Lecture 3 Overview

- Day2/Lab2 Questions
- Review Student Videos
- Lecture

Todays Topics

- Hashing/Salting
- Encryption
- SSL Encryption
- Cookies
- Sessions
- Image Manipulation
- Filesystem

- **First, create an SSL/Day3 Folder and set the root of MAMP to this folder.**

Hashing

http://us1.php.net/manual/en/faq.passwords.php#faq.passwords.fasthash
A hash is a one-way cryptographic algorithm to mathematically compute an input to a fixed-length result. Hashes are quickly computed with minimal processing resources (making them fast), and are often used as a checksum to compare two larger values.

MD5 - 32 character hexadecimal number
http://us1.php.net/manual/en/function.md5.php
http://www.faqs.org/rfcs/rfc1321.html
SHA1 - 40 character hexadecimal number
http://us1.php.net/manual/en/function.sha1.php
http://www.faqs.org/rfcs/rfc3174.html

Therefore, a string of "apple" could be hashed using MD5 to generate a checksum of "1f3870be274f6c49b3e31a0c6728957f"
Notice the result is a value with hexadecimal numbers (0-9, A-F)

```
$str = 'apple';
if (md5($str) === '1f3870be274f6c49b3e31a0c6728957f') {
    echo "Would you like a green or red apple?";
}

if (sha1($str) === 'd0be2dc421be4fcd0172e5afceea3970e2f3d940') {
    echo "Would you like a green or red apple?";
}
```

How Secure is Hashing?

While hashing is a one-way encryption (meaning it cannot be reversed or "unencrypted" by just reversing the algorithm), it is possible to guess combinations until you find a match.

The easiest way is to use a pre-built dictionary, hash the words and store in a database, then search by hashed value for a match. (These are often called Rainbow Tables.)

How long does it take to find a hashed value?
https://crackstation.net/

```
(md5($str) === '1f3870be274f6c49b3e31a0c6728957f')
(sha1($str) === 'd0be2dc421be4fcd0172e5afceea3970e2f3d940')
```

Salting

http://us1.php.net/manual/en/faq.passwords.php#faq.passwords.fasthash
Because hash values can be guessed (imagine looping through all possible combinations until you find one that works), they are generally considered insecure along for use as storing passwords.

Salting adds an additional value that is unknown to an attempted hacker to make hashes more secure. Concatenate the salt to the string you need to hash.

```
$str = 'apple';
$salt = 'mysectretpassword';
   //without salting
   if (md5($str) === '1f3870be274f6c49b3e31a0c6728957f') {
    echo "valid user";
   }
   //with salting
   if (md5($salt.$str) === 'asda3870be2345kf431a0c6728957f') {
      echo "valid user";
   }
```

