

1

Lab

PHỤC VỤ MỤC ĐÍCH GIÁO DỤC
FOR EDUCATIONAL PURPOSE ONLY

Tổng quan các lỗ hổng bảo mật web thường gặp

Thực hành môn Bảo mật web và ứng dụng

Tháng 9/2024

Lưu hành nội bộ

<Ng nghiêm cấm đăng tải trên internet dưới mọi hình thức>

A. TỔNG QUAN

1. Mục tiêu

- Giúp sinh viên có cái nhìn tổng quan hơn về các lỗ hổng web thường gặp thông qua tìm hiểu các lỗ hổng thuộc top 10 OSWAP 2021.
- Ở bài thực hành 1, sẽ tìm hiểu top từ 1 đến 5 của OSWAP. Sinh viên cần hiểu rõ cách thức lỗ hổng này xảy ra và cách tiếp cận để khai thác nó, đồng thời có giải pháp để khắc phục các lỗ hổng.

2. Thời gian thực hành

- Thực hành tại lớp: **5** tiết tại phòng thực hành.
- Hoàn thành báo cáo kết quả thực hành: tối đa **07** ngày.

B. CHUẨN BỊ MÔI TRƯỜNG

1. Docker

- Môi trường thực hành sử dụng docker được cung cấp

```
docker load -i owasp.tar
```

- Kiểm tra image được load vào.

```
docker images
```

- Chạy môi trường bài thực hành

```
docker run --rm -p 8000:8000 owasp
```

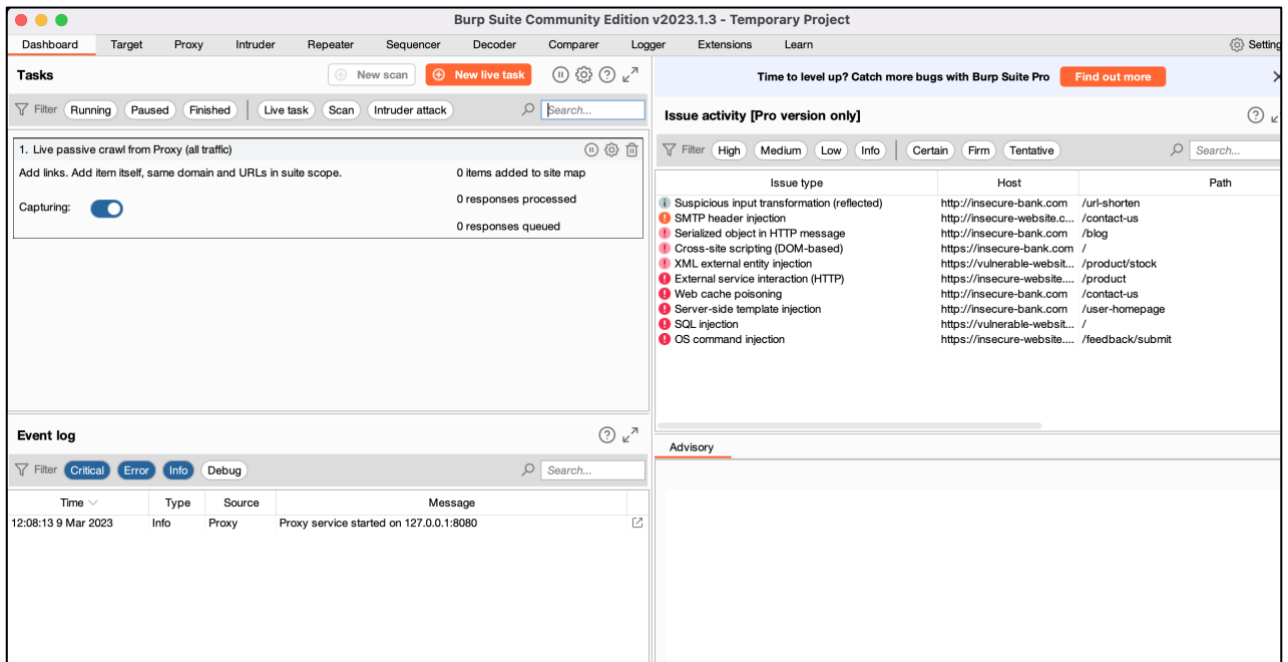
2. Phần mềm yêu cầu

- Phần mềm Burp Suite được cung cấp hoặc bất kỳ một phần mềm proxy nào mà sinh viên sử dụng quen thuộc.

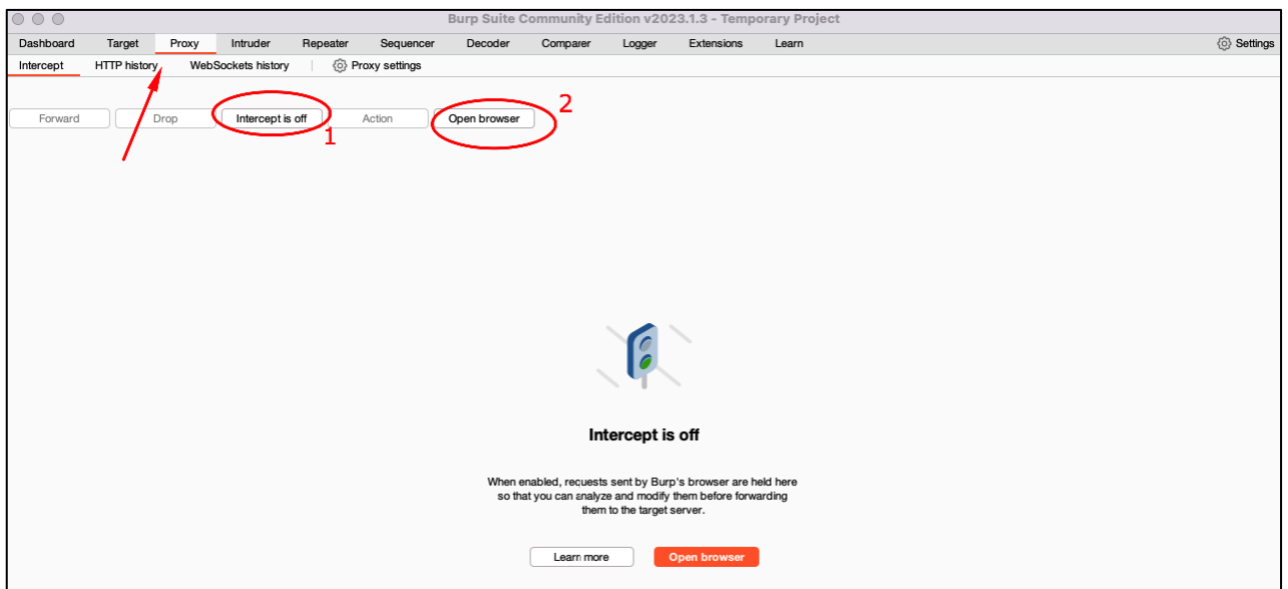
C. THỰC HÀNH

1. Hướng dẫn sử dụng Burp Suite

- Cài đặt phần mềm Burp Suite đã được cung cấp. Tùy theo hệ điều hành đang sử dụng mà chọn phần mềm phù hợp.
- Burp Suite là một phần mềm sử dụng để hỗ trợ khai thác các lỗ hổng bảo mật web, nó tích hợp nhiều công cụ khác nhau để thực hiện việc xâm nhập và kiểm tra web. Trong các bài thực hành bên dưới ta sẽ sử dụng 2 chức năng của burpsuite là proxy và repeater để tiến hành khai thác lỗ hổng trên các bài tập.
- Giao diện của burpsuite:



- Mở tab proxy để xem giao diện proxy của ứng dụng



***1: Intercept is off** là proxy được bật nhưng không chặn lại bất kỳ gói tin nào, lúc này proxy hoạt động để ghi nhận lịch sử câu request. Nếu **Intercept is on** là tiến hành chặn gói tin gửi lên server (tùy theo mục đích sử dụng, có thể chỉnh sửa gói tin hoặc đọc thông tin rồi forward nó đi)

***2:** Chức năng mở 1 browser để thực hiện kiểm tra chức năng web.

Chuyển qua tab nhỏ **HTTP history**. Ở đây sẽ lưu trữ danh sách các câu truy vấn đến server, bao gồm request và response của câu đó.

The screenshot shows the Burp Suite interface. The top bar includes tabs for Dashboard, Target, Proxy, Intruder, Repeater, Sequencer, Decoder, Comparer, Logger, Extensions, and Learn. The 'Proxy' tab is active, showing 'HTTP history' and 'WebSockets history'. The 'HTTP history' table lists various requests, with the last one (OWASP Pygoat) highlighted. Below the table, the 'Request' and 'Response' details are shown. The 'Request' tab is selected, displaying the raw request data. The 'Response' tab is also visible, showing the raw response data. The 'Inspector' panel on the right shows the request and response headers.

