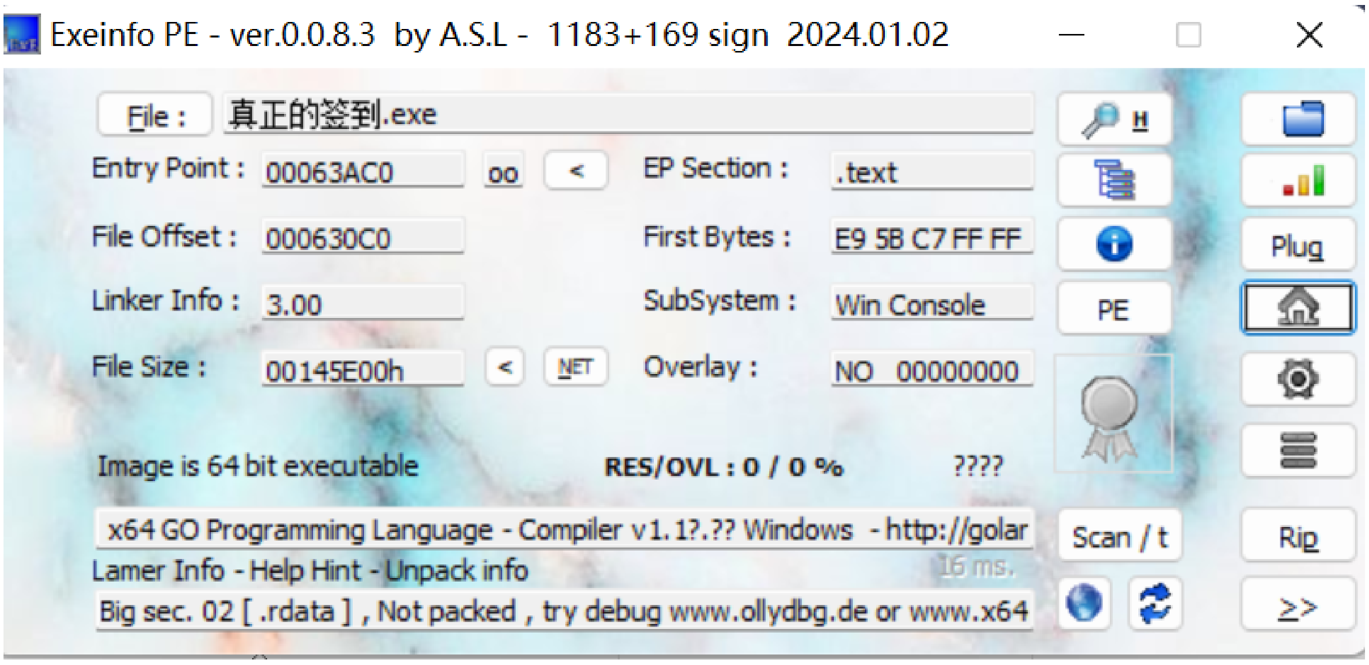
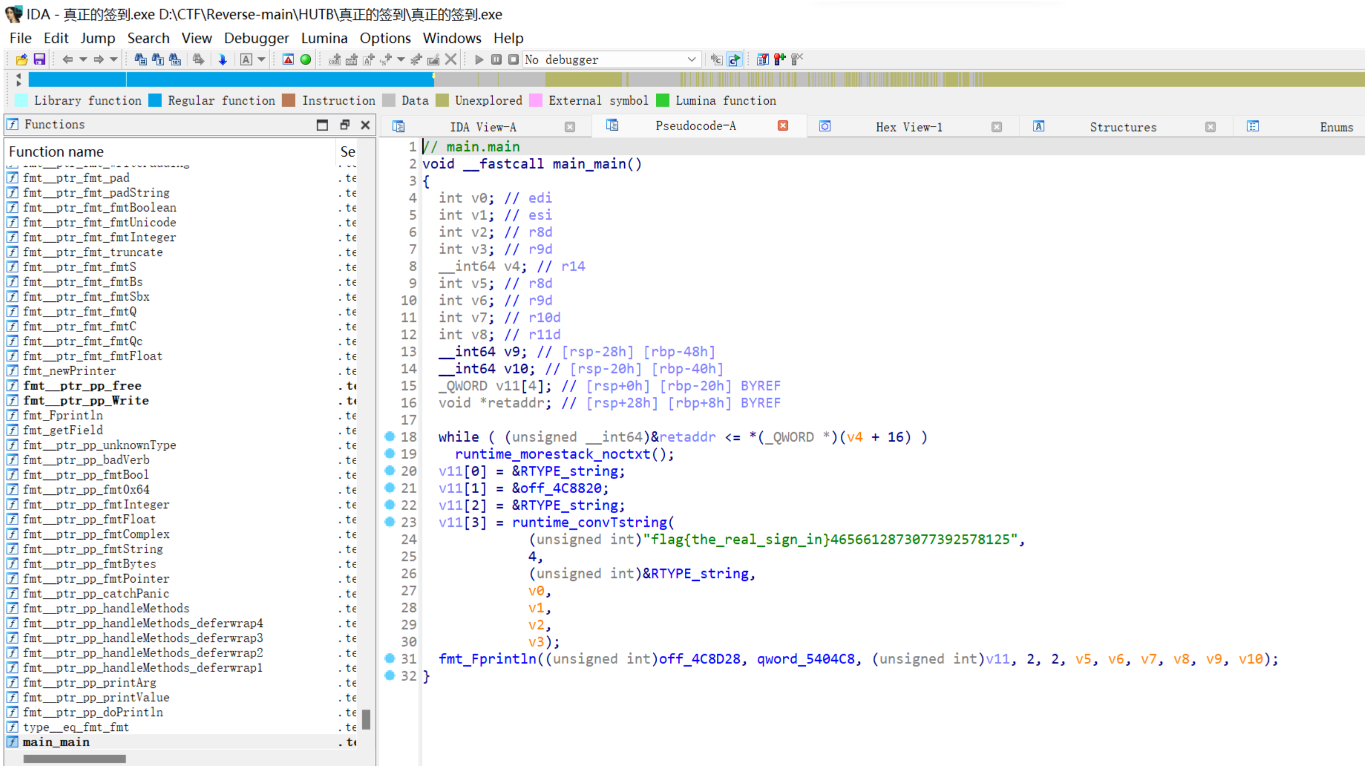
真正的签到

先用`Exeinfo`查壳: 无壳，64位，go语言


Exeinfo PE - ver.0.0.8.3 by A.S.L - 1183+169 sign 2024.01.02

File : 真正的签到.exe
Entry Point : 00063AC0
EP Section : .text
File Offset : 000630C0
First Bytes : E9 5B C7 FF FF
Linker Info : 3.00
SubSystem : Win Console
File Size : 00145E00h
Overlay : NO 00000000
Image is 64 bit executable
RES/OVL : 0 / 0 %
????
x64 GO Programming Language - Compiler v1.1?.?? Windows - http://golar
Lamer Info - Help Hint - Unpack info
16 ms.
Big sec. 02 [.rdata] , Not packed , try debug www.ollydbg.de or www.x64

