Introduction:

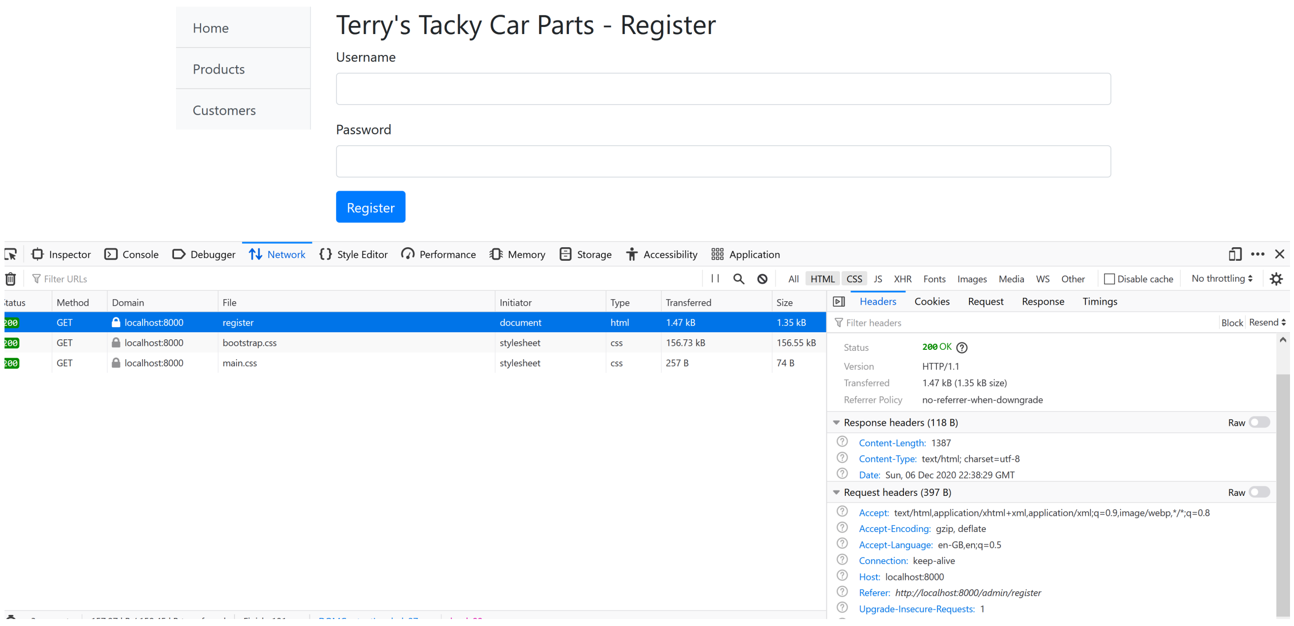
Thanks sharing challenge to be completed for Senior Cloud Security Engineer role.

I thoroughly enjoyed to work on this exercise and including my findings in this document scripts written in Go to exploit security vulnerability

Startup:
I followed the steps mentioned in Readme documentation to unpack and setup vulnerable-service and able to run it my local windows environment - Windows WSL2 - Ubuntu

Analysis:
I analysed Terry's Tacky Car Parts application and able to browse through various pages– Home, Products, Customers and functions to Login and Register.

I tried to register new user through register screen, but not seen any functionality enabled for registration. I monitored the network traffic being passed between URLs and not really seen any data being passed, no cookies were being transferred. Just seen HTTP 200 response from server.



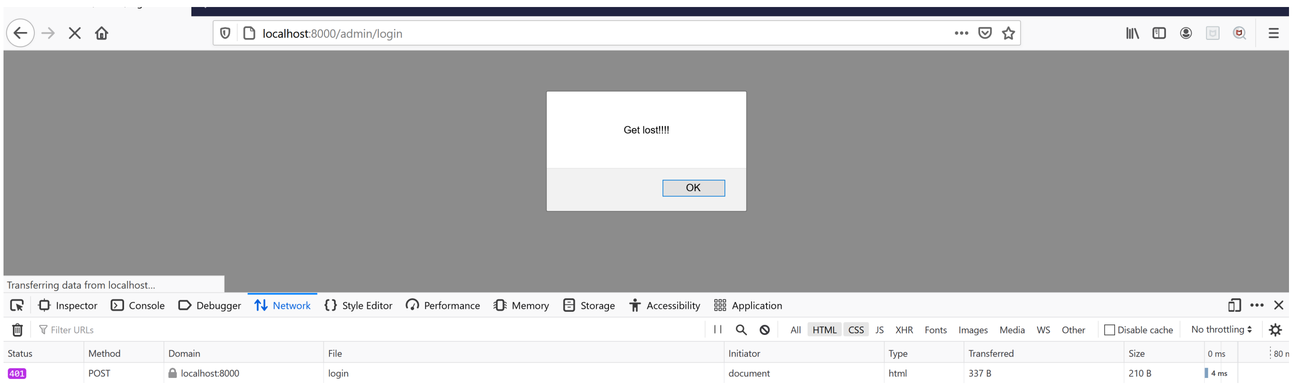
Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	localhost:8000	register	document	html	1.47 kB	1.35 kB
200	GET	localhost:8000	bootstrap.css	stylesheet	css	156.73 kB	156.55 kB
200	GET	localhost:8000	main.css	stylesheet	css	257 B	74 B

```
Status            200 OK
Version           HTTP/1.1
Transferred       1.47 kB (1.35 kB size)
Referrer Policy   no-referrer-when-downgrade

Response headers (118 B)
Content-Length: 1387
Content-Type: text/html; charset=utf-8
Date: Sun, 06 Dec 2020 22:38:29 GMT

Request headers (397 B)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en;q=0.5
Connection: keep-alive
Host: localhost:8000
Referer: http://localhost:8000/admin/register
Upgrade-Insecure-Requests: 1
```

