# **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:

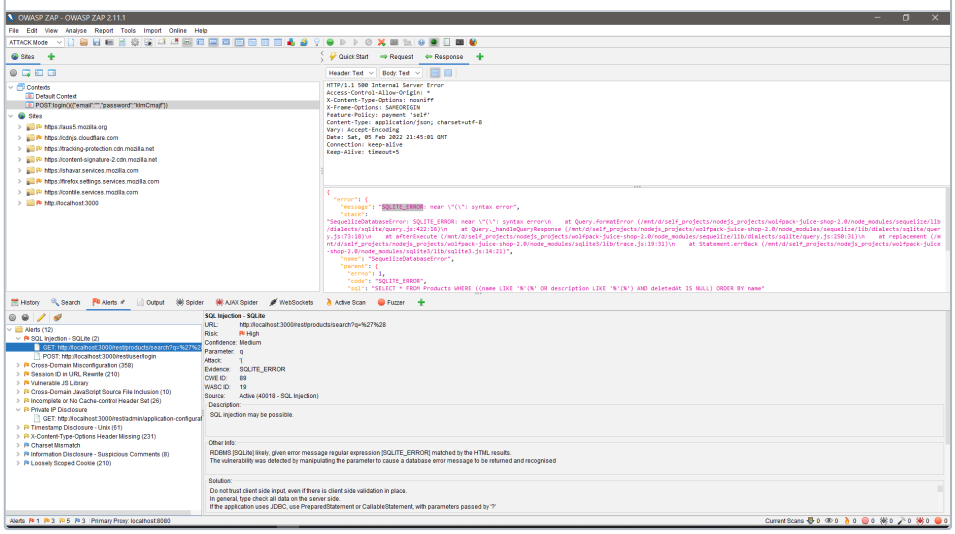
Vishnu Challa, vchalla2
Srujan Ponnur, sponnur
Varun Kumar Veginati, vvegina

# **Q2** Report Upload

30 Points

Upload a screenshot that shows a summary of alert categories/results from the automated ZAP scan.

zap_alerts.PNG Download



# **Q3** Potential Vulnerability

30 Points

From the report output, select an alert type/category (other than SQL Injection) that you find interesting.

## **Q3.1** Description
15 Points

In 3-4 sentences, briefly describe why you think the alert is a false positive or an actual concern.

Vulnerable JS Library - The wolfpack Juice shop is using the jquery library of version 2.2.4. This is an outdated version with a lot of vulnerabilities. Ex: There were numerous cross-site scripting issues that were reported along with prototype pollution vulnerability within the "extend" function.

## **Q3.2** Impact
15 Points

In 3-4 sentences, briefly describe how the software developers could mitigate the threat associated with the vulnerability indicated in the alert.

The current stable version of Jquery available is 3.6.0. By updating the jquery and all the other dependencies within the application to the latest stable version we can mitigate and solve some of the vulnerabilities with the legacy version of the jquery library,

# **Q4** Fuzzing
30 Points

Use ZAP to find an endpoint other than `login` that is susceptible to SQL injection or cross-site scripting. *NOTE: unlike the example with SQL Injection where a response of `OK` indicated an attack was most likely successful, you will have to manually try the cross-site scripting payloads that have a response of `OK` since the payload text may have been submitted successfully but not executed as a control instruction.*

**If you are unable to locate another endpoint that is susceptible to SQL injection or cross-site scripting, then you will need to report on endpoints you checked and how they were checked.**

## **Q4.1** Steps

15 Points

List the steps that need to be performed for to check whether the attack is successful or not. **If you were not able to locate a vulnerable endpoint, then list 5 different endpoints that you manually checked based on the ZAP results**

1. Tried SQL injection fuzzing on search end-point. Applied fuzzing on query parameter. But didn't work out. Attaching the curl command that we tried below.
curl --request GET \  --url 'http://localhost:3000/rest/products/search?q=apple'

