

# [BUGKU][CTF][Reverse][2020] Reverse writeup 1-7 暂时肝不动了

原创

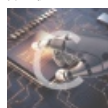
CryptWinter 于 2021-01-14 15:59:44 发布 256 收藏 1

分类专栏: CTF 文章标签: BUGKU CTF 2020 Reverse writeup

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/dadongwudi/article/details/112599804>

版权



[CTF 专栏收录该内容](#)

17 篇文章 2 订阅

订阅专栏

Reverse 入门逆向

步骤: ida main函数 按R

```
public _main
_main proc near

    argc= dword ptr 8
    argv= dword ptr 0Ch
    envp= dword ptr 10h

; __unwind {
push    ebp
mov     ebp, esp
and     esp, 0FFFFFF0h
sub     esp, 30h
call   __main
mov     dword ptr [esp], offset aHiThisIsABabyr ; "Hi~ this is a babyre"
call   _printf
mov     byte ptr [esp+2Fh], 'f'
mov     byte ptr [esp+2Eh], 'l'
mov     byte ptr [esp+2Dh], 'a'
mov     byte ptr [esp+2Ch], 'g'
mov     byte ptr [esp+2Bh], '{'
mov     byte ptr [esp+2Ah], 'R'
mov     byte ptr [esp+29h], 'e'
mov     byte ptr [esp+28h], '-'
mov     byte ptr [esp+27h], 'l'
mov     byte ptr [esp+26h], 's'
mov     byte ptr [esp+25h], '-'
mov     byte ptr [esp+24h], 'S'
mov     byte ptr [esp+23h], '0'
mov     byte ptr [esp+22h], '-'
mov     byte ptr [esp+21h], 'C'
mov     byte ptr [esp+20h], '0'
mov     byte ptr [esp+1Fh], '0'
mov     byte ptr [esp+1Eh], 'L'
mov     byte ptr [esp+1Dh], '}'
mov     eax, 0
leave
retn
; } // starts at 401460
```

Reverse sign in

关键字:

知识点: Android逆向分析。(常用工具: 安卓模拟器、JEB、Cyberchef、Androidkiller)

步骤:

少琳.

1. 用jeb打开，找到MainActivity，右键解析成java分析，需要反转后toString()

```
public void checkPassword(String arg4) {
    if(arg4.equals(new String(Base64.decode(new StringBuffer(this.getFlag()).reverse().toString(), 0)))) {
        this.showMsgToast("Congratulations !");
    }
    else {
        this.showMsgToast("Try again.");
    }
}

private String getFlag() {
    return this.getBaseContext().getString(0x7F0B0020);
}

protected void onCreate(Bundle arg2) {
    super.onCreate(arg2);
    this.setContentViews(0x7F09001B);
    this.findViewById(0x7F07004D).setOnClickListener(new View.OnClickListener() {
        public void onClick(View arg2) {
            MainActivity.this.checkPassword(MainActivity.this.findViewById(0x7F070045).getText().toString());
        }
    });
}
```

2. 用jeb打开，找到MainActivity，右键解析成java分析

```
<string id="0x7f0b0010" name="abc_font_family_headline_material" type="string" />
<string id="0x7f0b0011" name="abc_font_family_menu_material" type="string" />
<string id="0x7f0b0012" name="abc_font_family_subhead_material" type="string" />
<string id="0x7f0b0013" name="abc_font_family_title_material" type="string" />
<string id="0x7f0b0014" name="abc_search_hint" type="string" />
<string id="0x7f0b0015" name="abc_searchview_description_clear" type="string" />
<string id="0x7f0b0016" name="abc_searchview_description_query" type="string" />
<string id="0x7f0b0017" name="abc_searchview_description_search" type="string" />
<string id="0x7f0b0018" name="abc_searchview_description_submit" type="string" />
<string id="0x7f0b0019" name="abc_searchview_description_voice" type="string" />
<string id="0x7f0b001a" name="abc_shareactionprovider_share_with" type="string" />
<string id="0x7f0b001b" name="abc_shareactionprovider_share_with_application" type="string" />
<string id="0x7f0b001c" name="abc_toolbar_collapse_description" type="string" />
<string id="0x7f0b001d" name="app_name" type="string" />
<string id="0x7f0b001e" name="search_menu_title" type="string" />
<string id="0x7f0b001f" name="status_bar_notification_info_overflow" type="string" />
<string id="0x7f0b0020" name="toString" type="string" />
<string id="0x7f0c0000" name="AlertDialog.AppCompat" type="style" />
<string id="0x7f0c0001" name="AlertDialog.AppCompat.Light" type="style" />
```

```
sans-serif</string>
<string name="abc_font_family_subhead_material">
    sans-serif</string>
<string name="abc_font_family_title_material">
    sans-serif-medium</string>
<string name="abc_search_hint">
    Search.</string>
<string name="abc_searchview_description_clear">
    Clear query</string>
<string name="abc_searchview_description_query">
    Search query</string>
<string name="abc_searchview_description_search">
    Search</string>
<string name="abc_searchview_description_submit">
    Submit query</string>
<string name="abc_searchview_description_voice">
    Voice search</string>
<string name="abc_shareactionprovider_share_with">
    Share with</string>
<string name="abc_shareactionprovider_share_with_application">
    Share with %s</string>
<string name="abc_toolbar_collapse_description">
    Collapse</string>
<string name="app_name">
    sdnisc_apk1</string>
<string name="search_menu_title">
    Search</string>
<string name="status_bar_notification_info_overflow">
    999</string>
<string name="toString">
    qq1V17u0z817hF17f1Xdwk3Y1k2Y7TX7T+37hvm7</string>
```

