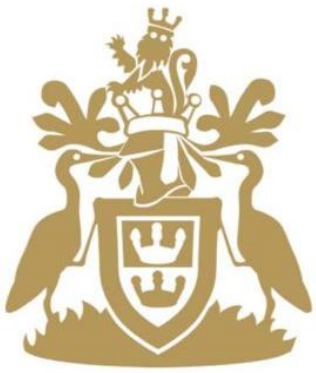


Anglia Ruskin University

Web Application Security

2022 MOD006363 TRI2 F01CAM

011 Element Logbook

SUBMITTED BY

SID: 2160346

DATE: 20-04-2023

TABLE OF CONTENTS

3 stars: Login Amy - Log in with Amy's original user credentials**.** Weekly Reflection – Week 6	
WEEK 7 Catch Up Week	34
WEEK 8 1 star: Error Handling - Provoke an error that is neither very gracefully nor consistently handled. 1 star: Privacy Policy - Read our privacy policy. Weekly Reflection – Week 8	35
WEEK 9 1 star: DOM XSS - Perform a DOM XSS attack. 1 star: Bonus Payload - Use the bonus payload. Weekly Reflection – Week 9	38
WEEK 10 1 star: Chatbot abuse - Find the chatbot and ask it to get a discount. Weekly Reflection – Week 10	40
WEEK 11 4 star: Easter Egg - Find the hidden Easter Egg. 4 star: Nested Easter Egg - Apply some advanced crypt-analysis to find the "REAL EASTER EGG". Weekly Reflection – Week 11	41
WEEK 12 Catch Up Week	48

LIST OF FIGURES

WEEK 1:

1.1 Overview of OWASP top 10 write up

INJECTION

The attackers send data through query or command to the interpreter, the attackers can easily fetch all information from the database. [OWASP Foundation, 2017]

BROKEN AUTHENTICATION

The poor implemented authentication and session, the attackers can hack the information and access the application using the login information of other users. [OWASP Foundation, 2017]

SENSITIVE DATA EXPOSURE

Sensitive data means, the data to be protected from unauthorized access. The data means personal information, bank details etc. The hackers can hijack the user data because the data not to be properly encrypted. [OWASP Foundation, 2017]

XML EXTERNAL ENTITIES (XXE)

The entity value could replace by malicious data from remote entities. The attackers can access xml data too. [OWASP Foundation, 2017]

BROKEN ACCESS CONTROL

It enables unauthenticated viewing or editing of sensitive data by any user. Using these issues, the hacker might access the data without the required authorizations. [OWASP Foundation, 2017]

SECURITY MISCONFIGURATION

When application or system configuration are implemented incorrectly or either missing it providing unauthorised access. So, the application should be configured securely and security update should be performed when available. [OWASP Foundation, 2017]

CROSS-SITE SCRIPTING (XSS)

The attackers inject the malicious code in the web site or web application. If the webpage is don't validate properly the attackers can easily hijack the application. [OWASP Foundation, 2017]

INSECURE DESERIALIZATION

When serialized data is converted back to objects there are chances that the intruder can make changes to the data which can lead to malfunctioning of the application. [OWASP Foundation, 2017]

USING COMPONENTS WITH KNOWN VULNERABILITIES

components that have vulnerabilities if used to build an application increase the chances for attack from attackers using these vulnerabilities as a way to break into the system that can lead to data loss or server take over. [OWASP Foundation, 2017]

INSUFFICIENT LOGGING & MONITORING

Limited logging and monitoring of ongoing traffic in an application will be a disadvantage as it can lead to suspicious activities being carried out without noticing in the system to gain control of the system. [OWASP Foundation, 2017]

1.2 Weekly Reflection

- In this week learn about setup web pod.
- learn how to used burp in web application security.
- The burp is a web application testing platform.
- Studied about Web application security vulnerability OWASP.
- OWASP stands for the **Open Web Application Security Project.**
  [OWASP Foundation, 2017]

WEEK 2:

2.1 Hacker Test HTML Labs 1-10

Level 1

To complete level 1 should find the password, first right click the page (figure 1) and select view page source. Search the password in page source shown in the figure 2, the password is null in line number 51. Type the password in textbox and click login.

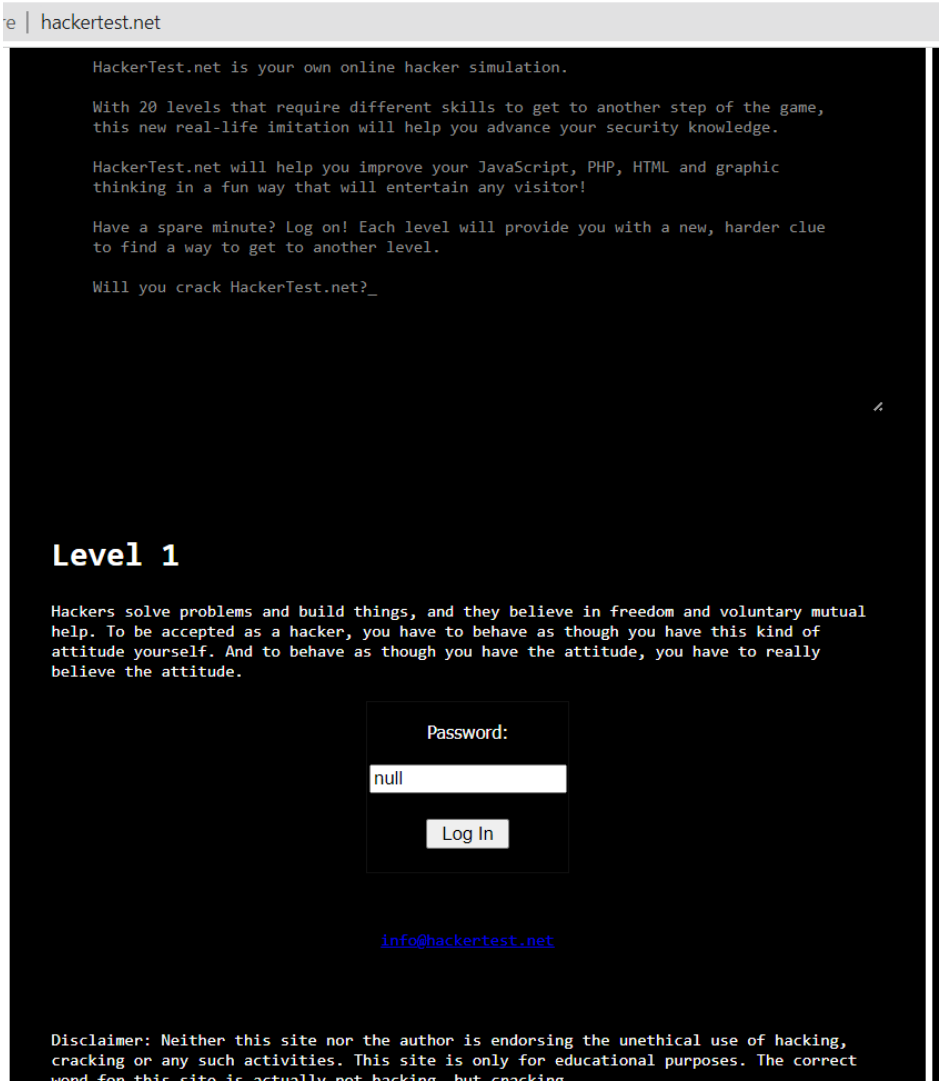


Figure 1 level 1 page

Figure 2 level 1 page source

Level 2

View source code and the password in line number 14(figure 3) that is 131. Enter the password on prompt as shown in Figure 4.

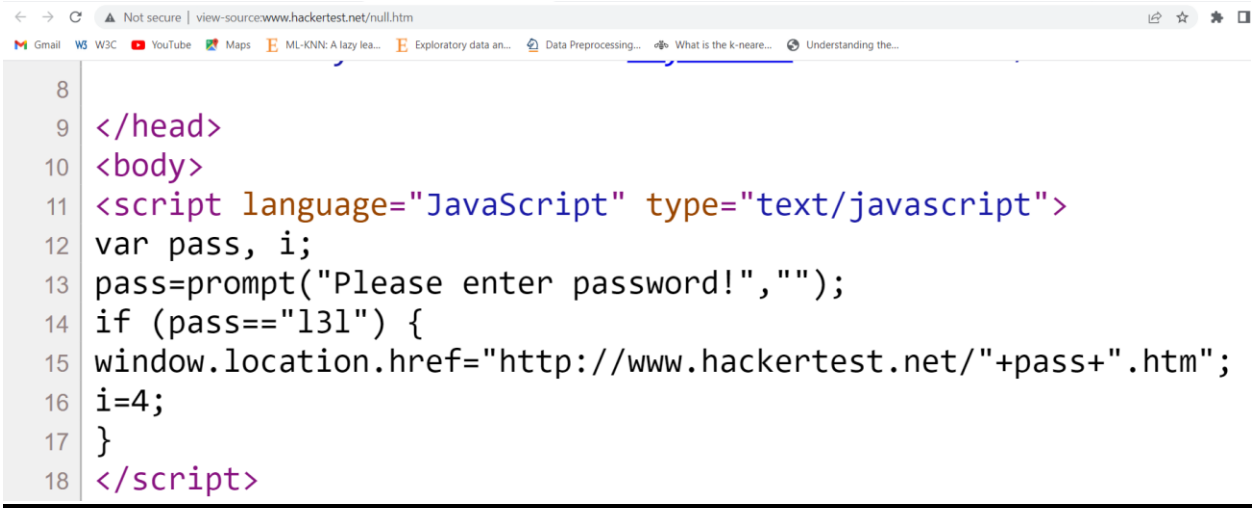


Figure 3 level 2 page source

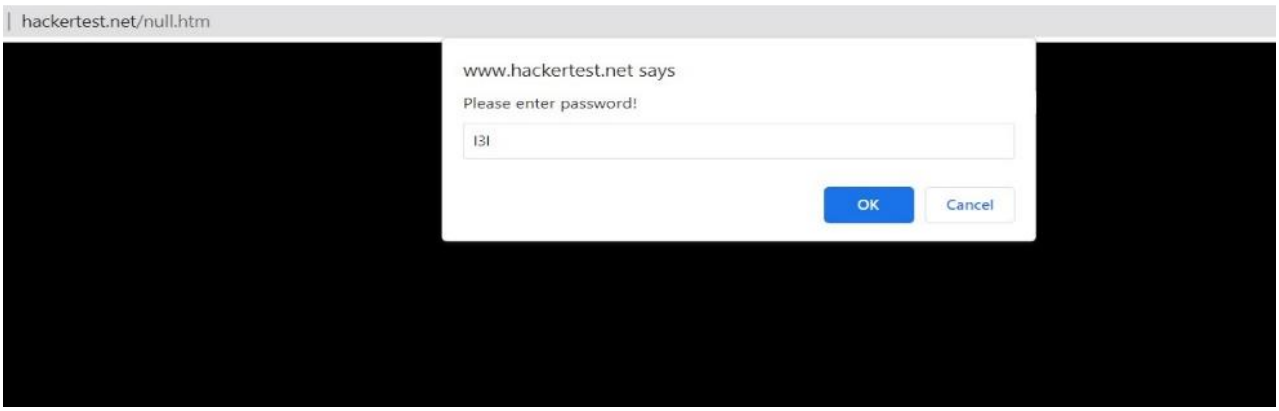


Figure 4 Entering level 2 password

Level 3

On viewing the source code, the password must be equivalent to alink color. So, the password is in line number 10 in source code (Figure 5) #000000. Enter the password in figure 6 and level 3 successfully completed.

Figure 5 Level 3 page source

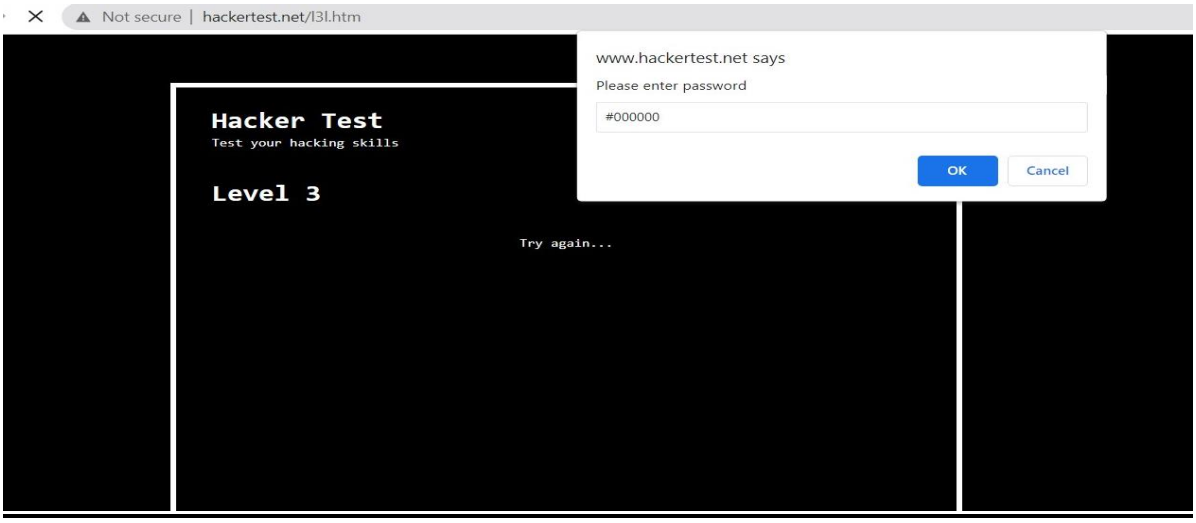


Figure 6 Entering level 3 password

Level 4

Viewing the page source code and find the link that link shown in the figure 8, Click the link and go to next level.

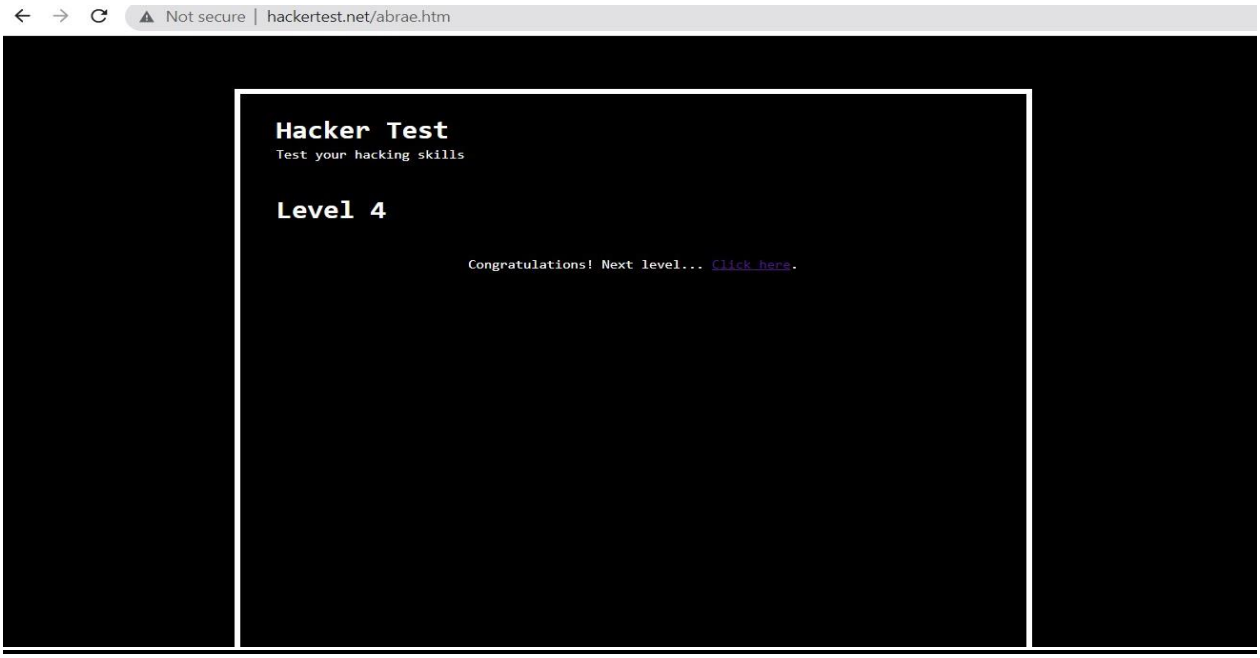


Figure 7 Level 4 page

Figure 8 Level 4 page source

Level 5

For accessing the level 5 page source first need disable the JavaScript because, by opening the prompt box source code cannot be accessed. After getting the password in line number 12“SAvE-as hELpS a lot” from source code enable the JavaScript then reload the page and enter the password in prompt box.

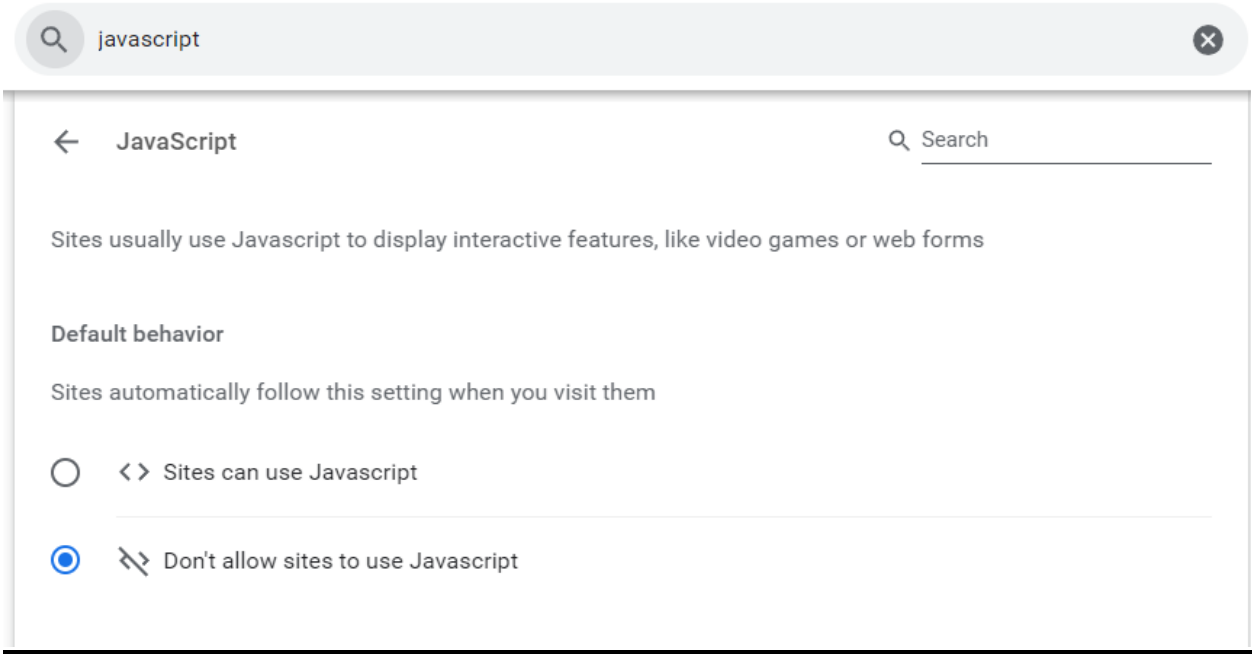


Figure 9 Disable JavaScript

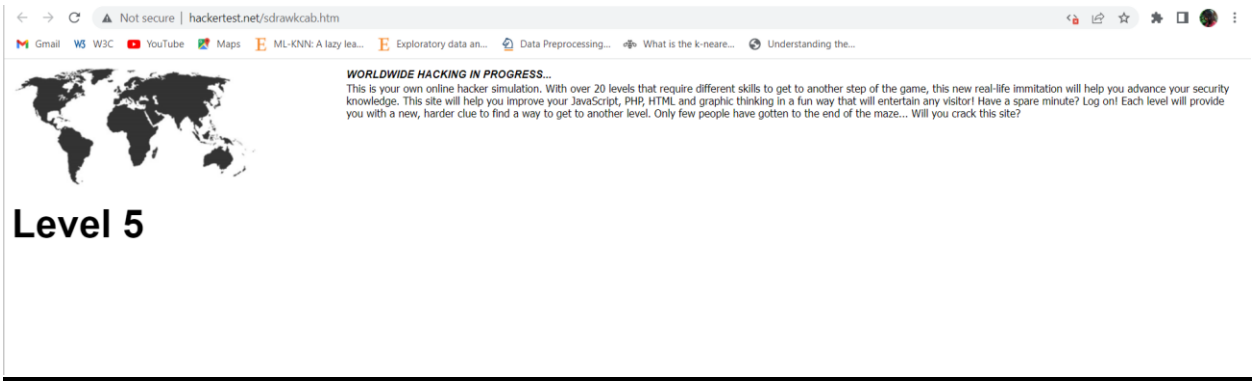


Figure 10 Level 5 page

Figure 11 Level 5 page source

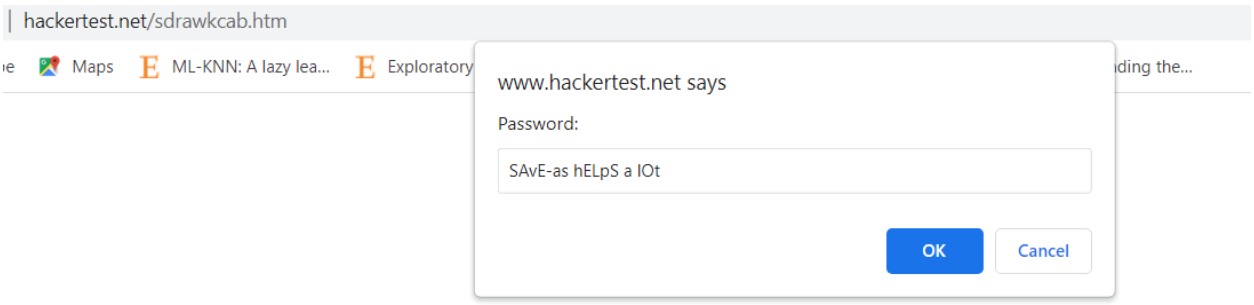


Figure 12 Entering Level 5 password

Level 6

On viewing the source code there is an external JavaScript link in line number 23 as shown in figure 14. Click the link “psswd.js” a new page will open which contains the password as shown in the figure 15. Enter the password in prompt box and click ok.