2. Tried XSS fuzzing on track-order end-point. Applied fuzzing on track id. But didn't work out. Attaching the curl command that we tried below.
curl --request GET \  --url 'http://localhost:3000/rest/track-order/fe01-52a134dba3f5752c'

3. Tried SQL injection fuzzing on captcha id while submitting a complaint. But didn't work out. Attaching the curl command that we tried below.

```
curl --request POST \
  --url http://localhost:3000/api/Feedbacks/ \
  --header 'Content-Type: application/json' \
  --data '{
        "UserId": 17,
        "captchaId": 1,
        "captcha": "-26",
        "comment": "abncda (***o)",
        "rating": 4
}'
```

4. Tried XSS fuzzing on email while create new user. But none of them worked when we went to the administration end-point when logged in as admin. Attaching the curl command that we tried below.
curl --request POST \

```
 --url http://localhost:3000/api/Users/ \
 --header 'Content-Type: application/json' \
 --data '{
      "email": "abc@email.com",
      "password": "abcdefghi",
      "passwordRepeat": "abcdefghi",
      "securityQuestion": {
            "id": 1,
            "question": "Your eldest siblings middle name?",
            "createdAt": "2022-02-07T19:35:31.387Z",
            "updatedAt": "2022-02-07T19:35:31.387Z"
      },
      "securityAnswer": "sample"
}'
```

5. Tried XSS fuzzing on product review. But didn't work out. When the product review is opened, none of the alerts popped up. Attaching the curl command we tried below.

```
curl --request PUT \
 --url http://localhost:3000/rest/products/3/reviews \
 --header 'Content-Type: application/json' \
 --data '{
      "message": "abcd",
      "author": "<iframe src=\"javascript:alert(`hello`)\">"
}'
```

6. Tried XSS fuzzing on the name in "My Payment Options". But didn't work out. When the "My Payment Options" tab is opened, none of the alerts tried as part of fuzzing popped up. Attaching the curl command we tried below.

```
curl --request POST \
 --url http://localhost:3000/api/Cards/ \
 --header 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzd
WNjZXNzIiwiZGF0YSI6eyJpZCI6MTcsInVzZXJuYW1lIjoiIiwiZW1h
aWwiOiJkZW1vIiwicGFzc3dvcmQiOiJmZTAxY2UyYTdmYmFjOG
ZhZmFlZDdjOTgyYTA0ZTIyOSIsInJvbGUiOiJjdXN0b21lciIsImRlb
HV4ZVRva2VuIjoiIiwibGFzdExvZ2luSXAiOiIxMjcuMC4wLjEiLCJ
wcm9maWxlSW1hZ2UiOiJhc3NldHMvcHVibGljL2ltYWdlcy91cGx
vYWRzL2RlZmF1bHQuc3ZnIiwidG90cFNlY3JldCI6IiIsImlzQWN0
aXZlIjp0cnVlLCJjcmVhdGVkQXQiOiIyMDIyLTAyLTA5IDA0OjM1O
jA1LjEyNyArMDA6MDAiLCJ1cGRhdGVkQXQiOiIyMDIyLTAyLTA5I
```

