CTF Walkthrough

Group member:

Zhiqi Li, Qiao Jiang, Simin Zhou,

Shuofeng Wang, Yue Chen, Ziyang Lin,

Mou Zhang, Zichen Wang, Jiawei Guo

Zheng Qin

Flag #1

Flag Type: Reverse Engineering

Flag ID: Check /home/seed

System: VM1

Operating System: Ubuntu 12.04 32-bit

Estimated time to completion: 15-20 mins

VM Username: seed

VM Password: dees

Storyline

Hello everyone!

We are the hacker community HateOpera. Our goal is to stop all opera shows in the world. According to our research, a great opera is being planned, and we cannot tolerate its existence. If we can leak the content of the opera in advance, we can prevent its showing. We have sent Jimmy, a professional hacker, to the theater, and he has finished all preparatory work. Now your mission is to grab all the important data you can find from the theater's computers.

Jimmy has left some directions in the theater's computer. In order not to be found by the theater's staff, the first direction is hidden in a binary file. He believes that you can find out the hidden message. Now, try to figure out the direction hidden in the binary file!

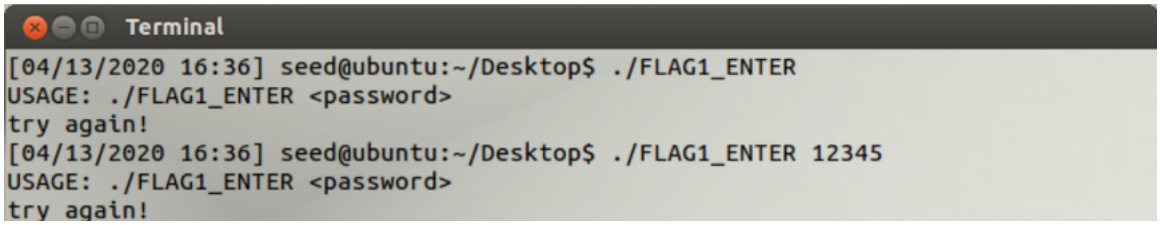
Walkthrough

This simple challenge is related to reverse engineering techniques. Reverse engineering techniques are fundamental to security assessments as they assist with locating vulnerabilities, but more importantly they provide a foundation for understanding why they even occur. For external security assessments, it additionally offers a method to achieve design comprehension.

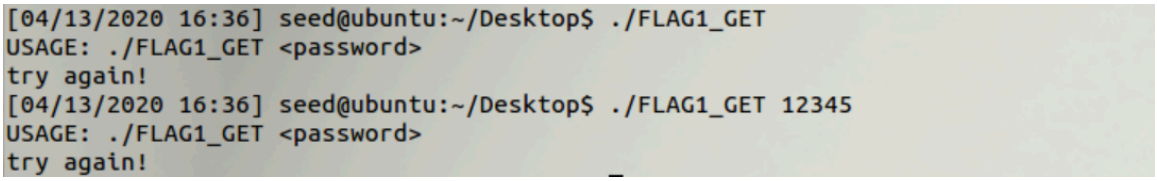
You will be provided two executable files to figure out the content of the flag. You may need to find out the password from FLAG1_ENTER file and use this password to get the content of flag1 in FLAG1_GET.

Here are the details about how to get the flag1.

1. Execute file FLAG1_ENTER in terminal with command: "./FLAG1_ENTER <password>". You need to figure out the format of the correct command. Once you tried "./FLAG1_ENTER", you may get a hint "USAGE", telling you that there should be a password followed behind.

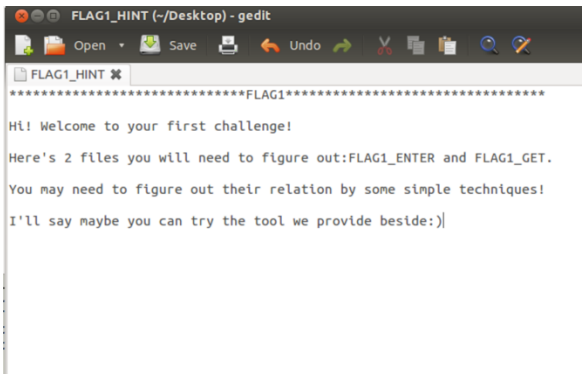


You may also try to execute file FLAG1_GET and it will give you the same result. This also means you need to find out the password first.

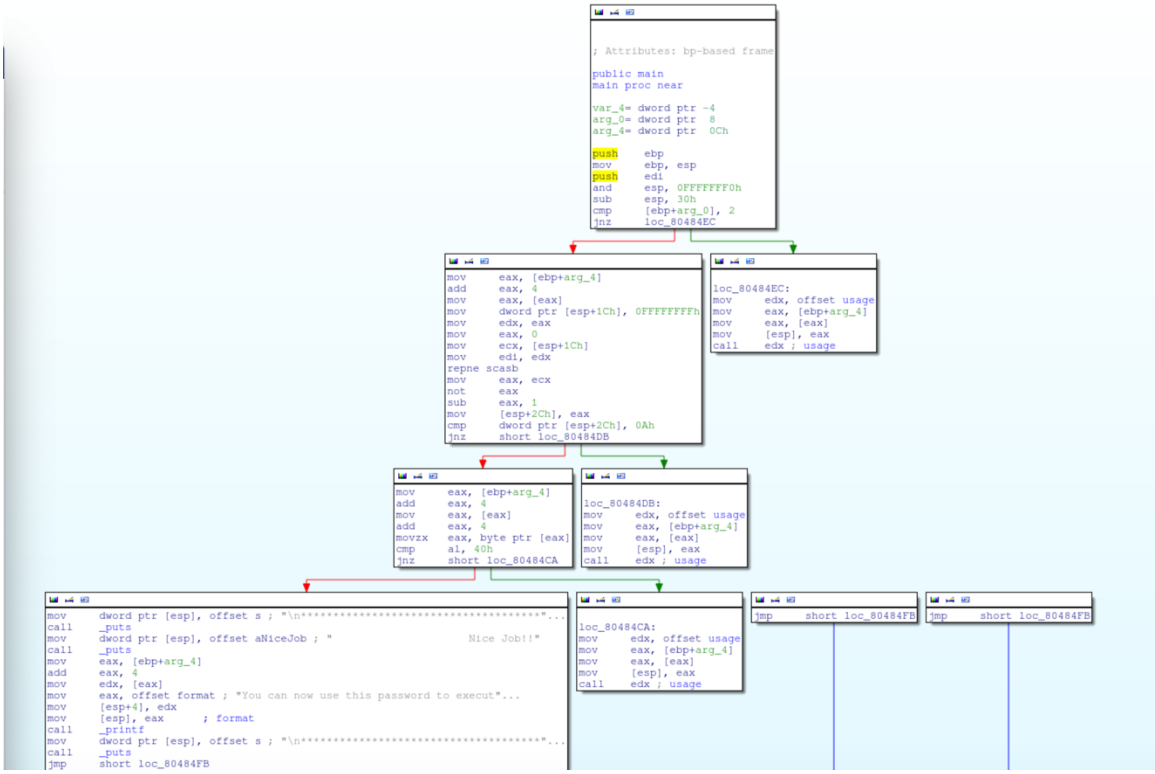


2. Figure out the <password> by using technique such as source code analysis.

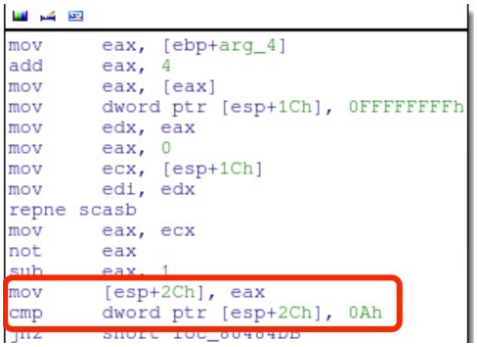
To find out the <password>, you need to look into the file we provided. File FLAG1_GET is a file with execute only permission, there's no way to read into it. So, you need to start with file FLAG1_ENTER. It is an executable and readable ELF file and you can use reverse engineering techniques to read its source code. We provide you an analysis tool in desktop, as mentioned in FLAG1_HINT.