*1: Lịch sử HTTP được ghi nhận

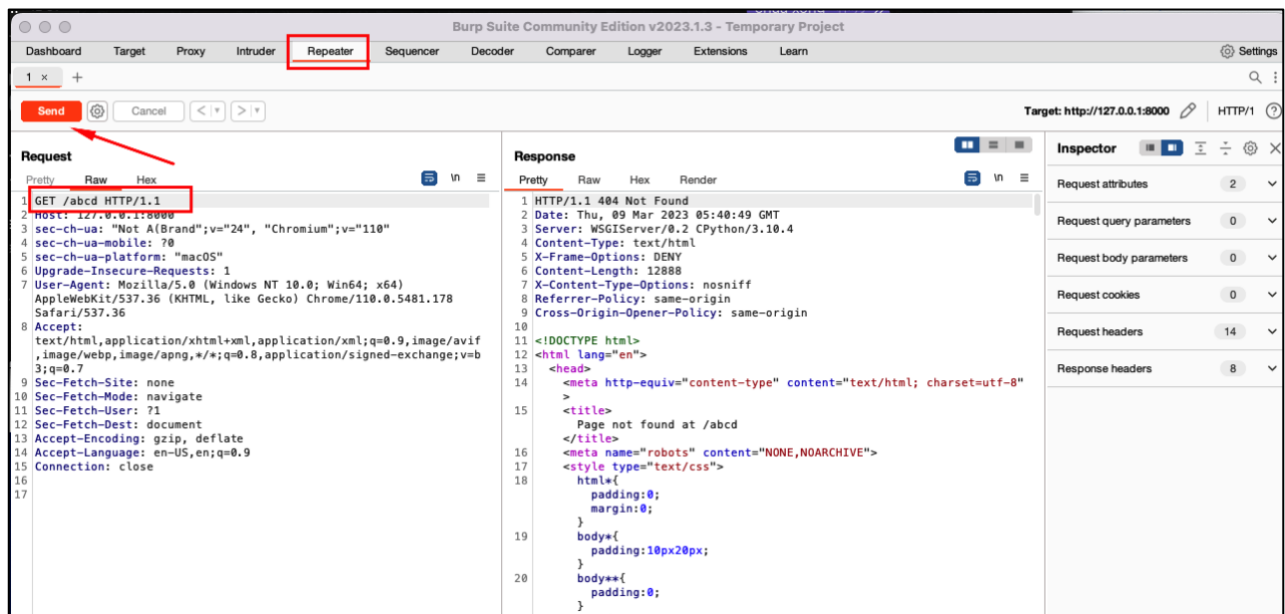
*2: Nội dung câu Request

*3: Response phản hồi được trả về

- Để thực hiện kiểm tra câu request nghi ngờ, click chuột phải vào câu đó, chọn repeater.

The screenshot shows the Burp Suite interface with the 'Proxy' tab active. The 'HTTP history' table is visible, and the last request (OWASP Pygoat) is selected. A right-click context menu is open over this request, showing options like 'Add to scope', 'Scan', 'Send to Intruder', 'Send to Repeater', 'Send to Sequencer', 'Send to Comparer (request)', and 'Send to Comparer (response)'. The 'Send to Repeater' option is highlighted. The 'Request' and 'Response' details are also visible below the table.

- Chuyển qua tab repeater, ở giao diện này sẽ có 2 phần chính là request và response. Tiến hành sửa request ở tab bên trái, chỉnh sửa các tham số như url, các parameter,... sau đó nhấn Send. Câu truy vấn sẽ được gửi lên máy chủ và kết quả câu request sẽ được hiển thị ở tab response.



2. Danh mục các lỗ hổng thuộc top 10 OSWAP 2021

Top 10 oswap là một tài liệu giành cho việc nâng cao nhận thức cho các nhà phát triển và nhà bảo mật ứng dụng web.

a) A01:2021-Broken Access Control

i. Mô tả

- Kiểm soát truy cập thực hiện việc thực thi chính sách sao cho người dùng không thể thực hiện các hành động khác ngoài các quyền họ được dự định. Lỗi phân quyền sẽ dẫn đến một số hậu quả như việc tiết lộ thông tin trái phép, sửa đổi hoặc phá huỷ tất cả dữ liệu hoặc chức năng kinh doanh ngoài giới hạn của người dùng đó được phép. Các lỗ hổng kiểm soát truy cập phổ biến:
- Vi phạm nguyên tắc đặc quyền tối thiểu hoặc từ chối theo mặc định, trong đó quyền truy cập chỉ được cấp cho các khả năng, vai trò hoặc người dùng cụ thể nhưng cuối cùng lại mở sẵn cho mọi người. Xem thêm: <https://www.checkpoint.com/cyber-hub/network-security/what-is-the-principle-of-least-privilege-polp/>
- Bỏ qua kiểm soát truy cập bằng cách sửa đổi URL, trạng thái nội bộ của ứng dụng, trang HTML hoặc sửa đổi các yêu cầu API
- Cho phép truy cập hoặc chỉnh sửa tài khoản người dùng khác bằng cách cung cấp định danh duy nhất của tài khoản đó (tham chiếu đối tượng trực tiếp không an toàn)
- Truy cập API với không có kiểm soát truy cập nào dành cho các phương thức POST, PUT và DELETE
- Nâng cao đặc quyền. Ví dụ như khi đăng nhập tài khoản người dùng thường thì hành động như người dùng quản trị hoặc khi không đăng nhập ứng dụng nhưng hành động như người dùng bình thường.

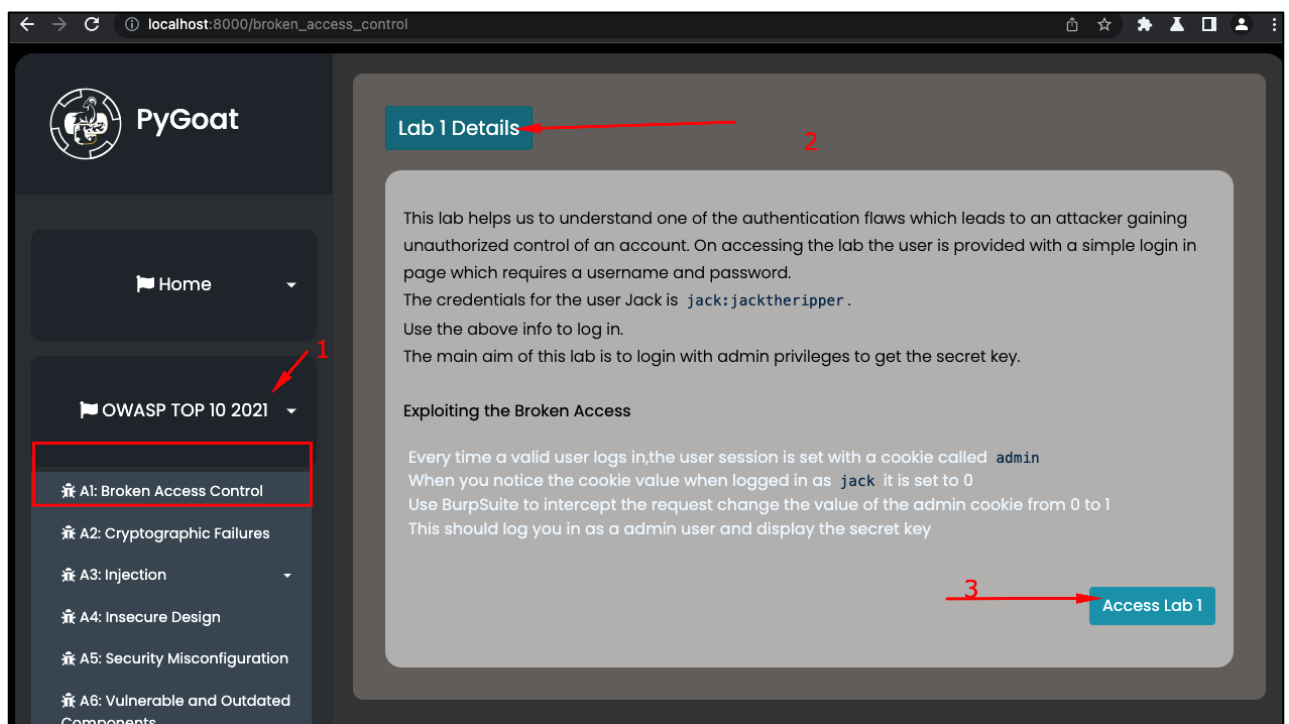
- Thao túng siêu dữ liệu (metadata), chẳng hạn như phát lại mã JSON Web Token (JWT), mã kiểm soát truy cập, giả mạo cookie hoặc các trường ẩn, để từ đó nâng cao đặc quyền hoặc lạm dụng tính không thể bị vô hiệu hoá trước hạn của JWT token.
- API có thể được truy cập từ nguồn trái phép hoặc không đáng tin cậy do cấu hình sai CORS. Tham khảo: <https://portswigger.net/web-security/cors>

b. Kịch bản tấn công thực hành:

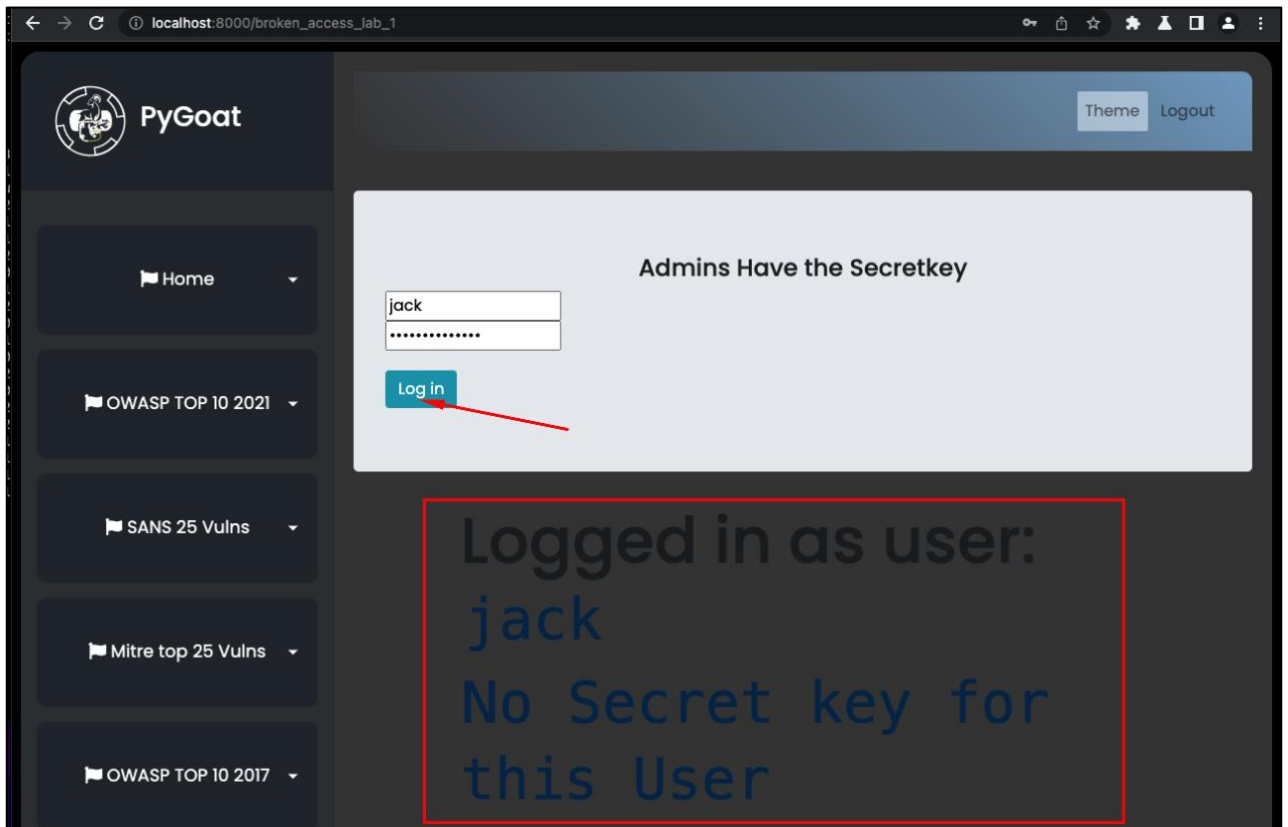
- Bài thực hành này giúp sinh viên hiểu được một trong những lỗi xác thực dẫn đến kẻ tấn công giành được quyền kiểm soát trái phép tài khoản. Khi truy cập bài thực hành, học viên được cung cấp một trang đăng nhập đơn giản với thông tin như sau:
- Thông tin đăng nhập của user Jack với tên đăng nhập là jack và mật khẩu là jacktheripper
- Yêu cầu của bài thực hành này là đăng nhập bằng quyền của quản trị viên để lấy khoá bí mật.

***Chú ý là các thao tác truy cập thông qua browser được Burpsuite cung cấp lúc ban đầu để có thể ghi nhận được lịch sử câu truy vấn.*

- Bước 1:** Truy cập bài thực hành tại **http://localhost:8000 => OSWAP TOP 10 2021 => A1: Broken Access Control => Lab 1 Details**



- Bước 2:** Đăng nhập vào trang web với tài khoản và mật khẩu được cung cấp của user Jack



- **Bước 3:** Trở lại giao diện HTTP history của Burpsuite để kiểm tra lịch sử câu truy vấn để hiểu rõ logic của ứng dụng.

The screenshot displays the Burp Suite interface. The top menu bar includes Dashboard, Target, Proxy, Intruder, Repeater, Sequencer, Decoder, Comparer, Logger, Extensions, and Learn. The 'Proxy' tab is active, showing a list of intercepted HTTP requests. The table below summarizes the visible requests:

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	IP	
25	http://localhost:8000	POST	/broken_access_lab_1		✓	200	12905	HTML		Broken Access Control			127.0.0.1	adm
24	http://localhost:8000	GET	/broken_access_lab_1			200	12754	HTML		Broken Access Control			127.0.0.1	
23	http://localhost:8000	GET	/broken_access_control			200	18417	HTML		Broken Access Control			127.0.0.1	
22	https://passwordleakcheck-pa...	POST	/v1/leaks:lookupSingle		✓	400	523	script				✓	74.125.24.95	
20	http://localhost:8000	GET	/			200	12184	HTML		OWASP Pygoat			127.0.0.1	
19	http://localhost:8000	POST	/login/		✓	302	736	HTML					127.0.0.1	csrf
18	http://localhost:8000	GET	/favicon.ico			404	13186	HTML	ico	Page not found at /favico...			127.0.0.1	
17	https://fonts.gstatic.com	GET	/s/poppins/v20/pxiByp8kv8JHgFvRLD...			200	8654		woff2			✓	142.251.12.94	
16	https://fonts.gstatic.com	GET	/s/poppins/v20/pxiEyp8kv8JHgFvRJL...			200	8698		woff2			✓	142.251.12.94	
15	https://fonts.gstatic.com	GET	/s/poppins/v20/pxiByp8kv8JHgFvRLG...			200	8562		woff2			✓	142.251.12.94	
12	https://code.jquery.com	GET	/jquery-3.3.1.slim.min.js			200	70348	script	js			✓	69.16.175.10	
11	https://cdnjs.cloudflare.com	GET	/ajax/libs/popper.js/1.14.0/umd/popper...			200	21497	script	js			✓	104.17.25.14	
9	https://stackpath.bootstrapcdn.com	GET	/bootstrap/4.1.3/js/bootstrap.min.js			200	51583	script	js			✓	104.18.10.207	
8	https://use.fontawesome.com	GET	/releases/v5.0.13/js/fontawesome.js			200	28522	script	js			✓	172.64.132.15	
7	https://use.fontawesome.com	GET	/releases/v5.0.13/js/solid.js			200	352144	script	js			✓	172.64.132.15	
6	http://localhost:8000	GET	/static/Lab/ssrl.js			200	2915	script	js				127.0.0.1	
5	http://localhost:8000	GET	/static/Lab/xss.js			200	1378	script	js				127.0.0.1	
2	http://localhost:8000	GET	/login/			200	13754	HTML		OWASP Pygoat			127.0.0.1	csrf
1	http://localhost:8000	GET	/			302	317	HTML					127.0.0.1	

