

buuctf - re 刷题记录 1-18

原创

wwwzzlll 于 2021-10-29 18:52:06 发布 76 收藏

分类专栏: [CTF系列问题 #re](#) 文章标签: [re](#)

版权声明: 本文为博主原创文章, 遵循[CC 4.0 BY-SA](#)版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/yzl_007/article/details/121040744

版权



[CTF系列问题 同时被 2 个专栏收录](#)

36 篇文章 1 订阅

订阅专栏



[re](#)

7 篇文章 0 订阅

订阅专栏

buuctf - re 刷题记录

[buuctf - re 刷题记录](#)

- 1、easyre
- 2、reverse1
- 4、内涵的软件
- 5、新年快乐
- 6、xor
- 7、helloworld
- 8、reverse3
- 9、不一样的flag
- 10、SimpleRev
- 11、Java逆向解密
- 12、[GXYCTF2019]luck_guy
- 14、findit
- 15、[BJDCTF2020]JustRE
- 16、简单注册器
- 17、[GWCTF 2019]pyre
- 18、[ACTF新生赛2020]easyre

1、easyre

shift+f12

| Strings window | | | |
|------------------|----------|------|--------------------------|
| Address | Length | Type | String |
| .rdata:000000... | 00000005 | C | %d%d |
| .rdata:000000... | 00000017 | C | flag(this_Is_a_EaSyRe) |
| .rdata:000000... | 00000019 | C | sorry,you can't get flag |
| .rdata:000000... | 0000000F | C | std::exception |

2. reverse1

7 char v5; // [rsp+0h] [rbp-20h]
8 int j; // [rsp+24h] [rbp+4h]
9 char Str1; // [rsp+48h] [rbp+28h]
10 unsigned __int64 v8; // [rsp+128h] [rbp+108h]
11
12 v0 = &v5;
13 for (i = 82i64; i; --i)
14 {
15 *(_DWORD *)v0 = -858993460;
16 v0 += 4;
17 }
18 for (j = 0; ; ++j)
19 {
20 v8 = j;
21 v2 = j_strlen(Str2);
22 if (v8 > v2)
23 break;
24 if (Str2[j] == 111)
25 Str2[j] = 48;
26 }
27 sub_1400111D1("input the flag:");
28 sub_14001128F("%20s", &Str1);
29 v3 = j_strlen(Str2);
30 if (!strncmp(&Str1, Str2, v3))
31 sub_1400111D1("this is the right flag!\n")
32
33 else
34 sub_1400111D1("wrong flag\n");
35 sub_14001113B(&v5, &unk_140019D00);
36 return 0i64;
37 }

双击跟进str2

```
.data:000000014001C000 ;org 14001C000h  
.data:000000014001C000 ; char Str2[]  
.data:000000014001C000 Str2 db '{hello_world}',0 ; DATA XREF: sub_1400118C0+4B↑o  
.data:000000014001C000 ; sub_1400118C0+67↑o ...  
.data:000000014001C00E align 10h
```

24行到25行 o替换为0, flag{hell0_w0rld}, 虽然很简单还是写写程序多熟练

程序如下：

c语言:

```
1 #include<stdio.h>
2 #include<string.h>
3 int main()
4 {
5     int v8,v2;
6     char str2[]="hello_world";
7     for (int j = 0; ; ++j )
8     {
9         v8 = j;
10        v2 = strlen(str2);
11        if ( v8 > v2 )
12            break;
13        if ( str2[j] == 'o' )
14            str2[j] = '0';
15    }
16    printf("flag{%s}",str2);
17 }
```

C:\Users\86177\Desktop\buure\程序\reverse1.e
flag{hell0_w0rld}

Process exited with return value 0
Press any key to continue . . .

python:

```
untitled C:/Users/86177/PycharmProjects/untitled
  ctf
    RSA.py
    RSA和数e.py
    sessions.py
字符替换 x
"G:\PyCharm 2019.3\1\Scripts\python.exe" C:/Users/86177/PycharmProjects/untitled/ctf/字符替换.py
flag{hell0_w0rld}

Process finished with exit code 0
```

3、reverse2

| Address | Length | Type | String |
|--------------------------|--------|-----------------------------|--------|
| LOAD:000000... 0000001C | C | /lib64/ld-linux-x86-64.so.2 | |
| LOAD:000000... 0000000A | C | libc.so.6 | |
| LOAD:000000... 0000000F | C | _isoc99_scanf | |
| LOAD:000000... 00000005 | C | puts | |
| LOAD:000000... 00000005 | C | fork | |
| LOAD:000000... 00000011 | C | _stack_chk_fail | |
| LOAD:000000... 00000007 | C | printf | |
| LOAD:000000... 00000007 | C | strlen | |
| LOAD:000000... 00000008 | C | waitpid | |
| LOAD:000000... 00000007 | C | strcmp | |
| LOAD:000000... 00000012 | C | _libc_start_main | |
| LOAD:000000... 0000000F | C | _gmon_start_ | |
| LOAD:000000... 0000000A | C | GLIBC_2.7 | |
| LOAD:000000... 0000000A | C | GLIBC_2.4 | |
| LOAD:000000... 0000000C | C | GLIBC_2.2.5 | |
| .rodata:0000... 00000010 | C | input the flag: | |
| .rodata:0000... 00000005 | C | %20s | |
| .rodata:0000... 0000000C | C | wrong flag! | |
| .rodata:0000... 00000018 | C | this is the right flag! | |
| .eh_frame:00... 00000006 | C | /*\$\" | |
| .data:000000... 00000011 | C | hacking_for_fun} | |

简单查看下字符，疑似后半段flag，跟进对应字符查看

关键代码：

[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-oNfwtf8-1635742416948)
(C:\Users\86177\AppData\Roaming\Typora\typora-user-images\image-20211029190456094.png)]

跟进flag

[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-lcjGyqS5-1635742416950)
(C:\Users\86177\AppData\Roaming\Typora\typora-user-images\image-20211029190749227.png)]

看代码应该也是替换字符

模仿反编译的代码写下程序

[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-swsQGaV1-1635742416952)
(C:\Users\86177\AppData\Roaming\Typora\typora-user-images\image-20211029191358828.png)]

这两题都是简单的替换，只是程序一个是windows的exe，另一个是linux的可执行文件

python代码更简单，可以多学习学习，方便很多

```
str="hacking_for_fun"

str2=str.replace("i","1")
print("flag["+str2.replace("r","1")]) #str.replace(old, new[, max])
```

4、内涵的软件

动态调试一下

Debug View Structures

IDA View-EIP Pseudocode-A

```
.rdata:004250FC db 25h ; %
.rdata:004250FD db 64h ; d
.rdata:004250FE db 0C3h ; 
.rdata:004250FF db 0EBh ; 
.rdata:00425100 db 0A3h ; 
.rdata:00425101 db 0ACh ; 
.rdata:00425102 db 0C7h ; 
.rdata:00425103 db 0EBh ; 
.rdata:00425104 db 0C4h ; 
.rdata:00425105 db 0CDh ; 
.rdata:00425106 db 0D0h ; 
.rdata:00425107 db 0C4h ; 
.rdata:00425108 db 0B5h ; 
.rdata:00425109 db 0C8h ; 
.rdata:0042510A db 0B4h ; 
.rdata:0042510B db 0FDh ; 
.rdata:0042510C db 0A3h ; 
.rdata:0042510D db 0A1h ; 
.rdata:0042510E db 0Ah ; 
.rdata:0042510F db 0 ; 
.rdata:00425110 db 0 ; 
.rdata:00425111 db 0 ; 
.rdata:00425112 db 0 ; 
.rdata:00425113 db 0 ; 
.rdata:00425114 db 0 ; 
.rdata:00425115 db 0 ; 
.rdata:00425116 db 0 ; 
.rdata:00425117 db 0 ; 
.rdata:00425118 aDbapp49d3c93df db 'DBAPP{49d3c93df25caad81232130f3d2ebfad}',0 ; DATA XREF: _main_0+1F1o
.rdata:00425119 db 0 ; 
.rdata:00425140 db 0 ; 
.rdata:00425141 db 0 ; 
.rdata:00425142 db 0 ; 
```