用`IDA64`打开，找到main函数，直接就有flag


IDA - 真正的签到.exe D:\CTF\Reverse-main\HUTB\真正的签到\真正的签到.exe

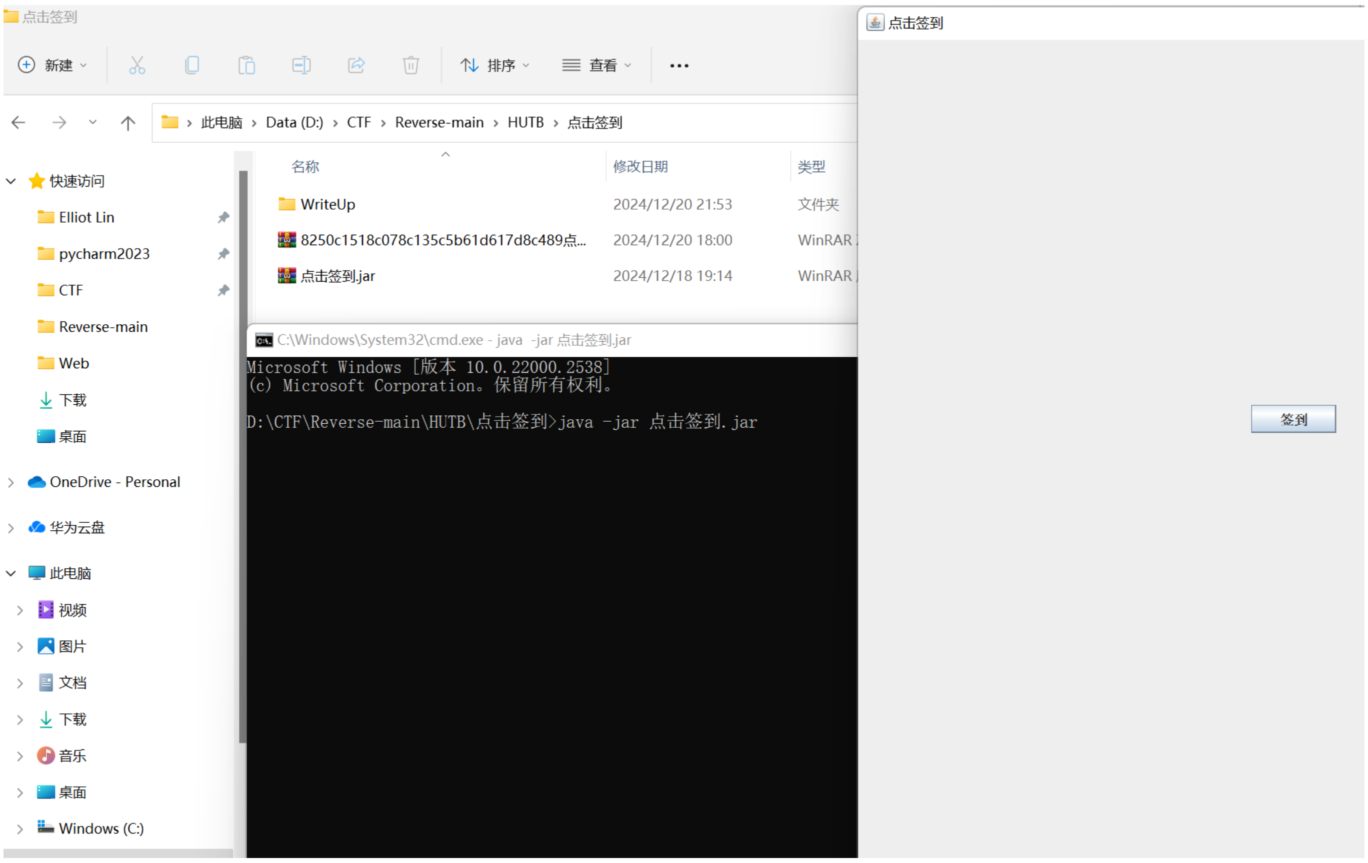
```
Function name                                  Se
fmt__ptr_fmt_pad                               .te
fmt__ptr_fmt_padString                         .te
fmt__ptr_fmt_fmtBoolean                        .te
fmt__ptr_fmt_fmtUnicode                        .te
fmt__ptr_fmt_fmtInteger                        .te
fmt__ptr_fmt_truncate                          .te
fmt__ptr_fmt_fmtS                              .te
fmt__ptr_fmt_fmtBs                             .te
fmt__ptr_fmt_fmtSbx                            .te
fmt__ptr_fmt_fmtQ                              .te
fmt__ptr_fmt_fmtC                              .te
fmt__ptr_fmt_fmtQc                             .te
fmt__ptr_fmt_fmtFloat                          .te
fmt_newPrinter                                 .te
fmt__ptr_pp_free                               .te
fmt__ptr_pp_Write                              .te
fmt_Fprintln                                   .te
fmt_getField                                   .te
fmt__ptr_pp_unknownType                        .te
fmt__ptr_pp_badVerb                            .te
fmt__ptr_pp_fmtBool                            .te
fmt__ptr_pp_fmt0x64                            .te
fmt__ptr_pp_fmtInteger                         .te
fmt__ptr_pp_fmtFloat                           .te
fmt__ptr_pp_fmtComplex                         .te
fmt__ptr_pp_fmtString                          .te
fmt__ptr_pp_fmtBytes                           .te
fmt__ptr_pp_fmtPointer                         .te
fmt__ptr_pp_catchPanic                         .te
fmt__ptr_pp_handleMethods                      .te
fmt__ptr_pp_handleMethods_deferwrap4           .te
fmt__ptr_pp_handleMethods_deferwrap3           .te
fmt__ptr_pp_handleMethods_deferwrap2           .te
fmt__ptr_pp_handleMethods_deferwrap1           .te
fmt__ptr_pp_printArg                           .te
fmt__ptr_pp_printValue                         .te
fmt__ptr_pp_doPrintln                          .te
type__eq_fmt_fmt                               .te
main_main                                      .te
```

```
// main.main
void __fastcall main_main()
{
  int v0; // edi
  int v1; // esi
  int v2; // r8d
  int v3; // r9d
  __int64 v4; // r14
  int v5; // r8d
  int v6; // r9d
  int v7; // r10d
  int v8; // r11d
  __int64 v9; // [rsp-28h] [rbp-48h]
  __int64 v10; // [rsp-20h] [rbp-40h]
  _QWORD v11[4]; // [rsp+0h] [rbp-20h] BYREF
  void *retaddr; // [rsp+28h] [rbp+8h] BYREF

  while ( (unsigned __int64)&retaddr <= *(_QWORD *)(v4 + 16) )
    runtime_morestack_noctxt();
  v11[0] = &RTYPE_string;
  v11[1] = &off_4C8820;
  v11[2] = &RTYPE_string;
  v11[3] = runtime_convTstring(
             (unsigned int)"flag{the_real_sign_in}465661287307739257812 5",
             4,
             (unsigned int)&RTYPE_string,
             v0,
             v1,
             v2,
             v3);
  fmt_Fprintln((unsigned int)off_4C8D28, qword_5404C8, (unsigned int)v11, 2, 2, v5, v6, v7, v8, v9, v10);
}
```

```
flag{the_real_sign_in}
```