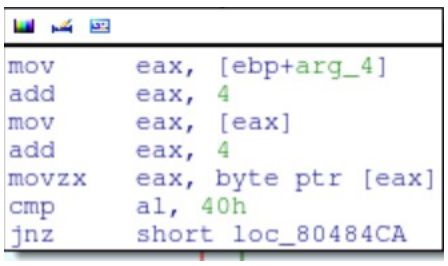
This tool will show you clear logical conditions in the code and make the source code more readable to you. Looking into its graph, you may figure out the meaning of essential assembly code in this file.



As you can see from the graph, the condition of getting “Nice Job”, i.e., the correct password format depends on the sections shown as below:



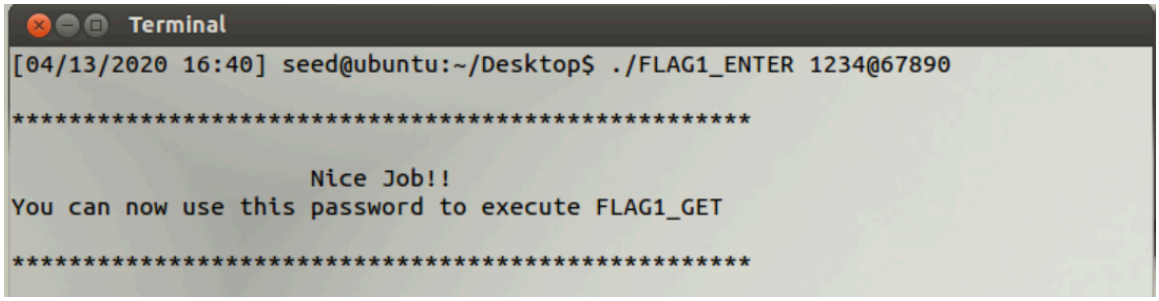
Firstly, the input parameter should be 10 characters long. The assembly code shows that the 11th parameter should be “0Ah”, which represents “newline”.



And then, the 5th character in the input parameter should be “@”. The assembly code shows that if al equals 40h, which represent “@”, then the program will jump to “Nice Job” section. Looking at the above code, we will see “ebp=0c”, which is 12. And in this part, ebp adds 8 and becomes 20, which represent 5th character in the parameter. Therefore, you can say if the 5th character in the parameter is “@”, the program will print “Nice Job”.

As of this step, you have successfully obtained the correct password format: 10 character long and with a “@” as the 5th character.

3. Then you can use the correct password, such as 1234@67890 to execute FLAG1_ENTER:



And then enter FLAG1_GET to get the flag!

Flag #2

Flag Type: Non-Advanced

Flag ID: VM2's PWD:bestctf01

System: VM1

Operating System: Ubuntu 12.04 32-bit

Estimated time to completion: 15-20 mins

VM Username: seed

VM Password: dees

Storyline

After you get the hidden message from the binary file, you know that Jimmy left something in the /home/seed. From our research, we are sure that there is something important in the root directory of VM1. Maybe the things left by Jimmy can be used to grab the secret about the opera. Try to figure it out!

Walkthrough

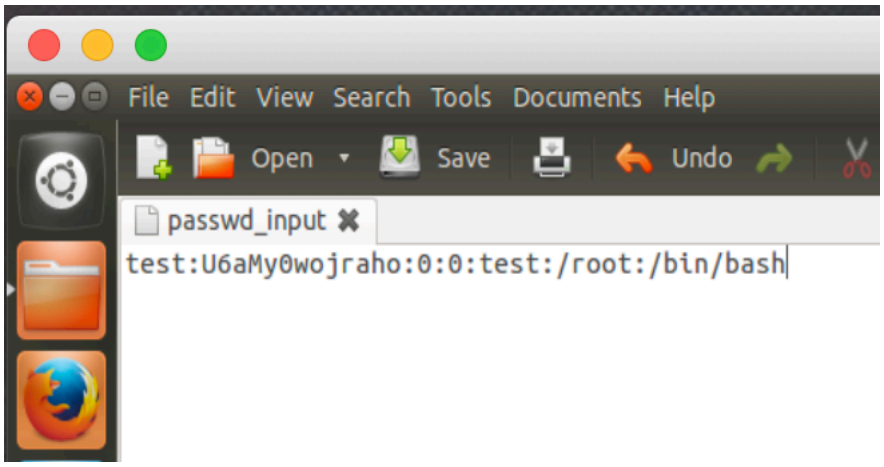
Race condition in software occurs when two component threads of execution access a shared resource in a way that unintentionally produces different results depending on the sequence or timing of the processes or threads.

First of all, we provide a program which aims at writing something into /tmp/XYZ. The system will check if the real user has the write permission to the file before opening the file. Therefore, the system will check the real user id in the process of access(). After passing this check, open() system call will be executed, which is another check. But the difference is that in this check, it only checks the effective user ID.

Here is what we will do. Before running the privileged program, we create a regular file XYZ in the /tmp directory, which is linked to another our own file. Since this is our own file, we will pass the access() check. Right after this check and before the program reaches open(), we quickly change "/tmp/XYZ" to a symbolic link pointing to "/etc/passwd". When the program

gets to open(), it will actually open the password file. Since the open() system call only checks the effective user ID, which is root, it will be able to open the password file for write.

We provided two preset programs to finish all things above. What you have to do is to decide the content which will be inserted into /etc/passwd and can help you get the root privilege. You have to use the given mysterious code to achieve this goal. Observe the content in the password file, you can know what you should add into it.



Here is the example of input. By adding this entry into /etc/passwd file. You can get a new user called test whose password is empty. Then execute the following codes in the directory of /home/seed to use the two provided programs:

After the program shows “stop”, you can execute “su test” and press enter to get root privilege. In this way, you can get the FLAG2 content in the root directory.


Seed 12.04 [Running]
Terminal
[04/15/2020 15:36] root@ubuntu:/home/seed# cd /
[04/15/2020 15:36] root@ubuntu:/# cat FLAG2
FLAG2{VM2's PWD:bestctf01}
[04/15/2020 15:36] root@ubuntu:/#