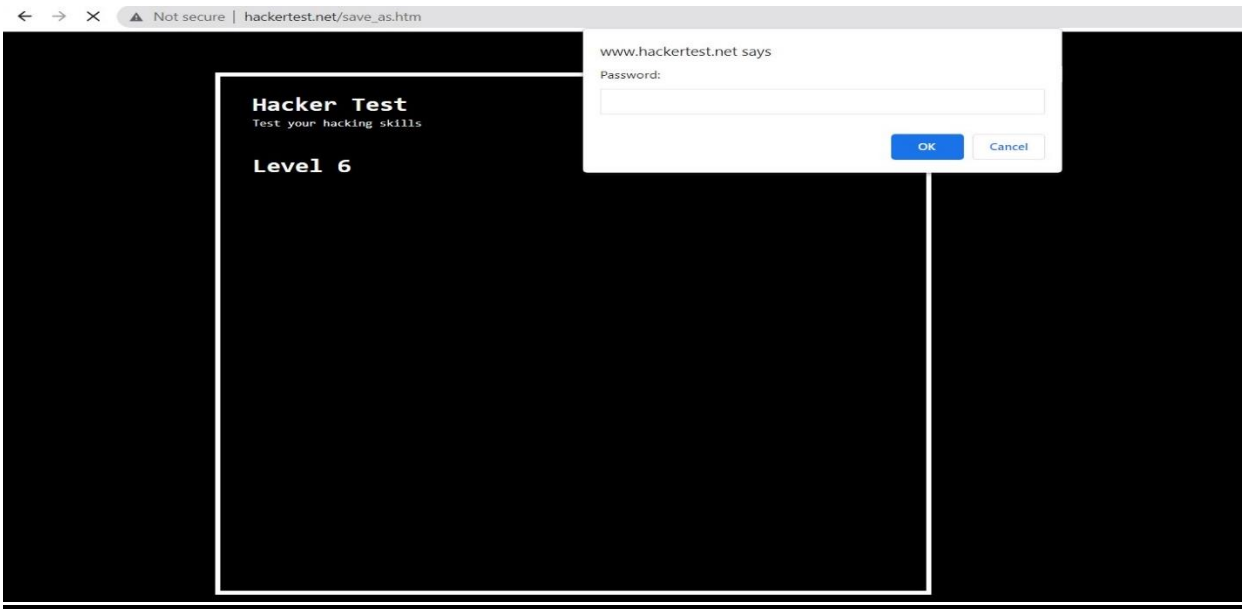


Figure 13 Level 6 page

```
<h1>Level 6</h1><SCRIPT SRC="psswd.js" LANGUAGE="JavaScript" type="text/javascript"></script>

<br><br>

<center><p>Try again ...</p></center>



<!-- Google AdSense was here but now removed -->


<!-- Start of StatCounter Code -->
<script type="text/javascript" language="javascript">
var sc_project=2497325;
var sc_invisible=0;
var sc_partition=23;
var sc_security="66a93703";
var sc_remove_link=1;
</script>
```

Figure 14 Level 6 Page source

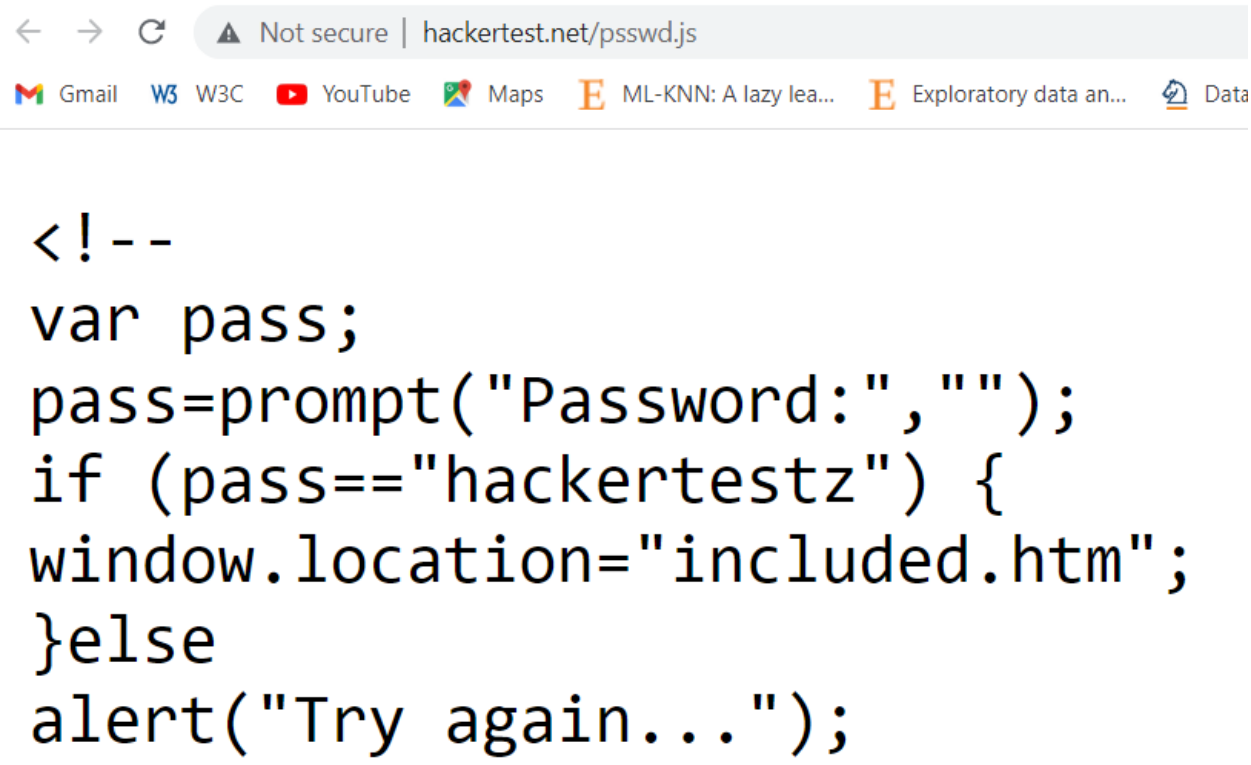


Figure 15 Source code of psswd.js

Level 7

On viewing the source code there is path to an image in line number 11 as shown in figure 16. On opening the image by following the path the user name and password can be seen as shown in figure 17. Enter that user name and password in textbox, then submit.

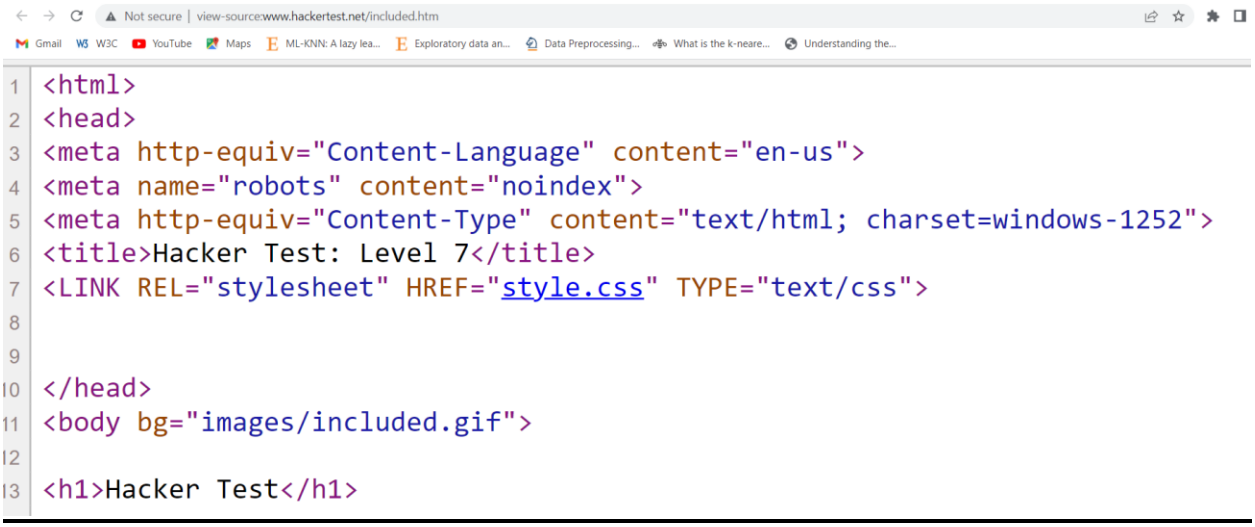


Figure 16 Level 7 source page

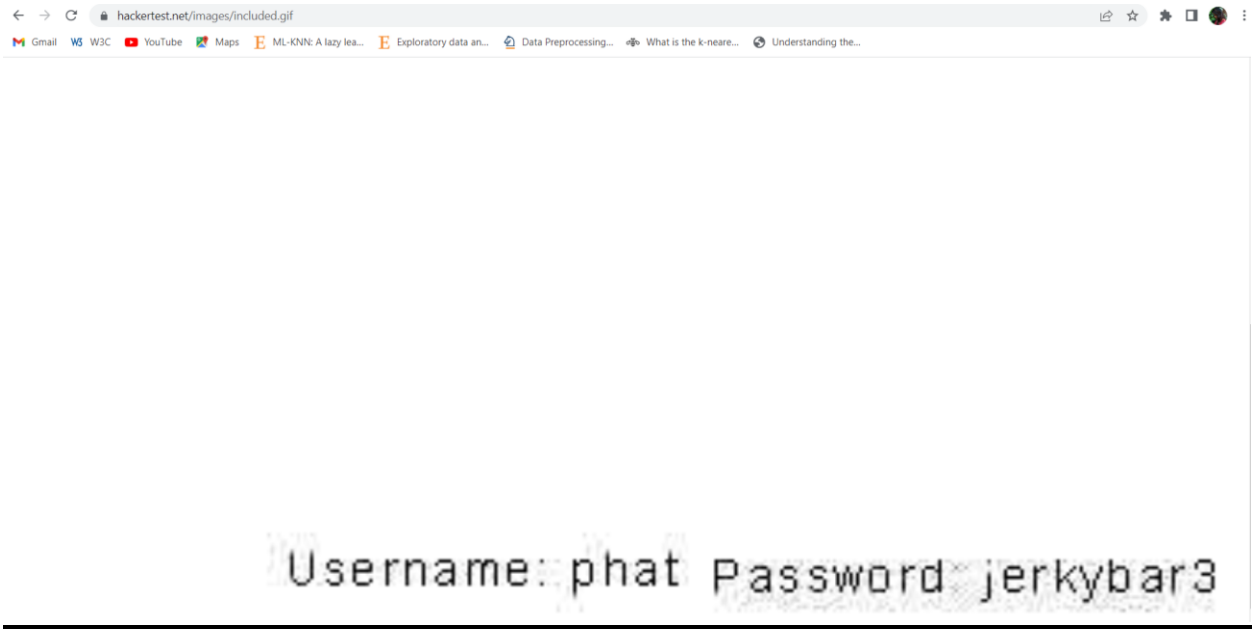


Figure 17 Username and Password for level 7

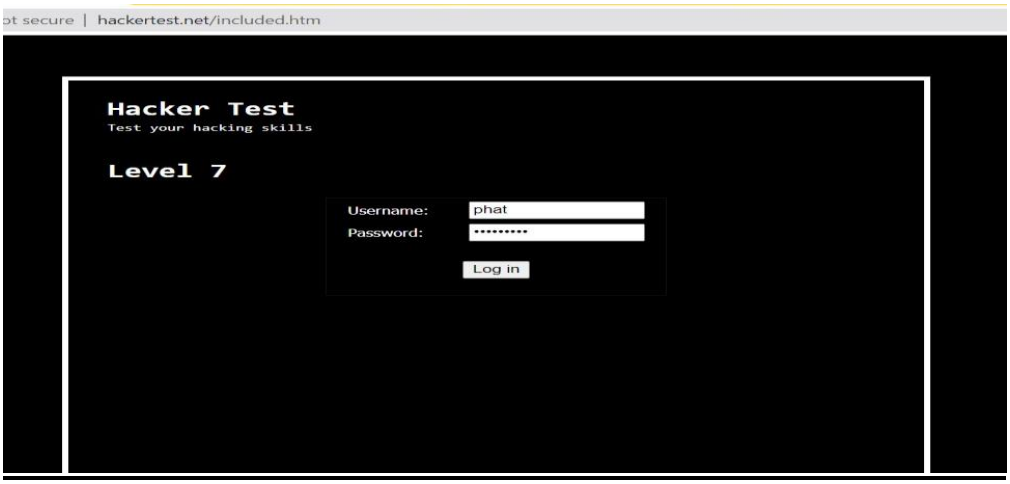


Figure 18 Entering username and password of level 7

Level 8

On viewing the source code there is path in form action in line number 35 as shown in figure 19. On opening the path there is an image path in line number 5 as shown in figure 20. On loading the image using the image path it is said to look for a photoshop doc as shown in figure 21. So, on changing phat.gif to phat.psd (figure 22) photoshop file is downloaded. This file is opened in a Photoshop application as shown in figure 23. Got username and password while unmasking the files figure(23), and login with username and password.