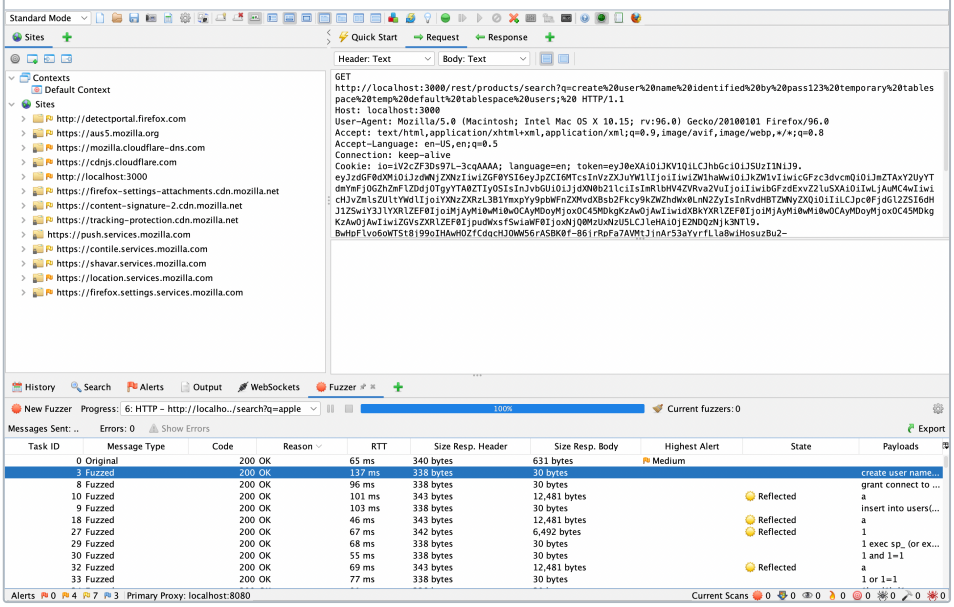
```
DA0OjM3OjQyLjE3OCArMDA6MDAiLCJkZWxldGVkQXQiOm51
bGx9LCJpYXQiOjE2NDQzODE2NDQsImV4cCI6MTY0NDM5OT
Y0NH0.tvFTCpuFYdQRAUQX1g55_Nr4dG5BXYF09cu_sxSMD6
cMwqkrzSw05o_WPFBLY3imP8ywkCi4cqT1dX-aVNYqFarQA-
RvVD0cWoH9Rz19ajL8-3j1krWh-
6AeJRmfXGUqstzjGTU1Xw9Pk0UCdqG5-
P7yjn5yggHkWfo8yMunQNM' \
  --header 'Content-Type: application/json' \
  --data '{
        "fullName": "abcdefg",
        "cardNum": 1234567891234567,
        "expMonth": "3",
        "expYear": "2082"
}'
```

## **Q4.2** Attack

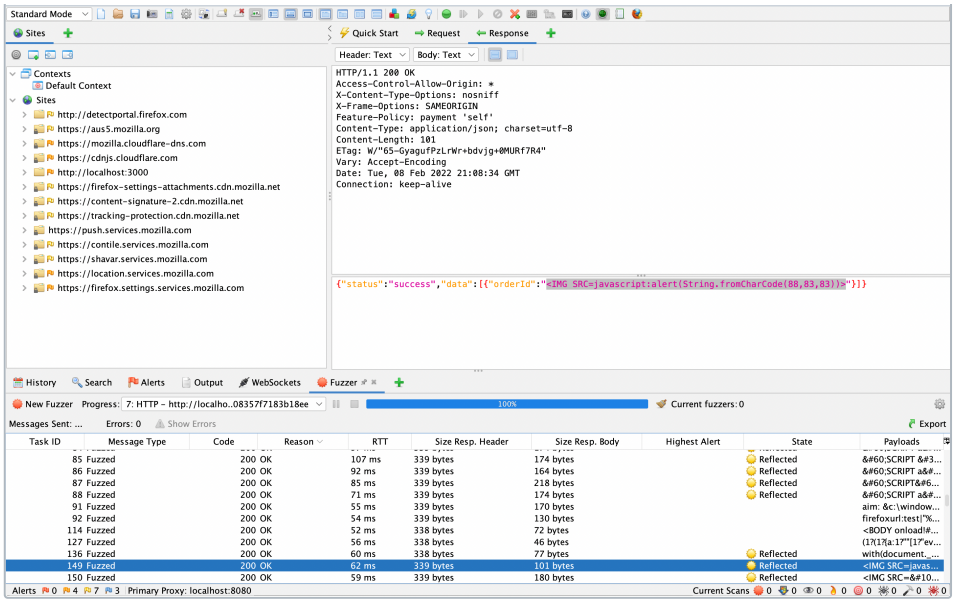
15 Points

Upload images that:
(1) Show the alert payload in ZAP.
(2) Show a screenshot of the successful attack in the web application. **If you were not able to locate a vulnerable endpoint, then provide one or more screenshot(s) to indicate what happened when you manually tried the payload in the webapp**

