORD_ZVXMVJ+FROM+USERS_BJDXPA—**để lấy tên người dùng và mật khẩu tương ứng



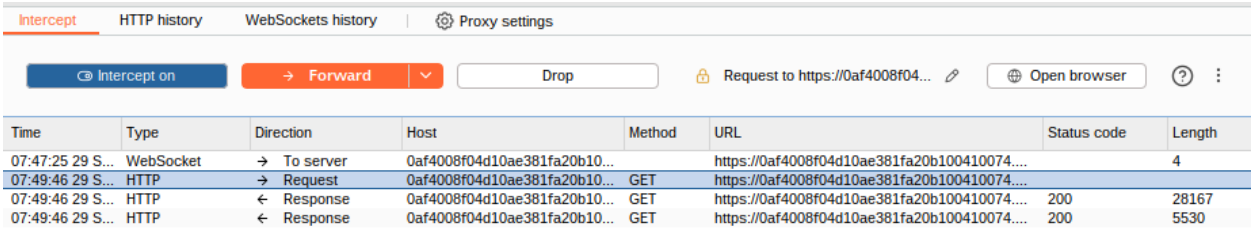
- Kết quả đăng nhập vào web sử dụng tên người dùng và mật khẩu tìm được:

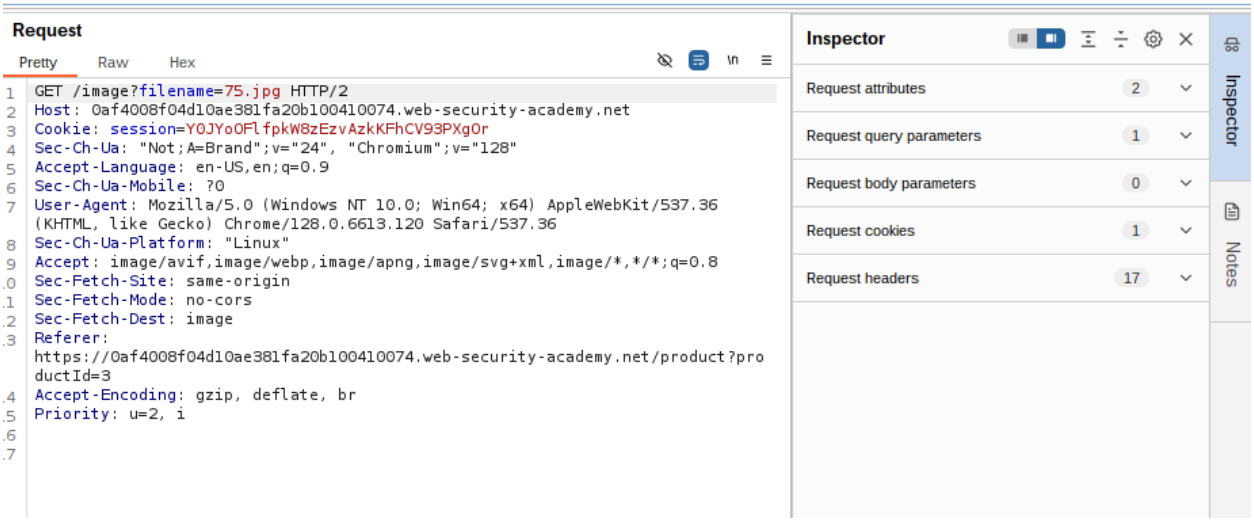
10.https://portswigger.net/web-security/file-path-traversal/lab-absolutepath-bypass

- Tiến hành **Access The Lab** xong chọn **intercept on** và **view details** 1 sản phẩm bất kì:

