IE3092 - Information Security Project

Year 3 Semester 2

Capture the Flag

Mid 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 analysists 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.

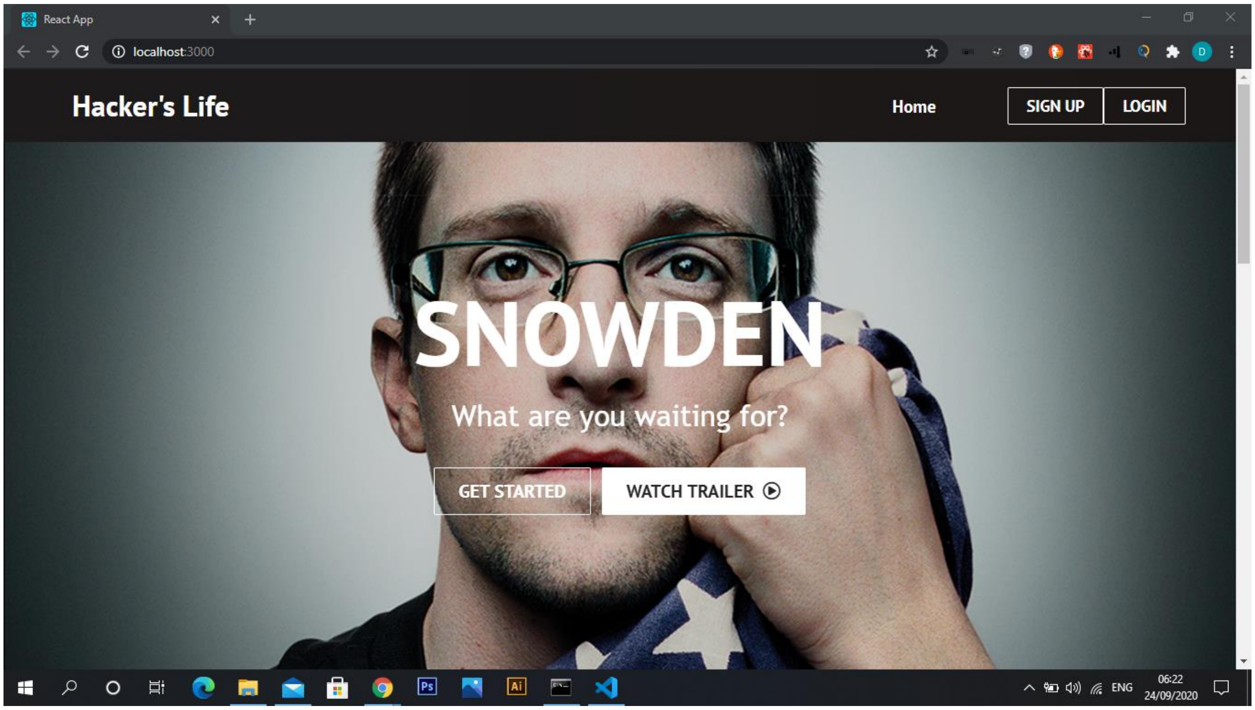
Implementation

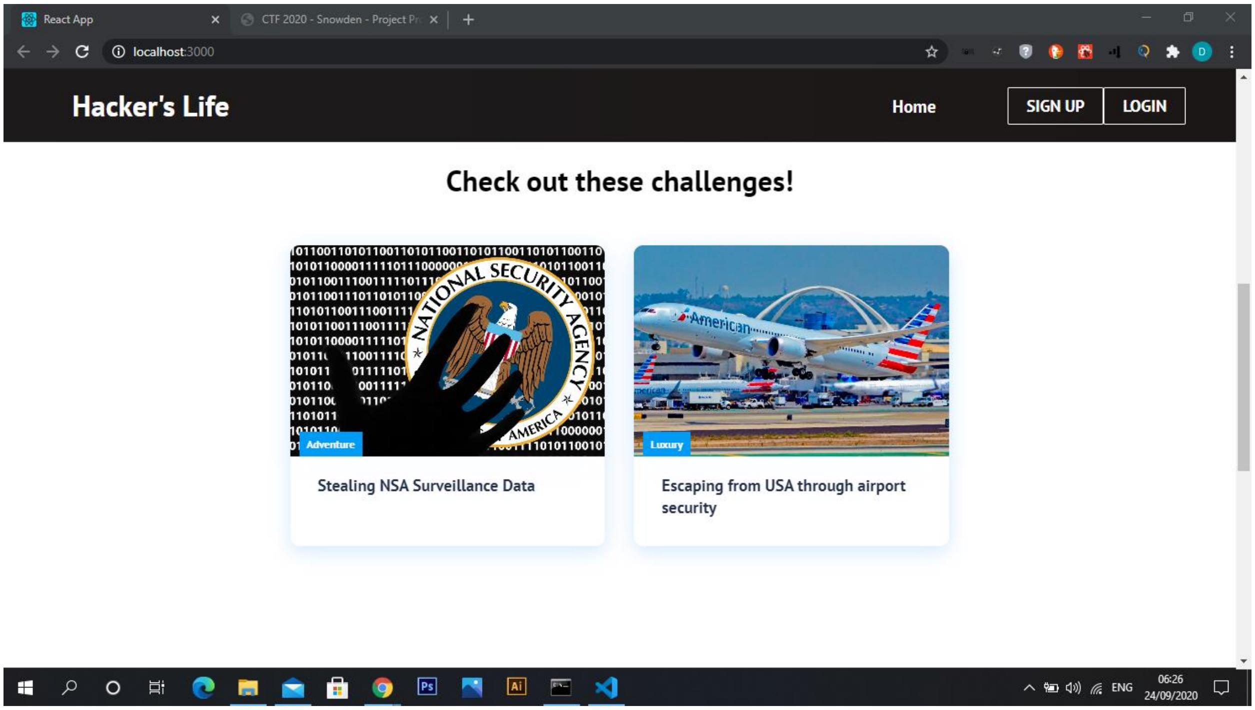
We have used docker containers on Amazon web services EC2 (Elastic Compute Cloud) per each main scenario. So, for the whole CTF box we have used two docker containers on our backend amazon ec2 server. There are separate docker compose yml files per each and every container which is responsible for controlling 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.

Now for the scenarios and it`s tasks were implemented as explained below.

Web App

This is the interface of our CTF box. We have used Snowden`s picture as a background because we are creating this CTF box based on him also we planned this interface to attract more users to this CTF box which will help us to sell this product more.