点击签到

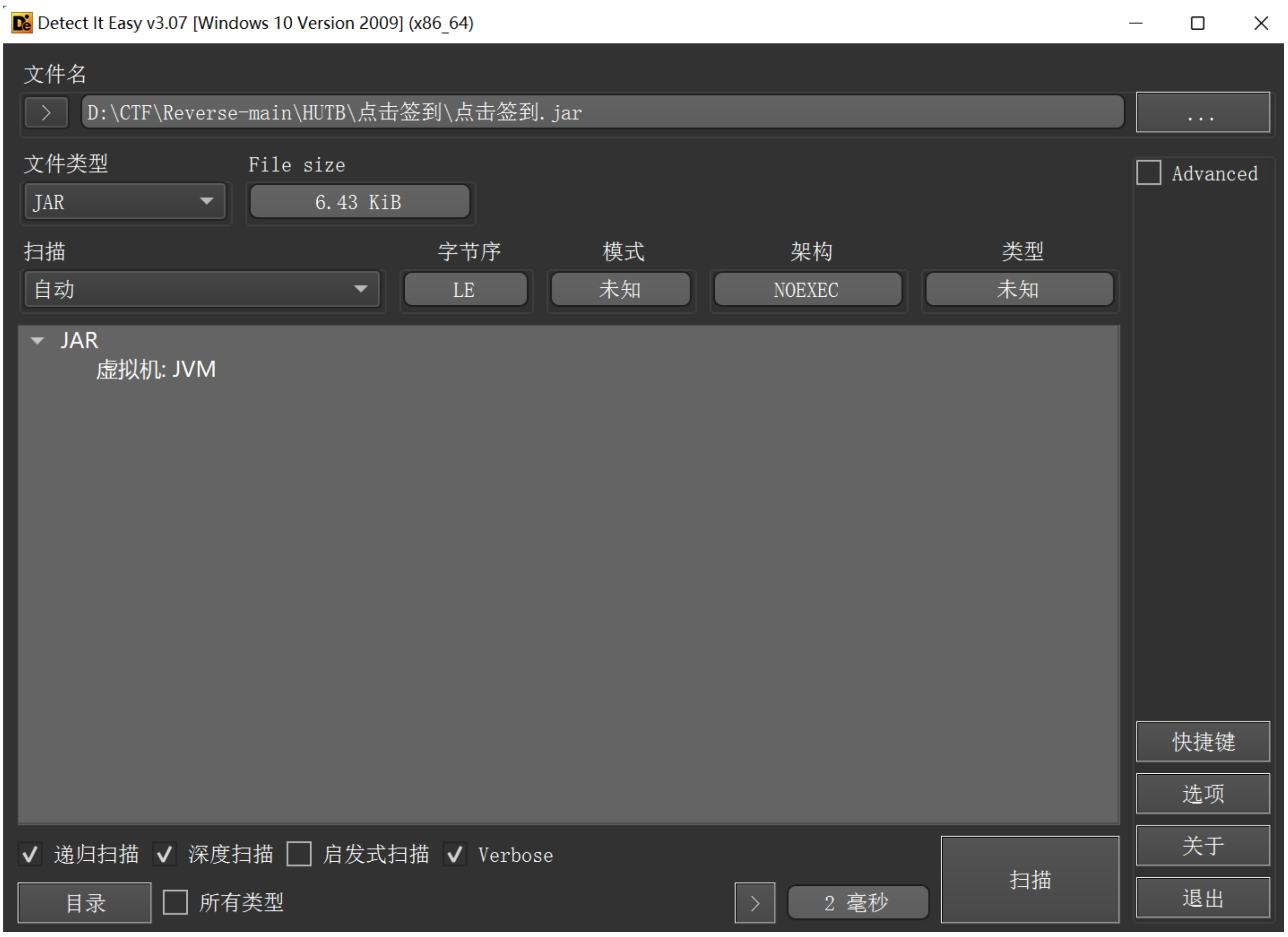
看到jar包，先用java关联jar包运行一下



```
Microsoft Windows [版本 10.0.22000.2538]
(c) Microsoft Corporation。保留所有权利。

D:\CTF\Reverse-main\HUTB\点击签到>java -jar 点击签到.jar
```

但是签到按钮是按不到的，会闪现

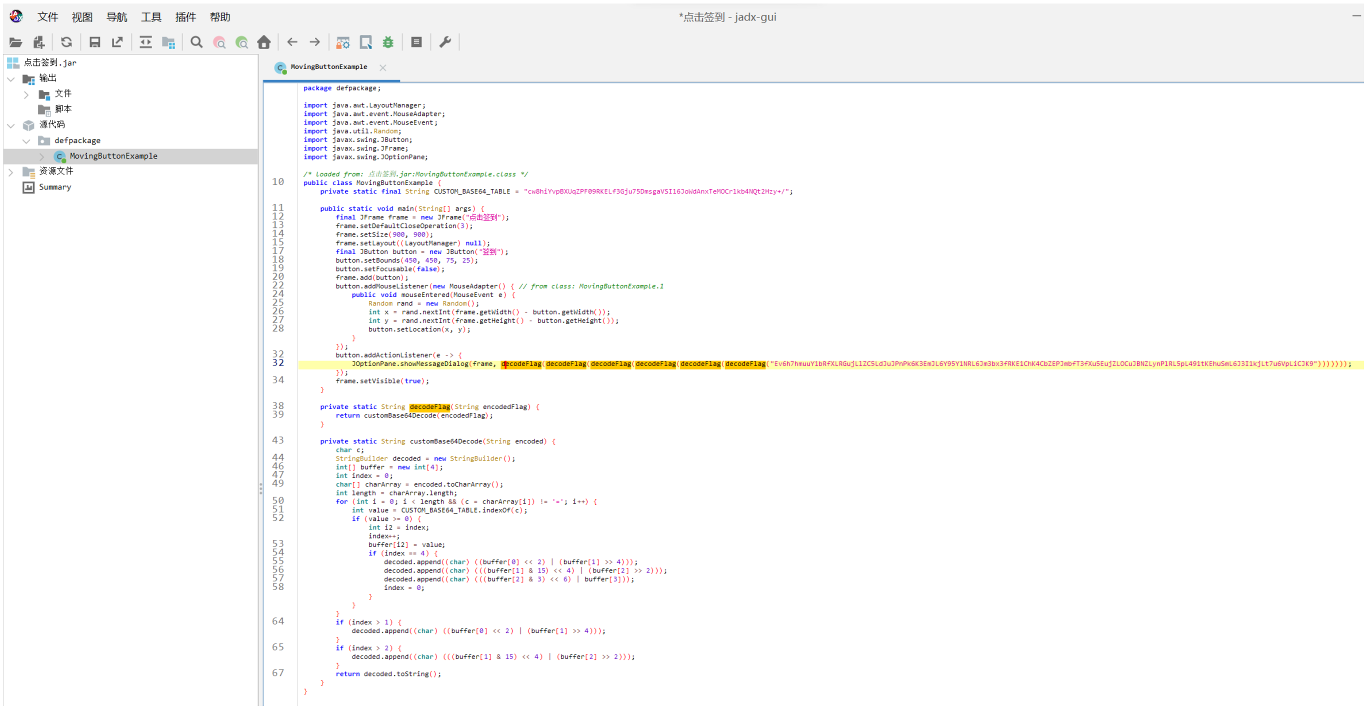
于是先用die查壳，无壳



文件名: D:\CTF\Reverse-main\HUTB\点击签到\点击签到.jar
文件类型: JAR
File size: 6.43 KiB
扫描: 自动
字节序: LE
模式: 未知
架构: NOEXEC
类型: 未知

```
JAR
    虚拟机: JVM
```