3.在线或者使用脚本反转 ,base64解码

```
a = '991YiZW0z81ZhFjZfJXdwk3X1k2XzIXZIt3ZhxmZ'  
b = ''  
b = a[::-1]  
print(b)
```

```
ZmxhZ3tIZXlzM2k1X3kwdXJfZjFhZ18zOWZiY199|
```

编码

base64



```
flag{Her3_i5_y0ur_flag_39fbc_}
```

<https://blog.csdn.net/dadongwudi>

参考链接:<https://www.cnblogs.com/myqzs/p/13724482.html>

## Reverse EASY re

关键字: ida

步骤:

1.main f5 F5翻译为伪C代码

```
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     int v3; // eax
4     __int128 v5; // [esp+0h] [ebp-44h]
5     int64 v6; // [esp+10h] [ebp-34h]
6     int v7; // [esp+18h] [ebp-2Ch]
7     __int16 v8; // [esp+1Ch] [ebp-28h]
8     char v9; // [esp+20h] [ebp-24h]
9
10    _mm_storeu_si128((__m128i *)&v5, _mm_loadu_si128((const __m128i *)&xmmword_413E34));
11    v7 = 0;
12    v6 = qword_413E44;
13    v8 = 0;
14    printf("欢迎来到DUTCTF呦\n");
15    printf("这是一道很可爱很简单的逆向题呦\n");
16    printf("输入flag吧:");
17    scanf("%s", &v9);
18    v3 = strcmp((const char *)&v5, &v9);
19    if ( v3 )
20        v3 = -(v3 < 0) | 1;
21    if ( v3 )
22        printf(aFlag_0);
23    else
24        printf((const char *)&unk_413E90);
25    system("pause");
26    return 0;
27 }
```

2.strcmp()对面输入的值是否等于xmmword\_413E34位置的值，双击xmmword\_413E34 跟过去，发现了flag（按R显示字符串）

```
.rdata:00413E34 xmmword_413E34    xmmword '0tem0c1eW{FTCTUD}'
.rdata:00413E34                                ; DATA XREF: _main+10↑r
.rdata:00413E44 qword_413E44    dq ' }FTCTUD'                ; DATA XREF: _main+27↑r
.rdata:00413E4C ; char aDutctf[]
.rdata:00413E4C aDutctf        db '欢迎来到DUTCTF呦',0Ah,0  ; DATA XREF: _main+1A↑o
.rdata:00413E5E                                align 10h
.rdata:00413E60 ; char asc_413E60[]
.rdata:00413E60 asc_413E60     db '这是一道很可爱很简单的逆向题呦',0Ah,0
.rdata:00413E60                                ; DATA XREF: _main+3D↑o
.rdata:00413E80 ; char aFlag[]
.rdata:00413E80 aFlag         db '输入flag吧:',0          ; DATA XREF: _main+47↑o
.rdata:00413E8C ; char aS[]
.rdata:00413E8C aS           db '%s',0                  ; DATA XREF: _main+55↑o
.rdata:00413E8F                                align 10h
.rdata:00413E90 unk_413E90     db 66h ; f                  ; DATA XREF: _main+91↑o
.rdata:00413E91                                db 6Ch ; l
.rdata:00413E92                                db 61h ; a
```

