ATTACK DEFENSE

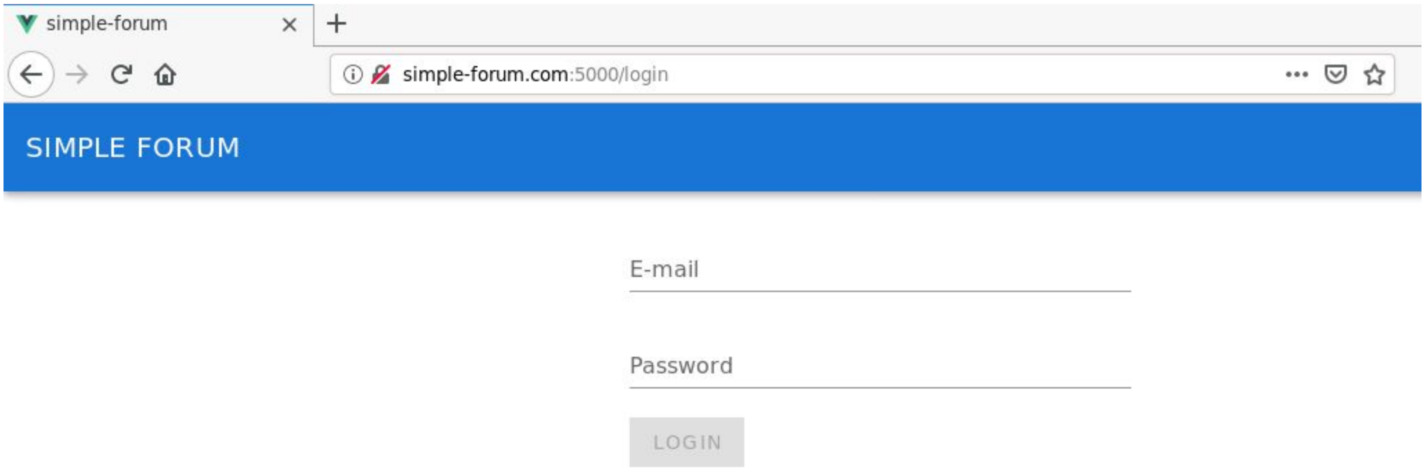
by PentesterAcademy

Name	Insecure Local Storage
URL	https://attackdefense.com/challengedetails?cid=1974
Type	REST: API Security

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Step 1: Viewing with the Forum webapp.

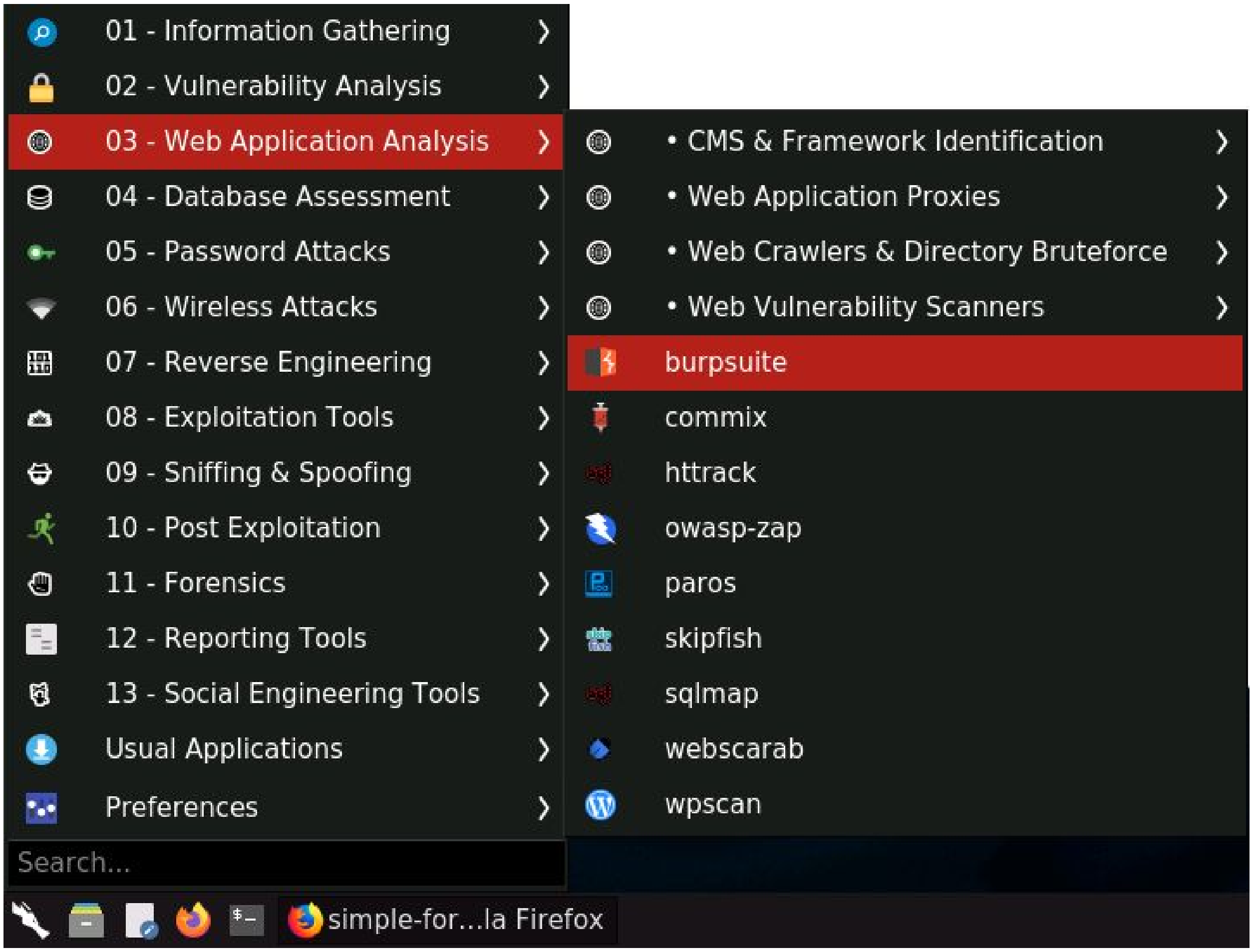
When the lab starts up, the webapp opens up in the browser:



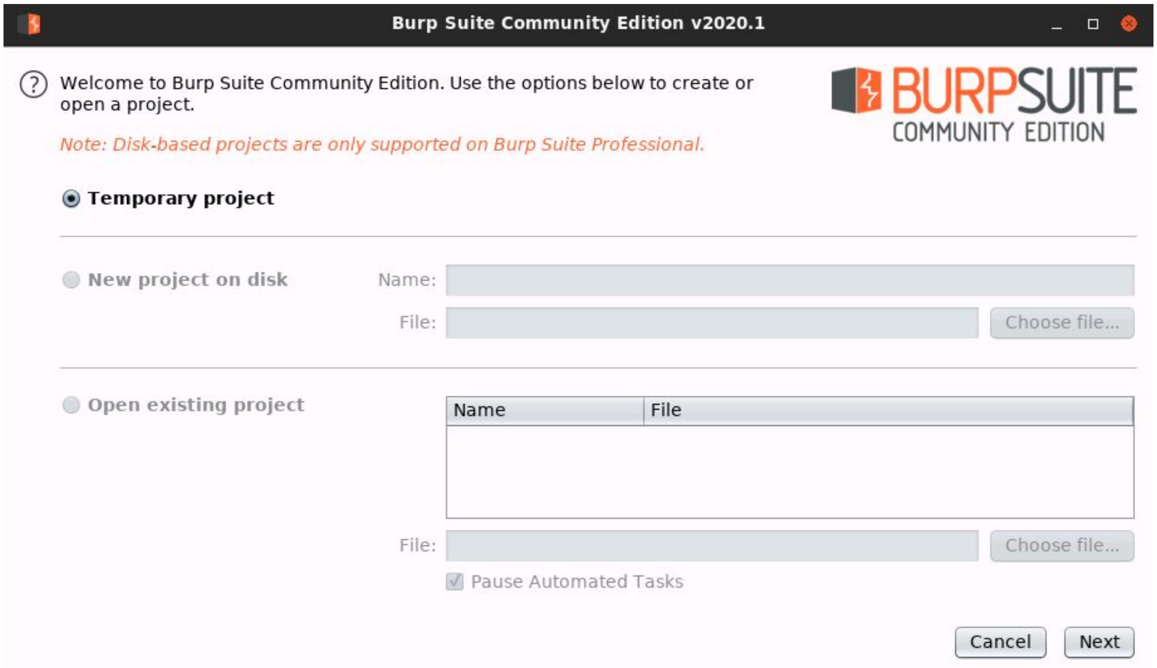
Step 2: Configuring the browser to use BurpSuite proxy and making BurpSuite intercept all the requests made to the API.

Launch BurpSuite.

Select Web Application Analysis > burpsuite



The following window will appear:



Burp Suite Community Edition v2020.1

Welcome to Burp Suite Community Edition. Use the options below to create or open a project.

Note: Disk-based projects are only supported on Burp Suite Professional.

● **Temporary project**

○ **New project on disk** Name: File:

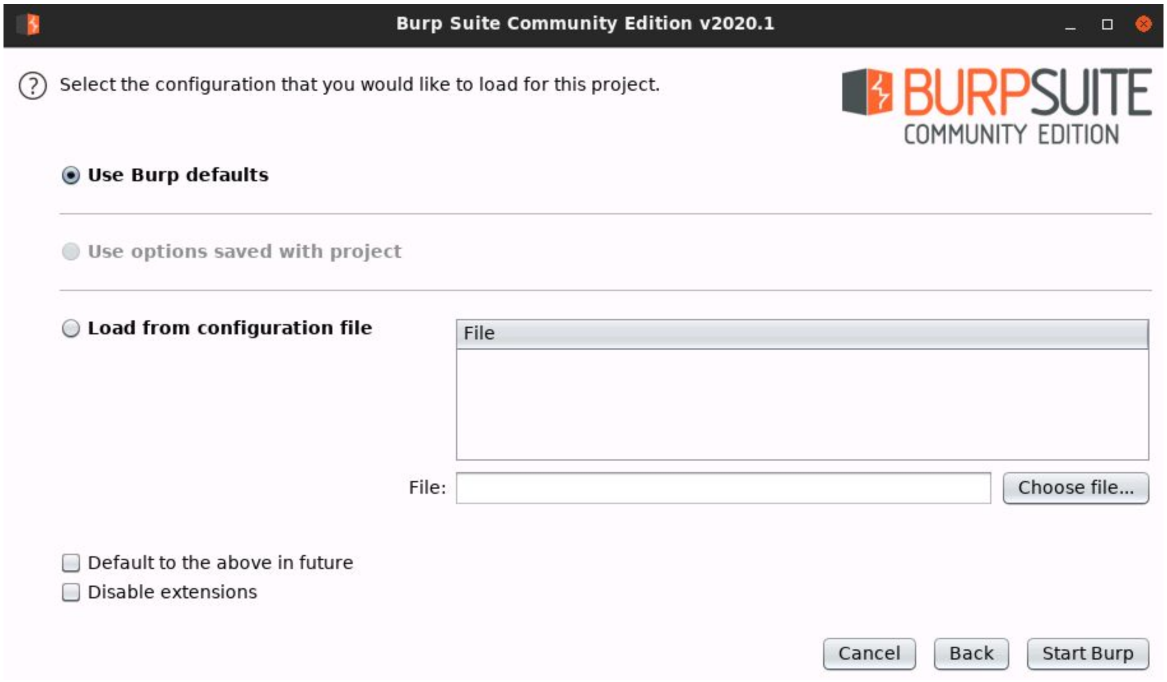
○ **Open existing project**

File:

☑ Pause Automated Tasks

Click Next.

Finally, click Start Burp in the following window:



Burp Suite Community Edition v2020.1

Select the configuration that you would like to load for this project.

● **Use Burp defaults**

○ **Use options saved with project**

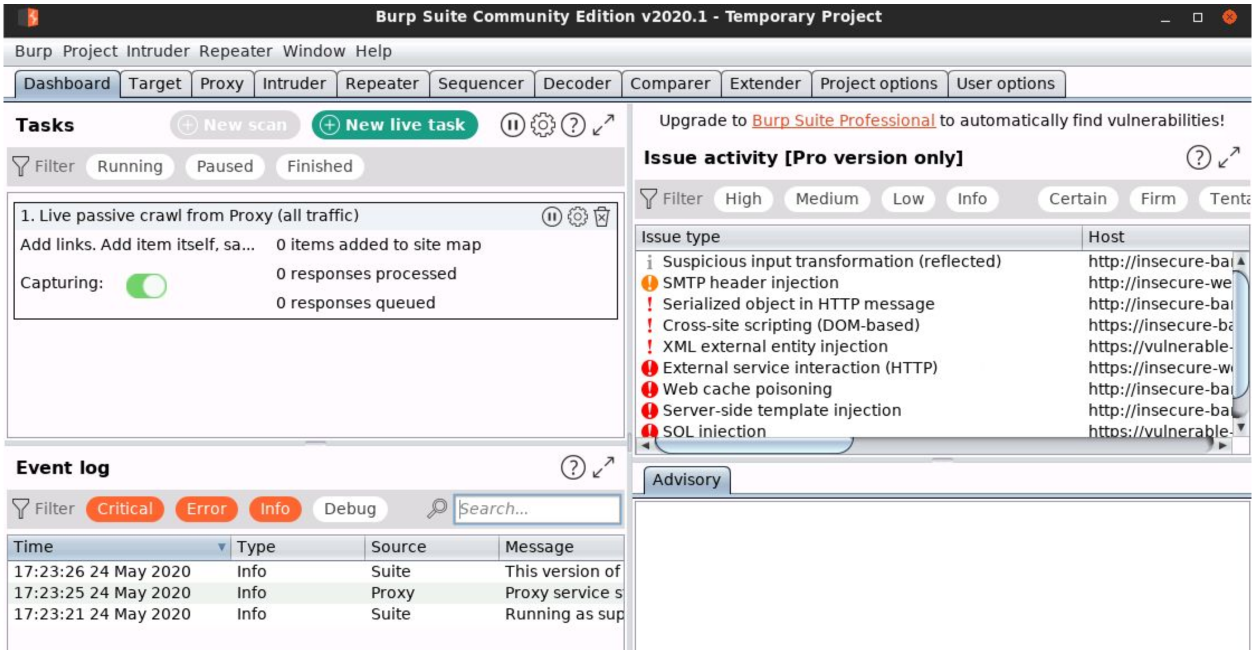
○ **Load from configuration file**

File:

□ Default to the above in future

□ Disable extensions

The following window will appear after BurpSuite has started:



Burp Suite Community Edition v2020.1 - Temporary Project

Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
Tasks
New scan
New live task
Filter Running Paused Finished

1. Live passive crawl from Proxy (all traffic)

Add links. Add item itself, sa... 0 items added to site map

Capturing: 0 responses processed

0 responses queued

Upgrade to Burp Suite Professional to automatically find vulnerabilities!