然后用`jadx-gui`反编译，找到主函数



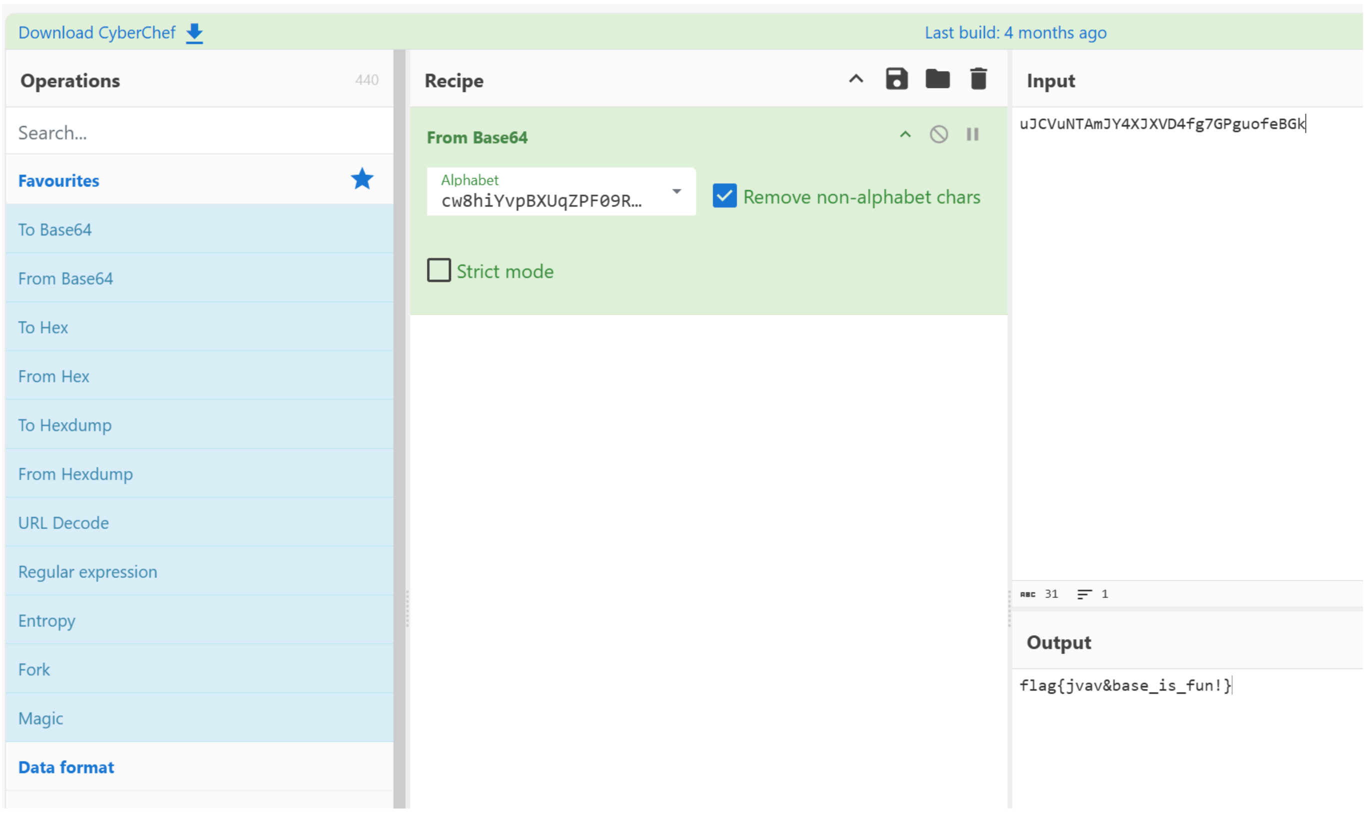
```java
package defpackage;

import java.awt.LayoutManager;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.util.Random;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JOptionPane;

/* loaded from: 点击签到.jar:MovingButtonExample.class */
public class MovingButtonExample {
    private static final String CUSTOM_BASE64_TABLE = "cw8hiYvpBXUqZPF09RKELf3Gju75DmsgaVSI16JoWdAnxTeMOCrlkb4NQt2Hzy+/";

    public static void main(String[] args) {
        final JFrame frame = new JFrame("点击签到");
        frame.setDefaultCloseOperation(3);
        frame.setSize(900, 900);
        frame.setLayout((LayoutManager) null);
        final JButton button = new JButton("签到");
        button.setBounds(450, 450, 75, 25);
        button.setFocusable(false);
        frame.add(button);
        button.addMouseListener(new MouseAdapter() { // from class: MovingButtonExample.1
            public void mouseEntered(MouseEvent e) {
                Random rand = new Random();
                int x = rand.nextInt(frame.getWidth() - button.getWidth());
                int y = rand.nextInt(frame.getHeight() - button.getHeight());
                button.setLocation(x, y);
            }
        });
        button.addActionListener(e -> {
            JOptionPane.showMessageDialog(frame, decodeFlag(decodeFlag(decodeFlag(decodeFlag(decodeFlag(decodeFlag("Ev6h7hmuuY1bRfXLRGujL1ZC5LdJuJPnPk6K3EmJL6Y95Y1NRL6Jm3bx3fRKE1ChK4CbZEPJmbfT3fXu5EujZLOCuJBNZLynP1RL5pL491tKEhuSmL6J3I1kjLt7u6VpLiCJK9")))))));
        });
        frame.setVisible(true);
    }

    private static String decodeFlag(String encodedFlag) {
        return customBase64Decode(encodedFlag);
    }

    private static String customBase64Decode(String encoded) {
        char c;
        StringBuilder decoded = new StringBuilder();
        int[] buffer = new int[4];
        int index = 0;
        char[] charArray = encoded.toCharArray();
        int length = charArray.length;
        for (int i = 0; i < length && (c = charArray[i]) != '='; i++) {
            int value = CUSTOM_BASE64_TABLE.indexOf(c);
            if (value >= 0) {
                int i2 = index;
                index++;
                buffer[i2] = value;
                if (index == 4) {
                    decoded.append((char) ((buffer[0] << 2) | (buffer[1] >> 4)));
                    decoded.append((char) (((buffer[1] & 15) << 4) | (buffer[2] >> 2)));
                    decoded.append((char) (((buffer[2] & 3) << 6) | buffer[3]));
                    index = 0;
                }
            }
        }
        if (index > 1) {
            decoded.append((char) ((buffer[0] << 2) | (buffer[1] >> 4)));
        }
        if (index > 2) {
            decoded.append((char) (((buffer[1] & 15) << 4) | (buffer[2] >> 2)));
        }
        return decoded.toString();
    }
}
```

发现是`base64换表`，而且还加密了6次

```
base64表:
cw8hiYvpBXUqZPF09RKELf3Gju75DmsgaVSI16JoWdAnxTeMOCrlkb4NQt2Hzy+/
```

```
加密6次后:
Ev6h7hmuuY1bRfXLRGujLlZC5LdJuJPnPk6K3EmJL6Y95Y1NRL6Jm3bx3fRKE1ChK4CbZEPJmbfT3fXu5E
ujZLOCuJBNZLynPlRL5pL491tKEhuSmL6J3I1kjLt7u6VpLiCJK9
```