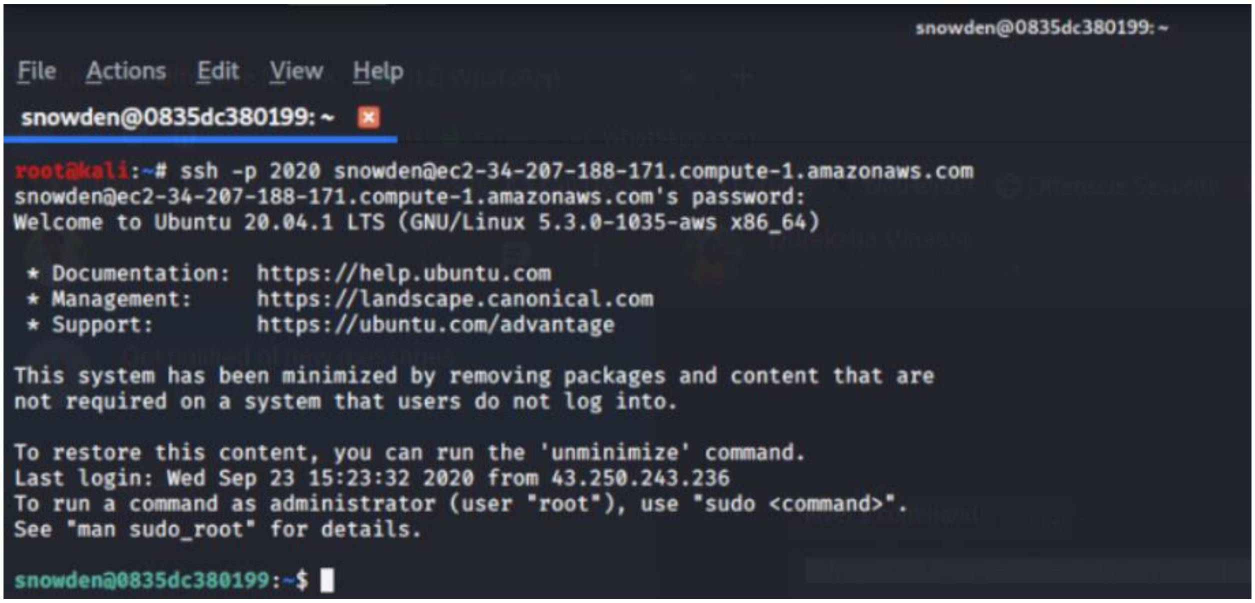

React App
CTF 2020 - Snowden - Project Pr
localhost:3000
Hacker's Life
Home
SIGN UP
LOGIN
Check out these challenges!
Adventure
Stealing NSA Surveillance Data
Luxury
Escaping from USA through airport security
ENG
06:26
24/09/2020

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.

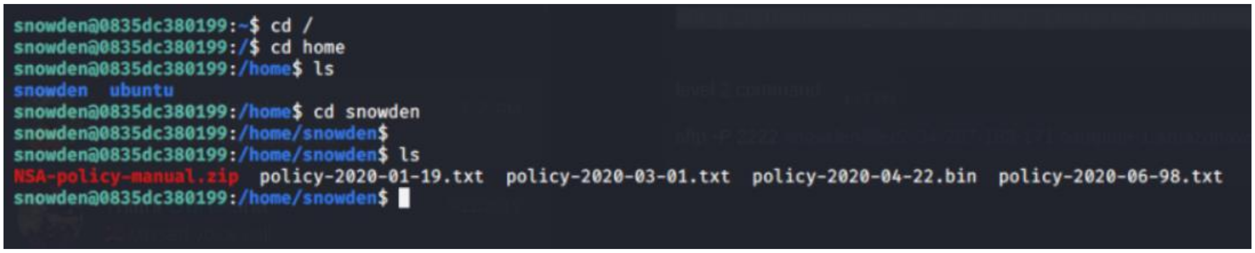
Command, username and password are mentioned below.