0002510F| 0042510F: .rdata:0042510F

但是其实这里就是输出的时候没有引入参数，值开始就已经给出了

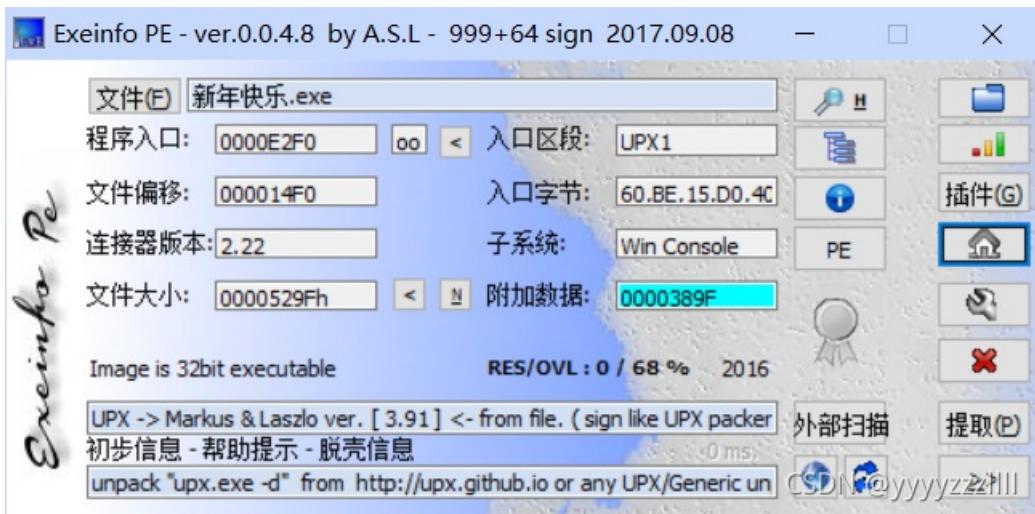
Pseudocode-A IDA View-A Pseudocode-B

```
1 int main_0()
2 {
3     int result; // eax@5
4     char v1; // [sp+Ch] [bp-4Ch]@1
5     char v2; // [sp+4Ch] [bp-Ch]@4
6     int v3; // [sp+50h] [bp-8h]@1
7     int v4; // [sp+54h] [bp-4h]@1
8
9     memset(&v1, 0xCCu, 0x4Cu);
10    v4 = 5;
11    v3 |= (int)"DBAPP{49d3c93df25caad81232130f3d2ebfad}";
12    while ( v4 > 0 )
13    {
14        printf("距离出现答案还有%d秒，请耐心等待！\n", v4);
15        sub_401000();
16        --v4;
17    }
18    printf("\n\n这里本来应该是答案的，但是细心的程序员忘记把变量写进来了，你要不回头试试看：(Y/N)\n");
19    v2 = 1;
20    scanf("%c", &v2);
21    if ( v2 == 'Y' )
22    {
23        printf("OD君爱破解或者IDA这些逆向软件都挺好的！");
24        result = sub_401000();
25    }
26    else if ( v2 == 'N' )
27    {
28        printf("那没办法了，猜是猜不出的。");
29        result = sub_401000();
30    }
31    else
32    {
33        printf("输入错误，没有提示。"); 
```

000010A4 main_0:11

5、新年快乐

查壳 upx



直接ida分析的话是打不开的，upx是压缩壳，程序基本上都被压缩加密的
脱壳

```
H:\CTFToolkit-v1.1_19.8.11\CTFToolkit-v1.1_19.8.11\逆向综合\upx-3.96-win64\upx-3.96-win64>upx.exe -d 新年快乐.exe
Ultimate Packer for eXecutables
Copyright (C) 1996 - 2020
UPX 3.96w      Markus Oberhumer, Laszlo Molnar & John Reiser   Jan 23rd 2020

File size        Ratio       Format       Name
-----<-----<-----<-----<
27807 <-     21151    76.06%    win32/pe    新年快乐.exe

Unpacked 1 file.

H:\CTFToolkit-v1.1_19.8.11\CTFToolkit-v1.1_19.8.11\逆向综合\upx-3.96-win64\upx-3.96-win64> CSDN @yyyyzzz|||
```

主函数

```
IDA View-A  Pseudocode-A  Hex View-1
1 int __cdecl main(int argc, const char **argv, const char **env)
2 {
3     char Str2[14]; // [esp+12h] [ebp-3Ah] BYREF
4     char Str1[44]; // [esp+20h] [ebp-2Ch] BYREF
5
6     __main();
7     strcpy(Str2, "HappyNewYear!");
8     memset(Str1, 0, 32);
9     printf("please input the true flag:");
10    scanf("%s", Str1);
11    if ( !strcmp(Str1, Str2, strlen(Str2)) )
12        return puts("this is true flag!");
13    else
14        return puts("wrong!");
15 }
```

CSDN @yyyyzzz|||

flag就是flag{HappyNewYear!}

6、xor

定位到关键代码

```
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     int i; // [rsp+2Ch] [rbp-124h]
4     char __b[264]; // [rsp+40h] [rbp-110h] BYREF
5
6     memset(__b, 0, 0x100ull);
7     printf("Input your flag:\n");
8     get_line(__b, 256LL);
9     if ( strlen(__b) != 33 )
10        goto LABEL_7;
11    for ( i = 1; i < 33; ++i )
12        __b[i] ^= __b[i - 1];
13    if ( !strncmp(__b, global, 0x21ull) )
14        printf("Success");
15    else
16    LABEL_7:
17        printf("Failed");
18    return 0;
19 }
```

CSDN @yyyzzzlll

找到数据

shift +E 导出数据

```
unsigned char ida_chars[] =
{
0x66, 0x0A, 0x6B, 0x0C, 0x77, 0x26, 0x4F, 0x2E, 0x40, 0x11,
0x78, 0x0D, 0x5A, 0x3B, 0x55, 0x11, 0x70, 0x19, 0x46, 0x1F,
0x76, 0x22, 0x4D, 0x23, 0x44, 0x0E, 0x67, 0x06, 0x68, 0x0F,
0x47, 0x32, 0x4F, 0x00
};
```

与前一位异或

```
data = [0x66, 0x0A, 0x6B, 0x0C, 0x77, 0x26, 0x4F, 0x2E, 0x40, 0x11,
        0x78, 0x0D, 0x5A, 0x3B, 0x55, 0x11, 0x70, 0x19, 0x46, 0x1F,
        0x76, 0x22, 0x4D, 0x23, 0x44, 0x0E, 0x67, 0x06, 0x68, 0x0F,
        0x47, 0x32, 0x4F, 0x00]
flag = chr(data[0])
for i in range(len(data)):
    flag += chr(data[i]^data[i-1])

print(flag)
```

7、helloworld

解压，反编译dex文件

main函数直接找到

```

11 package com.example.helloworld;
12 import android.os.Bundle;
13 import android.support.v7.app.ActionBarActivity;
14 import android.view.Menu;
15 import android.view.MenuItem;
16
17 public class MainActivity extends ActionBarActivity {
18     /* access modifiers changed from: protected */
19     @Override // android.support.v7.app.ActionBarActivity, android.support.v4.app.FragmentActivity
20     public void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22         setContentView(R.layout.activity_main);
23         "flag(7631a988259a00816deda84af829430a)".compareTo("xxxxxxxxxxxxxxxxxxxxxxxxxxxx");
24     }
25
26     public boolean onCreateOptionsMenu(Menu menu) {
27         getMenuInflater().inflate(R.menu.main, menu);
28         return true;
29     }
30
31     public boolean onOptionsItemSelected(MenuItem item) {
32         if (item.getItemId() == 2131034172) {
33             return true;
34         }
35     }
36
37     return super.onOptionsItemSelected(item);
38 }

```

CSDN @yyyyzzzlll

8、reverse3



无壳，32位,ida分析，简单看一眼字符，很可能是base64的简单加密

| | |
|------|--|
| Type | String |
| C | offset |
| C | <u>base64input</u> |
| C | input |
| C | ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+= |
| C | please enter the flag: |
| C | wrong flag!\n |
| C | tack around the variable ' |
| C | ' was corrupted. |
| C | he variable ' |
| C | ' is being used without being initialized. |
| C | rigth flag!\n |
| C | The value of ESP was not properly saved across a function call. This is usually a result of calling a function declared with one calling convention and using a different convention inside. |
| C | A cast to a smaller data type has caused a loss of data. If this was intentional, you should mask the source of the cast with the appropriate mask. |
| C | Stack memory... corrupted! |