用`Cyberchef`，解密得flag



From Base64
Alphabet cw8hiYvpBXUqZPF09R...
Remove non-alphabet chars ✓
Strict mode ☐

```
uJCVuNTAmJY4XJXVD4fg7GPguofeBGk
```
31 1

```
flag{jvav&base_is_fun!}
```

```
flag{jvav&base_is_fun!}
```

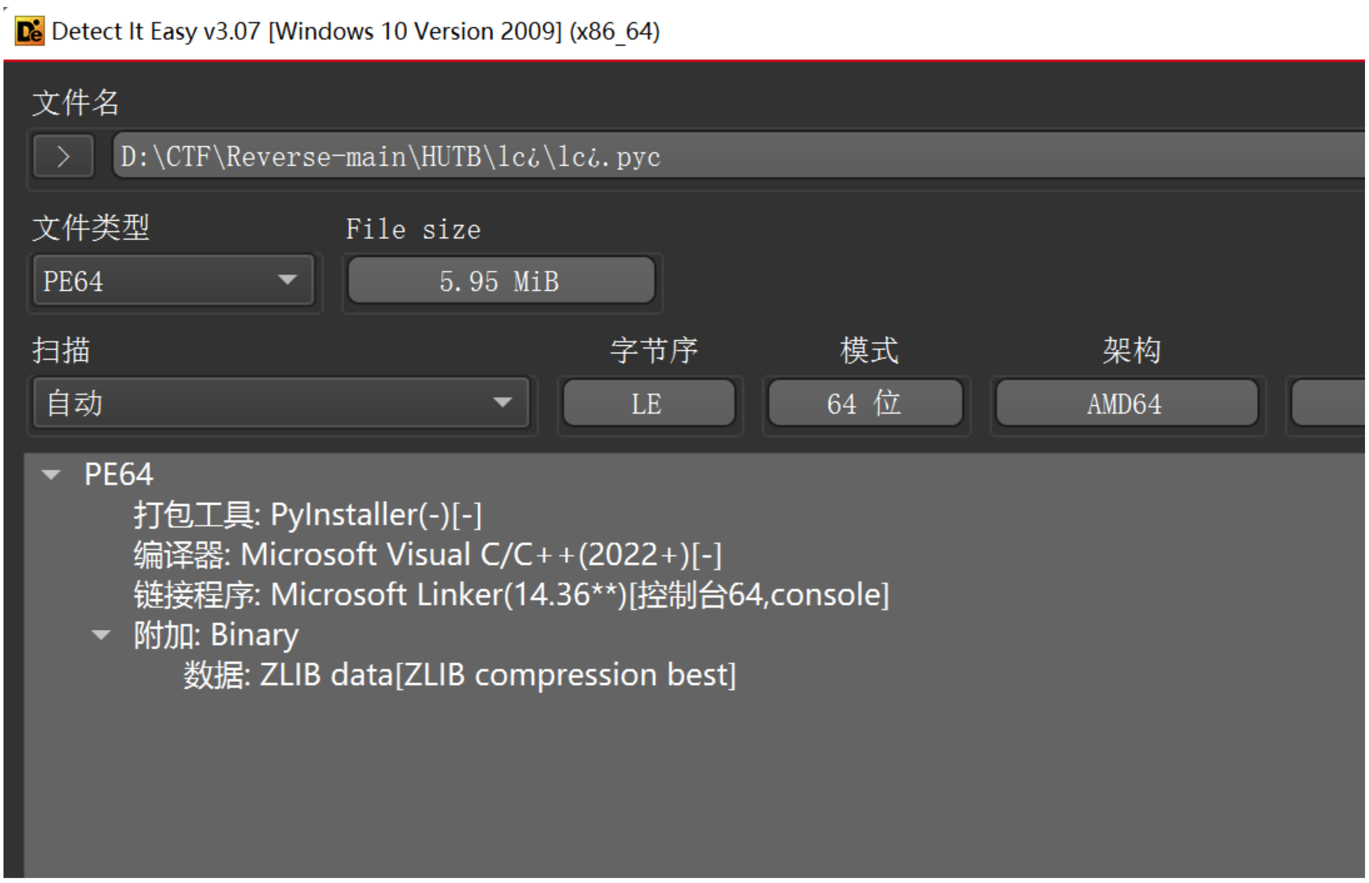
---

lc¿

提示：不记得了，但是说是python

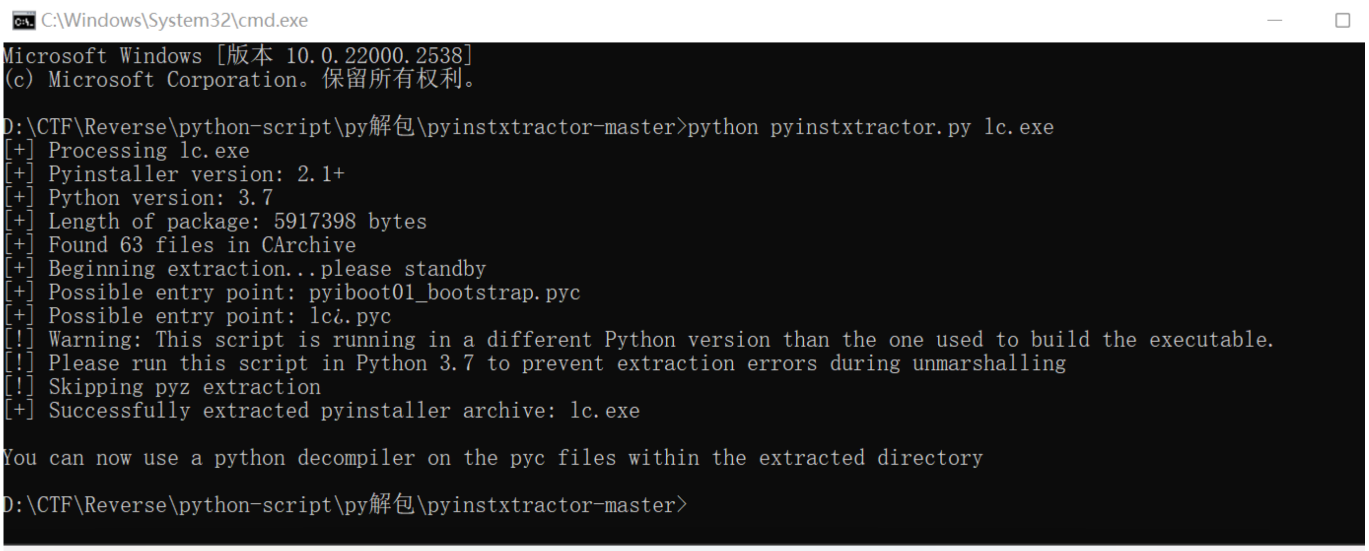
一开始单纯的，把exe改成pyc,想着直接用`pycdc`反编译,后面一直错误，反编译不了

