### Assignment 4: Cross Site Script Inclusion (XSSI) Attacks

Consider a banking web site bank.com where after login the user is taken to a user information page

https://bank.com/accountInfo.html

The page shows the user's account balances. Here accountInfo.html is a static page: it contains the page layout, but no user data. Towards the bottom of the page a script is included as:

<script src="//bank.com/userdata.js"> (\*)

The contents of userdata.js is as follows:

displayData({"name": "John Doe",

"AccountNumber": 12345,

"Balance": 45})

The function displayData is defined in accountInfo.html and uses the provided data to populate the page with user data.

The script userdata.js is generated dynamically and is the only part of the page that contains user data. Everything else is static content.

Suppose that after the user logs in to his or her account at bank.com the site stores the user's session token in a browser cookie.

1. Consider user John Doe who logs into his account at bank.com and then visits the URL https://evil.com/. Explain how the page atevil.com can cause all of John Doe's data to be sent to evil.com. Please provide the code contained in the page at evil.com.
2. How would you keep accountInfo.html as a static page, but prevent the attack from part (a)? You need only change line (\*) anduserdata.js. Make sure to explain why your defense prevents the attack.   
   **Hint:** Try loading the user's data in a way that gives bank.com access to the data, but does not give evil.com access. In particular,userdata.js need not be a Javascript file.