re学习笔记（93）攻防世界 - mobile进阶区 - Illusion

原创

Forgo7ten 于 2021-11-08 15:34:52 发布 96 收藏

分类专栏： Android逆向 # reverse ctf小白成长ing 文章标签： CTF 攻防世界 Android逆向 Reverse

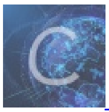
本文链接：https://blog.csdn.net/Palmer9/article/details/121208591

版权

Android逆向 同时被 3 个专栏收录

31 篇文章 5 订阅

订阅专栏



reverse

113 篇文章 6 订阅

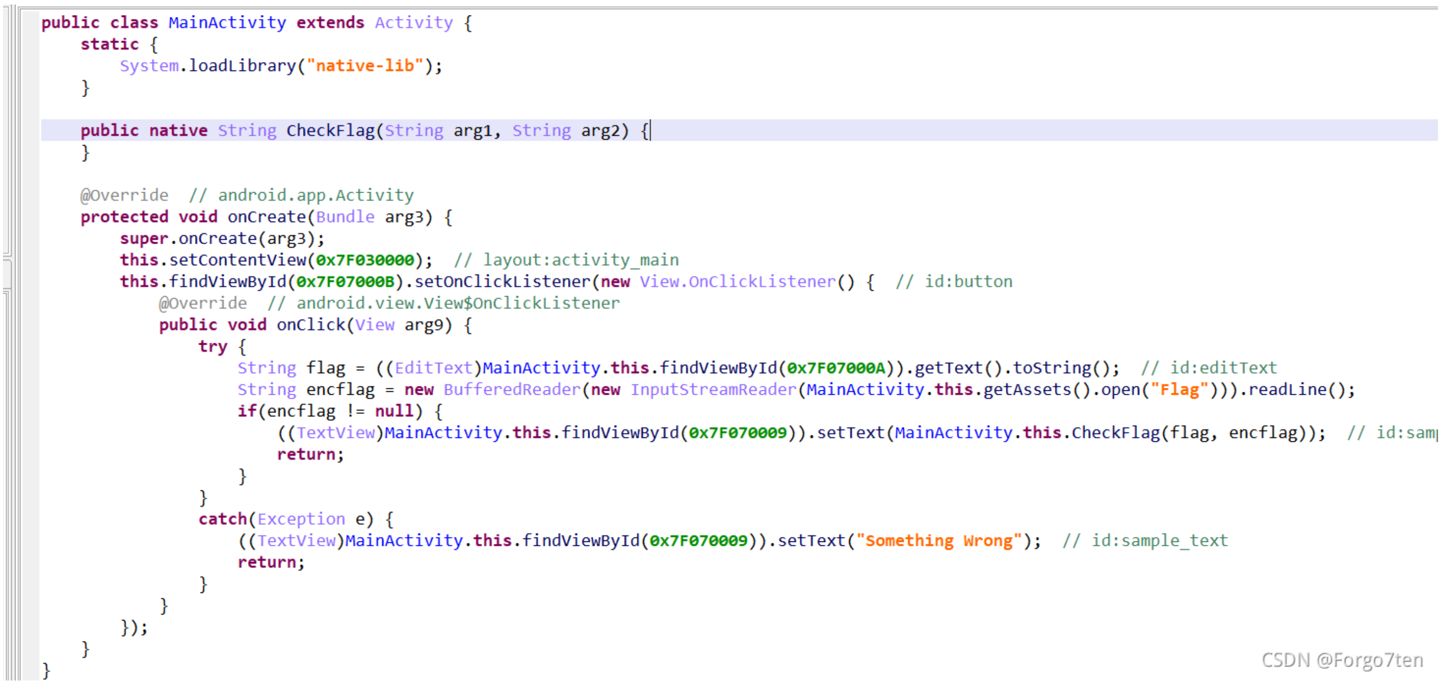
订阅专栏

ctf小白成长ing

112 篇文章 6 订阅

订阅专栏

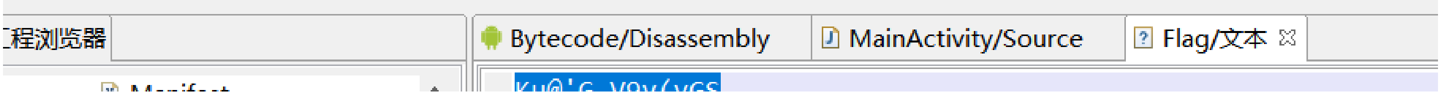
jeb载入查看MainActivity

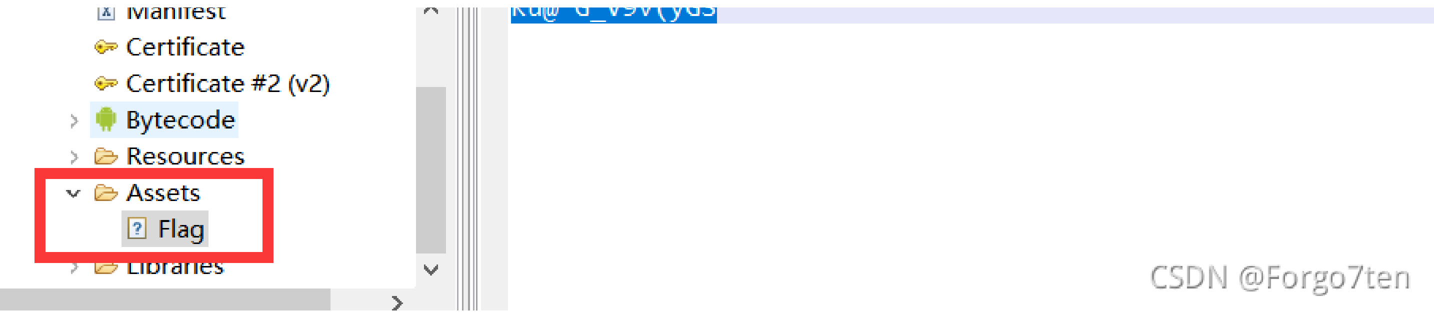


```
public class MainActivity extends Activity {
    static {
        System.loadLibrary("native-lib");
    }

    public native String CheckFlag(String arg1, String arg2) {
    }

    @Override  // android.app.Activity
    protected void onCreate(Bundle arg3) {
        super.onCreate(arg3);
        this.setContentView(0x7F030000);  // layout:activity_main
        this.findViewById(0x7F07000B).setOnClickListener(new View.OnClickListener() {  // id:button