3.字符串拼接后反转，得到flag

DUTCTF{We1c0met0DUTCTF}

思路二：解压得到一个可执行文件，然后用Notepad++打开，直接搜索DUTCTF即可得到

## Reverse Easy\_vb

关键字: ida

步骤:

ida 打开 往下翻 MCTF 替换 flag

```
ThunRTMain .text
.text:00401980 dd 0FCFB3D2Eh, 1068A0FAh, 838A7h, 0B571332Bh, 735C3A44h
.text:00401980 dd 5C74666Fh, 0ABBE4256h, 0E6B0F2BCh, 3642565Ch, 424C4F2Eh
.text:00401980 dd 0
.text:004019AC dword_4019AC ; DATA XREF: .text:00402254↓o
.text:004019AC ; .text:00402260↓o ...
.text:004019B8 dd 6, 9, 401990h, 4019ACh, 4042D0h, 2 dup(0)
.text:004019D4 dword_4019D4 ; DATA XREF: .text:00402264↓o
.text:004019D4 dd 577A020h, 33AD4EDAh, 11CF6699h, 0AA00CB7h, 93D36000h
.text:004019D4 ; DATA XREF: .text:00402264↓o
.text:004019D4 dd 6562614Ch, 336Ch, 33AD4F3Ah, 11CF6699h, 0AA00CB7h
.text:00401A08 aLabel1 db 'Label1',0
.text:00401A0F align 10h
.text:00401A10 aLabel2 db 'Label2',0
.text:00401A17 align 4
.text:00401A18 dword_401A18 ; DATA XREF: .text:0040227C↓o
.text:00401A18 dd 33AD4EE2h, 11CF6699h, 0AA00CB7h, 93D36000h, 74786554h
.text:00401A18 ; DATA XREF: .text:0040227C↓o
.text:00401A18 dd 31h, 6D6D6F43h, 32646E61h, 0
.text:00401A3C dword_401A48 ; DATA XREF: .text:00402398↓o
.text:00401A48 ; .text:0040241E↓o
.text:00401A48 ; DATA XREF: .text:004023A9↓o
.text:00401A5C aMctfN3tRev1sE4: text "UTF-16LE", 'MCTF{ N3t Rev 1s E4y }',0
.text:00401A5C ; .text:004023A9↓o
.text:00401A8C aTryAgain: ; DATA XREF: .text:00402473↓o
.text:00401A90 text "UTF-16LE", 'Try again!',0
.text:00401AA6 align 4
.text:00401AA8 dword_401AA8 ; DATA XREF: .text:0040264F↓o
.text:00401AA8 ; .text:00402B0B↓o ...
```

## Reverse Timer(阿里CTF)

关键字:

知识点:

步骤:

1. 下载文件发现是apk，先安装运行下发现有一个倒计时，只是时间为200000秒。猜测是让时间走完获取flag。

2. JEB查看

```
package net.bluelotus.tomorrow.easyandroid;

import android.os.Bundle;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    int beg = (((int) (System.currentTimeMillis() / 1000)) + 200000);
    int k = 0;
    int now;
    long t = 0;

    public native String stringFromJNI2(int i);

    public static boolean is2(int n) {
        if (n <= 3) {
            if (n > 1) {
                return true;
            }
            return false;
        } else if (n % 2 == 0 || n % 3 == 0) {
            return false;
        } else {
            int i = 5;
        }
    }
}
```

```

        while (i * i <= n) {
            if (n % i == 0 || n % (i + 2) == 0) {
                return false;
            }
            i += 6;
        }
        return true;
    }
}

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView((int) R.layout.activity_main);
    final TextView tv1 = (TextView) findViewById(R.id.textView2);
    final TextView tv2 = (TextView) findViewById(R.id.textView3);
    final Handler handler = new Handler();
    handler.postDelayed(new Runnable() {
        public void run() {
            MainActivity.this.t = System.currentTimeMillis();
            MainActivity.this.now = (int) (MainActivity.this.t / 1000);
            MainActivity.this.t = 1500 - (MainActivity.this.t % 1000);
            tv2.setText("AliCTF");
            if (MainActivity.this.beg - MainActivity.this.now <= 0) {
                tv1.setText("The flag is:");
                tv2.setText("alictf{" + MainActivity.this.stringFromJNI2(MainActivity.this.k) + "}");
            }
            MainActivity mainActivity;
            if (MainActivity.is2(MainActivity.this.beg - MainActivity.this.now)) {
                mainActivity = MainActivity.this;
                mainActivity.k += 100;
            } else {
                mainActivity = MainActivity.this;
                mainActivity.k--;
            }
            tv1.setText("Time Remaining(s):" + (MainActivity.this.beg - MainActivity.this.now));
            handler.postDelayed(this, MainActivity.this.t);
        }
    }, 0);
}

public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}

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

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