The bottom section shows the details of the selected POST request (ID 25) to /broken_access_lab_1. The request body is as follows:

```

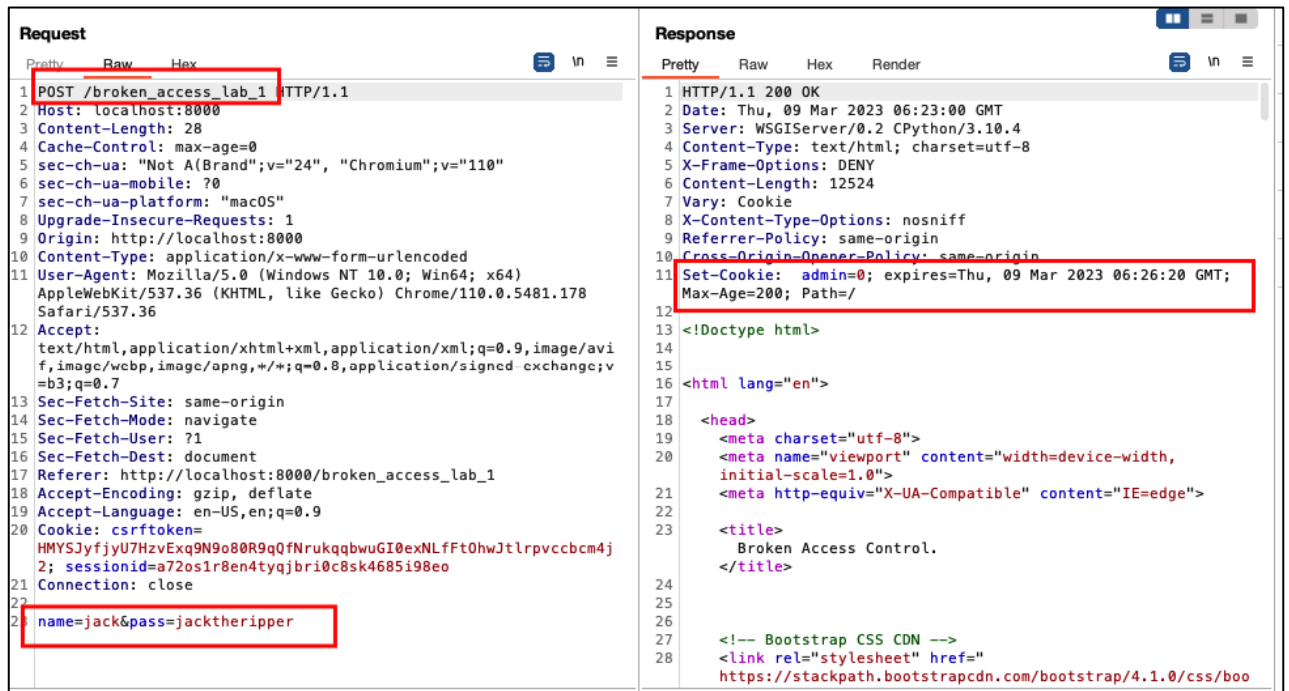
1 POST /broken_access_lab_1 HTTP/1.1
2 Host: localhost:8000
3 Content-Length: 28
4 Cache-Control: max-age=0
5 sec-ch-ua: "Not A(Brand";v="24", "Chromium";v="110"
6 sec-ch-ua-mobile: ?0
7 sec-ch-ua-platform: "macOS"
8 Upgrade-Insecure-Requests: 1
9 Origin: http://localhost:8000
10 Content-Type: application/x-www-form-urlencoded
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.5481.178
Safari/537.36
12 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avi
f,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v
=b3;q=0.7
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-User: ?1
16 Sec-Fetch-Dest: document
17 Referer: http://localhost:8000/broken_access_lab_1
18 Accept-Encoding: gzip, deflate
19 Accept-Language: en-US,en;q=0.9
20 Cookie: csrfToken=
HMY5JyfyU7HzvExq9N9o80R9qQfNrukqbwuGI0exNLff0hwJtlrpvccbcm4j
2; sessionId=a72qs1r8en4tyqjbr10c8sk4685198oe
21 Connection: close
22
23 name=jack&pass=jacktheripper
    
```

The response (ID 250) is an HTML page titled "Broken Access Control." It contains a login form with the following structure:

```

<div class="jumbotron">
  <h4 style="text-align:center">
    Admins Have the Secretkey
  </h4>
  <div class="login">
    <form method="post" action="/broken_access_lab_1">
      <input id="input" type="text" name="name"
        placeholder="User Name">
      <br>
      <input id="input" type="password" name="pass"
        placeholder="Password">
      <br>
      <button style="margin-top:20px" class="btn
        btn-info" type="submit">
        Log in
      </button>
    </form>
  </div>
</div>
    
```

- Tại đây ta thấy được danh sách rất dài các câu request, các bạn có thể đọc hết toàn bộ các câu để hiểu rõ hơn client và server trao đổi với nhau những gì, #id có số nhỏ hơn là những câu truy vấn được gửi trước.
- Sau khi kiểm tra, ta có thể chú ý đến câu request cuối cùng được ghi nhận.

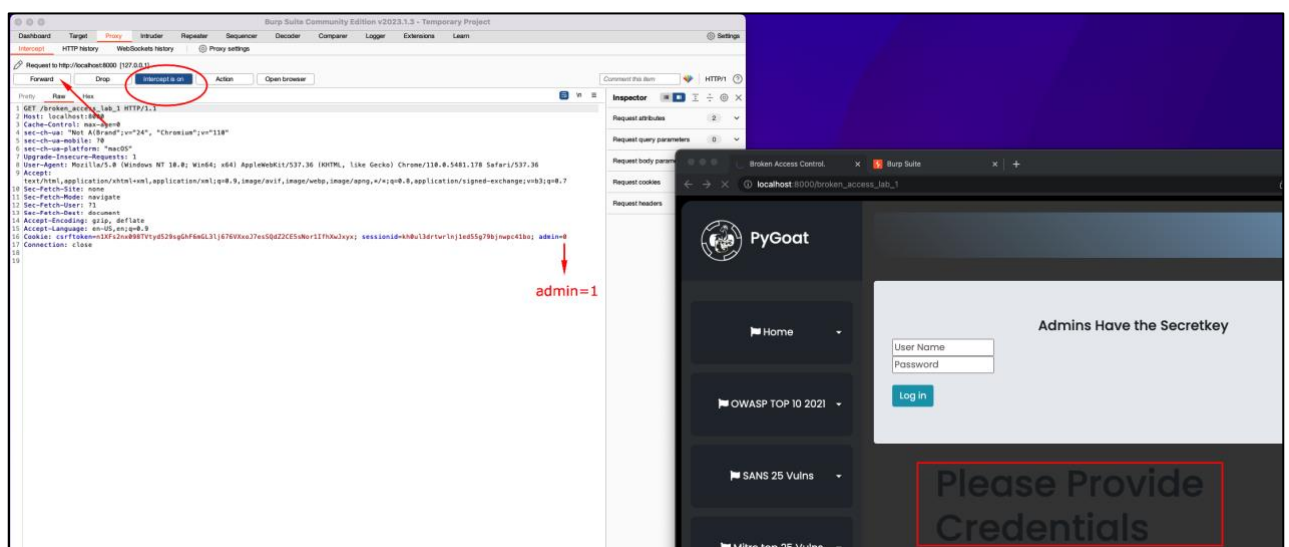


Chậm lại và suy nghĩ 1: Câu truy vấn tại đây đang làm gì?

- Như vậy chúng ta đã hiểu rõ được logic của ứng dụng, lúc này để có thể đọc được giá trị của Secretkey của admin, ta có thể nghĩ đến việc thay admin=1 trong cookie để kiểm tra. Chú ý là cookie được set với thời gian expire rất ngắn, do đó có thể đăng nhập lại nếu không thấy cookie admin=0
- Có 2 cách để thực hiện việc kiểm tra:

Cách 1: Chặn gói tin tại tab Intercept và sửa đổi trực tiếp tại đó.

- Thử bằng gói GET với admin=1 trong cookie đến máy chủ. Lúc này ta bật **Intercept is on** và truy cập đến `http://localhost:8000/broken_access_lab_1`



- Sau khi chỉnh sửa xong bấm forward để gói tin đi qua. Kiểm tra kết quả tại browser và nội dung trả về trong tab HTTP proxy.

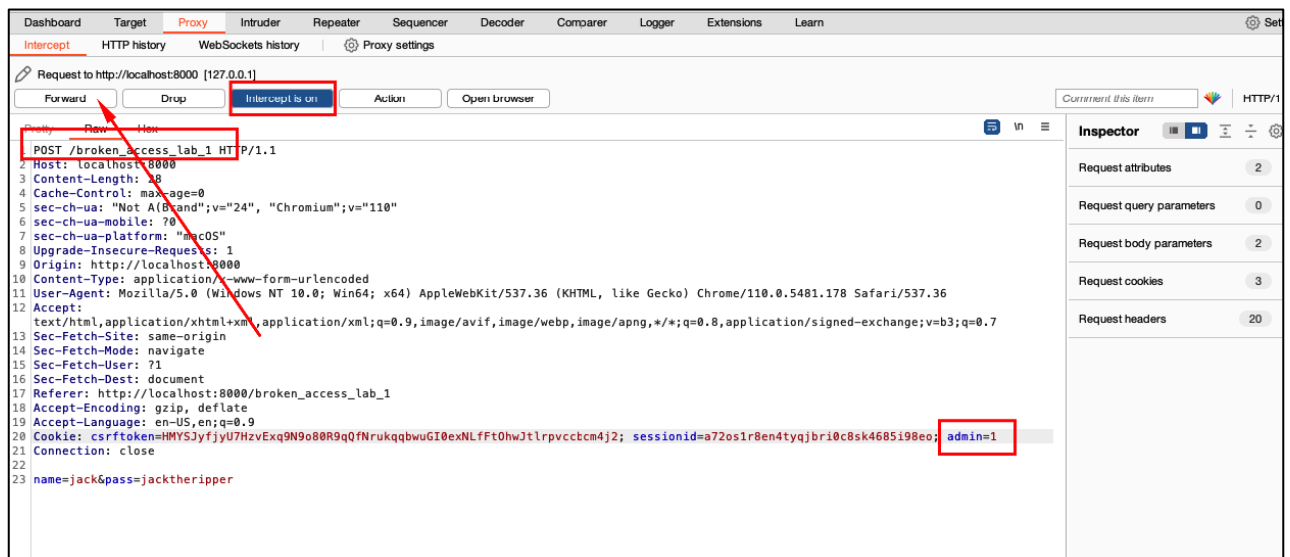
The screenshot displays the Burp Suite interface. At the top, a table lists intercepted HTTP requests. The 'Original request' tab is selected, showing the raw HTTP request details. The 'Response' tab is also visible, showing the raw HTTP response details. The 'Inspector' tab is on the right, showing the request and response attributes, cookies, headers, and body.

