

LAB 8 : Kỹ thuật upload file trong PHP

Mục tiêu:

- Nắm và hiểu rõ cơ chế kỹ thuật upload một hoặc nhiều file trong php.
- Biết cách ràng buộc các file do người dùng upload lên server để tránh vấn đề bị hack hoặc tạo backdoor.

1. Tiến hành upload một hoặc nhiều file trong php

- **Bước 1:** Tạo file php với tên là upload_mssv của bạn. Ví dụ upload_0412006.php
- **Bước 2:** Tiến hành thiết kế giao diện tìm kiếm như sau:

Choose file for Upload: Không có tệp nào được chọn

```
<html>
  <head>
    <title> xu lý upload file</title>
  </head>
  <body>
    <!-- Lưu ý mọi thao tác upload dữ liệu đều phải nằm trong form -->
    <!-- Trong form: action -> chính là file/link để nhận dữ liệu xử lý. -->
    <!-- Nếu # hoặc rỗng thì là chính trang/file/link hiện tại nhận dữ liệu xử lý -->
    <!-- Trong form: Method: Post/Get -> Phương thức dùng truyền dữ liệu -->
    <!-- Lưu ý, upload file -> có thêm trường enctype="multipart/form-data"-->
    <form action = "#" method="Post" enctype="multipart/form-data">
      <table border="0">
        <tr>
          <td>Choose file for Upload:</td>
          <td>
            <!-- Để upload file từ client thì dùng thẻ input -->
            <!-- Type: file (cho phép người dùng chọn file từ máy) -->
            <!-- name: nó là cái biến dùng để chỉ file đã chọn -->
            <td><input name="userfile" type="file"></td>
            <!--Tạo một cái nút để upload file lên server -->
            <td><input type="submit" value="Upload"></td>
          </tr>
        </table>
      </form>
    </body>
  </html>
```

- **Bước 3:** Tiến hành viết code xử lý (php) ở phía server để nhận xử lý file, hiển thị thông tin file ra dùng biến toàn cục \$_FILES.

```
<?php
//để xử lý các file thì dùng tới biến toàn cục $_FILES như $_REQUEST, $_Post, ...
//Lưu ý userfile chính là tên của control chứa tên file ở thiết kế
if (isset ($_FILES['userfile']))
{
    $file = $_FILES['userfile'];
    echo " <h2> File Information </h2>";
    echo " <b>Trường name (tên của file):</b> ".$file['name']."</br>";
    echo " <b>Trường type (kiểu của file):</b> ".$file['type']."</br>";
    echo " <b>Trường tmp_name (đường dẫn lưu tạm file trên server):</b> "
        . $file['tmp_name']."</br>";
    echo " <b>Trường error (mã lỗi nếu có):</b> ".$file['error']."</br>";
    echo " <b>Trường size (kích thước file):</b> ".$file['size']."</br>";

    //gọi hàm error get last để lấy về lỗi cuối cùng
    //phát sinh và in ra mình hình bạn xem dùng print_r
    print_r (error_get_last());
}

?>
</body>
</html>
```

- **Bước 4:** Tiến hành chạy chương trình, chọn 1 file bất kỳ và xem kết quả.

Choose file for Upload: ps_adds.sql

File Information

Trường name (tên của file): ps_adds.sql

Trường type (kiểu của file): application/octet-stream

Trường tmp_name (đường dẫn lưu tạm file trên server): C:\wamp64\tmp\phpBDE7.tmp

Trường error (mã lỗi nếu có): 0

Trường size (kích thước file): 32458















- **Bước 5:** Ở bước trên ta mới thực hiện được quá trình upload file và server mới chỉ lưu tạm file đó trên server thôi. Bây giờ ta muốn lưu file đó vào nơi mong muốn thì ta dùng hàm `move_uploaded_file` như sau:

```
    . $file['tmp_name'] . "</br>";
echo " <b>Trường error (mã lỗi nếu có):</b> ".$file['error']."</br>";
echo " <b>Trường size (kích thước file):</b> ".$file['size']."</br>";

//lưu file lại trên server dùng hàm move_uploaded_file
//Tham số 1 là đường dẫn lưu tạm trên server ở trên
//Tham số 2 là đường dẫn đến nơi lưu trên server.
if (move_uploaded_file($file['tmp_name'], $file['name']))
    echo 'upload thanh cong';
else
    echo 'upload that bai';

//gọi hàm error get last để lấy về lỗi cuối cùng
//phát sinh và in ra mình hình bạn xem dùng print_r
print_r (error_get_last());
```

- **Bước 6:** Tiến hành chạy chương trình, chọn 1 file bất kỳ rồi upload. Sau đó vào nơi chứa file code kiểm tra có file upload lên chưa.

