

Web Exercises

Part 2: CSRF

Barton P. Miller

Computer Sciences Department
University of Wisconsin

bart@cs.wisc.edu

Elisa Heymann

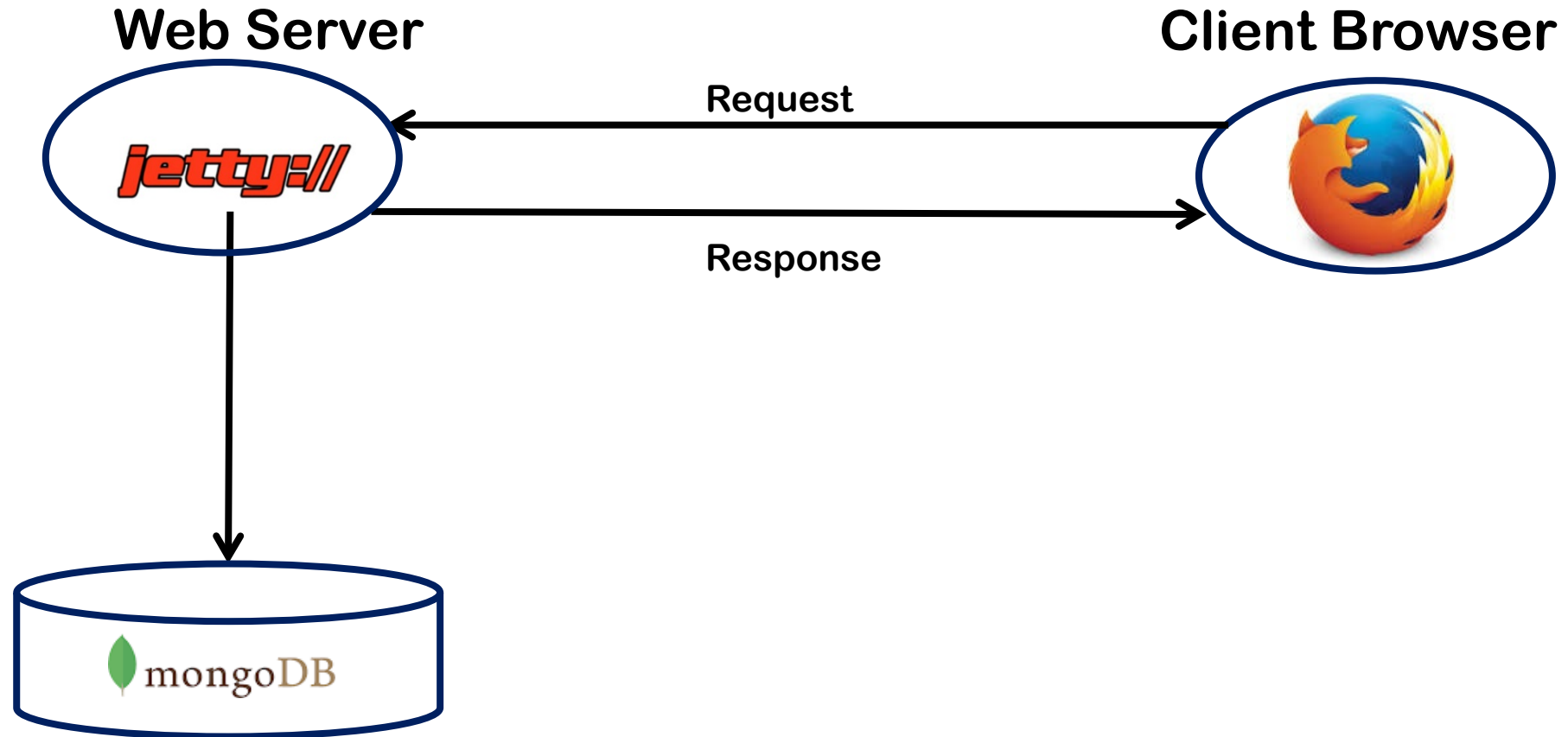
Computer Sciences Department
University of Wisconsin
Universitat Autònoma de Barcelona