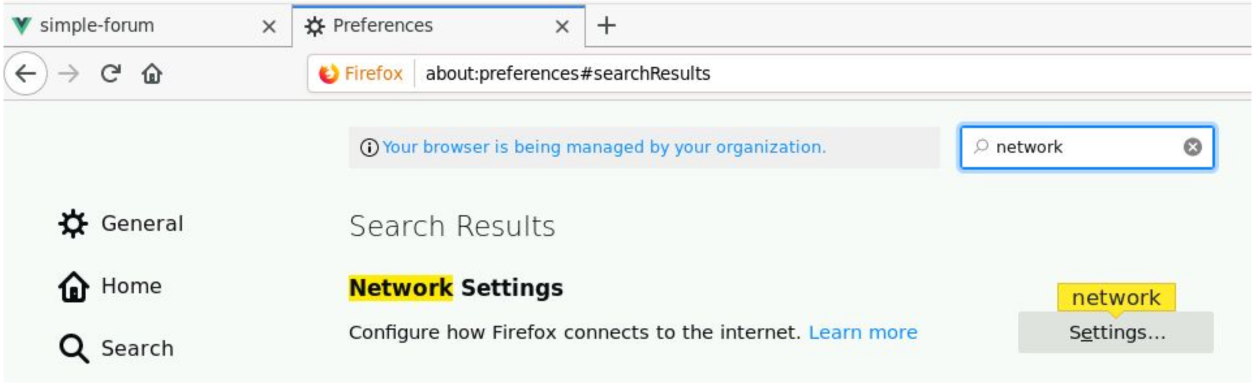
Issue activity [Pro version only]
Filter High Medium Low Info Certain Firm Tenta

Issue type	Host
Suspicious input transformation (reflected)	http://insecure-ba
SMTP header injection	http://insecure-we
Serialized object in HTTP message	http://insecure-ba
Cross-site scripting (DOM-based)	https://insecure-ba
XML external entity injection	https://vulnerable-
External service interaction (HTTP)	https://insecure-w
Web cache poisoning	http://insecure-ba
Server-side template injection	http://insecure-ba
SQL injection	https://vulnerable-

Time	Type	Source	Message
17:23:26 24 May 2020	Info	Suite	This version of
17:23:25 24 May 2020	Info	Proxy	Proxy service s
17:23:21 24 May 2020	Info	Suite	Running as sup

Configure the browser to use the Burp proxy listener as its HTTP Proxy server.

Open the browser preference settings and search for network proxy settings.



Your browser is being managed by your organization.

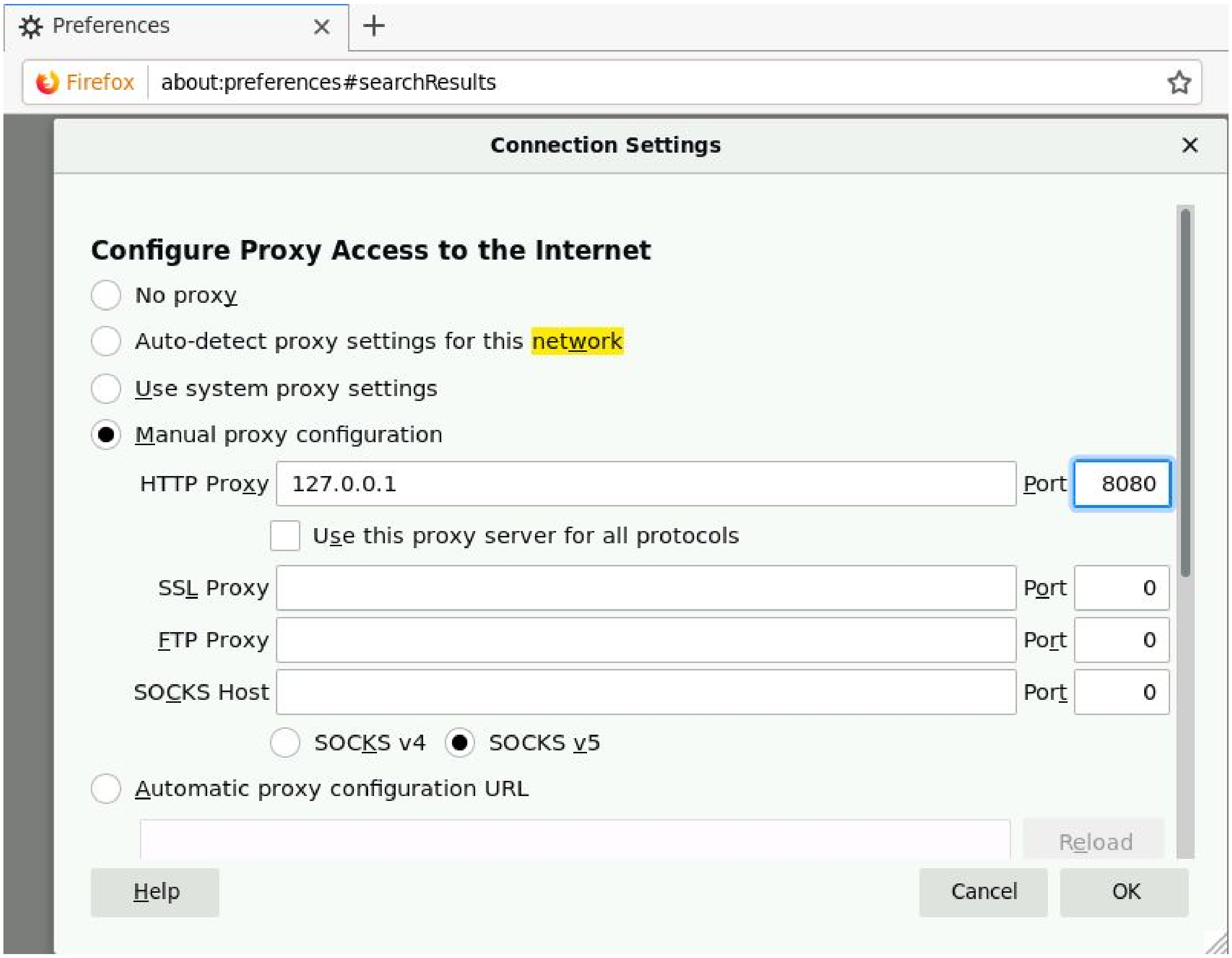
network

Search Results

Network Settings

Configure how Firefox connects to the internet. Learn more

Select Manual Proxy Configuration and set the HTTP Proxy address to localhost and the port to 8080.



