 PSG COLLEGE OF TECHNOLOGY  
DEPARTMENT OF APPLIED MATHEMATICS AND COMUTATIONAL SCIENCES

20XC58-ETHICAL HACKING LAB

PACKAGE REPORT

TITLE: VOYAGE TO THE GRAND LINE

­­­

TEAM MEMBERS:

1. 21PC17 – JYOTHISH K S
2. 21PC19 – NANDA PRANESH S
3. 21PC24 – S R ASHUWANTHH
4. 21PC25 – S VARUN

ABSTRACT

This abstract introduces a unique and engaging cybersecurity challenge in the form of a vulnerable black box, inspired by the world of the anime "One Piece." The black box is constructed using an Ubuntu Server image and is designed to emulate the concept of Capture The Flag (CTF) competitions, providing those who solve this box with an immersive and thrilling experience.

The objective of this challenge is to capture a total of 17 flags hidden within the black box, each representing a different cybersecurity skill or knowledge domain. Those who’re solving this box, are tasked with exploring and exploiting various vulnerabilities in a controlled environment, much like real-world cybersecurity scenarios. As they progress through the challenge, they will encounter a series of puzzles, obstacles, and security configurations inspired by the adventures of the "One Piece" universe.

This black box challenge not only tests peoples' technical abilities but also encourages creative problem-solving and teamwork. It provides a platform for individuals to hone their cybersecurity skills while offering an enjoyable and immersive experience rooted in the popular anime series "One Piece."

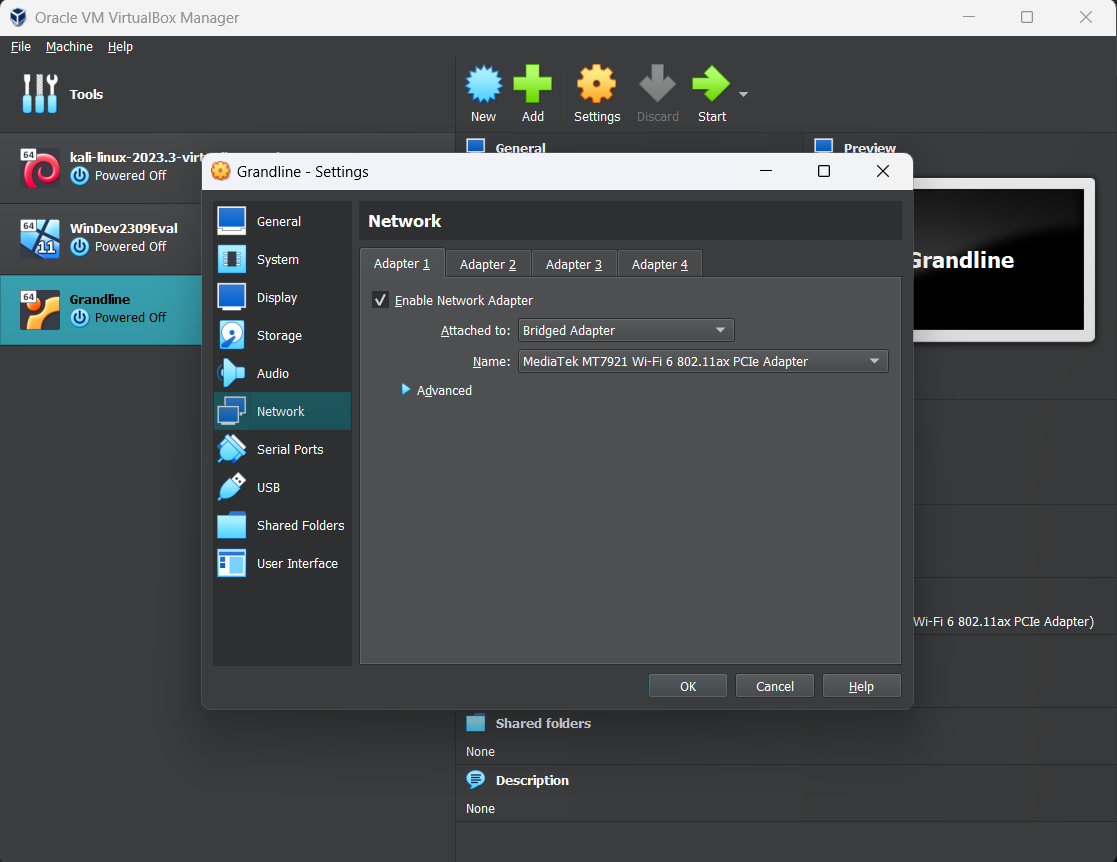
In summary, this black box challenge offers an exciting and educational opportunity for cybersecurity enthusiasts to test their mettle and embark on a thrilling journey to capture all 17 flags while paying homage to the adventurous world of "One Piece."

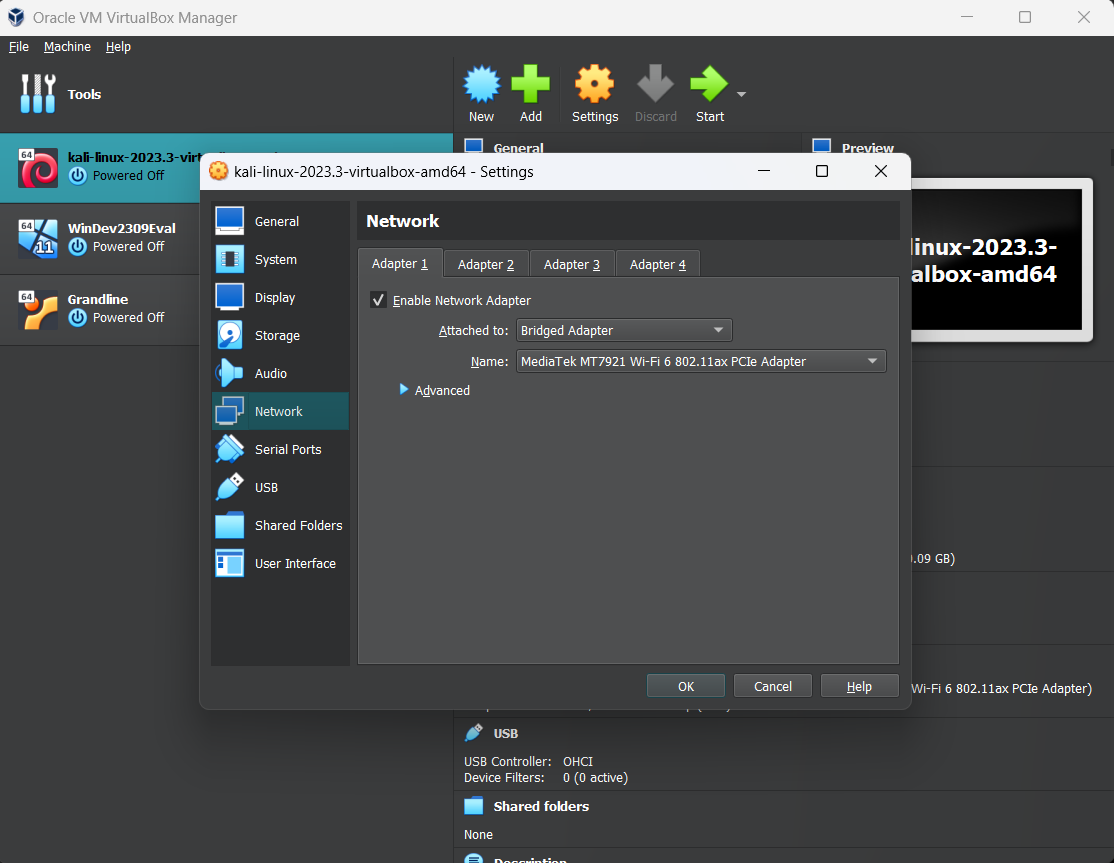
REQUIREMENTS

* Oracle VM VirtualBox
* Ubuntu Server image
* Kali Linux virtual machine

FLAGS

Before finding the flags, our first step here is to bridge the network between the downloaded ova file and the existing Kali Linux image. This is done so as for the image “GrandLine” gets its own IP, and exist in the same network as in the Kali Linux.