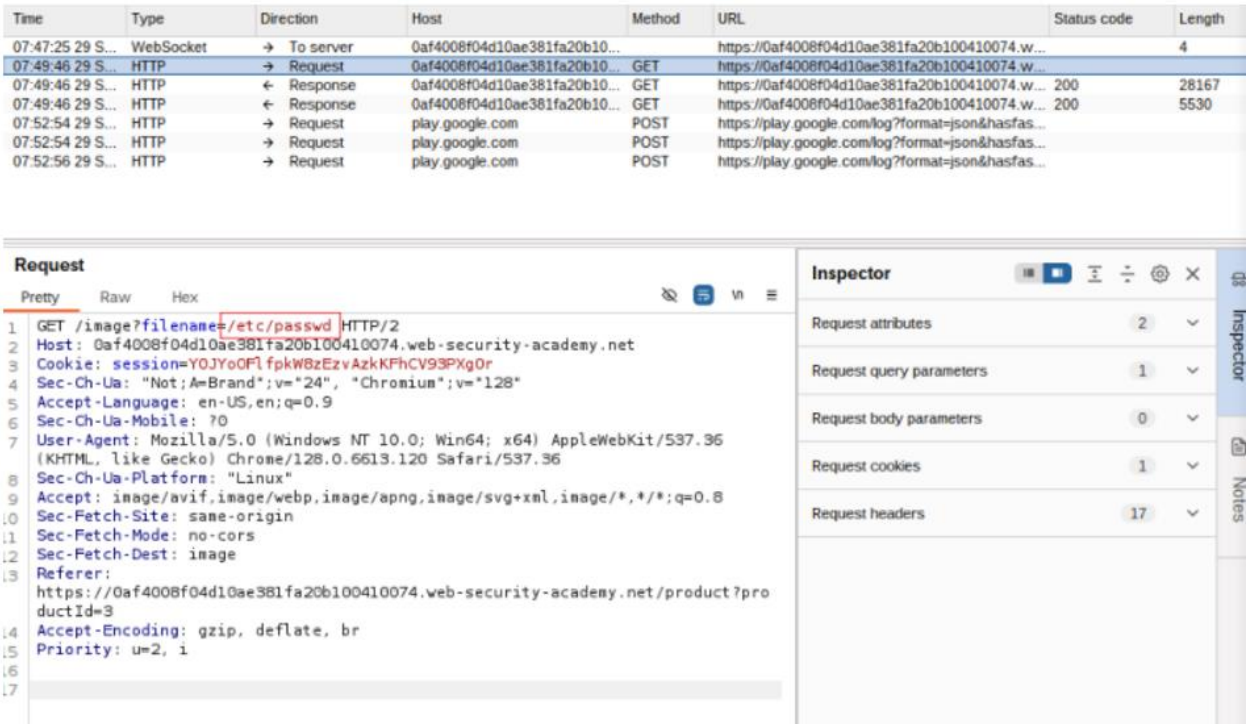


- Ta chọn **Forward** 2 lần sẽ hiển thị ra dòng **GET /image?filename=75.jpg HTTP/2 :**

