Cybersecurity — Cross Site Request Forgery Lab

Complete the following tasks on ***YOUR*** own computer, not the VM. You will need to have Node.js installed on your computer as well as a text editor of your choice. Node.js download: https://nodejs.org/en/download/ For your submission, answer any questions asked throughout the lab and state any observations. Submit both your answers and zip file.

1. Download the zip file attached to this assignment. It has the code that I wrote during the demo for executing an attack using a GET request.
2. Open your terminal and cd into the lab folder.
3. Enter **npm install** to install all dev dependencies.
4. Enter **npm start** to start server.
5. Open your web browser to http://localhost:3000/ for the **target site.** Login as **bob** (password is **test**). Open your web browser to http://localhost:3001/ for the **malicious site**.
   a. Look at the html pages for both. What is the current balance on the target site?
   b. Refresh the malicious page, then refresh the target page. What happened to the balance? Why?
6. Look through the other files, particularly **server.js**. Where is the vulnerability for a CSRF attack?
7. In the *evil-examples.html* file, comment out or remove the malicious img tag and malicious link.
8. Using JavaScript, conduct a CSRF attack via the *evil-examples.html* file. You should steal $10 from bob's account and put it in alice's account. Enter your malicious JavaScript inside the ***<script></script>*** tags. Refer back to the lecture on CSRF as a reference on how to write your code to execute the attack.

- **IMPORTANT:** Look at ***app.post('/transfer' …*** in the **server.js** file. This will tell you what to assign **action** in your form as well as what you should call the **name** attributes in your form.

- Remember, with a POST request, the function should automatically submit once a user opens the web page. The text inputs should be hidden from the user.

9. Briefly explain one countermeasure you could implement in the web browser to prevent a CSRF. You don’t need to implement the countermeasure.
10. Submit your answers and updated program as a zip file.