            @Override  // android.view.View$OnClickListener
            public void onClick(View arg9) {
                try {
                    String flag = ((EditText)MainActivity.this.findViewById(0x7F07000A)).getText().toString();  // id:editText
                    String encflag = new BufferedReader(new InputStreamReader(MainActivity.this.getAssets().open("Flag"))).readLine();
                    if(encflag != null) {
                        ((TextView)MainActivity.this.findViewById(0x7F070009)).setText(MainActivity.this.CheckFlag(flag, encflag));  // id:samp
                        return;
                    }
                }
                catch(Exception e) {
                    ((TextView)MainActivity.this.findViewById(0x7F070009)).setText("Something Wrong");  // id:sample_text
                    return;
                }
            }
        });
    }
}
```

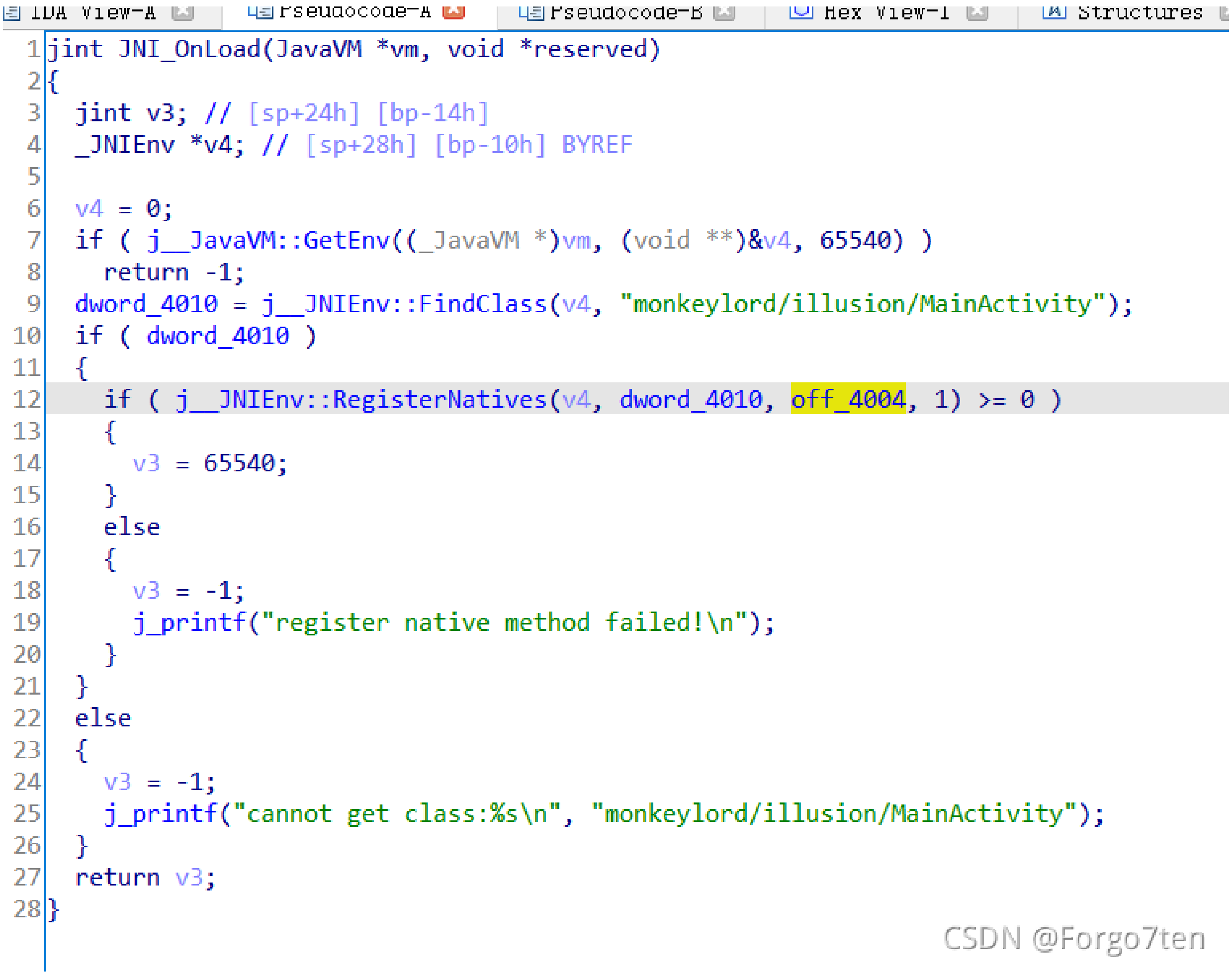
可以发现是将用户输入，与encflag传入了native方法中去，native方法的返回值就是显示结果