- Có thể thấy tại đây nội dung gói tin đã được ghi nhận là được chỉnh sửa, có thể xem nội dung chỉnh sửa bằng cách chỉnh từ **Original request** thành **Edited request**.
- Kết quả trả về trên browser **Please Provide Credentials**.

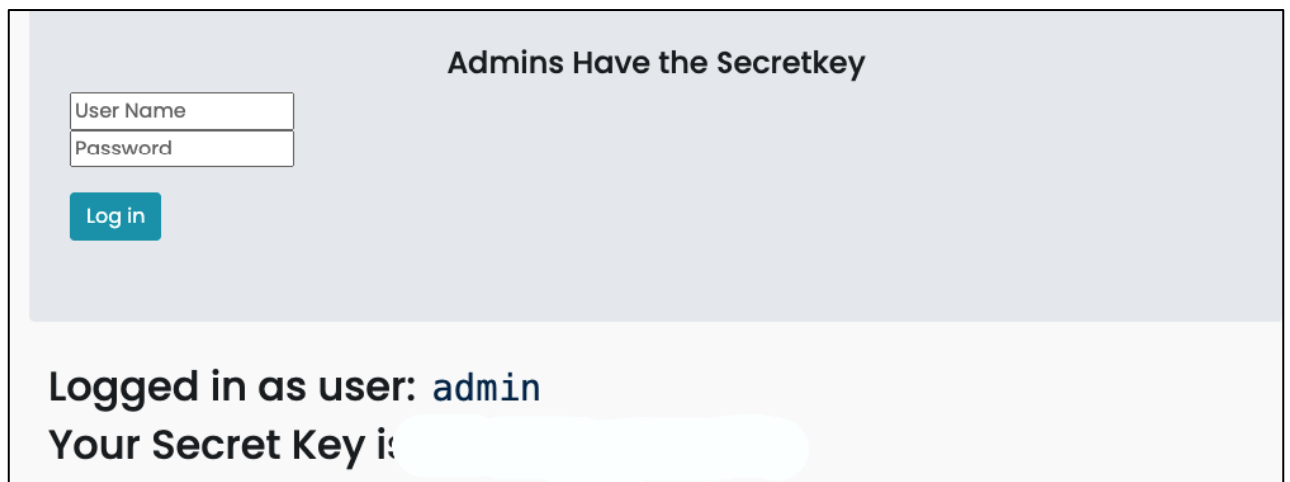
Admins Have the Secretkey

Please Provide Credentials

- Logic ứng dụng này là mỗi khi truy cập trang này đều phải cung cấp tài khoản và mật khẩu, do đó ta cung cấp tài khoản và mật khẩu tiến hành login lại.
- Cũng tiến hành chặn ở giữa bằng **Intercept is on** và thay đổi cookie admin=1 và login. Sau đó bấm forward để gói tin gửi lên máy chủ.



- Xem kết quả tại browser:



- Hoặc tại tab response:



Cách 2: Sử dụng repeater.

Bài tập 1: Sử dụng repeater để thực hành bài tập trên.

Bài tập 2: Báo cáo lỗ hổng đang được thực hành. Có thể sử dụng format theo mẫu sau:

#Tiêu đề: Tiêu đề rõ ràng và ngắn gọn bao gồm loại lỗ hổng và tài sản bị ảnh hưởng.

#Mô tả lỗ hổng: Lỗ hổng này là gì? Trong các bước rõ ràng, làm thế nào để bạn tái tạo nó?

Tóm tắt:

Các bước để thực hiện lại và bằng chứng:

1. bước 1

2. bước 2

3. bước 3

Tài liệu hỗ trợ và tham khảo:

* Liệt kê bất kỳ tài liệu bổ sung nào (ví dụ: ảnh chụp màn hình, nhật ký, v.v.)

#Mức độ ảnh hưởng của lỗ hổng: Tác động bảo mật nào mà kẻ tấn công có thể đạt được?

#Khuyến cáo khắc phục: Làm thế nào để vá lỗ hổng này?

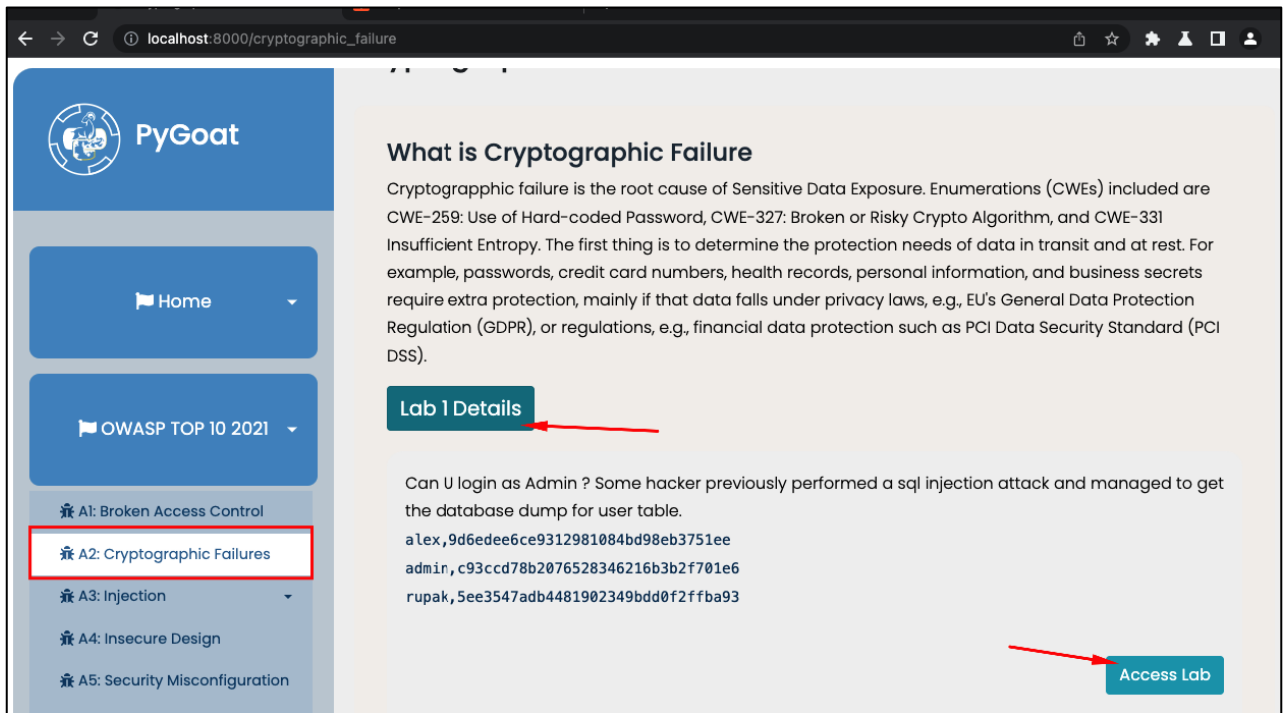
b) A02:2021 – Cryptographic Failures

c) a. Mô tả:

- Điều đầu tiên phải xác định nhu cầu bảo vệ dữ liệu khi ở trạng thái truyền và trạng thái nghỉ. Ví dụ như mật khẩu, thông tin số thẻ tín dụng, bản ghi sức khỏe, thông tin cá nhân và bí mật doanh nghiệp cần được bảo vệ. Tham khảo thêm các luật riêng tư về dữ liệu: Quy định bảo vệ dữ liệu chung (GDPR) của EU, hoặc các quy định: ví dụ như bảo vệ dữ liệu tài chính chẳng hạn như tiêu chuẩn bảo mật dữ liệu PCI (PCI DSS).
- Lỗi mật mã là nguyên nhân chính dẫn đến lỗi Tiết lộ thông tin dữ liệu nhạy cảm. Các CWE có thể tham khảo như CWE-259: Sử dụng mật khẩu mã hoá cứng, CWE-327: Thuật toán mã hoá bị hỏng hoặc rủi ro, CWE-331...