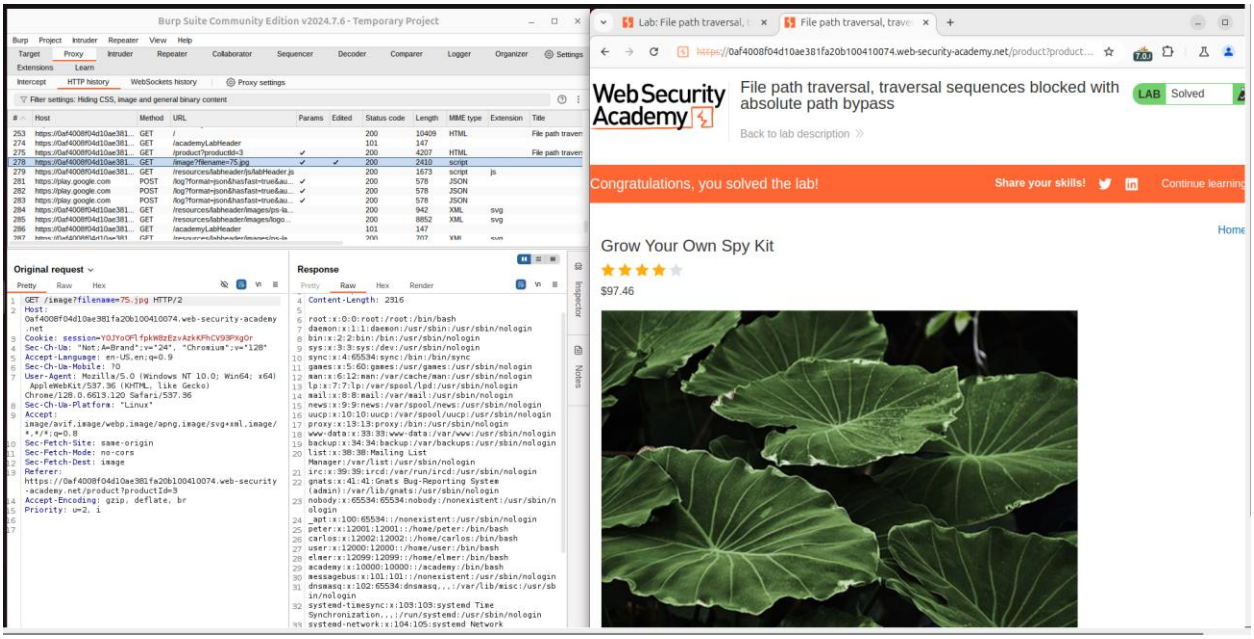




- Ta chỉnh sửa **filename=75.jpg** thành **filename=/etc/passwd**

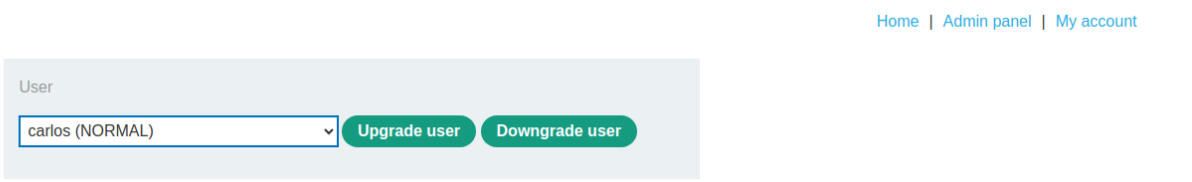


- Tiếp tục **Forward** xong chọn **intercep off** và load lại trang ta nhận được kết quả như hình:

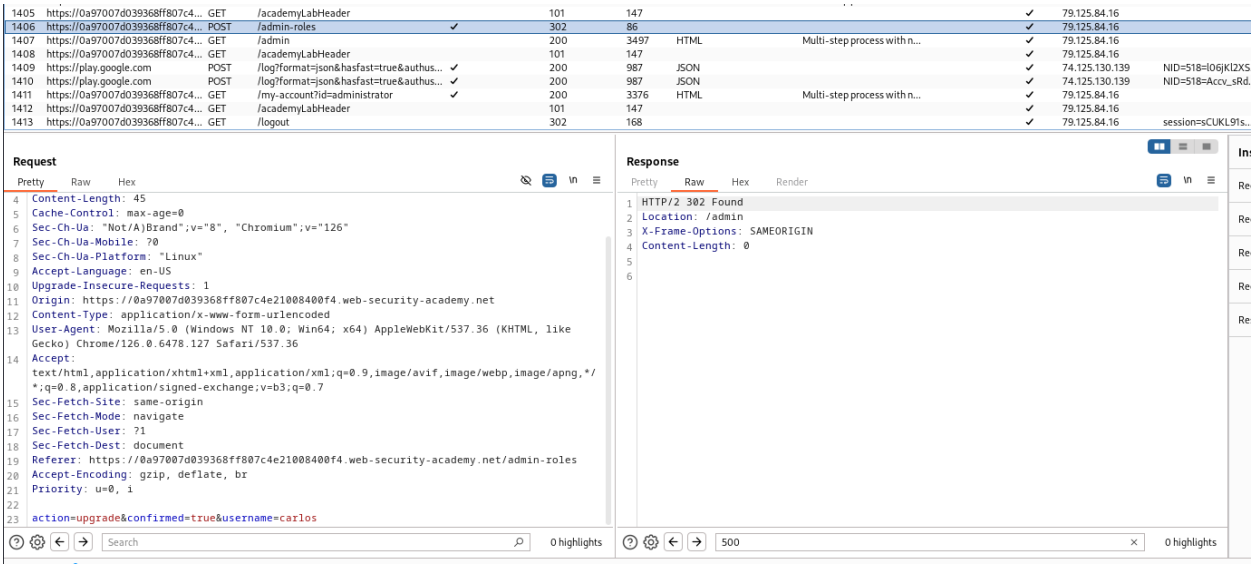


11.https://portswigger.net/web-security/access-control/lab-multi-stepprocess-with-no-access-control-on-one-step

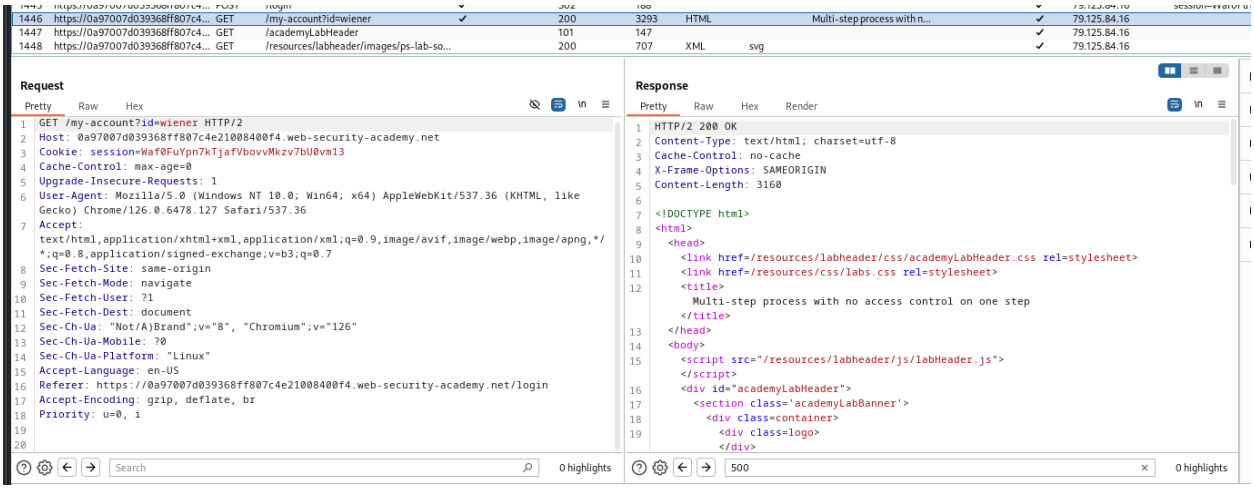
- Đầu tiên ta đăng nhập vào tài khoản admin, sử dụng tính năng nâng cấp quyền tài khoản để nâng quyền người dùng carlos lên admin:



