

FILE UPLOADS WITH PHP

Andrew Sheehan

Metropolitan College
Boston University

CLIENT



```
<form
  id="met_cs602_php"
  method="post"
  enctype="multipart/form-data"
  onsubmit="return g(this)">
```

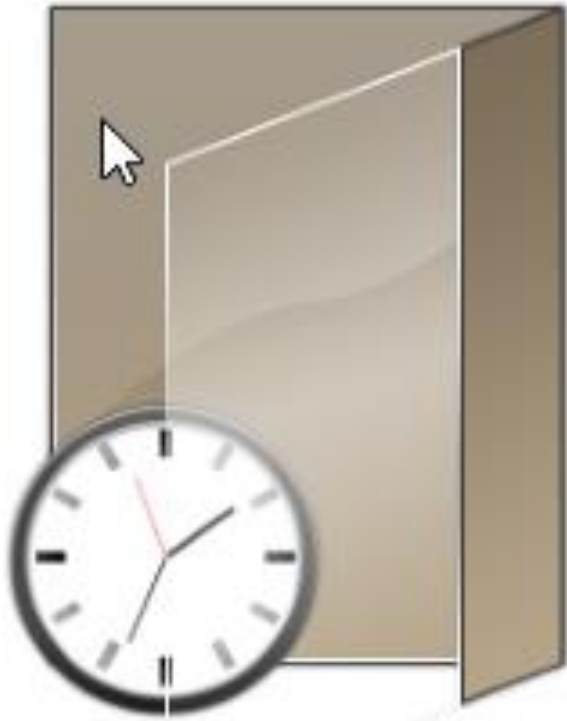
```
<input
  name="resume"
  accept="img/png img/jpeg"
  multiple /* many images, if you declare it */
  type="file"/>
```

UPLOADING

When a user submits the form, the file is uploaded, and the server saves it to a temporary file on the server.



UPLOADING IT'S TEMPORARY



The temporary upload is deleted
when the script ends!



UPLOADING IS IT ON? (PHP.INI)



```
file_uploads = On
```

SUPERGLOBAL

`$_FILES`



An associative array of items
uploaded to the current script
via HTTP POST

FILE UPLOADS

ATTRIBUTES

Index	Description
'name'	The original name of the uploaded file.
'size'	The size of the uploaded file in bytes.
'tmp_name'	The temporary name of the uploaded file on the web server.
'type'	The MIME type of the uploaded file as sent by the user's browser.
'error'	The error code associated with the file. Common values are UPLOAD_ERR_OK (no error), UPLOAD_ERR_INI_SIZE (file was too large), and UPLOAD_ERR_PARTIAL (upload was not completed).

MOVE_UPLOADED_FILE(\$T, \$N)

Uploading: get the file up to your server. You then have to place it to its final location.

Function	Description
<code>move_uploaded_file(\$tmp, \$new)</code>	Moves an uploaded file from its temporary location to a permanent location. If successful, returns TRUE.

CLIENT CHECK ~ VERIFY

```
<script>
  check_file_size = () => {
    const fileTarget = document.getElementById('fileUpload');

    if (fi.files.length > 0) {
      const limit = fi.files.length-1;
      const DEFINED_MAX = 4096;

      for (let i = 0; i <= limit; i++)
      {
        const fsize = fi.files.item(i).size;
        const file = Math.round((fsize / 1024));

        if (file >= DEFINED_MAX) {
          handle(new Error("Your selected file is too large"));
          break;
        }
      }
    }
  }
</script>
```

\$_FILES

1ST LEVEL CHECKS



```
// Undefined | Multiple Files | $_FILES Corruption Attack
// If this request falls under any of them, treat it invalid.
if (
    !isset($_FILES['upfile']['error']) ||
    is_array($_FILES['upfile']['error'])
) {
    throw new RuntimeException('Invalid parameters.');
```

\$_FILES MORE CHECKS



```
// Check $_FILES['upfile']['error'] value.
switch ($_FILES['upfile']['error']) {
    case UPLOAD_ERR_OK:
        break;
    case UPLOAD_ERR_NO_FILE:
        throw new RuntimeException('No file sent.');
    case UPLOAD_ERR_INI_SIZE:
    case UPLOAD_ERR_FORM_SIZE:
        throw new RuntimeException('Exceeded filesize limit.');
    default:
        throw new RuntimeException('Unknown errors.');
}
```

\$ _FILES

MORE CHECKS



```
// Check if file already exists
if (file_exists($target_file)) {
    echo "Sorry, file already exists.";
    $uploadOk = 0;
}
```

\$ _FILES EXAMPLE

```
$tmp_name = $_FILES['file1']['tmp_name'];  
$path = getcwd() . DIRECTORY_SEPARATOR . 'images';  
$name = $path . DIRECTORY_SEPARATOR . $_FILES['file1']['name'];  
$success = move_uploaded_file($tmp_name, $name);  
if ($success) {  
    $upload_message = $name . ' has been uploaded.';  
}
```