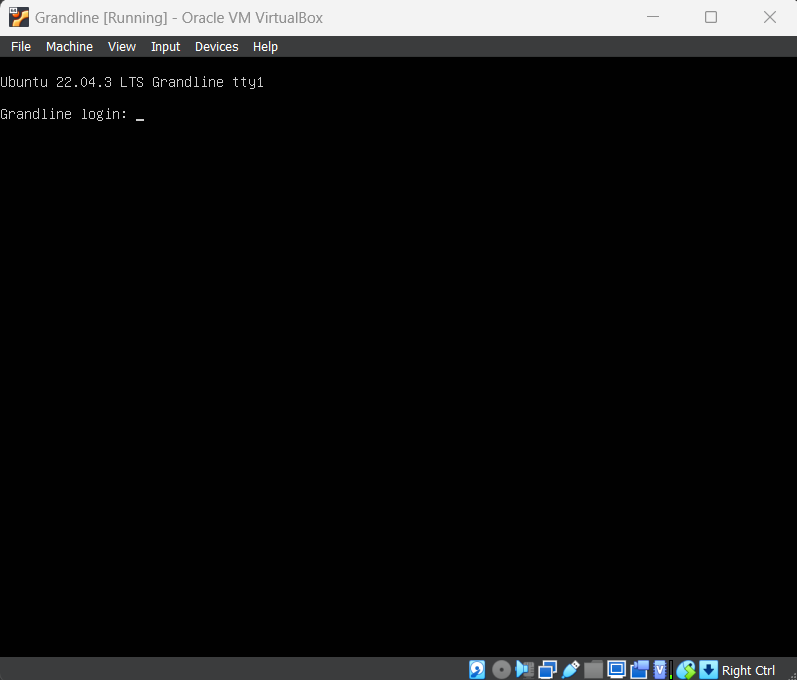




Make sure the Kali Linux is still connected to NAT network in any other adapter, to ensure that Kali Linux is connected with the outside network.



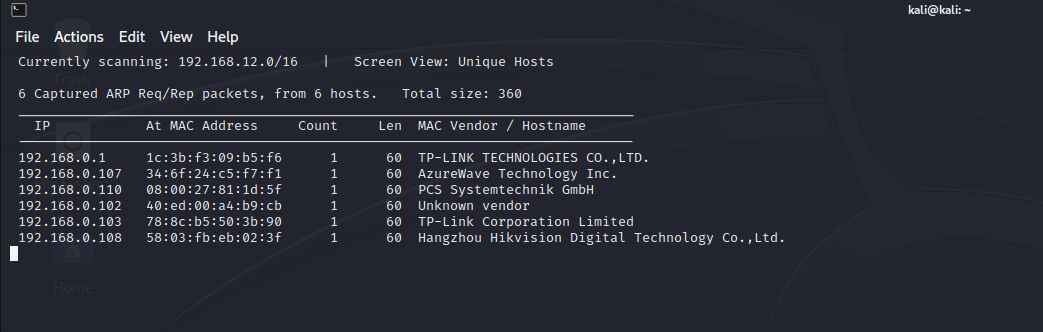
Now open both the machines. The machine Grandline will ask for username and password, which is unknown.



In order to find username and password, we get the help of Kali Linux, and hence entering the journey in finding the first flag.

FLAG 1:

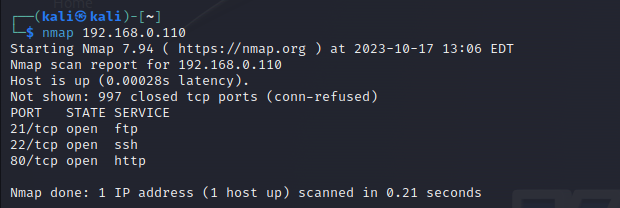
Kali Linux >> Terminal >> sudo netdiscover



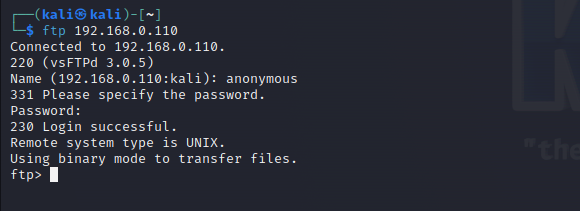


IP address => 192.168.0.110(It will differ every time we change the network, but this IP will be mentioned throughout the report for understanding purposes).

nmap <your IP> (To see the ports open in that device)

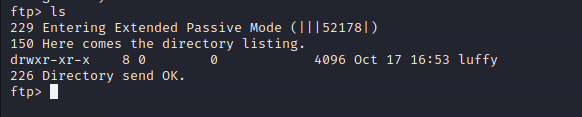


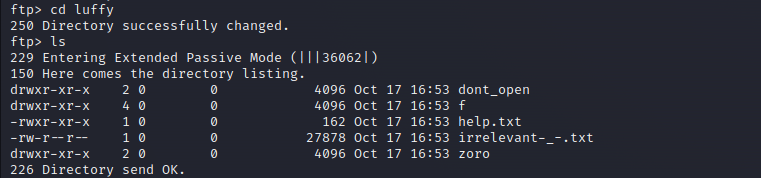
As in the image above, the ftp port is open. Hence, we try anonymous login through ftp port to see if it’s open.

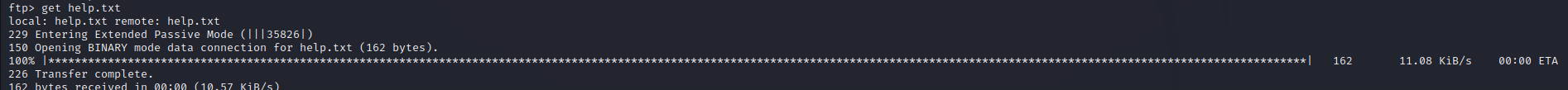


The anonymous login is successful.

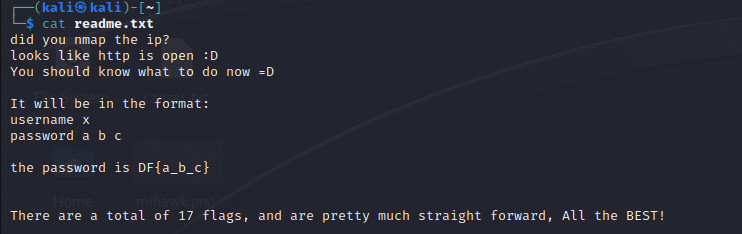
ls > cd luffy> get help.txt:



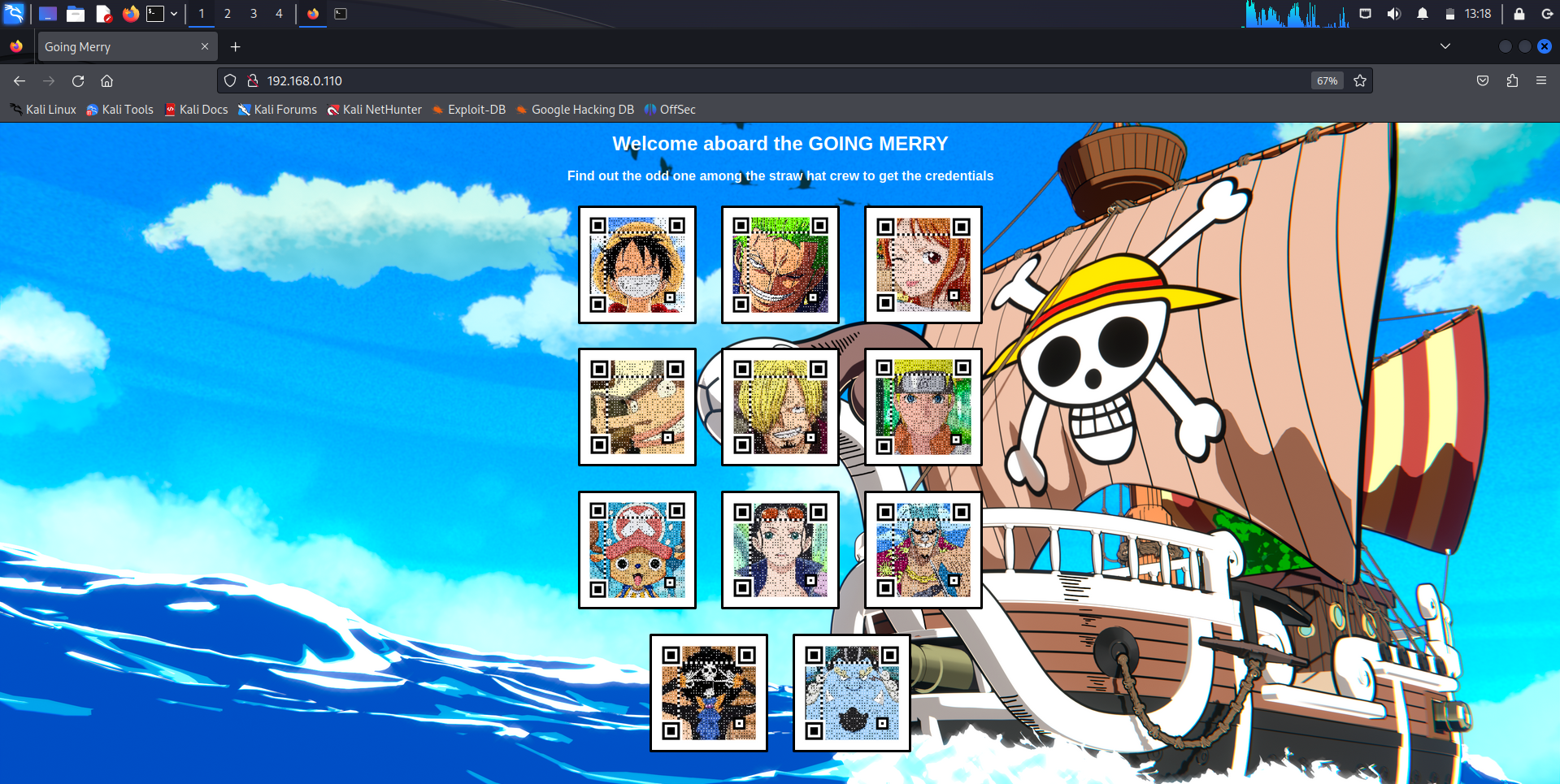




cat help.txt:



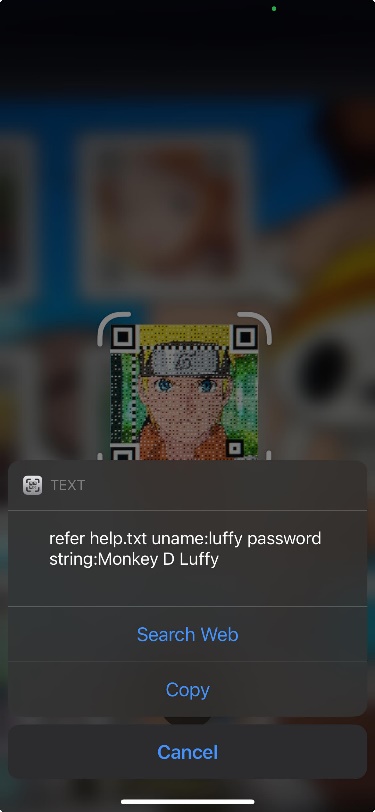
As in nmap results, http seems to be open. Now we try to access the machine through http portal.

Browser >> <IP ADDRESS>:80



Looks like there’s a set of QR codes, and we’re supposed to find the odd one out.

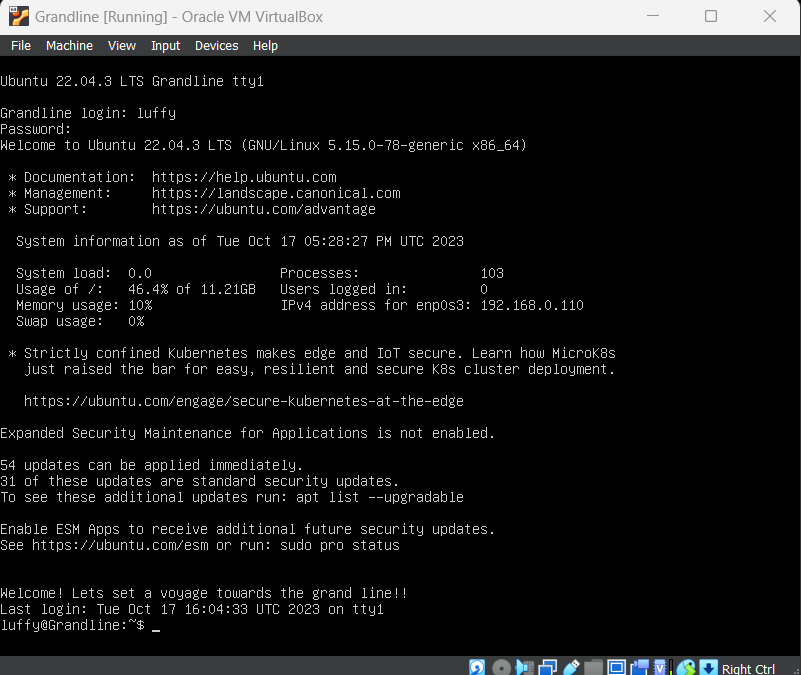
If you’re an anime freak, you’d find out that Naruto(circled) is the odd one out, in an instant. Scanning the QR gives:



Referring the text document help.txt, we hence get the username and password :

