iranphp articles

عنوان مقاله : مدیریت فایل های ارسالی توسط کاربران
نگارنده : مهدی ساغری
آدرس پست الکترونیک : .....
تاریخ نگارش : .....

مدیریت فایل های ارسالی توسط کاربران:

حتما تا بحال سایت هایی را دیده اید که به کاربران اجازه Upload فایل از طریق صفحات (فرم های Html) را می دهند به عنوان مثال در قسمت Profileهای یاهو که شما به عنوان یک کاربر عادی می توانید تصویر خود را که یک فایل با فرمت Gif یا Jpg است و یا نمونه پیغام صوتی خود را به Profileتان اضافه کنید و یا بخش آرشیو سایت های تحقیقاتی که به اعضای خود اجازه می دهند نتایج کار خود را در قالب یک فایلPDF و یا بصورت Zipشده بر روی سایت قرار دهند .

به مطالب گفته شده در بالا حتما به این نتجه رسیده اید که در اینگونه سیستم ها سه نکته اساسی قرار داد .

۱-چکونگی ارسال فایل با استفاده از پروتکل HTTP به جای FTP .

۲-بررسی فرمت فایل و جلوگیری از Upload فایل های غیر مجاز .

۳-بررسی اندازه فایل و جلوگیری از ارسال فایلهای حجیم .

برای دریافت فایل از کاربر ابتدا باید فرمی به شکل زیر ایجاد کرد :

```
<form name="form1" method="post" action="_URL_" enctype="multipart/form_data">
<input type="file" name="userfile">
<input type="submit" name="Submit" value="Submit">
</form>
```

عبارت multipart/form_data فرم بالا را از فرمهایی که تا بحال با آن کار کرده ایم متمایز می کند . در فرم بالا قبل از دکمه Submit کادر مخصوص در یافت فایل قرار گرفته در واقع این همان فایلی است که کاربر قصد ارسال به سرور را دارد .

در این فرم _URL_ اشاره به برنامه PHP ای دارد که قصد دارید عمل دریافت فایل را در آنجا انجام دهید .

تا اینجا فرم مورد نیاز طراحی شده است حال با یک برنامه ساده مثل نمونه زیر می توان فایل را از Client خوانده و بر روی سرور ذخیره کرد .

```
<?
  if(isset( $Submit ))   {
if ($_FILES['userfile']['type'] == "image/gif"){   copy ($_FILES['imagefile']['tmp_name'],
"files/".$_FILES['imagefile']['name'])
  or die ("Could not copy");

  echo "";
  echo "Name: ".$_FILES['imagefile']['name']."";
  echo "Size: ".$_FILES['imagefile']['size']."";
  echo "Type: ".$_FILES['imagefile']['type']."";
  echo "Copy Done....";
  }
  else {
  echo "";
  echo "Could Not Copy, Wrong Filetype (" .$_FILES['imagefile']['name'].")";

  }
  }?>
```

این برنامه در صورتی که فایل ارسالی یک فایل GIF باشد آنرا با همان نامی که روی کامپیوتر کاربر قرارداشته در شاخه Files بر روی سرور ذخیره می کند .

البته همانطور که میدانید متغییر هایی که برای Upload فایل تعریف می شوند بسته به نسخه PHP و تنظیمات آن متفاوت می باشند .

بعد از یک Upload موفق ، هنگامی که track_vars فعال باشد آرایه های$_FILES و $HTTP_POST_FILES ایجاد می گردند .

سرانجام اگر register_globals فعال باشد متغییرها به صورت globals ایجاد خواند شد .

توجه : track_vars از نسخه ۴,۰,۳ به طور پیش فرض فعال است . از PHP 4.1.0 به بعد ترجیحا از متغییر عمومی$_FILES به جای

$HTTP_POST_FILES استفاده می گردد .

$HTTP_POST_FILES/$_FILES : به منظور مهیا کردن مشخصات فایل (های) ارسال شده ایجاد شده اند . محتوای این دو متغییر به شرح زیر است

البته توجه کنید که 'userfile' نام انتخابی ما در هنگام ساخت فرم HTML است

```
<?
$HTTP_POST_FILES['userfile']['name'];
?>
```

نام اصلی فایل بر روی کامپیوتر کاربر .

```
<?
$HTTP_POST_FILES['userfile']['type'];
?>
```

شناسه محتوای (MIME type) فایل . در برنامه فوق تنها به فایلهای GIF اجازه ذخیره شدن داده می شود حال چنانچه بخواهیم کاربر را محدود به ارسال فایل های Wav نماییم

باید مقدار این متغییر را با "audio/wav" مقایسه کنیم :

```
<? $HTTP_POST_FILES['userfile']['size'] ?>[code]
اندازه فایل برحسب بایت .

[code]<? $HTTP_POST_FILES['userfile']['tmp_name'] ?>
```

نام موقتی که فایل در هنگام Upload موقتا با آن نام بر روی سرور ذخیره می گردد .