I also observed functionality surrounding “Login” functionality. As attaching in below screen, I just seen “401” HTTP response from server as attached in below screen and getting redirected to “login” screen again. I also did similar analysis over Products and Customer pages.   Apart form navigation to same login screen, I didn't noticed any functional flows between these pages.



localhost:8000/admin/login

Get lost!!!!

OK

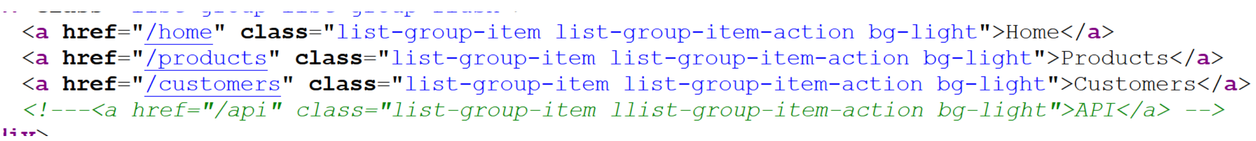
Transferring data from localhost...

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application

```
Status  Method  Domain          File   Initiator  Type  Transferred  Size
401     POST    localhost:8000  login  document   html  337 B        210 B   4 ms
```

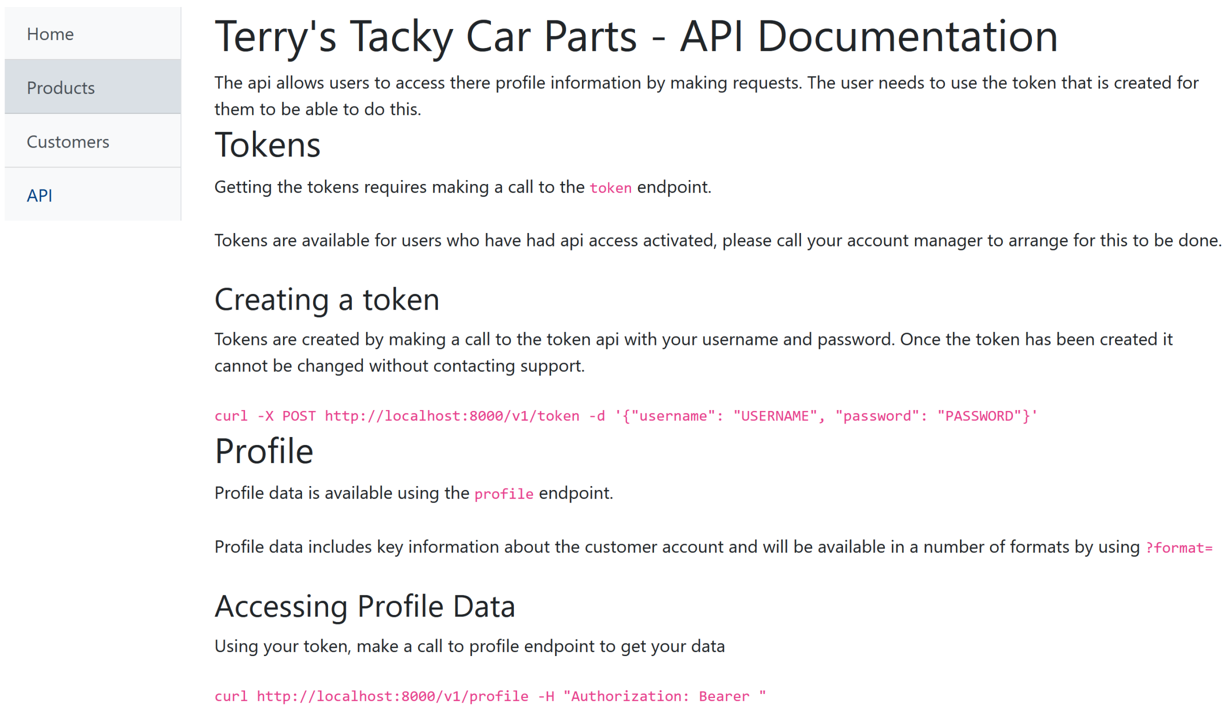
Therefore, I came to conclusion that there is no need to try out any Injection or Broken Authentication attacks.