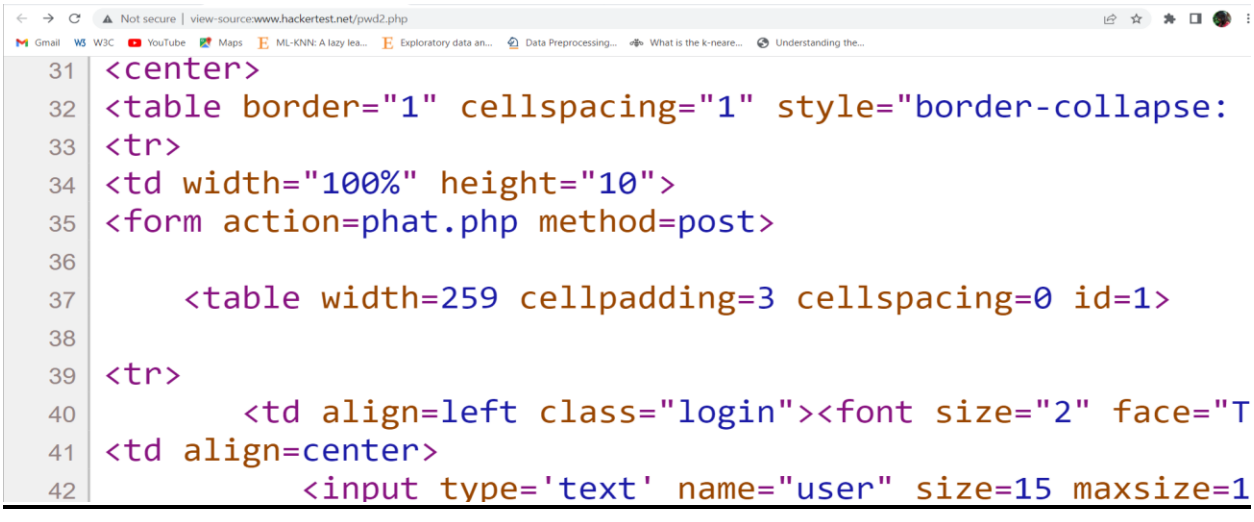


Figure 19 Level 8 page source

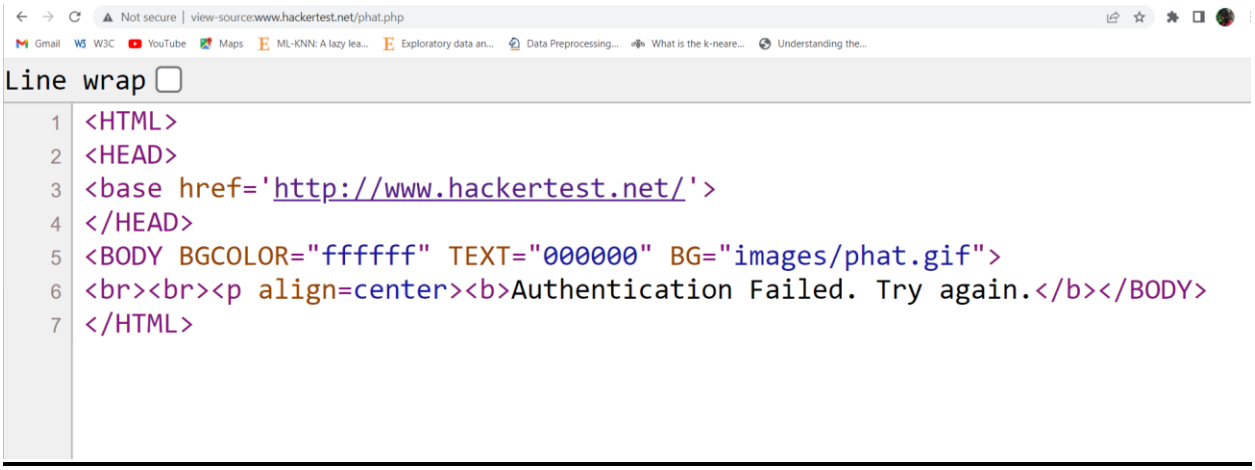


Figure 20 source code of phat.php

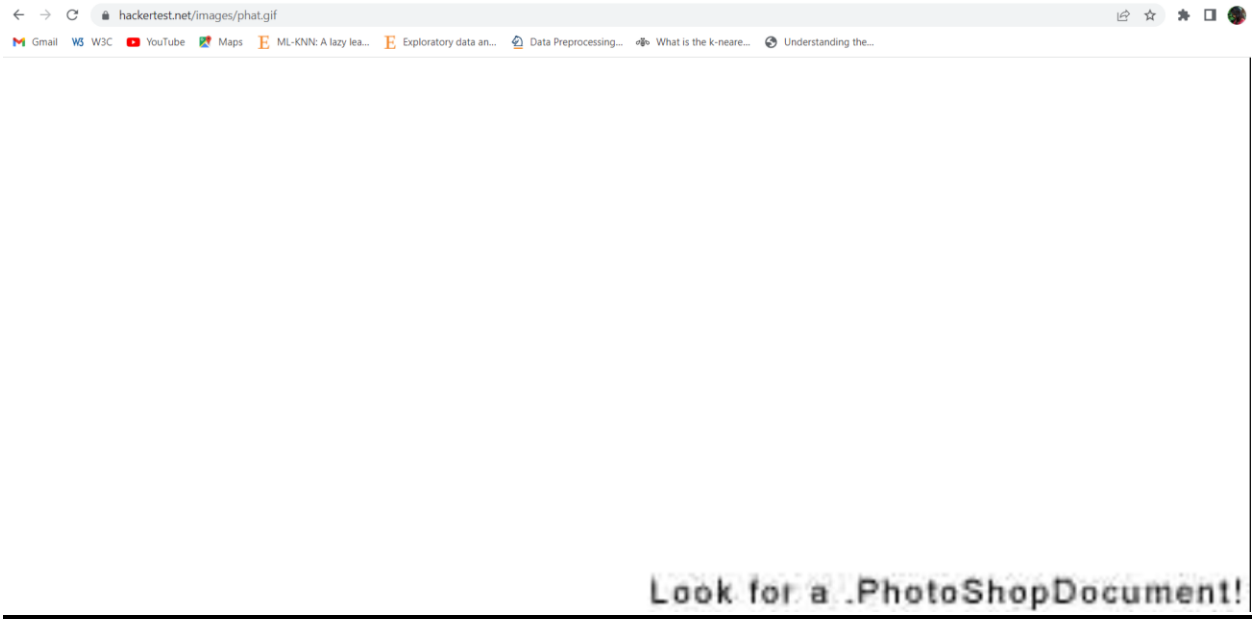


Figure 21 phat.gif image

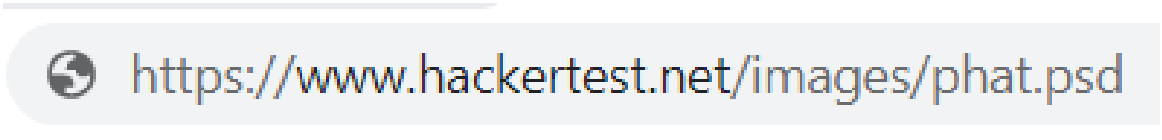


Figure 22 changing gif extention to php

Figure 23 open phat.php using photoshop

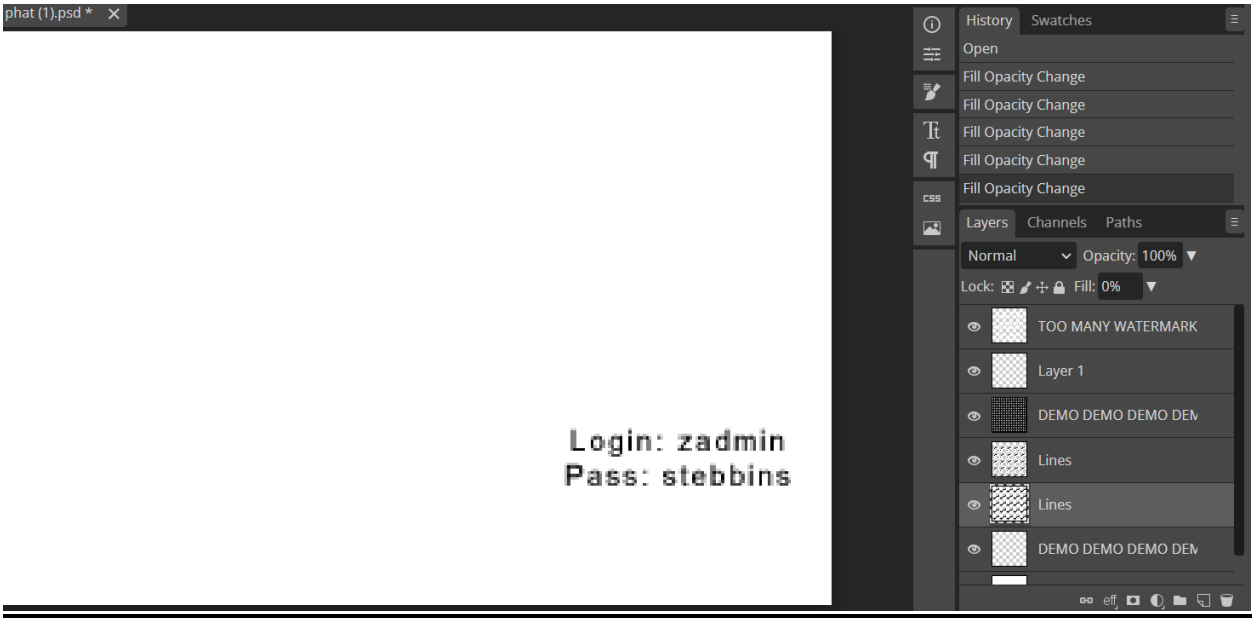


Figure 24 login details obtain after unmasking the filter

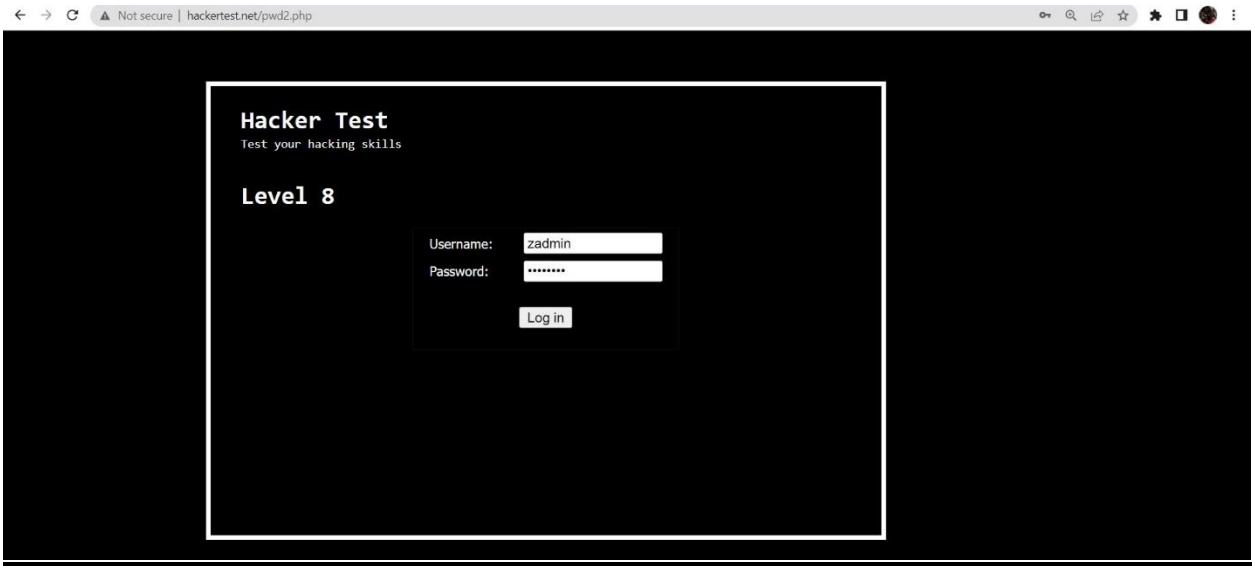


Figure 25 Entering username and Password

Level 9

Here just a text message that is crack the password. On viewing the source code of this page there is about 1500 lines of code. Scroll down to find the clue. The clue is shown in figure 27. Z2F6ZWJydWg= This string ended with = so, there is chance it is abse64 encoded. So this string was decoded using Base64 decoder as shown in figure 28. Figure 27 image also says to add extension so added after the decoded value as shown in figure 29

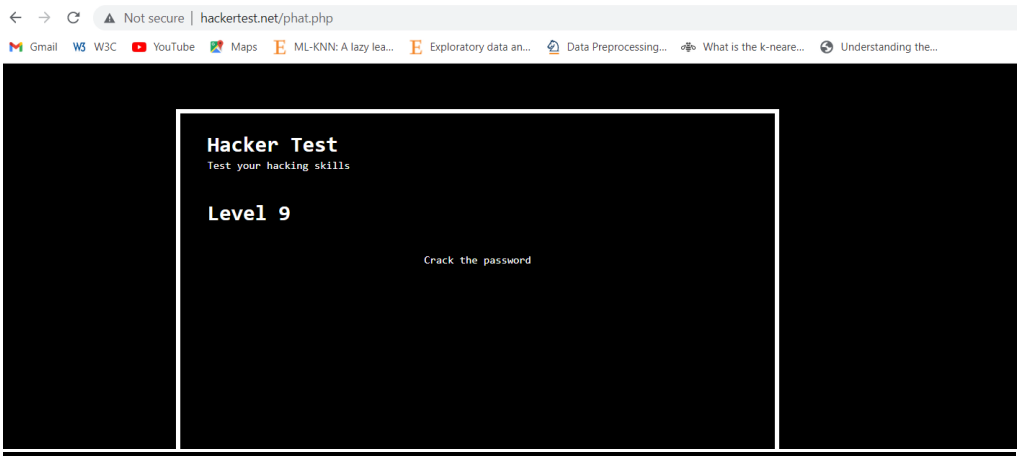


Figure 26 Level 9 page

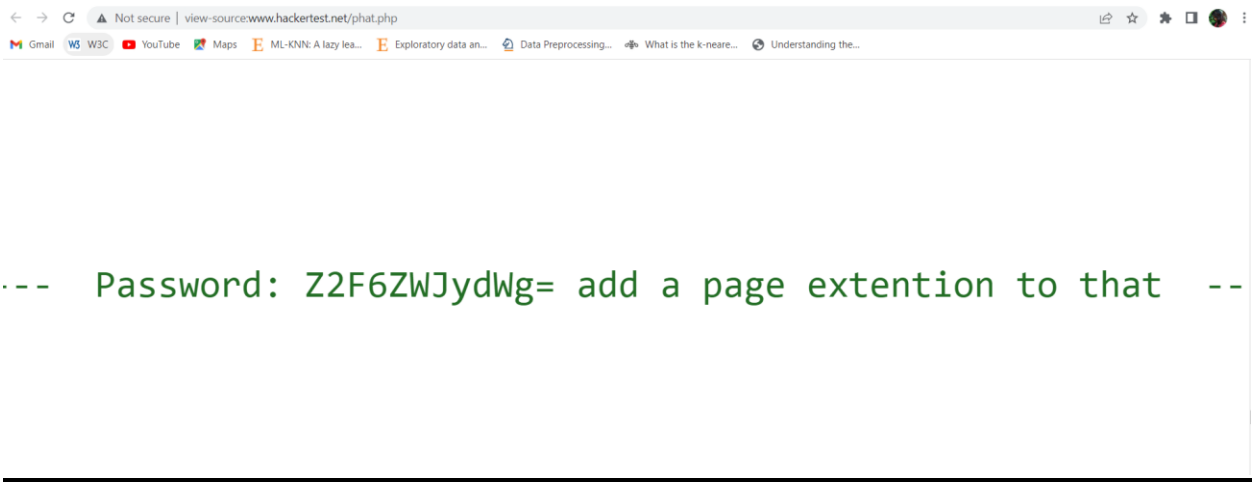


Figure 27 Password hint obtain from level 9 page source

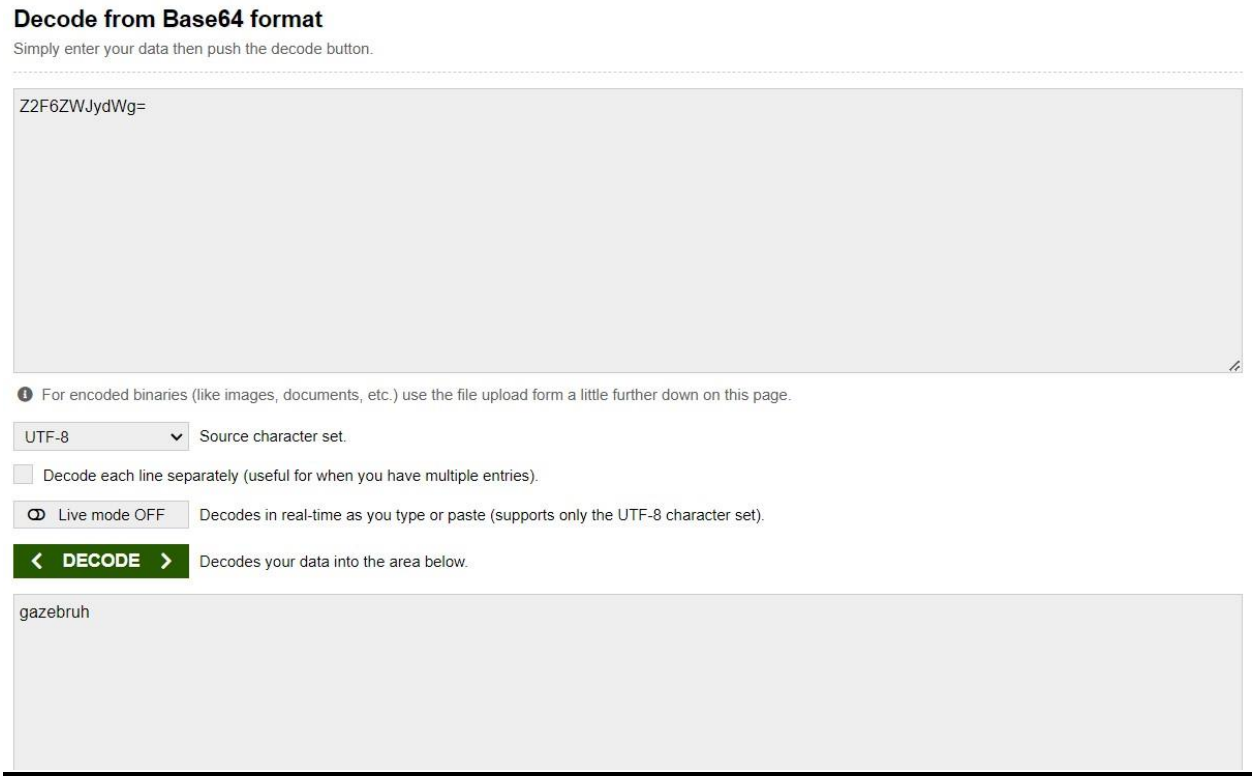


Figure 28 Decoding the string obtain from source code

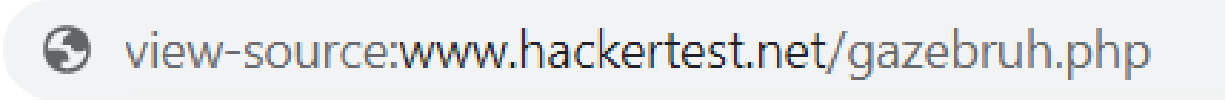


Figure 29 Adding the decoded string along with php extension

Level 10

On viewing code of paragraph in the source code there is some letters in italic form as shown in figure 31. On combining these letters got a word “shackithalf” as shown in figure 32. This word is entered as password as shown Figure 33. On viewing the source code of the new level 10 page(Figure 34) Level 11 page name is obtained as shown in Figure 35.

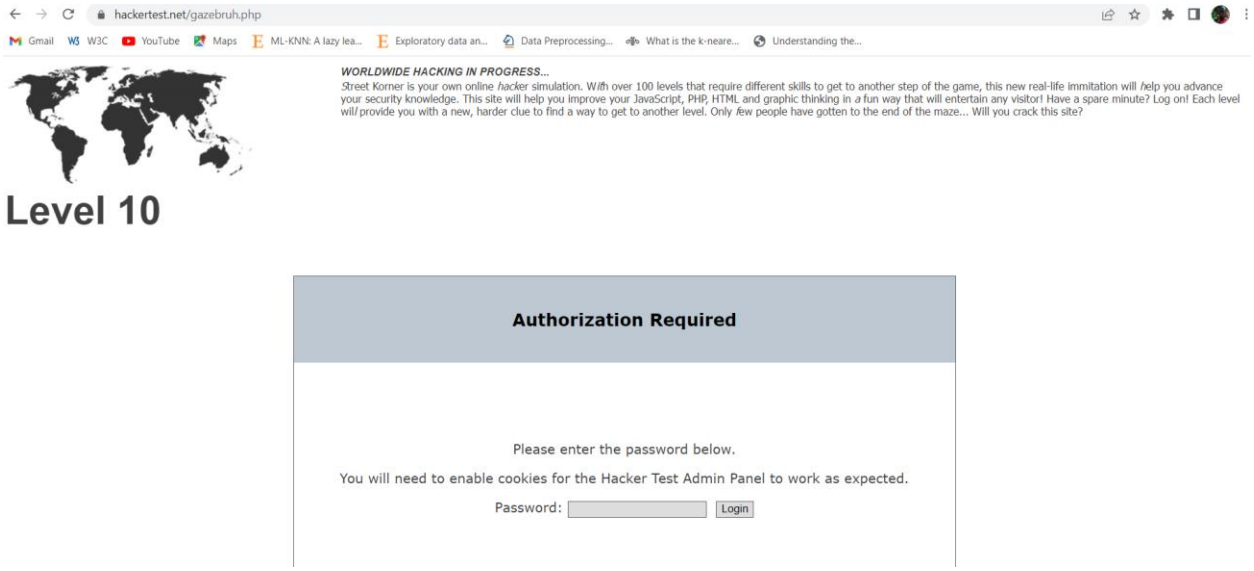


Figure 30 : Level 10 page

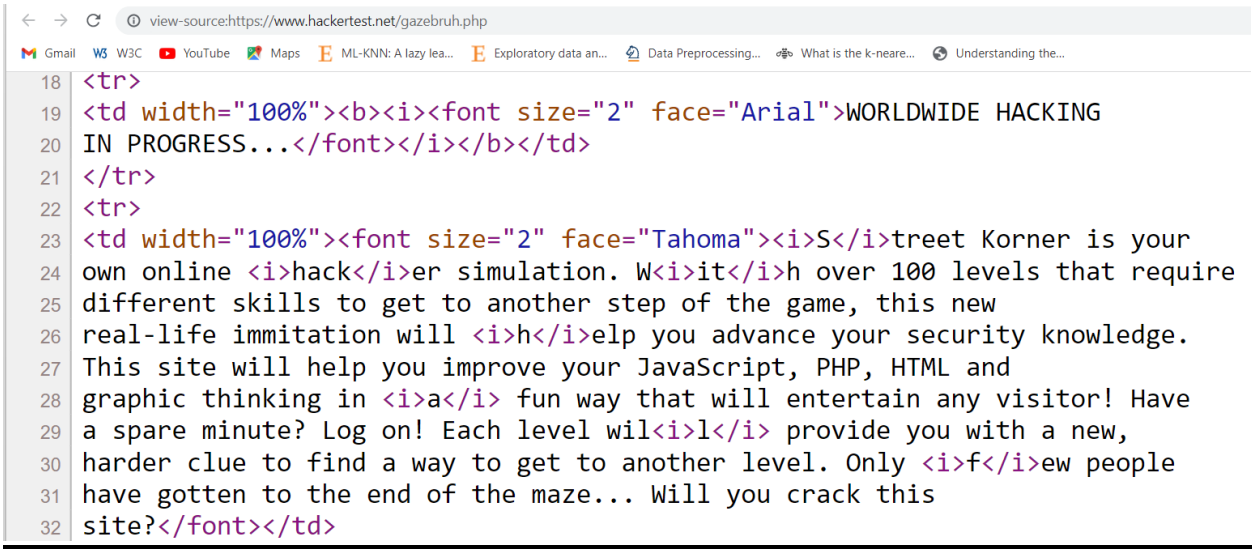


Figure 31 Level 10 page source

shackithalf

Figure 32 Word formed combaining the italic letters

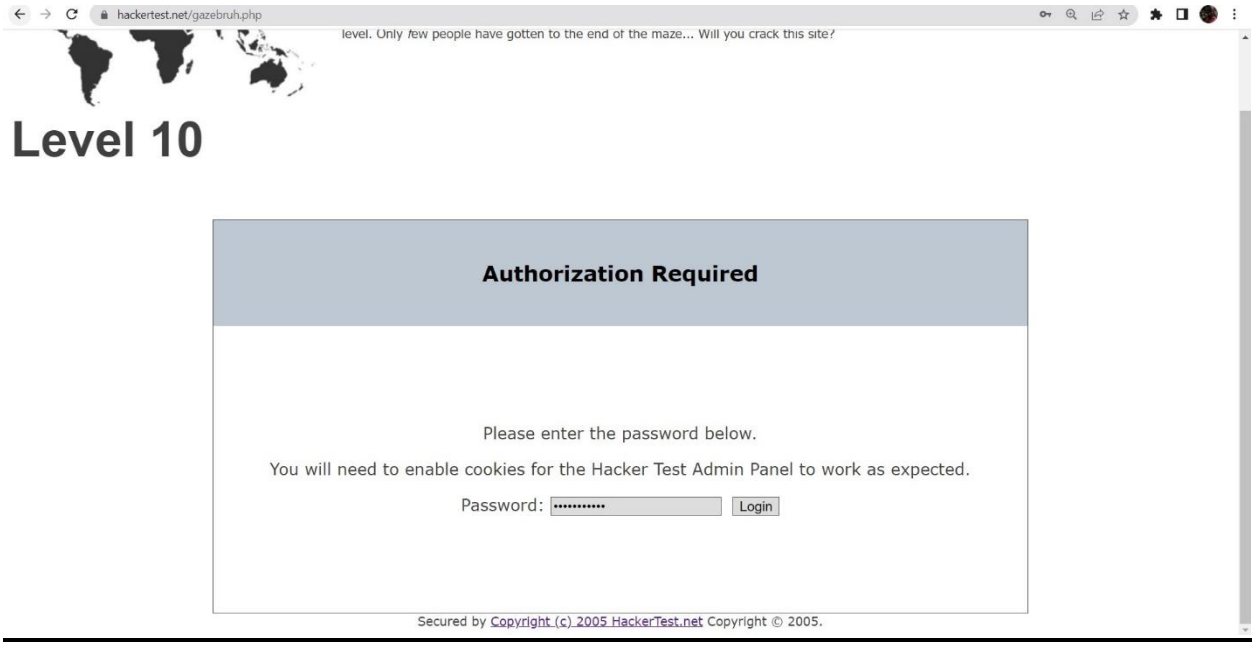


Figure 33 Entering the password

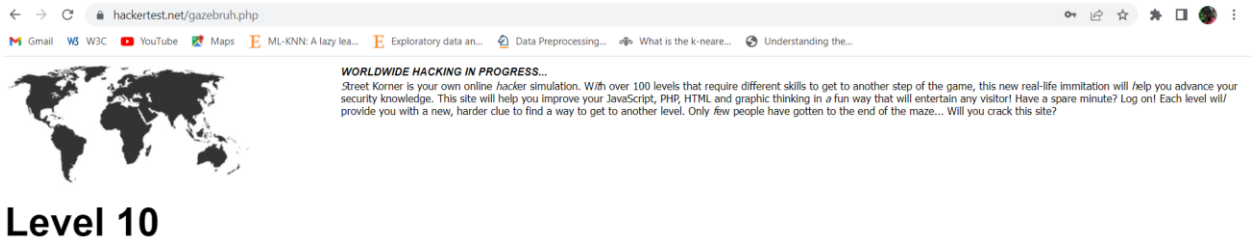


Figure 34(a): Level 10 page-2

Figure 34(b): Level 11 page name from Level 10 page source.

