

Forum:[Hack Forums](#)

Thread: [Trying to get a file to upload to the correct directory.](#)

Babadook - January 26, 2019, 6:16 am

Edit: Edited original post with a new issue. x2

Edit x3 Thank you everyone. I got everything working!

TCP - January 26, 2019, 6:42 am

Is \$target_file_p the one being used for the upload of the image as in the script? It looks like "../uploads/Photos" is missing a / before the filename

PHP Code

```
<?php
$dir_p = "../uploads/Photos/";
```

Babadook - January 26, 2019, 7:50 am

(Jan 26, 2019 07:42 AM)Exp500 Wrote: [\[CITE\]](#) Is \$target_file_p the one being used for the upload of the image as in the script? It looks like "../uploads/Photos" is missing a / before the filename

PHP Code

```
<?php
$dir_p = "../uploads/Photos/";
```

Thank you so much. I knew it'd be something simple like that.

Disease - January 26, 2019, 10:58 am

(Jan 26, 2019 08:50 AM)Babadook Wrote: [\[CITE\]](#) Thank you so much. I knew it'd be something simple like that.

A helpful debugging technique, when running down issues like this, is to watch the values of important variables. In this case, you knew the upload was working. It just wasn't being moved to the correct directory. You could've checked the value of the \$target_file_p variable. Either by printing it out via print()/echo() or by inspecting it via print_r()/var_dump().

newQuery - January 26, 2019, 3:43 pm

Define a constant to your ROOT's path directory and use it for creating a link to the upload path.

Babadook - February 1, 2019, 4:18 am

I've updated with a new issue. Thank you everyone for the help on the previous error I had.

+Haze - February 1, 2019, 10:41 am

(Jan 26, 2019 07:16 AM)Babadook Wrote: [\[CITE\]](#) Edit: Edited original post with a new issue. x2

For some reason this assigns the first \$var instead of sorting by the file extension.

Example: When I upload a png it should be assigned a \$var = 3;

However, it goes to the first fileType it comes to and is assigned a \$var = 1;

PHP Code

```
<?php
if (isset($_POST["submit"])) {
    if (filesize($_FILES["fileToUpload"]["tmp_name"]) < $df) { //gets free space)
```

```

$uploadOk = 1;
$error = 1;
if ($fileType == "epub" || $fileType == "mobi" || $fileType == "pdf" || $fileType == "azw" ||
$fileType == "lit") {
    $var = 1;
    $error = 2;
    goto midBreak;
} elseif ($fileType == "wav" || $fileType == "mp3" || $fileType == "ac3" || $fileType == "flac" ||
$fileType == "ape" || $fileType == "aac" || $fileType == "m4a") {
    $var = 2;
    $error = 3;
    goto midBreak;
} elseif ($fileType == "bmp" || $fileType == "png" || $fileType == "jpg" || $fileType == "jpeg" ||
$fileType == "gif") {
    $var = 3;
    $error = 4;
    goto midBreak;
} elseif ($fileType == "apk" || $fileType == "exe" || $fileType == "bat" || $fileType == "app" ||
$fileType == "vb") {
    $var = 4;
    $error = 5;
    goto midBreak;
} elseif ($fileType == "mp4" || $fileType == "ogg" || $fileType == "wmv" || $fileType == "avi" ||
$fileType == "webm" || $fileType == "mov") {
    $var = 5;
    $error = 6;
    goto midBreak;
} elseif ($fileType == "sh" || $fileType == "php") { //Blocks potentially harmful files
    $error = 7;
    goto theEnd;
} else {
    $error = 8;
goto dirUpload;
}
} else {
    $fEnd = 1;
    goto sizeFailure;
}
}

```

Just a heads up buddy, it's deemed bad programming practice to use goto statements in your code. It creates spaghetti code that becomes increasingly difficult to read and maintain.

Also,

Code (\$fileType == "mp4" || \$fileType == "ogg" || \$fileType == "wmv" || \$fileType == "avi" || \$fileType == "webm" || \$fileType == "mov") is a bit messy, try something like.

```

Code function isExtensionValid($fileType, $validExtensions) {
    foreach($validExtensions as $ext) {
        if ($ext == $fileType) {
            return true;
        }
    }
}

```

```

    }

    return false;
}
That way you can just do

```

```

Code if (isExtensionValid($fileType, ["mp4", "ogg", "wmv", "avi", "webm", "mov"])) {
    $var = 4;
    $error = 5;
    callFunction();
}

```

I wrote this on top of my head so it may not fully work, but this approach makes it easier to maintain and read your code.

