PHP: Database Demo, File Uploads, Mail, Security

CS174

Chris Pollett

Oct 31, 2007.

Outline

- Database Demo
- More Form Handling
- Preventing Bogus Form Spam
- Mail
- Injection Attacks

Database Demo

- Last day, showed the basics of making a connection from PHP to a mysql database.
- I didn’t demo that this actually works.
- So today lets look at an example.
- Using phpMyAdmin, I made a database my_db with one table foo which in turn has one integer column bar.
- I inserted 3 rows into this table.

More Database Demo

- Next I created the following mysql_test.php file:

```
<?php

$db = mysql_connect("localhost","root", "");

mysql_select_db("my_db");

$query = "select bar from foo";

$result = mysql_query($query);

$num_rows = mysql_num_rows($result);

for($i=0; $i < $num_rows; $i++)
{
   $row = mysql_fetch_array($result);
   print "<p>".$row['bar']."</p>";
}
mysql_close();
?>
```

- This was placed under localhost and run giving me the three lines for bar I had inserted into the table.

File Uploads

- We are now going to look at a couple of useful things PHP can do with regard to form processing.
- We have already seen that usually information sent from forms is provided to our PHP scripts in the global variables: $_REQUEST, $_POST, $_GET
- These variables though are not used to handle file uploads. Instead, the variable $_FILES is used.

Example

- Consider the form:

```
<form enctype="multipart/form-data" method="post" action="test_upload1.php"
   >
  <input type="hidden" name="MAX_FILE_SIZE" value="1000000" /><!-- The
   size is also controlled by php.ini -->
  <input type="file" name="docname" />
  <input type="submit" value="Upload" />
</form>
```

- When test_upload1.php is run, the global variable $_FILES[“docname”] will be set to something like:

```
Array(
    [name] => mystyles.css
    [type] => text/css
    [tmp_name] => /private/var/folders/k-/k-GHnyslGhyOhMqq80ZXgk+++TI/-Tmp-/phpcSLlhk
    [error] => 0    [size] => 157)
```

- Hence, we can then do a command like:

```
move_uploaded_file($_FILES["docname"]["tmp_name"],
    "$where_we_want");
```

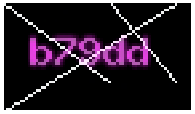
  to get the file where we would like.

Security

- We are now going to spend the rest of the lecture looking at various things vaguely connected with security.
- Often data from clients comes to your server via some form. For example, the file file upload processing we just did.
- One annoyance you will have to deal with is robots that find your web forms and upload garbage to your site, spamming you.
- One solution to this problem is to use Captcha's.

Captcha

- CAPTCHA stands for **C**ompletely **A**utomated **P**ublic **T**uring **T**est to tell **C**omputers and **H**umans **A**part.
- These were developed at Carnegie Mellon around 2000 by Luis von Ahn, Manuel Blum, Nicholas Hopper, and John Langford.
- The basic idea is that you put on your form an image of a distorted string.
- You hope the robot cannot decipher the string from the image so won't be able to fill out that portion of the form correctly:

Please type the following text: 

Making a Simple Captcha in PHP

- The previous CAPTCHA might be created with the code:

```
<p><b>Please type in the following text:</b>
<?php
    $md5 = md5(microtime() * mktime());
    $captcha_string = substr($md5,0,5);
     $captcha_img = imagecreatetruecolor(70, 40);
      $color = imagecolorallocate($captcha_img, 255, 0, 255);
      $line = imagecolorallocate($captcha_img,233,239,239);
     imagestring($captcha_img, 5, 10, 10, $captcha_string, $color);
     imageline($captcha_img,0,0,39,29,$line);
     imageline($captcha_img,40,0,64,29,$line);
     imageline($captcha_img,0,40,64,0,$line);  imagejpeg($captcha_img,
     "images/captcha.jpg",100); imagedestroy($captcha_img);
$_SESSION['key'] = md5($captcha_string);  ?>
<img src="images/captcha.jpg" alt="captcha" style="position:relative;
     top:15px;"/>
<input type="text" name="key" size="5" /></p>
```

Sending a Mail Message

- It is often useful to collect a person’s e-mail address with a form.
- By mailing, a person a special code that allows them to complete a registration process, one can verify that one has a real e-mail address of a real person.
- The simplest way to do this is to use the mail() command:

```
 $message = “Here is a mail message”;
 mail(“Someone@somewhere.com”,
         “Here is the title”,
         $message,
         "From: cpollett@somewhereelse.com”);
```

- This could be combined with a captcha to try to reduce the risk of your site spamming other sites.

Injection Attacks and Prevention

- Another kind of attack on a web-site’s forms is to carryfully fill out form variables to break the PHP script behind the forms variables.
- Consider the following SQL which might be used to insert into a database:

```
$sql = "INSERT INTO   users (reg_username,reg_password,                    reg_email)
    VALUES
   ('{$_POST['reg_username']}', $_POST[ '$reg_password’],
    '{$_POST['reg_email']}')";
```

- What if the posted reg_username is:

```
      bad_guy', 'mypass', ''), ('good_guy  ?
```

- You can use PHP commmands like: mysql_escape_string() or addslashes around the posted variable to prevent this problem.