定位到主函数

IDA View-A Pseudocode-A Strings Hex View-1 Structures

```

4 const char *v4; // eax
5 size_t v5; // eax
6 char v7; // [esp+0h] [ebp-188h]
7 char v8; // [esp+0h] [ebp-188h]
8 signed int j; // [esp+DCh] [ebp-ACh]
9 int i; // [esp+E8h] [ebp-A0h]
10 signed int v11; // [esp+E8h] [ebp-A0h]
11 char Destination[108]; // [esp+F4h] [ebp-94h] BYREF
12 char Str[28]; // [esp+160h] [ebp-28h] BYREF
13 char v14[8]; // [esp+17Ch] [ebp-Ch] BYREF
14
15 for ( i = 0; i < 100; ++i )
16 {
17     if ( (unsigned int)i >= 0x64 )
18         j_report_rangecheckfailure();
19     Destination[i] = 0;
20 }
21 sub_41132F("please enter the flag:", v7);
22 sub_411375("%20s", (char)Str);
23 v3 = j_strlen(Str);
24 v4 = (const char *)sub_4110BE(Str, v3, v14);
25 strncpy(Destination, v4, 0x28u);
26 v11 = j_strlen(Destination);
27 for ( j = 0; j < v11; ++j )
28     Destination[j] += j;
29 v5 = j_strlen(Destination);
30 if ( !strcmp(Destination, Str2, v5) )
31     sub_41132F("right flag!\n", v8);
32 else
33     sub_41132F("wrong flag!\n", v8);
34 return 0;
35 }
```

CSDN @yyyyzzz|||

可以适当修改来增加代码的可读性

输入的字符经过一个过程与str2比较，找到str2: e3niflH9b_C@n@dH

24行经历了一个变换

| | |
|----|--|
| 22 | sub_411375("%20s", (char)Str); |
| 23 | v3 = j_strlen(Str); |
| 24 | v4 = (const char *)sub_4110BE(Str, v3, v14); |
| 25 | strncpy(Destination, v4, 0x28u); |

定位到这个加密函数

Instruction Data Unexplored External symbol Lumina function

IDA View-A Pseudocode-B Pseudocode-A Stack of _main_0 Strings Structures Enums Imports Exports

```

28 while ( v11 > 0 )
29 {
30     byte_41A144[2] = 0;
31     byte_41A144[1] = 0;
32     byte_41A144[0] = 0;
33     for ( i = 0; i < 3 && v11 >= 1; ++i )
34     {
35         byte_41A144[i] = *v13;
36         --v11;
37         ++v13;
38     }
39     if ( !i )
40         break;
41     switch ( i )
42     {
43         case 1:
44             *((_BYTE *)v12 + v4) = aBcd...ijklmn[(int)(unsigned __int8)byte_41A144[0] >> 2];
45             v5 = v4 + 1;
46             *((_BYTE *)v12 + v5) = aBcd...ijklmn[((byte_41A144[1] & 0xF0) >> 4) | (16 * (byte_41A144[0] & 3))];
47             *((_BYTE *)v12 + ++v5) = aBcd...ijklmn[64];
48             *((_BYTE *)v12 + ++v5) = aBcd...ijklmn[64];
49             v4 = v5 + 1;
50             break;
51         case 2:
52             *((_BYTE *)v12 + v4) = aBcd...ijklmn[(int)(unsigned __int8)byte_41A144[0] >> 2];
53             v6 = v4 + 1;
54             *((_BYTE *)v12 + v6) = aBcd...ijklmn[((byte_41A144[1] & 0xF0) >> 4) | (16 * (byte_41A144[0] & 3))];
55             *((_BYTE *)v12 + ++v6) = aBcd...ijklmn[((byte_41A144[2] & 0xC0) >> 6) | (4 * (byte_41A144[1] & 0xF))];
56             *((_BYTE *)v12 + ++v6) = aBcd...ijklmn[64];
57             v4 = v6 + 1;
58             break;

```

CSDN @yyyyzzz|||

代码也看不懂，但是开头查看字符那里，有点像是base64加密

继续跟进一下

IDA View-A Pseudocode-B Pseudocode-A Stack of _main_0 Strings Structures Enums

```

.rdata:00417B2C db 0
.rdata:00417B2D db 0
.rdata:00417B2E db 0
.rdata:00417B2F db 0
.rdata:00417B30 aBcd...ijklmn db 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
.rdata:00417B30 ; DATA XREF: .text:004117E8t
.rdata:00417B30 ; .text:00411827t ...
.rdata:00417B30 db 0
.rdata:00417B72 align 4
.rdata:00417B74 ; const char aPleaseEnterThe[]


```

CSDN @yyyyzzz|||

大概率是base64了

str2 还经过了28行这里的移位，先还原回来，再进行解密即可，直接解密base64也是解不了的，@这种字符解不了

解密脚本

```

import base64
str = "e3nifiH9b_C@n@dH"

flag =''
for i in range(len(str)):
    flag+=chr(ord(str[i])-i)

data = base64.b64decode(flag)
print(data)

#b'{i_love_you}'

```

9、不一样的flag

定位到关键代码

```
1 int __cdecl __noreturn main(int argc, const char **argv, const char **envp)
2 {
3     _BYTE v3[29]; // [esp+17h] [ebp-35h] BYREF
4     int v4; // [esp+34h] [ebp-18h]
5     int v5; // [esp+38h] [ebp-14h] BYREF
6     int i; // [esp+3Ch] [ebp-10h]
7     char v7[12]; // [esp+40h] [ebp-Ch] BYREF
8
9     __main();
10    v3[26] = 0;
11    *(_WORD *)&v3[27] = 0;
12    v4 = 0;
13    strcpy(v3, "*11110100001010000101111#");
14    while ( 1 )
15    {
16        puts("you can choose one action to execute");
17        puts("1 up");
18        puts("2 down");
19        puts("3 left");
20        printf("4 right\n:");
21        scanf("%d", &v5);
22        if ( v5 == 2 )
23        {
24            ++*(_DWORD *)&v3[25];
25        }
26        else if ( v5 > 2 )
27        {
28            if ( v5 == 3 )
29            {
30                --v4;
31            }
32            else
33            {
34                if ( v5 != 4 )
35 LABEL_13:
36                    exit(1);
37                    ++v4;
38                }
39            else
40            {
41                if ( v5 != 1 )
42                    goto LABEL_13;
43                --*(_DWORD *)&v3[25];
44            }
45        for ( i = 0; i <= 1; i++ )
46        {
47            if ( *(int *)&v3[4 * i + 25] < 0 || *(int *)&v3[4 * i + 25] > 4 )
48                exit(1);
49            }
50            if ( v7[5 * *(_DWORD *)&v3[25] - 41 + v4] == 49 )
51                exit(1);
52            if ( v7[5 * *(_DWORD *)&v3[25] - 41 + v4] == 35 )
53            {
54                puts("\nok, the order you enter is the flag!");
55                exit(0);
56            }
57        }
58    }
59 }
```

CSDN @yyyyzzz|||

CSDN @yyyyzzz|||

大概率是个迷宫题

从下面这段代码可以看出

```

for ( i = 0; i <= 1; ++i )
{
    if ( *(int *)&v3[4 * i + 25] < 0 || *(int *)&v3[4 * i + 25] > 4 )
        exit(1);
}
if ( v7[5 * *(_DWORD *)&v3[25] - ')' + v4] == '1' )
    exit(1);
if ( v7[5 * *(_DWORD *)&v3[25] - ')' + v4] == '#' )
{
    puts("\nok, the order you enter is the flag!");
    exit(0);
}

```

CSDN @yyyyzzz|||

*1111

01000

01010

00010

1111#

构成如上这样一个迷宫，从*开始不能触碰1到#结束，然后输入1234分别代表上下左右

很了然了，直接输出

flag{222441144222}

10、SimpleRev

找到main函数，似乎没有什么用，大概就是输入一个值跳到一个关卡的意味

```

1 int __cdecl __noreturn main(int argc, const char **argv, const char **envp)
2 {
3     int v3; // eax
4     char v4; // [rsp+Fh] [rbp-1h]
5
6     while ( 1 )
7     {
8         while ( 1 )
9         {
10            printf("Welcome to CTF game!\nPlease input d/D to start or input q/Q to quit this program: ");
11            v4 = getchar();
12            if ( v4 != 'd' && v4 != 'D' )
13                break;
14            Decry();
15        }
16        if ( v4 == 'q' || v4 == 'Q' )
17            Exit("Welcome to CTF game!\nPlease input d/D to start or input q/Q to quit this program: ", argv);
18        puts("Input fault format!");
19        v3 = getchar();
20        putchar(v3);
21    }
22 }
```

