0xGame2023 Week4 Writeup

> 最后一周的题都非常难，本noob只会做几道比较简单的题o(╥﹏╥)o。。。其他的题都是花了大量时间（真的很多很多）但都一无所获。。。o(╥﹏╥)o问题不大，虽然肯定没法使自己的排名再提升了（差距太大了），但是这个过程中学到的知识和积累的经验都是无价的ヾ(◍°∇°◍)ﾉﾞ。

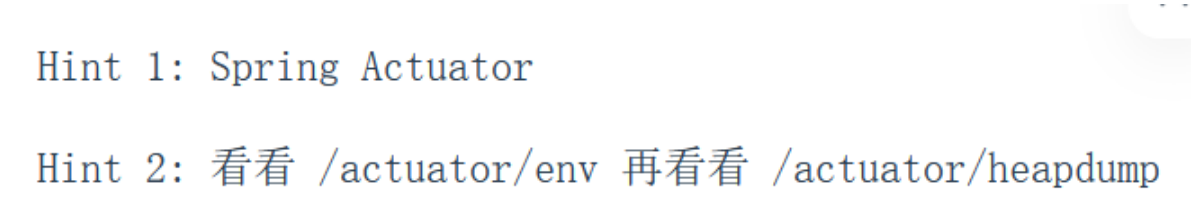
Web

Spring

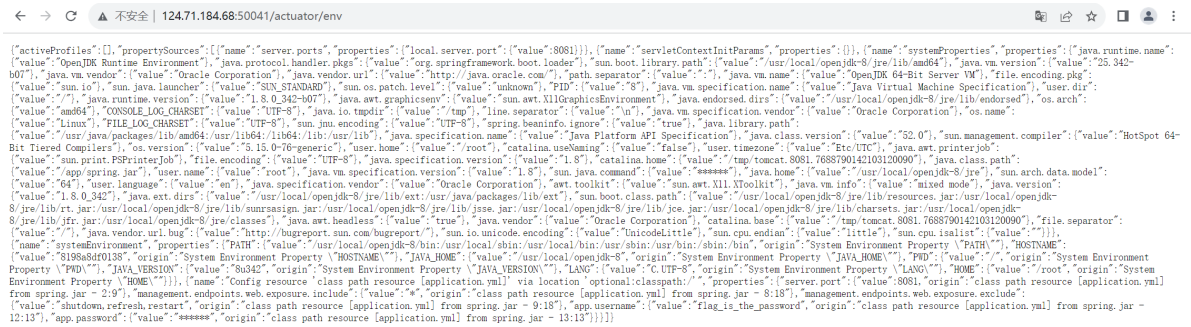
题目

do you know what is actuator?