```

首先初始化了beg为当前时间加上200000。`(System.currentTimeMillis() / 1000)`是获得系统的时间，单位为毫秒，转换为秒。

看 `onCreate` 方法，找到关键处

```
if (MainActivity.this.beg - MainActivity.this.now <= 0) {
    tv1.setText("The flag is:");
    tv2.setText("alictf{" +
MainActivity.this.stringFromJNI2(MainActivity.this.k) + "}");
}
```

所以 `MainActivity.this.beg - MainActivity.this.now <= 0` 就是过了得时间。如果过了200000秒则出现flag。flag是使用native层来打印。

思路：能不能直接跳过200000秒直接出现flag呢？

有一个关键变量 `k`，往下看，看看k有没有什么运算。

```
if (MainActivity.is2(MainActivity.this.beg - MainActivity.this.now)) {
    mainActivity = MainActivity.this;
    mainActivity.k += 100;
} else {
    mainActivity = MainActivity.this;
    mainActivity.k--;
}
```

将差值用is2函数判断，如果true，就k+100，如果false，就k-1。那就要看下is2函数

```
public static boolean is2(int n) {
    if (n <= 3) {
        if (n > 1) {
            return true;
        }
        return false;
    } else if (n % 2 == 0 || n % 3 == 0) {
        return false;
    } else {
        int i = 5;
        while (i * i <= n) {
            if (n % i == 0 || n % (i + 2) == 0) {
                return false;
            }
            i += 6;
        }
        return true;
    }
}
```

<https://blog.csdn.net/dadongwudi>

直接照着写一个即可，然后可以算出关键变量k

解密脚本，算出k = 1616384

```
def is2(n):
    if(n <= 3):
        if(n > 1):
            return True
        return False
    elif(n % 2 == 0 or n % 3 == 0):
        return False
    else:
        i = 5
        while(i * i <= n):
            if (n % i == 0 or n % (i + 2) == 0):
                return False
            i += 6
        return True

k=0

for i in range(200000,0,-1):
    k = k + 100 if is2(i) else k - 1
print(k)
```



```
(base) PS D:\CTF\script> python .\ali.py  
1616384
```

