GAZI UNIVERSITY
FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING

SECURE CODING

ASSIGNMENT-1

03-03-2019

PHP Secure Login

Student Number : 141180033
Name : Burak IŞIK
E-Mail : buraksk94@gmail.com

Introduction

In this assignment, A secure login screen will be created with PHP. The purpose is prevent some attacks such as Cross Site Scripting , SQL Injection and brute force when users when users are entering form inputs.Functions such as strip_tags(), trim(), filter_var(), htmlspecialchars(), utf8_decode(), htmlentities(), stripslashes(), get_magic_quotes_gpc() and mysqli_real_escape_string() will be used to prevent these attacks and create a secure login screen.

Reminding: In some cases, the input control is deliberately bypassed from the database in order to clearly show the effect of the functions.

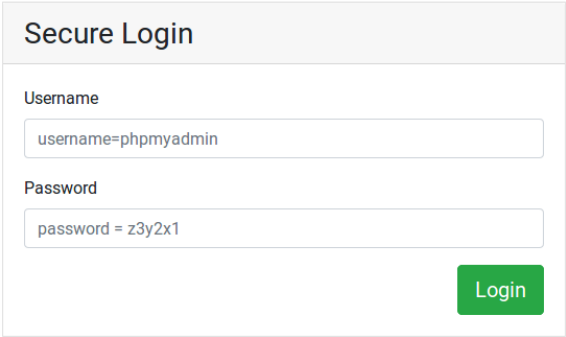
1. strip_tags()

The strip_tags() function strips a string from HTML, XML, and PHP tags.

Usage in my codes

```
static function my_strip_tags($variable) {
        return strip_tags($variable);
 }
```

In the below screen shot is my login page



{"statusCode":404,"errorMessage":"User not found"}

when user is enter his/her username as " burak" the output will be will be as below.

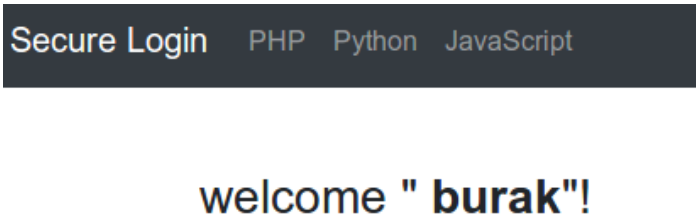


Figure 1.

If strip_tag() function is used when perform input validation, the output will be as below.

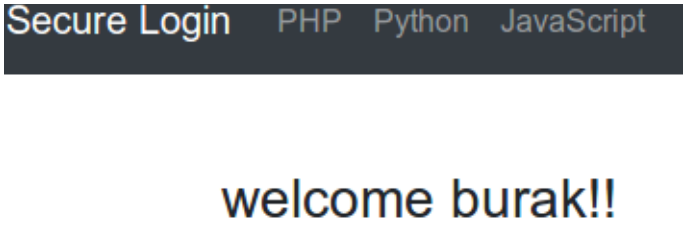


Figure 2.

2. trim()

The trim() method removes whitespace from both sides of a string.

Usage in my code

```
static function my_trim($variable) {
        return trim($variable);
 }
```

if user press space key on the keyboard before enter the user name, There will be space between hello and burak as figure 3. If we use trim the output will be as figure 2.

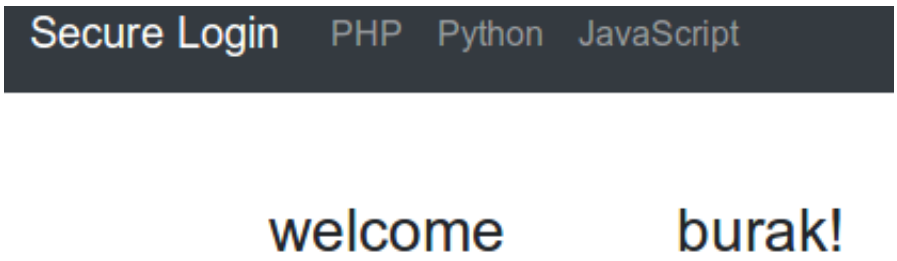


Figure 3.