提示

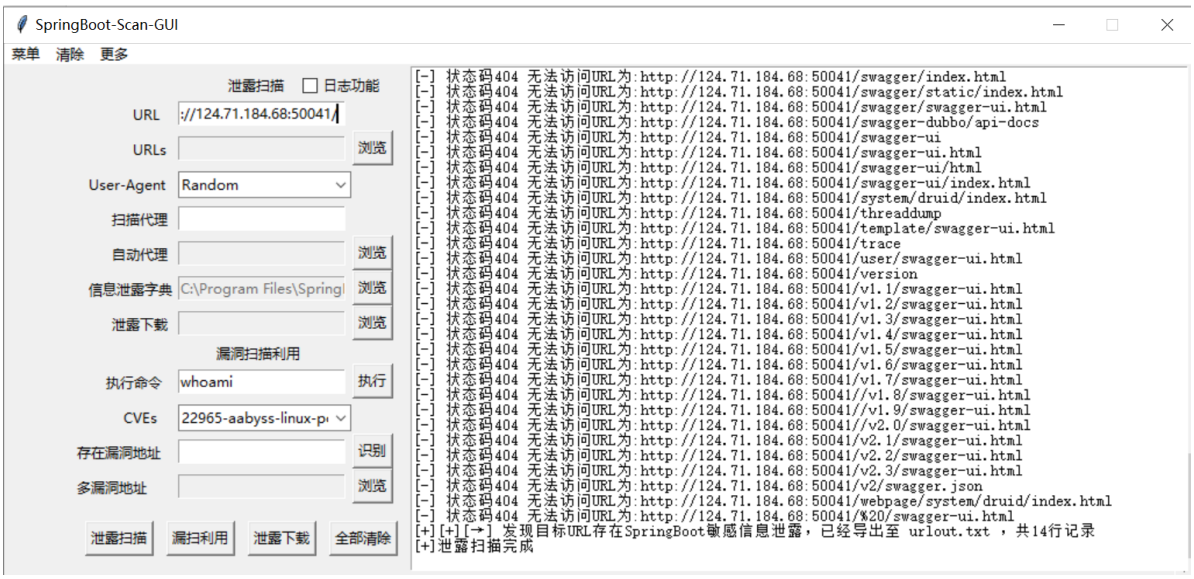


根据提示查看



{"activeProfiles":[],"propertySources":[{"name":"server.ports","properties":{"local.server.port":{"value":8081}}},{"name":"servletContextInitParams","properties":{}},{"name":"systemProperties","properties":{"java.runtime.name":{"value":"OpenJDK Runtime Environment"},"java.protocol.handler.pkgs":{"value":"org.springframework.boot.loader"},"sun.boot.library.path":{"value":"/usr/local/openjdk-8/jre/lib/amd64"},"java.vm.version":{"value":"25.342-b07"},"java.vm.vendor":{"value":"Oracle Corporation"},"java.vendor.url":{"value":"http://java.oracle.com/"},"path.separator":{"value":":"},"java.vm.name":{"value":"OpenJDK 64-Bit Server VM"},"file.encoding.pkg":{"value":"sun.io"},"sun.java.launcher":{"value":"SUN_STANDARD"},"sun.os.patch.level":{"value":"unknown"},"PID":{"value":"8"},"java.vm.specification.name":{"value":"Java Virtual Machine Specification"},"user.dir":{"value":"/"},"java.runtime.version":{"value":"1.8.0_342-b07"},"java.awt.graphicsenv":{"value":"sun.awt.X11GraphicsEnvironment"},"java.endorsed.dirs":{"value":"/usr/local/openjdk-8/jre/lib/endorsed"},"os.arch":{"value":"amd64"},"CONSOLE_LOG_CHARSET":{"value":"UTF-8"},"java.io.tmpdir":{"value":"/tmp"},"line.separator":{"value":"\n"},"java.vm.specification.vendor":{"value":"Oracle Corporation"},"os.name":{"value":"Linux"},"FILE_LOG_CHARSET":{"value":"UTF-8"},"sun.jnu.encoding":{"value":"UTF-8"},"spring.beaninfo.ignore":{"value":"true"},"java.library.path":{"value":"/usr/java/packages/lib/amd64:/usr/lib64:/lib64:/lib:/usr/lib"},"java.specification.name":{"value":"Java Platform API Specification"},"java.class.version":{"value":"52.0"},"sun.management.compiler":{"value":"HotSpot 64-Bit Tiered Compilers"},"os.version":{"value":"5.15.0-76-generic"},"user.home":{"value":"/root"},"catalina.useNaming":{"value":"false"},"user.timezone":{"value":"Etc/UTC"},"java.awt.printerjob":{"value":"sun.print.PSPrinterJob"},"file.encoding":{"value":"UTF-8"},"java.specification.version":{"value":"1.8"},"catalina.home":{"value":"/tmp/tomcat.8081.7688790142103120090"},"java.class.path":{"value":"/app/spring.jar"},"user.name":{"value":"root"},"java.vm.specification.version":{"value":"1.8"},"sun.java.command":{"value":"******"},"java.home":{"value":"/usr/local/openjdk-8/jre"},"sun.arch.data.model":{"value":"64"},"user.language":{"value":"en"},"java.specification.vendor":{"value":"Oracle Corporation"},"awt.toolkit":{"value":"sun.awt.X11.XToolkit"},"java.vm.info":{"value":"mixed mode"},"java.version":{"value":"1.8.0_342"},"java.ext.dirs":{"value":"/usr/local/openjdk-8/jre/lib/ext:/usr/java/packages/lib/ext"},"sun.boot.class.path":{"value":"/usr/local/openjdk-8/jre/lib/resources.jar:/usr/local/openjdk-8/jre/lib/rt.jar:/usr/local/openjdk-8/jre/lib/sunrsasign.jar:/usr/local/openjdk-8/jre/lib/jsse.jar:/usr/local/openjdk-8/jre/lib/jce.jar:/usr/local/openjdk-8/jre/lib/charsets.jar:/usr/local/openjdk-8/jre/lib/jfr.jar:/usr/local/openjdk-8/jre/classes"},"java.awt.headless":{"value":"true"},"java.vendor":{"value":"Oracle Corporation"},"catalina.base":{"value":"/tmp/tomcat.8081.7688790142103120090"},"file.separator":{"value":"/"},"java.vendor.url.bug":{"value":"http://bugreport.sun.com/bugreport/"},"sun.io.unicode.encoding":{"value":"UnicodeLittle"},"sun.cpu.endian":{"value":"little"},"sun.cpu.isalist":{"value":""}}},{"name":"systemEnvironment","properties":{"PATH":{"value":"/usr/local/openjdk-8/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin","origin":"System Environment Property \"PATH\""},"HOSTNAME":{"value":"8198a8df0138","origin":"System Environment Property \"HOSTNAME\""},"JAVA_HOME":{"value":"/usr/local/openjdk-8","origin":"System Environment Property \"JAVA_HOME\""},"PWD":{"value":"/","origin":"System Environment Property \"PWD\""},"JAVA_VERSION":{"value":"8u342","origin":"System Environment Property \"JAVA_VERSION\""},"LANG":{"value":"C.UTF-8","origin":"System Environment Property \"LANG\""},"HOME":{"value":"/root","origin":"System Environment Property \"HOME\""}}},{"name":"Config resource 'class path resource [application.yml]' via location 'optional:classpath:/'","properties":{"server.port":{"value":8081,"origin":"class path resource [application.yml] from spring.jar - 2:9"},"management.endpoints.web.exposure.include":{"value":"*","origin":"class path resource [application.yml] from spring.jar - 8:18"},"management.endpoints.web.exposure.exclude":{"value":"shutdown,refresh,restart","origin":"class path resource [application.yml] from spring.jar - 9:18"},"app.username":{"value":"flag_is_the_password","origin":"class path resource [application.yml] from spring.jar - 12:13"},"app.password":{"value":"******","origin":"class path resource [application.yml] from spring.jar - 13:13"}}}]}