CSDN @yyyyzzz|||

关键代码应该是Decry() 定位到该函数，

将字符拼接好

```

unsigned __int64 Decry()
{
    char v1; // [rsp+Fh] [rbp-51h]
    int v2; // [rsp+10h] [rbp-50h]
    int v3; // [rsp+14h] [rbp-4Ch]

```

```
int i; // [rsp+18h] [rbp-48h]
int len; // [rsp+1Ch] [rbp-44h]
char src[8]; // [rsp+20h] [rbp-40h] BYREF
__int64 v7; // [rsp+28h] [rbp-38h]
int v8; // [rsp+30h] [rbp-30h]
__int64 v9[2]; // [rsp+40h] [rbp-20h] BYREF
int v10; // [rsp+50h] [rbp-10h]
unsigned __int64 v11; // [rsp+58h] [rbp-8h]

v11 = __readfsqword(0x28u);
*(_QWORD *)src = 'SLCDN';
v7 = 0LL;
v8 = 0;
v9[0] = 'wodah';
v9[1] = '\0';
v10 = 0;
text = (char *)join(key3, v9); // text = killshadow
strcpy(key, key1);
 strcat(key, src); // key1=ADSFKNDCLS
v2 = 0;
v3 = 0;
getchar();
len = strlen(key);
for ( i = 0; i < len; ++i )
{
    if ( key[v3 % len] > '@' && key[v3 % len] <= 'Z' )
        key[i] = key[v3 % len] + ' ';
    ++v3;
}
printf("Please input your flag:");
while ( 1 )
{
    v1 = getchar();
    if ( v1 == '\n' )
        break;
    if ( v1 == ' ' )
    {
        ++v2;
    }
    else
    {
        if ( v1 <= '`' || v1 > 'z' )
        {
            if ( v1 > '@' && v1 <= 'Z' )
            {
                str2[v2] = (v1 - '\' - key[v3 % len] + 'a') % '\x1A' + 'a';
                ++v3;
            }
        }
        else
        {
            str2[v2] = (v1 - 39 - key[v3 % len] + 97) % 26 + 97;
            ++v3;
        }
        if ( !(v3 % len) )
            putchar(32);
        ++v2;
    }
}
if (!strcmp(text, str2))
```

```

    if ( !SCMP(text, str2) )
        puts("Congratulation!\n");
    else
        puts("Try again!\n");
    return __readfsqword(0x28u) ^ v11;
}

```

最终结果是要str2同text做比较，已经得到text = killshadow和key=ADSFKNDCLS

然后通过代码的变化推算出str2,大概逻辑就是将key转化成小写，输入v1 然后经过转化生成str2[]

反过来想，我们输入的是字母，而且是大写字母，如果转化后满足与对应的text[]相等就是我们要输入的字符

代码

python

```

s = 'ABCDEFGHIJKLMNPQRSTUVWXYZ'

key = 'adsfkndcls'
text = 'killshadow'

flag = ''
for i in range(len(text)):
    str2=text[i]
    for j in s:
        if str2 == chr((ord(j) - 39 - ord(key[i % len(key)])) + 97) % 26 + 97:
            flag+=j
print('flag{'+flag+'}')
#flag{KLDQCUDFZO}

```

11、Java逆向解密

直接进行分析

```

package defpackage;

import java.util.ArrayList;
import java.util.Scanner;

/* renamed from: Reverse reason: default package */
public class Reverse {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.println("Please input the flag : ");
        String str = s.next();
        System.out.println("Your input is : ");
        System.out.println(str);
        Encrypt(str.toCharArray());
    }

    public static void Encrypt(char[] arr) {
        int[] KEY;
        ArrayList<Integer> Resultlist = new ArrayList<>();
        for (char c : arr) {
            Resultlist.add(Integer.valueOf((c + '@') ^ 32));
        }
        ArrayList<Integer> KEYList = new ArrayList<>();
        for (int i : new int[]{180, 136, 137, 147, 191, 137, 147, 191, 148, 136, 133, 191, 134, 140, 129, 135, 191, 65}) {
            KEYList.add(Integer.valueOf(i));
        }
        System.out.println("Result:");
        if (Resultlist.equals(KEYList)) {
            System.out.println("Congratulations!");
        } else {
            System.out.println("Error!");
        }
    }
}

```

python

```

list=[180, 136, 137, 147, 191, 137, 147, 191, 148, 136, 133, 191, 134, 140, 129, 135, 191, 65]

flag=''
for i in range(len(list)):
    flag+=chr(list[i]-ord('@')^32)

print('flag{'+flag+'}')
#flag{This_is_the_flag_!}

```

它这里是java，也用java写一写吧

```

package main;

public class main {
    public static void main(String[] args)
    {
        int[] list = {180, 136, 137, 147, 191, 137, 147, 191, 148, 136, 133, 191, 134, 140, 129, 135, 191, 65};
        String flag="";
        for(int i=0;i<list.length;i++){
            char x = (char)((list[i]- '@')^32);
            flag+=x;
        }
        System.out.println("flag{" + flag + "}");
    }
}
//flag{This_is_the_flag_!}

```

12、[GXYCTF2019]luck_guy

```

1 int welcome()
2 {
3     puts("Welcome to Cyber SWAT2019.");
4     puts("Designed by Solar,wish you can enjoy it and have fun.");
5     return puts("Good luck (^_^)");
6 }

```

输入数字

```

1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     int v4; // [rsp+14h] [rbp-Ch] BYREF
4     unsigned __int64 v5; // [rsp+18h] [rbp-8h]
5
6     v5 = __readfsqword(0x28u);
7     welcome(argc, argv, envp);
8     puts("____");
9     puts("try to patch me and find flag");
.0     v4 = 0;
.1     puts("please input a lucky number");
.2     __isoc99_scanf("%d", &v4);
.3     patch_me(v4);
.4     puts("OK,see you again");
.5     return 0;
.6 }

```

CSDN @yyyyzzz|||

跟进patch_me(),如果数字为偶数，进入get_flag函数

```
1 int __fastcall patch_me(int a1)
2 {
3     if ( a1 % 2 == 1 )
4         return puts("just finished");
5     else
6         return get_flag();
7 }
```

跟进get_flag()函数

```
1 unsigned __int64 get_flag()
2 {
3     unsigned int v0; // eax
4     int i; // [rsp+4h] [rbp-3Ch]
5     int j; // [rsp+8h] [rbp-38h]
6     __int64 s; // [rsp+10h] [rbp-30h] BYREF
7     char v5; // [rsp+18h] [rbp-28h]
8     unsigned __int64 v6; // [rsp+38h] [rbp-8h]
9
10    v6 = __readfsqword(0x28u);
11    v0 = time(0LL);
12    srand(v0);
13    for ( i = 0; i <= 4; ++i )
14    {
15        switch ( rand() % 200 )
16        {
17            case 1:
18                puts("OK, it's flag:");
19                memset(&s, 0, 0x28uLL);
20                strcat((char *)&s, f1);
21                strcat((char *)&s, &f2);
22                printf("%s", (const char *)&s);
23                break;
24            case 2:
25                printf("Solar not like you");
26                break;
27            case 3:
28                printf("Solar want a girlfriend");
29                break;
30            case 4:
31                s = 0x7F666F6067756369LL;
32                v5 = 0;
```

CSDN @yyyyzzz|||

大概就是这些关键代码。接着分析,20、21行，有两个参数，`f1=GX{do_not_`

`f2指向的字符是s这里，大概是将两者拼接起来，那么关键点就在31行这一段16进制字符串了`

```

● 19     memset(&s, 0, 0x28uLL);
● 20     strcat((char *)&s, f1);
● 21     strcat((char *)&s, &f2);
● 22     printf("%s", (const char *)&s);
● 23     break;
● 24 case 2:
● 25     printf("Solar not like you");
● 26     break;
● 27 case 3:
● 28     printf("Solar want a girlfriend");
● 29     break;
● 30 case 4:
● 31     s = 0x7F666F6067756369LL;
● 32     v5 = 0;
● 33     strcat(&f2, (const char *)&s);
● 34     break;
● 35 case 5:
● 36     for ( j = 0; j <= 7; ++j )
● 37     {
● 38         if ( j % 2 == 1 )
● 39             *(&f2 + j) -= 2;
● 40         else
● 41             --*(&f2 + j);
● 42     }
● 43     break;
● 44 default:
● 45     puts("emmm,you can't find flag 23333");
● 46     break;
● 47 }
● 48 }
● 49 return __readfsqword(0x28u) ^ v6;
● 50 }
```