Abruptum - February 1, 2019, 3:00 pm

(Feb 1, 2019 11:41 AM)+Haze Wrote: [CITE](#) Just a heads up buddy, it's deemed bad programming practice to use goto statements in your code. It creates spaghetti code that becomes increasingly difficult to read and maintain.

Also,
 Code (\$fileType == "mp4" || \$fileType == "ogg" || \$fileType == "wmv" || \$fileType == "avi" || \$fileType == "webm" || \$fileType == "mov")
 is a bit messy, try something like.

```

Code function isExtensionValid($fileType, $validExtensions) {
    foreach($validExtensions as $ext) {
        if ($ext == $fileType) {
            return true;
        }
    }

    return false;
}
That way you can just do

```

```

Code if (isExtensionValid($fileType, ["mp4", "ogg", "wmv", "avi", "webm", "mov"])) {
    $var = 4;
    $error = 5;
    callFunction();
}

```

I wrote this on top of my head so it may not fully work, but this approach makes it easier to maintain and read your code.

A even better solution would be to use [in_array\(\)](#)

PHP Code

```

<?php
if (in_array($fileType, ["mp4", "ogg", "wmv", "avi", "webm", "mov"])) {
    $var = 4;
    $error = 5;
    callFunction();
}

```

And pleas never never never never never EVER use goto..... The results can only be garbage.

Do it like this :

PHP Code

```
<?php
define("EBOOK_FILE", 1);
define("AUDIO_FILE", 2);
define("IMAGE_FILE", 3);
define("EXECUTABLE_FILE", 4);
define("VIDEO_FILE", 5);
define("UNSUPPORTED_FILE", 6);
$fileType = "png";

switch(getFileExtensionType($fileType)){
    case EBOOK_FILE:
        echo "its a Ebook";
        break;
    case AUDIO_FILE:
        echo "its Music";
        break;
    case IMAGE_FILE:
        echo "its a Image";
        break;
    case EXECUTABLE_FILE:
        echo "its a Executable";
        break;
    case VIDEO_FILE:
        echo "its a Video";
        break;
    case UNSUPPORTED_FILE:
        echo "Unsupported (maybe some sort of attack)";
        break;
}

function getFileExtensionType($fileType){
    if (in_array($fileType, ["epub", "mobi", "pdf", "azw", "lit"])){
        return EBOOK_FILE;
    }elseif(in_array($fileType, ["wav", "mp3", "ac3", "flac", "ape", "aac", "m4a"])){
        return AUDIO_FILE;
    }elseif (in_array($fileType, ["bmp", "png", "jpg", "jpeg", "gif"])){
        return IMAGE_FILE;
    }elseif (in_array($fileType, ["apk", "exe", "bat", "app", "vb"])){
        return EXECUTABLE_FILE;
    }elseif (in_array($fileType, ["mp4", "ogg", "wmv", "avi", "webm", "mov"])){
        return VIDEO_FILE;
    }else{
        return UNSUPPORTED_FILE;
    }
}
```

+Haze - February 1, 2019, 3:47 pm

(Feb 1, 2019 04:00 PM)Abruptum Wrote: [\[CITE\]](#) x

I'm not all that familiar with PHP as I used to but yeah that's a better cleaner solution.

Babadook - February 1, 2019, 6:23 pm

Thank all of you for your help. I'll start testing your suggestions. If it's not obvious by now, I'm teaching myself as I go. The goto statements, I know that nobody likes them, but without them, my code ran through all of my if statements whether they were true or false so I needed something to skip the codes it didn't need. Hopefully I can get everything working without them. Thank you again!

J?ey - February 1, 2019, 7:50 pm

(Feb 1, 2019 07:23 PM)Babadook Wrote: [\[CITE\]](#) Thank all of you for your help. I'll start testing your suggestions. If it's not obvious by now, I'm teaching myself as I go. The goto statements, I know that nobody likes them, but without them, my code ran through all of my if statements whether they were true or false so I needed something to skip the codes it didn't need. Hopefully I can get everything working without them. Thank you again!

When you're utilizing if with else-if's, the first conditional that is met true the rest of the else-if/else's are skipped over.

Babadook - February 1, 2019, 9:50 pm

Down to my last error, everything to this point has been fixed. Just need to figure out where I'm messing up the file_exists part.