或者扫描找到发生泄露的URL再查看

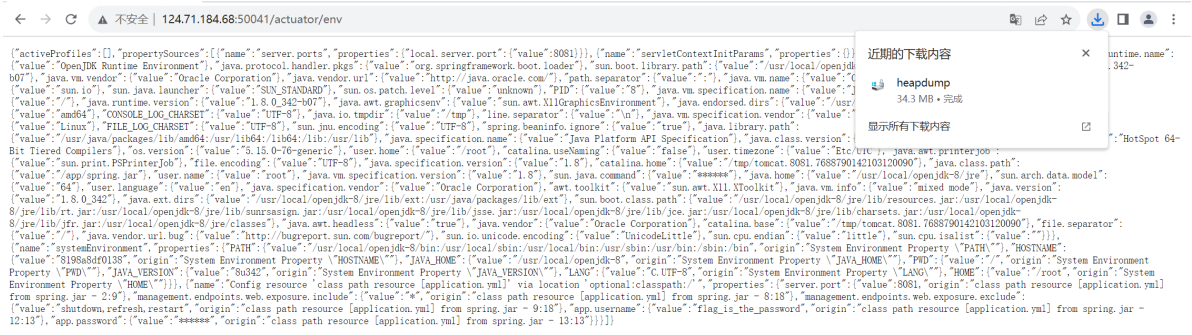


urlout - 记事本

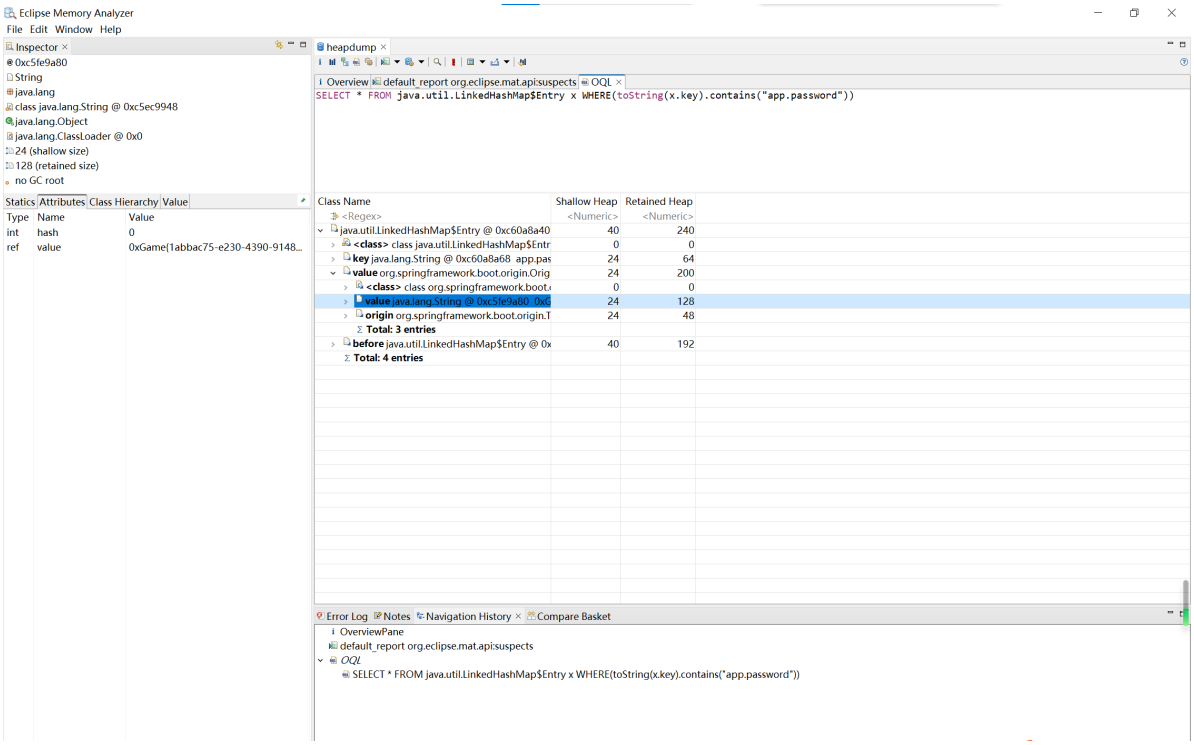
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

http://124.71.184.68:50041/actuator
http://124.71.184.68:50041/actuator/beans
http://124.71.184.68:50041/actuator/caches
http://124.71.184.68:50041/actuator/conditions
http://124.71.184.68:50041/actuator/configprops
http://124.71.184.68:50041/actuator/env
http://124.71.184.68:50041/actuator/health
http://124.71.184.68:50041/actuator/info
http://124.71.184.68:50041/actuator/loggers
http://124.71.184.68:50041/actuator/metrics
http://124.71.184.68:50041/actuator/mappings
http://124.71.184.68:50041/actuator/scheduledtasks
http://124.71.184.68:50041/actuator/scheduledtasks
http://124.71.184.68:50041/actuator/threaddump