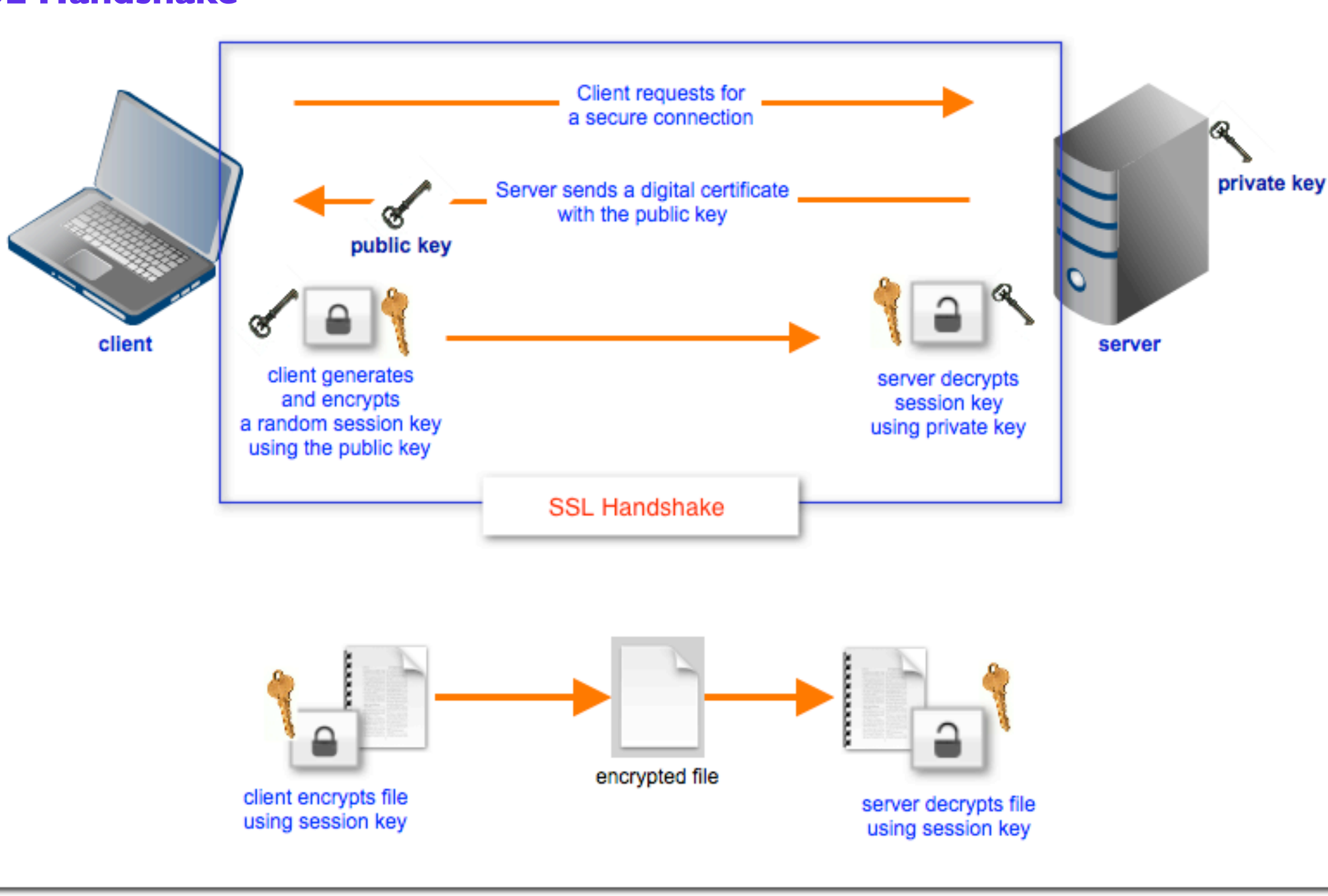
Encryption

http://us2.php.net/manual/en/function.mcrypt-encrypt.php

string mcrypt_encrypt (string $cipher , string $key , string $data , string $mode [, string $iv])

```
<?php
class Cipher {
    private $securekey, $iv;
    function __construct($textkey) {
        $this->securekey = hash('sha256',$textkey,TRUE);
        $this->iv = mcrypt_create_iv(32);
    }
    function encrypt($input) {
        return base64_encode(mcrypt_encrypt(MCRYPT_RIJNDAEL_256, $this->securekey, $input,
MCRYPT_MODE_ECB, $this->iv));
    }
    function decrypt($input) {
        return trim(mcrypt_decrypt(MCRYPT_RIJNDAEL_256, $this->securekey,
base64_decode($input), MCRYPT_MODE_ECB, $this->iv));
    }
}

$cipher = new Cipher('secret passphrase');

$encryptedtext = $cipher->encrypt("hide me");
echo "->encrypt = $encryptedtext<br />";

$decryptedtext = $cipher->decrypt($encryptedtext);
echo "->decrypt = $decryptedtext<br />";

var_dump($cipher);
?>
```

SSL Handshake



Source: http://www.jscape.com/blog/bid/84690/Choosing-Key-Lengths-for-Encrypted-File-Transfers

Create a Self-Signed SSL Certificate for MAMP

Open up Terminal and cd /Applications/MAMP/conf/apache/

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
server.key -out server.crt
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:Florida
Locality Name (eg, city) []:Winter Park
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Full Sail LLC
Organizational Unit Name (eg, section) []:WDDBS
Common Name (e.g. server FQDN or YOUR name) []:localhost
Email Address []:wbunner@fullsail.com
```

For the Common Name, usually would be FQDN (www.hostname.com or subdomain.host.com), otherwise the browser would display a certificate/name mismatch.

This generates a *Private Key* (server.key) and a self-signed *Public Key* (server.crt) in the form of a certificate.

The Private Key is the only key that exists that can decrypt the information - this should be stored in a safe location (NOT in any directory accessible by the web server) and backed up in the case of production websites.

In certificates signed by a trusted 3rd party, a *Certificate Request* (sometimes referred to as CSR) will be sent (not containing the private key), and the 3rd party will sign the request and generate the certificate. Often the 3rd party certificate chain will be included with your certificate so that the root authority is trusted by your browser.

Enable Secure Socket Layer in MAMP

Open up MAMP/conf/apache/httpd.conf

Uncomment the Include line for httpd-ssl.conf