elisa@cs.wisc.edu

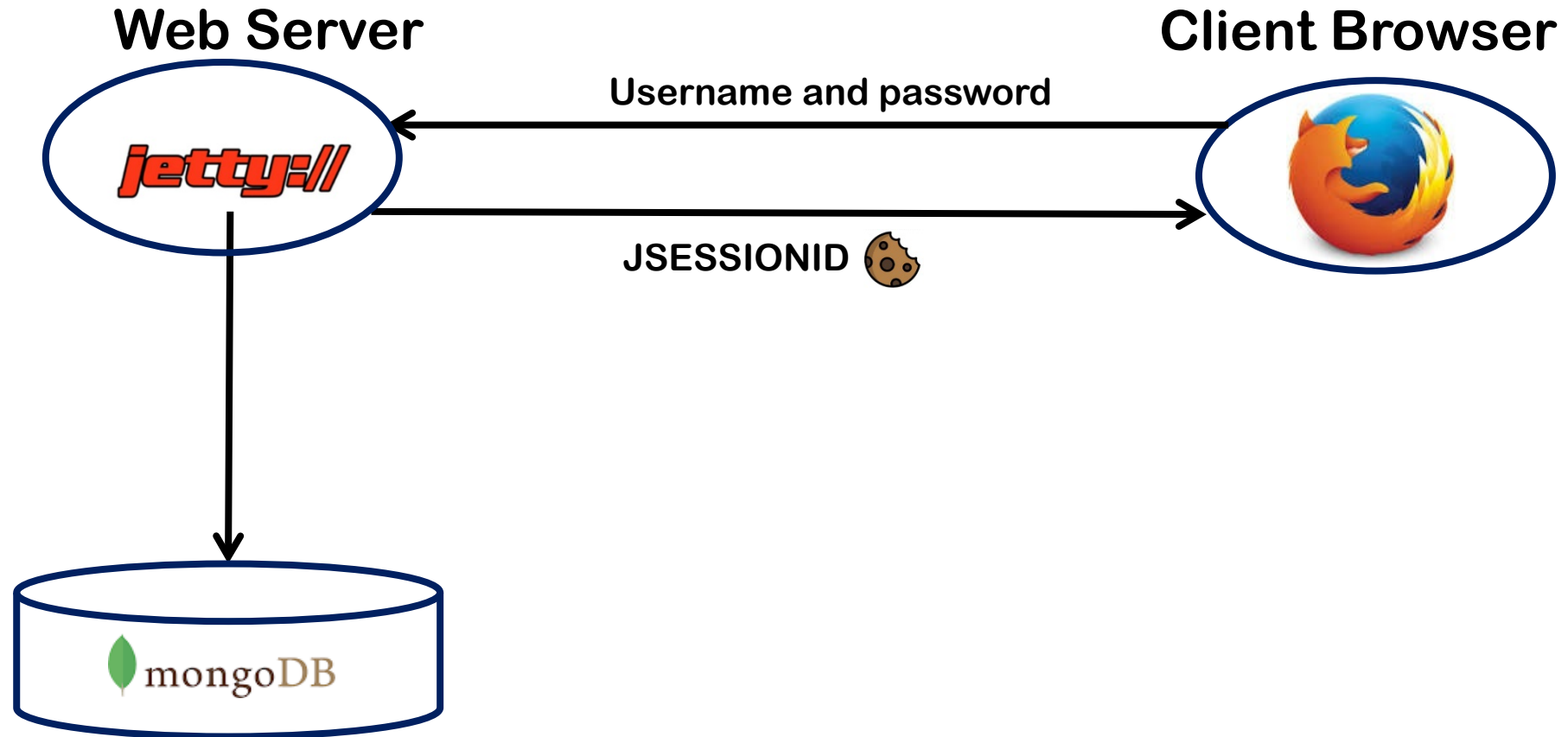
WisClick. Our Web Application

- **A simple game.**
 - Players click on a button to earn credits.
 - Users can manage their profiles.
 - Users can transfer (some of) their credits to others.
 - There is a top five ranking.
 - Users can view other users' page through the top five ranking.
- **Vulnerable to web attacks.**
- **Exploit those vulnerabilities.**