从env中可以看出password即为flag，访问heapdump下载内存镜像



使用Eclipse Memory Analyzer进行OQL查询，查得password（即flag）



Crypto

Normal ECC

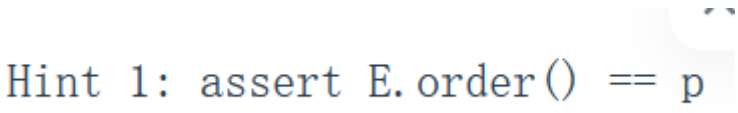
题目

这周的ECC由于数字较大，无法用上周的Pohlig-Hellman方法求解，故此需换用其他方法

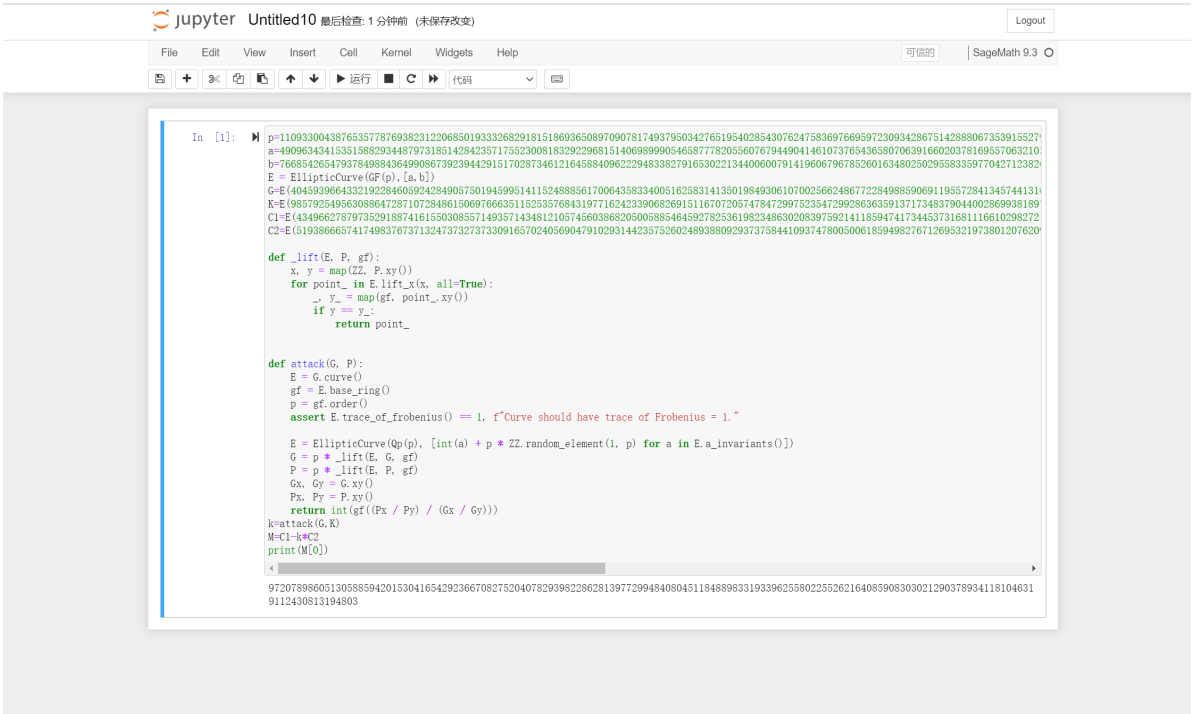
```
from Crypto.Util.number import getPrime
from Crypto.Cipher import AES
from random import getrandbits
from hashlib import md5
from secret import flag,M

def MD5(m):return md5(str(m).encode()).hexdigest()
assert '0xGame{'+MD5(M[0])+'}' == flag
p = 
110933004387653577876938231220685019333268291815186936508970907817493795034276519540285430762475836976695972309342867514288806735391552792323043011239314191
a = 
490963434153515882934487973185142842357175523008183292296815140698999054658777820556076794490414610737654365807063916602037816955706321036900113929329671
b = 
766854265479378498843649908673923944291517028734612164588409622294833827916530221344006007914196067967852601634802502955833597704271238261119799500231646
assert p>a
assert p>b
E = EllipticCurve(GF(p),[a,b])
assert E.order() == p
M = E(M)

G = E.random_point()
k = getPrime(int(128))
K = k*G
```

```
r = getrandbits(64)

C1 = M + r*K
C2 = r*G

print(f'p={p}\na={a}\nb={b}')
print(f'G={G.xy()}')
print(f'K={K.xy()}')
print(f'C1={C1.xy()}')
print(f'C2={C2.xy()}')

'''
p=110933004387653577876938231220685019333268291815186936508970907817493795034276
519540285430762475836976695972309342867514288806735391552792323043011239314 19
a=490963434153515882934487973185142842357175523008183292296815140698999054658777
820556076794490414610737654365807063916602037816955706321036900113929329671
b=766854265479378498843649908673923944291517028734612164588409622294833827916530
221344006007914196067967852601634802502955833597704271238261119799500231646 6
G=
(404593966433219228460592428490575019459951411524888561700643583340051625831413 5
019849306107002566248677228498859069119557284134574413164612914441502516162,
284779462783898486680885373079779475894415923975590365209214613793295981613700 69
540453188215319847155621351346812568367947353887453540659947456618329264 04)
K=
(985792549563088647287107284861506976663511525357684319771624233906826915116707 2
057478472997523547299286363591371734837904400286993818976404285783613138603,
998186532993887790457930620042959969048009395155501025880921074045812058650763 81
004687228077173900337842902152171859216901037579118709334972405788676797 16)
C1=
(434966278797352918874161550308557149357143481210574560386820500588546459278253 6
198234863020839759214118594741734453731681116610298272107088387481605173124,
108357083023554257987293929934513371627732530004405663336116106332349292941597 43
316615308778168947697567386109223430056006489876900001115634567822674333770)
C2=
(519386665741749837673713247373273733091657024056904791029314423575260248938809 2
937375844109374780050061859498276712695321973801207620914447727053101524592,
684299154840371832195648774293174908478389728255128448106858267664482339440737 09
981086863390629746545043641709130273947340794395587464848664751111934 1978)
'''
```