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



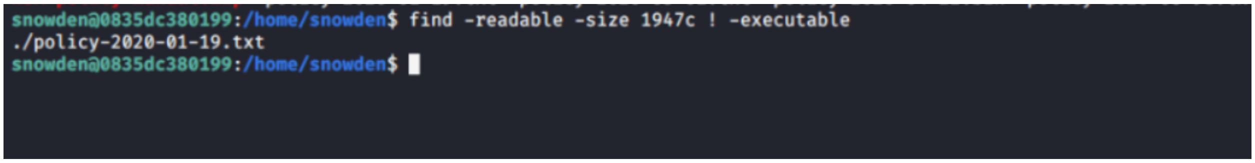
```
root@kali:~# ssh -p 2020 snowden@ec2-34-207-188-171.compute-1.amazonaws.com
snowden@ec2-34-207-188-171.compute-1.amazonaws.com's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.3.0-1035-aws x86_64)

 * 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.



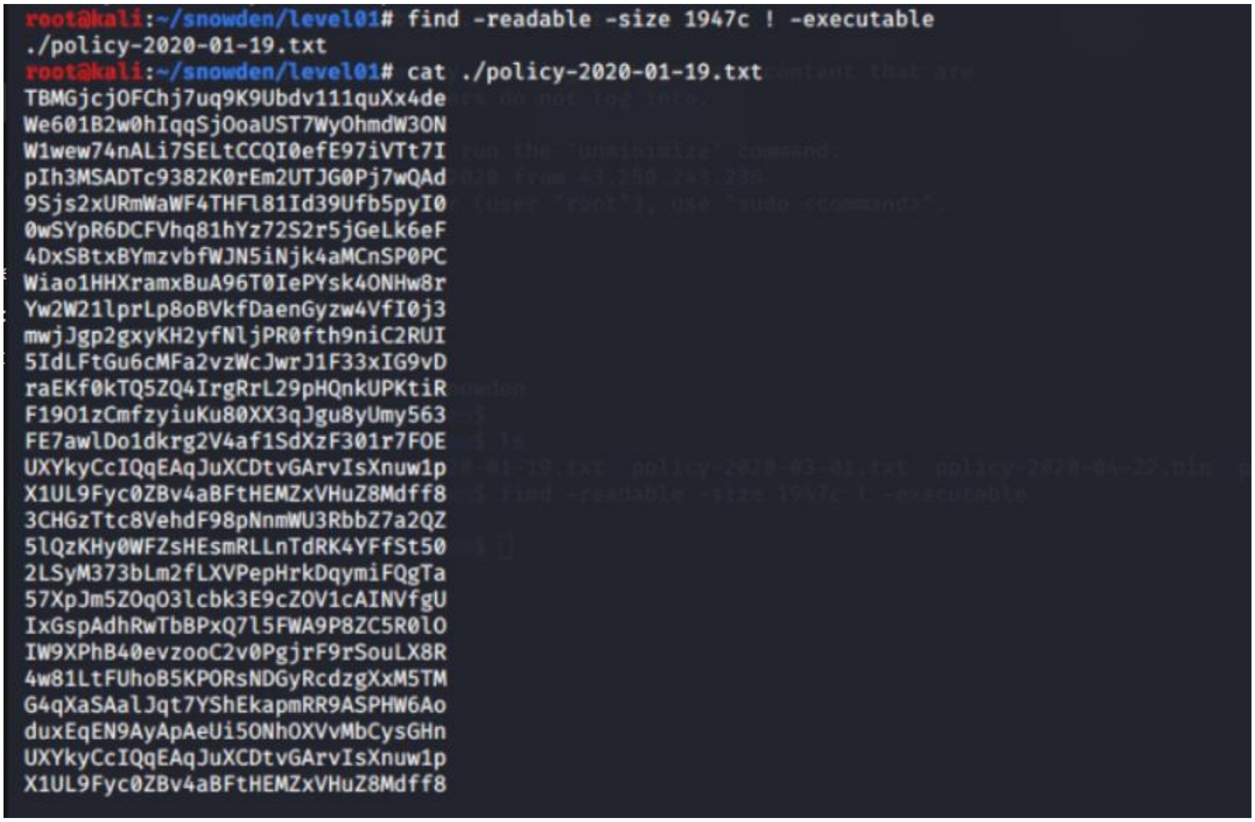
```
snowden@0835dc380199:~$ cd /
snowden@0835dc380199:/$ cd home
snowden@0835dc380199:/home$ ls
snowden  ubuntu
snowden@0835dc380199:/home$ cd snowden
snowden@0835dc380199:/home/snowden$
snowden@0835dc380199:/home/snowden$ ls
NSA-policy-manual.zip  policy-2020-01-19.txt  policy-2020-03-01.txt  policy-2020-04-22.bin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