```
548  # Secure (SSL/TLS) connections
549  Include /Applications/MAMP/conf/apache/extra/httpd-ssl.conf
```

Also, configure the location of your DocumentRoot. (You usually set for http under MAMP preferences, for https this needs to be explicitly set as well.)

```
74  <VirtualHost _default_:443>
75
76  #   General setup for the virtual host
77  DocumentRoot "/Applications/MAMP/Library/htdocs"
78  ServerName www.example.com:443
79  ServerAdmin you@example.com
```

MAMP has issues with OS X running on port 443 unless you start it with escalated permissions.
Stop Apache through MAMP control panel.
To restart MAMP again:
sudo /Applications/MAMP/bin/startApache.sh

Connect with your browser to https://localhost and see if the page displays. Check the certificate once the page loads.

Web Page Requests

Traditional HTML requests by nature result in short-term, stateless interactions.

1. Browser open connection to server (IP address, port 80) requests page from web server (HTTP GET Request)

2. Server retrieves or generates HTML response and sends back to browser through IP connection made by initial request (HTTP Response)

3. Server and Browser close IP connection when data transfer is complete.

For EACH HTTP request sent to/received by server these steps take place. This includes EACH .js .css .jpeg .png etc. file that is included on a single web page.

Because each transaction is a standalone request, there is no way for the server to know which client made each request.

Could we track by IP address and client port?
Could we store data on the browser client that could be read back later?

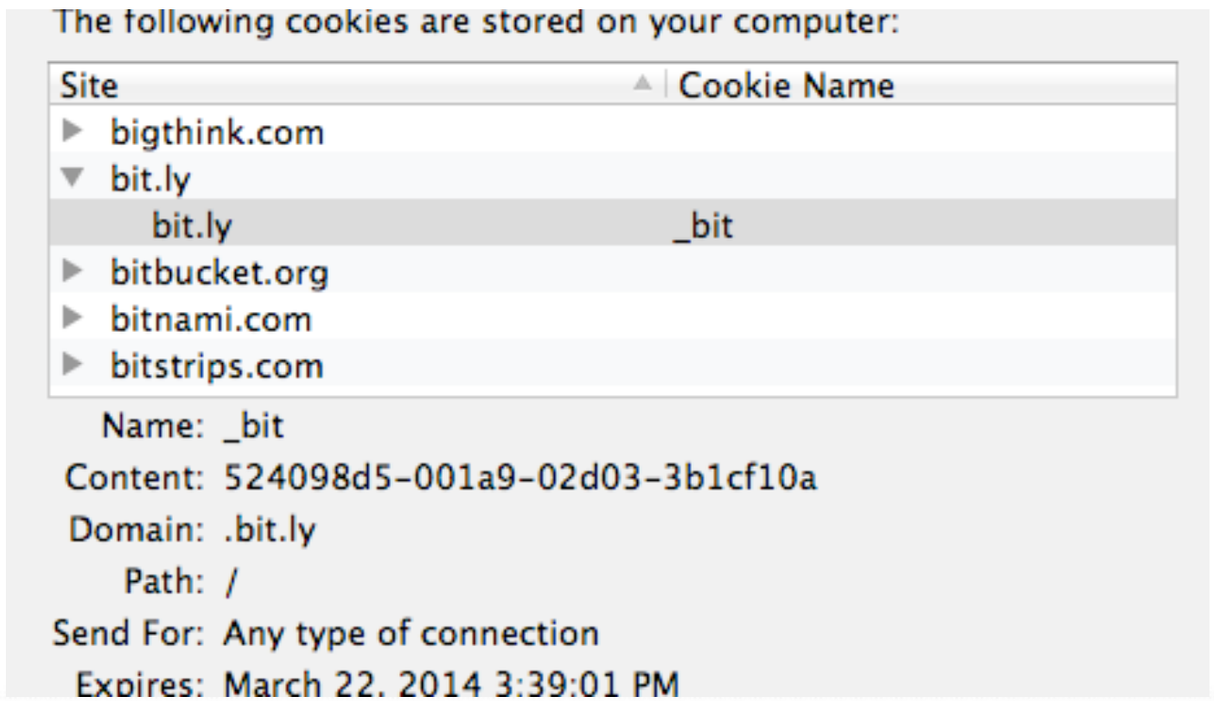
Cookies

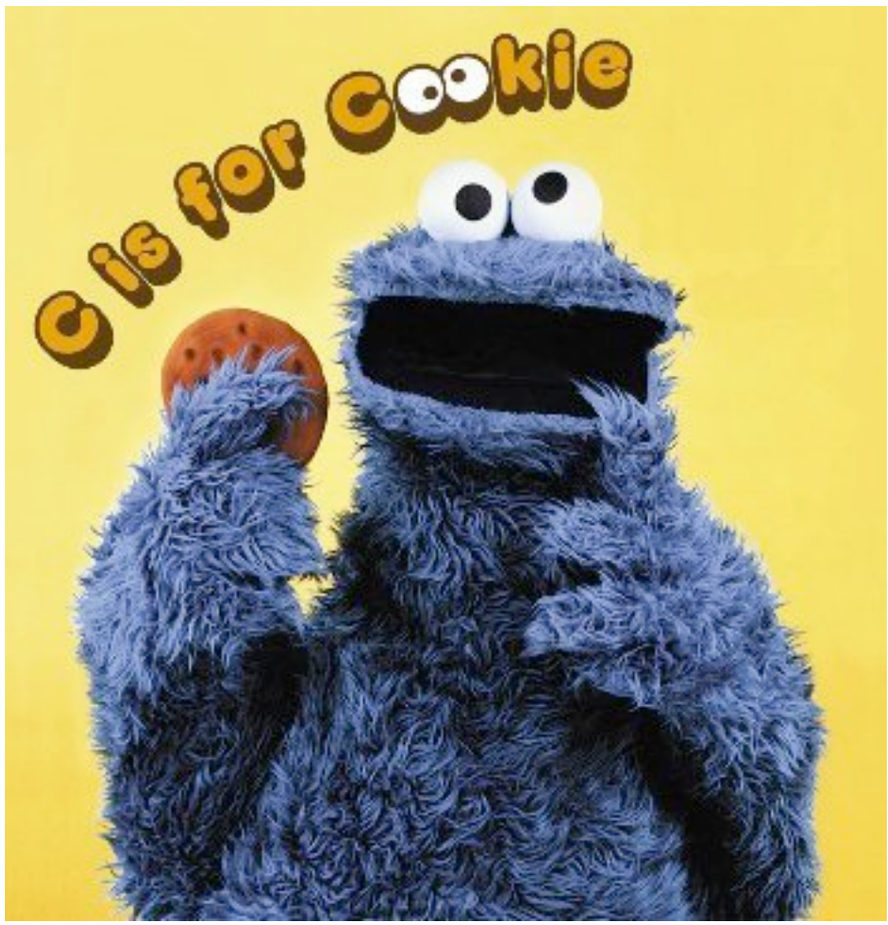
Cookies are small files that are stored locally on the client browser that can be set by a web server and then requested back by the web server later.

Cookies contain a server name (domain), path, expiration date, key value (cookie name) , and data value (content). Third-party cookies are cookies that can be set or retrieved by web sites other than the one you are visiting.

Cookies are stored unencrypted.

(And, unless sent through HTTPS usually transported without encryption.)





http://www.flickr.com/photos/shellewill79/5333262917/sizes/o/
Some rights reserved by Michelle O'Connell Photography

http://youtu.be/-qTlGg3l5y8

Setting Cookies/Deleting Cookies

http://us1.php.net/manual/en/function.setcookie.php

bool setcookie (string $name [, string $value [, int $expire = 0 [, string $path [, string $domain [, bool $secure = false [, bool $httponly = false]]]]]])

```php
$value = 'something from somewhere';
setcookie("TestCookie", $value);
setcookie("TestCookie", $value, time()+3600);  /* expire in 1 hour */

// Print an individual cookie
echo $_COOKIE["TestCookie"];

