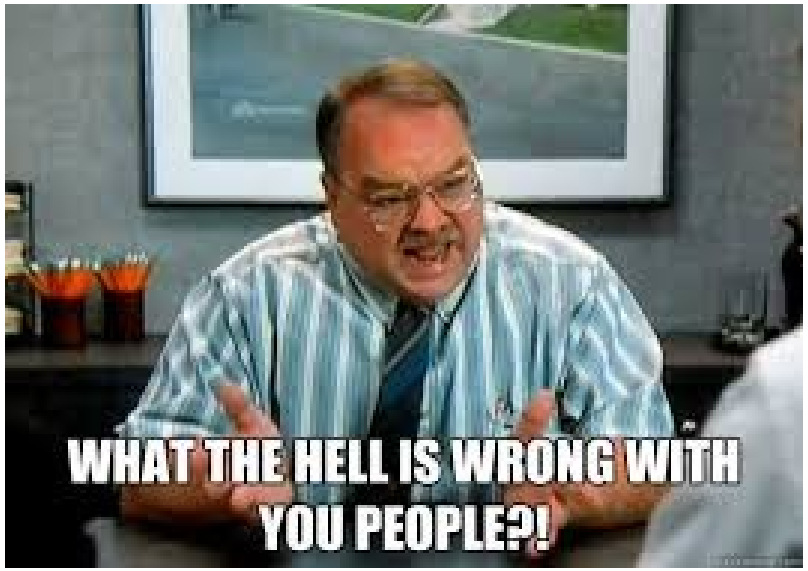
Dealing with users

Dr. Justin F. Brunelle
jbrunelle@cs.odu.edu
http://www.cs.odu.edu/~jbrunelle/cs518

Topics

- Server-side validation
- Error Handling
- Input sanitization



Why do we need input validation?

- Typos
- Malicious Input
- We'll look at validating:
  - Simple string values
  - Integer values
  - Formatted text input

Server vs Client

- Client-side security == no security
  - malicious or broken clients
- JavaScript (client-side) might be disabled

Useful PHP Functions

- empty()
- is_numeric()
- is_bool()
- is_array()
- is_object()

http://us2.php.net/manual/en/ref.var.php

Best practices for error handling

- Generate errors on same page
  - don't load a different URI
- Load separate URI that is just for handling errors
- Reload same page with an error argument

- PHP Error Settings:

```
error_reporting(-1);
ini_set('display_errors', 'On');
```

```
function dump_error_to_file($errno, $errstr) {
   file_put_contents('/tmp/phpErrors',
     date('Y-m-d H:i:s - ')
   . $errstr, FILE_APPEND);
}
set_error_handler('dump_error_to_file');
```

Same URI

```
if (empty($var1)) {
  $errors .= "$var1 should not be empty";
}
if (!is_numeric($var2)) {
  $errors .= "$var2 should be a number";
} // check all anticipated error conditions …
if (empty($errors)){
  // do interesting work
}
else {
  internal_error_function($errors);
}

function internal_error_function ($errors) {
  // generate pretty HTML response
  // provide link to start over
}
```

Different URI

```
if (empty($var1)) {
  $errors .= "$var1 should not be empty";
}
if (!is_numeric($var2)) {
  $errors .= "$var2 should be a number";
} // check all anticipated error conditions …
if (empty($errors)) {
  // do interesting work
}
else {
  $errors = urlencode($errors);
  header("Location: http://foo.edu/error.php?error=$errors");
}
```

URI with Error in Argument

```
if (empty($var1)) {
  $errors .= "$var1 should not be empty";
}
if (!is_numeric($var2)) {
  $errors .= "$var2 should be a number";
} // check all anticipated error conditions
…
if (empty($errors)) {
   // do interesting work
}
else {
  $errors = urlencode($errors);
  header("Location:".$_SERVER["REQUEST_URI"]."?errors=$errors");
}
```

Encode URLs

- RFC1738 requires "unsafe" and "reserved" characters to be encoded in URIs
  - Reserved examples: "/", ":", "?"
  - Unsafe examples: [space], "%", "#"
- PHP urlencode(), urldecode()
- More info, examples, rationales: http://www.blooberry.com/indexdot/html/topics/urlencoding.htm

Standard String Formats

- To do formatted text (e.g., date) validation, use regular expressions
- DD-MM-YYYY format for date
  - [0-9]{2})-([0-9]{2})-([0-9]{4})
  - (2 digits between 0-9
  - "-" character
  - 2 digits between 0-9
  - "-" character
  - 4 digits between 0-9

RegEx

```
if ( !preg_match("/([0-9]{2})-([0-9]{2})-([0-9]{4})/",
$_POST['movie_release'], $reldatepart) )
{
   $error .= "Please+enter+a+date+with+the+dd-mm-yyyy+format";
}
```

- if $_POST['movie_release'] is 31-05-1969 then:
  - $reldatepart[0] = 31-05-1969
  - $reldatepart[1] = 31
  - $reldatepart[2] = 05
  - $reldatepart[3] = 1969

Date Time

- Change the date string into a timestamp (to store in the database) using mktime()
- Takes as parameters: hour, min, seconds, month, day, year
- If valid date, returns the number of seconds since Jan 1, 1970
- If not a valid date (e.g., 99-99-9999), returns -1

```
$movie_release = mktime (0, 0, 0,
$reldatepart['2'], $reldatepart['1'],
$reldatepart['3']);
```

...and with MySQL

- UNIX_TIMESTAMP()
  - takes YYYY-MM-DD HH:MM:SS format string
  - creates a timestamp

```
$reldate = $reldatepart['3'] . "-" . $reldatepart['2'] . "-" .
     $reldatepart['1'] . " 00:00:00";
$sql = "INSERT INTO movie_main (release_date) " .
     "VALUES (UNIX_TIMESTAMP('$reldate'))";
```

Handling HTML

- htmlentities(), html_entity_decode()

```
<!--?php
$orig = "I'll \"walk\" the <em>dog</em> now";
$a = htmlentities($orig);
$b = html_entity_decode($a); echo $a; // I'll
&quot;walk&quot; the &lt;em&gt;dog&lt;/em&gt; now
echo $b; // I'll "walk" the <em>dog</em> now
?-->
```

Configuring Apache (httpd.conf)

- No access to httpd.conf?
  - put ErrorDocument directives in .htaccess

ErrorDocument 404 "<h1>Sorry, nothing here</h1>"

ErrorDocument 404 /404.html

```
# # Customizable error responses come in three flavors:
# 1) plain text 2) local redirects 3) external redirects
#
# Some examples:
#ErrorDocument 500 "The server experienced an error."
#ErrorDocument 404 /missing.html ErrorDocument 404 /error.php?
404
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
#
...
```

.htaccess

- Embedded protection of files
- Handled by apache
- http://www.htaccess-guide.com/how-to-use-htaccess/

Useful tools

- JavaScript input validation
  - http://www.codeave.com/javascript/category.asp?u_category=Forms
- SQL injection attacks
  - https://dev.mysql.com/doc/refman/5.5/en/information-functions.html
  - http://www.veracode.com/security/sql-injection
- PHP Regex
  - http://us2.php.net/regex

Emailing from PHP

- mail($to,$subject,$body,$header);
- HTML or text
- Headers!!

```
MIME-Version: 1.0\r\n
Content-type: text/html; charset=iso-8859-1\r\n
From: Apache Error <host@yourdomain.com>\r\n
```

http://www.cs.odu.edu/~jbrunelle/cs518/examples/mailtest.php