CSDN @yyyyzzzllll

观察到case5: 8位数的变换，将s的值转化一下就是：0x7F,0x66,0x6F,0x60,0x67,0x75,0x63,0x69

```

f1 = 'GXY{do_not_'
list=[0x7F,0x66,0x6F,0x60,0x67,0x75,0x63,0x69][::-1]

flag=''
for i in range (8):
    if i%2==1:
        s=chr(list[i]-2)
    else:
        s=chr(list[i]-1)
    flag+=s
print(f1+flag)
#GXY{do_not_hate_me}
```

代码有点长还没看，做实验去了

14、findit

下载是一个apk文件，直接解压找到class.dex，丢进jex反编译，找到主函数

```

16     public void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.activity_main);
19         final EditText edit = (EditText) findViewById(R.id.widget2);
20         final TextView text = (TextView) findViewById(R.id.widget1);
21         final char[] a = {'T', 'h', 'i', 's', 'I', 's', 'T', 'h', 'e', 'F', 'l', 'a', 'g', 'H', 'o', 'm', 'e'};
22         final char[] b = {'p', 'v', 'k', 'q', '{', 'm', '1', '6', '4', '6', '7', '5', '2', '6', '1', '3', '1', '4', 'm', '4', '9', '1', 'n', '1'};
23         ((Button) findViewById(R.id.widget3)).setOnClickListener(new View.OnClickListener() {
24             /* class com.example.findit.MainActivity$AnonymousClass1 */
25
26             public void onClick(View v) {
27                 char[] x = new char[17];
28                 char[] y = new char[38];
29                 for (int i = 0; i < 17; i++) {
30                     if ((a[i] < 'I' && a[i] >= 'A') || (a[i] < 'i' && a[i] >= 'a')) {
31                         x[i] = (char) (a[i] + 18);
32                     } else if ((a[i] < 'A' || a[i] > 'Z') && (a[i] < 'a' || a[i] > 'z')) {
33                         x[i] = a[i];
34                     } else {
35                         x[i] = (char) (a[i] - '\b');
36                     }
37                 }
38                 if (String.valueOf(x).equals(edit.getText().toString())) {
39                     for (int i2 = 0; i2 < 38; i2++) {
40                         if ((b[i2] < 'A' || b[i2] > 'Z') && (b[i2] < 'a' || b[i2] > 'z')) {
41                             y[i2] = b[i2];
42                         } else {
43                             y[i2] = (char) (b[i2] + 16);
44                             if ((y[i2] > 'Z' && y[i2] < 'a') || y[i2] > 'z') {
45                                 y[i2] = (char) (y[i2] - 26);
46                             }
47                         }
48                     }
49                     text.setText(String.valueOf(y));
50                     return;
51                 }
52                 text.setText("答案错了肿么办。。。不给你又不好意思。。。哎呀好纠结啊~~~~");
53             }
54         });
55     }
56
57     public boolean onOptionsItemSelected(MenuItem item) {
58
59     }
60
61     public void onOptionsItemSelected(MenuItem item) {
62
63     }
64
65     public void onOptionsMenuClosed(Menu menu) {
66
67     }
68
69     public void onOptionsMenuOpened(Menu menu) {
70
71     }
72
73     public void onOptionsMenuClosed(Menu menu) {
74
75     }
76
77     public void onOptionsMenuOpened(Menu menu) {
78
79     }
80
81     public void onOptionsMenuClosed(Menu menu) {
82
83     }
84
85     public void onOptionsMenuOpened(Menu menu) {
86
87     }
88
89     public void onOptionsMenuClosed(Menu menu) {
90
91     }
92
93     public void onOptionsMenuOpened(Menu menu) {
94
95     }
96
97     public void onOptionsMenuClosed(Menu menu) {
98
99     }
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
317
318
319
319
320
321
322
323
324
325
326
327
327
328
329
329
330
331
332
333
334
335
335
336
337
337
338
339
339
340
341
342
343
344
345
345
346
347
347
348
349
349
350
351
351
352
353
353
354
355
355
356
357
357
358
359
359
360
361
361
362
363
363
364
365
365
366
367
367
368
369
369
370
371
371
372
373
373
374
375
375
376
377
377
378
379
379
380
381
381
382
383
383
384
385
385
386
387
387
388
389
389
390
391
391
392
393
393
394
395
395
396
397
397
398
399
399
400
401
401
402
403
403
404
405
405
406
407
407
408
409
409
410
411
411
412
413
413
414
415
415
416
417
417
418
419
419
420
421
421
422
423
423
424
425
425
426
427
427
428
429
429
430
431
431
432
433
433
434
435
435
436
437
437
438
439
439
440
441
441
442
443
443
444
445
445
446
447
447
448
449
449
450
451
451
452
453
453
454
455
455
456
457
457
458
459
459
460
461
461
462
463
463
464
465
465
466
467
467
468
469
469
470
471
471
472
473
473
474
475
475
476
477
477
478
479
479
480
481
481
482
483
483
484
485
485
486
487
487
488
489
489
490
491
491
492
493
493
494
495
495
496
497
497
498
499
499
500
501
501
502
503
503
504
505
505
506
507
507
508
509
509
510
511
511
512
513
513
514
515
515
516
517
517
518
519
519
520
521
521
522
523
523
524
525
525
526
527
527
528
529
529
530
531
531
532
533
533
534
535
535
536
537
537
538
539
539
540
541
541
542
543
543
544
545
545
546
547
547
548
549
549
550
551
551
552
553
553
554
555
555
556
557
557
558
559
559
560
561
561
562
563
563
564
565
565
566
567
567
568
569
569
570
571
571
572
573
573
574
575
575
576
577
577
578
579
579
580
581
581
582
583
583
584
585
585
586
587
587
588
589
589
590
591
591
592
593
593
594
595
595
596
597
597
598
599
599
600
601
601
602
603
603
604
605
605
606
607
607
608
609
609
610
611
611
612
613
613
614
615
615
616
617
617
618
619
619
620
621
621
622
623
623
624
625
625
626
627
627
628
629
629
630
631
631
632
633
633
634
635
635
636
637
637
638
639
639
640
641
641
642
643
643
644
645
645
646
647
647
648
649
649
650
651
651
652
653
653
654
655
655
656
657
657
658
659
659
660
661
661
662
663
663
664
665
665
666
667
667
668
669
669
670
671
671
672
673
673
674
675
675
676
677
677
678
679
679
680
681
681
682
683
683
684
685
685
686
687
687
688
689
689
690
691
691
692
693
693
694
695
695
696
697
697
698
699
699
700
701
701
702
703
703
704
705
705
706
707
707
708
709
709
710
711
711
712
713
713
714
715
715
716
717
717
718
719
719
720
721
721
722
723
723
724
725
725
726
727
727
728
729
729
730
731
731
732
733
733
734
735
735
736
737
737
738
739
739
740
741
741
742
743
743
744
745
745
746
747
747
748
749
749
750
751
751
752
753
753
754
755
755
756
757
757
758
759
759
760
761
761
762
763
763
764
765
765
766
767
767
768
769
769
770
771
771
772
773
773
774
775
775
776
777
777
778
779
779
780
781
781
782
783
783
784
785
785
786
787
787
788
789
789
790
791
791
792
793
793
794
795
795
796
797
797
798
799
799
800
801
801
802
803
803
804
805
805
806
807
807
808
809
809
810
811
811
812
813
813
814
815
815
816
817
817
818
819
819
820
821
821
822
823
823
824
825
825
826
827
827
828
829
829
830
831
831
832
833
833
834
835
835
836
837
837
838
839
839
840
841
841
842
843
843
844
845
845
846
847
847
848
849
849
850
851
851
852
853
853
854
855
855
856
857
857
858
859
859
860
861
861
862
863
863
864
865
865
866
867
867
868