Preferences

Firefox about:preferences#searchResults

Connection Settings

Configure Proxy Access to the Internet

○ No proxy

○ Auto-detect proxy settings for this network

○ Use system proxy settings

● Manual proxy configuration

HTTP Proxy 127.0.0.1 Port 8080

□ Use this proxy server for all protocols

SSL Proxy Port 0

FTP Proxy Port 0

SOCKS Host Port 0

○ SOCKS v4 ● SOCKS v5

○ Automatic proxy configuration URL

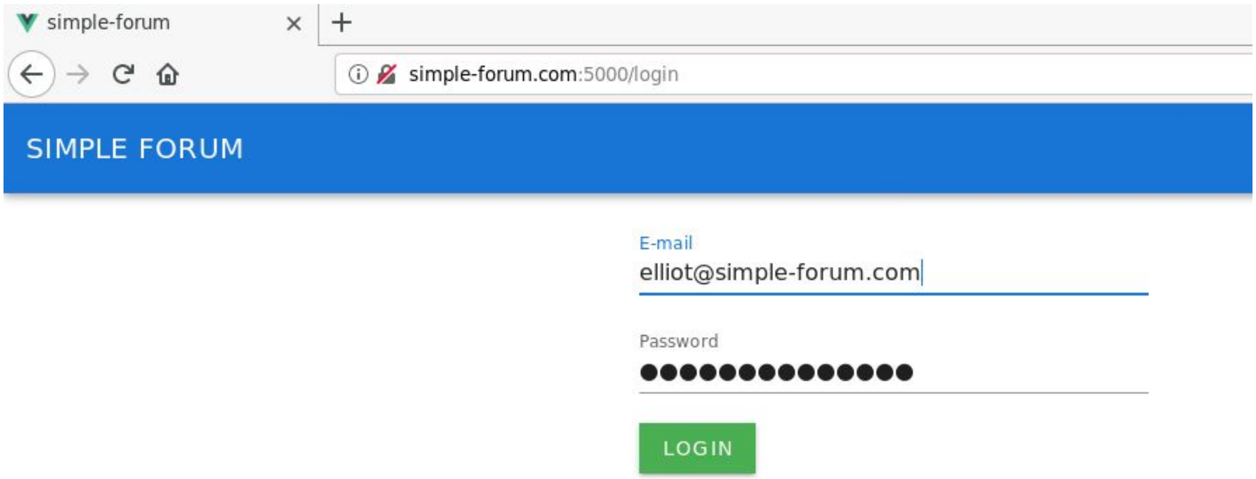
Click OK.

Everything required to intercept the requests has been set up.

Step 3: Login into the forum.

Login into the forum using the provided credentials:

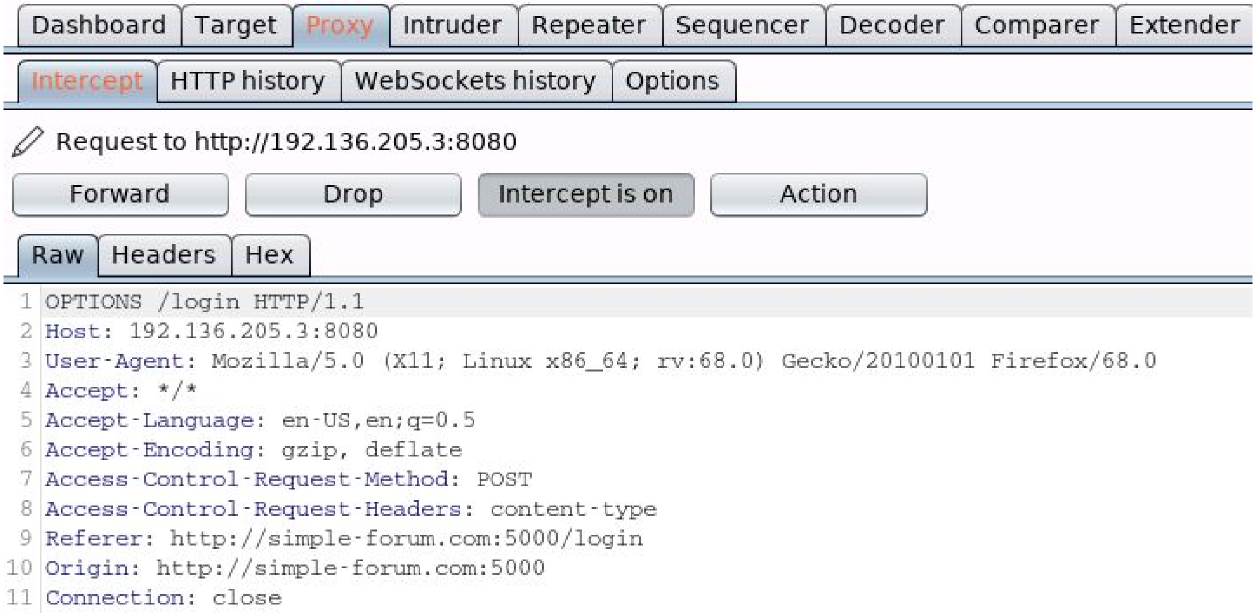
Email: elliot@simple-forum.com
Password: elliotalderson



E-mail
elliot@simple-forum.com

Password
●●●●●●●●●●●●●●

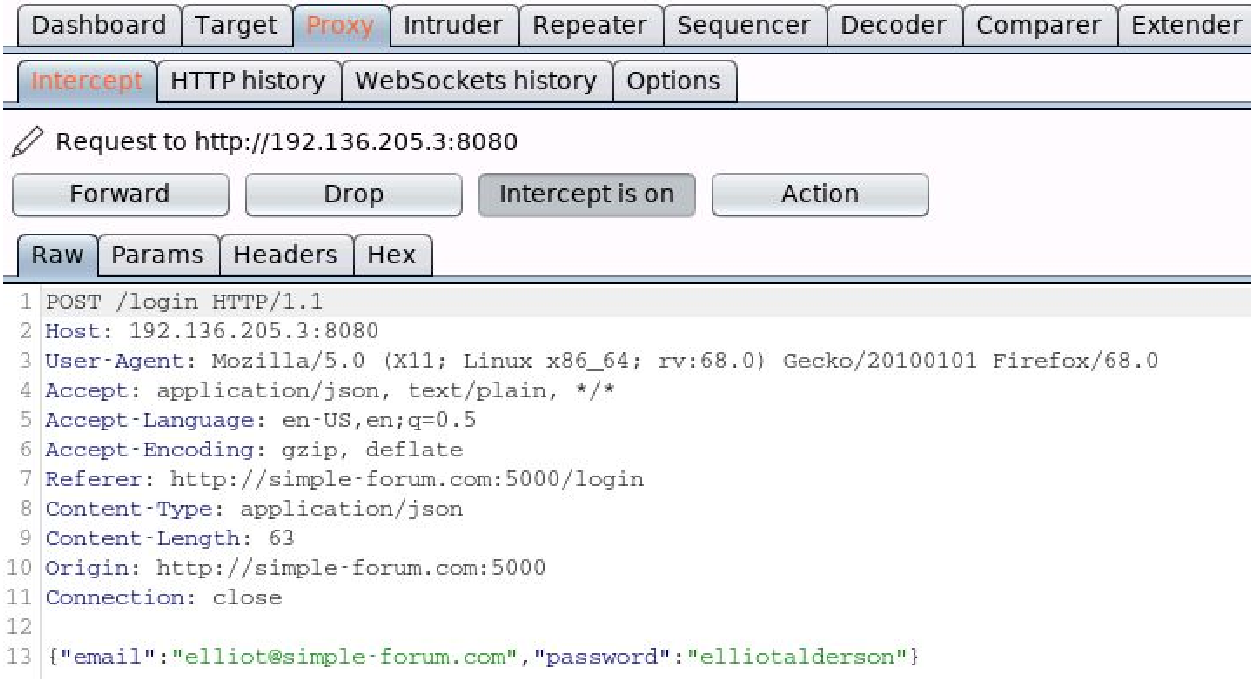
Check the intercepted request:



Request to http://192.136.205.3:8080

```
OPTIONS /login HTTP/1.1
Host: 192.136.205.3:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Access-Control-Request-Method: POST
Access-Control-Request-Headers: content-type
Referer: http://simple-forum.com:5000/login
Origin: http://simple-forum.com:5000
Connection: close
```

Forward the intercepted request:

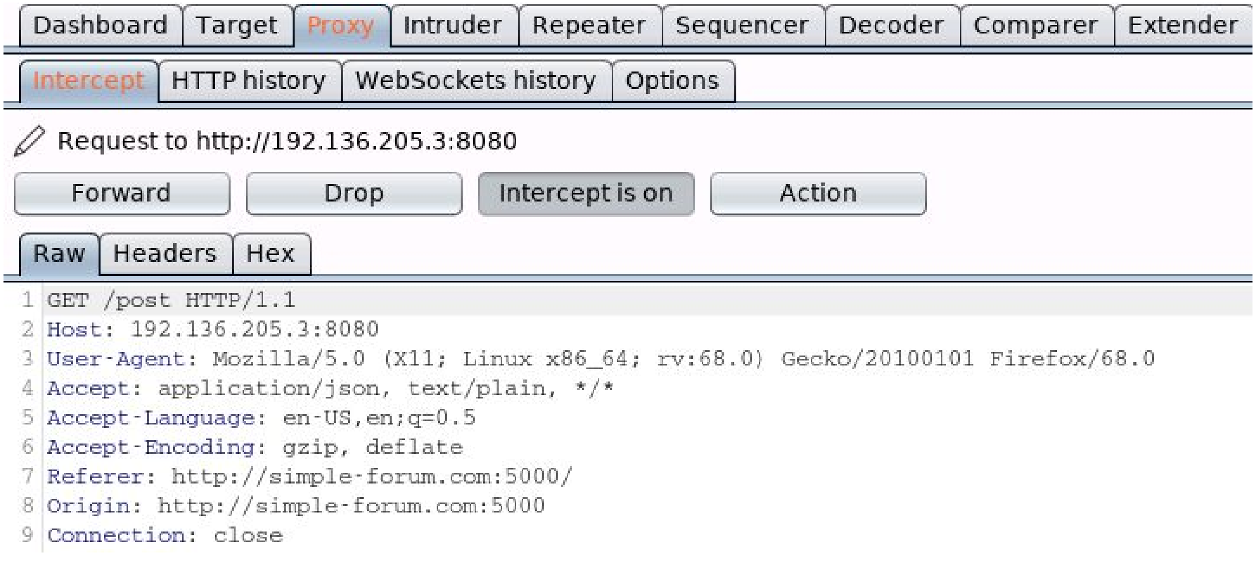


Request to http://192.136.205.3:8080

```
POST /login HTTP/1.1
Host: 192.136.205.3:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://simple-forum.com:5000/login
Content-Type: application/json
Content-Length: 63
Origin: http://simple-forum.com:5000
Connection: close

{"email":"elliot@simple-forum.com","password":"elliotalderson"}
```

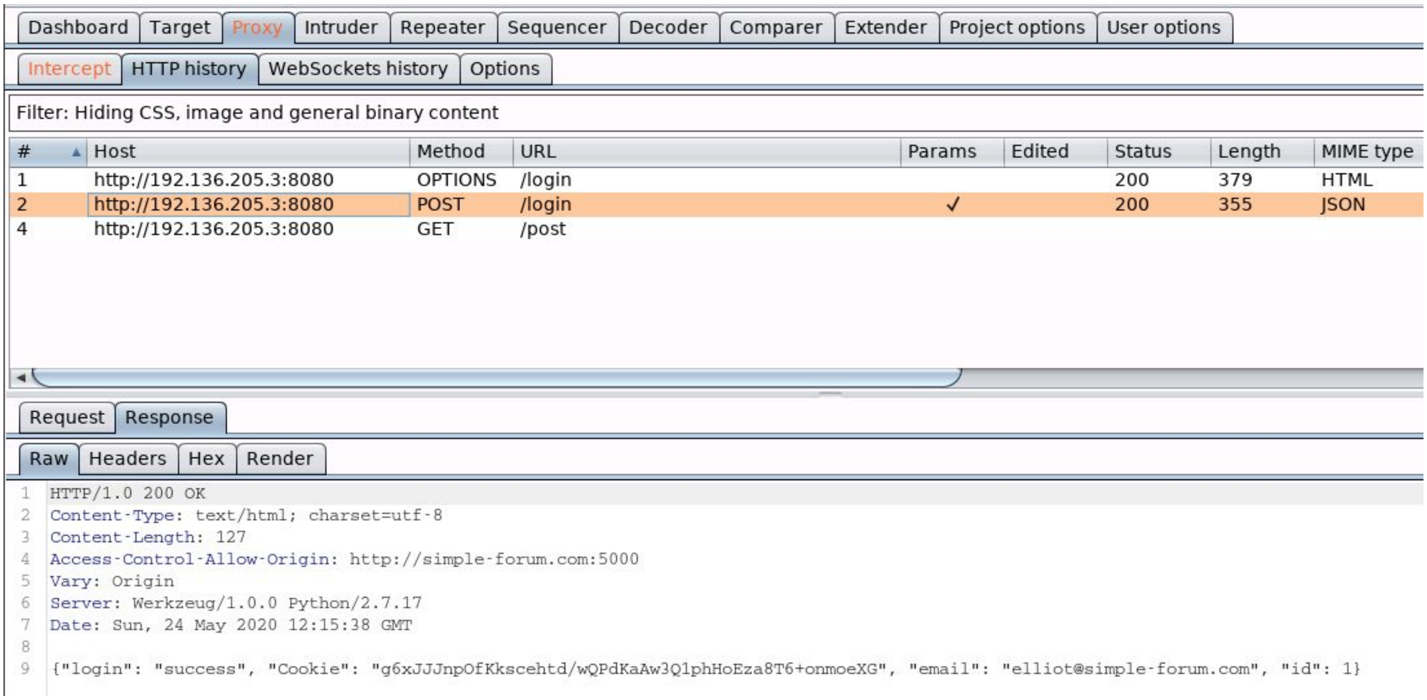
Forward the above request.