I continued looking source code of html displayed in browser and noticed following commented lined being visible in source code.



```
  <a href="/home" class="list-group-item list-group-item-action bg-light">Home</a>
  <a href="/products" class="list-group-item list-group-item-action bg-light">Products</a>
  <a href="/customers" class="list-group-item list-group-item-action bg-light">Customers</a>
  <!---<a href="/api" class="list-group-item llist-group-item-action bg-light">API</a> -->
```

So, I tried accessing URL http://localhost:8000/api and able to see active webpage



Home
Products
Customers
API

Terry's Tacky Car Parts - API Documentation

The api allows users to access there profile information by making requests. The user needs to use the token that is created for them to be able to do this.

Tokens

Getting the tokens requires making a call to the `token` endpoint.

Tokens are available for users who have had api access activated, please call your account manager to arrange for this to be done.

Creating a token

Tokens are created by making a call to the token api with your username and password. Once the token has been created it cannot be changed without contacting support.

```
curl -X POST http://localhost:8000/v1/token -d '{"username": "USERNAME", "password": "PASSWORD"}'
```

Profile

Profile data is available using the `profile` endpoint.

Profile data includes key information about the customer account and will be available in a number of formats by using `?format=`

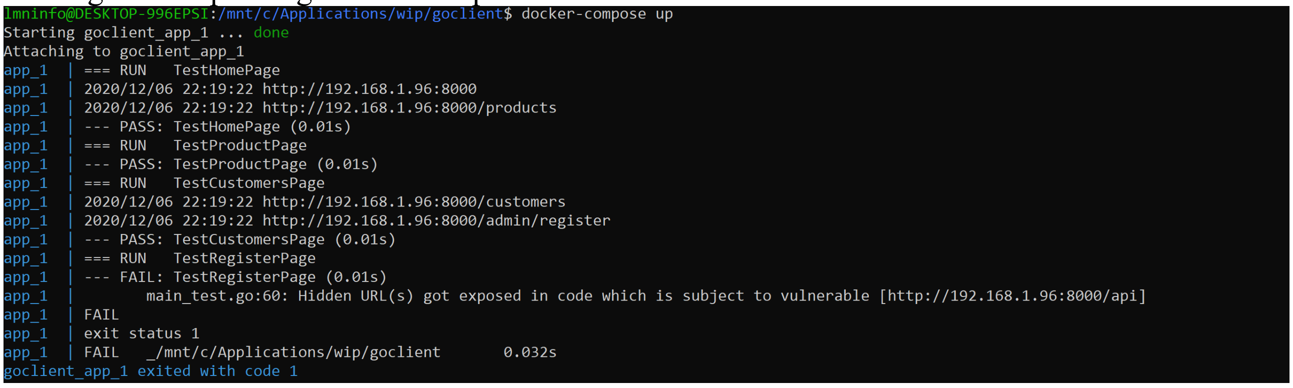
Accessing Profile Data

Using your token, make a call to profile endpoint to get your data

```
curl http://localhost:8000/v1/profile -H "Authorization: Bearer "
```

Noticed that this page contains additional information about how to generate tokens and accessing profile data which are possible candidates to go further exploiting service.   I would think this is the flag which has been buried in application which I would think comes under "Sensitive data exposure" attack.

I also done same exercise in Go language and able list down all URLs vs Active URL's and wrote few tests to find this vulnerability. Attaching screenshot of the output in my local machine after running test script using docker-compose.



```
lmninfo@DESKTOP-996EPSI:/mnt/c/Applications/wip/goclient$ docker-compose up
Starting goclient_app_1 ... done
Attaching to goclient_app_1
app_1  | === RUN   TestHomePage
app_1  | 2020/12/06 22:19:22 http://192.168.1.96:8000
app_1  | 2020/12/06 22:19:22 http://192.168.1.96:8000/products
app_1  | --- PASS: TestHomePage (0.01s)
app_1  | === RUN   TestProductPage
app_1  | --- PASS: TestProductPage (0.01s)
app_1  | === RUN   TestCustomersPage
app_1  | 2020/12/06 22:19:22 http://192.168.1.96:8000/customers
app_1  | 2020/12/06 22:19:22 http://192.168.1.96:8000/admin/register
app_1  | --- PASS: TestCustomersPage (0.01s)
app_1  | === RUN   TestRegisterPage
app_1  | --- FAIL: TestRegisterPage (0.01s)
app_1  |        main_test.go:60: Hidden URL(s) got exposed in code which is subject to vulnerable [http://192.168.1.96:8000/api]
app_1  | FAIL
app_1  | exit status 1
app_1  | FAIL   _/mnt/c/Applications/wip/goclient       0.032s
goclient_app_1 exited with code 1
```

Attaching source code in my Github account along with steps.