仔细一看才发现用了`PyInstaller`打包



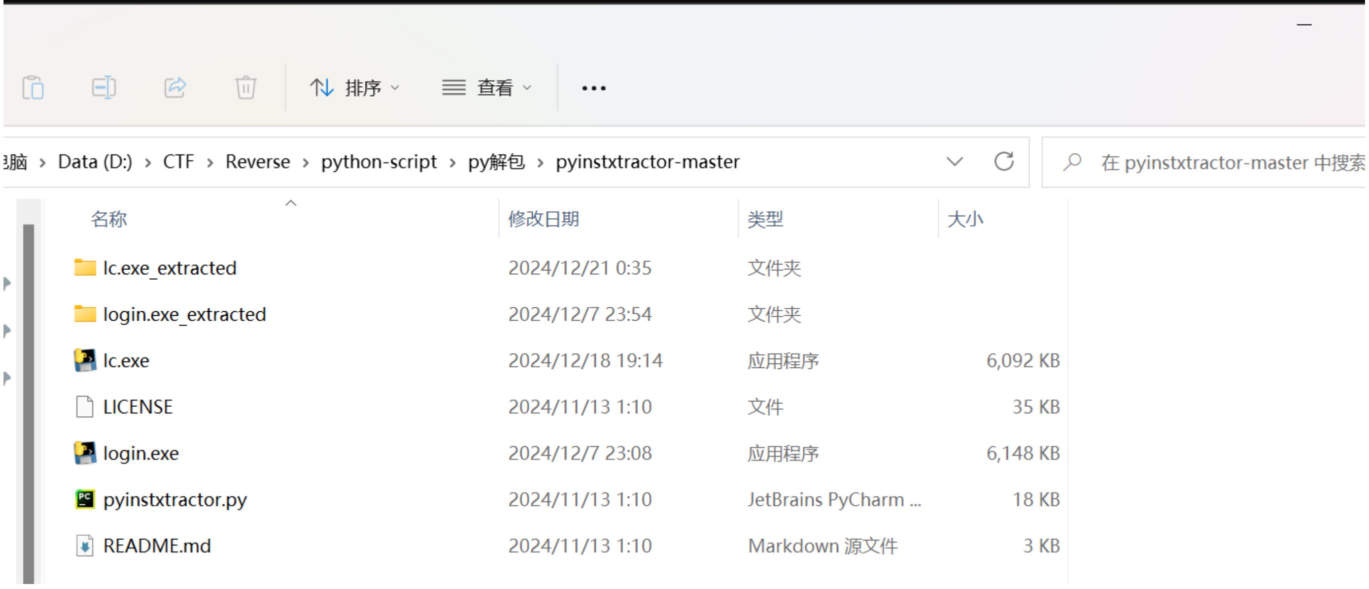
```
D:\CTF\Reverse-main\HUTB\lc¿\lc¿.pyc
PE64
5.95 MiB
自动
LE
64 位
AMD64
PE64
    打包工具: PyInstaller(-)[-]
    编译器: Microsoft Visual C/C++(2022+)[-]
    链接程序: Microsoft Linker(14.36**)[控制台64,console]
    附加: Binary
        数据: ZLIB data[ZLIB compression best]
```

用 `pyinstxtractor.py`解包，生成`lc.exe_extracted文件夹`



C:\Windows\System32\cmd.exe

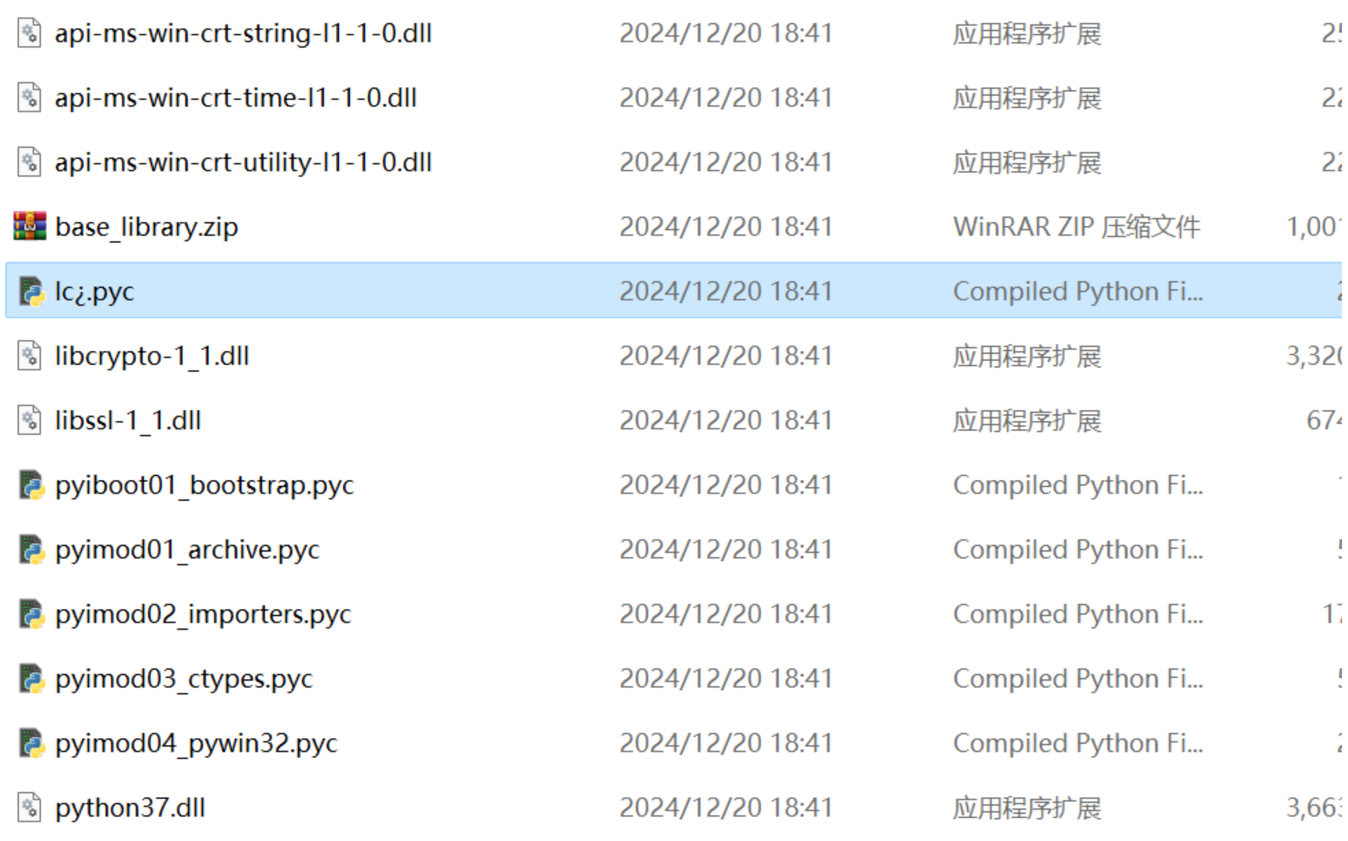
```
Microsoft Windows [版本 10.0.22000.2538]
(c) Microsoft Corporation。保留所有权利。

D:\CTF\Reverse\python-script\py解包\pyinstxtractor-master>python pyinstxtractor.py lc.exe
[+] Processing lc.exe
[+] Pyinstaller version: 2.1+
[+] Python version: 3.7
[+] Length of package: 5917398 bytes
[+] Found 63 files in CArchive
[+] Beginning extraction...please standby
[+] Possible entry point: pyiboot01_bootstrap.pyc
[+] Possible entry point: lc¿.pyc
[!] Warning: This script is running in a different Python version than the one used to build the executable.
[!] Please run this script in Python 3.7 to prevent extraction errors during unmarshalling
[!] Skipping pyz extraction
[+] Successfully extracted pyinstaller archive: lc.exe

You can now use a python decompiler on the pyc files within the extracted directory

D:\CTF\Reverse\python-script\py解包\pyinstxtractor-master>
```