WisClick



WisClick



WisClick. Our web application

You will use two accounts:

username: **attacker**

password: **theattacker**

username: **victim**

password: **thevictim**

To reset the database at anytime run:

```
cd ~/Web_Attack/src  
mongo ResetMongo.js
```

Run WisClick

On a console run the web server:

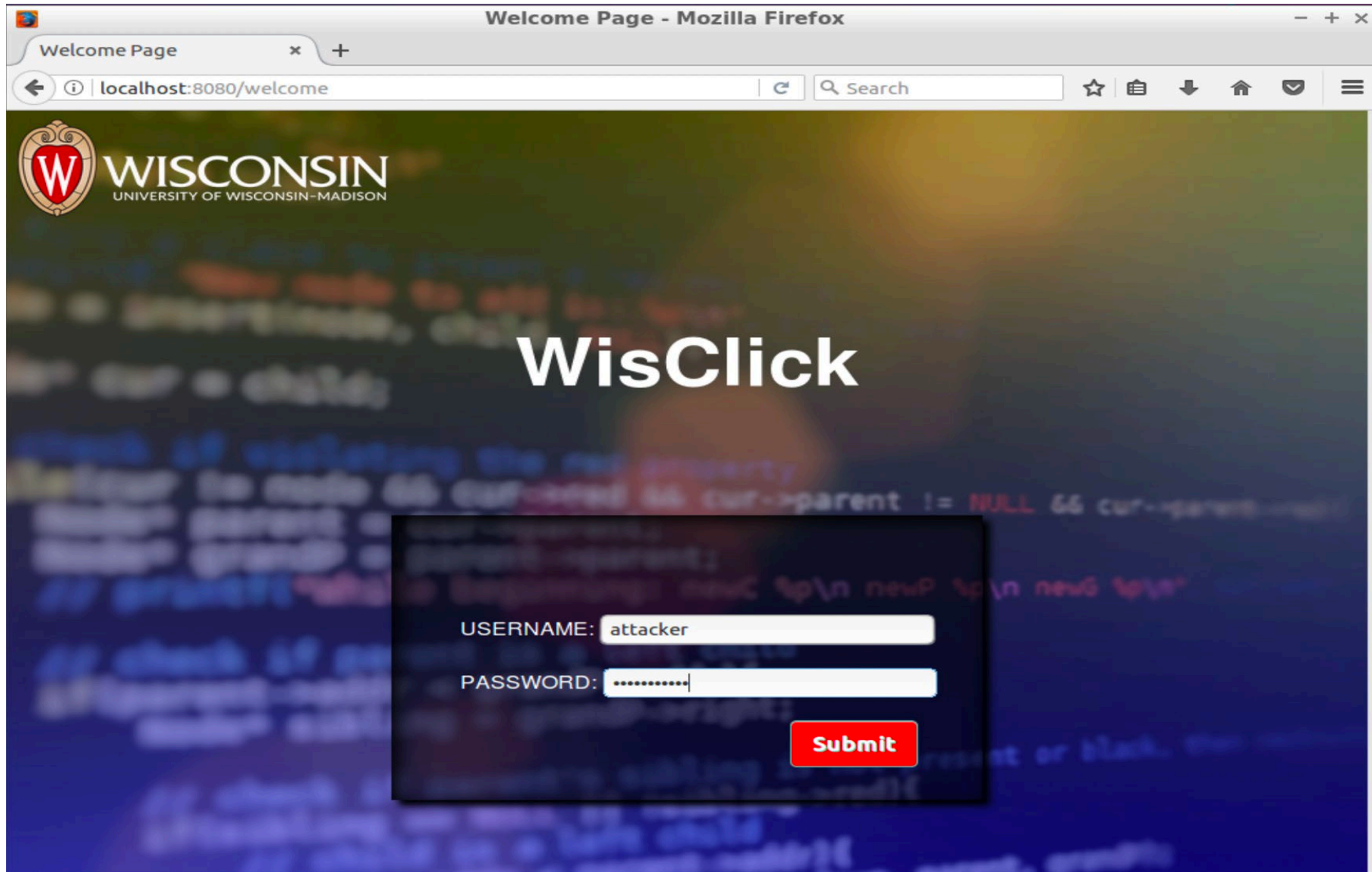
```
cd Web_Attack  
ant  
cd build/classes  
./run.sh
```