// Delete Cookie example - set the expiration date to one hour ago
setcookie ("TestCookie", "", time() - 3600);
```

- Cookies will not become visible until the next loading of a page that the cookie should be visible for. To test if a cookie was successfully set, check for the cookie on a next loading page before the cookie expires. Expire time is set via the expire parameter.
- Cookies must be deleted with the same parameters as they were set with. If the value argument is an empty string, or FALSE, and all other arguments match a previous call to setcookie, then the cookie with the specified name will be deleted from the remote client. This is internally achieved by setting value to 'deleted' and expiration time to one year in past.
- Because setting a cookie with a value of FALSE will try to delete the cookie, you should not use boolean values. Instead, use 0 for FALSE and 1 for TRUE.

Exercise: Set and Retrieve Cookies

Using the setcookie() function and $_COOKIE[] superglobal, test setting

1. setcookie.php - Set a cookie value to the browser and echo confirmation.
2. viewcookie.php - Retrieve a cookie from the browser and echo out
3. deletecookie.php - Remove the cookie and echo back confirmation.

Call:
setcookie
viewcookie
deletecookie
then viewcookie again - What happens?
then delete again.

```php
$value = 'something from somewhere';
setcookie("TestCookie", $value);
setcookie("TestCookie", $value, time()+3600);  /* expire in 1 hour */

// Print an individual cookie
echo $_COOKIE["TestCookie"];

// Delete Cookie example - set the expiration date to one hour ago
setcookie ("TestCookie", "", time() - 3600);
```

Sessions

http://us1.php.net/manual/en/book.session.php

Remember that there is no persistence between HTML request/response pairs from the client/web server. Sessions allow some persistence to emulate a connection-oriented environment.

PHP sets default session length in php.ini file (usually 30 minutes, but can vary).

```
session_start();
if (!isset($_SESSION['count'])) {
  $_SESSION['count'] = 0;
} else {
  $_SESSION['count']++;
}
```

```
session_start();
unset($_SESSION['count']); //remove the session variable count

// Finally, destroy the session.
session_destroy();
```

Exercise: PHP Sessions

1. Create two PHP scripts:
setsession.php - Start a session and set session variables based on GET parameters you pass through the querystring. Set at least 3 variables.
getsession.php - Retrieve an existing session and echo out the session variables you set in setsession.
Wait 60 seconds and try to retrieve getsession again - does it still retrieve variables?

2. Add the following line to both scripts.
ini_set('session.gc_maxlifetime', 60); //set the session timeout to 60 seconds
Stop and restart Apache, then run setsession/getsession again.
Wait for over 60 seconds and run getsession again.

```php
session_start();
if (!isset($_SESSION['count'])) {
  $_SESSION['count'] = 0;
} else {
  $_SESSION['count']++;
}
session_start();
unset($_SESSION['count']); //remove the session variable count
// Finally, destroy the session.
session_destroy();
```

Image Manipulation

http://us1.php.net/manual/en/ref.image.php
http://us1.php.net/manual/en/image.examples-png.php
GD is a graphics and image manipulation library that is accessible through PHP.
Image *Resources* (references to objects) are passed to/from functions.

Example #1 PNG creation with PHP

```
<?php

header("Content-type: image/png");
$string = $_GET['text'];
$im     = imagecreatefrompng("images/button1.png");
$orange = imagecolorallocate($im, 220, 210, 60);
$px     = (imagesx($im) - 7.5 * strlen($string)) / 2;
imagestring($im, 3, $px, 9, $string, $orange);
imagepng($im);
imagedestroy($im);

?>
```

This example would be called from a page with a tag like: *<img src="button.php?text=text">*. The above *button.php* script then takes this "*text*" string and overlays it on top of a base image which in this case is "*images/button1.png*" and outputs the resulting image. This is a very convenient way to avoid having to draw new button images every time you want to change the text of a button. With this method they are dynamically generated.

Other graphics libraries such as ImageMagick can be used as well by installing additional libraries and *wrappers* to access these libraries through PHP.

PHP Image Creation

Some of the useful PHP image methods.