排序
查看
脑 › Data (D:) › CTF › Reverse › python-script › py解包 › pyinstxtractor-master
在 pyinstxtractor-master 中搜索

名称	修改日期	类型	大小
lc.exe_extracted	2024/12/21 0:35	文件夹	
login.exe_extracted	2024/12/7 23:54	文件夹	
lc.exe	2024/12/18 19:14	应用程序	6,092 KB
LICENSE	2024/11/13 1:10	文件	35 KB
login.exe	2024/12/7 23:08	应用程序	6,148 KB
pyinstxtractor.py	2024/11/13 1:10	JetBrains PyCharm ...	18 KB
README.md	2024/11/13 1:10	Markdown 源文件	3 KB

在`lc.exe_extracted文件夹`里找到`lc.pyc`

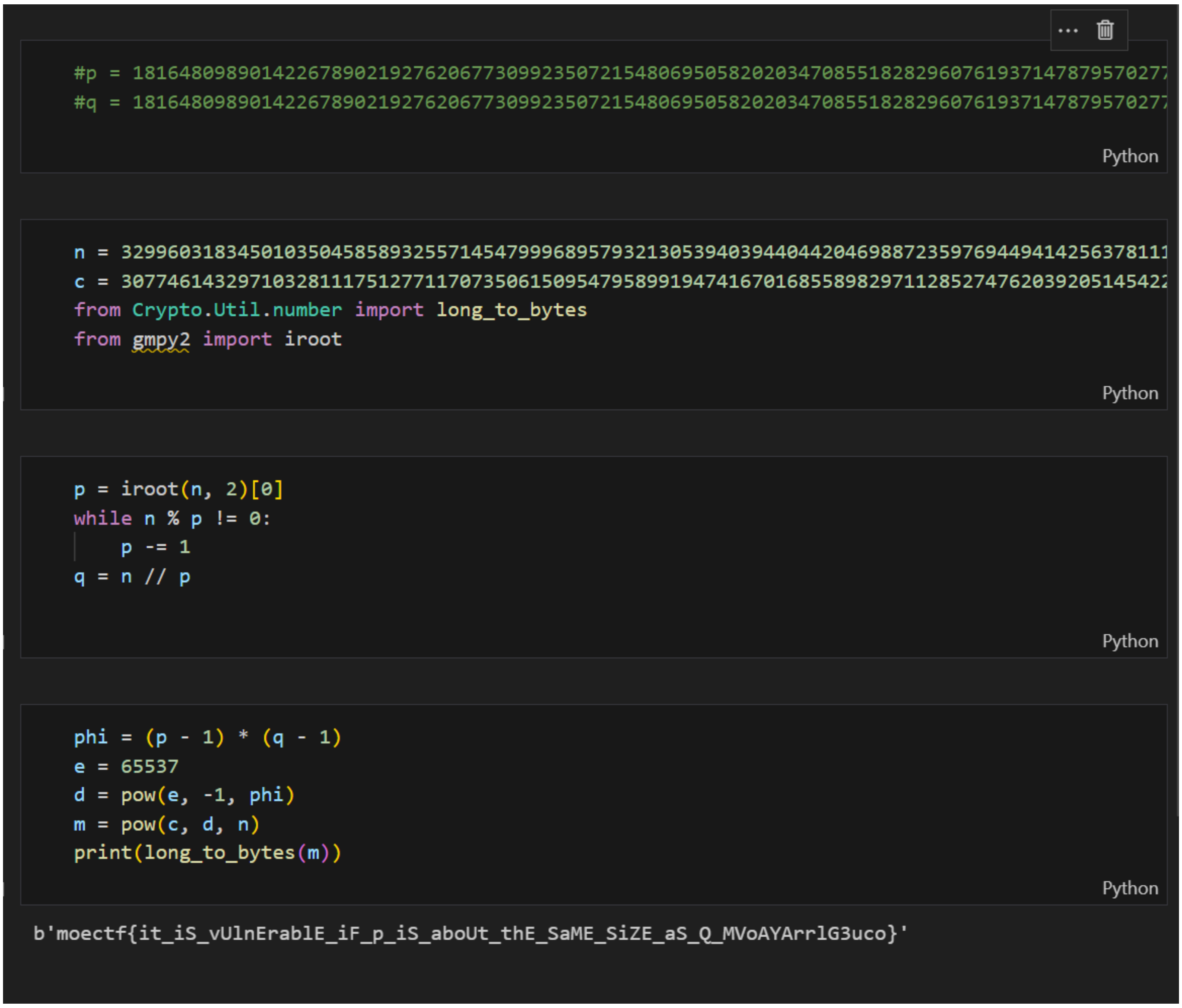


```
api-ms-win-crt-string-l1-1-0.dll    2024/12/20 18:41    应用程序扩展            25
api-ms-win-crt-time-l1-1-0.dll      2024/12/20 18:41    应用程序扩展            22
api-ms-win-crt-utility-l1-1-0.dll   2024/12/20 18:41    应用程序扩展            22
base_library.zip                    2024/12/20 18:41    WinRAR ZIP 压缩文件     1,00
lc¿.pyc                             2024/12/20 18:41    Compiled Python Fi...
libcrypto-1_1.dll                   2024/12/20 18:41    应用程序扩展            3,32
libssl-1_1.dll                      2024/12/20 18:41    应用程序扩展            67
pyiboot01_bootstrap.pyc             2024/12/20 18:41    Compiled Python Fi...
pyimod01_archive.pyc                2024/12/20 18:41    Compiled Python Fi...
pyimod02_importers.pyc              2024/12/20 18:41    Compiled Python Fi...   17
pyimod03_ctypes.pyc                 2024/12/20 18:41    Compiled Python Fi...
pyimod04_pywin32.pyc                2024/12/20 18:41    Compiled Python Fi...
python37.dll                        2024/12/20 18:41    应用程序扩展            3,66
```

最后用pycdc反编译pyc,得到python代码

```
def hint():
    print('Rsa')
    print('p和q很接近哦')

if __name__ == '__main__':
    print('n = 
1514158276342195581925404214710254440236861174216388878176509008349573701089239041
4303206064971161690095468260109214755318435715386176702148980805904090506865492166
0994010965494136629953524211626163626676880506977939356425246922919208825094053268
0835777419440023336684000258536476186715972905868449561222228623694186212540908835
3147041225896774692317241635988687686984697517087249953776324748416170514112457033
3213571629803137007297262837708874607189116723053540266603307000261449866334869404
6509199512320676615478399142113039235878616192149228851327723554948864316162101397
4849758920117109758764420854228779898068317')
    print('c = 
6178794643928434161739023306889397128020055578901057529700316058628342851885977403
7914480785744678382885239164293443033260780388480029214015529936868892215640989561
1426635185547788381194934802572080368424849053875934732912958605685553239880547505
7787263565468485191683410525901052001664245705489501950325712240324062464375844368
3166592953077902615214217944977169951729588151794697637737492352964635743141880929
1709960149876659488878650765038217725072126703971484517571364552698083812272727341
1814162789459072852328640597548883993448105141213457020252705294875782114742559941
965078906839563983823333479934985791051343')
```