### 3. 实现

的话，用Androidkiller打开项目，因为跳转后输出了The flag is,所以搜索该字符串,双击跟过去

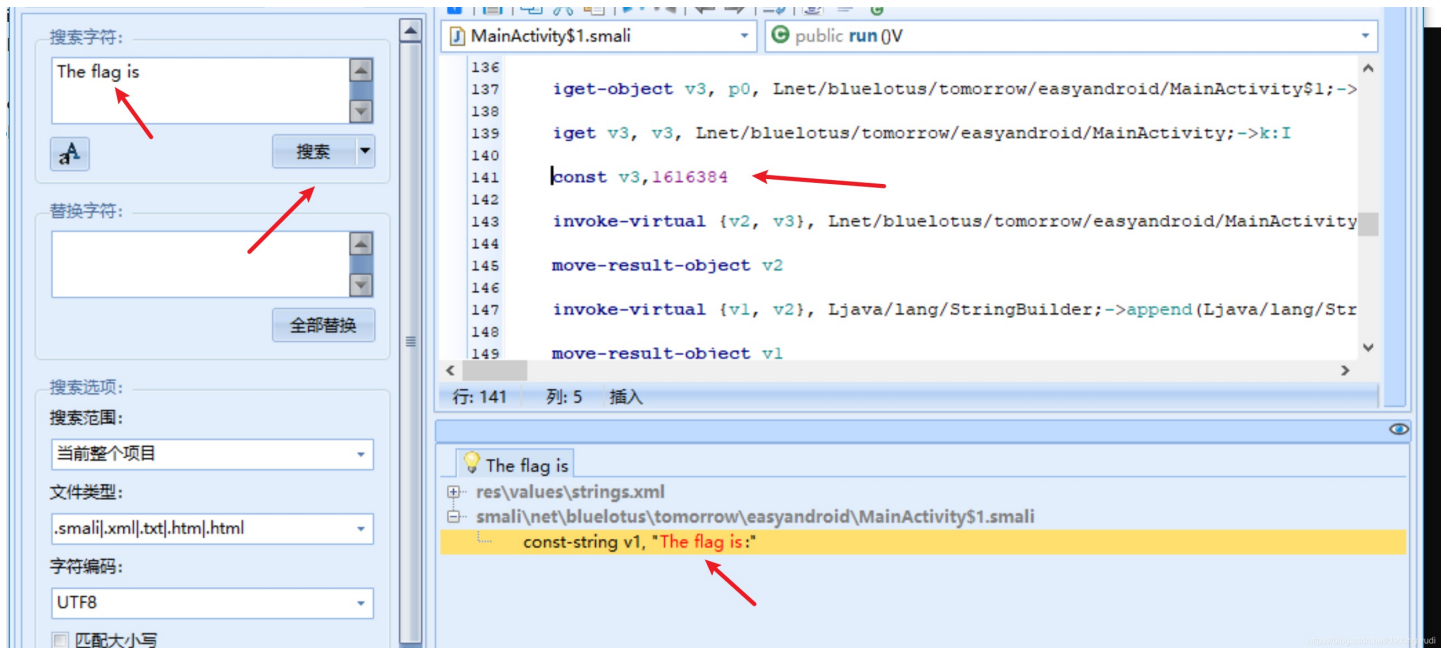
3.1 第113行的if-gtz v0, :cond\_0。if-ltz是如果大于0跳转，那改成如果小于0跳转就跳过了200000秒等待了。对应的语句为if-ltz v0, :cond\_0。

3.2 然后要找到赋值k的位置，看第129行-149行，因为k的值是在alictf(和)之间传入的。

看到了139行的iget v3, v3, Lnet/bluelotus/tomorrow/easyandroid/MainActivity;->k:I，知道v3是k的值。

于是在下面赋值const v3,1616384

然后保存，编译，安装运行就出现flag。（jdk=1.8 apktool>=2.3）



```
112  
113  
114
```

```
if-ltz v0, :cond_0
```

```
140  
141
```

```
const v3,1616384
```

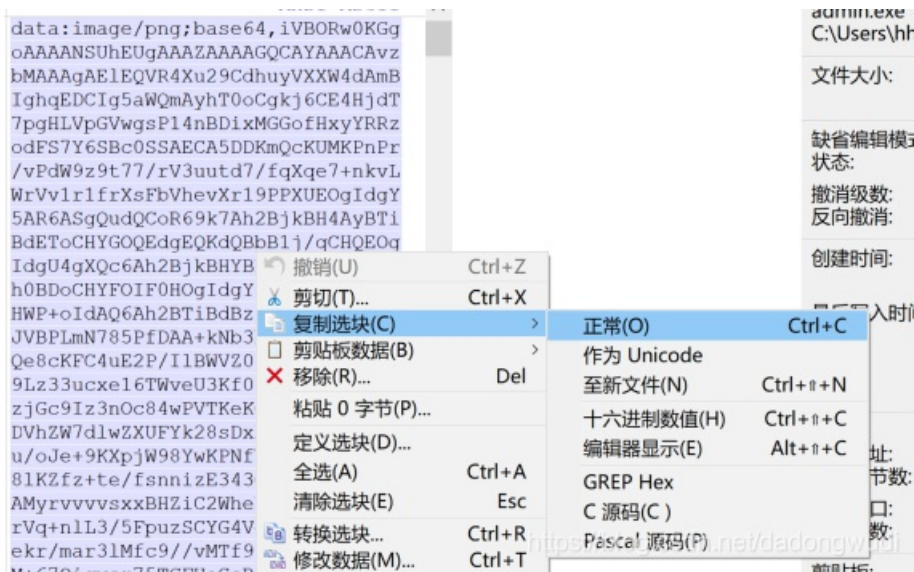


Reverse 逆向入门

关键字: winhex

步骤:

1.winhex 打开 复制黏贴到浏览器



2.QR扫描出结果





自动扩张

尺寸(M)

4

图片像素

148 × 148

编码(C)

