Unit 4 Module 4: Files, Sessions, & Cookies

Dylan Lane McDonald

CNM STEMulus Center
Web Development with PHP

September 18, 2014

Outline

Files

Files can be uploaded using HTML forms using the `file` type of the `<input>` tag. Uploading files presents the following three challenges:

1. Configuring `php.ini`
2. Protecting against malicious & incompetent users
3. Saving the files to a safe place

Files

Files can be uploaded using HTML forms using the `file` type of the `<input>` tag. Uploading files presents the following three challenges:

1. Configuring `php.ini`
2. Protecting against malicious & incompetent users
3. Saving the files to a safe place

`php.ini`'s `post_max_size` and `max_filesize` need to be set to size parameters that can accommodate the maximum size of the files you wish to upload. This can be set by the systems administrator. These parameters can **NOT** be effectively set by the `ini_set()` function.

Files: Malicious & Incompetent Users

Always do very restrictive verification on files. Verify **both** on extension and MIME type.

```
$goodExts = array("jpg", "png");
$goodTypes = array("image/jpeg", image/png");
$name = $_FILES["file"]["name"];
$extension = end(explode(".", $name));
$type = $_FILES["file"]["type"];
if(in_array($extension, $goodExts) === false || in_array(
    $type, $goodTypes) === false) {
  // bad file: throw an exception?
}
```

Listing 1: Verifying a File

Files: Saving Files

The file must then be written to a "safe" directory that is writable by Apache's "nobody" user. A safe directory would be a directory that is:

- isolated from the rest of the file system
- only accessible to the Apache "nobody" user
- constantly monitored by for threats by both humans and anti-virus software

Files: Saving Files

The file must then be written to a "safe" directory that is writable by Apache's "nobody" user. A safe directory would be a directory that is:

- isolated from the rest of the file system
- only accessible to the Apache "nobody" user
- constantly monitored by for threats by both humans and anti-virus software

Once a safe directory has been established, the files can then be saved by using the move_uploaded_file() function. **Be sure to only move the file after the files has been verified as being secure.**

Cookies

Definition

A **cookie** *is a variable that is sent with the HTTP headers and is stored locally in the browser and sent to the server.*

Sending and receiving this data, user preferences and settings can now persist across pages as the user browses the site.

Cookies can be set to expire sometime in the future (a persistent cookie) or when the browser is closed (a session cookie).

The term *cookie* comes from the term *magic cookie*, which is a term for shared data between cooperating programs. "I give him a packet, he gives me back a cookie." [1]

Setting Cookies

Cookies are set with the HTTP headers, so must, by definition, be sent before any page content is rendered. If one tries to set a cookie after any or all of the content is rendered, this will result in PHP script error. All cookie data is in the superglobal `$_COOKIE`.

```php
// set a session cookie
setcookie("name", "value");
// set a one hour persistent cookie
setcookie("name", "value", time() + 3600);
// destroy a cookie outright
setcookie("name", "value", time() - 3600);
```

Listing 2: Setting a Cookie

Sessions

Definition

A **session** *is a container for persistent or semi-persistent data during the user's experience on the site.*

PHP will create a session for the visitor and an associated, system-generated session ID. Once the session is started, the session variables are available in the superglobal `$_SESSION`.

The session ID is the heart of the session and can be passed via either cookies or as a URL parameter. Cookies are optimal, but PHP will transparently fall back when cookies are unavailable. [2]

PHP Sessions

Like cookies, sessions must be initialized with the HTTP headers, so the session must be setup before any content is generated.

```php
// create the session
session_start();
// add data to the session
$_SESSION["foo"] = "bar";
// delete data from the session
unset($_SESSION["baz"]);
// destroy the session
session_destroy();
```

Listing 3: Using a PHP Session

Works Cited


Eric S. Raymond.
Jargon file: Cookie.
http://www.catb.org/~esr/jargon/html/C/cookie.html.


PHP Project.
Passing the session id.
http://www.php.net/manual/en/session.idpassing.php.