Hint



经过noob千辛万苦的搜索，终于找到了这么一串代码o(╥﹏╥)o，使用sagemath求解



根据题目对M[0]进行操作得到flag

```
from hashlib import md5
m=9720789860513058859420153041654292366708275204078293982286281397729948408045118488983319339625580225526216408590830302129037893411810463191124308131948038
def MD5(m):return md5(str(m).encode()).hexdigest()
print('0xGame{'+MD5(m)+'}' )
```

Reverse

序列9-二进制学徒

> 虽说确实简单，但这道题放出来58秒就被人秒了。。。萌新害怕o(╥﹏╥)o

应该是本周最简单的一道题了。

解压得到一个pyc文件，直接对其反编译，即可看见flag

python工具

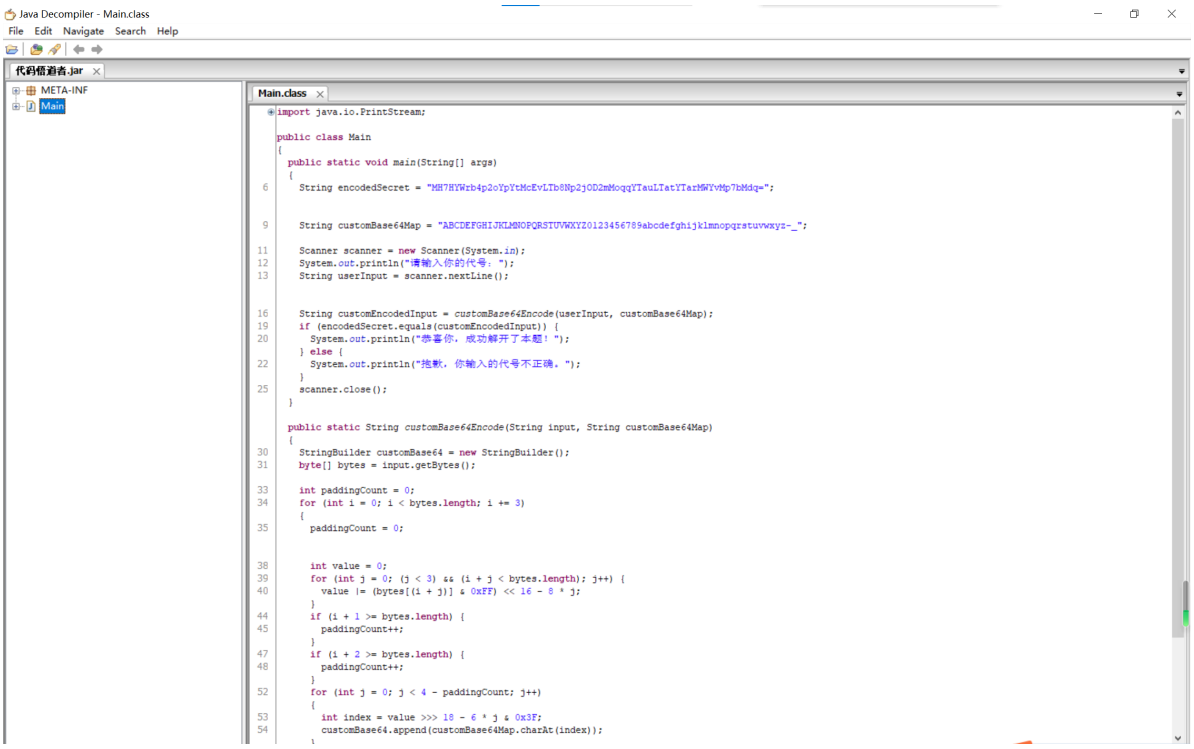
请选择pyc文件进行解密。支持所有Python版本

浏览... 未选择文件。

```python
#!/usr/bin/env python
# visit https://tool.lu/pyc/ for more information
# Version: Python 3.11

'''
0xGame{15cf69f6-69f9-6b18-1464-7785f106b0d3}
'''

def main():
    print('我的flag去哪儿了？')

if __name__ == '__main__':
    main()
    return None
```