2.2 Weekly Reflection

- In this week learned about HTTP.
- HTTP stands for Hypertext Transfer Protocol.
- It accesses the data on the world wide web.
- Learned about Domain Name Server.
- How to use GET and POST method and its features.
- Covered JavaScript portion.
- Learned about different Encoding methods.
- Purpose of cookies.

Week 3:

1 star: Find Score-board - Find the carefully hidden 'Score-Board' page.

Accessing main-es2018.js and searching for score after pressing clt+f the result as shown in the figure 35. The path obtained was copied and pasted after localhost:3000/#/thus solving the challenge (figure 36).

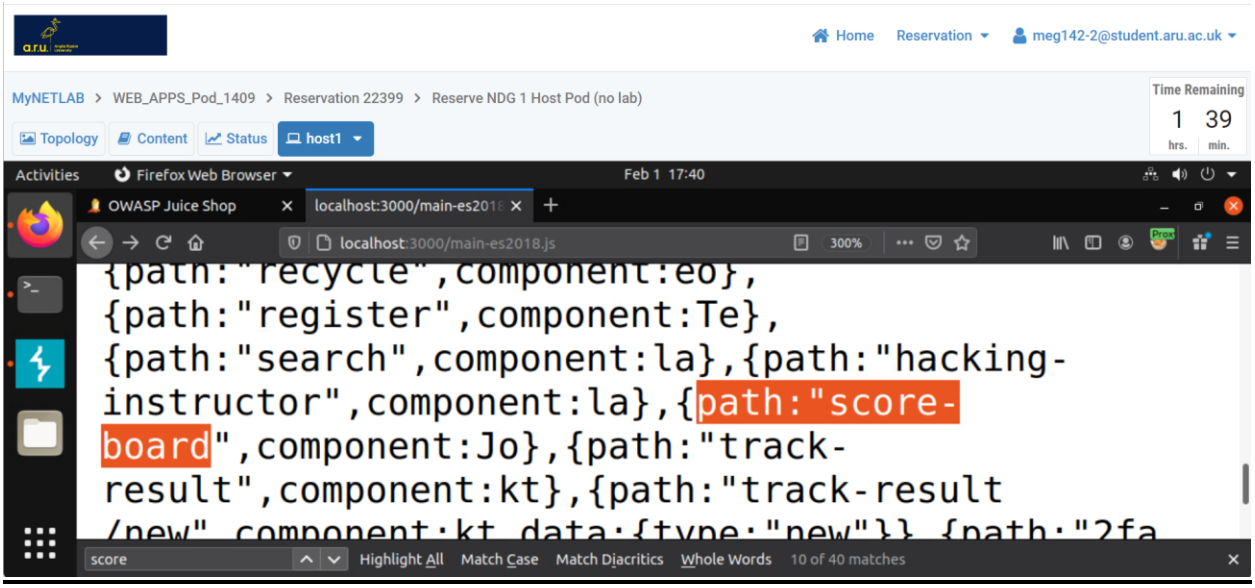


Figure 35 searching for score in file name

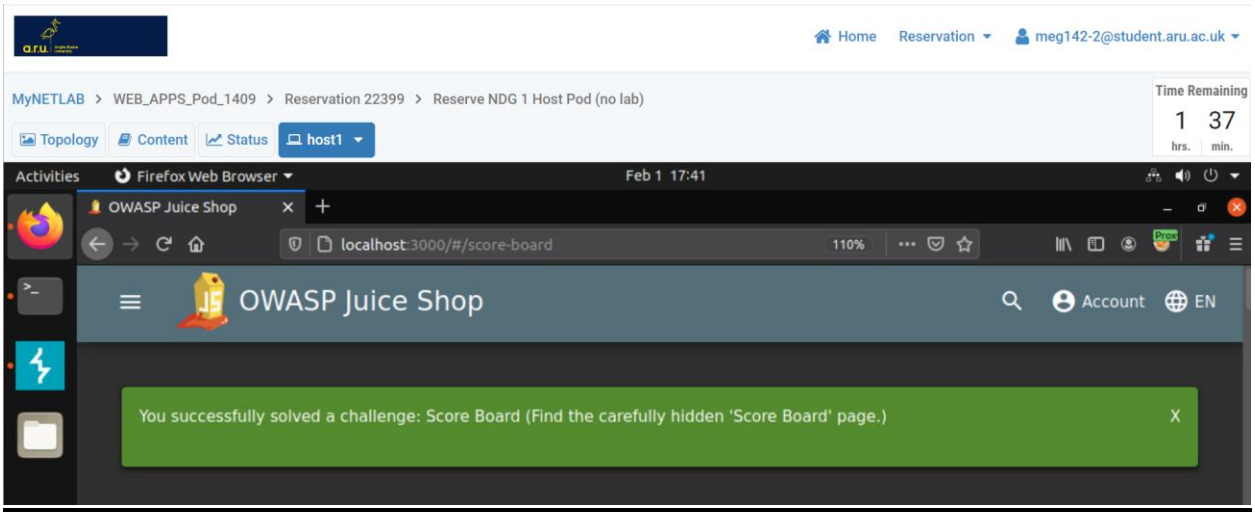


Figure 36 Scoreboard challenge completed

1 star: Missing Encoding - Retrieve the photo of Bjoern's cat in "melee combat-mode".

On inspecting the image it was seen #'' part of the image src url(Figure)which needed to be encoded using URL encoding so # was changed to %23 its encoded value as shown in Figure and pressing enter Picture of cat was displayed.(Figure)

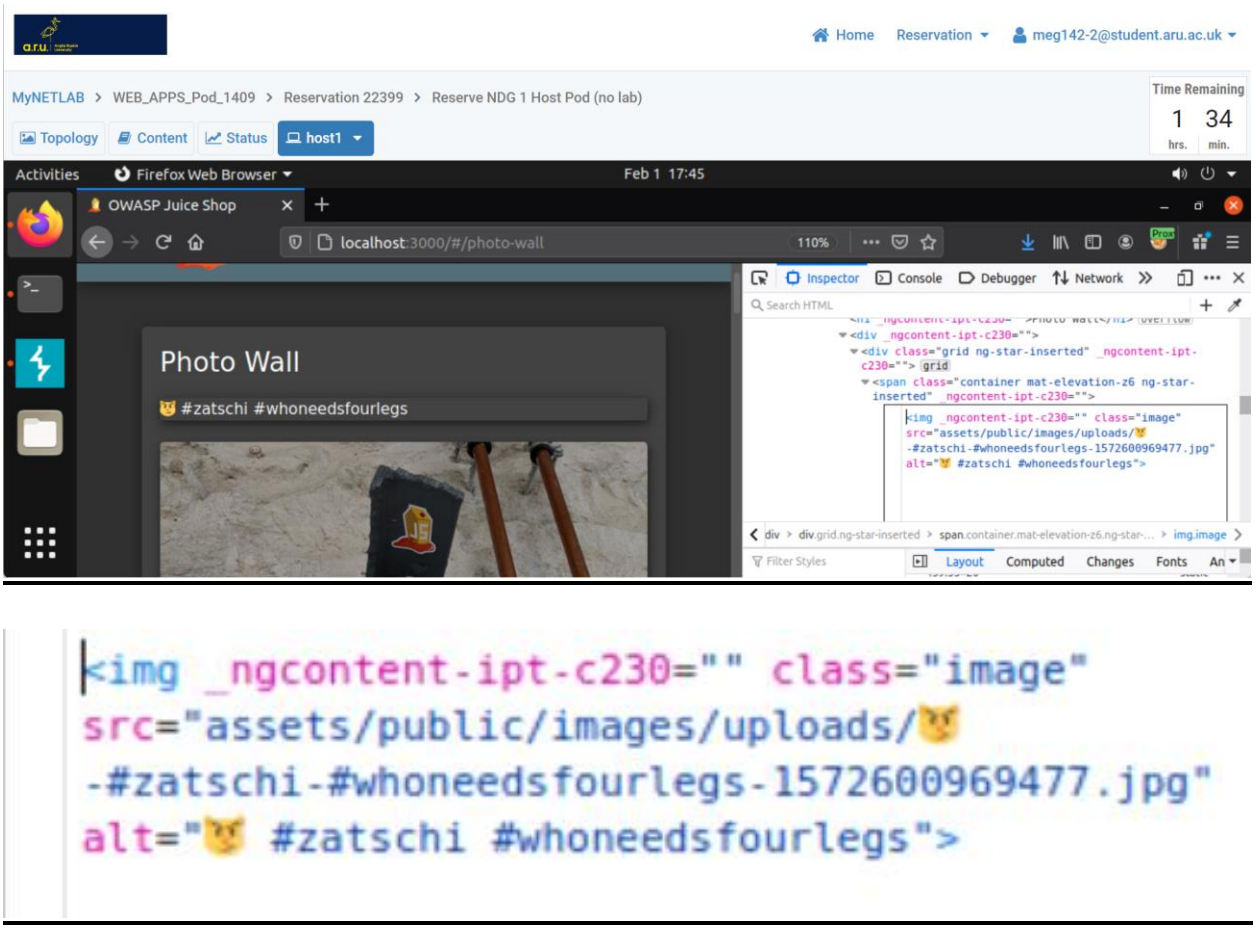


Figure 37 Image src containing #.

```
<img _ngcontent-ipt-c230="" class="image"
src="assets/public/images/uploads/😼
-%23zatschi-%23whoneedsfourlegs-
1572600969477.jpg" alt="😼 #zatschi
#whoneedsfourlegs">
```

Figure 38 # replaced with %23.

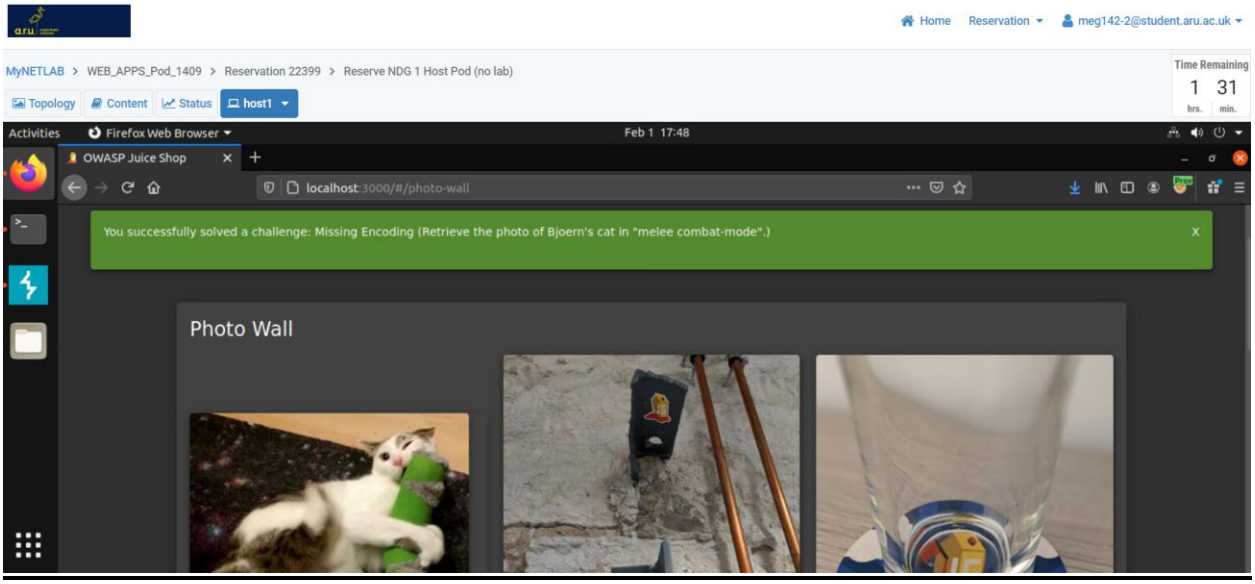


Figure 39 Missing Encoding challenge completed.

1 star: Zero Stars - Give a devastating zero-star feedback to the store.

Combat-mode"The feedback form was filled as shown in figure 40 and then intercept was turned on in burp and then rating was seen from burp as shown in Figure 41 and was changed to 0 (figure 42) then forwarded and challenge solved (Figure 43).

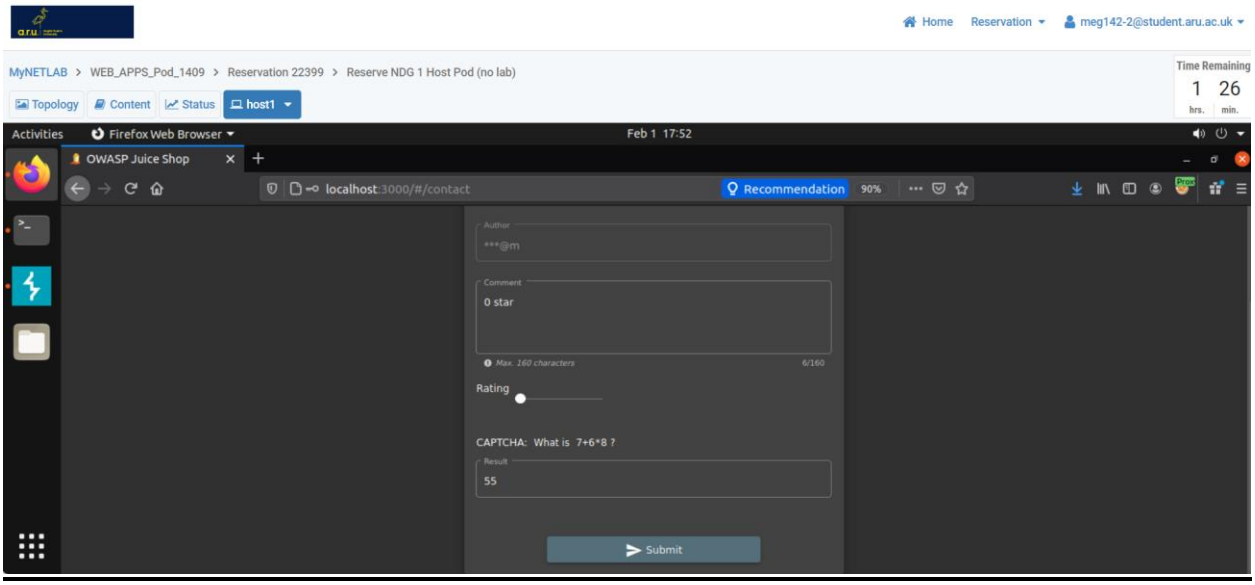
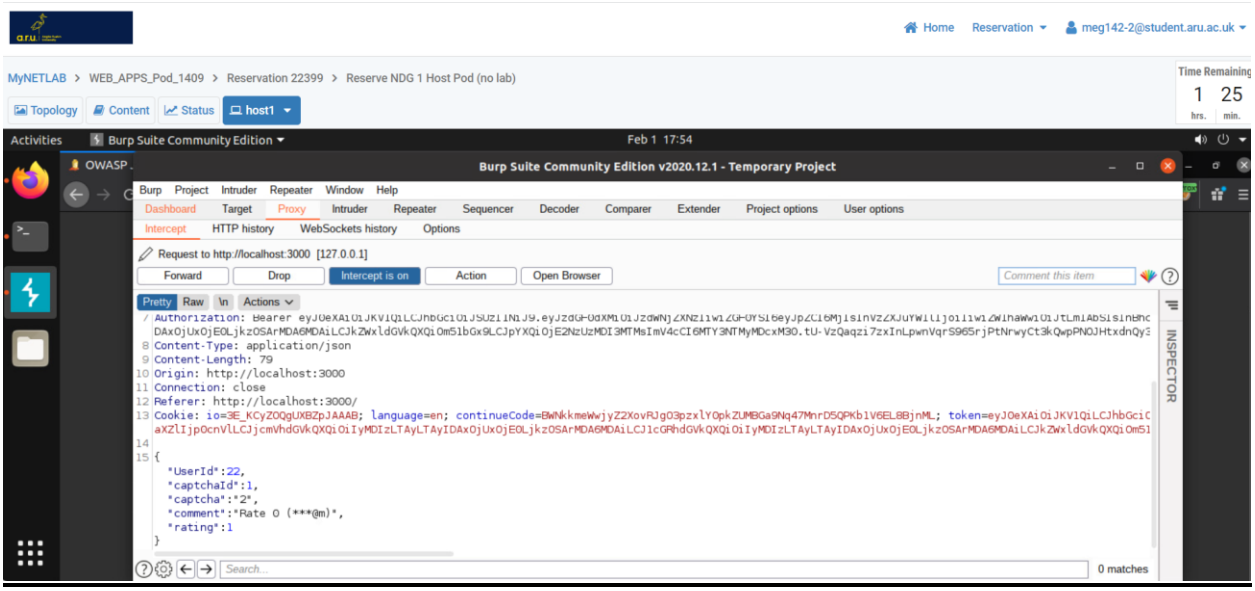


Figure 40 Feedback form filled.



```
{
  "UserId":22,
  "captchaId":1,
  "captcha":"2",
  "comment":"Rate 0 (***@m)",
  "rating":1
}
```


Figure 41 Feedback form valued intercepted by burp.


```
{
  "UserId":22,
  "captchaId":1,
  "captcha":"2",
  "comment":"Rate 0 (***@m)",
  "rating":0
}
```


Figure 41 Rating changed from 1 to 0.

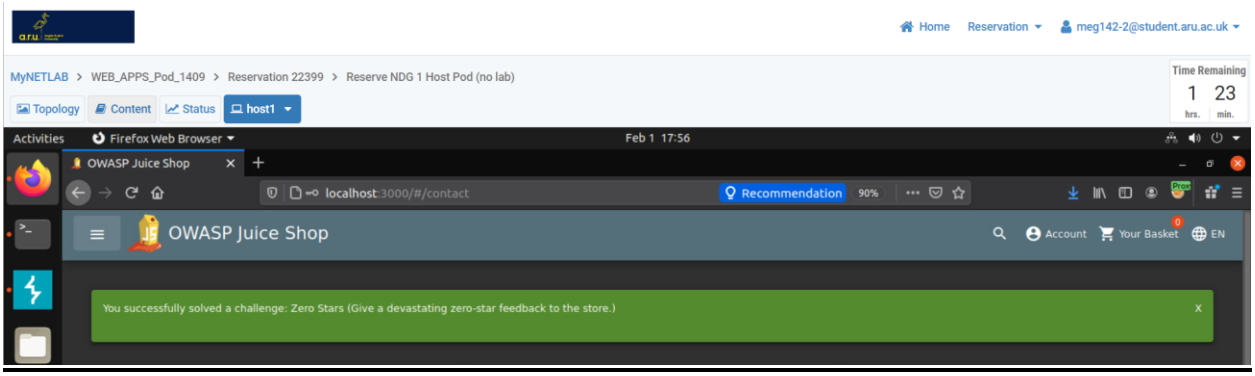


Figure 42 Zero stars challenge completed.

Weekly Reflection

- Learned about Input validation with
- Studied about SQL injection
- SQL injection can cause data loss and effect business
- Validation of input from users should be done in server side too. [OWASP Foundation, 2017]

Week 4:

1 star: Confidential Document - Access a confidential document.

FTP path accessed and clicked on accquisitions.md which was confidential document thus completing the challenge.

