Quiz 28: Security SOLUTION

CSCI 110 Section 1

Friday, November 18, 2016

1) Is the following web application code vulnerable to an XSS attack? If so, how should it be fixed? [30 points]

```
import cgi

# Assume page_header and page_footer are defined here

class MainPage(webapp.RequestHandler):

  def render_string(self, s):
    self.response.out.write(s)

  def get(self):
    self.response.headers.add_header("X-XSS-Protection", "0")

    if not self.request.get('query'):
      # Show main search page
      self.render_string(page_header + main_page_markup + page_footer)
    else:
      query = self.request.get('query', '[empty]')

      # Our search engine broke, we found no results :-(
      message = ("Sorry, no results were found for <b>"
         + cgi.escape(query) + "</b>.")
      message += " <a href='?'>Try again</a>."

      # Display the results page
      self.render_string(page_header + message + page_footer)

    return
```

No, it is not vulnerable. The user input in the variable 'query' gets escaped via the call to cgi.escape() before being put in HTML.

2) Is the following web application code vulnerable to SQL injection? If so, how should it be fixed? [30 points]

```
// Get the 'var' parameter out of the POST request
$var = $_POST['var'];

// Use the value of that parameter in an SQL query
mysql_query("SELECT * FROM sometable WHERE id = $var");
```

Yes, it is vulnerable. User input in the variable '$var' does not get escaped before being put in an SQL query. To fix this, escape the value of $_POST['var'] before storing it in the variable $var:

```
$var = mysql_real_escape_string($_POST['var']);
```

3) In 6 words or fewer, describe what the following function does. [40 points]

```
int mode(int a[]) {
    int currMode = -1;
    int maxCount = -1;

    for (int i = 0; i < a.length; ++i) {
        int count = 0;
        for (int j = 0; j < a.length; ++j) {
            if (a[j] == a[i]) ++count;
        }
        if (count > maxCount) {
            maxCount = count;
            currMode = a[i];
        }
    }

    return aVal;
}
```

"calculates the mode of an array"

I've improved the variable names for clarity. This is an O(n^2) implementation. Can you think of a way to do it in O(n) time with a HashMap?