b. Kịch bản tấn công thực hành

- Kịch bản thực hành này sẽ cho thấy điểm yếu trong mã hoá dữ liệu có thể gây ảnh hưởng đến ứng dụng như thế nào. Bối cảnh xảy ra khi kẻ tấn công trước đó đã thực thi thành công lỗi SQL injection và lấy được bảng thông tin đăng nhập của người dùng với username và 1 đoạn chuỗi ký tự. Truy cập bài thực hành tại: http://localhost:8000/cryptographic_failure/lab

**Chậm lại và suy nghĩ 2: Đoạn chuỗi ký tự trên là gì?**

- Hiện nay các thuật toán mã hoá đã trở nên phức tạp hơn trước, các nhà nghiên cứu mật mã học liên tục tìm ra các điểm yếu trong thuật toán và giải mã nó. Cùng với sự phát triển này thì các thuật toán mật mã cũ cũng phơi bày điểm yếu của nó. Như trong bài tập này, đoạn mã hash MD5 đã bị bruteforce và dễ dàng tìm kiếm thông qua các trang web online trên mạng. Ở đây chúng ta sẽ thử trang <https://www.md5online.org/md5-decrypt.html>

MD5 Decryption

Enter your MD5 hash below and cross your fingers :

c93ccd78b2076528346216b3b2f701e6

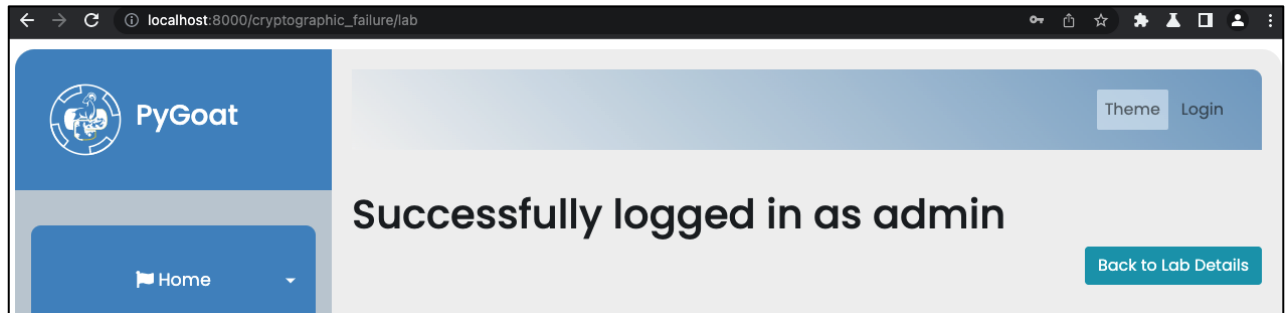
☒ Quick search (free) ☐ In-depth search (1 credit) 

Decrypt

Found : XXXXXXXXXX

(hash = c93ccd78b2076528346216b3b2f701e6)

- Dùng mật khẩu tìm được tiến hành đăng nhập vào bài tập thực hành để kiểm tra.



Bài tập 3: Báo cáo lỗ hổng đang được thực hành. Có thể sử dụng format theo mẫu sau:

#Tiêu đề: Tiêu đề rõ ràng và ngắn gọn bao gồm loại lỗ hổng và tài sản bị ảnh hưởng.

#Mô tả lỗ hổng: Lỗ hổng này là gì? Trong các bước rõ ràng, làm thế nào để bạn tái tạo nó?

Tóm tắt:

Các bước để thực hiện lại và bằng chứng:

1. bước 1

2. bước 2

3. bước 3

Tài liệu hỗ trợ và tham khảo:

* Liệt kê bất kỳ tài liệu bổ sung nào (ví dụ: ảnh chụp màn hình, nhật ký, v.v.)

#Mức độ ảnh hưởng của lỗ hổng: Tác động bảo mật nào mà kẻ tấn công có thể đạt được?

#Khuyến cáo khắc phục: Làm thế nào để vá lỗ hổng này?

d) A03:2021 – Injection

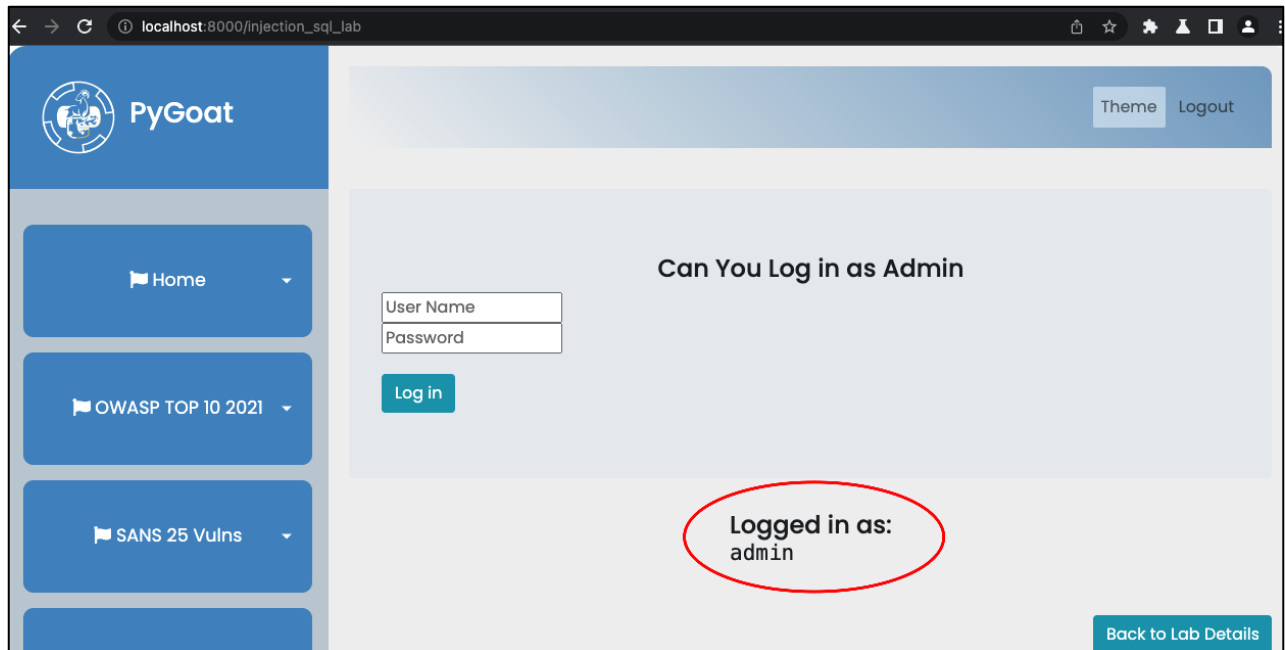
a. Mô tả

- Một ứng dụng có khả năng bị tấn công tiêm khi dữ liệu người dùng cung cấp không được xác thực hoặc lọc bởi ứng dụng.
- Tùy theo kiểu tiêm thì mã tấn công có thể khác nhau và tùy chỉnh theo cấu trúc của ứng dụng muốn tấn công, một số kiểu tiêm phổ biến như SQL injection, NoSQL injection, OS command injection, Object Relational Mapping(ORM) injection, LDAP injection, và Expression Language (EL) injection hoặc Object Graph Navigation Library (OGNL) injection.

b. Kịch bản tấn công thực hành

- Kịch bản thực hành này giúp cho học viên khai thác được lỗ hổng tiêm sql thông thường, gây ra khi thiếu xác thực đầu vào và truyền trực tiếp đầu vào vào câu truy vấn.

- Người dùng sẽ truy cập bài thực hành với trang đăng nhập được cung cấp. Người dùng có thể thử đăng nhập với tài khoản admin. Lỗi tiêm SQL có thể được nhận ra thông qua một vài thủ thuật như tiêm một kí tự ' vào bất kỳ trường nào. Nếu kết quả là một lỗi SQL thì lỗi tiêm SQL có thể đã xảy ra.
- Truy cập bài thực hành tại: http://localhost:8000/injection_sql_lab



Chậm lại và suy nghĩ 3: Nếu trang web thực hành bị lỗi tiêm SQL thì khai thác như thế nào?

Bài tập 4: Báo cáo lỗ hổng đang được thực hành. Có thể sử dụng format theo mẫu sau:

#Tiêu đề: Tiêu đề rõ ràng và ngắn gọn bao gồm loại lỗ hổng và tài sản bị ảnh hưởng.

#Mô tả lỗ hổng: Lỗ hổng này là gì? Trong các bước rõ ràng, làm thế nào để bạn tái tạo nó?.