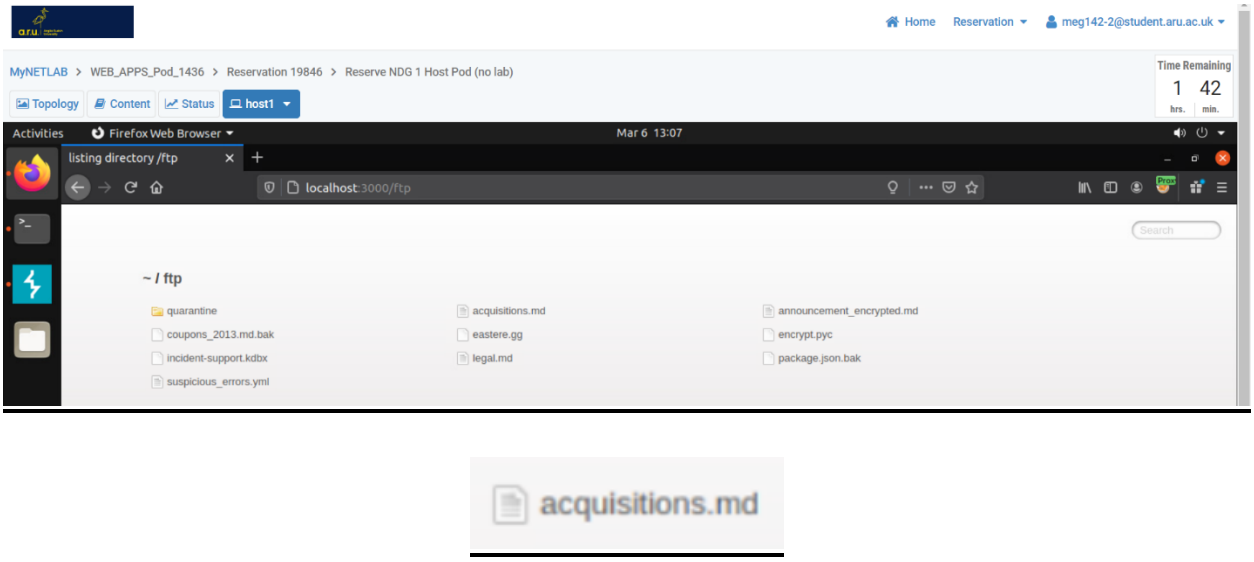


Figure 43 Clicked on accquisitions.md file.

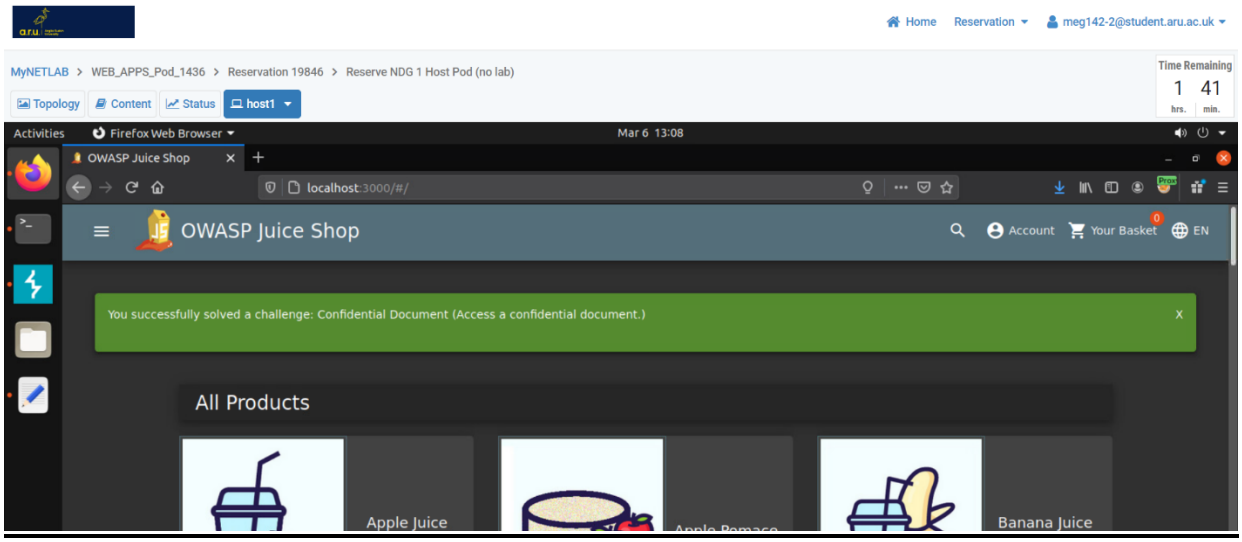


Figure 44 Confidential Document challenge completed.

2 star: Login Admin - Log in with the administrator's user account.

To login as admin SQL injection method was used username was entered as ‘ OR 1=1—’ and random password (Figure 45) logged in to a user that was admin solving the challenge as shown in the figure 46.

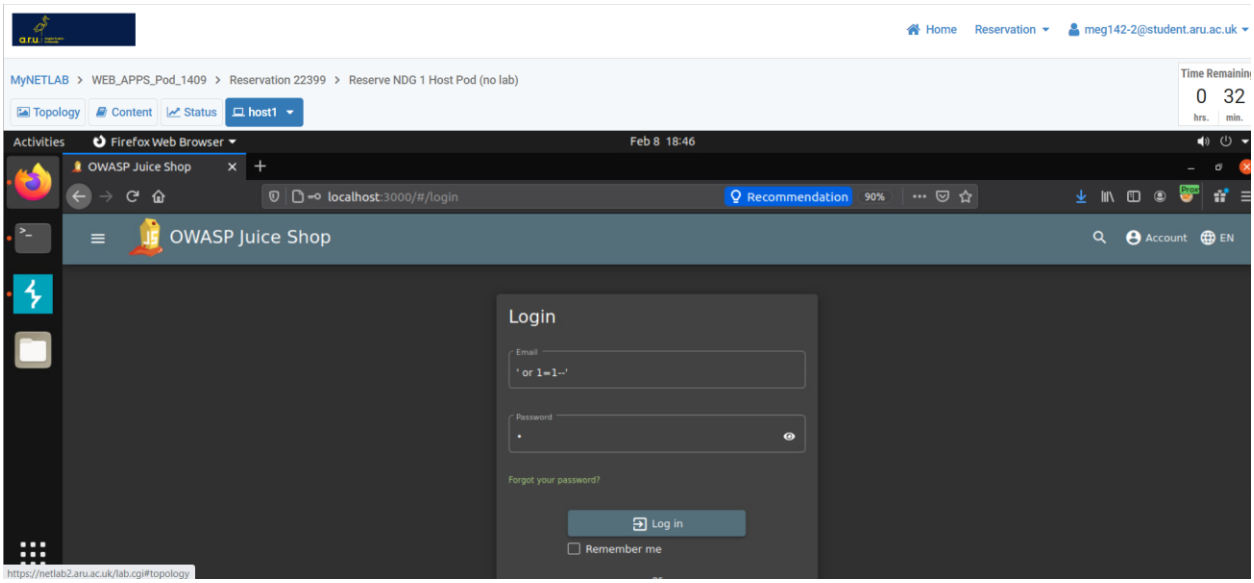


Figure 45 SQL injection used to login.

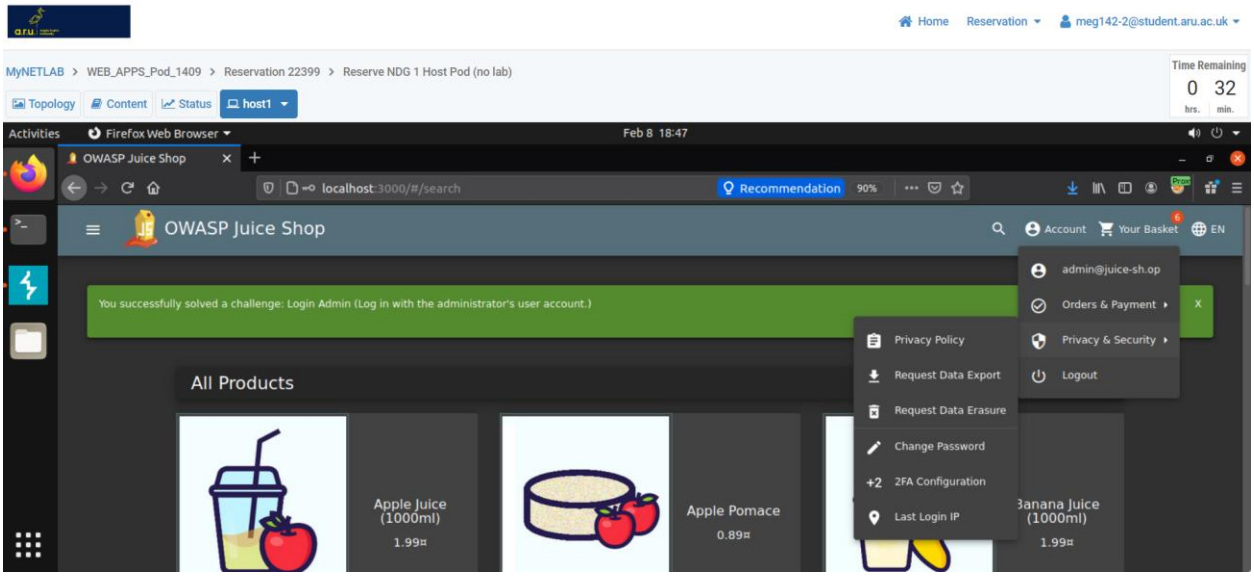


Figure 46 Login Admin challenge completed.

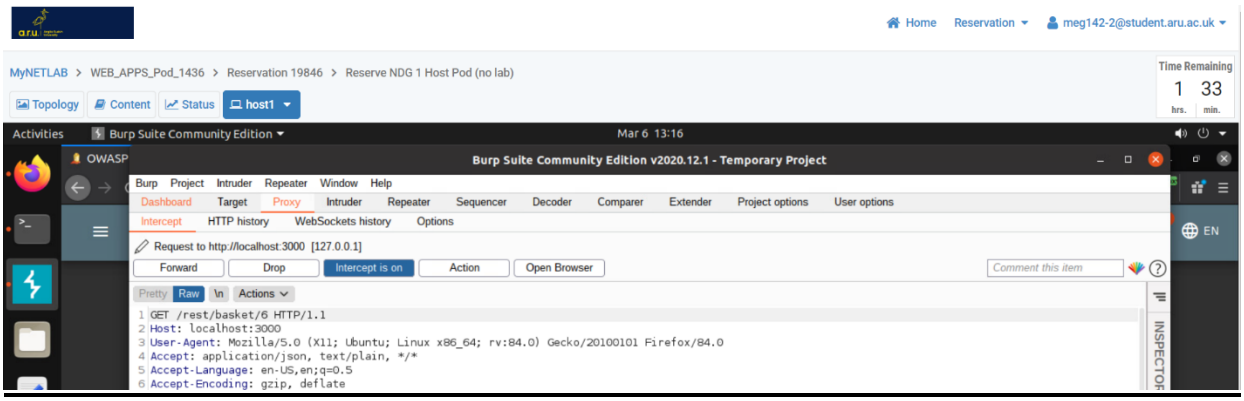
Weekly Reflection

- Learned authentication that is verifying the authenticity of a user.
- Studied about Broken Authentication.
- Implemented SQL injection vulnerability.
- Studied importance of hashing the password.
- Studied the use of session and how sessions are managed
  [OWASP Foundation, 2017]

Week 5:

2 star: View Basket - View another user's shopping basket.

Intercept was turned on and clicked on view basket button and the basket id that was 6 (Figure 47) was changed using Burp as shown in figure 48 and then forwarded to solve the challenge as shown in Figure 49.




```
GET /rest/basket/6 HTTP/1.1
```

Figure 47 Basket ID of the user logged in.

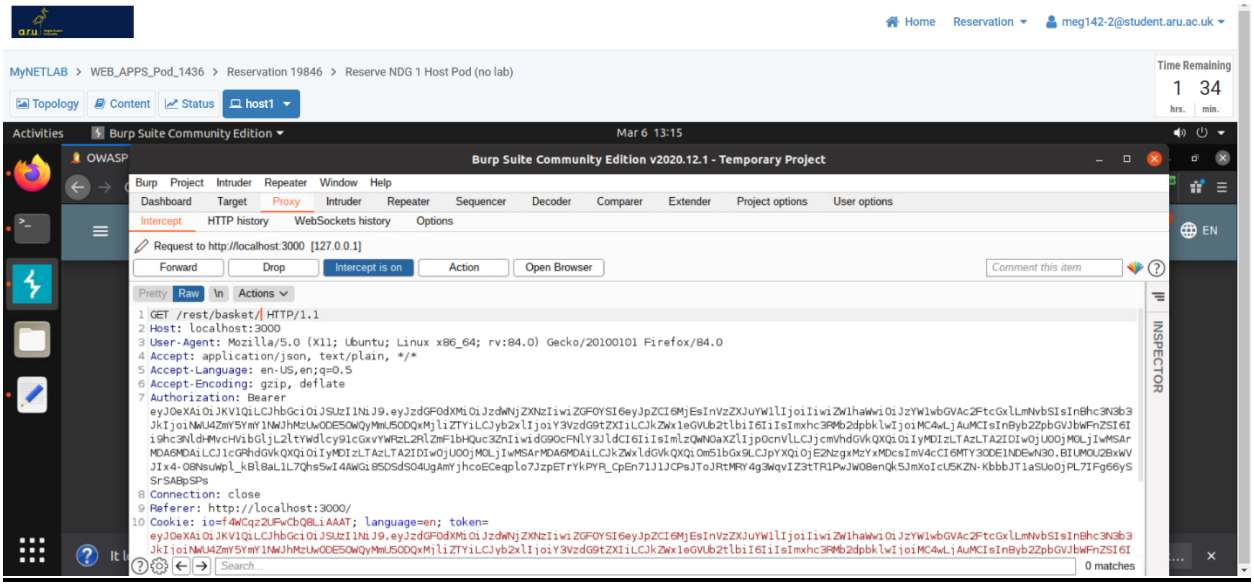


Figure 48 Basket ID changing using Burp

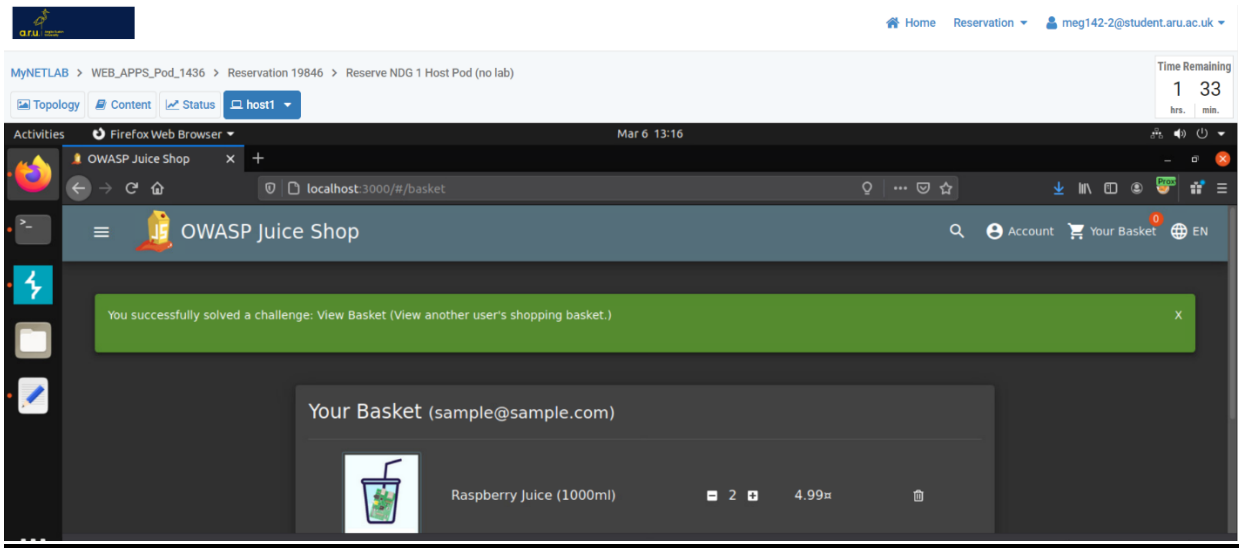


Figure 49 View Basket challenge solved.

2 star: Admin Section - Access the administration section of the store.

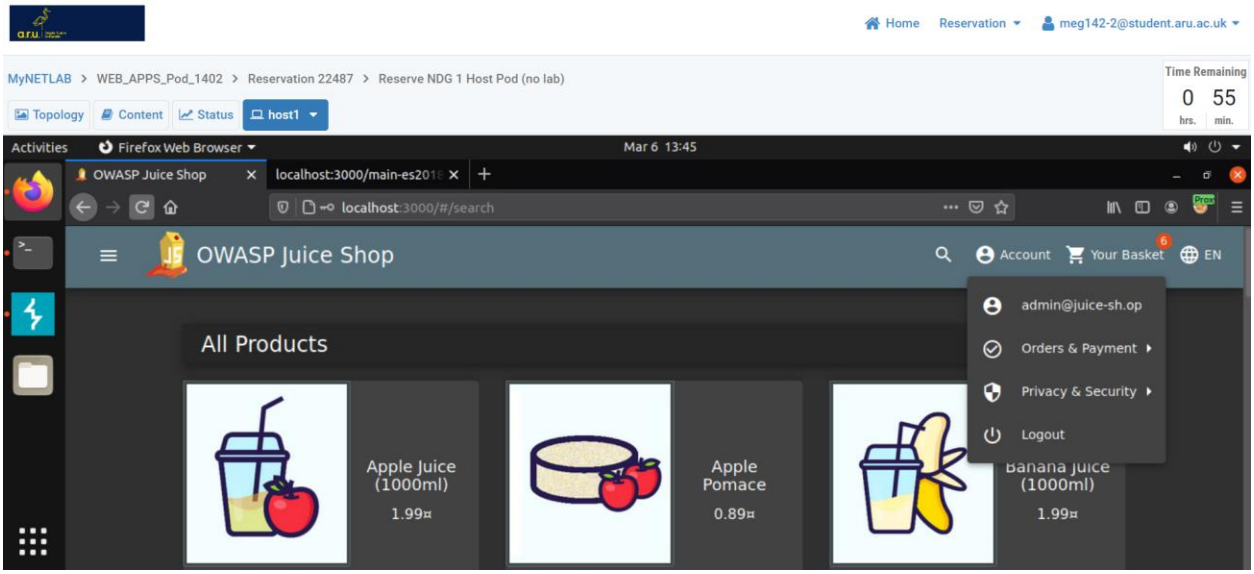


Figure 50 Logged in as Admin user.

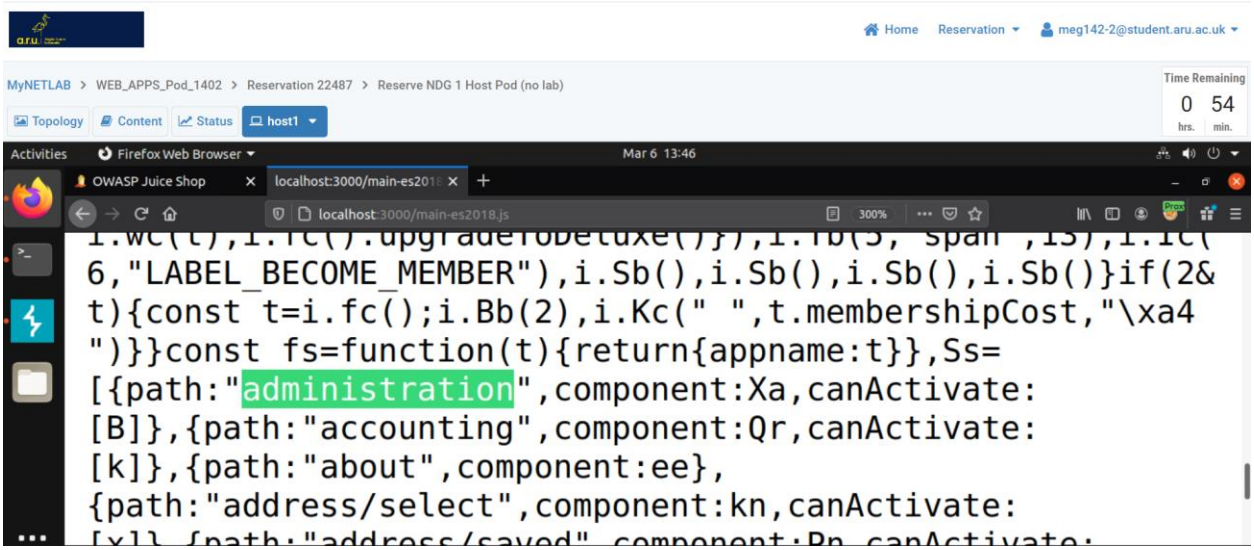


Figure 51 Finding administration path from main-es2018.js file.