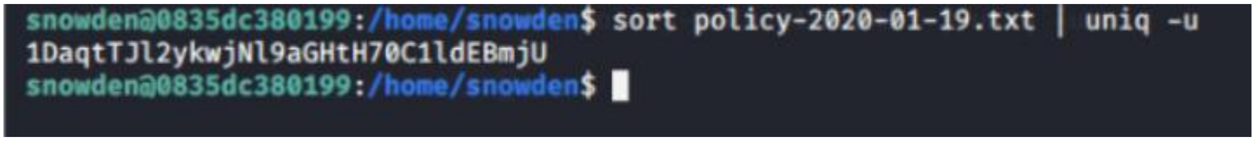
Then using cat command user can get a preview of what this file contains. As this below image shows it contains some 32-character ASCII values. If user has a keen eye there is a unique value, all the other values are repeated several times.



```
root@kali:~/snowden/level01# find -readable -size 1947c ! -executable
./policy-2020-01-19.txt
root@kali:~/snowden/level01# cat ./policy-2020-01-19.txt
TBMGjcjOFChj7uq9K9Ubdv111quXx4de
We601B2w0hIqqSjOoaUST7WyOhmdW3ON
W1wew74nALi7SELtCCQI0efE97iVTt7I
pIh3MSADTc9382K0rEm2UTJG0Pj7wQAd
9Sjs2xURmWaWF4THFl81Id39Ufb5pyI0
0wSYpR6DCFVhq81hYz72S2r5jGeLk6eF
4DxSBtxBYmzvbfWJN5iNjk4aMCnSP0PC
Wiao1HHXramxBuA96T0IePYsk4ONHw8r
Yw2W21lprLp8oBVkfDaenGyzw4VfI0j3
mwjJgp2gxyKH2yfNljPR0fth9niC2RUI
5IdLFtGu6cMFa2vzWcJwrJ1F33xIG9vD
raEKf0kTQ5ZQ4IrgRrL29pHQnkUPKtiR
F19O1zCmfzyiuKu80XX3qJgu8yUmy563
FE7awlDo1dkrg2V4af1SdXzF301r7FOE
UXYkyCcIQqEAqJuXCDtvGArvIsXnuw1p
X1UL9Fyc0ZBv4aBFtHEMZxVHuZ8Mdff8
3CHGzTtc8VehdF98pNnmWU3RbbZ7a2QZ
5lQzKHy0WFZsHEsmRLLnTdRK4YFfSt50
2LSyM373bLm2fLXVPepHrkDqymiFQgTa
57XpJm5ZOqO3lcbk3E9cZOV1cAINVfgU
IxGspAdhRwTbBPxQ7l5FWA9P8ZC5R0lO
IW9XPhB40evzooC2v0PgjrF9rSouLX8R
4w81LtFUhoB5KPORsNDGyRcdzgXxM5TM
G4qXaSAalJqt7YShEkapmRR9ASPHW6Ao
duxEqEN9AyApAeUi5ONhOXVvMbCysGHn
UXYkyCcIQqEAqJuXCDtvGArvIsXnuw1p
X1UL9Fyc0ZBv4aBFtHEMZxVHuZ8Mdff8
```

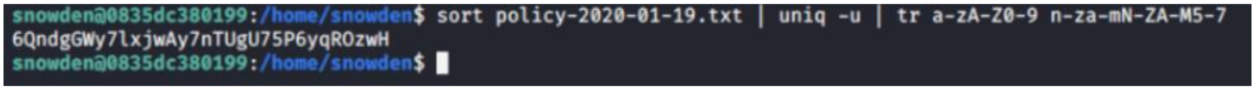
To get the unique value easily following command can be used.

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



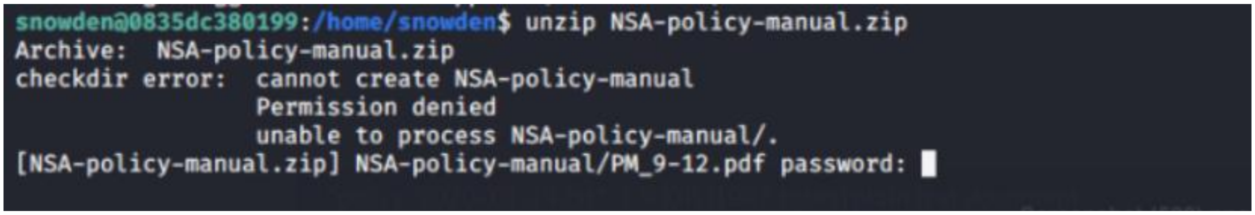
```
snowden@0835dc380199:/home/snowden$ sort policy-2020-01-19.txt | uniq -u
1DaqtTJl2ykwjNl9aGHtH70C1ldEBmjU
snowden@0835dc380199:/home/snowden$ █
```

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.



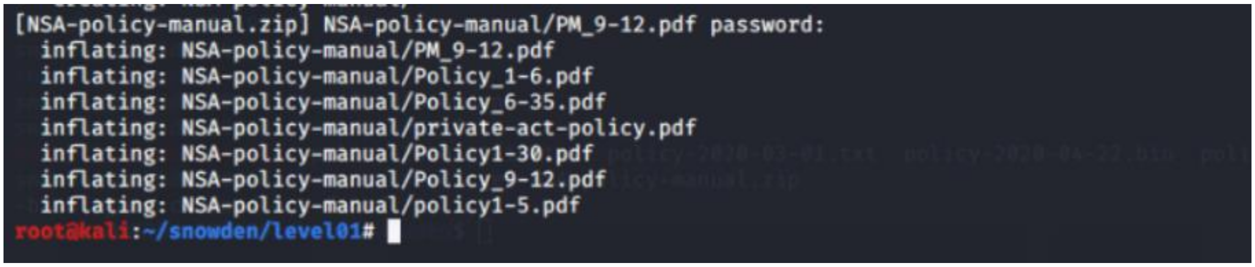
```
snowden@0835dc380199:/home/snowden$ sort policy-2020-01-19.txt | uniq -u | tr a-zA-Z0-9 n-za-mN-ZA-M5-7
6QndgGWy7lxjwAy7nTUgU75P6yqROzwH
snowden@0835dc380199:/home/snowden$ █
```