▾ 1.png Download

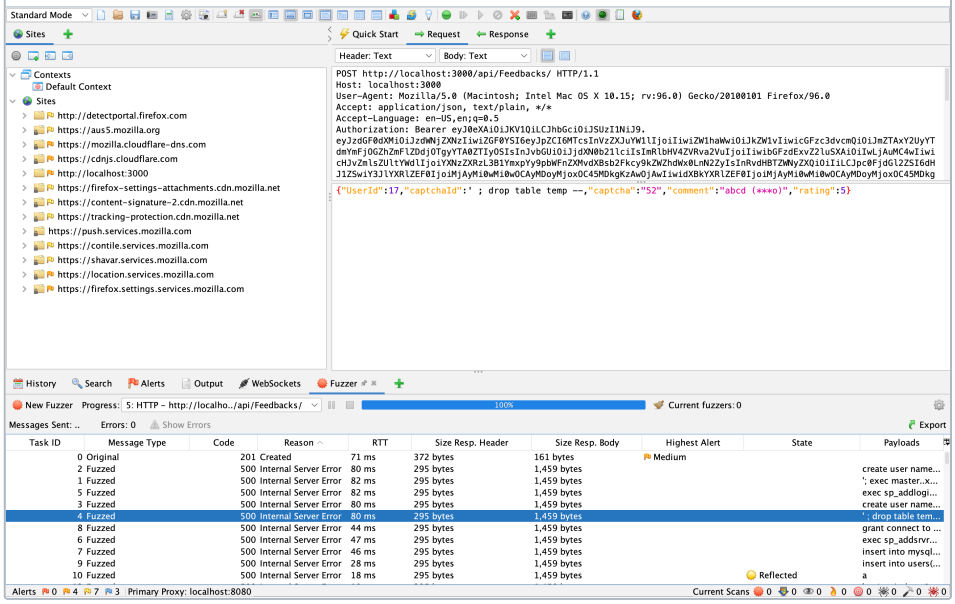



▾ 2.png Download



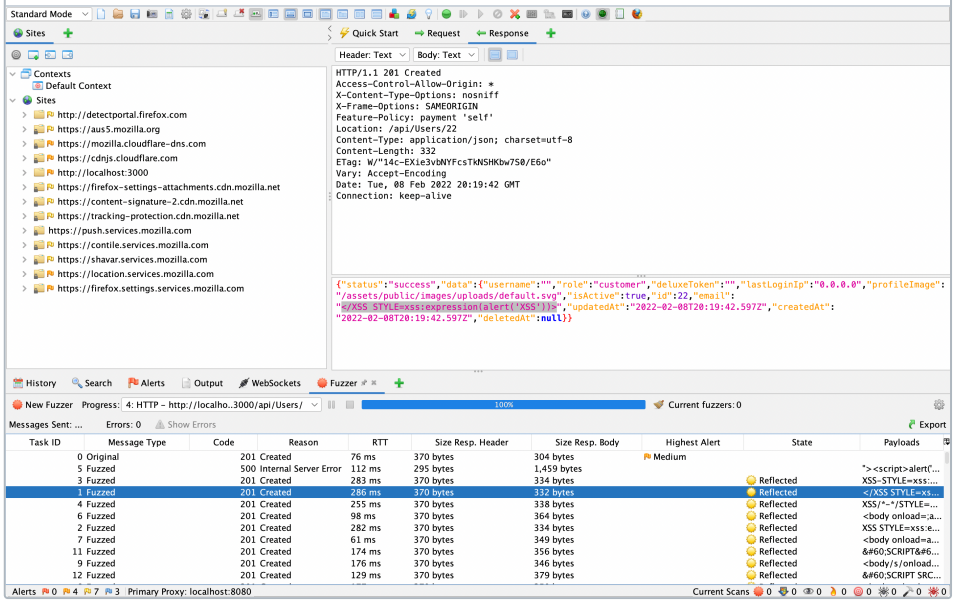
## 3.png

Download



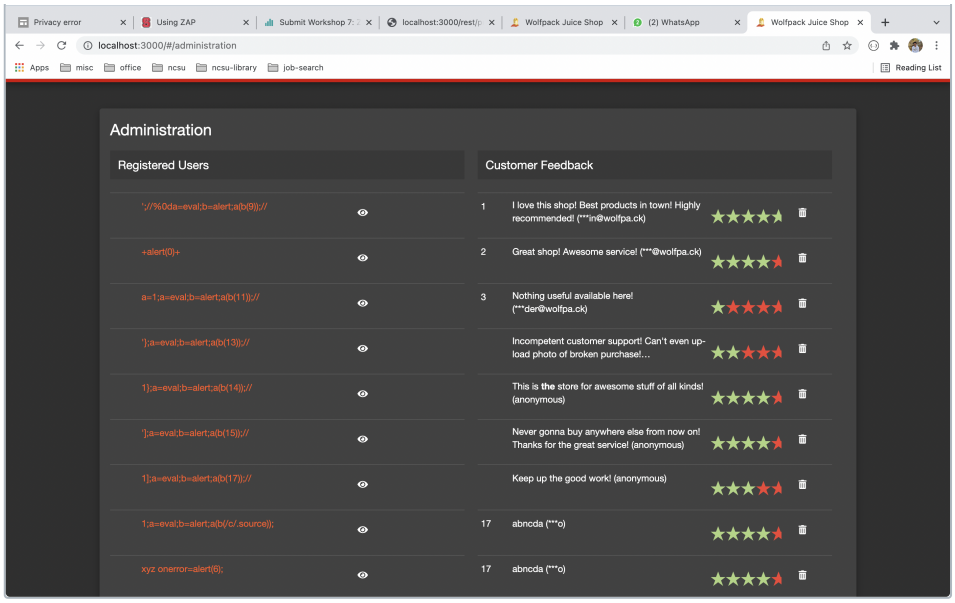
## 4-a.png

Download

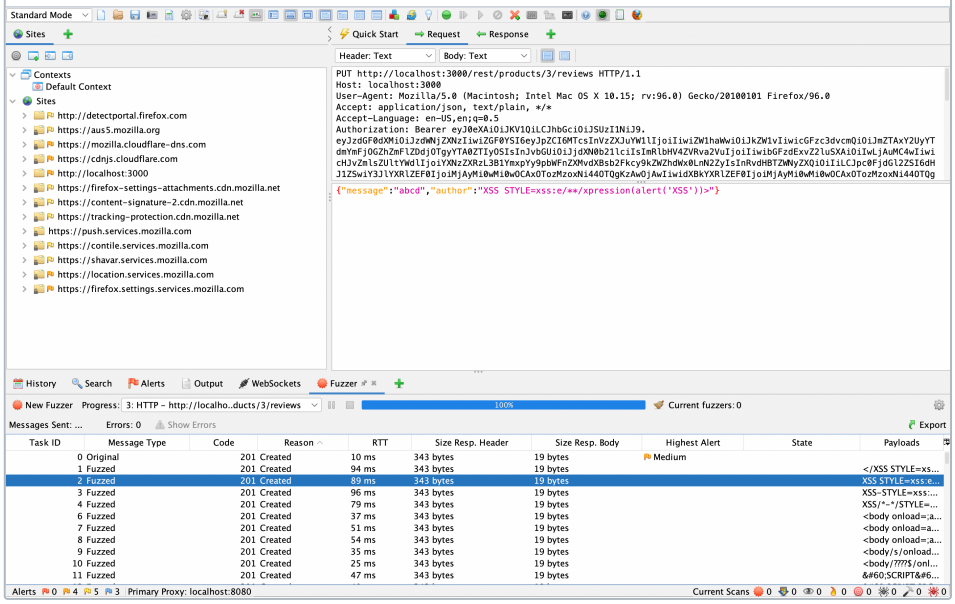


## 4-b.png

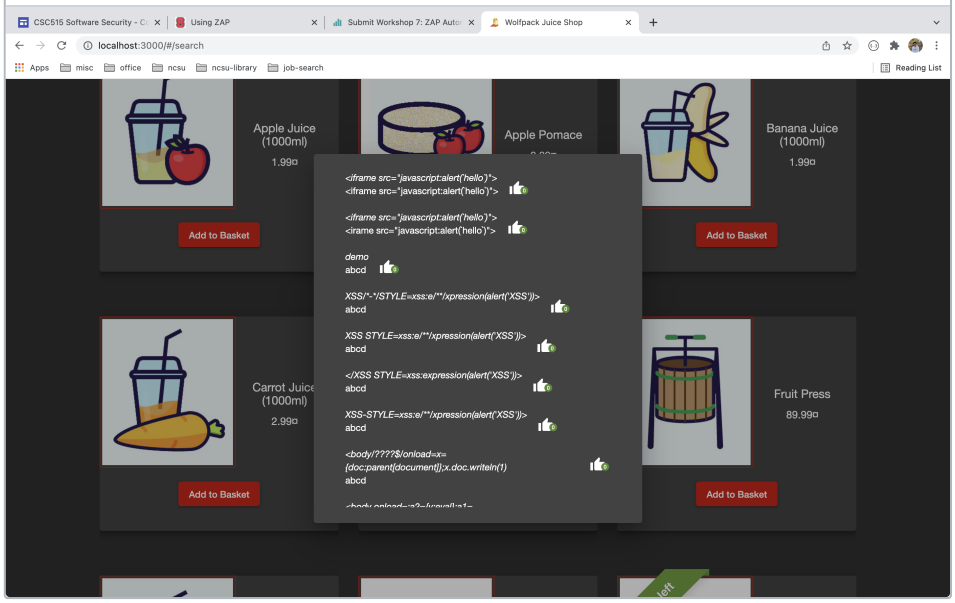
Download



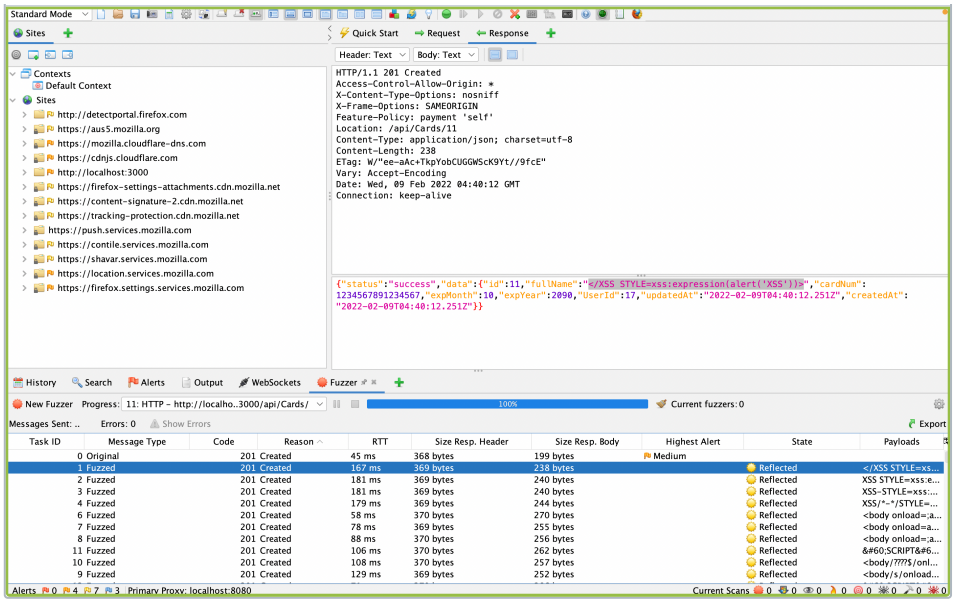

5-a.png Download



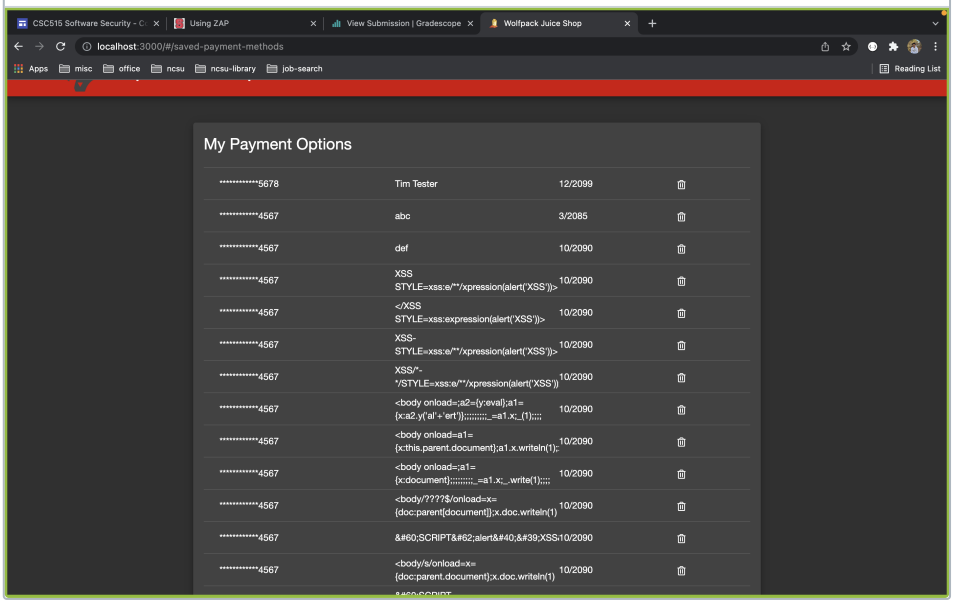
5-b.png Download




6-a.png Download



6-b.png Download




# Q5 Previous Results

10 Points

Try fuzzing fields that were previously identified as vulnerable to SQL injection (other than the `login` module) or