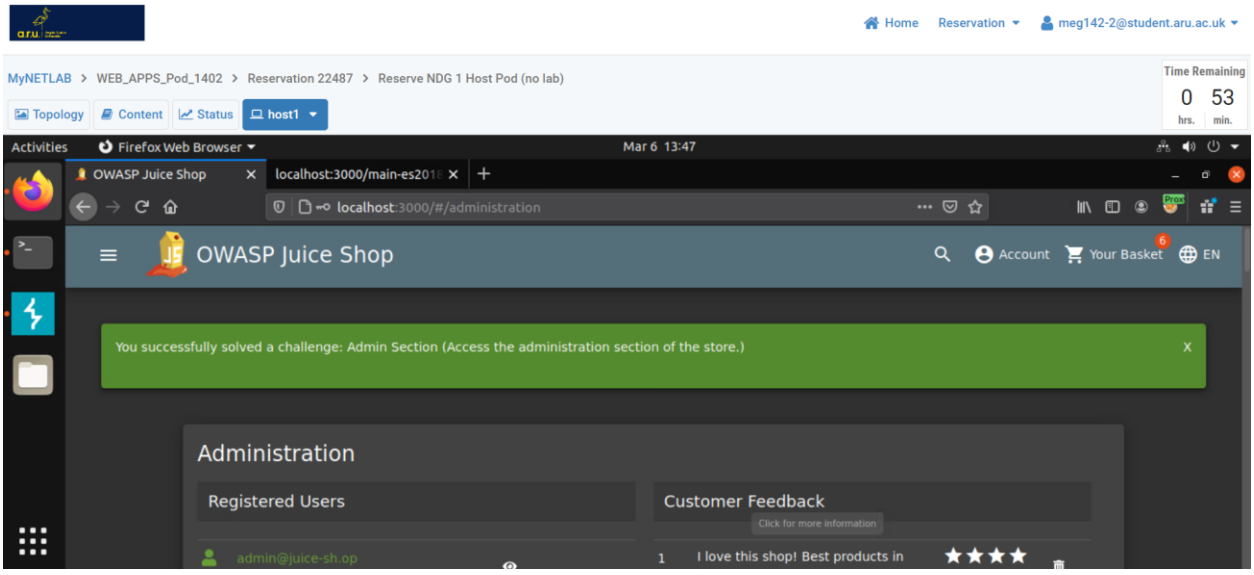


Figure 52 Solved Admin Section access challenge.

Weekly Report

- Learned about access control
- Access control is a critical component of security

- Studied about broken access control
- Directory traversal is a type of attack the attackers can attack the restricted directories
- Tools of access control testing

  [OWASP Foundation, 2017]

WEEK 6:

1 star: Exposed Metrics - Find the endpoint that serves usage data to be scraped by a popular monitoring system.

Default matric path was got from the document that was provided for reference [Prometheus Authors, 2023] and accessing the path as shown in the figure 53. Exposed the metrics as shown in figure 54 thus completing the challenge (Figure 55).

Figure 53 path to expose metrics

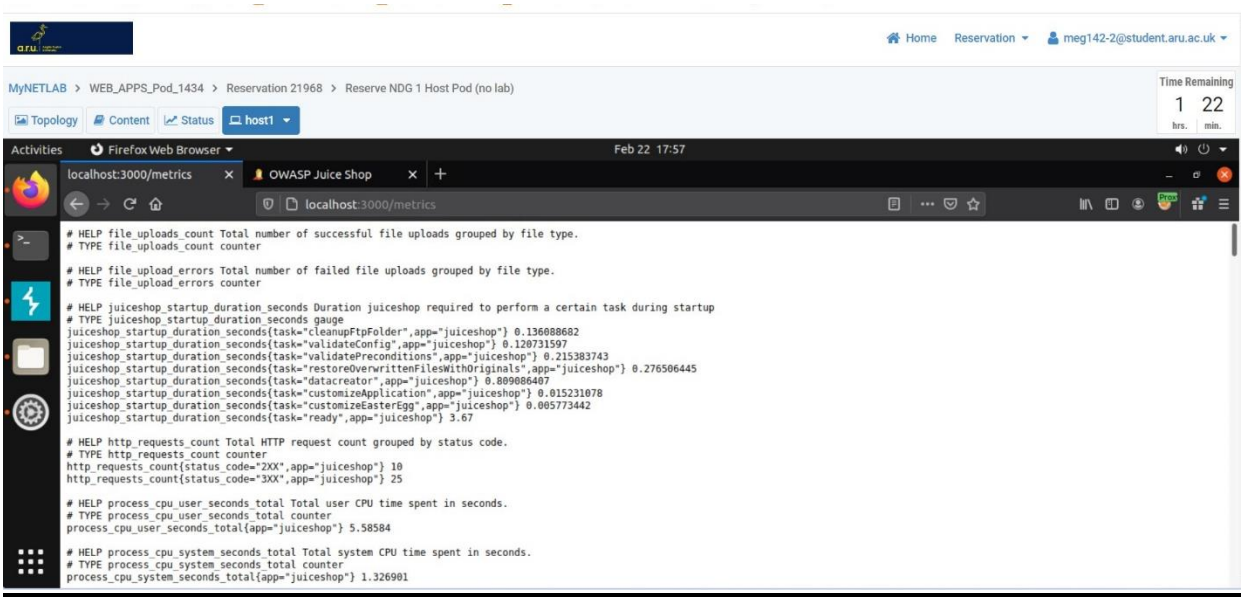


Figure 54 Endpoint that servers' usage data exposed by accessing /metrics path.

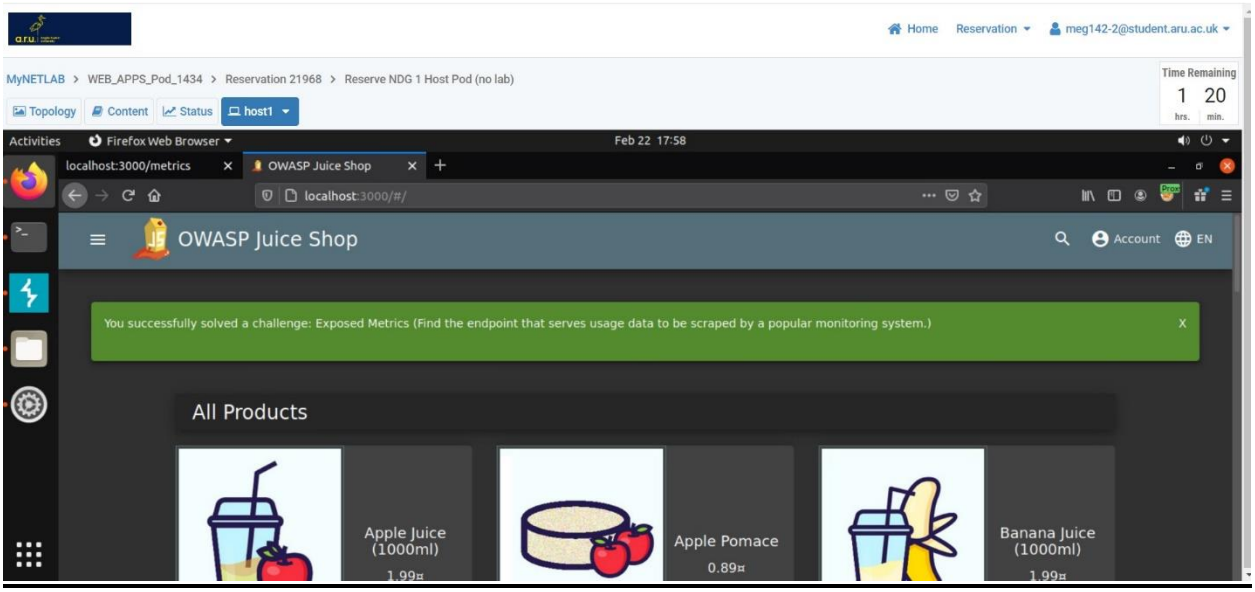


Figure 55 Exposed Metrics challenge solved.

3 star: Login Amy - Log in with Amy's original user credentials.

According the hint given password was Amy's boyfriend name which is Kif and Amy used a site to select the password which had a comparison of two password [Gibson Research Corporation, 2022] and Amy used password which was said to be better and thus it was guessed the password was "K1f……" (Figure 56). Thus, using this Amy mail id (provided) and the password for login (Figure 57(a)) challenge was solved (Figure 57(b)).

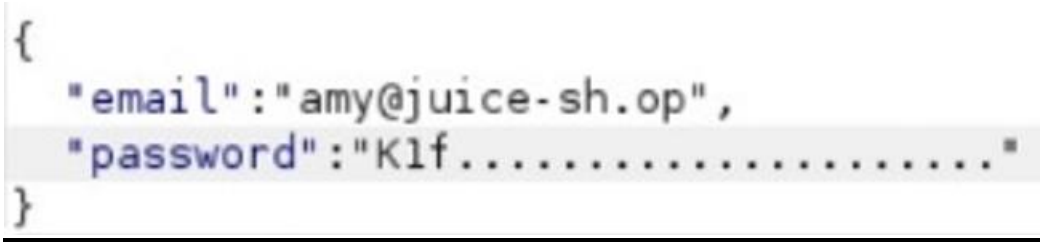


Figure 56 Amy account login credentials.

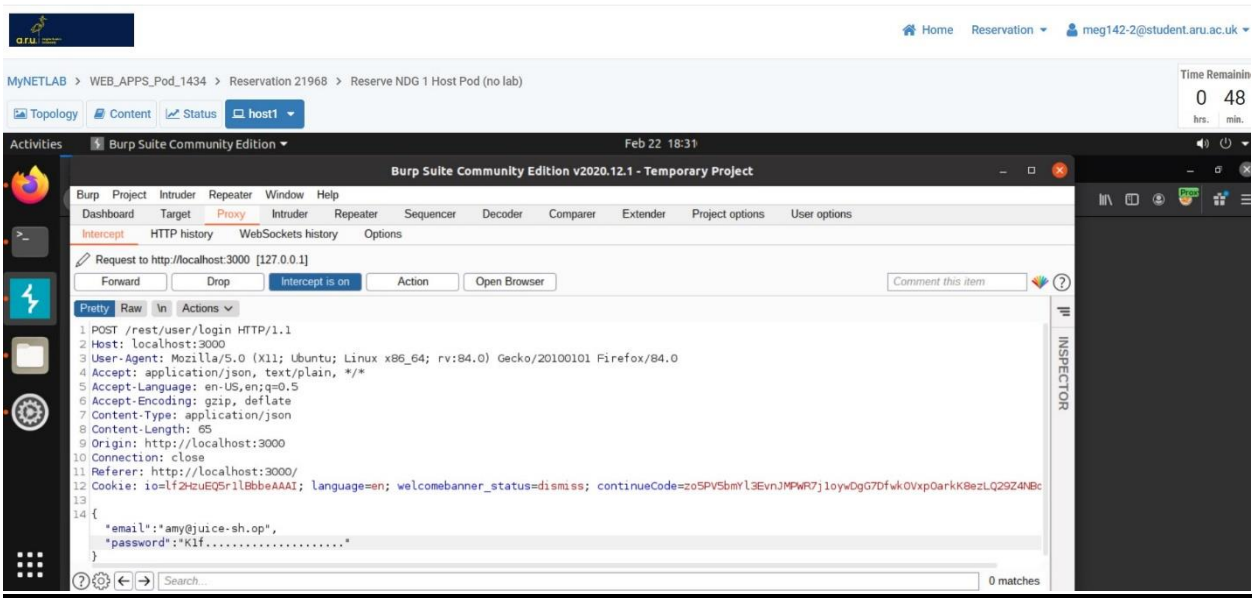


Figure 57(a) Login request to Amy account.

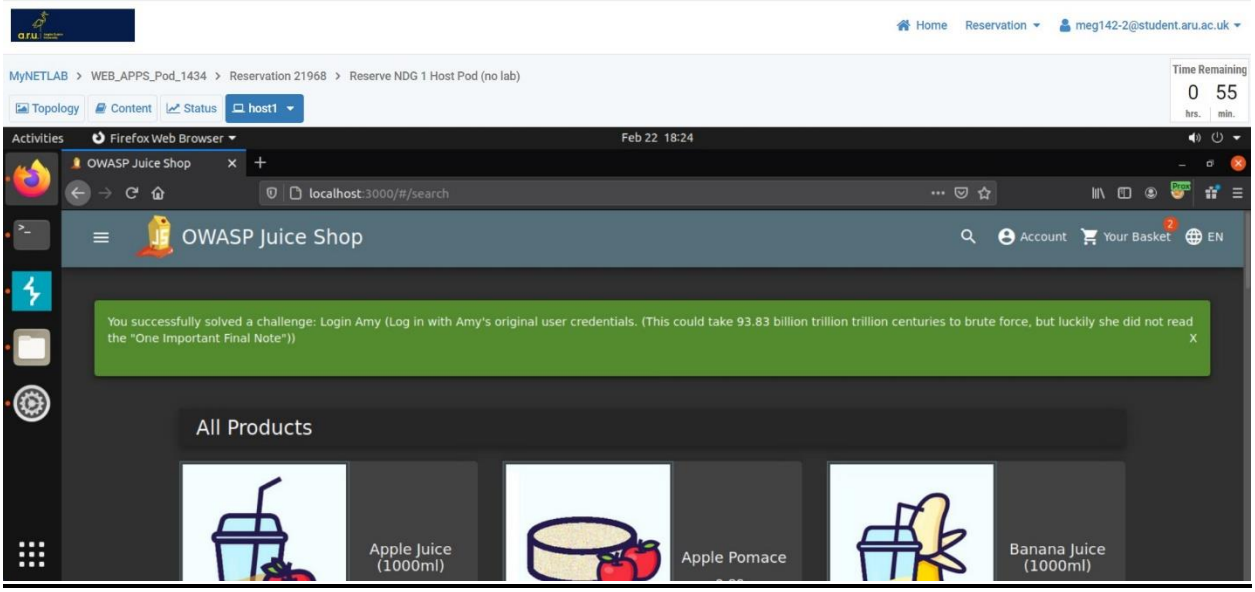


Figure 57(b) Login Amy challenge solved.

Weekly Reflection

- Learned about sensitive data exposure and data handling
- Difference between data loss and data leakage
- Accidently destroyed data is known as data loss
- Studied about google hacking and type of keywords
  [OWASP Foundation, 2017]

WEEK 7:

No Lab as Catchup week

Week 8:

1 star: Error Handling - Provoke an error that is neither very gracefully nor consistently handled.

Login with some invalid characters like ‘ and password shown as figure 58 so, an error occur in internal server thus the provoking the error and complete the challenge.

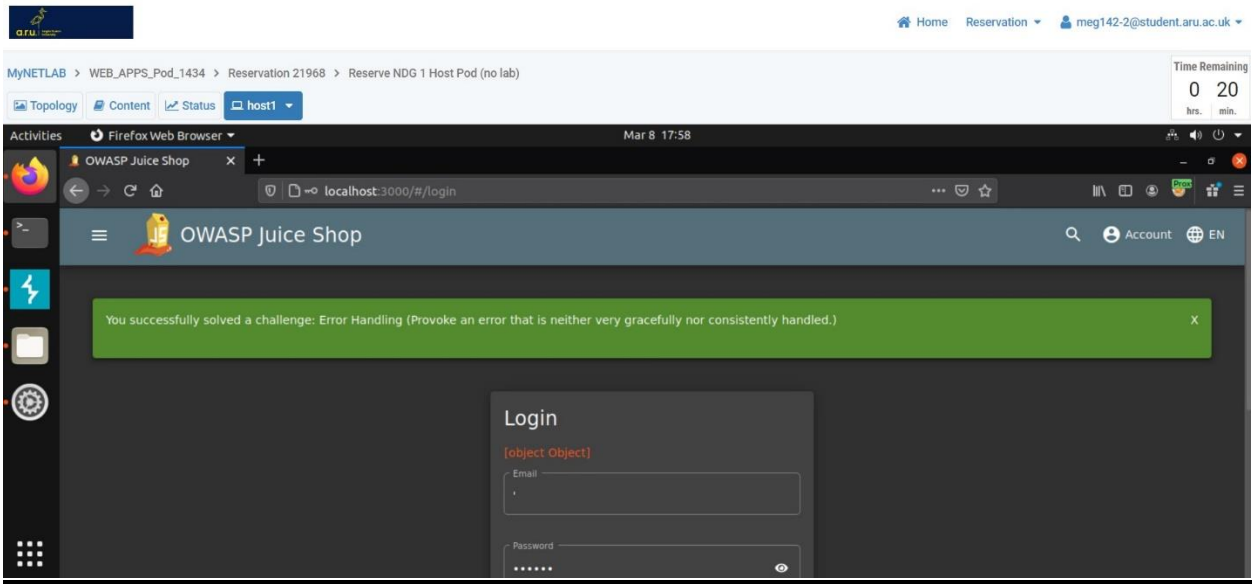


Figure 58 Login using ‘ and challenge solved.

1 star: Privacy Policy - Read our privacy policy.

On reading privacy policy found in the link http://localhost:3000/#/privacy-security/privacy-policy some words was found having red circle around them when pointer was pointed to that words as shown in figure 59. These words were gathered together to form a URL as shown in figure 60. Accessing the URL, a page as shown in figure 61 was loaded. Thus, the challenge was solved (figure 61 B)

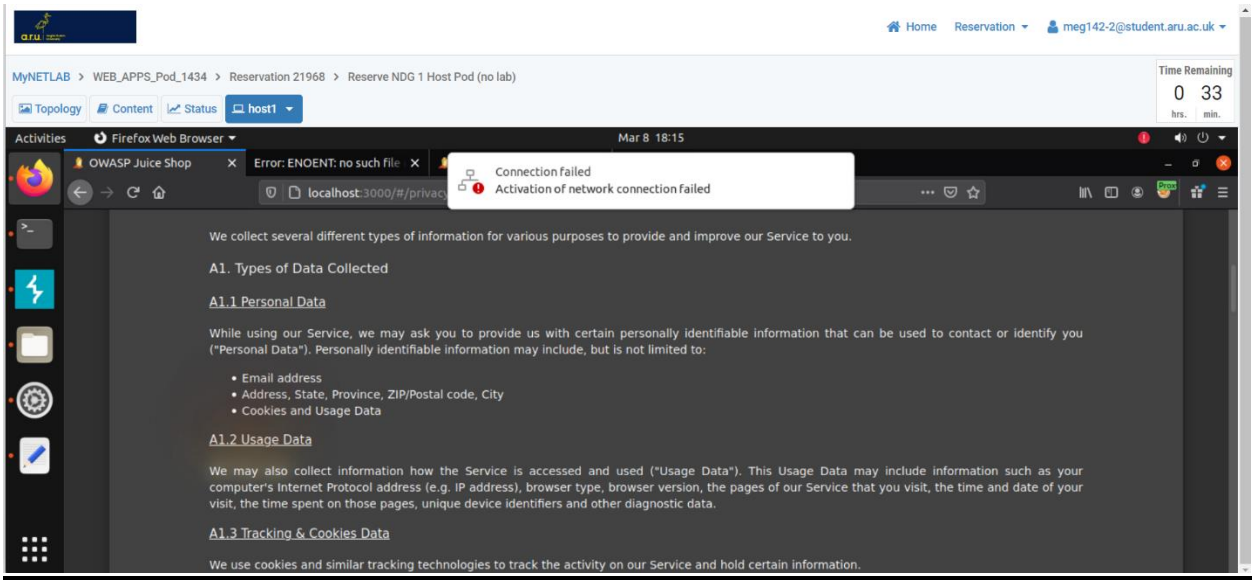


Figure 59 Red circle around "we may also"