869
869
870
871
871
872
873
873
874
875
875
876
877
877
878
879
879
880
881
881
882
883
883
884
885
885
886
887
887
888
889
889
890
891
891
892
893
893
894
895
895
896
897
897
898
899
899
900
901
901
902
903
903
904
905
905
906
907
907
908
909
909
910
911
911
912
913
913
914
915
915
916
917
917
918
919
919
920
921
921
922
923
923
924
925
925
926
927
927
928
929
929
930
931
931
932
933
933
934
935
935
936
937
937
938
939
939
940
941
941
942
943
943
944
945
945
946
947
947
948
949
949
950
951
951
952
953
953
954
955
955
956
957
957
958
959
959
960
961
961
962
963
963
964
965
965
966
967
967
968
969
969
970
971
971
972
973
973
974
975
975
976
977
977
978
979
979
980
981
981
982
983
983
984
985
985
986
987
987
988
989
989
990
991
991
992
993
993
994
995
995
996
997
997
998
999
999
1000
1001
1001
1002
1003
1003
1004
1005
1005
1006
1007
1007
1008
1009
1009
1010
1011
1011
1012
1013
1013
1014
1015
1015
1016
1017
1017
1018
1019
1019
1020
1021
1021
1022
1023
1023
1024
1025
1025
1026
1027
1027
1028
1029
1029
1030
1031
1031
1032
1033
1033
1034
1035
1035
1036
1037
1037
1038
1039
1039
1040
1041
1041
1042
1043
1043
1044
1045
1045
1046
1047
1047
1048
1049
1049
1050
1051
1051
1052
1053
1053
1054
1055
1055
1056
1057
1057
1058
1059
1059
1060
1061
1061
1062
1063
1063
1064
1065
1065
1066
1067
1067
1068
1069
1069
1070
1071
1071
1072
1073
1073
1074
1075
1075
1076
1077
1077
1078
1079
1079
1080
1081
1081
1082
1083
1083
1084
1085
1085
1086
1087
1087
1088
1089
1089
1090
1091
1091
1092
1093
1093
1094
1095
1095
1096
1097
1097
1098
1099
1099
1100
1101
1101
1102
1103
1103
1104
1105
1105
1106
1107
1107
1108
1109
1109
1110
1111
1111
1112
1113
1113
1114
1115
1115
1116
1117
1117
1118
1119
1119
1120
1121
1121
1122
1123
1123
1124
1125
1125
1126
1127
1127
1128
1129
1129
1130
1131
1131
1132
1133
1133
1134
1135
1135
1136
1137
1137
1138
1139
1139
1140
1141
1141
1142
1143
1143
1144
1145
1145
1146
1147
1147
1148
1149
1149
1150
1151
1151
1152
1153
1153
1154
1155
1155
1156
1157
1157
1158
1159
1159
1160
1161
1161
1162
1163
1163
1164
1165
1165
1166
1167
1167
1168
1169
1169
1170
1171
1171
1172
1173
1173
1174
1175
1175
1176
1177
1177
1178
1179
1179
1180
1181
1181
1182
1183
1183
1184
1185
1185
1186
1187
1187
1188
1189
1189
1190
1191
1191
1192
1193
1193
1194
1195
1195
1196
1197
1197
1198
1199
1199
1200
1201
1201
1202
1203
1203
1204
1205
1205
1206
1207
1207
1208
1209
1209
1210
1211
1211
1212
1213
1213
1214
1215
1215
1216
1217
1217
1218
1219
1219
1220
1221
1221
1222
1223
1223
1224
1225
1225
1226
1227
1227
1228
1229
1229
1230
1231
1231
1232
1233
1233
1234
1235
1235
1236
1237
1237
1238
1239
1239
1240
1241
1241
1242
1243
1243
1244
1245
1245
1246
1247
1247
1248
1249
1249
1250
1251
1251
1252
1253
1253
1254
1255
1255
1256
1257
1257
1258
1259
1259
1260
1261
1261
1262
1263
1263
1264
1265
1265
1266
1267
1267
1268
1269
1269
1270
1271
1271
1272
1273
1273
1274
1275
1275
1276
1277
1277
1278
1279
1279
1280
1281
1281
1282
1283
1283
1284
1285
1285
1286
1287
1287
1288
1289
1289
1290
1291
1291
1292
1293
1293
1294
1295
1295
1296
1297
1297
1298
1299
1299
1300
1301
1301
1302
1303
1303
1304
1305
1305
1306
1307
1307
1308
1309
1309
1310
1311
1311
1312
1313
1313
1314
1315
1315
1316
1317
1317
1318
1319
1319
1320
1321
1321
1322
1323
1323
1324
1325
1325
1326
1327
1327
1328
1329
1329
1330
1331
1331
1332
1333
1333
1334
1335
1335
1336
1337
1337
1338
1339
1339
1340
1341
1341
1342
1343
1343
1344
1345
1345
1346
1347
1347
1348
1349
1349
1350
1351
1351
1352
1353
1353
1354
1355
1355
1356
1357
1357
1358
1359
1359
1360
1361
1361
1362
1363
1363
1364
1365
1365
1366
1367
1367
1368
1369
1369
1370
1371
1371
1372
1373
1373
1374
1375
1375
1376
1377
1377
1378
1379
1379
1380
1381
1381
1382
1383
1383
1384
1385
1385
1386
1387
1387
1388
1389
1389
1390
1391
1391
1392
1393
1393
1394
1395
1395
1396
1397
1397
1398
1399
1399
1400
1401
1401
1402
1403
1403
1404
1405
1405
1406
1407
1407
1408
1409
1409
1410
1411
1411
1412
1413
1413
1414
1415
1415
1416
1417
1417
1418
1419
1419
1420
1421
1421
1422
1423
1423
1424
1425
1425
1426
1427
1427
1428
1429
1429
1430
1431
1431
1432
1433
1433
1434
1435
1435
1436
1437
1437
1438
1439
1439
1440
1441
1441
1442
1443
1443
1444
1445
1445
1446
1447
1447
1448
1449
1449
1450
1451
1451
1452
1453
1453
1454
1455
1455
1456
1457
1457
1458
1459
1459
1460
1461
1461
1462
1463
1463
1464
1465
1465
1466
1467
1467
1468
1469
1469
1470
1471
1471
1472
1473
1473
1474
1475
1475
1476
1477
1477
1478
1479
1479
1480
1481
1481
1482
1483
1483
1484
1485
1485
1486
1487
1487
1488
1489
1489
1490
1491
1491
1492
1493
1493
1494
1495
1495
1496
1497
1497
1498
1499
1499
1500
1501
1501
1502
1503
1503
1504
1505
1505
1506
1507
1507
1508
1509
1509
1510
1511
1511
1512
1513
1513
1514
1515
1515
1516
1517
1517
1518
1519
1519
1520
1521
1521
1522
1523
1523
1524
1525
1525
1526
1527
1527
1528
1529
1529
1530
1531
1531
1532
1533
1533
1534
1535
1535
1536
1537
1537
1538
1539
1539
1540
1541
1541
1542
1543
1543
1544
1545
1545
1546
1547
1547
1548
1549
1549
1550
1551
1551
1552
1553
1553
1554
1555
1555
1556
1557
1557
1558
1559
1559
1560
1561
1561
1562
1563
1563
1564
1565
1565
1566
1567
1567
1568
1569
1569
1570
1571
1571
1572
1573
1573
1574
1575
1575
1576
1577
1577
1578
1579
1579
1580
1581
1581
1582
1583
1583
1584
1585
1585
1586
1587
1587
1588
1589
1589
1590
1591
1591
1592
1593
1593
1594
1595
1595
1596
1597
1597
1598
1599
1599
1600
1601
1601
1602
1603
1603
1604
1605
1605
1606
1607
1607
1608
1609
1609
1610
1611
1611
1612
1613
1613
1614
1615
1615
1616
1617
1617
1618
1619
1619
1620
1621
1621
1622
1623
1623
1624
1625
1625
1626
1627
1627
1628
1629
1629
1630
1631
1631
1632
1633
1633
1634
1635
1635
1636
1637
1637
1638
1639
1639
1640
1641
1641
1642
1643
1643
1644
1645
1645
1646
1647
1647
1648
1649
1649
1650
1651
1651
1652
1653
1653
1654
1655
1655
1656
1657
1657
1658
1659
1659
1660
1661
1661
1662
1663
1663
1664
1665
1665
1666
1667
1667
1668
1669
1669
1670
1671
1671
1672
1673
1673
1674
1675
1675
1676
1677
1677
1678
1679
1679
1680
1681
1681
1682
1683
```

```

        for (int i2 = 0, i2 < 30, i2++) {
            if ((b[i2] < 'A' || b[i2] > 'Z') && (b[i2] < 'a' || b[i2] > 'z')) {
                y[i2] = b[i2];
            } else {
                y[i2] = (char) (b[i2] + 16);
                if ((y[i2] > 'Z' && y[i2] < 'a') || y[i2] >= 'z') {
                    y[i2] = (char) (y[i2] - 26);
                }
            }
        }
        text.setText(String.valueOf(y));
        return;
    }
    text.setText("答案错了肿么办。。。不给你又不好意思。。。哎呀好纠结啊~~~");
}
);
}

public boolean onOptionsItemSelected(MenuItem item) {
    if (item.getItemId() == 2131034176) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}
}