DoCoMo

电话簿信息

电子邮件

网页书签

文本信息

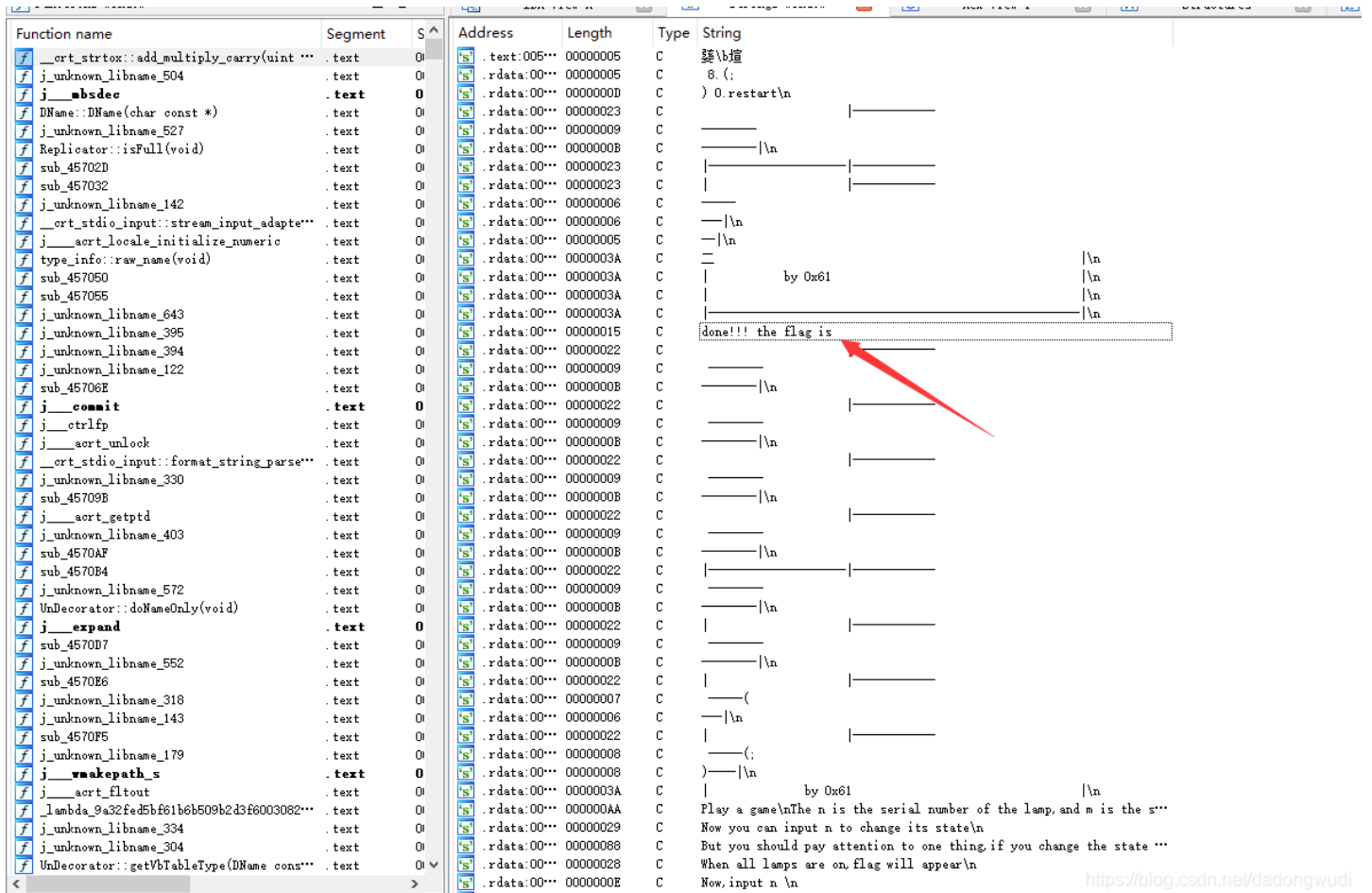