序列8-代码悟道者

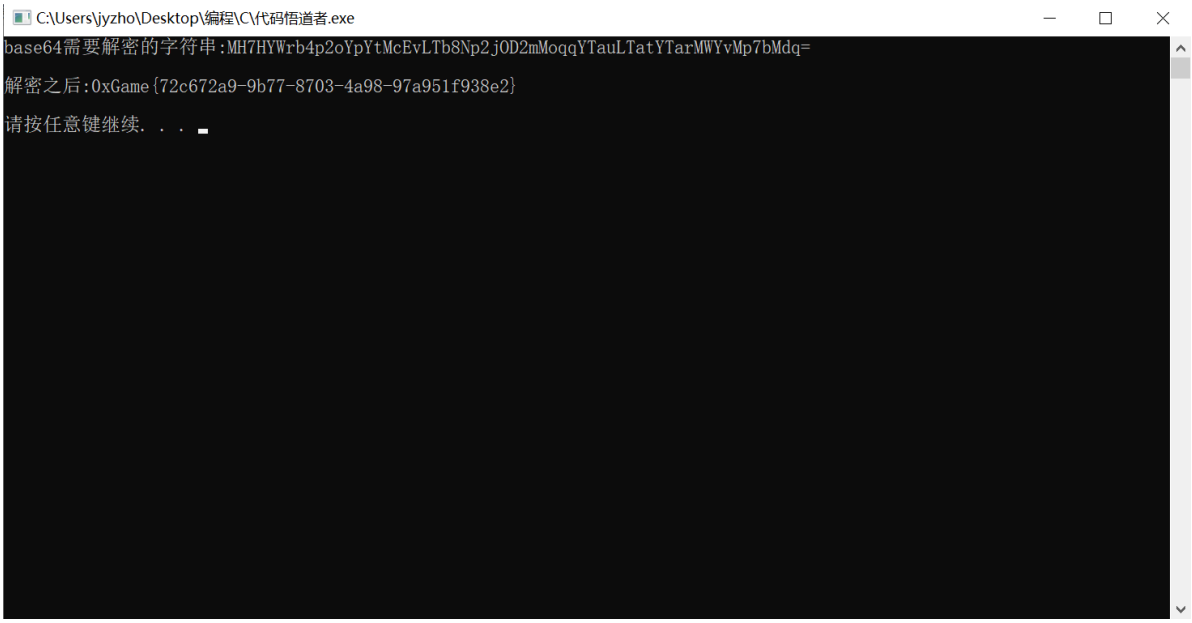
对下载得到的jar文件使用jd-gui进行反编译



从反编译得到的代码中可以看出，是对flag进行了一番自定义的base64加密（将map中的"+"和"/"替换成了"-"和"_"），于是编写相应代码对其进行解码，得到flag

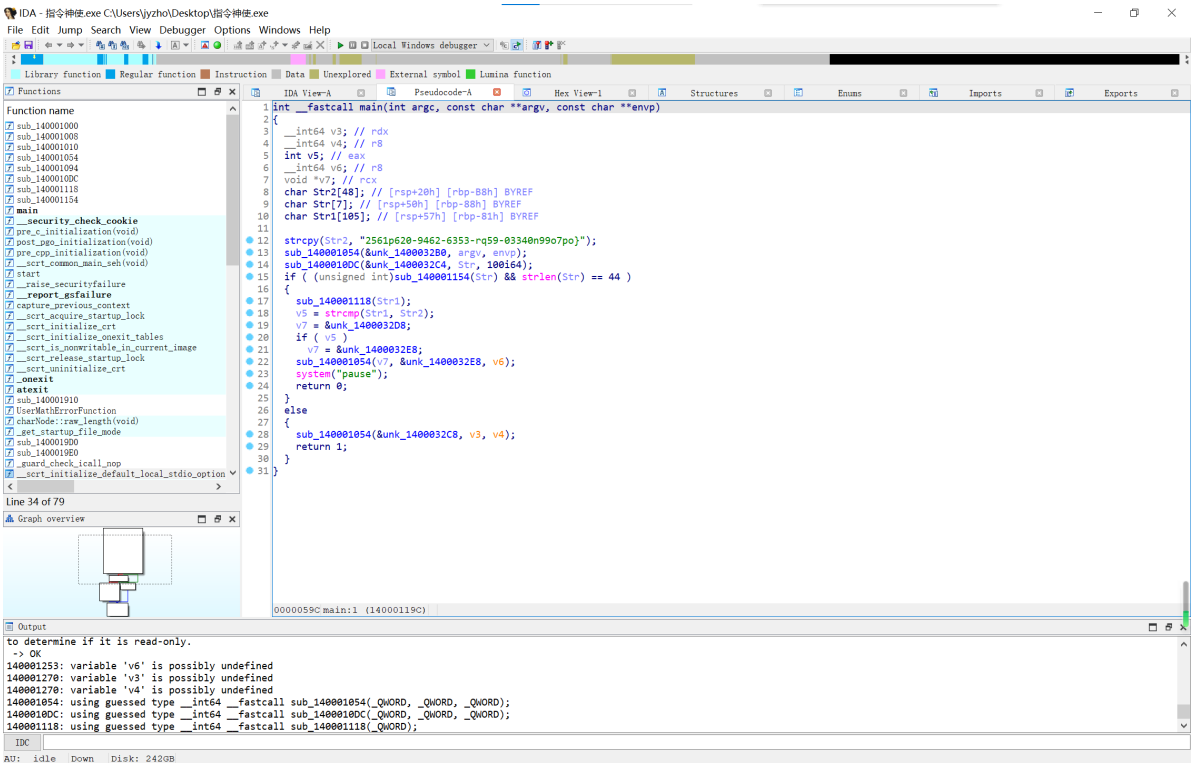
```c
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
char s[100];
int i,j;char
a[]="ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz-_";
```

```
void jiemi(int a,int b,int c,int d)
{
    s[j]=a<<2|b>>4;
    s[j+1]=b<<4|c>>2;
    s[j+2]=c<<6|d;
}
int main()
{
    int len,len1;
    int str1[100];
    char str[100];
    printf("base64需要解密的字符串:");
    gets(str);len=strlen(str);
    for(i=0;i<len;i++)
    {
        for(j=0;j<64;j++)
        {
            if(str[i]==a[j])
            {
                str1[i]=j;
                len1=i+1;
                }
            }
        }
        for(i=0,j=0;i<=len1-4;i+=4,j+=3)
        {
            jiemi(str1[i],str1[i+1],str1[i+2],str1[i+3]);
        }
        if(len1%4==2)
        {
            s[j]=str1[i]<<2|str1[i+1]>>4;
        }
        if(len1%4==3)
        {
            s[j]=str1[i]<<2|str1[i+1]>>4;
            s[j+1]=str1[i+1]<<4|str1[i+2]>>2;
        }
        printf("\n解密之后:");
        puts(s);
        putchar('\n');
        system("pause");
        return 0;
}
```