Flag #3

Flag Type: Cryptology

Flag ID: VM3's PWD:brilliantgroup

System: VM2

Operating System: Ubuntu 15.10 64-bit

Estimated time to completion: 10-15 mins

VM Username: student

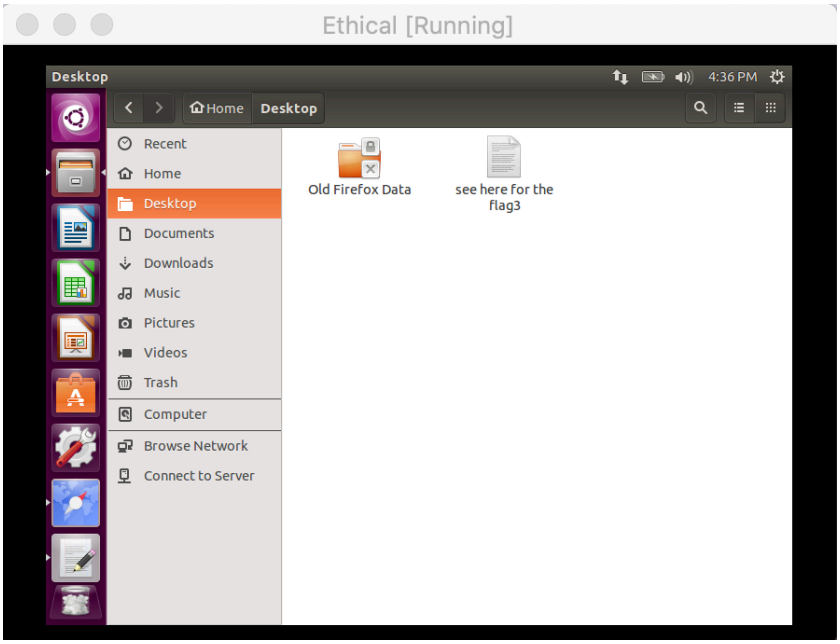
VM Password: bestctf01

Storyline

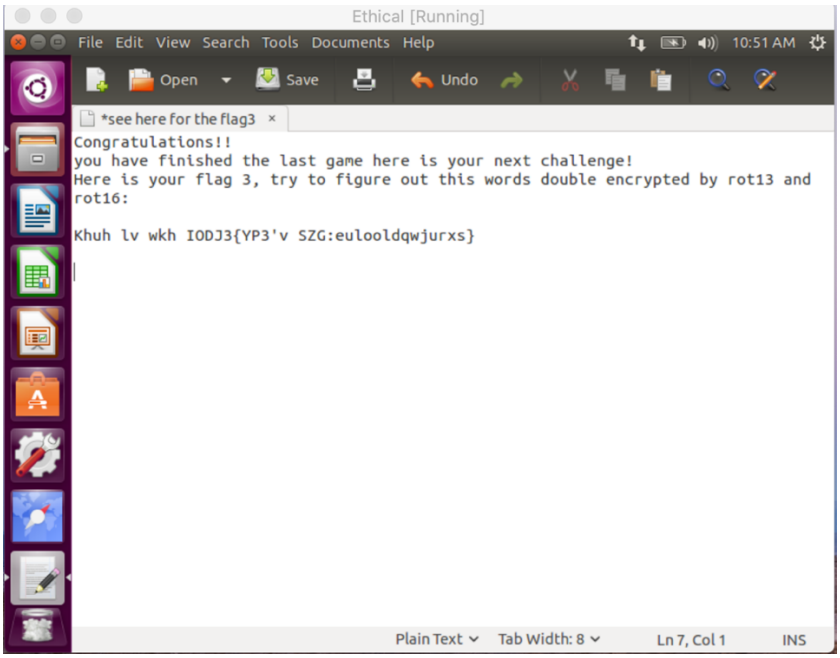
What a pity! The important file is not the secret about the opera itself. It is just a password to another computer. Since Jimmy brings you here, there must be something useful. Find out something valuable!

Walkthrough

Here is the flag to start.



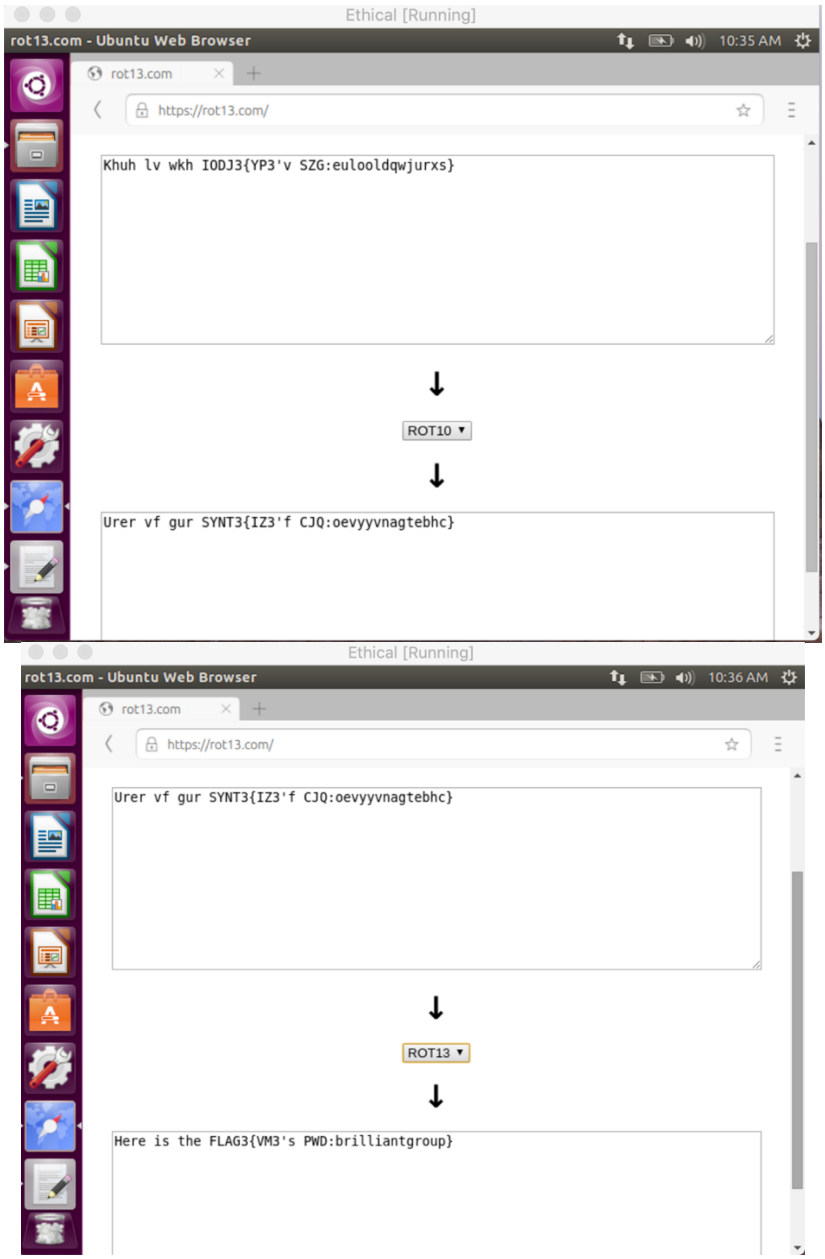
Here is the flag file, you need to convert "YP3'v SZG:eulooldqwjurxs" to passwords.



As I said we use rot13 first and rot16 next, so if you want to decode, you just need to decode rot16 first and rot13 next.

The principles behind the solution: ROT13 is a special case of Caesar cipher which was developed in ancient Rome. Because there are 26 letters (2×13) in basic Latin alphabet ROT13 is its own inverse; that is, to undo ROT13, the same algorithm is applied, so the same action can be used for encoding and decoding. Needless to say, to reverse ROT16, we just need to convert it back by running ROT(26-16) which is rot10, because there are 26 alphabets.

Therefore, what we have to do is to apply ROT(10) first and then ROT(13).



In this way, we can get the flag 3 successfully.