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: █
```

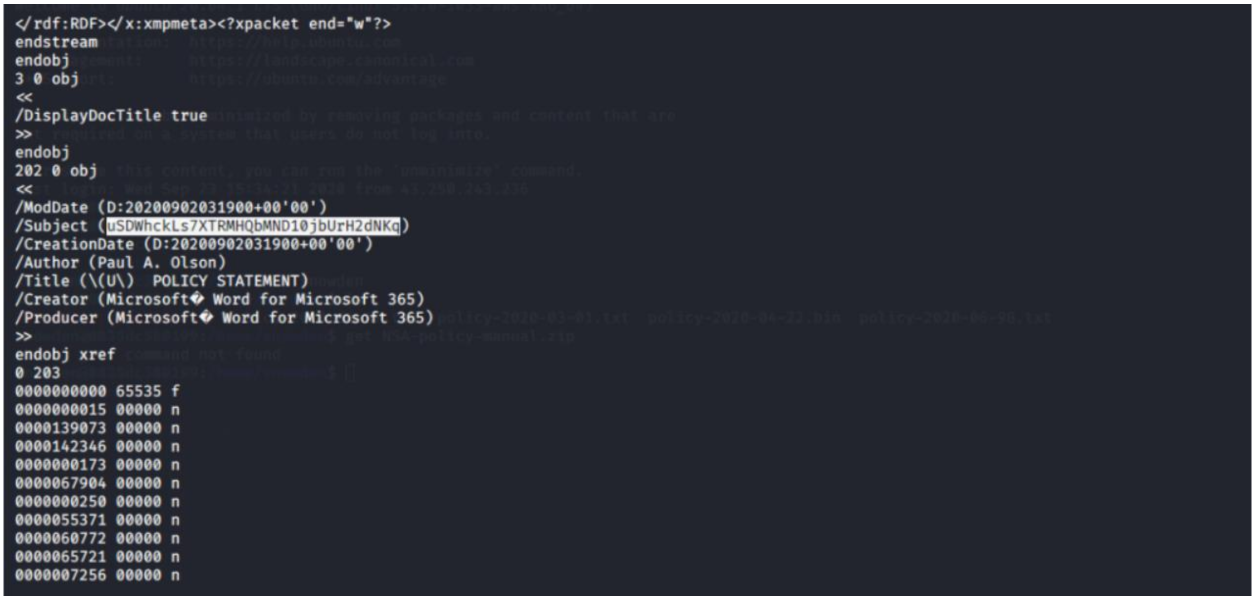
After unzipping followings are the files that contains in that zip folder.