توجه : متغییر $_FILES تنها در نسخه های ۴,۱,۰ و بالاتر شناخته شده است و نیز نسخه ۳ PHP از $HTTP_POST_FILES حمایت می کند .همچنین اگر در فایل PHP.INI پارامتر register_globals فعال باشد متغییرهای زیر موجود خواهند بود:

```
<?
$userfile
$userfile_name
$userfile_size
$userfile_type
?>
```

فایل ها به طور پیش فرض در دایرکتوری فایلهای موقت سرور ذخیره می شوند مگر اینکه مکان دیگری توسط upload_tmp_dir در فایل PHP.INI تعیین شده باشد . در پایان اجرای اسکریپت چناچه فایل ارسال شده از طرف کاربر را به مکان دیگری انتقال ندهید و یا نام آن را تغییر ندهید توسط سیستم این فایل پاک خواهد شد .

در صورت استفاده از PHP4.1.0 و بالاتر می توانید خط اول برنامه فوق را بصورت زیر بنویسید :

```
<?
if(is_uploaded_file($HTTP_POST_FILES['userfile']['tmp_name'])) <br />
  { <br />
  //In PHP 4.1.0 or later, $_FILES should be used instead of $HTTP_POST_FILES.
```

ارسال چندین فایل به طور همزمان:

همانطور که میدانید یک برنامه را می توان به چنیدن روش مختلف پیاده سازی کرد ولی چه خوب است که این پیاده سازی همراه با استفاده بهینه از گرامر زبان باشد .

برای اینکه کاربر بتواند بیش از یک فایل را در یک لحظه ارسال کند فرمی به شکل زیر طراحی کنید :

```
<form name="form1" method="post" action="_URL_" enctype="multipart/form_data">
<input type="file" name="userfile[]">
```

```
<input type="file" name="userfile[]">
<br />
  <input type="submit" name="Submit" value="Submit">
</form>
```

همانطور که مشاهده می شود این فرم دارای دو کادر دریافت فایل است که به نام آنها به صورت یک آرایه است . پس از ارسال این فرم بر روی سرور متغییر های :

```
<?
$HTTP_POST_FILES['userfile']['name'][0]
$HTTP_POST_FILES['userfile']['size'][0]
$HTTP_POST_FILES['userfile']['type'][0]
$HTTP_POST_FILES['userfile']['tmp_name'][0]
?>
```

دلالت بر مشخصات فایل اولی و متغییرهای:

```
<?
$HTTP_POST_FILES['userfile']['name'][1]
$HTTP_POST_FILES['userfile']['size'][1]
$HTTP_POST_FILES['userfile']['type'][1]
$HTTP_POST_FILES['userfile']['tmp_name'][1]
?>
```

دلالت بر مشخصات فایل دوم دارند. همینطور الی آخر می توان چندین فایل را ارسال کرد.

روش HTTP PUT

روش دیگری نیز برای ارسال فایل به سرور وجود دارد که بسیار ساده تر از روش بالا عمل می کند ولی چون این روش برای حفظ امنیت سرور نیاز به تنظیمات خاصی در سرورها دارد از توضیح آن در اینجا صرف نظر کرده ام . چنانچه مایل به آشنایی با این روش هستید به آدرس زیر مراجعه کنید :

http://www.php-center.de/en-html-manual/features.file-upload.put-method.html

درهنگام نوشتن چنین برنامه هایی مراقب مجوز هایی که به دایرکتوری ذخیره این فایل ها می دهید و همچنین بررسی اینکه فایل ارسال شده از طرف کاربر همان چیزی است که باید باشد باشید .فراموش نکنید که همیشه عده ای بر روی اینترنت به دنبال راهی برای نفوذ به کامپیوتر و یا وب سایت شما هستند بد نیست نگاهی هم به مقاله زیر داشته باشید :

http://securityresponse.symantec.com/avcenter/security/Content/2208.html