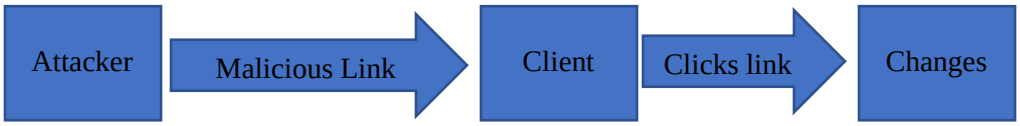
William Burkes
Test2

CSC441 Test1 Instructor: Beddhu Murali 3/23/2021

Max points 100. All questions carry equal weight

(1) True or False. XSS (Cross Site Scripting) is worse than CSRF (Cross Site Reference Forgery).
True

(2) Describe CSRF problem. Draw a diagram to illustrate.
CSRF is an attack which tricks an end user into doing actions they don't intend to do usually by clicking a link.



(3) Describe how would you protect against CSRF
REST and Anti-Frogery Tokens are some of the best ways to prevent CSRF attacks

(4) Use JavaScript frameworks such as Reactjs in combination with Webpack, makes it nearly impossible to use meta tags for serving CSRF token. However, you can use a fetch GET request to get the CSRF token before making a POST request. Assume that the server sent a CSRF token in a header named x-csrf-token. Write code to get this token from the header and save it in a variable (preferably in the Redux store).

```
async function getData(url = '', data = {}) {
    try {
      const response = await fetch(url, {
        method: 'GET',
        credentials: 'same-origin'
      });
      var cookie = response.headers.get('Set-Cookie');
      var x_csrf_token = response.headers.get('X-CSRF-Token');
      return {cookie, x_csrf_token};
    } catch (e) {
      console.log('postData:Exception: ', e);
    }
  }
```

(5) Write code to send the x-csrf-token using a header to the server.

```
return rp({
    method: 'GET',
    url: "https://test.url/testurl",
    headers: {
       'X-CSRF-Token': "yv34nYKZTUH63nEvuQ1bf48E5KLHz8scspsqDHgm"
    }
}).then(function (html) {
})
```

(6) Write code to protect a path on the server from CSRF attacks. You can assume the approach and npm modules discussed in the lectures as available.

```
app.post('/signup', csrfProtection, (req, res) => {
    const body = req.body;
    console.log('body: ', JSON.stringify(body))
    res.send(`${JSON.stringify(body)}\n`)
  })
```

(7) What is the purpose of express-session. What are your choices to store data using expresssession (on or off the server)? Which one you would prefer.

Express-session is used to preserve and store states and data. There needs to be a healthy mix of data that is stored on the server and in the client's browser each session to maintain a healthy balance of performance and responsiveness. I wouldn't prefer one over the other, but more of using them both in a way to maximise effeciency.

(8) Write a express post handler for the url '/echo' that simply returns (echoes) the data sent if it is json and sends a error message in json format if the data sent is not json.

```
app.post('/', function (req,res) {
      try{
        JSON.parse(req.body);
      } catch (e) {
          return false;
      }
      return res.body;
  })
```

(9) Write an express middleware function that validates a telephone number and sends an error message and blocks further processing if the telephone number is invalid. Otherwise, processing should proceed further.

```
app.get('/', function(req, res){
   let re = /(\d[\s-]?)?[\(\[\s-]{0,3}?\d{3}[\)\]\s-]{0,2}?\d{3}[\s-]?\d{4}/g;
    if (re.test(req.body)){
        return res.body;
    } else {
        return false;
    }
})
```

(10) What is express-static? How would you use it? Give a code sample using express-static.

Express-static serves static files from the server to the client through middleware functions in Express. The best use cases are for images, CSS, and Javascript files that aren't changing client-side.

```
var express = require('express');
var app = express();

app.use(express.static('public'));
app.use(express.static('images'));
```