bugku{inde\_9882ihsd8-0}

## Reverse 游戏过关

关键字: ida

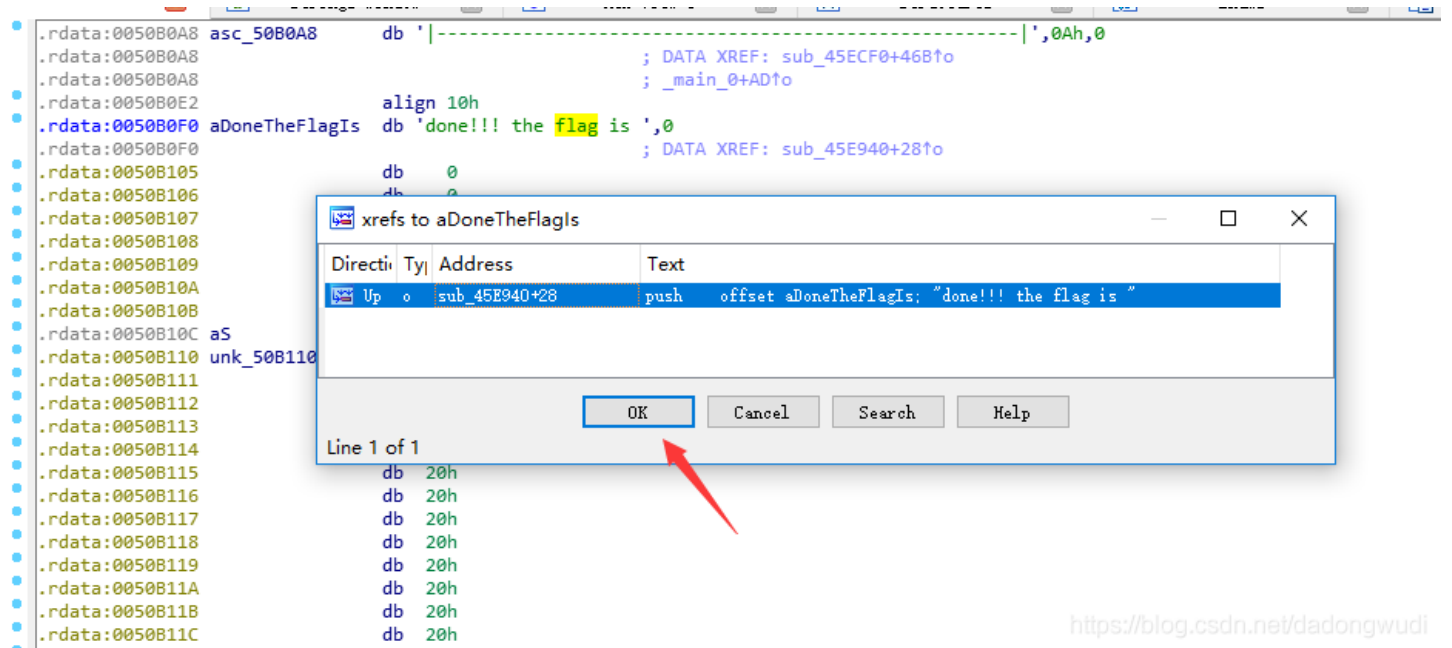
步骤:

1.首先就是看运行遍程序，了解下程序流程以及关键字字符串。然后打开ida  
Shift+F12查看下字符串，然后双击过去



Function name	Segment	S	Address	Length	Type	String
__crt_strtokx::add_multiply_carry(uint ...	.text	0	.text:005...	00000005	C	5\b\b\b
j_unknown_libname_504	.text	0	.rdata:00...	00000005	C	8 (;
j__absdec	.text	0	.rdata:00...	00000000	C	) 0.restart\n
DName::DName(char const *)	.text	0	.rdata:00...	00000023	C	
j_unknown_libname_527	.text	0	.rdata:00...	00000009	C	
Replicator::isFull(void)	.text	0	.rdata:00...	0000000B	C	
sub_45702D	.text	0	.rdata:00...	00000023	C	
sub_457032	.text	0	.rdata:00...	00000023	C	
j_unknown_libname_142	.text	0	.rdata:00...	00000006	C	
__crt_stdio_input::stream_input_adapte...	.text	0	.rdata:00...	00000006	C	
j__aort_locale_initialize_numeric	.text	0	.rdata:00...	00000005	C	
type_info::raw_name(void)	.text	0	.rdata:00...	0000003A	C	
sub_457050	.text	0	.rdata:00...	0000003A	C	by 0x61
sub_457055	.text	0	.rdata:00...	0000003A	C	
j_unknown_libname_643	.text	0	.rdata:00...	0000003A	C	
j_unknown_libname_395	.text	0	.rdata:00...	00000015	C	done!!! the flag is
j_unknown_libname_394	.text	0	.rdata:00...	00000022	C	
j_unknown_libname_122	.text	0	.rdata:00...	00000009	C	
sub_45706E	.text	0	.rdata:00...	0000000B	C	
j__commit	.text	0	.rdata:00...	00000022	C	
j__ctrlfp	.text	0	.rdata:00...	00000009	C	
j__aort_unlock	.text	0	.rdata:00...	0000000B	C	
__crt_stdio_input::format_string_parse...	.text	0	.rdata:00...	00000022	C	
j_unknown_libname_330	.text	0	.rdata:00...	00000009	C	
sub_45709B	.text	0	.rdata:00...	0000000B	C	
j__aort_getpttd	.text	0	.rdata:00...	00000022	C	
j_unknown_libname_403	.text	0	.rdata:00...	00000009	C	
sub_4570AF	.text	0	.rdata:00...	0000000B	C	
sub_4570B4	.text	0	.rdata:00...	00000022	C	
j_unknown_libname_572	.text	0	.rdata:00...	00000009	C	
Undecorator::doNameOnly(void)	.text	0	.rdata:00...	0000000B	C	
j__expand	.text	0	.rdata:00...	00000022	C	
sub_4570D7	.text	0	.rdata:00...	00000009	C	
j_unknown_libname_552	.text	0	.rdata:00...	0000000B	C	
sub_4570E6	.text	0	.rdata:00...	00000022	C	
j_unknown_libname_318	.text	0	.rdata:00...	00000007	C	
j_unknown_libname_143	.text	0	.rdata:00...	00000006	C	
sub_4570F5	.text	0	.rdata:00...	00000022	C	
j_unknown_libname_179	.text	0	.rdata:00...	00000008	C	
j__wakepath_s	.text	0	.rdata:00...	00000008	C	
j__aort_fltout	.text	0	.rdata:00...	0000003A	C	by 0x61
lambda_9a32fed5b6f61b6b509b2d3f8003082...	.text	0	.rdata:00...	000000AA	C	Play a game\nThe n is the serial number of the lamp, and m is the s...
j_unknown_libname_334	.text	0	.rdata:00...	00000029	C	Now you can input n to change its state\n
j_unknown_libname_304	.text	0	.rdata:00...	00000088	C	But you should pay attention to one thing if you change the state ...
Undecorator::getVtblType(DName cons...	.text	0	.rdata:00...	00000028	C	When all lamps are on, flag will appear\n
			.rdata:00...	0000000E	C	Now, input n \n

2.按Cirt+X交叉引用显示调用位置



Directi	Ty	Address	Text
Up	o	sub_45E940+28	push offset aDoneTheFlagIs; "done!!! the flag is "

Line 1 of 1

3.F5看下伪代码，两个数组按位异或再和0x13异或生成flag

```
array1 = [18, 64, 98, 5, 2, 4, 6, 3, 6, 48, 49, 65, 32, 12, 48, 65, 31, 78, 62, 32, 49, 32, 1, 57, 96, 3, 21, 9, 4, 62, 3, 5, 4, 1, 2, 3, 44, 65, 78, 3, 2, 16, 97, 54, 16, 44, 52, 32, 64, 89, 45, 32, 65, 15, 34, 18, 16, 0]
array2 = [123, 32, 18, 98, 119, 108, 65, 41, 124, 80, 125, 38, 124, 111, 74, 49, 83, 108, 94, 108, 84, 6, 96, 83, 44, 121, 104, 110, 32, 95, 1, 17, 101, 99, 123, 127, 119, 96, 48, 107, 71, 92, 29, 81, 107, 90, 85, 64, 12, 43, 76, 86, 13, 114, 1, 117, 126, 0]

flag = ''
for i in range(len(array1)):
    flag+= chr(array1[i] ^ array2[i] ^ 0x13 )
print(flag)
```

Reverse

关键字:

知识点:

步骤:

Reverse

关键字:

知识点:

步骤:

Reverse

关键字:

知识点:

步骤:

Reverse

关键字:

知识点:

步骤:

Reverse

关键字:

知识点:

步骤:

Reverse

关键字:

知识点:

步骤:

参考链接:

<https://www.codeqq.com/log/7Zjb2O7Z.html>

<https://blog.csdn.net/ahilll/article/details/84787700>