Security Module: Input Validation

Description

Input validation is a technique for validating all input before using it. An application that assumes input from the user is valid opens itself up to security vulnerabilities. Issues with data consistency render a system's data useless. User input that queries a database may contain malicious SQL that can alter the database in undesirables ways.

Objective

The objectives of this security module are to illustrate and explain the importance of input validation and the consequences that not validating data can have on a your system.

Activities

There are two activities for this module. This module is targeted at a server-side development class that uses PHP. The first activity is a discussion assignment over input validation and methods that can be used to achieve input validation. Second, students are given a PHP script that simulates a sign-up form for an online account. Students are asked to test the form and add input validation logic.

Module Contents

- Assignment 1: Input Validation Discussion
- Assignment 2: Add Proper Validation
- Source code for InputValidation script
Assignment 1: Input Validation Discussion

Input Validation

Input validation is a term used to describe the process of validating all input before using it. An application that assumes input from the user is valid opens itself up to security vulnerabilities. Issues with data consistency render a system’s data useless. User input that queries a database may contain malicious SQL that can alter the database in undesirable ways. Input validation starts by defining the data that will be collected from users. By defining the data, developers can validate the input data to ensure that it meets specifications.

The following is an example specification used for defining a sign-up form.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>Text</td>
<td>Yes</td>
<td>Consist of characters a-z, A-Z, and 0-9 Length between 6 and 20 (inclusive)</td>
</tr>
<tr>
<td>Email Address</td>
<td>Text</td>
<td>Yes</td>
<td>Must be a valid email <a href="mailto:Username@domain.com">Username@domain.com</a></td>
</tr>
<tr>
<td>First Name</td>
<td>Text</td>
<td>Yes</td>
<td>Consist of characters a-z and A-Z Length between 2 and 24 (inclusive)</td>
</tr>
<tr>
<td>Last Name</td>
<td>Text</td>
<td>Yes</td>
<td>Consist of characters a-z and A-Z Length between 2 and 24 (inclusive)</td>
</tr>
<tr>
<td>Age</td>
<td>Number</td>
<td>Yes</td>
<td>Number that must be between 18 and 110 inclusive Must be 18 or over</td>
</tr>
<tr>
<td>Address Line 1</td>
<td>Text</td>
<td>Yes</td>
<td>Consist of characters a-z and A-Z Length between 2 and 48 (inclusive)</td>
</tr>
<tr>
<td>Address Line 2</td>
<td>Text</td>
<td>No</td>
<td>Consist of characters a-z and A-Z Length between 2 and 48 (inclusive)</td>
</tr>
<tr>
<td>City</td>
<td>Text</td>
<td>Yes</td>
<td>Consist of characters a-z and A-Z Length between 2 and 48 (inclusive)</td>
</tr>
<tr>
<td>US Zipcode</td>
<td>Integer</td>
<td>Yes</td>
<td>Consist of numbers Length equal to 5</td>
</tr>
<tr>
<td>State</td>
<td>Text</td>
<td>Yes</td>
<td>Must be one of the 50 states</td>
</tr>
</tbody>
</table>

Table 1: Sign-up form specification

The following questions can be answered using the above narrative, specification table, and online PHP documentation.
1. What is the benefit of creating a specification for form data?

2. What PHP function can be used to check the length of a variable?

3. What PHP function can be used to perform a regex match? What is the regex to check for a-z, A-Z, or 0-9?

4. What PHP function can be used to check if variable is a number?

5. What PHP array function can be used to check if a value is in an array?
Assignment 2: Add Proper Validation

In this assignment, you will be given access to a fake sign-up form that will require certain information. Once you submit the form, the information will be presented back to you as you entered it. The form does no validation checks on the data provided. Your assignment is to add proper validation logic and thoroughly test against the form specifications.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>Text</td>
<td>Yes</td>
<td>Consist of characters a-z, A-Z, and 0-9 Length between 6 and 20 (inclusive)</td>
</tr>
<tr>
<td>Email Address</td>
<td>Text</td>
<td>Yes</td>
<td>Must be a valid email <a href="mailto:Username@domain.com">Username@domain.com</a></td>
</tr>
<tr>
<td>First Name</td>
<td>Text</td>
<td>Yes</td>
<td>Consist of characters a-z and A-Z Length between 2 and 24 (inclusive)</td>
</tr>
<tr>
<td>Last Name</td>
<td>Text</td>
<td>Yes</td>
<td>Consist of characters a-z and A-Z Length between 2 and 24 (inclusive)</td>
</tr>
<tr>
<td>Age</td>
<td>Number</td>
<td>Yes</td>
<td>Number that must between 18 and 110 inclusive Must be 18 or over</td>
</tr>
<tr>
<td>Address Line 1</td>
<td>Text</td>
<td>Yes</td>
<td>Consist of characters a-z and A-Z Length between 2 and 48 (inclusive)</td>
</tr>
<tr>
<td>Address Line 2</td>
<td>Text</td>
<td>No</td>
<td>Consist of characters a-z and A-Z Length between 2 and 48 (inclusive)</td>
</tr>
<tr>
<td>City</td>
<td>Text</td>
<td>Yes</td>
<td>Consist of characters a-z and A-Z Length between 2 and 48 (inclusive)</td>
</tr>
<tr>
<td>US Zipcode</td>
<td>Integer</td>
<td>Yes</td>
<td>Consist of numbers Length equal to 5</td>
</tr>
<tr>
<td>State</td>
<td>Text</td>
<td>Yes</td>
<td>Must be one of the 50 states</td>
</tr>
</tbody>
</table>

Table 2: Sign-up form specification

1. Test your solution to ensure that it works.

2. Turn in a screen shot of the sign-up form with invalid data showing that the form accepts invalid data.

3. Turn in a screen shot of the sign-up form with valid data.
Input Validation Source Code

```php
<?php


```

```
```

```
```

```html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

```php
<?php

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
First Name: 
Last Name: 
Age:
Address Line 1: 
Address Line 2: 
City:
US Zipcode:
State:

Username
Email Address
First Name
Last Name
Age

<!— Else show the form -->
<?php else: ?>
<form action="" id="subscription-form" method="post" style="width: 400px; margin: 100px auto 0 auto;">
<label for="username">Username</label><br />
<input id="username" name="username" size="16" /><br /><br />
<label for="email">Email Address</label><br />
<input id="email" name="email" size="16" /><br /><br />
<label for="firstname">First Name</label><br />
<input id="firstname" name="firstname" size="16" /><br /><br />
<label for="lastname">Last Name</label><br />
<input id="lastname" name="lastname" size="16" /><br /><br />
<label for="age">Age</label><br />
<input id="age" name="age" size="8" /><br /><br />
<label for="address1">Address Line 1</label><br />
<input id="address1" name="address1" size="32" />
<br />
<label for="address2">Address Line 2</label><br />
<input id="address2" name="address2" size="32" />
<br />
<label for="city">City</label><br />
<input id="city" name="city" size="16" />
<br />
<label for="zipcode">US Zipcode</label><br />
<input id="zipcode" name="zipcode" size="16" />
<br />
<label for="state">State</label><br />
<select id="state" name="state">
  <?php foreach ($states as $state): ?>
    <option value="<?php echo $state ?>"><?php echo $state ?></option>
  <?php endforeach; ?>
</select>
<br />
<input class="submit" type="submit" value="Submit" />
</form>
<?php endif; ?>
</div>
</body>
</html>