3. filter_var()

The filter_var() function filters a variable with the specified filter.

Usage in my code

```
static function my_filter_var($variable) {
        return filter_var($variable, FILTER_SANITIZE_STRING);
 }
```

if user enter the username as "<h1>burak!</h1>" , the output will be as figure 4.

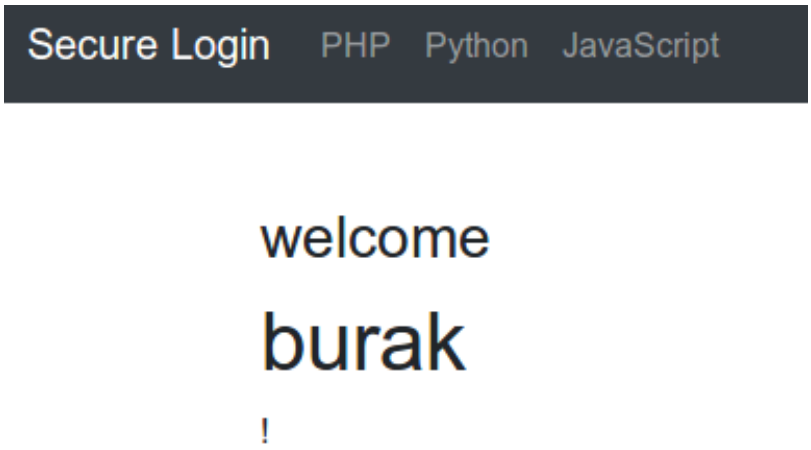


Figure 4.

if we use my_filter_var($p) function the output will appear as figure 2.

4. htmlspecialchars()

The htmlspecialchars() function converts some predefined characters to HTML entities.

Usage in my code

```
static function my_htmlspecialchars($variable) {
        return htmlspecialchars($variable);
 }
```

if user enter the username as "<u>burak<//u>" , the output will be as figure 5.

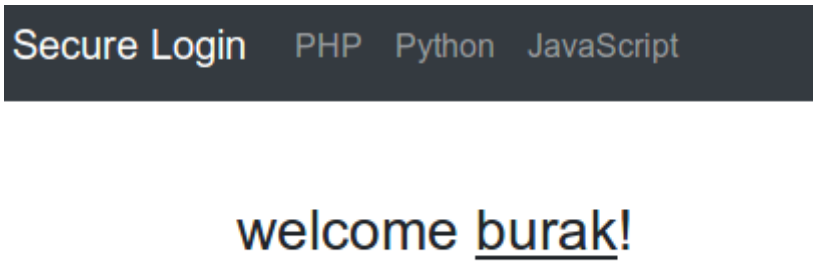
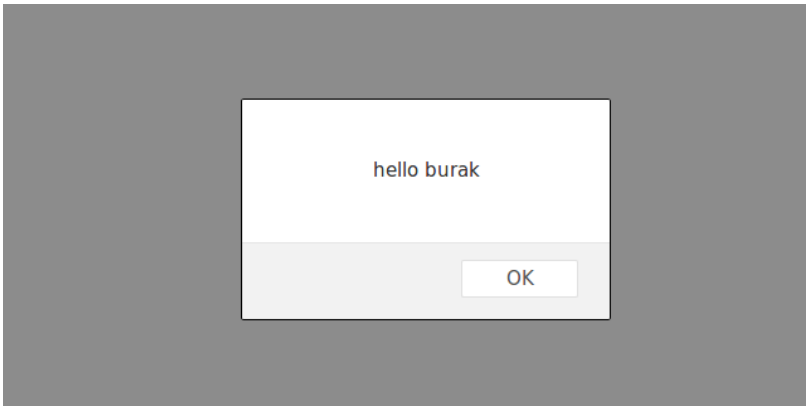


Figure 5.

when my_htmlspecialchars($p) function is used, the output will appear as figure 2.

if input enter as <script type="text/javascript"> alert("hello burak"); </script> the output will appear as below.