卡了很久，最后用了以前做西电题的脚本



```python
#p = 18164809890142267890219276206773099235072154806950582020347085518282960761937147879570277
#q = 18164809890142267890219276206773099235072154806950582020347085518282960761937147879570277
```

```python
n = 32996031834501035045858932557145479996895793213053940394404420469887235976944941425637811
c = 30774614329710328111751277117073506150954795899194741670168558982971128527476203920514542
from Crypto.Util.number import long_to_bytes
from gmpy2 import iroot
```

```python
p = iroot(n, 2)[0]
while n % p != 0:
    p -= 1
q = n // p
```

```python
phi = (p - 1) * (q - 1)
e = 65537
d = pow(e, -1, phi)
m = pow(c, d, n)
print(long_to_bytes(m))
```

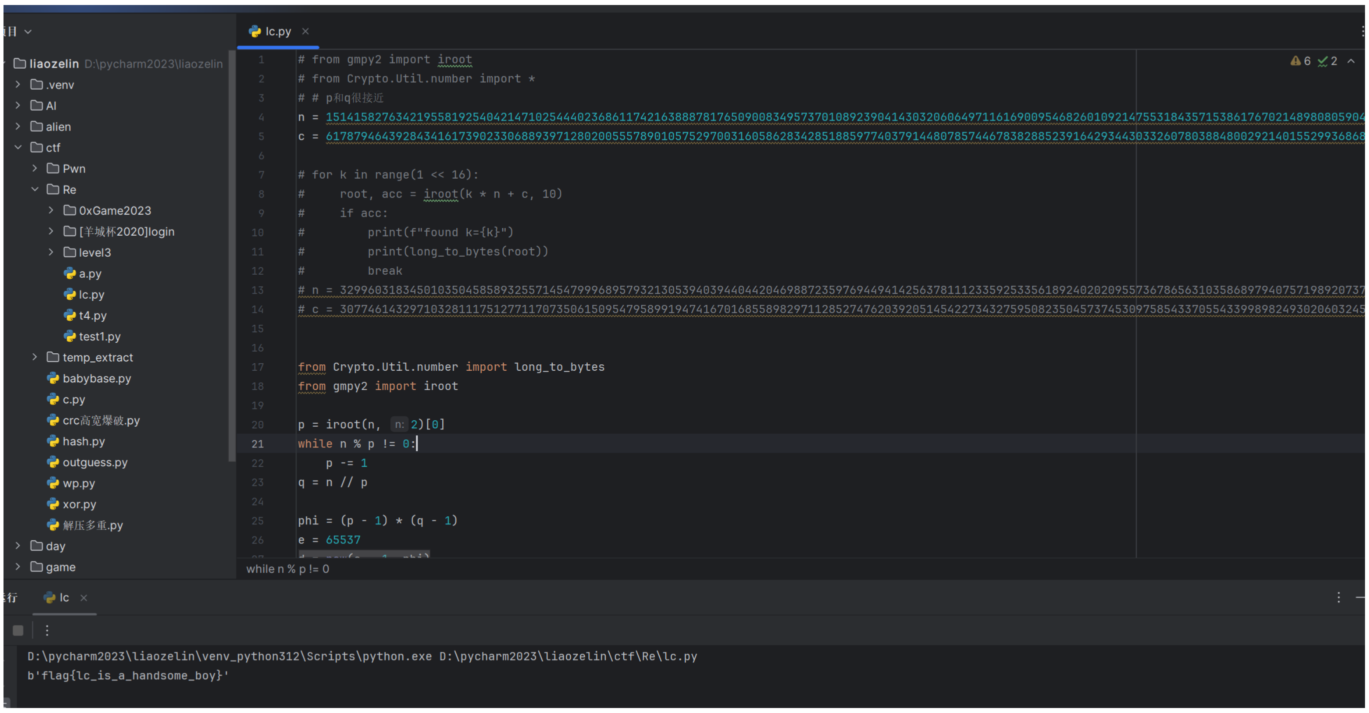
```
b'moectf{it_iS_vUlnErablE_iF_p_iS_aboUt_thE_SaME_SiZE_aS_Q_MVoAYArrlG3uco}'
```

因为p和q相差很小，于是先想到开根号

e没给， 于是默认是65537了

```python
from Crypto.Util.number import long_to_bytes
from gmpy2 import iroot

n =
151415827634219558192540421471025444023686117421638887817650900834957370108923904l
4303206064971161690095468260109214755318435715386176702148980805904090506865492166
0994010965494136629953524211626163626676880506977939356425246922919208825094053268
0835777419440023336684000258536476186715972905868449561222228623694186212540908835
3147041225896774692317241635988687686984697517087249953776324748416170514112457033
3213571629803137007297262837708874607189116723053540266603307000261449866334869404
6509199512320676615478399142113039235878616192149228851327723554948864316162101397
48497589201171097587644208542287798980683l7
c =
6178794643928434161739023306889397128020055578901057529700316058628342851885977403
7914480785744678382885239164293443033260780388480029214015529936868892215640989561
1426635185547788381194934802572080368424849053875934732912958605685553239880547505
7787263565468485191683410525901052001664245705489501950325712240324062464375844368
```

```python
316659295307790261521421794497716995172958815179469763773749235296463574314188092
9
170996014987665948887865076503821772507212670397148451757136455269808381227272734
1
181416278945907285232864059754888399344810514121345702025270529487578211474255994
1
96507890683956398382333479934985791051343

p = iroot(n, 2)[0]
while n % p != 0:
    p -= 1
q = n // p

phi = (p - 1) * (q - 1)
e = 65537
d = pow(e, -1, phi)
m = pow(c, d, n)
print(long_to_bytes(m))

# b'flag{lc_is_a_handsome_boy}'
```



```python
# from gmpy2 import iroot
# from Crypto.Util.number import *
# # p和q很接近
n = 15141582763421955819254042147102544402368611742163888781765090083495737010892390414303206064971161690095468260109214755318435715386176702148980805904
c = 61787946439284341617390233068893971280200555789010575297003160586283428518859774037914480785744678382885239164293443033260780388480029214015529936868

# for k in range(1 << 16):
#     root, acc = iroot(k * n + c, 10)
#     if acc:
#         print(f"found k={k}")
#         print(long_to_bytes(root))
#         break
# n = 329960318345010350458589325571454799968957932130539403944044204698872359769449414256378111233592533561892402020955736786563103586897940757198920737
# c = 307746143297103281117512771170735061509547958991947416701685589829711285274762039205145422734327595082350457374530975854337055433998982493020603245


from Crypto.Util.number import long_to_bytes
from gmpy2 import iroot

p = iroot(n, 2)[0]
while n % p != 0:
    p -= 1
q = n // p

phi = (p - 1) * (q - 1)
e = 65537
```

```
D:\pycharm2023\liaozelin\venv_python312\Scripts\python.exe D:\pycharm2023\liaozelin\ctf\Re\lc.py
b'flag{lc_is_a_handsome_boy}'
```