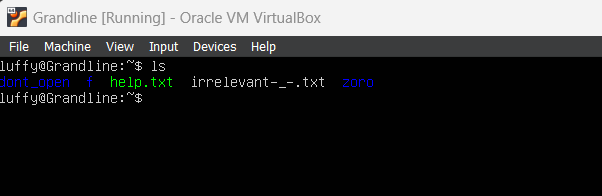
Username : luffy

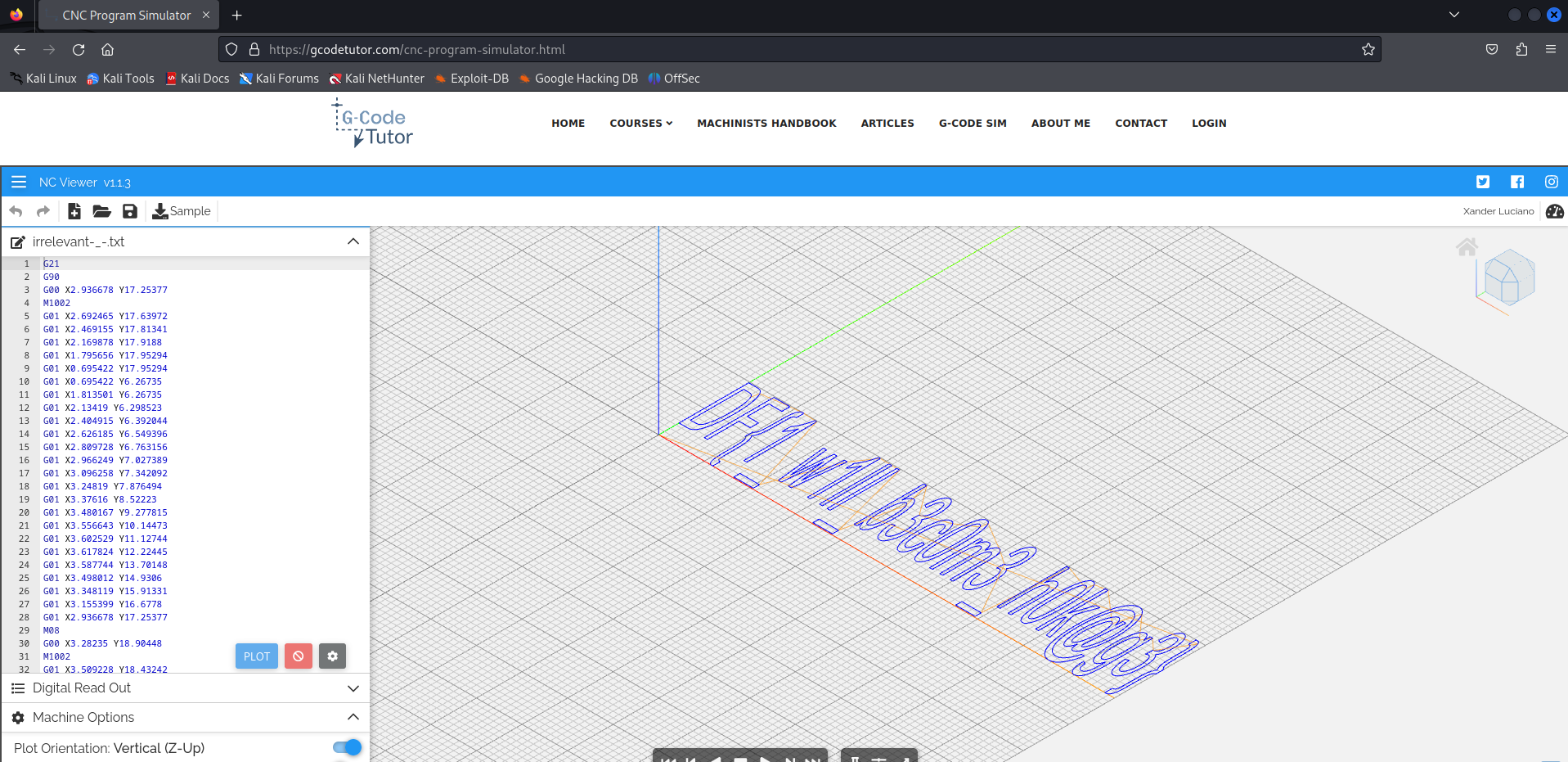
Password : DF{Monkey\_D\_Luffy} => Flag 1





FLAG 2:

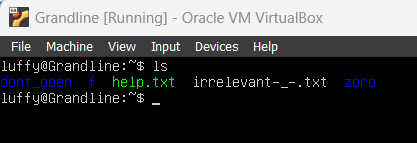


cat irrelevant-\_-.txt gives some hundreds of gibberish lines, which are not gibberish infact, but some g-code. We can download the file through ftp in linux and upload it in the nc viewer, which gives :  


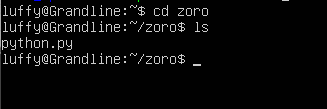
* Flag 2 : DF{1\_w1ll\_b3c0m3\_h0k@g3}

FLAG 3:

cd in the pwd shows a folder named zoro:



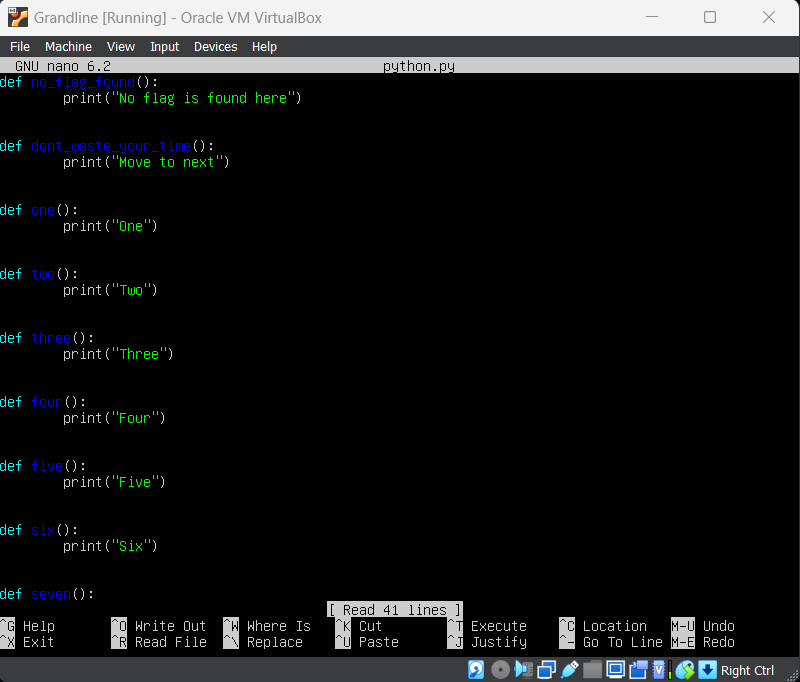
cd zoro>ls:



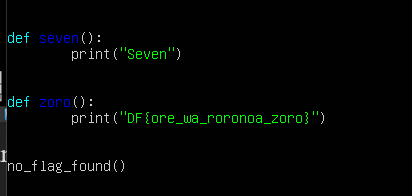
python3 python.py:



nano python.py:



Looks like the flag is nowhere to be found, but if we go down:

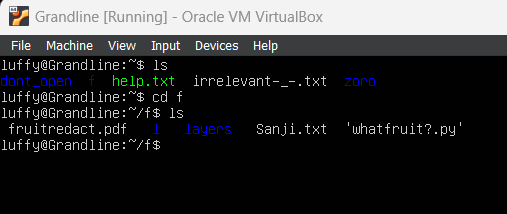


* Flag 3 : DF{ore\_wa\_roronoa\_zoro}

FLAG 4:

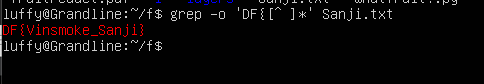
In home directory, there’s a folder ‘f’.

cd f>ls:



Sanji.txt seems like a text file. Since we know the format of the flag(“DF{<flag>}”), we can see if the flag is inside Sanji.txt with the help of grep command:

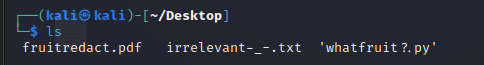
grep -o 'DF{[^ ]\*' Sanji.txt



* Flag 5: DF{Vinsmoke\_Sanji}

FLAG 5:

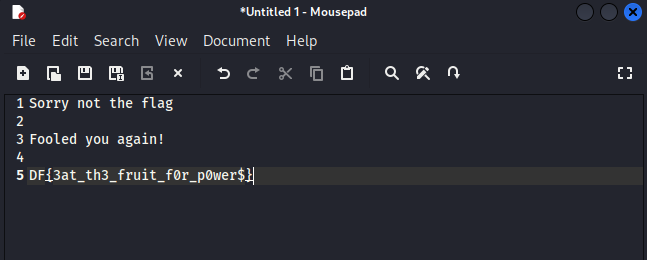
In the directory f, we can see there are two files, apart from Sanji.txt. We can download them in kali through ftp with the help of get command.



Opening fruitredact.pdf, we get to see three sentences that are redacted.



We can get them by just copy pasting them in the note pad.