5. utf8_decode()

The utf8_decode() function decodes a UTF-8 string to ISO-8859-1.

Usage in my code

```
static function my_utf8_decode($variable) {
        return utf8_decode($variable);
 }
```

if user enter the username as "Déjà vu;" , the output will be as figure 6.

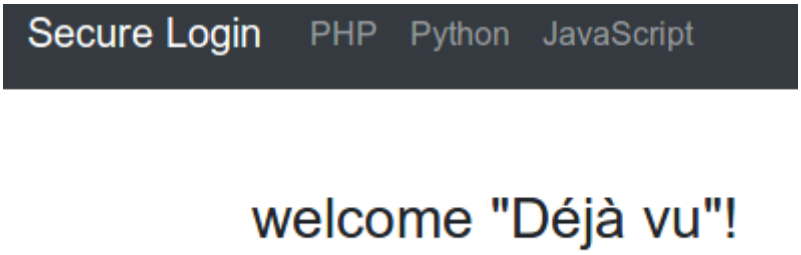


figure 6

if my_utf8_decode($p) function is used, the output will be as figure 7.

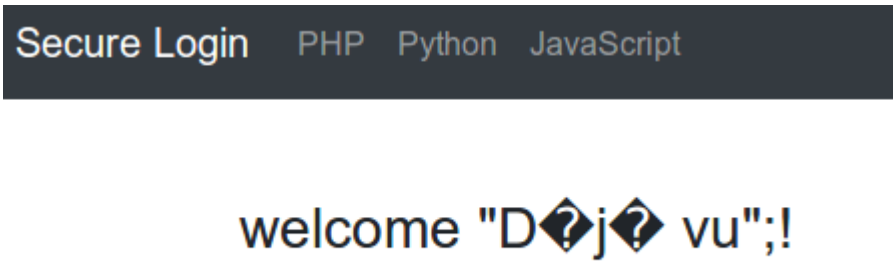


figure 7.

6. htmlentities()

Reserved characters in HTML replaced with character entities.

Usage in my code

```
static function my_htmlentities($variable) {
        return htmlentities($variable);
 }
```

if the user enter the username as 'maninthebit'; , the output will be as figure 8.

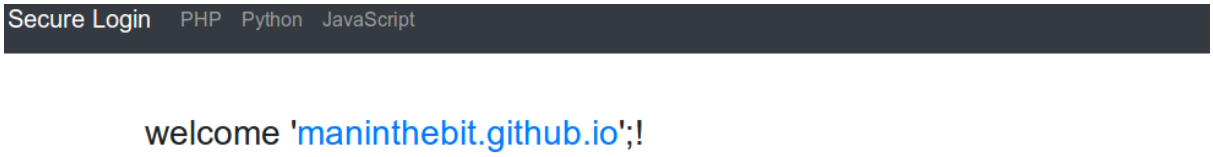


figure 8.

if my_htmlentities($p) function is used, the output will be as figure 9.

figure 9.

7. stripslashes()

The stripslashes() function removes backslashes added by the addslashes() function.

Usage in my code

```
static function my_stripslashes($variable) {
        return stripslashes($variable);
 }
```

if the user enter the username as "who don\'t like burak?" , the output will be as figure 10.