Request to http://192.136.205.3:8080

```
GET /post HTTP/1.1
Host: 192.136.205.3:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://simple-forum.com:5000/
Origin: http://simple-forum.com:5000
Connection: close
```

Check the response of the above request to /login in the HTTP History tab:



Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type
1	http://192.136.205.3:8080	OPTIONS	/login			200	379	HTML
2	http://192.136.205.3:8080	POST	/login	✓		200	355	JSON
4	http://192.136.205.3:8080	GET	/post					

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 127
Access-Control-Allow-Origin: http://simple-forum.com:5000
Vary: Origin
Server: Werkzeug/1.0.0 Python/2.7.17
Date: Sun, 24 May 2020 12:15:38 GMT

{"login": "success", "Cookie": "g6xJJJnpOfKkscehtd/wQPdKaAw3Q1phHoEza8T6+onmoeXG", "email": "elliot@simple-forum.com", "id": 1}
```

Notice that the response contains the email id of the user, cookie, login status (success in this case) and id.

The id is 1 in the response.

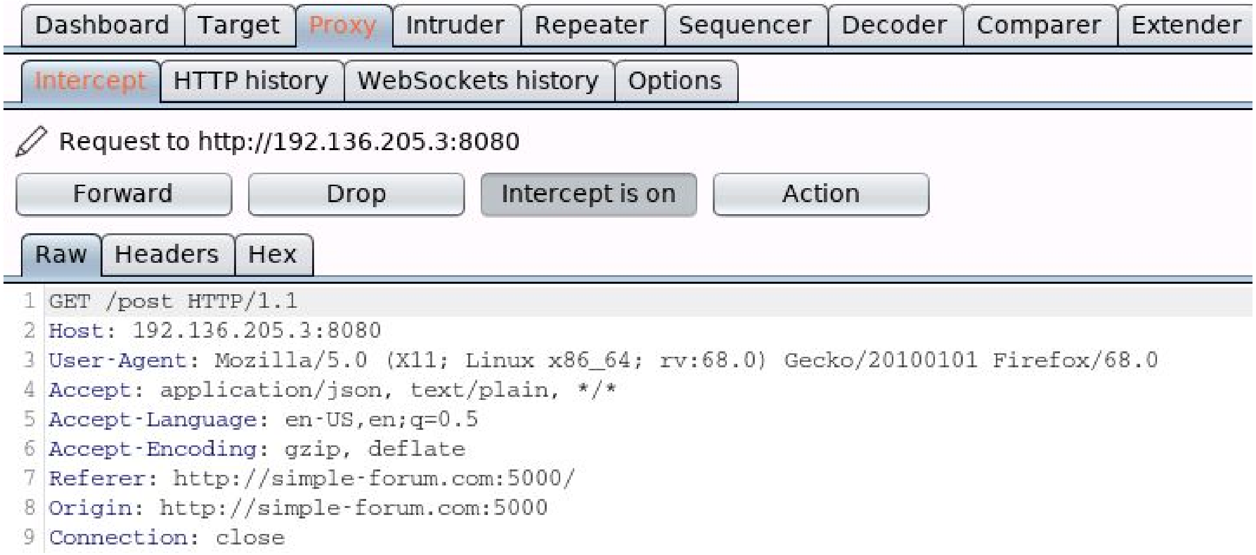
In the challenge description, it is mentioned that that:

- role = 0 (Admin)
- role = 1 (Authenticated)

Also, it is mentioned that the id and role parameters are the same.

So, this account belongs to an authenticated user.

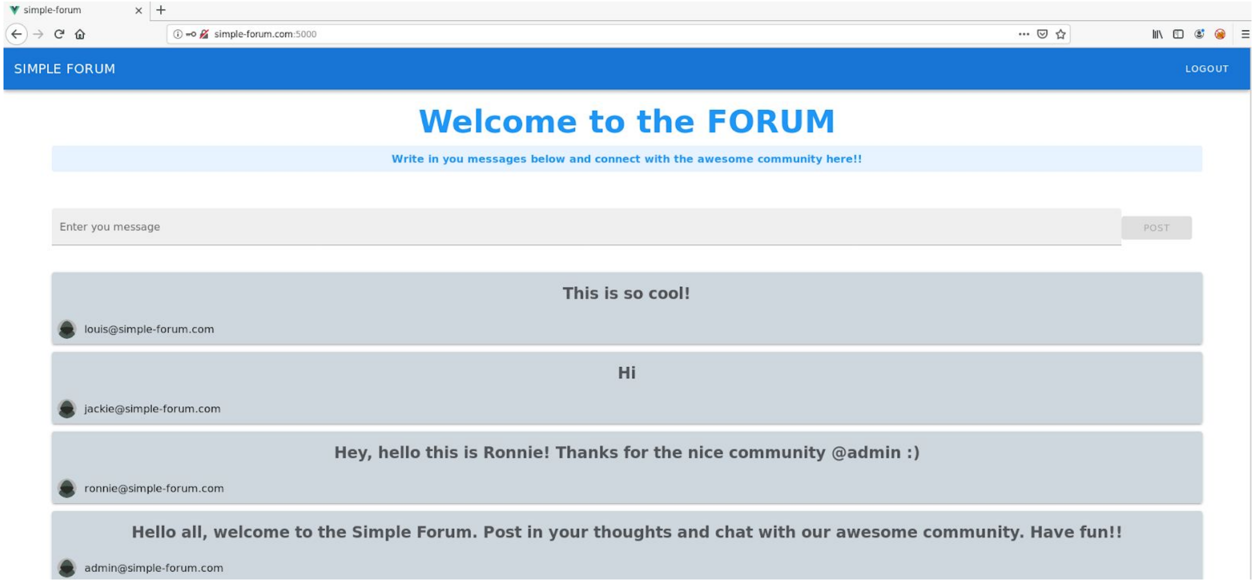
Now, forward the above intercepted request (/post):