```
[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 (uSDWhckLs7XTRMHQbMND10jbUrH2dNKq)
/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
```

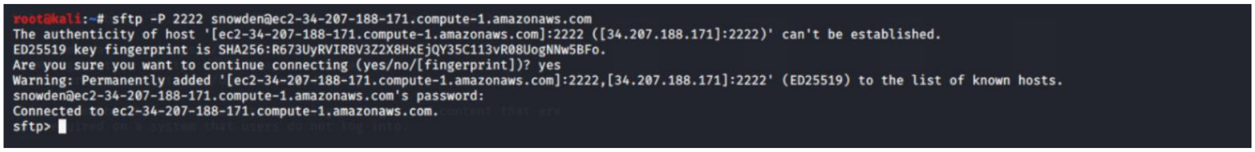
This highlighted text is the flag that user should find on level one. After finding this flag level one will be end.

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.

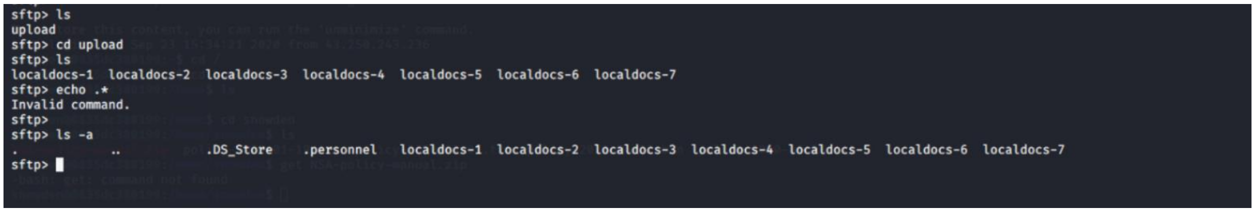
URL: sftp -P 2222 snowden@ec2-34-207-188-171.compute-1.amazonaws.com
Username:
Password: **Flag that caught on level 1**

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



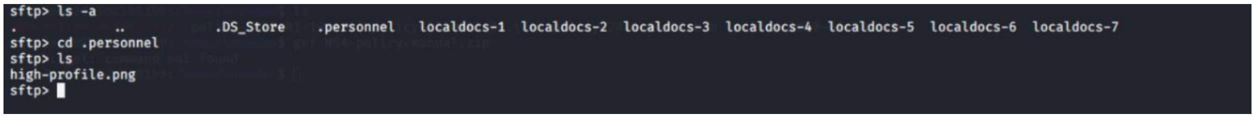
```
root@kali:~# sftp -P 2222 snowden@ec2-34-207-188-171.compute-1.amazonaws.com
The authenticity of host '[ec2-34-207-188-171.compute-1.amazonaws.com]:2222 ([34.207.188.171]:2222)' can't be established.
ED25519 key fingerprint is SHA256:R673UyRVIRBV3Z2X8HxEjQY35C113vR08UogNNw5BFo.
Are you sure you want 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> █
```

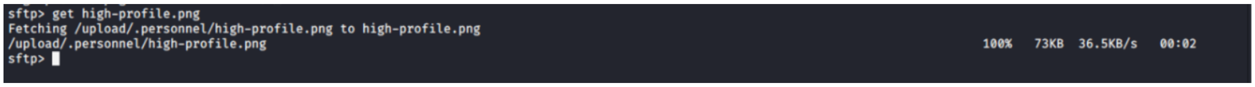
There are several files on that folder and we have hidden the folder that user needs to find to continue the CTF box. So, using ls-all command user can get all the folders and files including hidden folder on this directory. Hidden folder called personnel contains an image called high-profile.



```
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> █
```

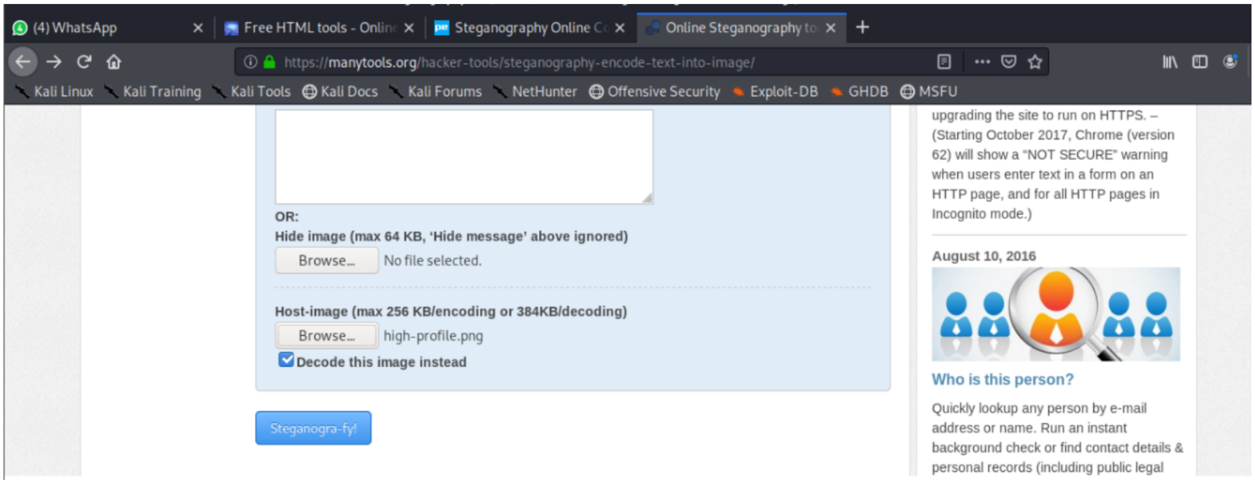
This image is an image of the high-profile VIP people who has access to these confidential data of NSA and responsible for National Security of the country. This image is just an avatar image which contains a message. We have used technology called steganography to encode a message to this image. So, 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                                                        100%   73KB  36.5KB/s   00:02
sftp> █
```