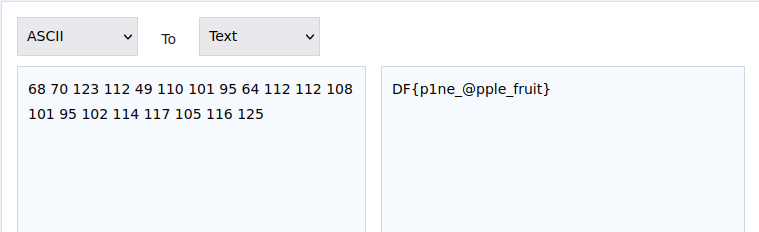
* FLAG 5 : DF{3at\_th3\_fruit\_f0r\_p0wer$}

FLAG 6:

Opening the whatfruit?.py shows a python script :



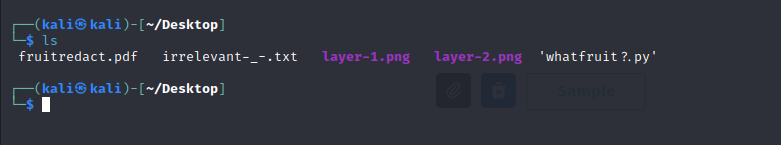
The functions are the full form of ASCII, from which we can conclude that the numbers mentioned are in ASCII format, and we just have to convert them back to text. Converting them back to text gives us :

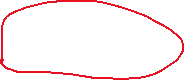


* Flag 6: DF{p1ne\_@pple\_fruit}

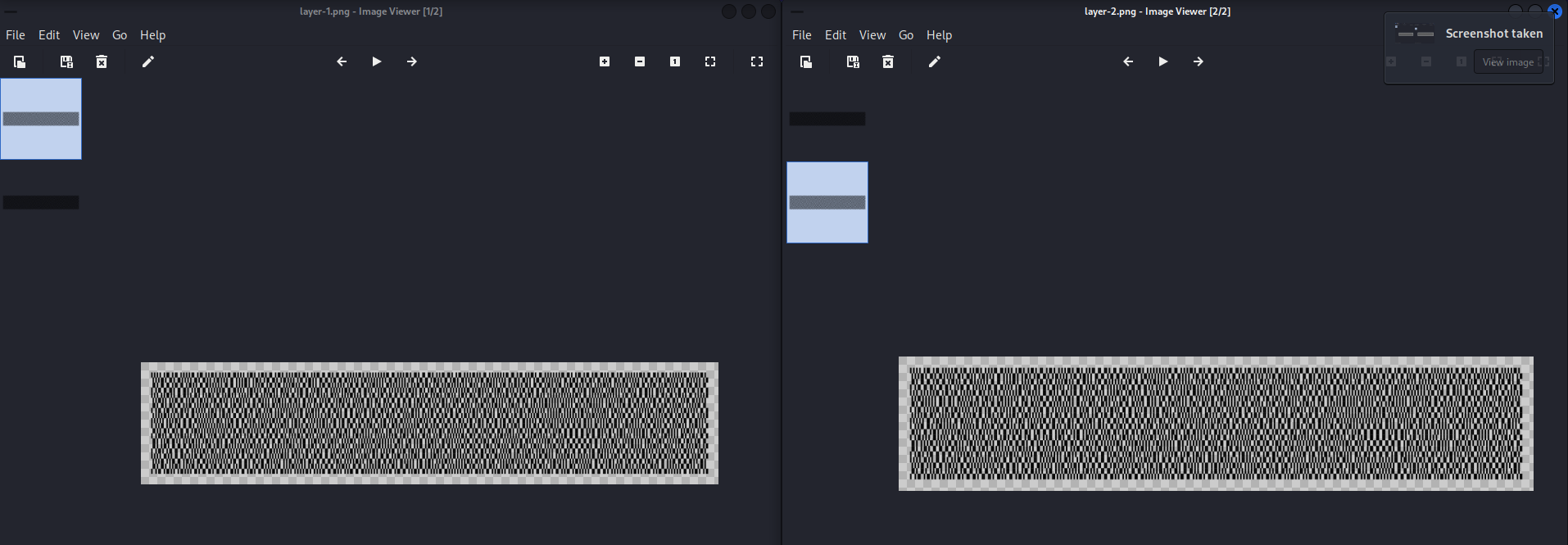
FLAG 7:

As we saw in the ‘f’ folder, there was a folder named ‘layers’. Opening them in kali through ftp connection shows us two images, layer-1.png and layer-2.png.

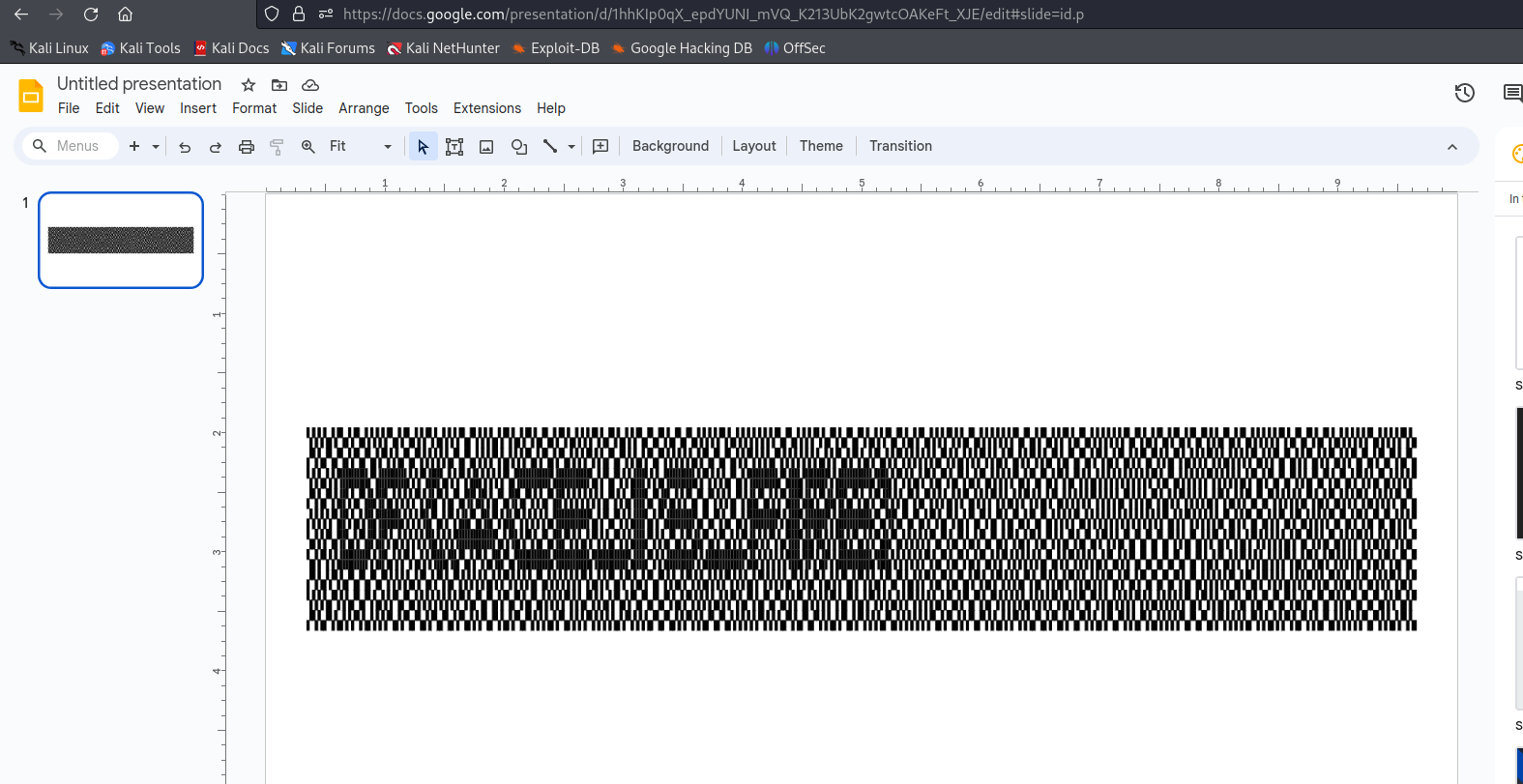




Those two are some sort of pixelated images:



What will happen if we stack them on top of each other? Lets try them :



* Flag 7: DF{ACE\_IS\_FIRE}

FLAG 8:

Now that we tried out everything in the directory ‘f’, let’s look at directory ‘l’.

cd l>ls:

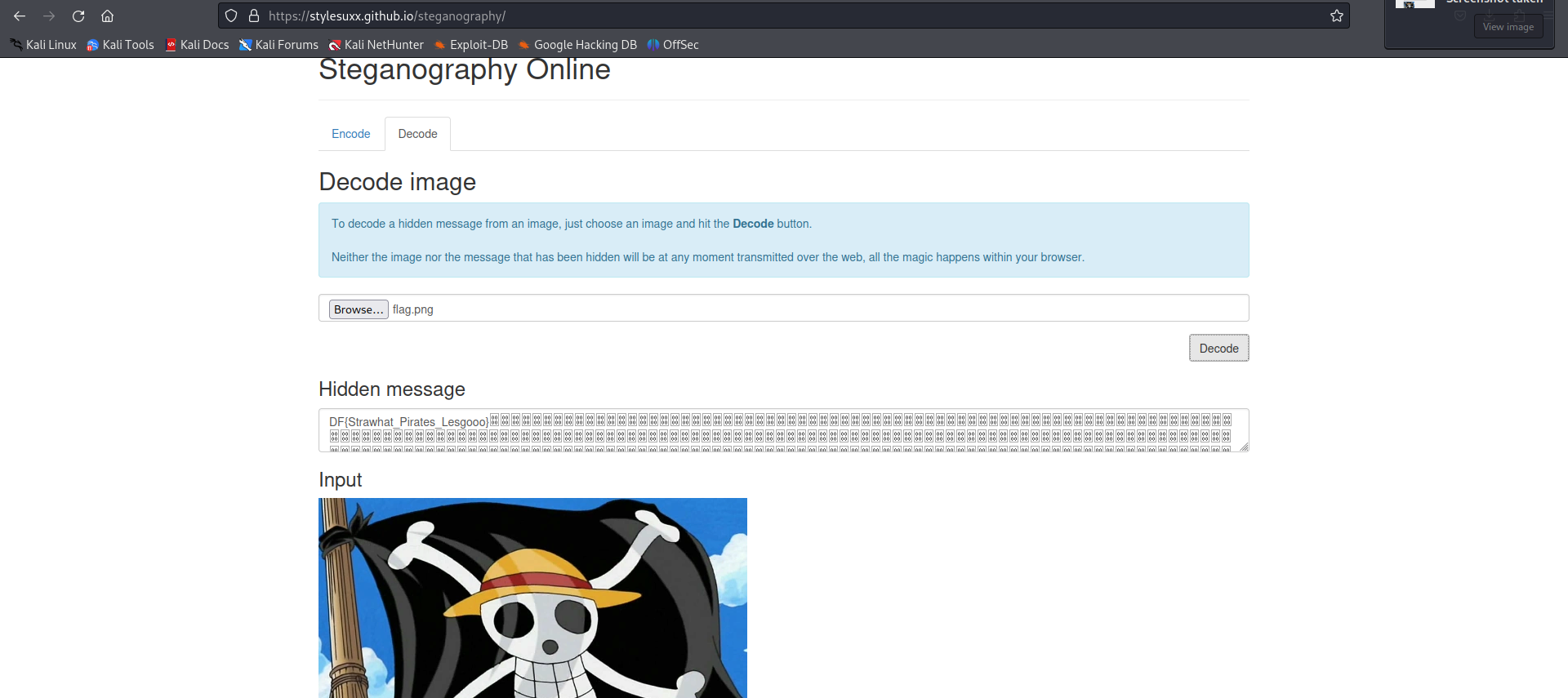


Let’s first download flag.png in kali.

After downloading them, when we open flag.png, we will see a literal flag. It’s the flag that represents the “Straw hat Pirates”.



It seems to be normal, but what if it’s some sort of steganographic junk? Lets try it out using steganography decoder.

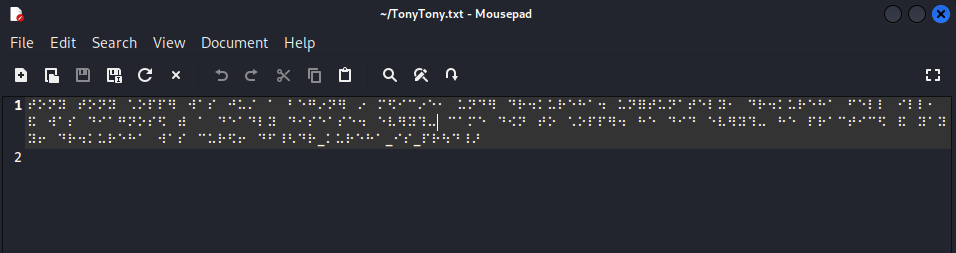


Oh look, there’s the flag!

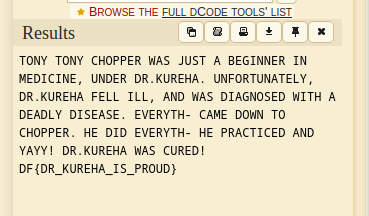
* Flag 8: DF{Strawhat\_Pirates\_Lesgooo}

FLAG 9:

The other text file, TonyTony.txt shows something like this:



Looks like a braille script, lets try decoding it:



* FLAG 9: DF{DR\_KUREHA\_IS\_PROUD}

FLAG 10:

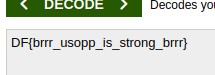
One of the files there was Syrup village. Opening them shows a gibberish, but there is another file saying “map\_to\_syrup”. Opening that file, we see:



@bash is a kind of cipher, so applying it on the text file “Syrup village”, we get:



It says look through the base, which may mean base64:

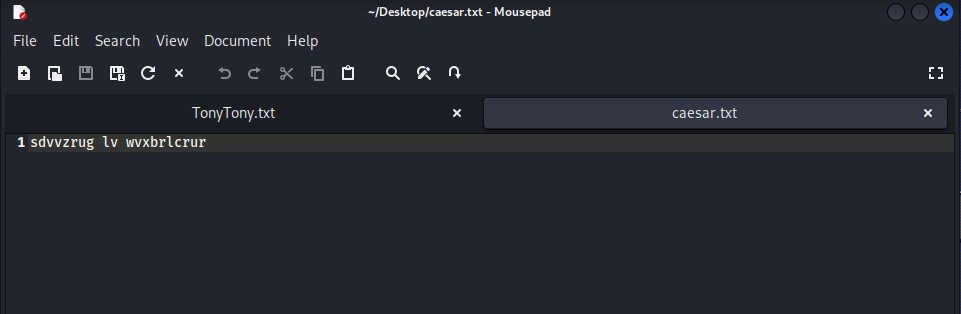


* Flag 10: DF{brr\_usopp\_is\_strong\_brrr}

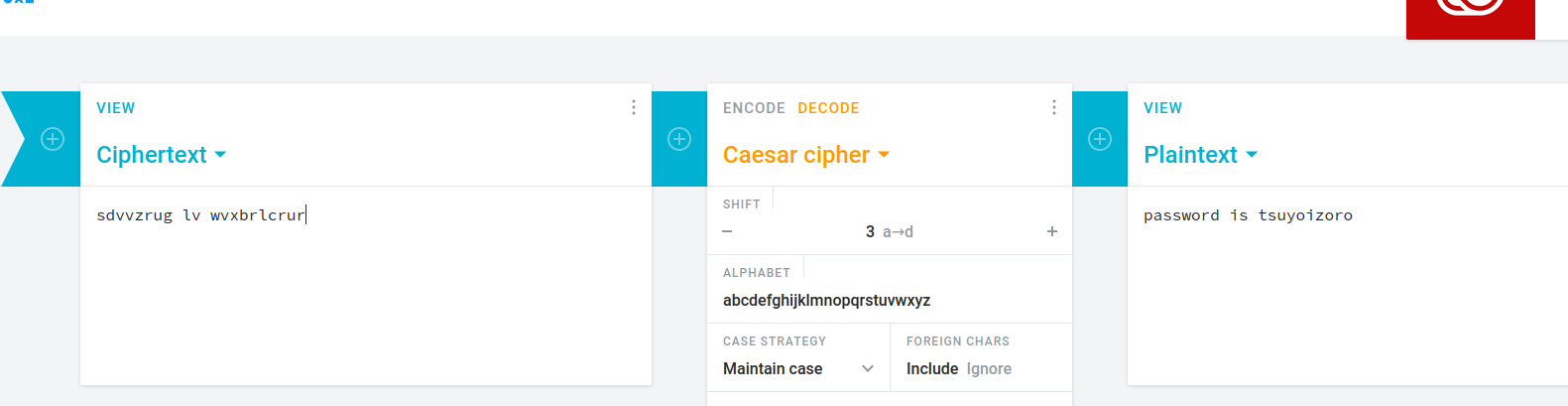
FLAG 11:

Inside pwd\_enc directory, we can see two files, ‘caesar.txt’ and ‘mihawk.png’. Let’s download them in kali.