```
GET /post HTTP/1.1
Host: 192.136.205.3:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://simple-forum.com:5000/
Origin: http://simple-forum.com:5000
Connection: close
```

Forward the above request and turn off the intercept mode in Burp.

Check the response in the webapp:



Welcome to the FORUM

Write in you messages below and connect with the awesome community here!!

Enter you message

This is so cool!
louis@simple-forum.com

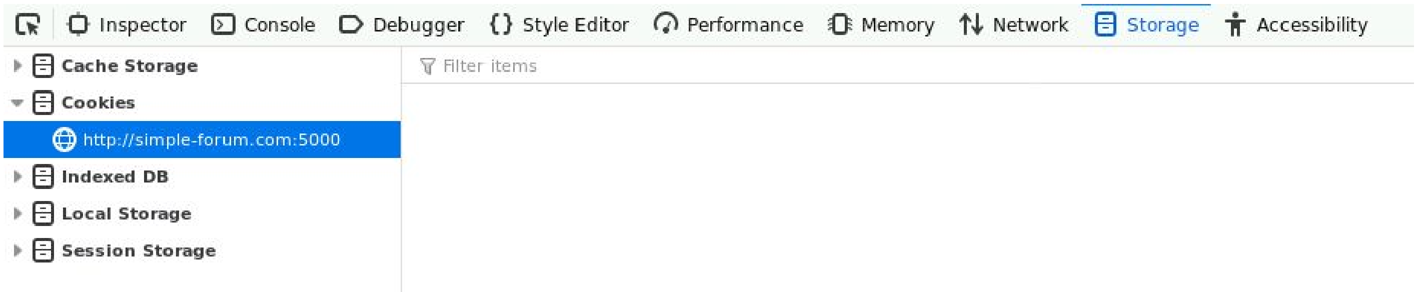
Hi
jackie@simple-forum.com

Hey, hello this is Ronnie! Thanks for the nice community @admin :)
ronnie@simple-forum.com

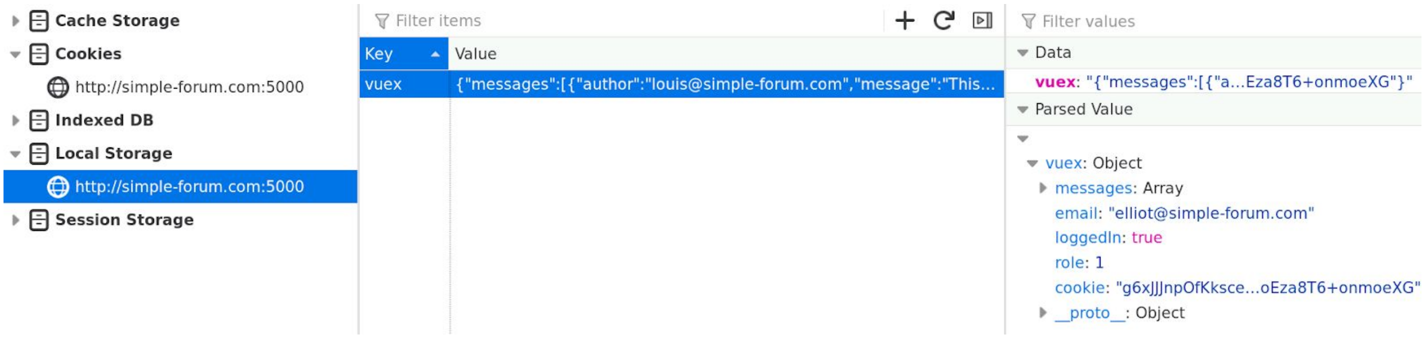
Hello all, welcome to the Simple Forum. Post in your thoughts and chat with our awesome community. Have fun!!
admin@simple-forum.com

Step 4: Inspecting and modifying the local storage to become admin.

Open the inspector window (CTRL + Shift + I) and check the Storage tab:



Check the Local Storage tab:

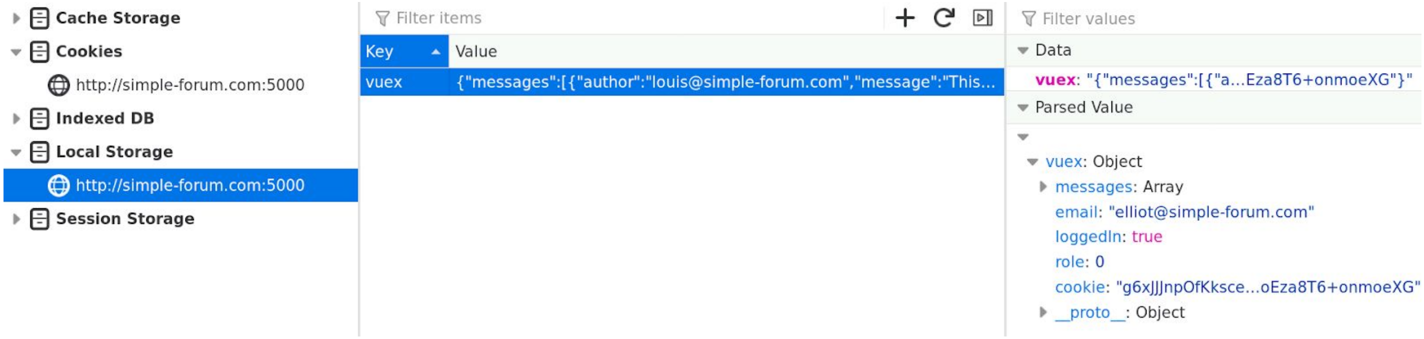


Key	Value
vuex	{"messages":[{"author":"louis@simple-forum.com","message":"This...

```
Data
  vuex: "{"messages":[{"a...Eza8T6+onmoeXG"}"
Parsed Value
  vuex: Object
    messages: Array
    email: "elliot@simple-forum.com"
    loggedIn: true
    role: 1
    cookie: "g6xJJJnpOfKksce...oEza8T6+onmoeXG"
    __proto__: Object
```

Notice that there is only one entry of vuex. Notice that there is a role parameter in the local storage.

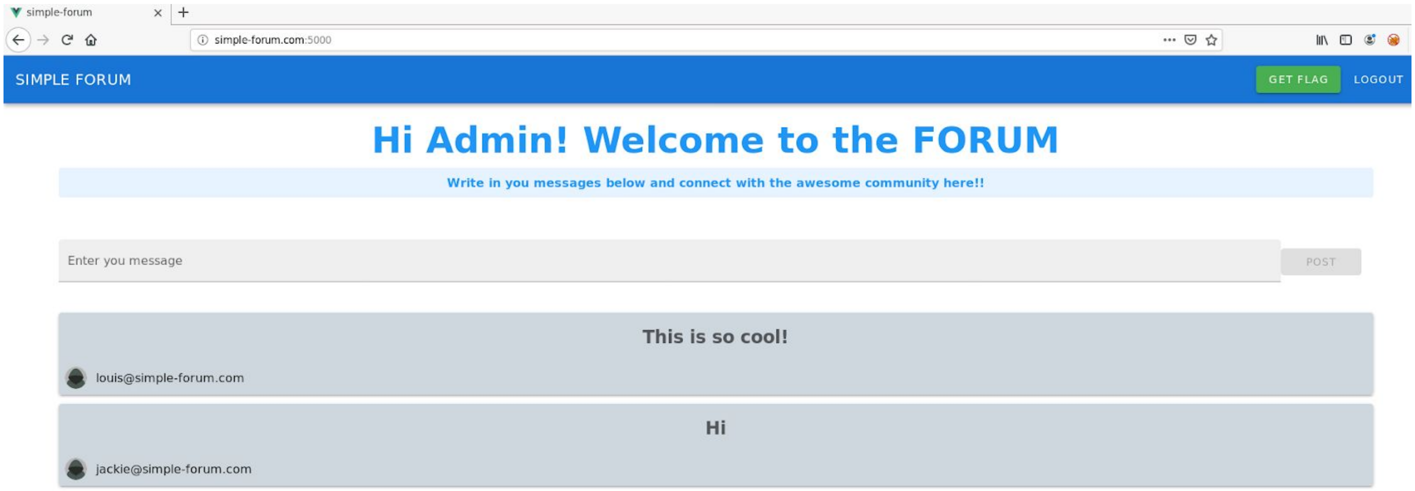
Double click on the value of that entry to edit the role and set it to 0 (admin):



Key	Value
vuex	{"messages":[{"author":"louis@simple-forum.com","message":"This...