	0412006.php	12/17/2017 2:15 PM	PHP Script	4 KB
	0412006_jquery.php	12/17/2017 4:01 PM	PHP Script	2 KB
<input type="checkbox"/>	 ajax.php	12/17/2017 1:19 PM	PHP Script	1 KB
	jquery.js	11/25/2017 10:56 ...	JavaScript File	71 KB
	new 1.txt	12/17/2017 8:51 A...	Text Document	5 KB
	pic1.jpg	12/17/2017 9:56 A...	JPG File	9 KB
<input checked="" type="checkbox"/>	 ps_adds.sql	12/23/2017 8:32 A...	SQL File	32 KB
	test.php	12/17/2017 2:26 PM	PHP Script	1 KB
	test.txt	12/17/2017 2:15 PM	Text Document	4 KB
	timSach.php	12/10/2017 8:07 A...	PHP Script	1 KB
	trangchu.php	11/19/2017 11:22 ...	PHP Script	1 KB
	xltimSach.php	12/10/2017 8:12 A...	PHP Script	1 KB
	xlUploadfile.php	12/23/2017 8:32 A...	PHP Script	3 KB
	xuly.php	12/10/2017 9:46 A...	PHP Script	1 KB

- **Bước 7:** Các bước trên là ta upload một file nhưng giờ ta muốn cùng lúc upload nhiều file thì sao? Nếu thế thì bạn có thể clone/copy_paste để tạo nhiều thẻ input với type là file.

```
<tr>
  <td>Choose file for Upload:</td>
  <td><input name="userfile1" type="file"></td>
</tr>
<tr>
  <td>Choose file for Upload:</td>
  <td><input name="userfile2" type="file"></td>
</tr>
<tr>
  <td>Choose file for Upload:</td>
  <td><input name="userfile3" type="file"></td>
</tr>
<tr>
  <td><input type="submit" value="Upload"></td>
</tr>
```

Rồi tiến hành vào clone/copy_paste xử lý code php ở phía server lên (cách này bạn tự làm).

- **Bước 8:** Ở đây mình không dùng thủ công như ở bước 7. Mình có cách làm ngắn gọn như sau bằng cách chỉnh thiết kế ở bước 2 thêm trường multiple và [].

```
<form action = "#" method="Post" enctype="multipart/form-data">
  <table border="1">
    <tr>
      <td>Choose file for Upload:</td>
      <td><!-- Muốn upload nhiều file thì chỉ cần thêm thuộc tính multiple -->
        <!-- và chỗ name mình thêm [] (mảng) -->
        <input name="userfile[]" type="file" multiple></td>
      <td><input type="submit" value="Upload"></td>
    </tr>
  </table>
</form>
```

- **Bước 9:** Trong đoạn code xử lý ở phía server tiến hành chỉnh sửa lại như sau:

```
<?php
if (isset ($_FILES['userfile']))
{
    //Lấy thông tin các file upload đưa vào biến $file
    $file = $_FILES['userfile'];
    //Lấy về số lượng các file upload thông qua hàm count
    $file_count = count($file['name']);
    //Tiến hành quét từng file và xử lý
    for($i=0;$i<$file_count;$i++)
    {
        echo "<h2> File Information </h2>";
        echo "<b>name:</b> ".$file['name'][$i]."</br>";
        echo "<b>type:</b> ".$file['type'][$i]."</br>";
        echo "<b>tmp_name:</b> ".$file['tmp_name'][$i]."</br>";
        echo "<b>error:</b> ".$file['error'][$i]."</br>";
        echo "<b>size:</b> ".$file['size'][$i]."</br>";

        if (move_uploaded_file($file['tmp_name'][$i], $file['name'][$i]))
            echo 'upload thanh cong';
        else
            echo 'upload that bai';
    }
}
?>
```

- **Bước 10:** Tiến hành chạy và kiểm tra kết quả.