On your web browser go to:

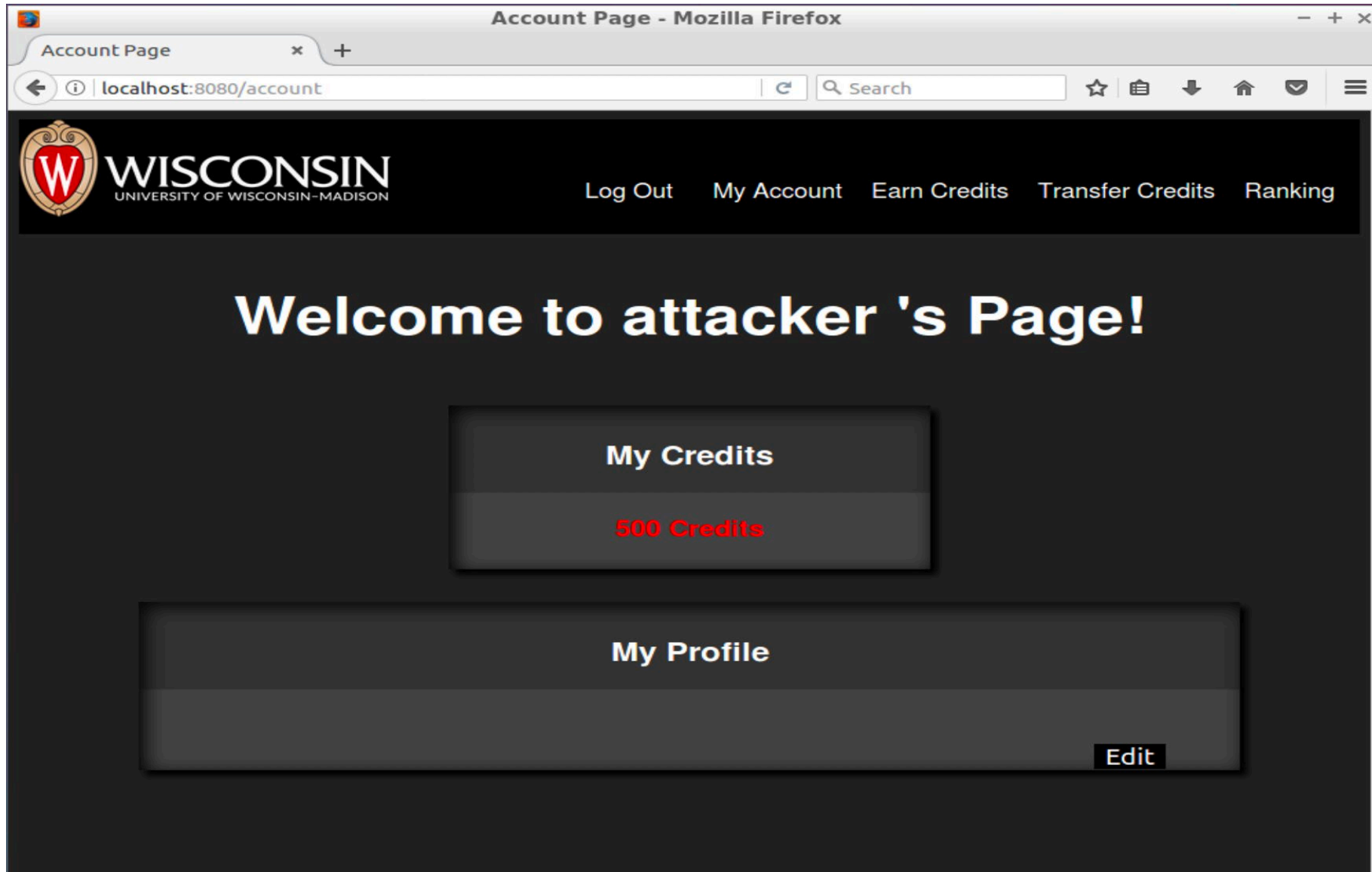
```
http://localhost:8080/welcome
```

Get familiar with the application:

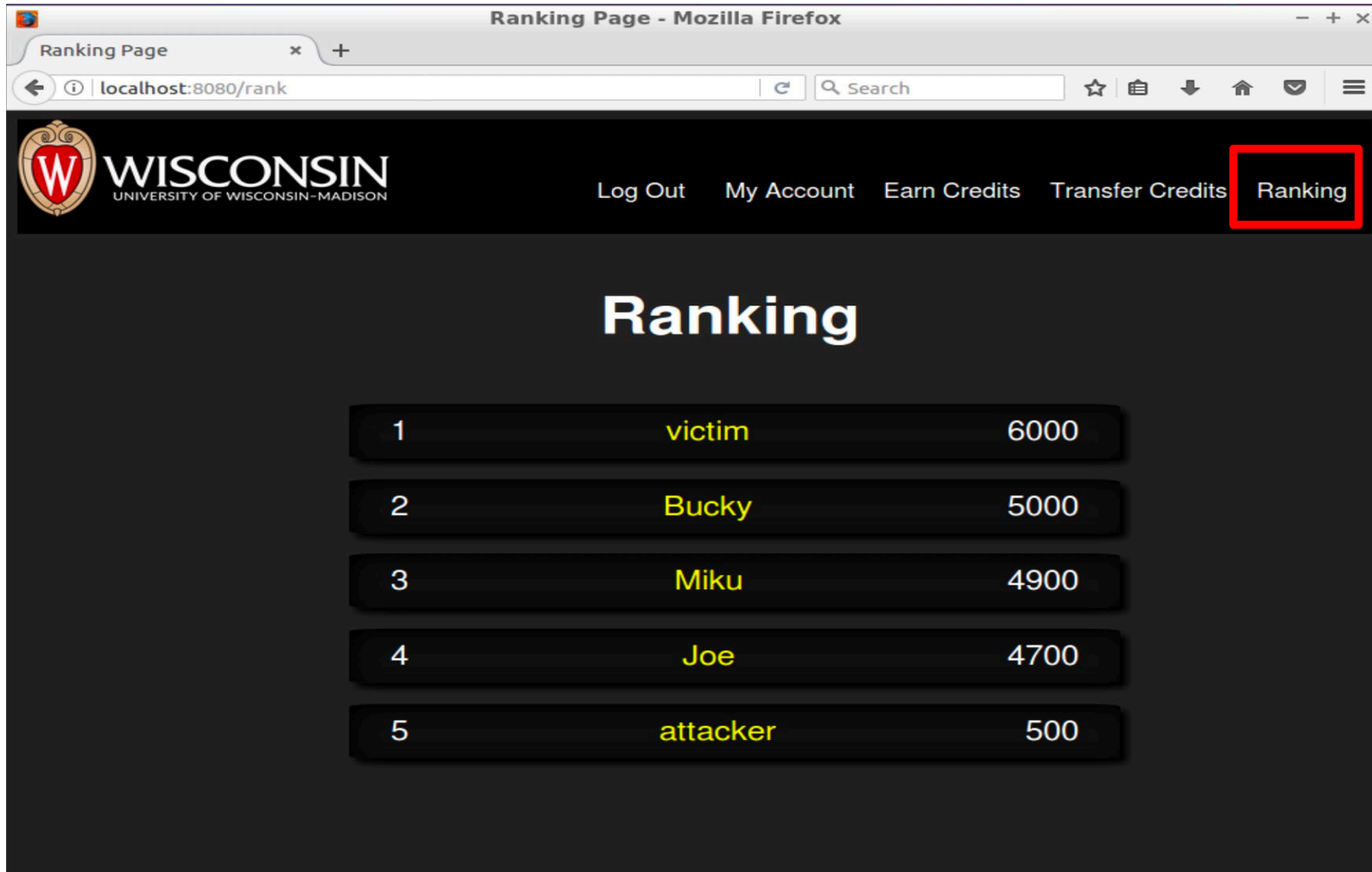
Run WisClick



Run WisClick



Run WisClick



Ranking Page - Mozilla Firefox