Flag #4

Flag Type: Non-Advanced

Flag ID: 5368756f66656e67

System: VM3

Operating System: Ubuntu 16.04 64-bit

Estimated time to completion: 15-20 mins

VM Username: mou

VM Password: brilliantgroup

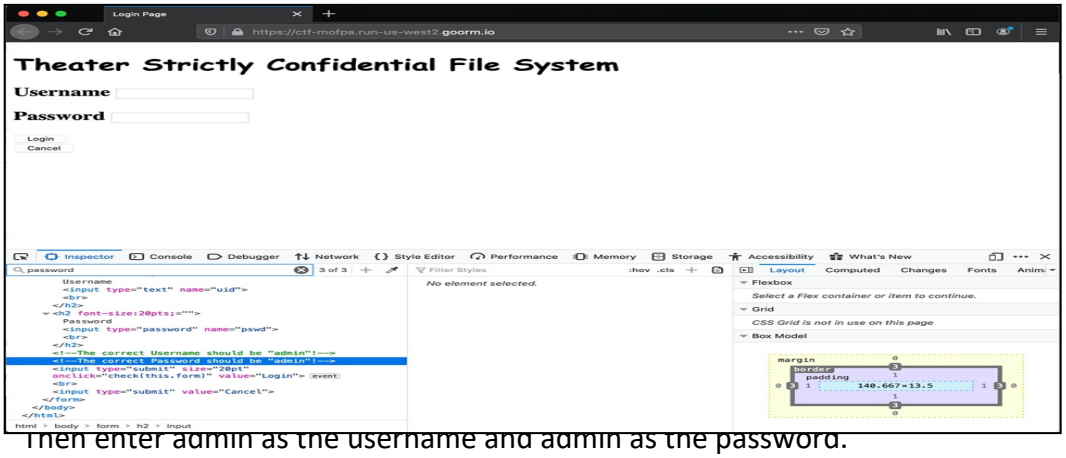
Storyline

Again! The hidden message is a password to another machine again! You are likely to consider that Jimmy is kidding you. But this time, if you observe carefully, you can find that this machine works as the server of the file system of the whole theater. Therefore, your work is to log in this file system and then find out something useful.

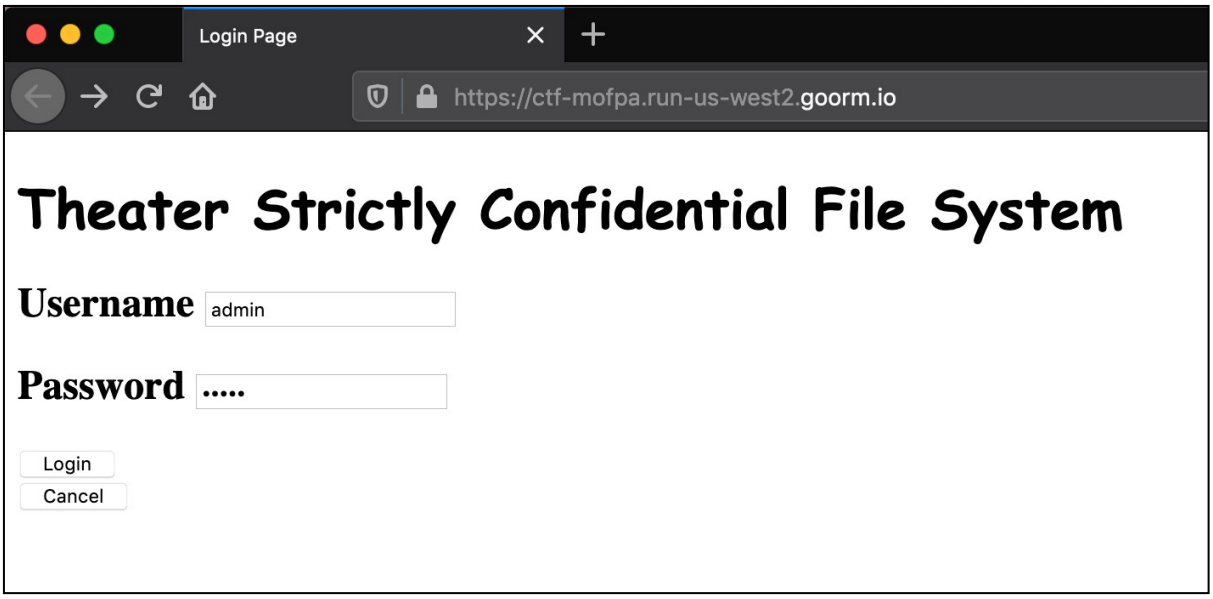
Walkthrough

If the server is unavailable, please contact swang246@jhu.edu.

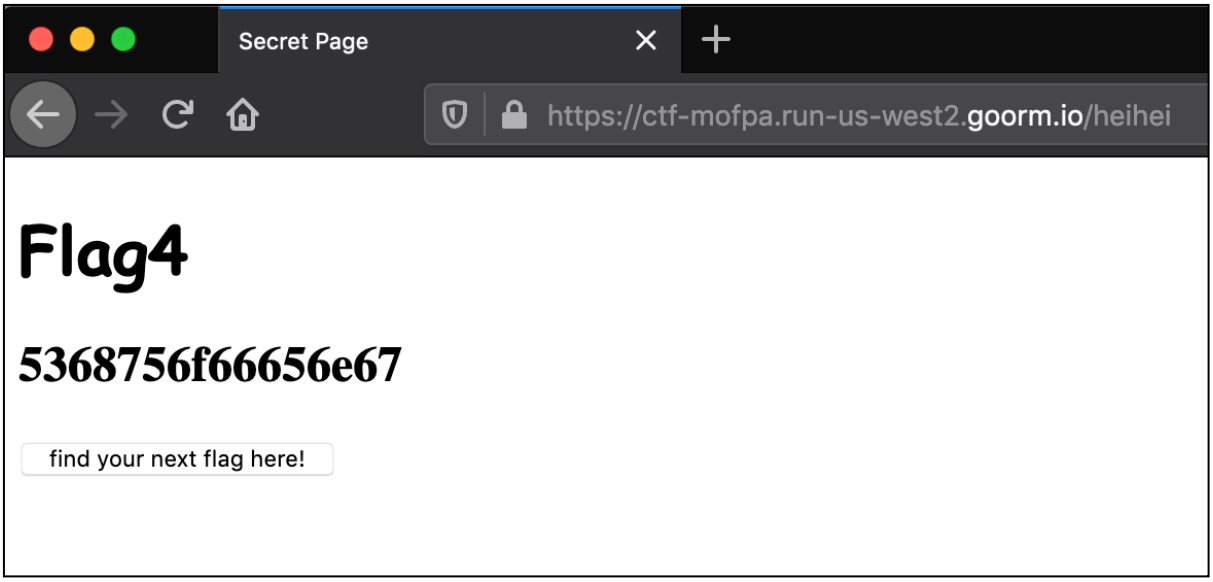
1. Check the source code of the web page, then you will find the hint in the comments of the html code, which says that “<!-- The correct Username should be "admin"! -->” and “<!-- The correct Password should be "admin"! -->”. You can also press Control + F to search for ‘username’ or ‘Password’ to find those comments.



2. Then enter admin as the username and admin as the password.



3. Then you will get the flag. Congratulations!



4. By clicking the link, you will jump to the next problem. Please do not discard this flag. It will be useful for the next flag.