2. Thêm các ràng buộc về dữ liệu upload file trong php

- **Bước 1:** Làm tiếp bài trên. Bây giờ ta muốn người dùng chỉ được phép chọn các file ảnh để upload lên server mà không được phép chọn các file khác như exe, ... để upload lên server để tránh vấn đề bị hack hoặc tạo backdoor. Ta tiến hành thêm tùy chọn **accept** sau vào phần thiết kế để ràng buộc ở phía client

```
<tr>
<td>Choose file for Upload:</td>
<!-- Muốn ràng buộc người dùng chỉ được phép chọn loại file mình mong muốn-->
<!-- thì dùng thuộc tính accept -->
<!-- Ở đây chỉ cho phép chọn file ảnh với định dạng .jpg, .jpeg, .png -->
<td><input name="userfile[]" type="file" accept=".jpg, .jpeg, .png" multiple></td>
<td><input type="submit" value="Upload"></td>
</tr>
```

- **Bước 2:** Bước 1 chỉ mới ràng buộc ở phía client. Ta phải double check lại ở phía server để đảm bảo an toàn. Ta thêm code check ở phía server cho trường hợp trên bằng cách thêm mảng \$type và code if như sau:

```

if (isset ($_FILES['userfile']))
{
    //Định nghĩa thêm mảng các dạng file mình được phép upload
    $types = array('image/jpeg', 'image/gif');

    $file = $_FILES['userfile'];
    $file_count = count($file['name']);
    for($i=0;$i<$file_count;$i++)
    {
        echo "<h2> File Information </h2>";
        echo "<b>name:</b> ".$file['name'][$i]."<br>";
        echo "<b>type:</b> ".$file['type'][$i]."<br>";
        echo "<b>tmp_name:</b> ".$file['tmp_name'][$i]."<br>";
        echo "<b>error:</b> ".$file['error'][$i]."<br>";
        echo "<b>size:</b> ".$file['size'][$i]."<br>";

        //Tiến hành kiểm tra loại của file upload có nằm trong danh sách
        //loại file cho phép upload
        if (in_array($file['type'][$i], $types)) {
            if (move_uploaded_file($file['tmp_name'][$i], $file['name'][$i]))
                echo 'upload thành công';
            else
                echo 'upload thất bại';
        } else {
            echo 'Định dạng file không được cho upload';
        }
    }
}

```

- **Bước 3:** Tiến hành chạy và thử nghiệm. Tìm cách đưa các file khác không phải hình ảnh lên server?
- **Bước 4:** Đối với các hệ thống thường sợ vấn đề người dùng upload 1 file có dung lượng rất lớn lên server (file nặng 100M, 1GB, 100GB, 1TB,...). Tại sao lại phải sợ như thế, các bạn tự tìm câu trả lời nhé? Để ràng buộc vấn đề này. Mình tiến hành thêm code để ràng buộc không cho người dùng/client chọn file có kích thước lớn hơn 2000000 bytes với thẻ input và type là hidden như sau:

```

<form action = "#" method="Post" enctype="multipart/form-data">
    <table border="1">
        <tr>
            <td>Choose file for Upload:</td>
            <td>
                <!--Để ràng buộc client không chọn file lớn hơn 2000000 byte -->
                <!--Ta dùng thẻ input với type là hidden như sau -->
                <input type="hidden" name="MAX_FILE_SIZE" value = "2000000"/>
                <td><input name="userfile[]" type="file" accept=".jpg, .jpeg, .png" multiple></td>
                <td><input type="submit" value="Upload"></td>
            </tr>
        </table>
    </form>

```

- **Bước 5:** Tiến hành code phía server để double check thêm vấn đề ràng buộc này như sau:

```
for ($i=0;$i<$file_count;$i++)
{
    echo "<h2> File Information </h2>";
    echo "<b>name:</b> ".$file['name'][$i]."</br>";
    echo "<b>type:</b> ".$file['type'][$i]."</br>";
    echo "<b>tmp_name:</b> ".$file['tmp_name'][$i]."</br>";
    echo "<b>error:</b> ".$file['error'][$i]."</br>";
    echo "<b>size:</b> ".$file['size'][$i]."</br>";

    if (in_array($file['type'][$i], $types)) {
        //Tiến hành kiểm tra kích thước file có nhỏ hơn 2000000 bytes không
        //Nếu không thì mình ko lưu file này lại trên server
        if ($file['size'][$i] < 2000000) {
            if (move_uploaded_file($file['tmp_name'][$i], $file['name'][$i]))
                echo 'Upload thành công';
            else
                echo 'Upload thất bại';
        } else {
            echo 'File upload lớn hơn 2000000 bytes';
        }
    } else {
        echo 'Định dạng file không được cho upload';
    }
}
```

- **Bước 6:** Tiến hành chạy và xem kết quả. Bạn hãy tìm cách vượt qua các ràng buộc trên?