Caesar.txt:



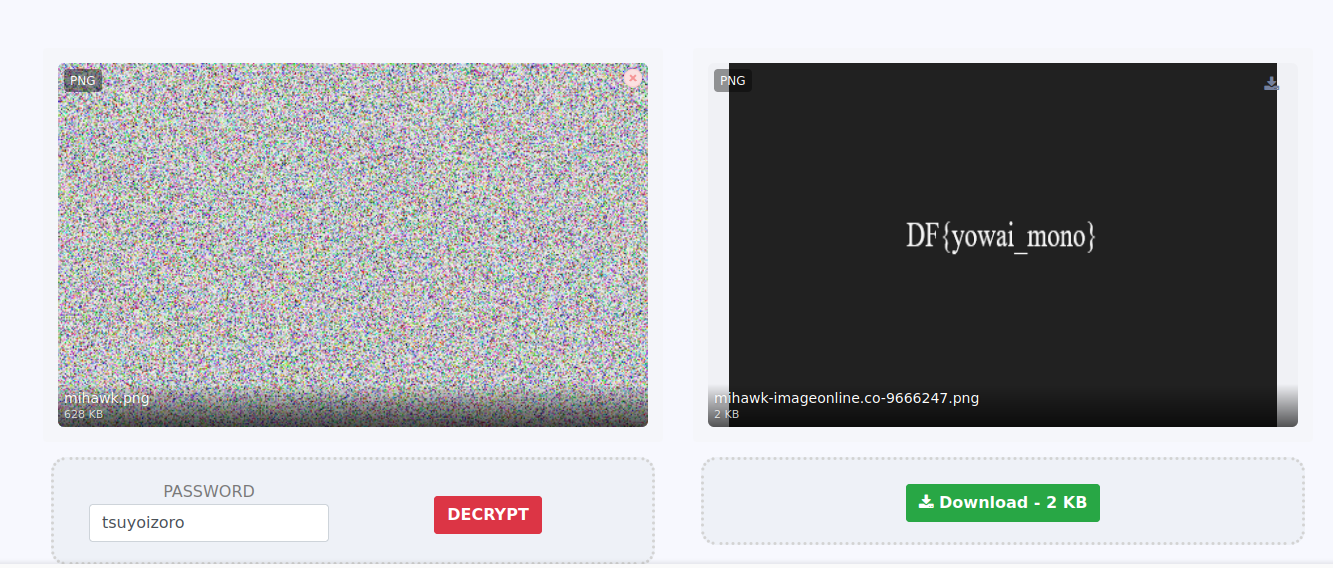
As the name tells, we can try to decode them using Caesar cipher.



It says password is tsuyoizoro. But what is it password for?? Looking at the folder name, it says “pwd\_enc”, which means password\_encryption. Maybe password was pointing to the text file and encryption was pointing towards the image. Let’s try decrypting the image with this password.

The image is:



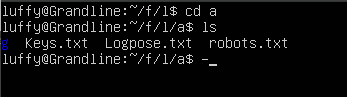


* FLAG 11: DF{yowai\_mono}

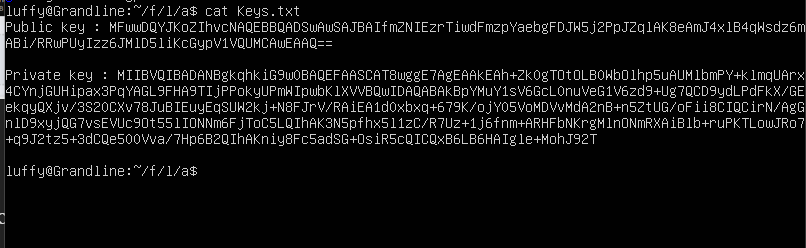
FLAG 12:

Now that we’re done with folder ‘l’, we’ll move on to folder ‘a’.

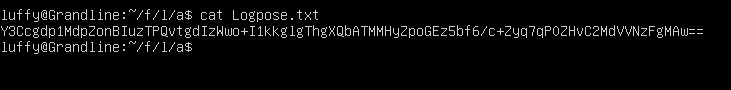
cd a>ls



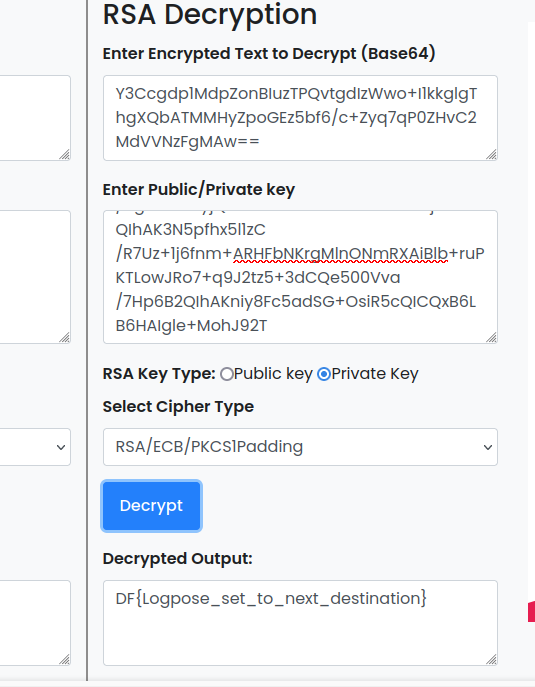
Looks like the content inside the text file Keys.txt seems to be some sort of public and private keys for a cipher.



And Logpose.txt seems to be encrypted:



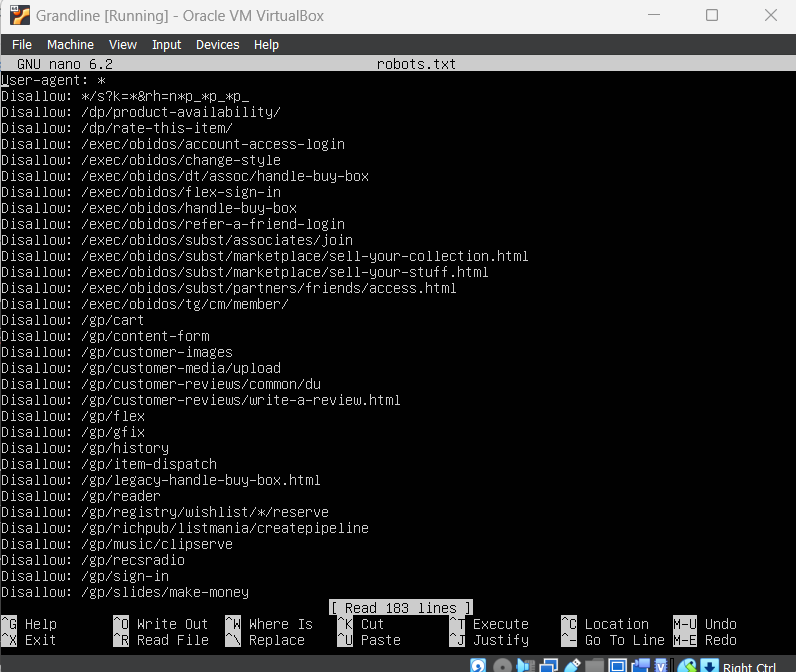
Let’s download them in kali and try RSA on them:



* Flag 12: DF{Logpose\_set\_to\_next\_destination}

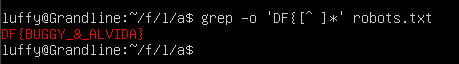
FLAG 13:

The other file in this folder is robots.txt, which is usually a web crawler file that can be found in popular webpages like amazon.com etc. On opening, it looks like this :



If we try grep, we’ll get:

grep -o 'DF{[^ ]\*' robots.txt

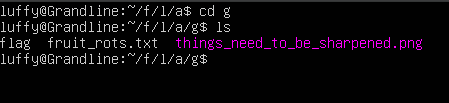


* Flag 13: DF{BUGGY\_&\_ALVIDA}

FLAG 14:

Now that we’re done with the folder ‘a’, we now go with folder ‘g’.

cd g>ls:





Upon entering this folder, we can notice something. The directories traversed all together seems to say the word : “flag”, similar to the file named “flag”. This shows the directory traversal aspect.

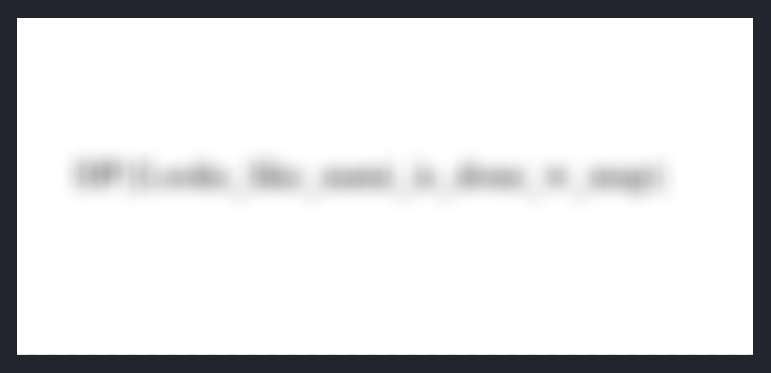
Upon opening the file flag, we get:



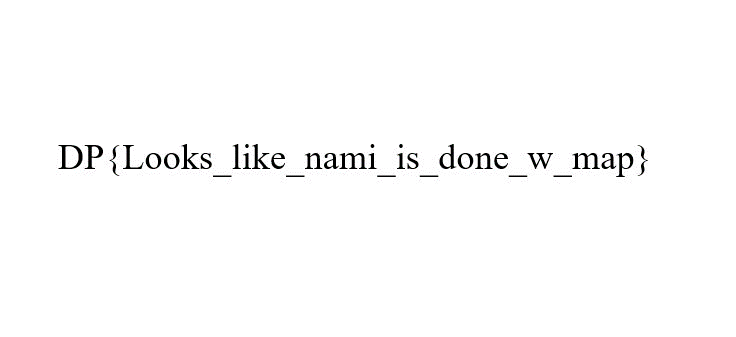
* Flag 14: DF{Welcome\_to\_alabasta}

FLAG 15:

Upon opening the image file “things\_need\_to\_be\_sharpened.png”, we see an image with a blurred out text:



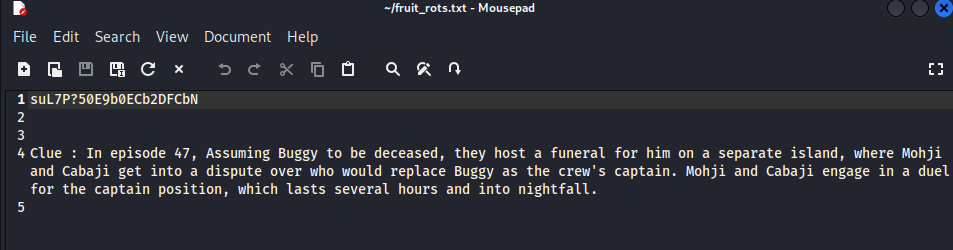
This can be rectified online using some tools:



* Flag 15: DP{Looks\_like\_nami\_is\_done\_w\_map}

FLAG 16:

The other text file is “fruit\_rots.txt”. Lets download it in kali.



The image seem to have a random text and a clue talking about episode 47 in One Piece. Connecting this clue with the name of the file, we conclude that rot47 has been used upon the given cipher text.



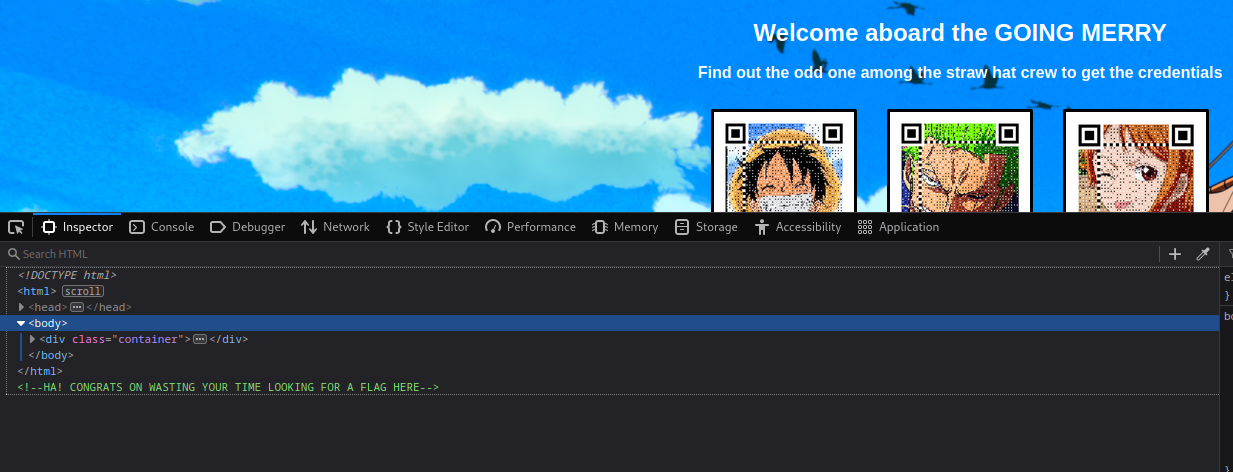
Flag 16: DF{f!nd\_th3\_tr3asur3}

This implies though we checked through all the files and directories, there’s still a flag left, which is the treasure i.e., ONE PIECE!!

FLAG 17:

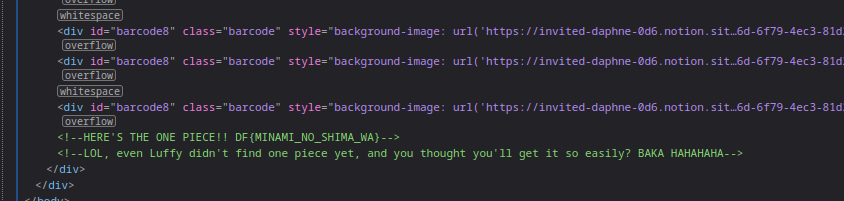
As said, we looked through all the files and folders. So where could the final flag be?

Guess what?! We forgot to inspect the html page!!





Oops, looks like the flag isn’t here. Or, maybe what if it’s there? Let’s go a little deeper!



Oh!! Looks like we found the One Piece! Wait a minute… this isn’t one piece! Hmm, at least we got to find the flag!

* Flag 17: DF{MINAMI\_NO\_SHIMA\_WA}

CONCLUSION

In conclusion, the "Pirate's Quest" black box challenge, as outlined in the abstract, promises an exhilarating fusion of cybersecurity and "One Piece" fandom. With 17 flags to capture, participants are in for a thrilling and educational adventure, where they can refine their skills while immersing themselves in the captivating universe of the anime.