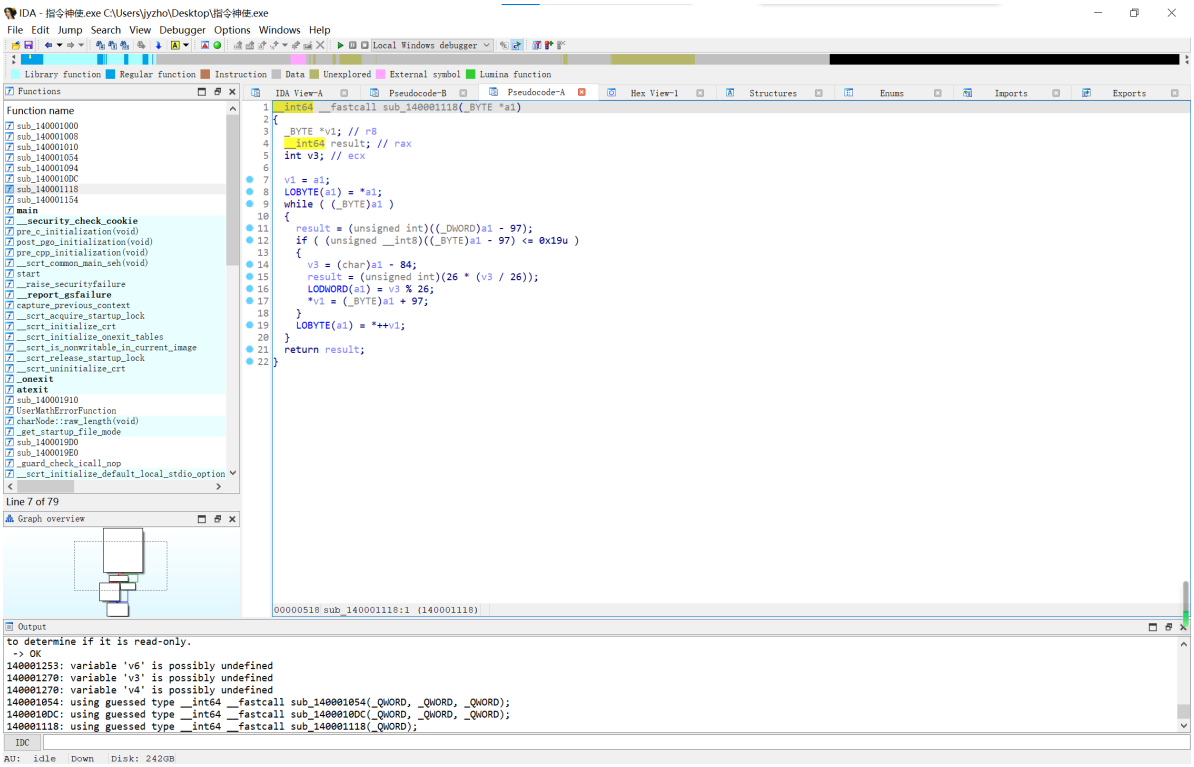


序列7-指令神使

使用IDA对下载得到的exe文件进行反编译，并生成伪代码



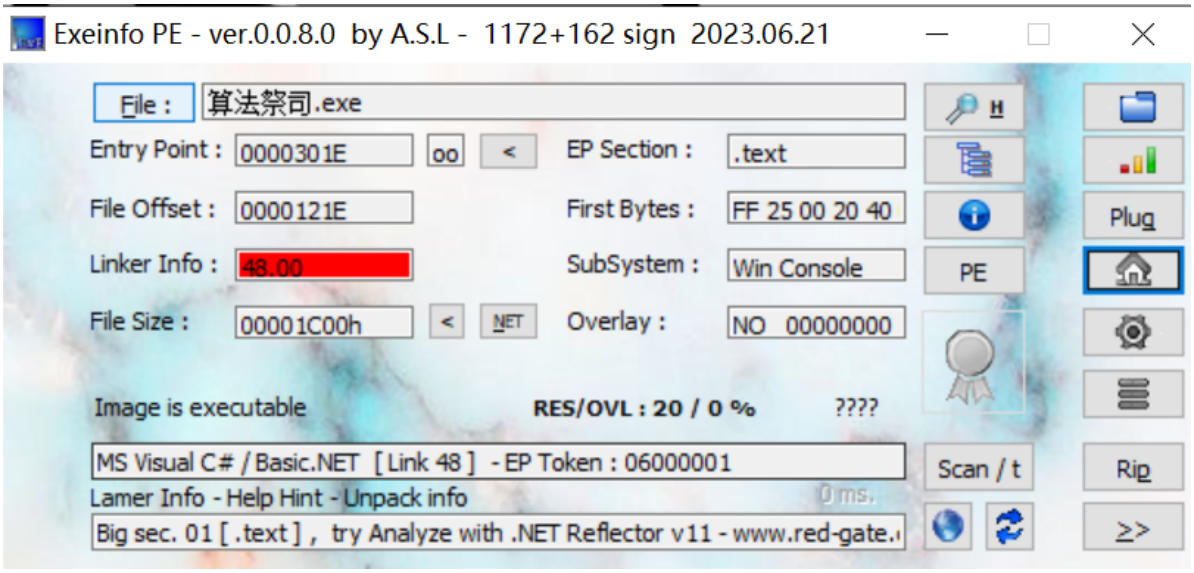
根据伪代码，得知对字符串进行了sub_140001118函数的操作，查看sub_140001118函数的伪代码