Tóm tắt:

Các bước để thực hiện lại và bằng chứng:

1. bước 1

2. bước 2

3. bước 3

Tài liệu hỗ trợ và tham khảo:

* Liệt kê bất kỳ tài liệu bổ sung nào (ví dụ: ảnh chụp màn hình, nhật ký, v.v.)

#Mức độ ảnh hưởng của lỗ hổng: Tác động bảo mật nào mà kẻ tấn công có thể đạt được?

#Khuyến cáo khắc phục: Làm thế nào để vá lỗ hổng này?

e) A04:2021 – Insecure Design

a. Mô tả

- Thiết kế không an toàn là một thể loại rộng đại diện cho các điểm yếu khác nhau, được thể hiện là thiết kế kiểm soát thiếu hoặc không hiệu quả. Thiết kế không an toàn không phải là nguồn của tất cả 10 loại rủi ro hàng đầu khác. Có một sự khác biệt giữa thiết kế không an toàn và thực hiện không an toàn. Việc phân biệt giữa lỗ hổng thiết kế và khiếm khuyết thực hiện vì một vài lý do, trong đó chúng có nguyên nhân gốc rễ gây ra lỗi và cách khắc phục khác nhau. Một thiết kế an toàn vẫn có các khiếm khuyết thực hiện, dẫn đến các lỗ hổng có thể được khai thác. Một thiết kế không an toàn thì khác, nó không thể sửa chữa bằng việc triển khai hoàn hảo vì theo thiết kế của nó, các kiểm soát bảo mật cần thiết không được tạo để chống lại các cuộc tấn công cụ thể.

b. Kịch bản tấn công thực hành

- Kịch bản này mô tả tình huống người dùng có thể có được 5 vé xem phim miễn phí với điều kiện là phải chờ đợi cho đến khi tất cả các vé được bán hết. Đối với tình huống này, chúng ta sẽ tận dụng lỗi thiết kế không an toàn để lấy toàn bộ vé xem phim về.
- Thực hành tại: http://localhost:8000/insecure-design_lab
- Nhập số vé muốn lấy và nhấn claim để lấy, nhấn claim lần nữa để kiểm tra còn bao nhiêu vé còn lại.

You can have atmost 5 tickits

Claim Upto 5 Free Tickits

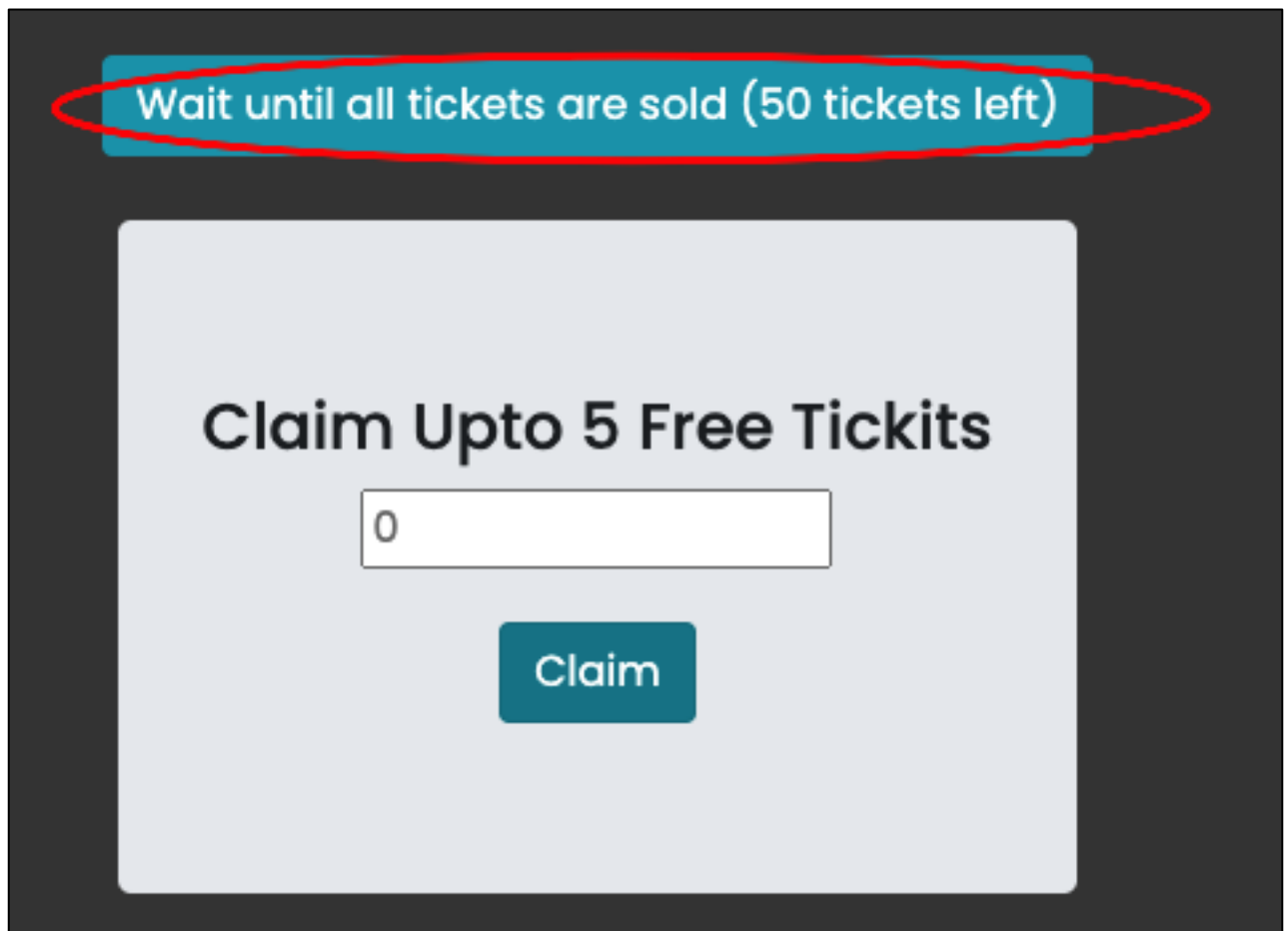
Claim

Watch Movie

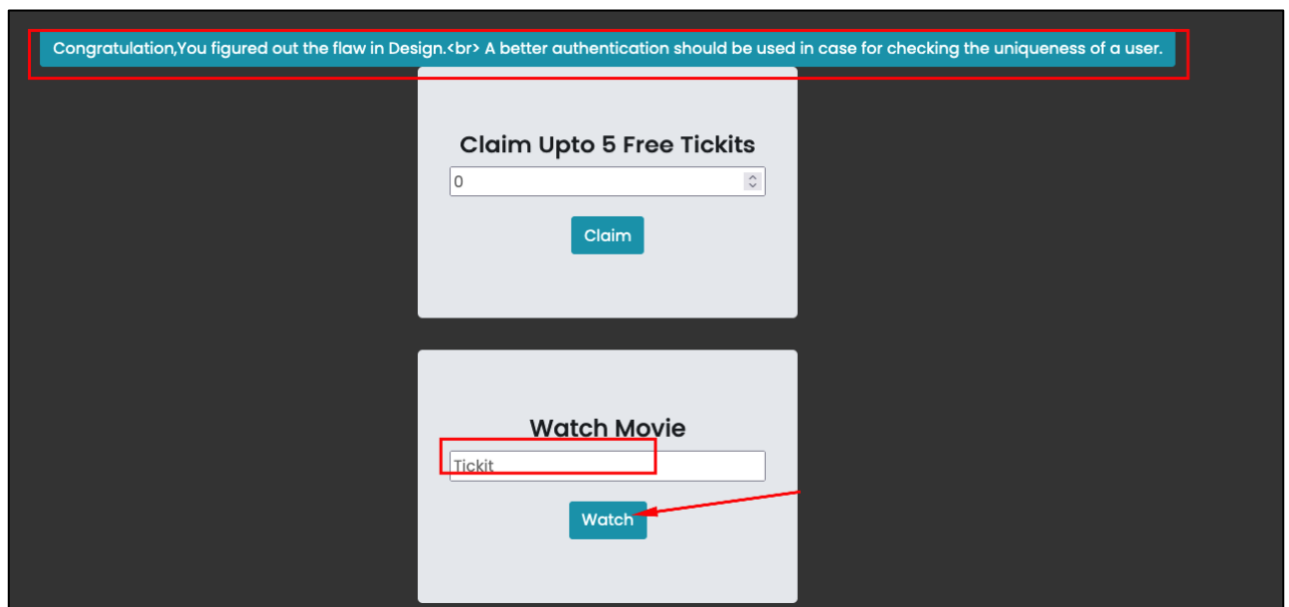
Watch

My Tickets

xrjzsSJZEt
ZUqnyEbXcn
PEilwtkjwe
WpVhthWsBf
mwBSUUqaEG



Sau khi toàn bộ vé đã bán hết ai có vé có thể bấm vào watch movie để xem phim



Chậm lại và suy nghĩ 4: Lỗi thiết kế không an toàn nằm ở đâu? Chú ý là web được tạo nhiều tài khoản.