cross-site scripting in previous labs. Did the provided payloads in ZAP identify the vulnerabilities? If so, give an example payload that was successful. It not, suggest a reason why the payloads may not have been successful.

1. Tried creating a new User.
2. New user creation uses a POST endpoint.
3. After creating a new user using sample email, tried XSS fuzzing on the email id in the ZAP application.
4. Though some of the payloads returned 201 status, it doesn't

actually create any impact.
5. When logged in as admin and redirected to the administration page (https://localhost:3000/#/administration), none of the HTML alerts used as payload worked.
6. The payloads might not have been successful because there might be a denylist within the application which blocked some of the syntaxes that were used in the payload.
7. Curl command used:

```
curl --request POST \
  --url http://localhost:3000/api/Users/ \
  --header 'Content-Type: application/json' \
  --data '{
        "email": "def@email.com",
        "password": "123456789",
        "passwordRepeat": "abcdefghi",
        "securityQuestion": {
                "id": 1,
                "question": "Your eldest siblings middle name?",
                "createdAt": "2022-02-07T19:35:31.387Z",
                "updatedAt": "2022-02-07T19:35:31.387Z"
        },
        "securityAnswer": "sample"
}'
```

# Workshop 7: ZAP Automated Scan & Fuzzing Tool

● GRADED

**GROUP**

Varun Kumar Veginati
Vishnu Challa
Srujan Ponnur
View or edit group

**TOTAL POINTS**

**100 / 100 pts**

**QUESTION 1**

## Study Group Information

**0** / 0 pts

**QUESTION 2**

Report Upload **30** / 30 pts

**QUESTION 3**

Potential Vulnerability **30** / 30 pts

3.1 Description **15** / 15 pts

3.2 Impact **15** / 15 pts

**QUESTION 4**

Fuzzing **30** / 30 pts

4.1 Steps **15** / 15 pts

4.2 Attack **15** / 15 pts

**QUESTION 5**

Previous Results **10** / 10 pts