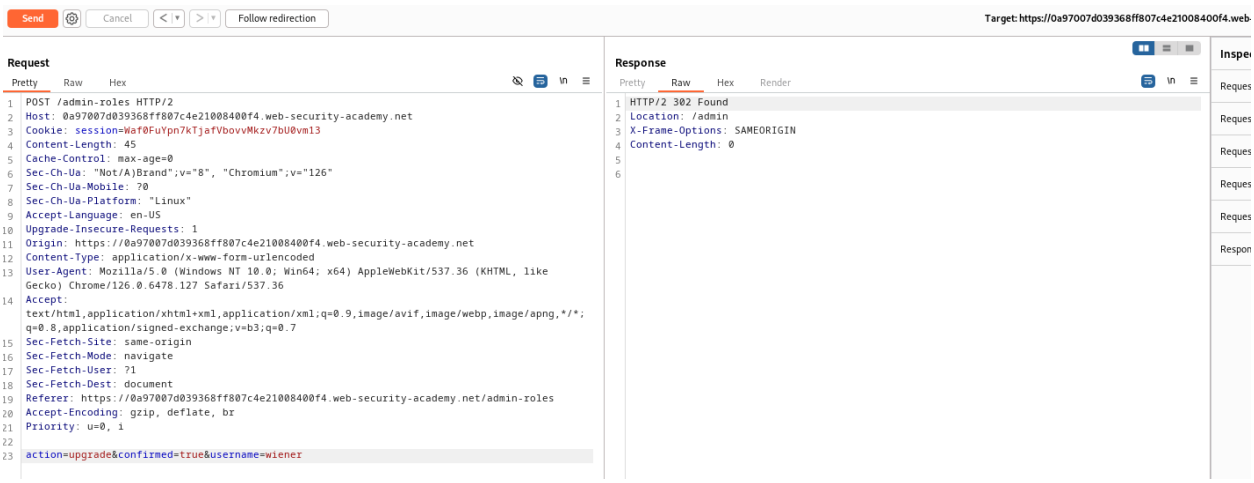
- Sử dụng burpsuite để proxy và bắt gói xác thức nâng cấp quyền người carlos, chuyển gói bắt được sang **repeater** để thực hiện tấn công replay :



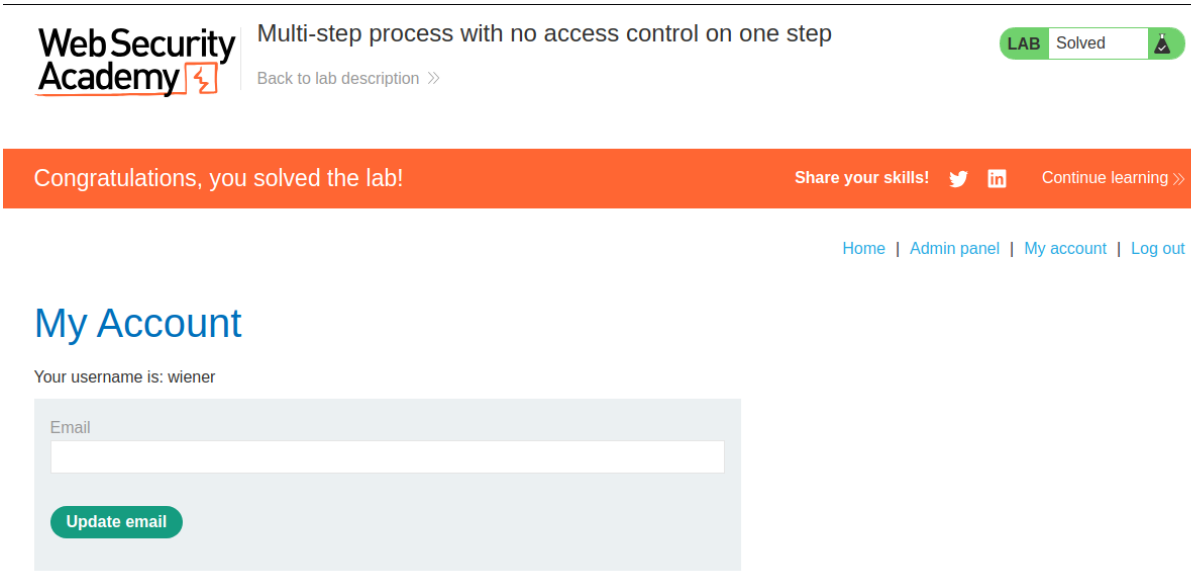
- Để thực hiện nâng cấp quyền người cho wiener khi không có tài khoản admin, ta sử dụng gói bắt được kết hợp với **cookie seesion** của phiên đăng nhập của **wiener** :



- Gửi gói tin bắt được với cookie session của **wiener** và tên người dùng được nâng cấp là **wiener** :



Kết quả thực hiện:


Web Security Academy
Multi-step process with no access control on one step
Back to lab description
LAB Solved
Congratulations, you solved the lab!
Share your skills!
Continue learning
Home | Admin panel | My account | Log out
My Account
Your username is: wiener
Email
Update email

12.