```
getimagesize($image_path);
imagecreatefromjpeg($filename);
imagejpeg, imagepng, imagegif();
imagedestroy();
imagecreatetruecolor();
imagecopyresampled();
imagecreate();
imagecolorallocate();
imagerectangle();
imagefttext();
```

```
header('Content-type: image/png');
```

PHP Image Creation

Image copy method

```
public function imageCopy($orgfile, $newfile){

    $n = imagecreatefromjpeg($orgfile);
    imagejpeg($n,$newfile);
    imagedestroy($n);
}
```

$n is a *resource* (or pointer) that contains the binary bitmap contents of the imported image.

PHP Image Creation

Image resize method

```
public function imageResize($orgfile, $newfile, $h, $w){


    $n = imagecreatefromjpeg($orgfile);
    $ar = getimagesize($orgfile);
    $orgw=  $ar[0];
    $orgh = $ar[1];

    $cont = imagecreatetruecolor($w, $h);
    imagecopyresampled($cont, $n, 0, 0, 0, 0, $w, $h, $orgw, $orgh);
    imagejpeg($cont,$newfile,100);
    imagedestroy($n);
}
```

CAPTCHA

http://www.captcha.net/

CAPTCHA is a method to verify that a human is interacting with a site, to avoid automated "bot" submissions from scripts or attempts to SPAM.

The term CAPTCHA (for Completely Automated Public Turing Test To Tell Computers and Humans Apart) was coined in 2000 by Luis von Ahn, Manuel Blum, Nicholas Hopper and John Langford of Carnegie Mellon University.

Trivia: What is a Turing Test?

http://www.turing.org.uk/scrapbook/test.html

PHP Image Creation

Message/Captcha Method

```
public function msg($msg){

    $container = imagecreate(300,200);
    $black = imagecolorallocate($container,0,0,0);
    $white = imagecolorallocate($container,255,255,255);
    $font = 'fonts/texb.ttf';
    imagefilledrectangle($container, 0, 0, 250,150,$black);
    imagerectangle($container,0,0,50,60,$white);
    imagefttext($container,12,0,0,12,$white,$font,$msg);
    header('Content-Type: image/png');
    imagepng($container,null);
    imagedestroy($container);

}
```

Exercise: Image Creation

1. Create a CAPTCHA-type image based on a word using the code from the previous page.
2.Once you have successfully created the image, embed it using an <img> tag on a sample webpage and have your script create the image based on a GET variable.
Ex:

```
public function msg($msg){

    $container = imagecreate(300,200);
    $black = imagecolorallocate($container,0,0,0);
    $white = imagecolorallocate($container,255,255,255);
    $font = 'fonts/texb.ttf';
    imagefilledrectangle($container, 0, 0, 250,150,$black);
    imagerectangle($container,0,0,50,60,$white);
    imagefttext($container,12,0,0,12,$white,$font,$msg);
    header('Content-Type: image/png');
    imagepng($container,null);
    imagedestroy($container);

}
```

Filesystem Functions

http://us2.php.net/manual/en/book.filesystem.php

Filesystem components allow interaction with local files on the webserver, or retrieving remote files through URLs.

```php
<?php
$file = file_get_contents('./people.txt', true);
$homepage = file_get_contents('http://www.example.com/');
echo $homepage;
?>
```

Writing Files

```php
<?php
$file = 'people.txt';
// The new person to add to the file
$person = "John Smith\n";
// Write the contents to the file,
// using the FILE_APPEND flag to append the content to the end of the
file
// and the LOCK_EX flag to prevent anyone else writing to the file at
the same time
file_put_contents($file, $person, FILE_APPEND | LOCK_EX);
?>
```

Today's Lab (Lab 3)

In this lab you will create an upload form for a user to register and upload an avatar image.

- The user should select a username, password, upload a JPEG file as their avatar, and verify a simple one-word CAPTCHA.
- The CAPTCHA should be a random word generated from a dictionary file that you read in from "dictionary.txt". Dictionary.txt should have one word per line and contain at least 10 words. (Hint: You may want to look at the file() function to assist you with this, and trim whitespace.)
- You should resize the image *proportionally* to no more than 100 pixels on the longest side, and save BOTH the original size image and thumbnail image to an ./images folder under your ./SSL/Day3 path.
- Save the captured form information in a Session variable, save the password as a *salted* hashed value.
- Once the user has signed up, redirect them to a "User Profile" page that displays the following information that was stored in the Session variable:
the user name,
hashed password,
and their thumbnail avatar image.

SUBMIT: Source code zipped and uploaded on FSO Lab 3; Link to screencast video explaining code uploaded on FSO Video 3.