```
Data
  vuex: "{"messages":[{"a...Eza8T6+onmoeXG"}"
Parsed Value
  vuex: Object
    messages: Array
    email: "elliot@simple-forum.com"
    loggedIn: true
    role: 0
    cookie: "g6xJJJnpOfKksce...oEza8T6+onmoeXG"
    __proto__: Object
```

Now, since the role is set to 0, refresh the page.



Hi Admin! Welcome to the FORUM

Write in you messages below and connect with the awesome community here!!

This is so cool!

louis@simple-forum.com

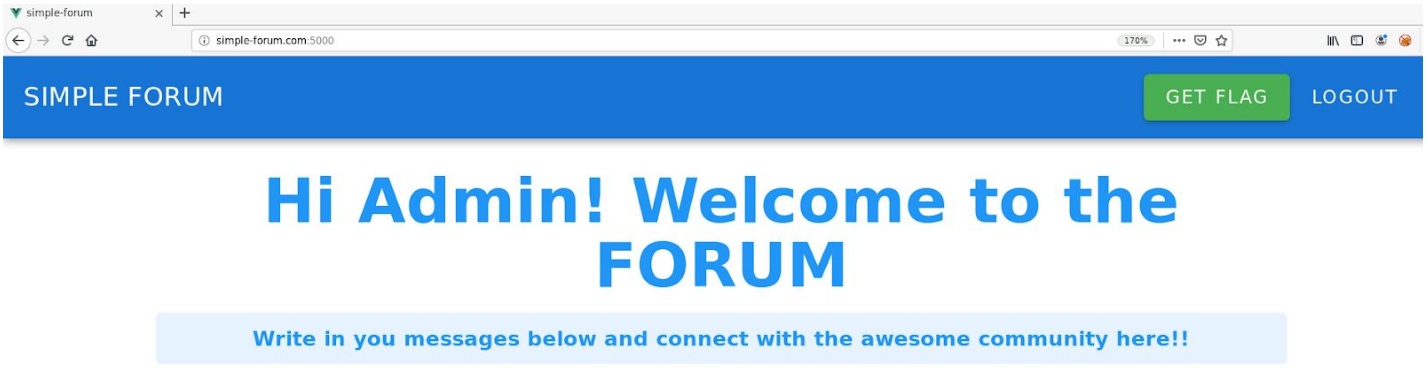
Hi

jackie@simple-forum.com

Notice that role is reflected in the webapp (admin).

Step 5: Retrieving the flag.

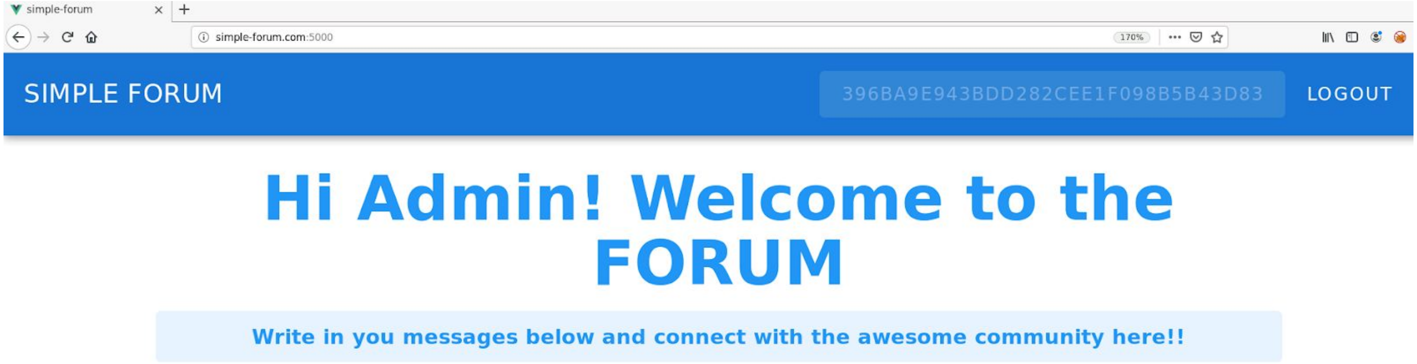
Notice that there is a button to get the flag on the top right.



Hi Admin! Welcome to the FORUM

Write in you messages below and connect with the awesome community here!!

Click on it to get the flag.



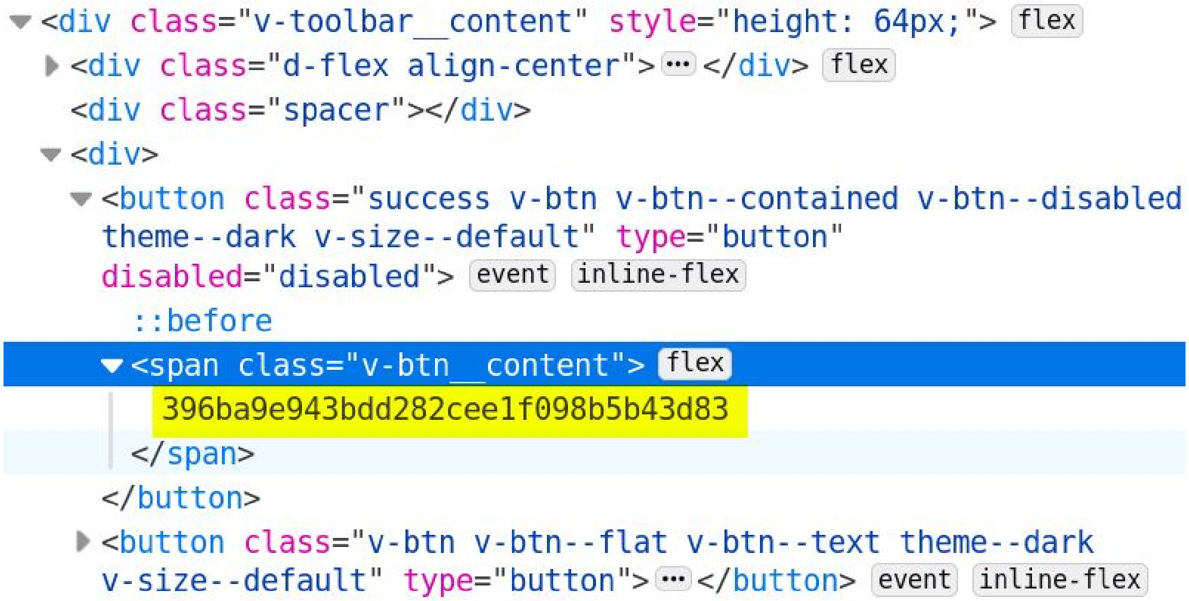
396BA9E943BDD282CEE1F098B5B43D83

Hi Admin! Welcome to the FORUM

Write in you messages below and connect with the awesome community here!!

The flag is revealed but it cannot be copied.

Inspect that element to get the flag value:



```
<div class="v-toolbar__content" style="height: 64px;"> flex
  <div class="d-flex align-center">…</div> flex
  <div class="spacer"></div>
  <div>
    <button class="success v-btn v-btn--contained v-btn--disabled
    theme--dark v-size--default" type="button"
    disabled="disabled"> event inline-flex
      ::before
      <span class="v-btn__content"> flex
        396ba9e943bdd282cee1f098b5b43d83
      </span>
    </button>
    <button class="v-btn v-btn--flat v-btn--text theme--dark
    v-size--default" type="button">…</button> event inline-flex
```

Flag: 396ba9e943bdd282cee1f098b5b43d83

Conclusion:

Never trust local storage for critical information. It can be easily tampered and therefore states must be maintained using cookie or JWT tokens may be used to provide that information.

References:

1. OWASP Top 10 (https://owasp.org/www-project-top-ten/)
2. Broken Authentication (https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A2-Broken_Authentication.html)