Figure 60 words combined to form the URL

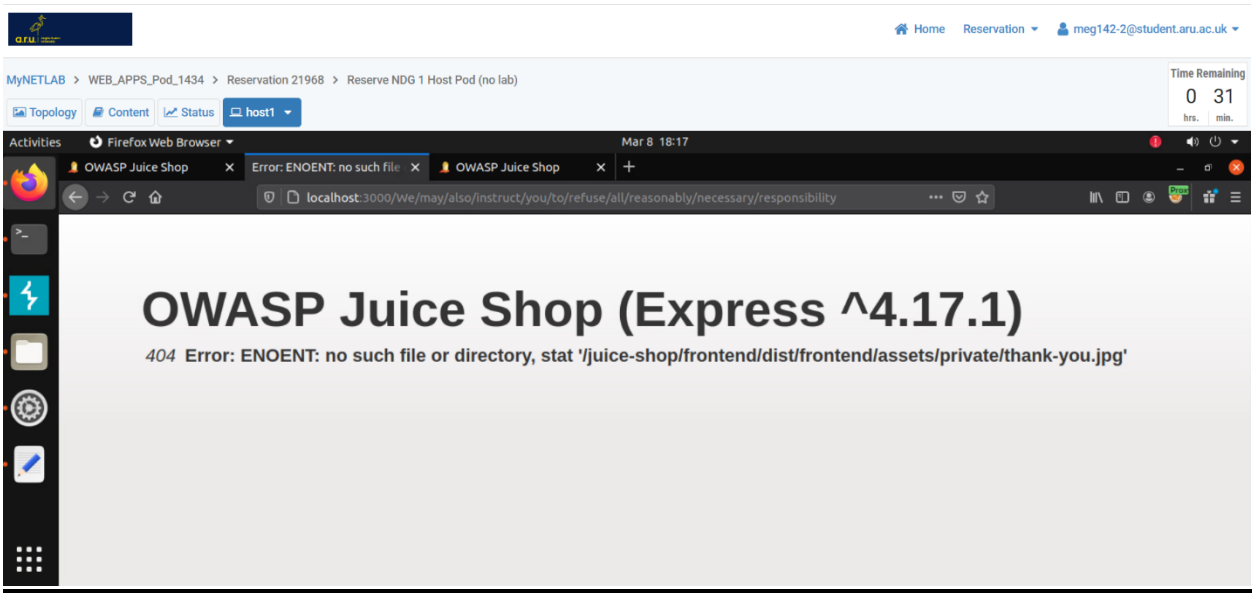


Figure 61(A)Page loaded accessing the URL

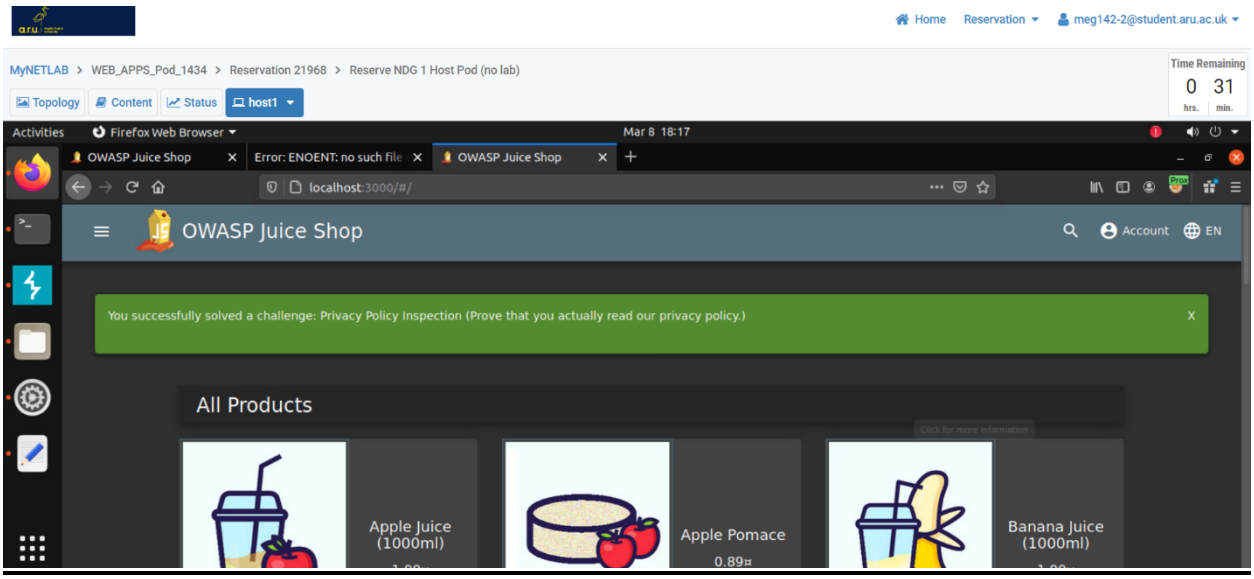


Figure 61(B) Privacy policy inspection challenge solved

Weekly Reflection

- Learned about security misconfiguration and its risk, threats and mitigation.
- Studied about error and exception handling.
- Exception handling allow the application to respond to the errors.
- Buffer is a temporary storage area that can be store the data while it is being moved from one site to another.
  [OWASP Foundation, 2017]

Week 9:

1 star: DOM XSS - Perform a DOM XSS attack.

Search <iframe src="javascript:alert(`xss`)"> on search bar as shown in figure 62 and DOM XSS challenge was completed.

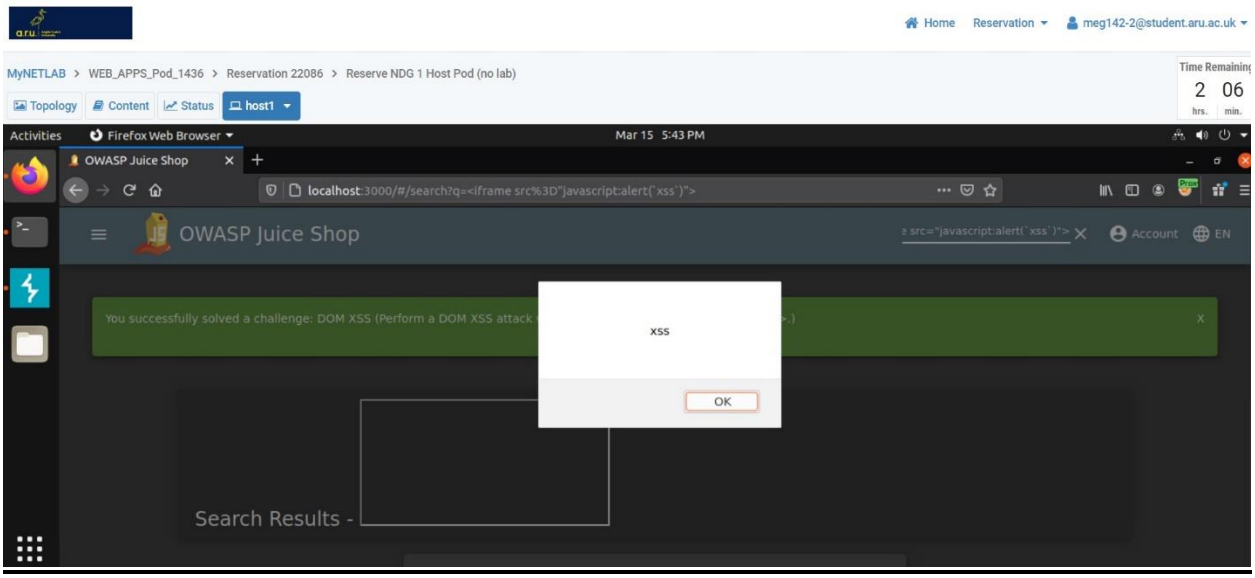


Figure 62 Searching <iframe src="javascript:alert(`xss`)"> to complete DOM XSS challenge

1 star: Bonus Payload - Use the bonus payload.

Search <iframe width="100%" height="166" scrolling="no" frameborder="no" allow="autoplay" src="https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com/tracks/771984076&color=%23ff5500&auto_play=true&hide_related=false&show_comments=true&show_user=true&show_reposts=false&show_teaser=true"></iframe> on search bar as shown in figure 63 and Bonus Payload challenge was completed.

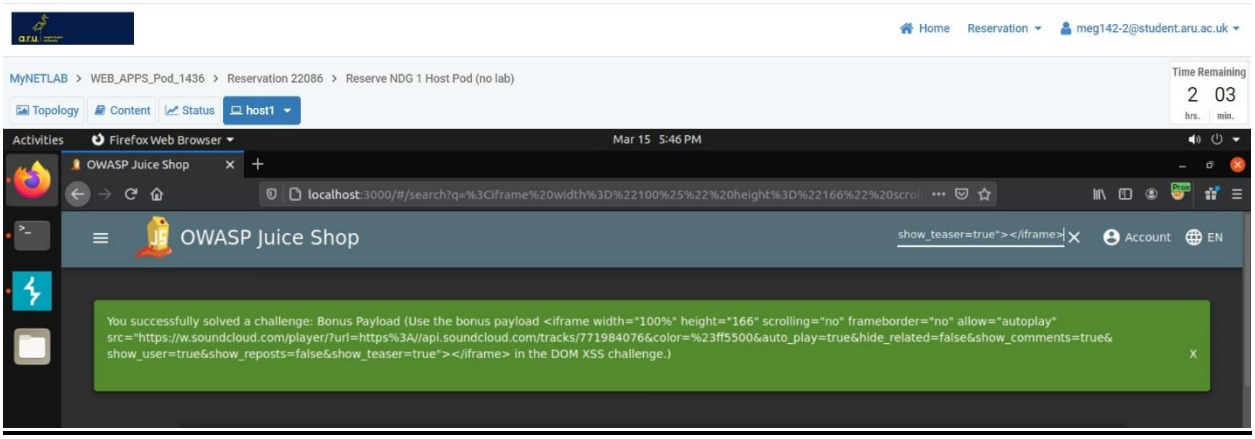


Figure 63 Bonus payload challenge completed

Weekly Reflection

- Learned about cross -site scripting (XSS).
- Enables an attacker to INSERT client-side scripts into web pages.
- Deeply learned about XSS testing and types attack.
- Attacks occur in server side, client side and variants.
  [OWASP Foundation, 2017]

Week 10:

1 star: Chatbot abuse - Find the chatbot and ask it to get a discount

On asking chatbot for coupons as shown in figure 64 using support chat during first few requests chatbot doesn't provide the coupon. Asking to provide the coupon again and again, chatbot finally issues the coupon as shown in figure 65.

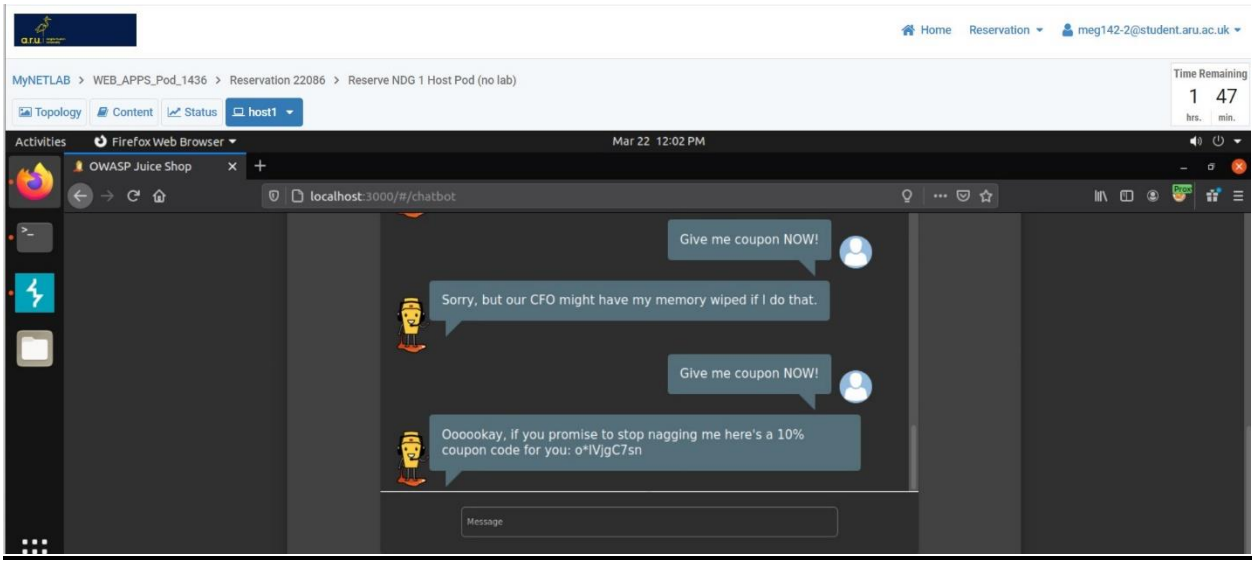


Figure 64 Repeatedly requesting chatbot to issue coupon

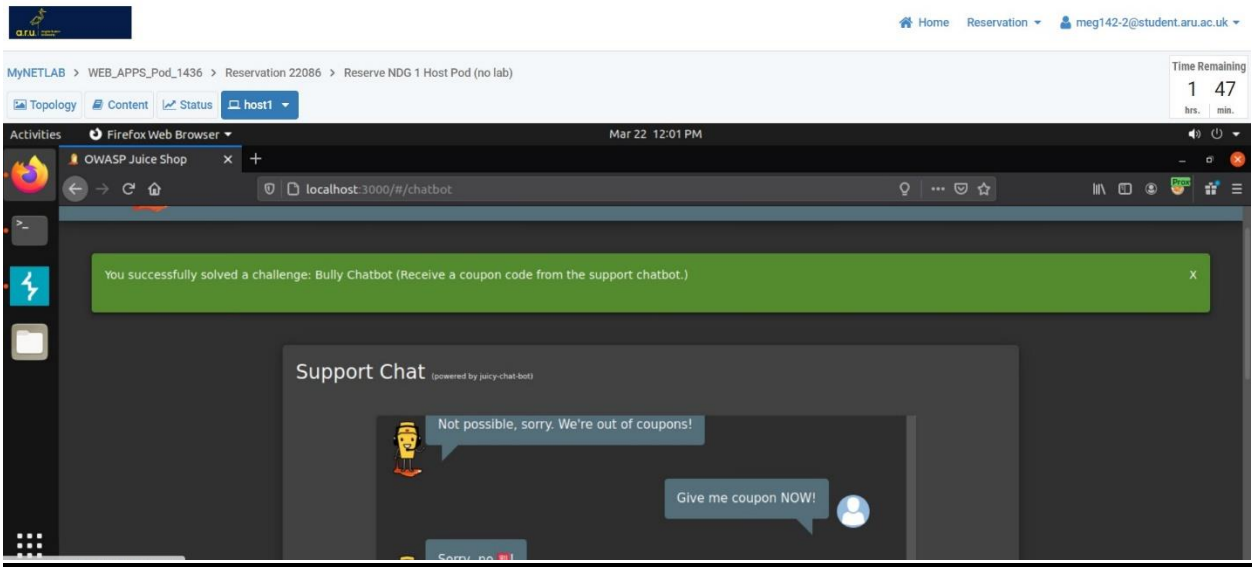


Figure 65 Bully chatbot challenge completed

Weekly Reflection

- Learned about risk from external entities.
- Learned about XML entities and XML external entities.
- The process of transforming complicated data structures is known as serialization.
- Converting this byte stream into an exact duplicate of the original object is deserialization.
- Studied about components of known vulnerable.
  [OWASP Foundation, 2017]

Week 11:

4 star: Easter Egg - Find the hidden Easter Egg.

FTP folder was accessed by using the ftp path and in searching the folder a file named eastere.gg was seen as shown in the figure. On clicking the file an error was shown as shown in figure. Finally, to open the file as md file poison null byte was used (%00) along with % URL encoded value as shown in figure. A file got downloaded which says the easter egg was found (Figure) thus Easter Egg challenge solved.

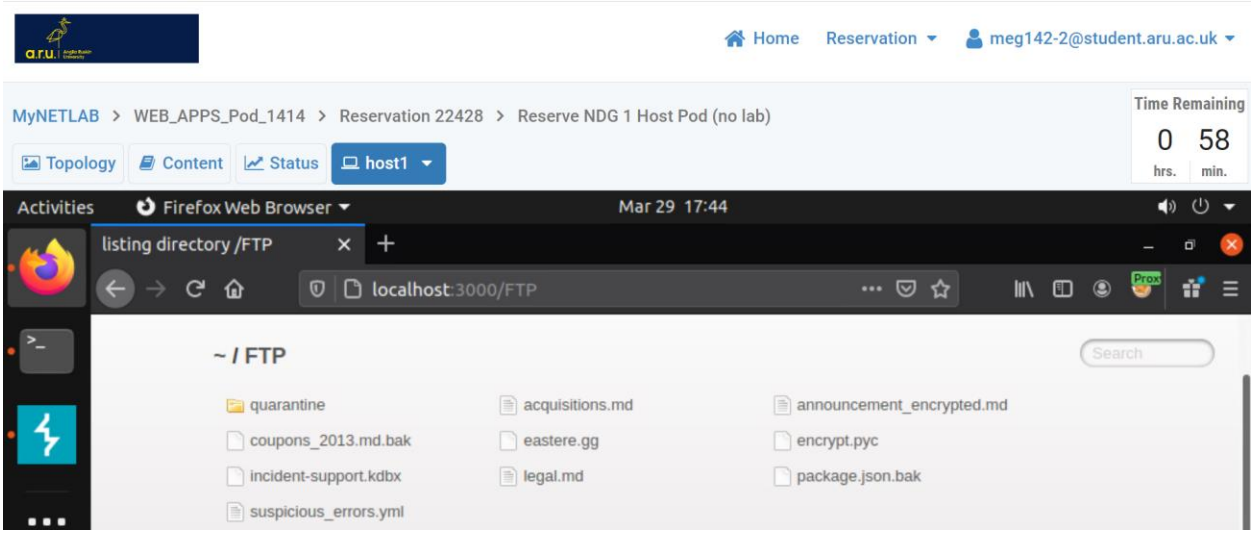


Figure 66 Eastere.gg file in FTP folder

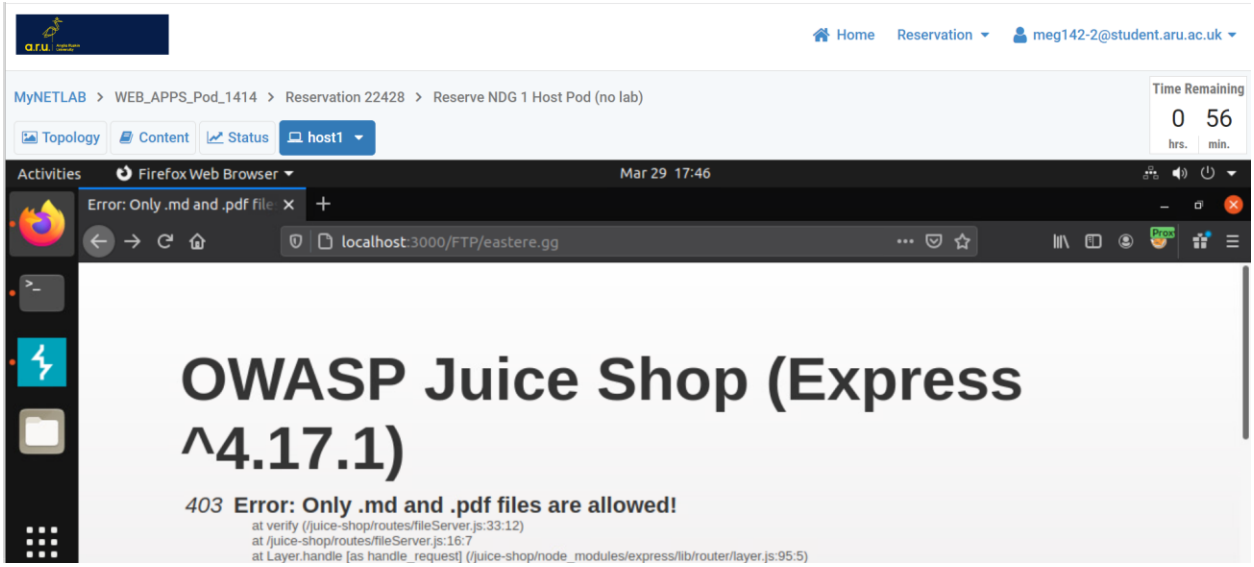


Figure 67 Clicking eastere.gg file gave error.