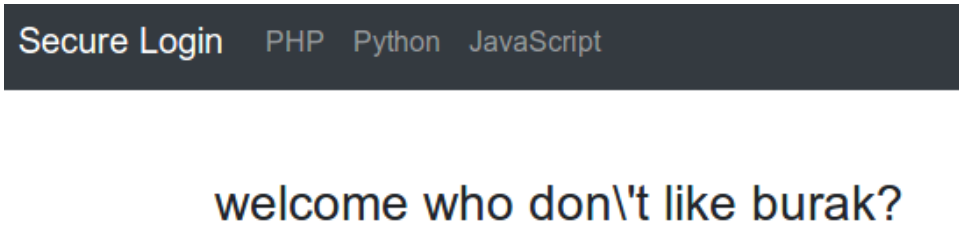


Figure 10.

if my_stripslashes($p) function is used, the output will be as figure 11.

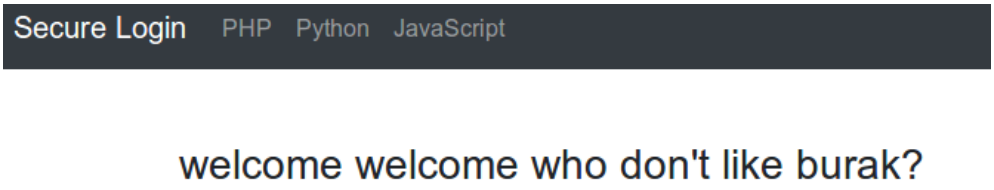


Figure 11.

8. addslashes()

The addslashes() function returns a string with backslashes in front of predefined characters.

Usage in my code

```
static function my_addslashes($variable) {
        return addslashes($variable);
 }
```

if the user enter the username as Who\'s is that? , the output will be as figure 12.

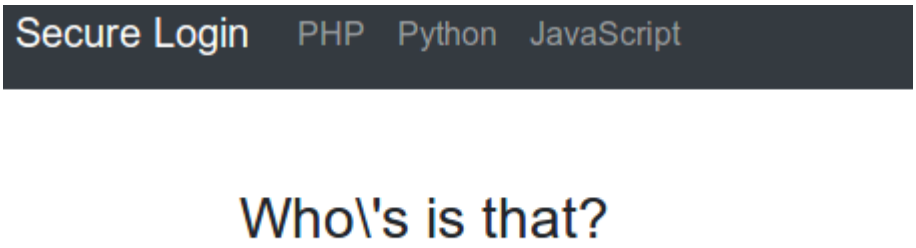


Figure 12.

if my_addslashes($p) function is used, the output will be as figure 13.

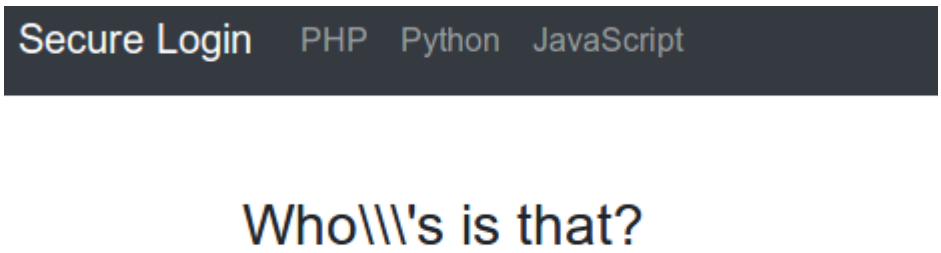


Figure 13.

9. get_magic_quotes_gpc()

get_magic_quotes_gpc() is a function that checks the configuration (php.ini) and returns 0 if magic_quotes_gpc is off (otherwise it returns 1). When magic_quotes are on, all ' (single-quote), " (double quote), \ (backslash) and NULs are escaped with a backslash automatically.

Usage in my code

```
static function my_get_magic_quotes_gpc($variable) {
        return get_magic_quotes_gpc($variable);
}
```

9. mysqli_real_escape_string()

The mysqli_real_escape_string() function escapes special characters in a string for use in an SQL statement.

Usage in my code

```
static function my_mysqli_real_escape_string($conn, $variable) {
        return mysqli_real_escape_string($conn, $variable);
}
```

when the user enter password as "' OR '='", this string bypass the login control. No matter password is true or false;