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



OR:
Hide image (max 64 KB, 'Hide message' above ignored)
Browse... No file selected.

Host-image (max 256 KB/encoding or 384KB/decoding)
Browse... high-profile.png
☑ Decode this image instead

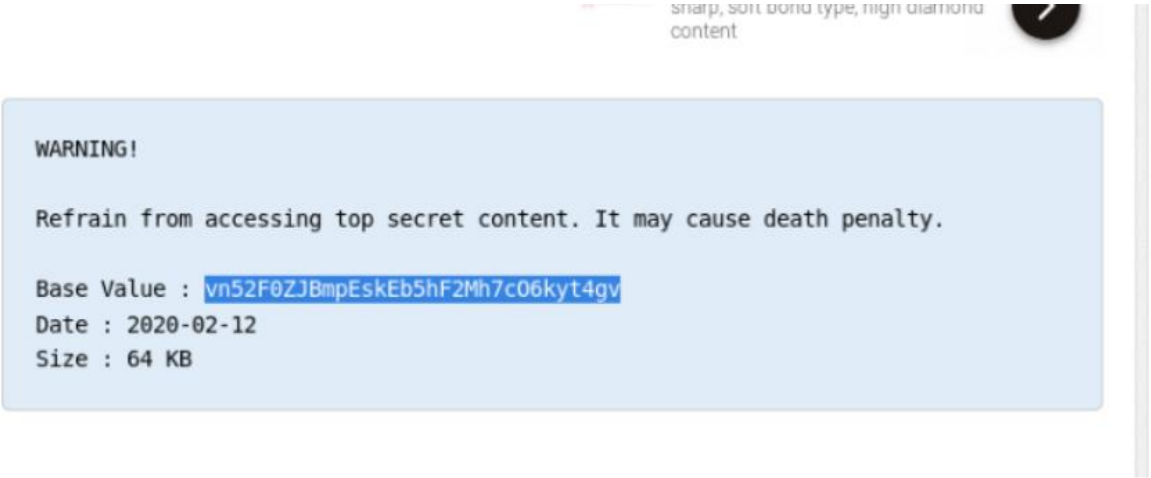
Steganogra-fy!

upgrading the site to run on HTTPS. – (Starting October 2017, Chrome (version 62) will show a "NOT SECURE" warning when users enter text in a form on an HTTP page, and for all HTTP pages in Incognito mode.)

August 10, 2016

Who is this person?

Quickly lookup any person by e-mail address or name. Run an instant background check or find contact details & personal records (including public legal



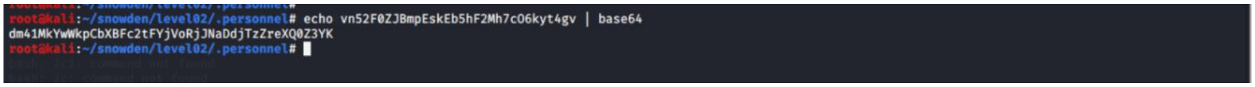
sharp, soft bond type, high diamond content

```
WARNING!

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

Base Value : vn52F0ZJBmpEskEb5hF2Mh7c06kyt4gv
Date : 2020-02-12
Size : 64 KB
```

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.



```
root@kali:~/snowden/level02/.personnel# echo vn52F0ZJBmpEskEb5hF2Mh7c06kyt4gv | base64
dm41MkYwWkpCbXBFc2tFYjVoRjJNaDdjTzZreXQ0Z3YK
root@kali:~/snowden/level02/.personnel# █
```

Future Progress Plan

There will be another level in this scenario and Scenario 2 will be implemented in future with more complex hacking techniques relevant to web security.