Flag #5

Flag Type: Non-Advanced

Flag ID: hellothisisamanda

System: VM3

Operating System: Ubuntu 16.04 64-bit

Estimated time to completion: 15-20 mins

VM Username: mou

VM Password: brilliantgroup

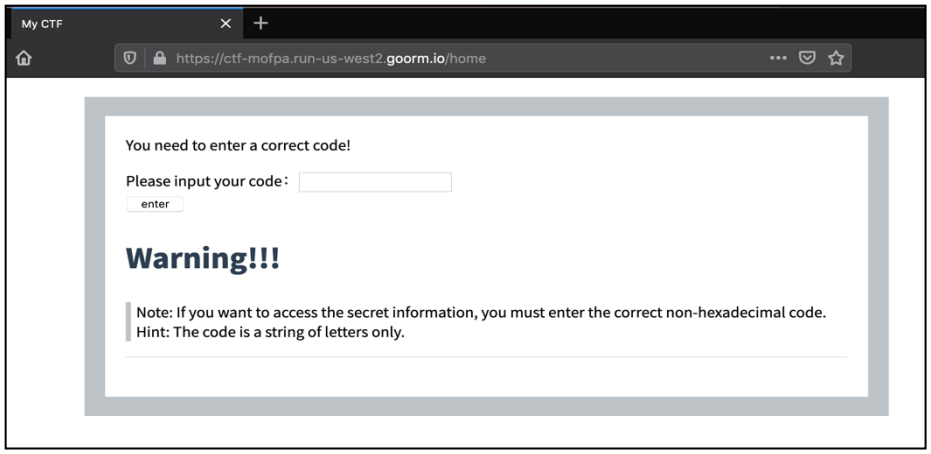
Storyline

OK. Now you succeed in logging into the theater's file system. And you find that you have to input a code to grab the information. Consider if you can get the code from the webpage. And what does the message Jimmy left mean? According to your knowledge, Jimmy is a lazy guy. He never does something meaningless. Try to understand what he wants to tell you!

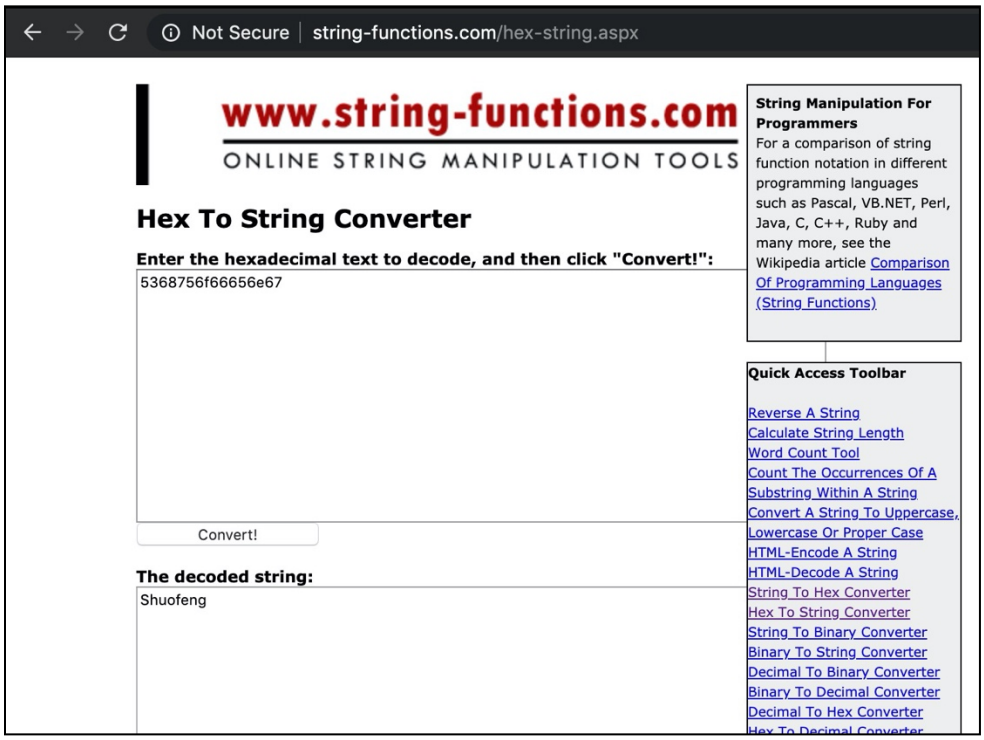
Walkthrough

If the server is unavailable, please contact swang246@jhu.edu.

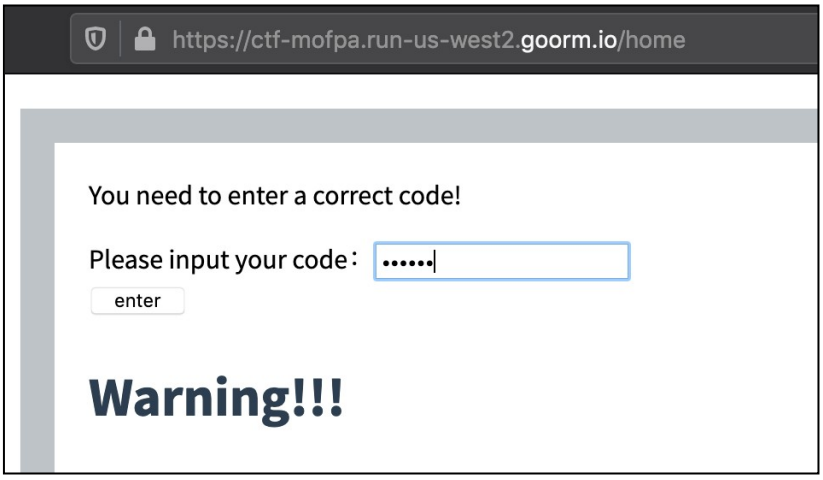
Flag 4 is a string encoded in some format, such as 2, 8, or hex. You need to decode flag 4 correctly to get the corresponding string. According to the intelligence, this string will work as the password to fetch the secret of the theater.



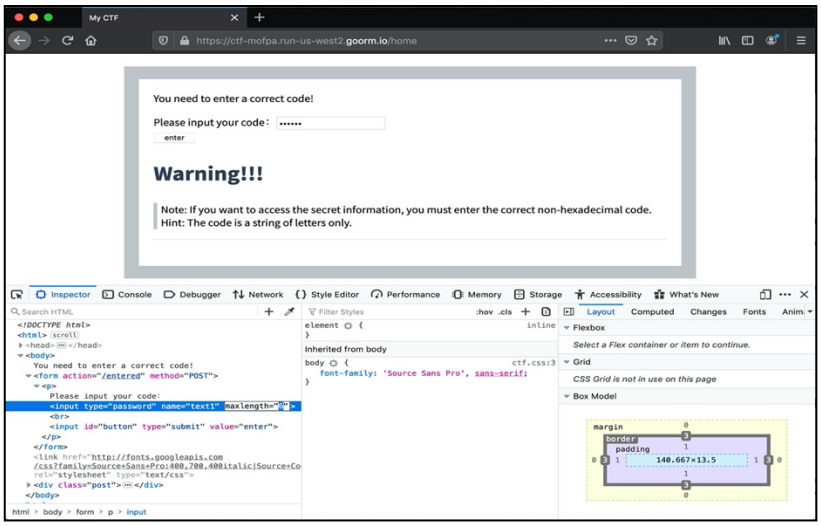
1. According to the hint of the webpage, the code is non-hex and is a string of letters only. Recall that flag 4 (5368756f66656e67) is a string of numbers and letters, which meets the characteristic of a hex string. Then utilize a hex to string converter (e.g. http://string-functions.com/hex-string.aspx) to try to decode the flag.



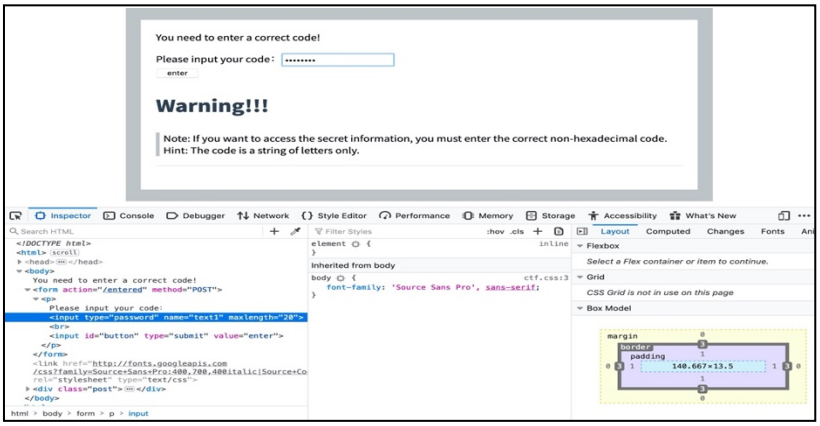
2. Try to enter the string Shuofeng as the code in the webpage. However, you will find that the textbox sets limitations on the input length, i.e. only six letters can be entered here.