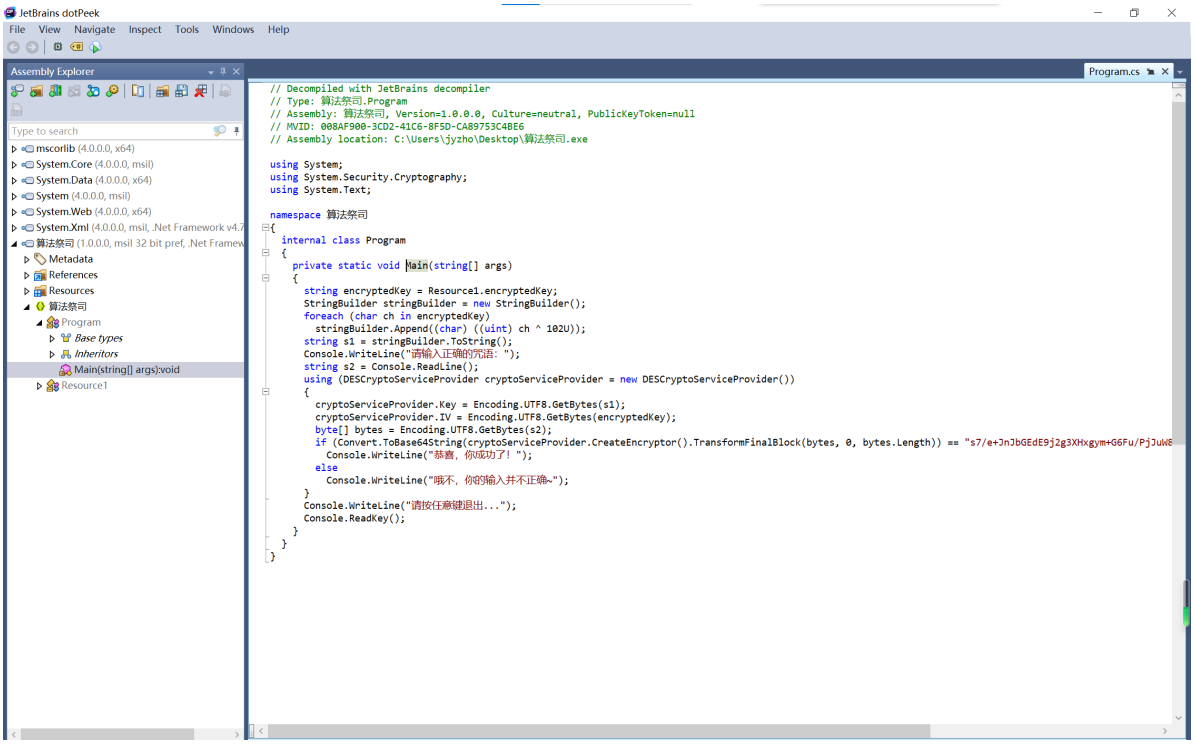
分析伪代码后可以看出，是对其进行了偏移量为13的凯撒加密，于是进行相应解密，得到flag

序列5-算法祭司

对下载下来的exe文件查壳，发现是C#语言编写的



于是用JetBrains dotPeek对其进行反编译，并找到其中的main函数



阅读程序后得知，是对flag进行了DES加密与一次base64加密，且其中的key也被加密了，于是首先在resource中找到encryptedkey

```
<data name="encryptedKey" xml:space="preserve">
  <value>STV&gt;!'+#</value>
```

其中有转义字符，经搜索知是大于号。根据源码对encryptedkey进行异或运算处理，得到key。观察代码得知是DES加密，而又同时存在key和iv，可推测得知是CBC_mode。于是开写脚本。

```
from Crypto.Cipher import DES
from base64 import *
key = "STV>!'+#"
keyy= ""
for i in key:
    keyy+=chr(ord(i)^102)
keyy=keyy.encode()
key=key.encode()
print(keyy)
encrypted_text =
b64decode(b's7/e+JnJbGEdE9j2g3XHxgym+G6Fu/PjJuW8ONeMKgemdqaWG9KVM8TfccOeRfaA')
des = DES.new(key=keyy, mode=DES.MODE_CBC,iv=key)
text = des.decrypt(encrypted_text)
print(text)
```

得到结果（上面的是key，下面的是flag）