Ranking Page

localhost:8080/rank

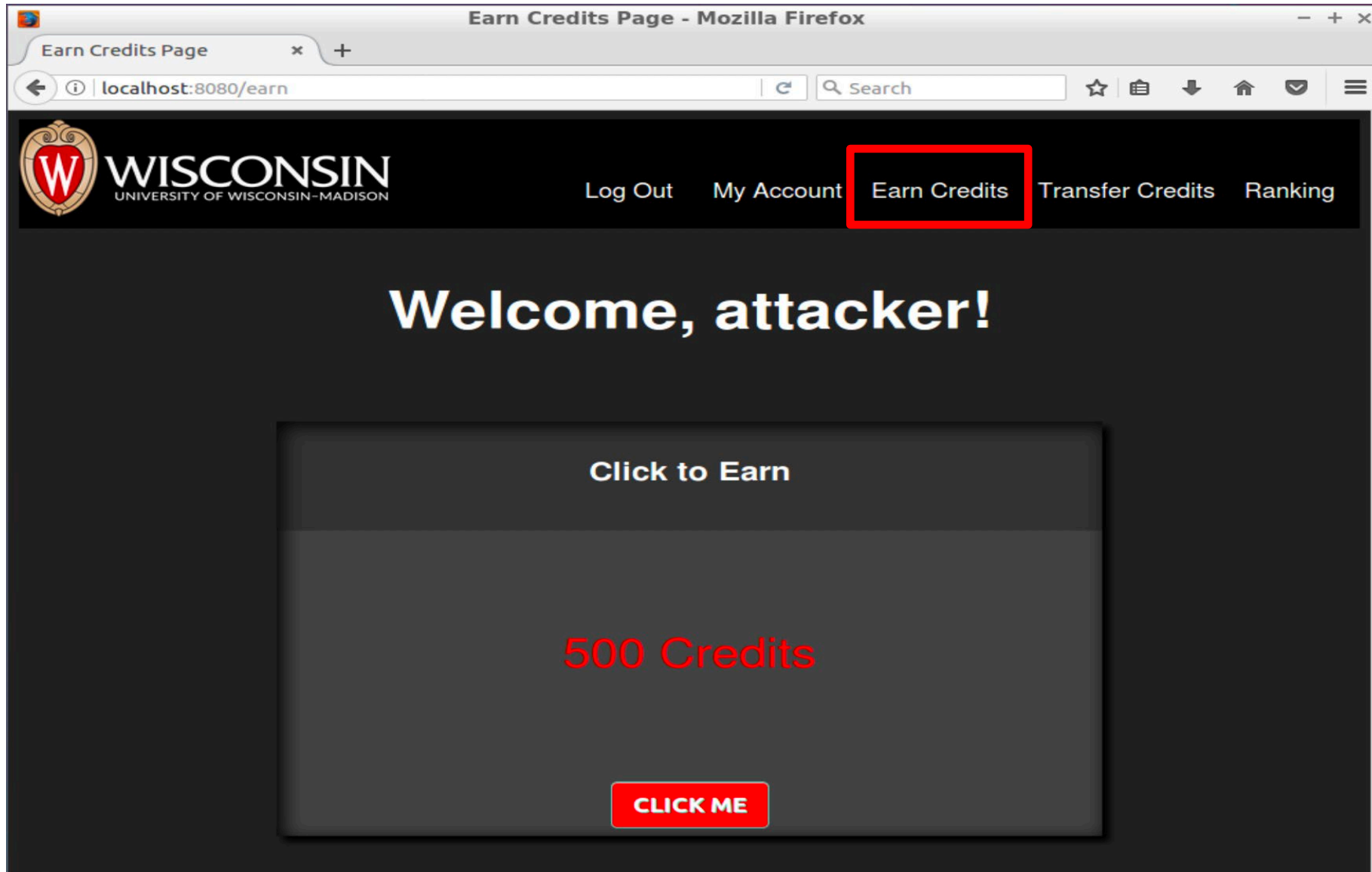
WISCONSIN UNIVERSITY OF WISCONSIN-MADISON