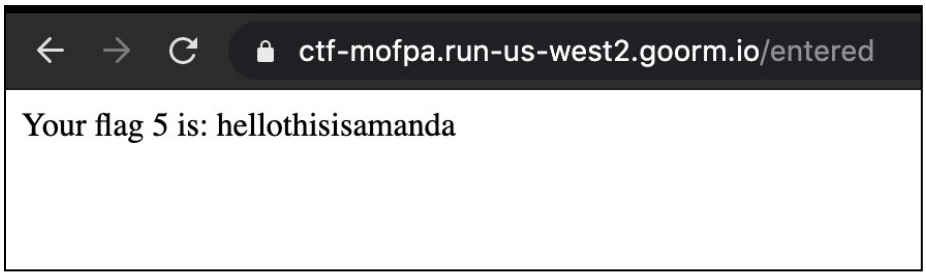
3. Check the source code, you can find that the max length is set to “6”. Then double click the number and change it to any larger number so that you can enter enough letters in the textbox.



4. Change it to “20” and then you will be able to enter the string Shuofeng.



5. After clicking on ‘enter’, you will get the flag 5. Congratulations.



Flag #6

Flag Type: Non-Advanced

Flag ID: vpl0ad_Ctf_6

System: VM3

Operating System: Ubuntu 16.04 64-bit

Estimated time to completion: 15-20 mins

VM Username: mou

VM Password: brilliantgroup

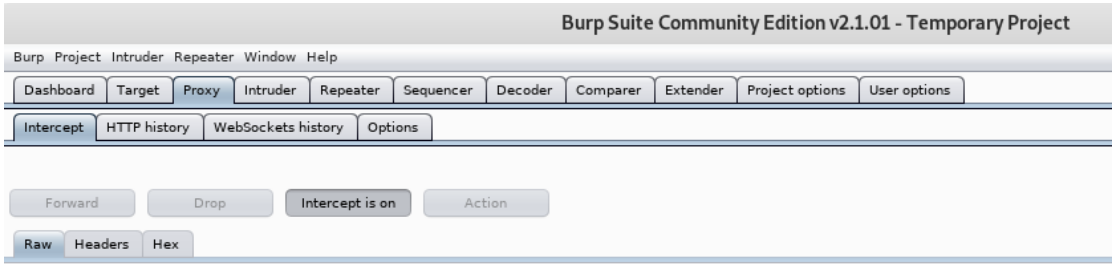
Storyline

From the theater's file system. You know that the famous actress, Amanda Chen, is going to act in this opera. Leaking this information out can affect the publicity of this opera to some extent. In order to get more information from this server, you'd better check all other websites deployed on this server.

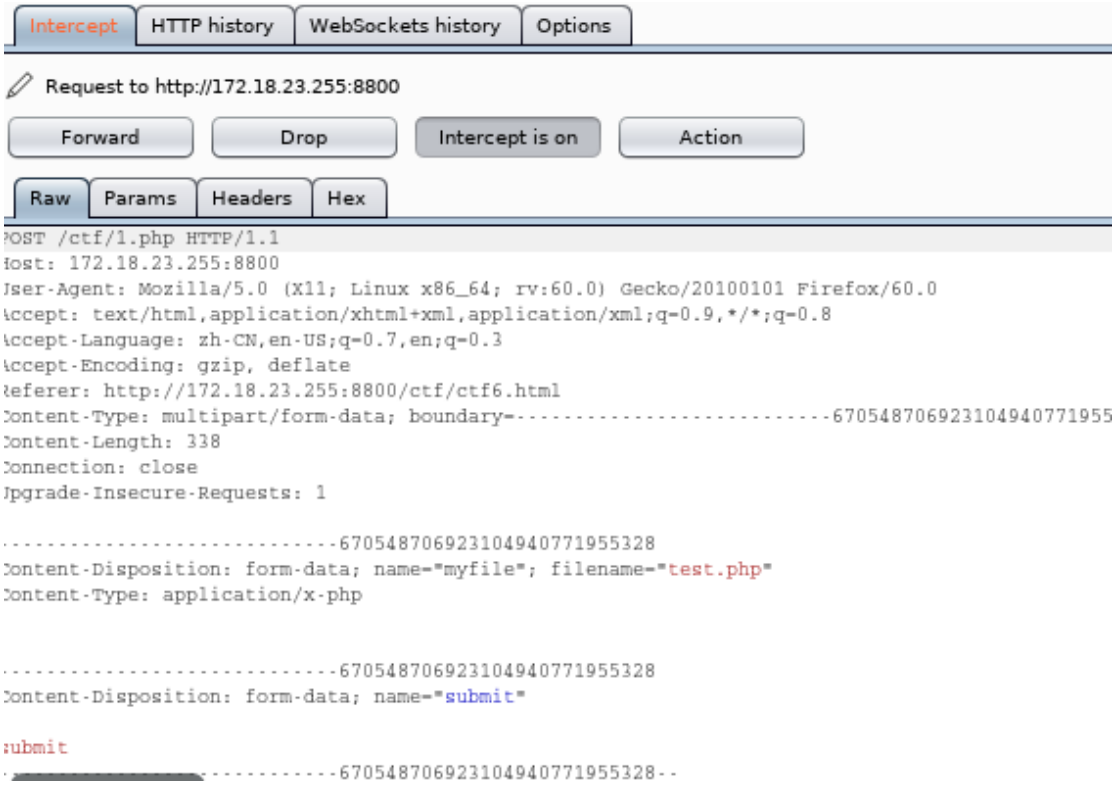
Walkthrough

1. Input vm3_ip_address:8800/ctf6.html, then the web browser will show like this:

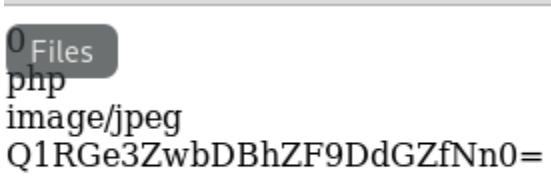
2. Create a php file.
3. Open the Burp Suite and turn on the proxy, like this:



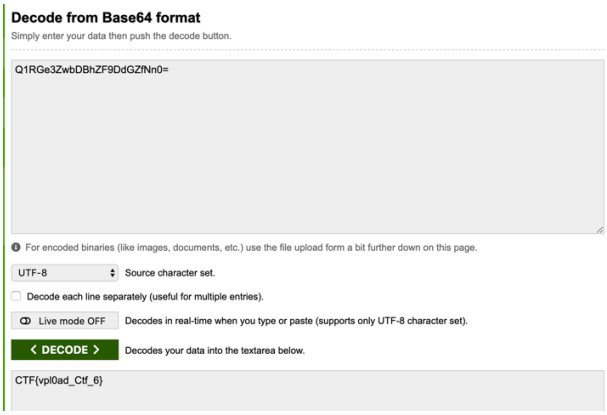
4. Upload the file, the Burp Suite will show:



5. Change the content-type from application/x-php to image/jpeg.



6. Use base64 to decode the text, you can get the flag.



Flag #7

Flag Type: Non-Advanced

Flag ID: ThisIsThefinal

System: VM3

Operating System: Ubuntu 16.04 64-bit

Estimated time to completion: 10-15 mins

VM Username: mou

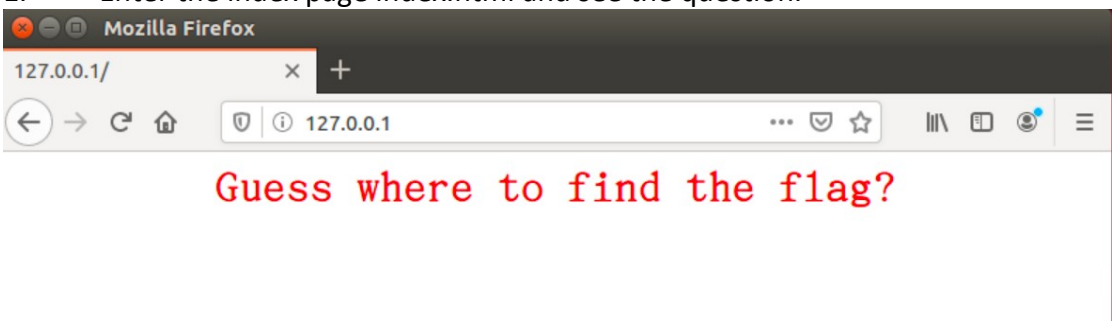
VM Password: brilliantgroup

Storyline

You've got another message in the theater's file system! Since you have no idea about whether this message is meaningful or not, what you should do is to check all websites you can find in this server. Maybe there is another message Jimmy left to you.

Walkthrough

1. Enter the index page index.html and see the question.

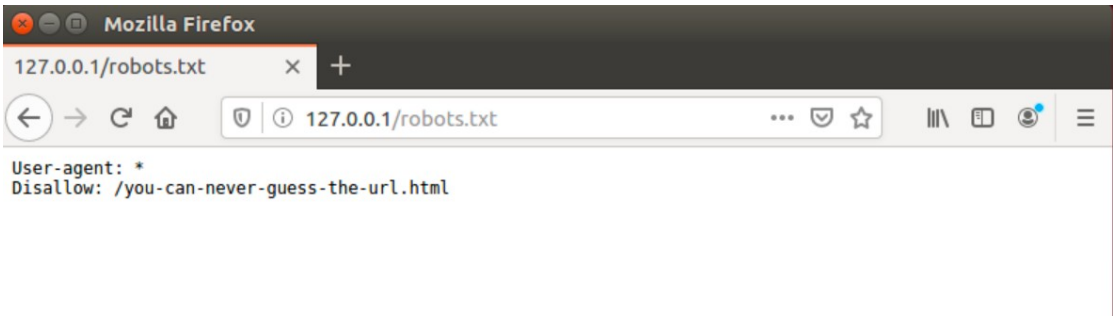


2. Guessing where the flag is.

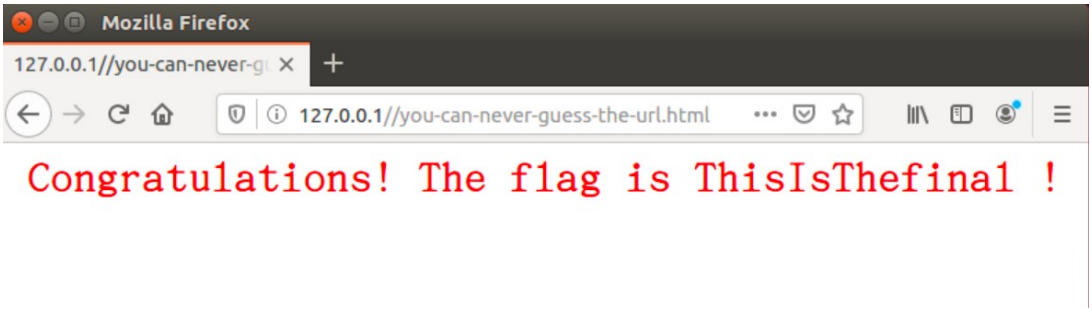
Hint1: It's on another page in the website.

Hint2: Think about robots.txt, which is a common page used to defend web crawlers.

3. Check the robot.txt and find the target webpage URL.



4. Get into the target page and find the flag.



Flag #8

Flag Type: Cryptology

Flag ID: hackhackglhfglhf

System: VM4

Operating System: Windows 10 64-bit

Estimated time to completion: 10-15 mins

VM Username: PWD is the last flag

VM Password: ThisIsThefinal

Storyline

Jimmy tells us this machine is the last one we have to attack. After entering this system, we find that we have the chance to break the electrical system of the theater. If we can cause an outrage during the opera show, the fame of this opera and this theater would be horrible. This is a good idea. But the problem is we have to get the password to enter the switching room, and it is protected by an encrypted string. If you can decrypt this password, we can make this opera much worse. Let's go! Let the audience know the feeling of watching opera in darkness!

Walkthrough

The cipher we used here is block cipher, it's pretty easy to explain. Every consecutive two letters become a block and then were turned into one letter of the cipher text.

So, we only need to take each possible combination of cipher text to brute-force the password. What's more, there are repeating parts in the cipher text, so just assume that the corresponding part of the plain text also looks in the same way.

```python
import subprocess

def possible_list(c):
    l = []
    target = ord(c)-97
    for i in range(26):
        if i <= target:
            j = target-i
        else:
            j = target+26-i
        l.append(chr(i+97)+chr(j+97))
    return l

psw = []
for i1 in possible_list("h"):
    for i2 in possible_list("m"):
        for i3 in possible_list("r"):
            for i4 in possible_list("m"):
                msg = i1+i2+i1+i2+i3+i4+i3+i4+"\n"
                psw.append(msg)

n = 0
while True:
    p = subprocess.Popen("password.exe",stdin=subprocess.PIPE,stdout=subprocess.PIPE)
    for i in range(962*n,962*(n+1)):
        p.stdin.write(psw[i].encode())
        print(i)
    result = p.communicate()[0].decode()
    print(result)
    c = result.count("Wrong")
    if c < 962:
        print("The password is: ",psw[962*n+c])
        break
    else:
        n += 1
```

Here is one of the solution codes, we used library subprocess since we have to interact with the executable file. Then we iterate every possible combination to find out the correct solution. The following screenshot is the output of the code. It's not hard to get the flag.

```
Please Enter Again:

###################

     Bingo!!!!!!

###################

Congratulations!!!!
You got the key having access to the next flag.
Good luck!!!


449
124547 hackhackglhfglhf
```

Flag #9

Flag Type: Non-Advanced

Flag ID: iloveopera

System: VM4

Operating System: Windows 10 64-bit

Estimated time to completion: 20-25 mins

VM Username: PWD is the last flag

VM Password: ThisIsThefinal

Storyline

During the process of finding valuable data in the machine, Jimmy found a compressed file called BGM. He thinks this file may contain a lot of background music of the opera. If we can get them and distribute them on the Internet, many people will not go to the theater to watch the opera. However, the format of the files in the rar file is not a common music format like mp3, wmv, etc. Try to figure it out and get the message hidden in the music!