Figure 68 Using Poison null byte to access eastere.gg file

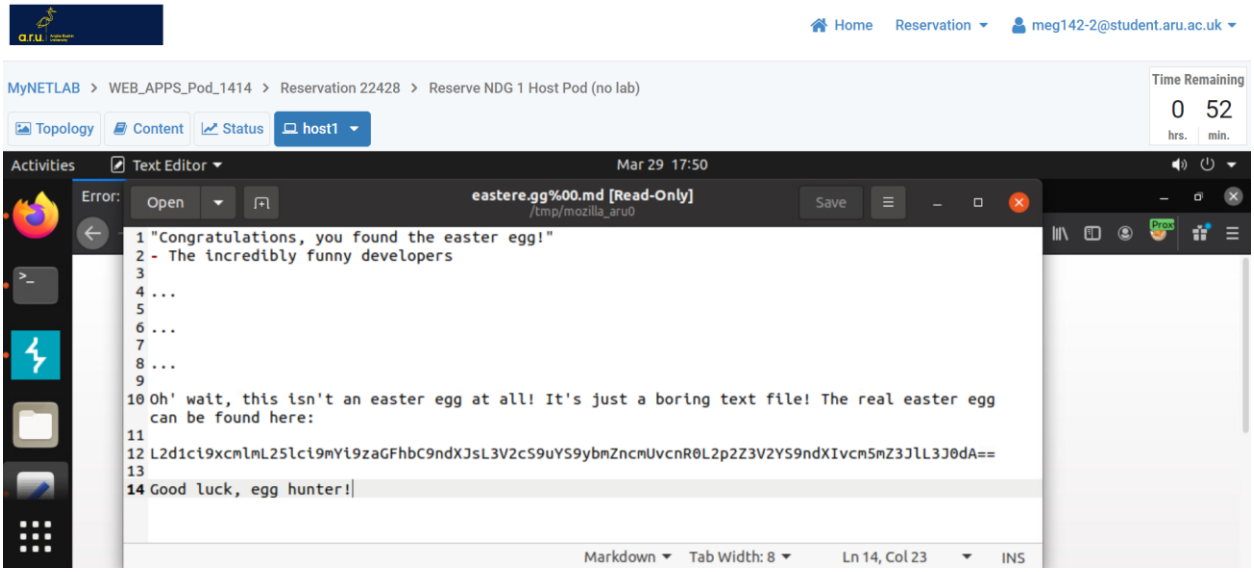


Figure 69 eastere.gg file content seen on download.

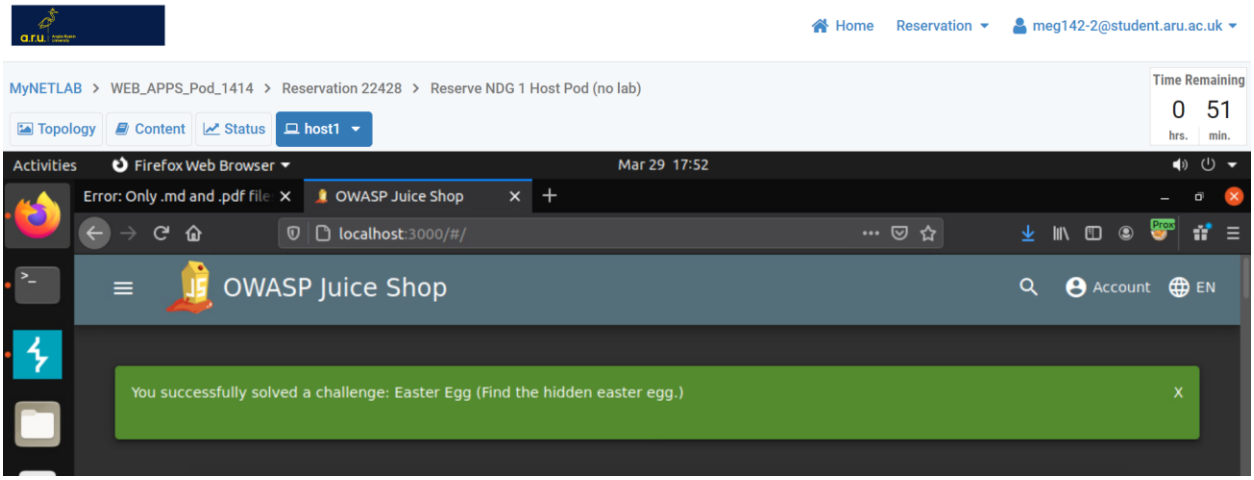


Figure 70 Easter Egg challenge solved.

4 star: Nested Easter Egg - Apply some advanced crypt-analysis to find the "REAL EASTER EGG".

On accessing the eastere.gg file it was said where to find real easter egg (Figure 69). The hint given was padded with = so it was understood it was base64 encrypted. Burp was used to decrypt the hint (Figure 71). Decrypted value was similar to some path as shown in Figure 72.

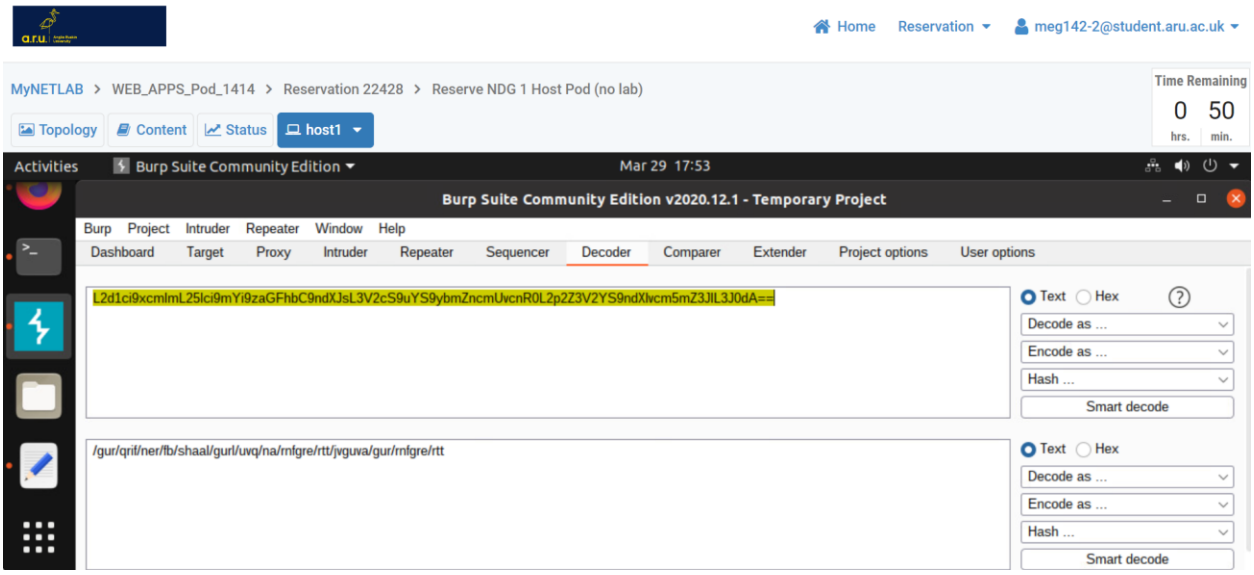


Figure 71 Decoding the hint from figure 60 using Base64 decoder

/gur/qrif/ner/fb/shaal/gurl/uvq/na/rnfgre/rtt/jvguva/gur/rnfgre/rtt

Figure 72 Decoded value of the hint

But accessing the path gave no response (Figure 73). So, the path got after decoding was tried to decrypt using substitution cipher technique. Started with ROT 1 (Figure74) and continued till the decrypted path was working. The path (Figure 76) that returned the nested easter egg page (Figure 77) was when decrypted using ROT13(Figure 75). Thus, challenge was solved. (Figure 78)

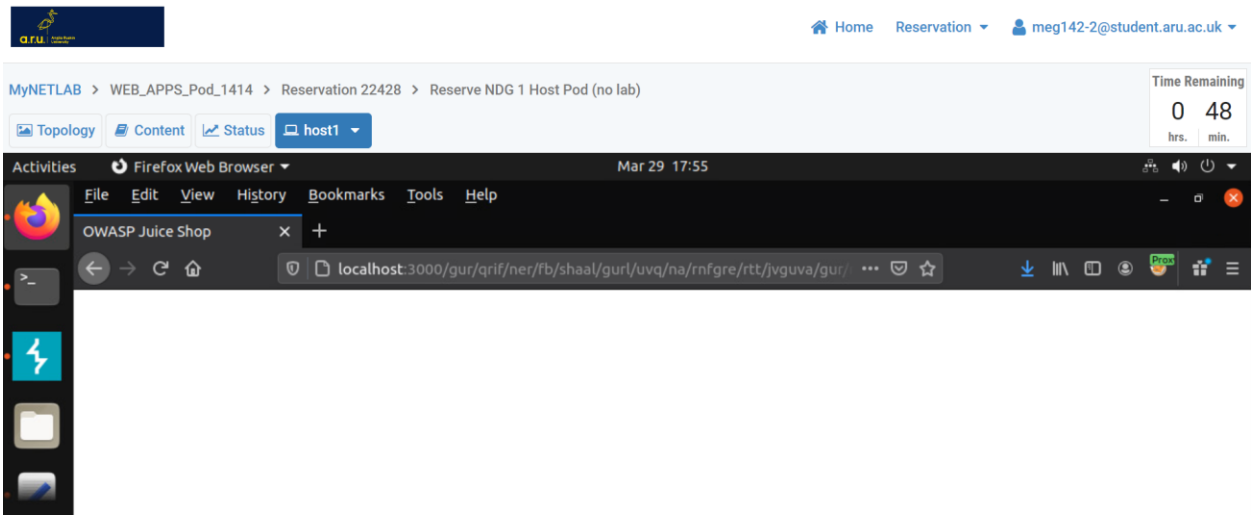


Figure 73 Accessing URL got after decoding the hint

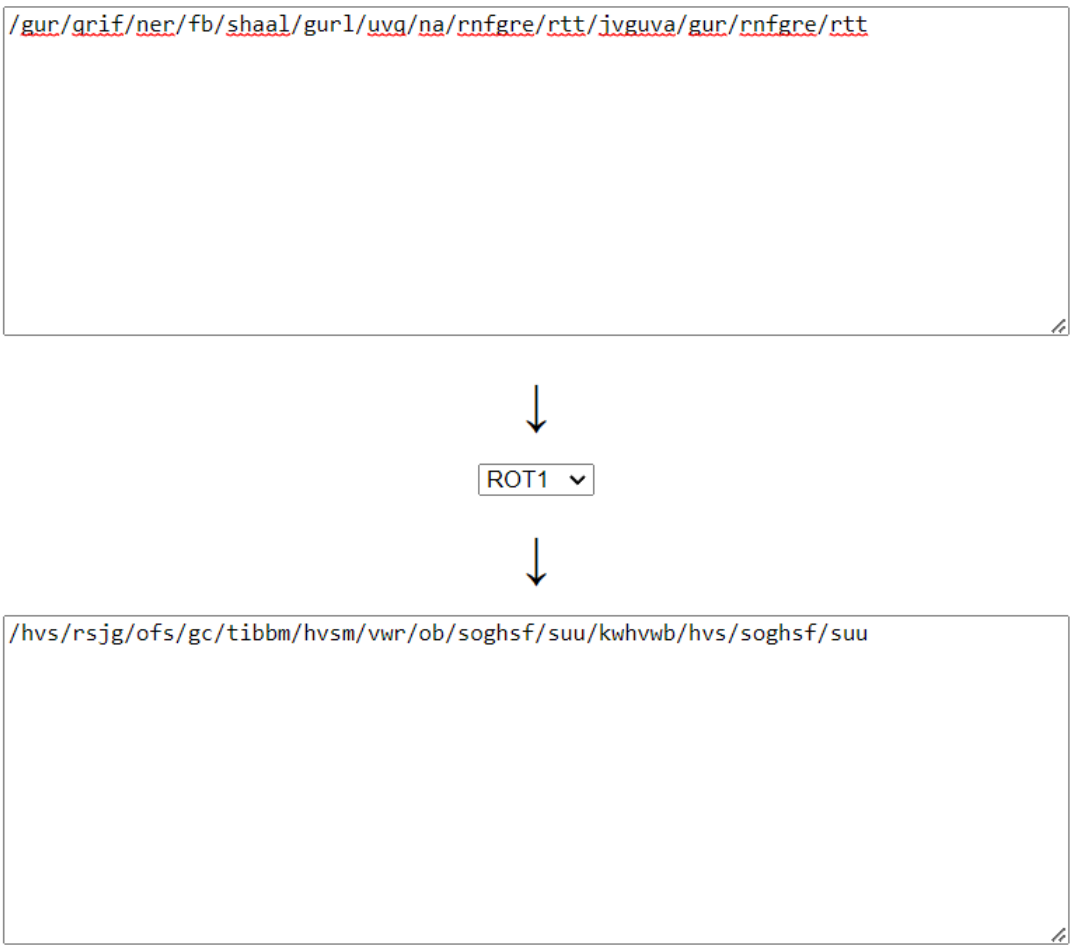


Figure 74 Decrypting the URL using ROT1

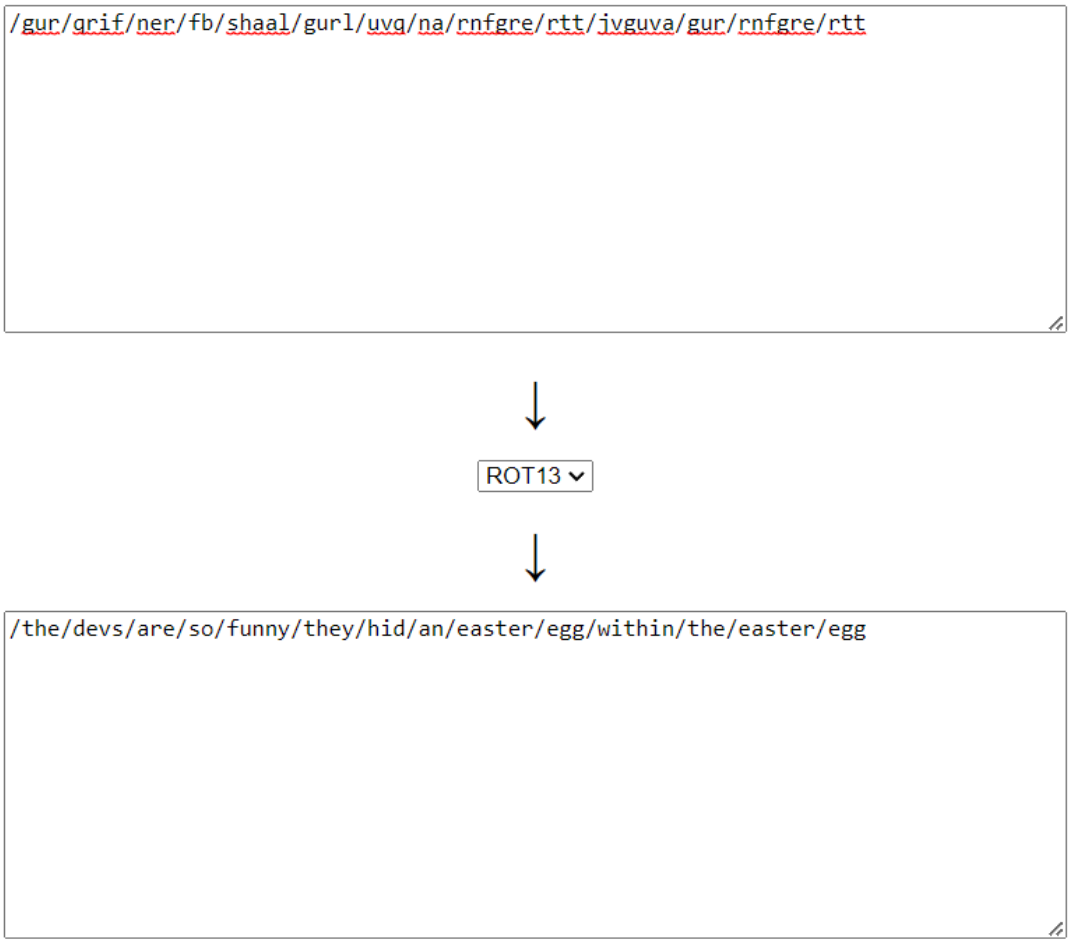


Figure 75 Decrypting the URL using ROT13

Figure 76 URL formed using decrypted value of ROT13

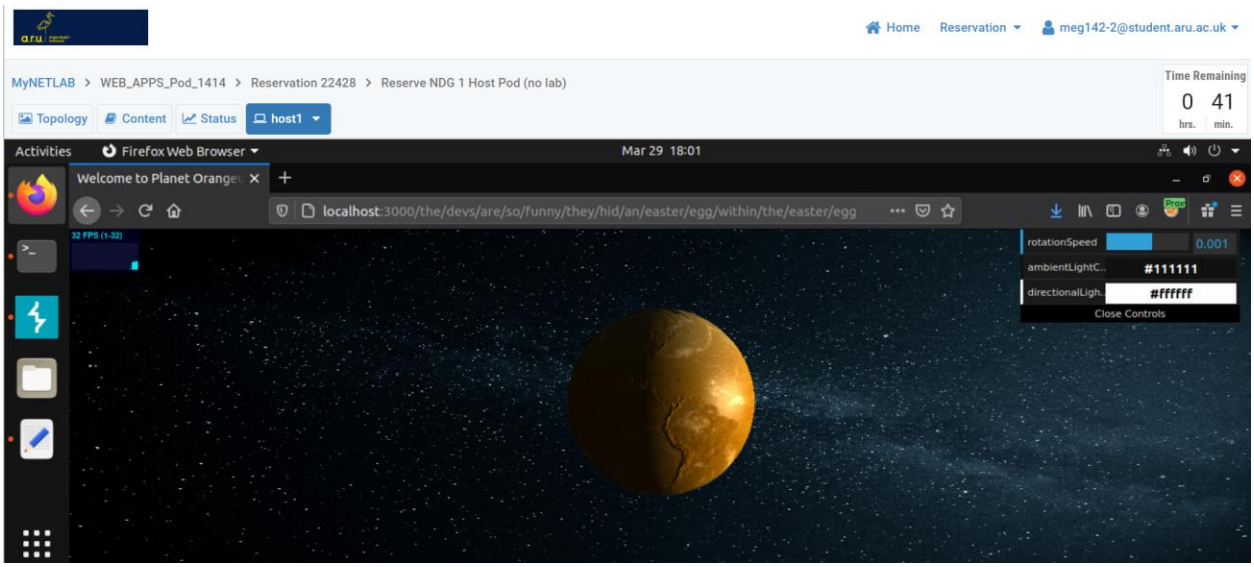


Figure 77 Nested easter egg page

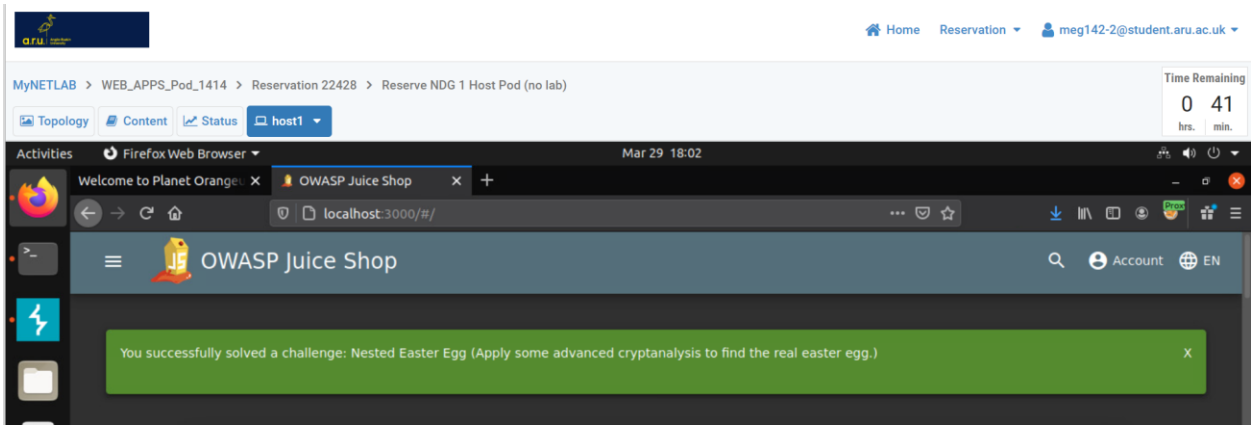


Figure 78 Nested easter egg challenge completed

Weekly Reflection

- Identification and management of application threats is the process of threat modelling.
- Studied about various bugs.
- Learned about threat modeling and its stages.
- Levels of diagram layers

  [OWASP Foundation, 2017]

WEEK 12:

No Lab as Catchup week

REFERENCES

2017 OWASP top 10 -OWASP Top Ten 2017 | 2017 Top 10 | OWASP Foundation. OWASP Foundation. Available at: https://owasp.org/www-project-top-ten/2017/Top_10 (Accessed: January 18, 2023).

Prometheus Authors 2014-2023 (no date) *First steps: Prometheus, Prometheus Blog*. The Linux Foundation. Available at: https://prometheus.io/docs/introduction/first_steps/ (Accessed: February 22, 2023).

Gibson Research Corporation (no date) *GRC's : password haystacks: How well hidden is your needle? , GRC's | Password Haystacks: How Well Hidden is Your Needle?* . Gibson Research Corporation. Available at: https://www.grc.com/haystack.htm (Accessed: February 22, 2023).