Log Out My Account Earn Credits Transfer Credits **Ranking**

Ranking

1	victim	6000
2	Bucky	5000
3	Miku	4900
4	Joe	4700
5	attacker	500

Run WisClick



Run WisClick

The screenshot shows a Mozilla Firefox browser window titled "Transfer Page - Mozilla Firefox". The address bar shows "localhost:8080/transfer". The page header features the University of Wisconsin-Madison logo and navigation links: "Log Out", "My Account", "Earn Credits", "Transfer Credits" (highlighted with a red box), and "Ranking". The main content area has the heading "Transfer Your Credits" and a central form box. The form box displays "Your Credits: 500" in red text. Below this, there are two input fields labeled "To:" and "Points:". At the bottom of the form box is a red "Submit" button.

WisClick Vulnerabilities

1. Cross-Site Scripting (XSS).
2. Cross-Site Request Forgery (CSRF).
3. Extracting Credentials and using them.

2. Cross-Site Request Forgery

A. Craft a script to steal some victim's credits.

- Log in as the attacker.
- Create a POST request using the same attack surface used for XSS.
- When the victim views the attacker's profile, some of their credits will be transferred to the attacker's account.

Hint:

- You need to understand how a valid transfer request is sent:
 - From the attacker's account go to the Transfer page. Create a transfer and observe the traffic with Web Inspector (Network tab).

2. Cross-Site Request Forgery

B. The attacker changes the victim's profile content, and every user who sees the victim's profile gets infected.

- Log in as the attacker.
- Observe the traffic when editing your own profile.

<http://localhost:8080/account>

After the victim views the attacker's page, they will get infected, so that if other users view the victim's profile, they will be infected as well. At the same time **the attacker also steals some credits from infected users.**

2. Cross-Site Request Forgery

- **Hint:**

Script that copies itself (and displays its content on an alert window):

```
<script id=replica>
  var headerTag = '<script id=\"replica\"
                  type=\"text/javascript\">';
  var jsCode = document.getElementById('replica').innerHTML;
  var tailTag = '</' + 'script>';
  var replicaCode = headerTag + jsCode + tailTag;
  alert(replicaCode);
</script>
```

2. Cross-Site Request Forgery

- Log in as the victim and view the attacker's profile.
- See how your profile changed and you lost credits.
- Log in as user Bucky, passwd badger and view the victim's profile.
- See how Bucky's profile changed and they lost credits.
- Include the result of that in your report: code, screenshots and explanation.

Web Exercises

Part 2: CSRF

Barton P. Miller

Computer Sciences Department
University of Wisconsin

bart@cs.wisc.edu

Elisa Heymann

Computer Sciences Department
University of Wisconsin
Universitat Autònoma de Barcelona

elisa@cs.wisc.edu