Walkthrough

1. According to the hint, we can know that the .opera file is encrypted by the simplest XOR method. It's easy to write a program and decrypt it. However, we don't know the number the file is XOR with. Luckily, there are only 256 possible answer so that we can try to get all the 256 files and check them.

ans.py

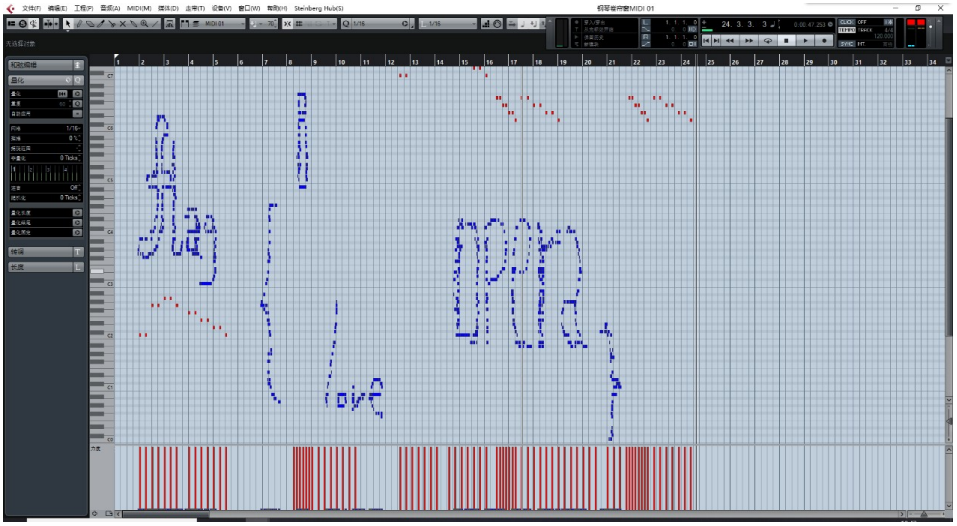
```
fin = open("flag.opera", "rb")
fin_data = fin.read()

for i in range(1, 0xff + 1):
    fout = open("flag.{}".format(i), "wb")
    data = bytearray(fin_data)
    for j, item in enumerate(data):
        data[j] = item ^ i
    fout.write(data)
    fout.close()
```

2. The best way to identify file attributes is to use the FILE command. So, we can use it and find out that flag.102 is different from other files. It's a MIDI file and recalling that this is the BGM of the opera, we can assume that this is the correct file.

```
flag.1:     data
flag.10:    data
flag.100:   data
flag.101:   data
flag.102:   Standard MIDI data (format 1) using 3 tracks at 1/480
flag.103:   unicos (cray) executable
flag.104:   data
flag.105:   data
flag.106:   data
flag.107:   data
```

3. Open it we can find out it's "Twinkle, Twinkle, Little Star" and we need to find the key inside it.

4. Since the flag is in the MIDI file, we can try to open it in a music editing software, and we can find the flag hiding in the audio track.



The flag is iloveopera.

Flag #10

Flag Type: Non-Advanced

Flag ID: Hee tian_www.hetianl ab.com

System: VM4

Operating System: Windows 10 64-bit

Estimated time to completion: 15-20 mins

VM Username: PWD is the last flag

VM Password: ThisIsThefinal

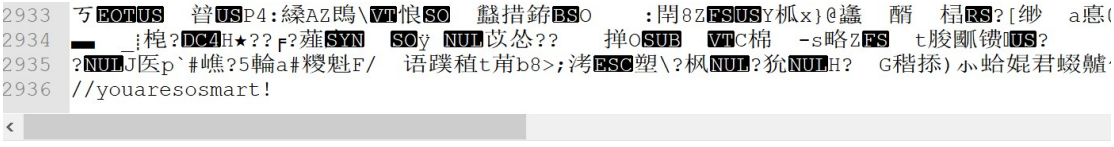
Storyline

Jimmy find a suspicious picture in this machine. According to the folder name, all files in this folder should be about the actor list of the opera. However, in this folder, he only found a picture. Jimmy guessed that this is a picture used to hide important information. Therefore, he hopes that you can help him extract the secret data from this picture. If you can achieve this goal, HateOpera can thoroughly destroy this opera show.

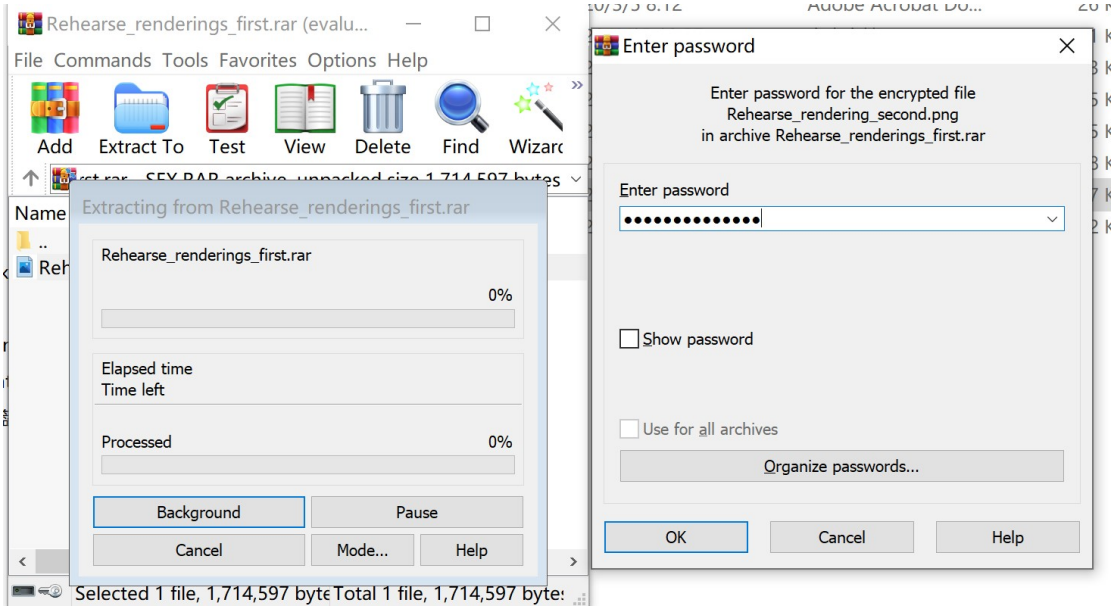
Walkthrough

This is a question about steganography

1. Open the jpeg file in Notepad++, you can find a hidden readable char at the end. It is a password.



2. Change the filetype from jpeg to rar. Extract docs from the rar file with the password from step 1.



3. I hide the text of the list of actors in another pic by LSB algorithm. You can write a script to extract the bin file and transform it into text. I offered the script I have written.

```python
from PIL import Image

def mod(x,y):
    return x%y;

def toasc(strr):
    return int(strr, 2)

def func(le,str1,str2):
    a=""
    b=""
    im = Image.open(str1)
    lenth = le*8
    width = im.size[0]
    height = im.size[1]
    count = 0
    for h in range(0, height):
        for w in range(0, width):
            pixel = im.getpixel((w, h))
            if count%3==0:
                count+=1
                b=b+str((mod(int(pixel[0]),2)))
                if count ==lenth:
                    break
            if count%3==1:
                count+=1
                b=b+str((mod(int(pixel[1]),2)))
                if count ==lenth:
                    break
            if count%3==2:
                count+=1
                b=b+str((mod(int(pixel[2]),2)))
                if count ==lenth:
                    break
        if count == lenth:
            break
    with open(str2,"wb") as f:
        for i in range(0,len(b),8):
            stra = toasc(b[i:i+8])
            f.write(chr(stra).encode())
            stra =""
    f.closed
le = 100
new = r"C:\Users\qin\Desktop\ethic\CTF\heetian_LSB.png"
tiqu = r"C:\Users\qin\Desktop\ethic\CTF\get_flag.txt"

func(le,new,tiqu)
```

Or you can just find there are hidden information by changing Bit planes setting by StegSlove like pic shown below.