```

模仿代码写出解密

```

package main;

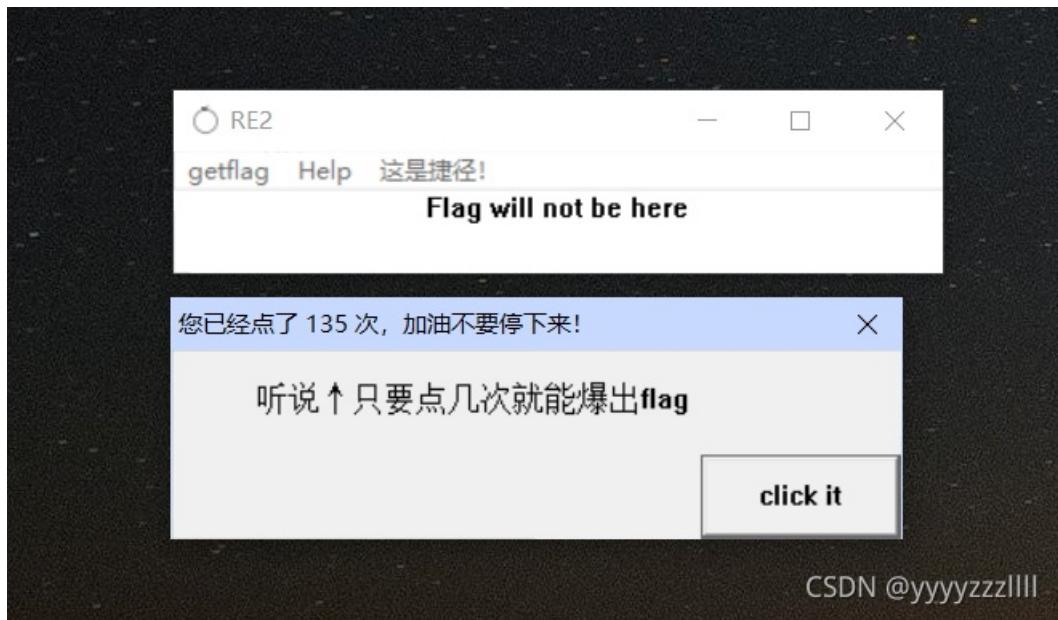
public class findit {
    public static void main(String[] args)
    {
        final char[] b = {'p', 'v', 'k', 'q', '{', 'm', '1', '6', '4', '6', '7', '5', '2', '6', '2', '0', '3', '3', '1', '4', 'm', '4', '9', 'l', 'n', 'p', '7', 'p', '9', 'm', 'n', 'k', '2', '8', 'k', '7', '5', '}};

        char[] y = new char[38];
        for (int i2 = 0; i2 < 38; i2++) {
            if ((b[i2] < 'A' || b[i2] > 'Z') && (b[i2] < 'a' || b[i2] > 'z')) {
                y[i2] = b[i2];
            } else {
                y[i2] = (char) (b[i2] + 16);
                if ((y[i2] > 'Z' && y[i2] < 'a') || y[i2] >= 'z') {
                    y[i2] = (char) (y[i2] - 26);
                }
            }
        }
        for(int j=0;j<38;j++){
            System.out.print(y[j]);
        }
    }
}
//flag{c164675262033b4c49bdf7f9cda28a75}

```

15、[BJDCTF2020]JustRE

打开后一个这软件



ida看看，发现一个字符，但是%d%d这不知道是什么鬼，继续分析

ion Data Unexplored External symbol Lumina function

IDA View-A Strings Hex View-1 Structures Enums

```
.data:00407028 ; _PVFV dword_407028
.data:00407028 dword_407028 dd 0 ; DATA XREF: _doexit:loc_401FDD↑o
.data:0040702C align 10h
.data:00407030 ; char aBjdDD2069a4579[]
.data:00407030 aBjdDD2069a4579 db ' BJD{%d%d2069a45792d233ac}',0 ; DATA XREF: DialogFunc+5A↑o
.data:00407030 align 4 ; DATA XREF: DialogFunc+5A↑o
.data:0040704B align 4
.data:0040704C ; char Format
.data:0040704C Format db 0A1h ; DATA XREF: DialogFunc+5E↑o
```

CSDN @yyyyzzz|||

大概是点击19999次出flag？？按照代码意思将19999和0填入%d%d BJD{1999902069a45792d233ac}

->flag{1999902069a45792d233ac}

```
INT_PTR __stdcall DialogFunc(HWND hWnd, UINT a2, WPARAM a3, LPARAM a4)
{
    CHAR String[100]; // [esp+0h] [ebp-64h] BYREF

    if ( a2 != 272 )
    {
        if ( a2 != 273 )
            return 0;
        if ( (_WORD)a3 != 1 && (_WORD)a3 != 2 )
        {
            sprintf(String, &Format, ++dword_4099F0);
            if ( dword_4099F0 == 19999 )
            {
                sprintf(String, " BJD{%d%d2069a45792d233ac}", 19999, 0);
                SetWindowTextA(hWnd, String);
                return 0;
            }
            SetWindowTextA(hWnd, String);
            return 0;
        }
        EndDialog(hWnd, (unsigned __int16)a3);
    }
    return 1;
}
```

CSDN @yyyyzzz|||

和findit类似，直接找主函数

```
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends ActionBarActivity {
    @Override // android.support.v7.app.ActionBarActivity, android.support.v4.app.FragmentActivity
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        if (savedInstanceState == null) {
            getSupportFragmentManager().beginTransaction().add(R.id.container, new PlaceholderFragment()).commit();
        }
        final TextView textView = (TextView) findViewById(R.id.textView1);
        final EditText editview = (EditText) findViewById(R.id.editText1);
        ((Button) findViewById(R.id.button1)).setOnClickListener(new View.OnClickListener() {
            /* class com.example.flag.MainActivity$AnonymousClass1 */
        });
    }

    public void onClick(View v) {
        int flag = 1;
        String xx = editview.getText().toString();
        if (!(xx.length() == 32 && xx.charAt(31) == 'a' && xx.charAt(1) == 'b' && (xx.charAt(0) + xx.charAt(2)) - 48 == 56)) {
            flag = 0;
        }
        if (flag == 1) {
            char[] x = "dd2940c04462b4dd7c450528835cca15".toCharArray();
            x[2] = (char) ((x[2] + x[3]) - 50);
            x[4] = (char) ((x[2] + x[5]) - 48);
            x[30] = (char) ((x[31] + x[9]) - 48);
            x[14] = (char) ((x[27] + x[28]) - 97);
            for (int i = 0; i < 16; i++) {
                char a = x[31 - i];
                x[31 - i] = x[i];
                x[i] = a;
            }
            textView.setText("flag{" + String.valueOf(x) + "}");
            return;
        }
        textView.setText("输入注册码错误");
    }
}

public boolean onCreateOptionsMenu(Menu menu) {
```

CSDN @yyyyzzz

直接看验证部分就可以其实，这种筛选部分直接可以不看，过滤不满足的字符然后跳转到return;

```
String xx = editview.getText().toString();

if (!(xx.length() == 32 && xx.charAt(31) == 'a' && xx.charAt(1) == 'b' && (xx.charAt(0) + xx.charAt(2)) - 48 == 56)) {
    flag = 0;
}

package main;

public class 简单注册表 {
    public static void main(String[] args) {
        int flag = 1;
        if (flag == 1) {
            char[] x = "dd2940c04462b4dd7c450528835cca15".toCharArray();
            x[2] = (char) ((x[2] + x[3]) - 50);
            x[4] = (char) ((x[2] + x[5]) - 48);
            x[30] = (char) ((x[31] + x[9]) - 48);
            x[14] = (char) ((x[27] + x[28]) - 97);
            for (int i = 0; i < 16; i++) {
                char a = x[31 - i];
                x[31 - i] = x[i];
                x[i] = a;
            }
            System.out.print("flag{" + String.valueOf(x) + "}");
        }
    }
}

//flag{59acc538825054c7de4b26440c0999dd}
```