Bài tập 5: Báo cáo lỗ hổng đang được thực hành. Có thể sử dụng format theo mẫu sau:

#Tiêu đề: Tiêu đề rõ ràng và ngắn gọn bao gồm loại lỗ hổng và tài sản bị ảnh hưởng.

#Mô tả lỗ hổng: Lỗ hổng này là gì? Trong các bước rõ ràng, làm thế nào để bạn tái tạo nó?.

Tóm tắt:

Các bước để thực hiện lại và bằng chứng:

1. bước 1

2. bước 2

3. bước 3

Tài liệu hỗ trợ và tham khảo:

* Liệt kê bất kỳ tài liệu bổ sung nào (ví dụ: ảnh chụp màn hình, nhật ký, v.v.)

#Mức độ ảnh hưởng của lỗ hổng: Tác động bảo mật nào mà kẻ tấn công có thể đạt được?

#Khuyến cáo khắc phục: Làm thế nào để vá lỗ hổng này?

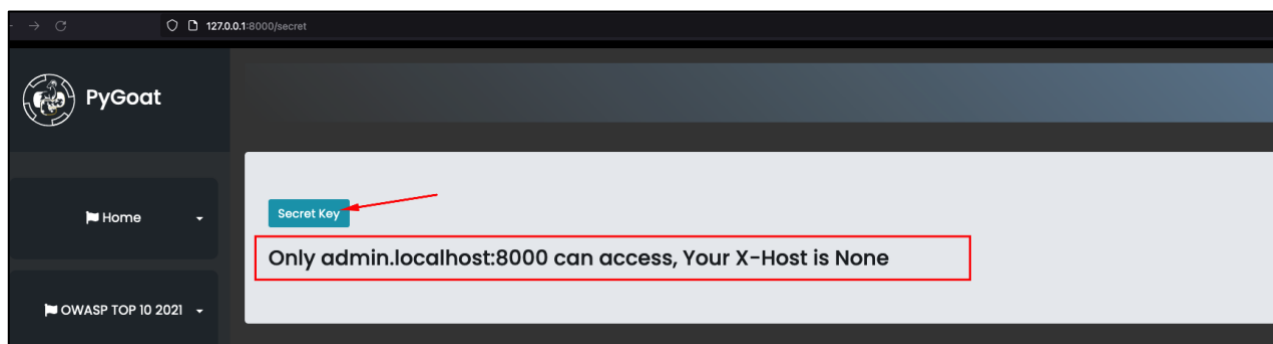
f) A05:2021 – Security Misconfiguration

a. Mô tả

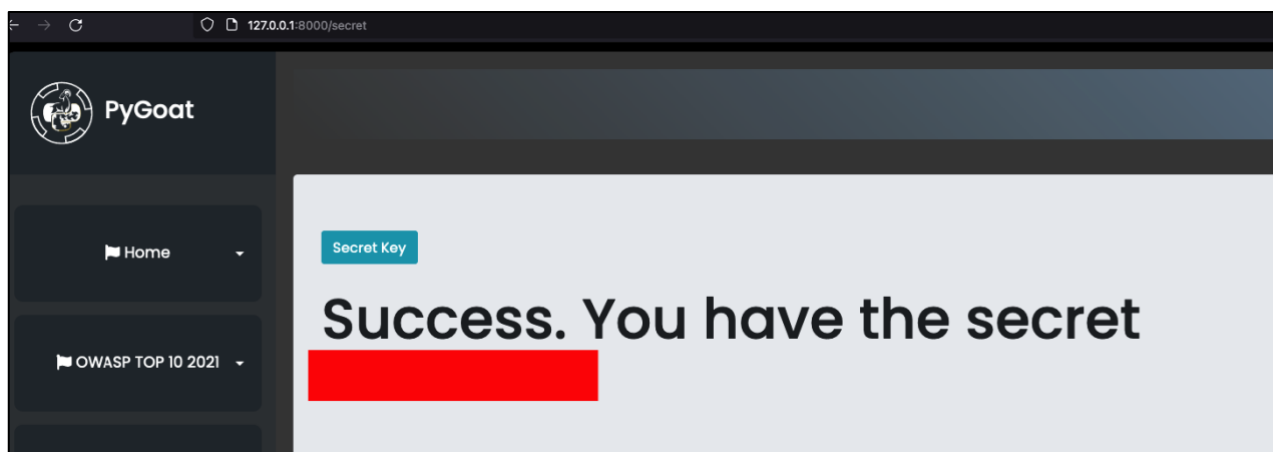
- Ứng dụng có thể gặp phải vấn đề cấu hình bảo mật sai trong một số trường hợp như:
- Các tính năng không cần thiết được bật hoặc cài đặt (ví dụ như các cổng, dịch vụ, trang, tài khoản, hoặc quyền không cần thiết)
- Tài khoản mặc định và mật khẩu được bật và không thay đổi.
- Xử lý thông báo lỗi quá mức cho người dùng
- Đối với các hệ thống được nâng cấp, tính năng bảo mật mới nhất bị vô hiệu hoá hoặc không được cấu hình an toàn.
- Phần mềm bị lỗi thời hoặc có lỗ hổng (xem thêm A06:2021-Vulnerable and Outdated Components)

b. Kịch bản tấn công thực hành

- Kịch bản thực hành này sẽ có một chức năng để lấy khoá bí mật, tuy nhiên nó chỉ có thể truy cập bởi người dùng quản trị.
- Truy cập bài tập tại: http://localhost:8000/sec_mis_lab
- Kết quả khi nhấn lấy khoá bí mật, thông báo hiện ra là chỉ có trang admin.localhost:8000 mới có thể truy cập vào chức năng này. Và thông báo X-Host lúc này là None



- Khi trở thành người dùng quản trị, khoá bí mật có thể nhìn thấy.



Chậm lại và suy nghĩ 5: X-Host is None là gì? Có kiểm soát được X-Host không.

Bài tập 6: Báo cáo lỗ hổng đang được thực hành. Có thể sử dụng format theo mẫu sau:

#Tiêu đề: Tiêu đề rõ ràng và ngắn gọn bao gồm loại lỗ hổng và tài sản bị ảnh hưởng.

#Mô tả lỗ hổng: Lỗ hổng này là gì? Trong các bước rõ ràng, làm thế nào để bạn tái tạo nó?

Tóm tắt:

Các bước để thực hiện lại và bằng chứng:

1. bước 1

2. bước 2

3. bước 3

Tài liệu hỗ trợ và tham khảo:

* Liệt kê bất kỳ tài liệu bổ sung nào (ví dụ: ảnh chụp màn hình, nhật ký, v.v.)

#Mức độ ảnh hưởng của lỗ hổng: Tác động bảo mật nào mà kẻ tấn công có thể đạt được?

#Khuyến cáo khắc phục: Làm thế nào để vá lỗ hổng này?

3. Bài tập luyện tập

1. http://localhost:8000/broken_access_lab_2
2. http://localhost:8000/broken_access_lab_3
3. http://localhost:8000/cryptographic_failure/lab2
4. http://localhost:8000/cryptographic_failure/lab3
5. http://localhost:8000/cmd_lab
6. <http://localhost:8000/ssti/lab>
7. http://localhost:8000/data_exp_lab
8. http://localhost:8000/sec_mis_lab3
9. <https://portswigger.net/web-security/sql-injection/examining-the-database/lab-listing-database-contents-oracle>
10. <https://portswigger.net/web-security/file-path-traversal/lab-absolute-path-bypass>
11. <https://portswigger.net/web-security/access-control/lab-multi-step-process-with-no-access-control-on-one-step>
12. <https://portswigger.net/web-security/logic-flaws/examples/lab-logic-flaws-infinite-money>
13. <https://portswigger.net/web-security/prototype-pollution/client-side/browser-apis/lab-prototype-pollution-client-side-prototype-pollution-via-browser-apis>

D. YÊU CẦU & ĐÁNH GIÁ

- Sinh viên tìm hiểu và thực hành theo hướng dẫn, thực hiện theo nhóm đã đăng ký.
- Nộp báo cáo kết quả gồm Code, CSDL được export và chi tiết những việc (Report) mà nhóm đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Báo cáo:
 - File .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
 - Đặt tên theo định dạng: [Mã lớp]-LabX_MSSV1.
 - Ví dụ: [NT213.K11.ANTN.1]-Lab1_1852xxxx-.
 - Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
 - Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Bài sao chép, trộm, ... sẽ được xử lý tùy mức độ vi phạm.

HẾT

Chúc các bạn hoàn thành tốt!