17、[GWCTF 2019]pyre

pyc文件，直接在线反编译成py，原理后面再学一下

```

#!/usr/bin/env python
# visit https://tool.lu/pyc/ for more information
print 'Welcome to Re World!'
print 'Your input1 is your flag~'
l = len(input1)
for i in range(l):
    num = ((input1[i] + i) % 128 + 128) % 128
    code += num

for i in range(l - 1):
    code[i] = code[i] ^ code[i + 1]

print code
code = [
    '\x1f',
    '\x12',
    '\x1d',
    '(',
    '0',
    '4',
    '\x01',
    '\x06',
    '\x14',
    '4',
    ',',
    '\x1b',
    'U',
    '?',
    'o',
    '6',
    '*',
    ':',
    '\x01',
    'D',
    ';',
    '%',
    '\x13']

```

解密：

```

code = ['\x1f', '\x12', '\x1d', '(', '0', '4', '\x01', '\x06', '\x14', '4', ',', '\x1b', 'U', '?', 'o', '6', '*', ':', '\x01', 'D',
';', '%', '\x13']

l = len(code)
flag=''
for i in range(l-2, -1, -1):
    code[i]=chr(ord(code[i])^ord(code[i+1]))

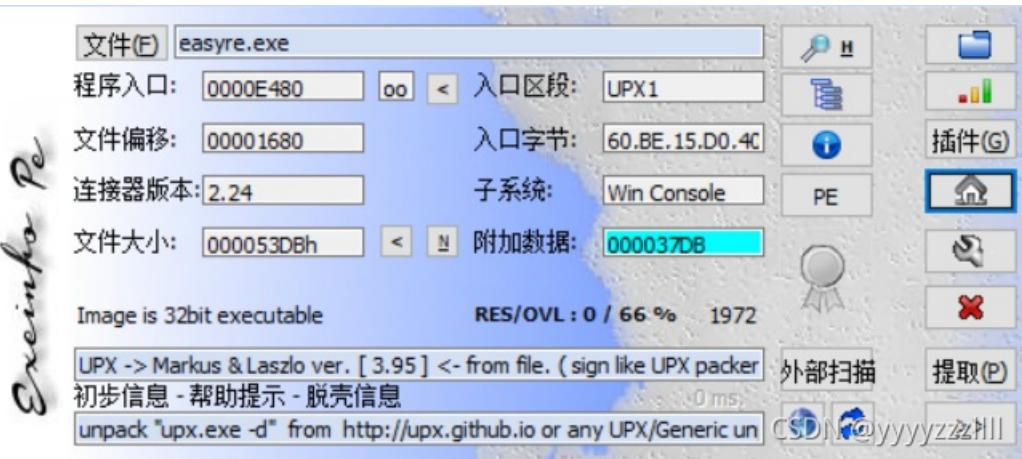
for i in range(len(code)):
    flag+=chr((ord(code[i])-i)%128)

print(flag)
# GWHT{Just_Re_1s_Ha66y!}

```

18、[CTF新生赛2020]easyre

UPX壳，脱去



```

/C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.18363.1556]
(c) 2019 Microsoft Corporation。保留所有权利。
/H: \Tool\Reverse\upx-3.96-win64\upx-3.96-win64>upx.exe -d easyre.exe
          Ultimate Packer for eXecutables
          Copyright (C) 1996 - 2020
/UPX 3.96w      Markus Oberhumer, Laszlo Molnar & John Reiser   Jan 23rd 2020
/----- File size ----- Ratio ----- Format ----- Name -----
/----- 28123 <- 21467 76.33% win32/pe easyre.exe
/----- Unpacked 1 file.
/H: \Tool\Reverse\upx-3.96-win64\upx-3.96-win64>

```

CSDN @yyyyzzz|||

然后IDA32打开，貌似有个什么加密

| Address | Length | Type | String |
|--------------------|----------|------|--|
| \$.data:00402001 | 0000005F | C |]jzyxwwutsrqponmlkjihgfedcba`_^]\\\\[ZYXWVUTSRQPONMLKJIHGFECDBA@?>=<;9876543210/-,+*)(`%\$# !" |
| \$.rdata:004030... | 0000000E | C | libgcc-13.dll |
| \$.rdata:004030... | 00000014 | C | _Jv_RegisterClasses |
| \$.rdata:004030... | 0000000E | C | Please input: |
| \$.rdata:004030... | 00000011 | C | You are correct! |
| \$.rdata:004030... | 00000018 | C | Mingw runtime failure:\n |
| \$.rdata:004030... | 00000031 | C | VirtualQuery failed for %d bytes at address %p\n |
| \$.rdata:004030... | 00000032 | C | Unknown pseudo relocation protocol version %d.\n |
| \$.rdata:004030... | 0000002A | C | Unknown pseudo relocation bit size %d.\n |
| \$.rdata:004030... | 00000013 | C | GCC: (tdm-1) 4.9.2 |
| \$.rdata:004031... | 00000013 | C | GCC: (tdm-1) 4.9.2 |
| \$.rdata:004031... | 00000013 | C | GCC: (tdm-1) 4.9.2 |
| \$.rdata:004031... | 00000013 | C | GCC: (tdm-1) 4.9.2 |

CSDN @yyyyzzz|||

定位到主函数

```

int __cdecl main(int argc, const char **argv, const char **envp)
{
    char v4[12]; // [esp+12h] [ebp-2Eh] BYREF
    int v5[3]; // [esp+1Eh] [ebp-22h]
    char v6[5]; // [esp+2Ah] [ebp-16h] BYREF
    int v7; // [esp+2Fh] [ebp-11h]
    int v8; // [esp+33h] [ebp-Dh]
    int v9; // [esp+37h] [ebp-9h]
    char v10; // [esp+3Bh] [ebp-5h]
    int i; // [esp+3Ch] [ebp-4h]

    __main();
    qmemcpy(v4, "*F'\"N,\"(I?+@", sizeof(v4));
    printf("Please input:");
    scanf("%s", v6);
    if ( v6[0] != 65 || v6[1] != 67 || v6[2] != 84 || v6[3] != 70 || v6[4] != 123 || v10 != 125 )
        return 0;
    v5[0] = v7;
    v5[1] = v8;
    v5[2] = v9;
    for ( i = 0; i <= 11; ++i )
    {
        if ( v4[i] != _data_start_[*((char *)v5 + i) - 1] ) //v4[i]得与_data_start处理后相等
            return 0;
    }
    printf("You are correct!");
    return 0;
}

```

需要是以ASCTF{}包含一个文件，大概中间是12个字符，与“][]{zyxwvutsrqponmlkjihgfedcba`_]
[ZYXWVUTSRQPONMLKJIHGfedcba@?>=”进行加密

找到那些字符

```

sub    esp, 40h
call   __main
mov    byte ptr [esp+12h], 2Ah ; '*'
mov    byte ptr [esp+13h], 46h ; 'F'
mov    byte ptr [esp+14h], 27h ; ' '
mov    byte ptr [esp+15h], 22h ; ' '
mov    byte ptr [esp+16h], 4Eh ; 'N'
mov    byte ptr [esp+17h], 2Ch ; ','
mov    byte ptr [esp+18h], 22h ; ' '
mov    byte ptr [esp+19h], 28h ; '('
mov    byte ptr [esp+1Ah], 49h ; 'I'
mov    byte ptr [esp+1Bh], 3Fh ; '?'
mov    byte ptr [esp+1Ch], 2Bh ; '+'
mov    byte ptr [esp+1Dh], 40h ; '@'
mov    dword ptr [esp], offset Format ; "Please input:"
call   _printf
lea    eax, [esp+2Ah]
mov    [esp+4], eax
mov    dword ptr [esp], offset as ; "%s"

```

第一次做这种，抄了个脚本

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

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

print ("flag{" + flag + "}")
#flag{U9X_1S_W6@T?}
```