而encflag是从assets文件夹下的flag文件中获取的

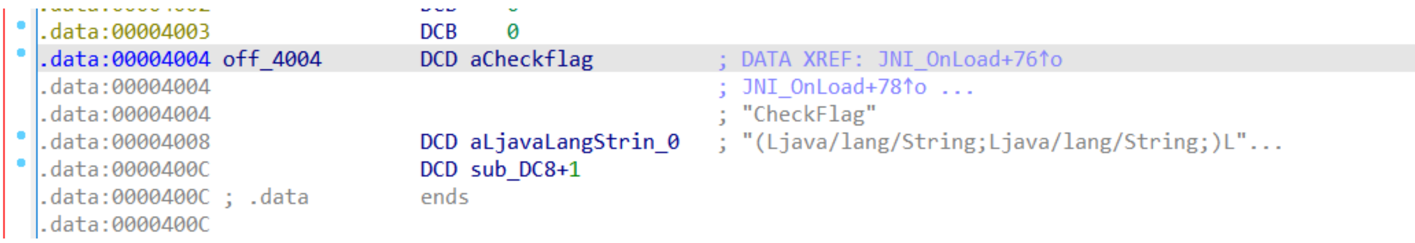




之后IDA查看so文件
JNI_OnLoad动态注册了函数

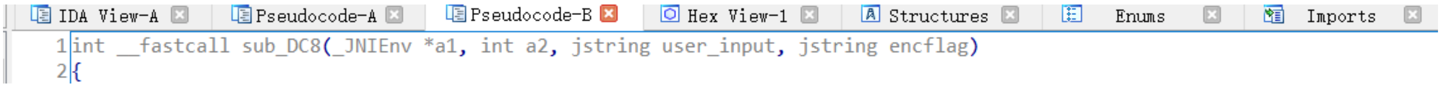


```
jint JNI_OnLoad(JavaVM *vm, void *reserved)
{
  jint v3; // [sp+24h] [bp-14h]
  _JNIEnv *v4; // [sp+28h] [bp-10h] BYREF

  v4 = 0;
  if ( j__JavaVM::GetEnv((_JavaVM *)vm, (void **)&v4, 65540) )
    return -1;
  dword_4010 = j__JNIEnv::FindClass(v4, "monkeylord/illusion/MainActivity");
  if ( dword_4010 )
  {
    if ( j__JNIEnv::RegisterNatives(v4, dword_4010, off_4004, 1) >= 0 )
    {
      v3 = 65540;
    }
    else
    {
      v3 = -1;
      j_printf("register native method failed!\n");
    }
  }
  else
  {
    v3 = -1;
    j_printf("cannot get class:%s\n", "monkeylord/illusion/MainActivity");
  }
  return v3;
}
```

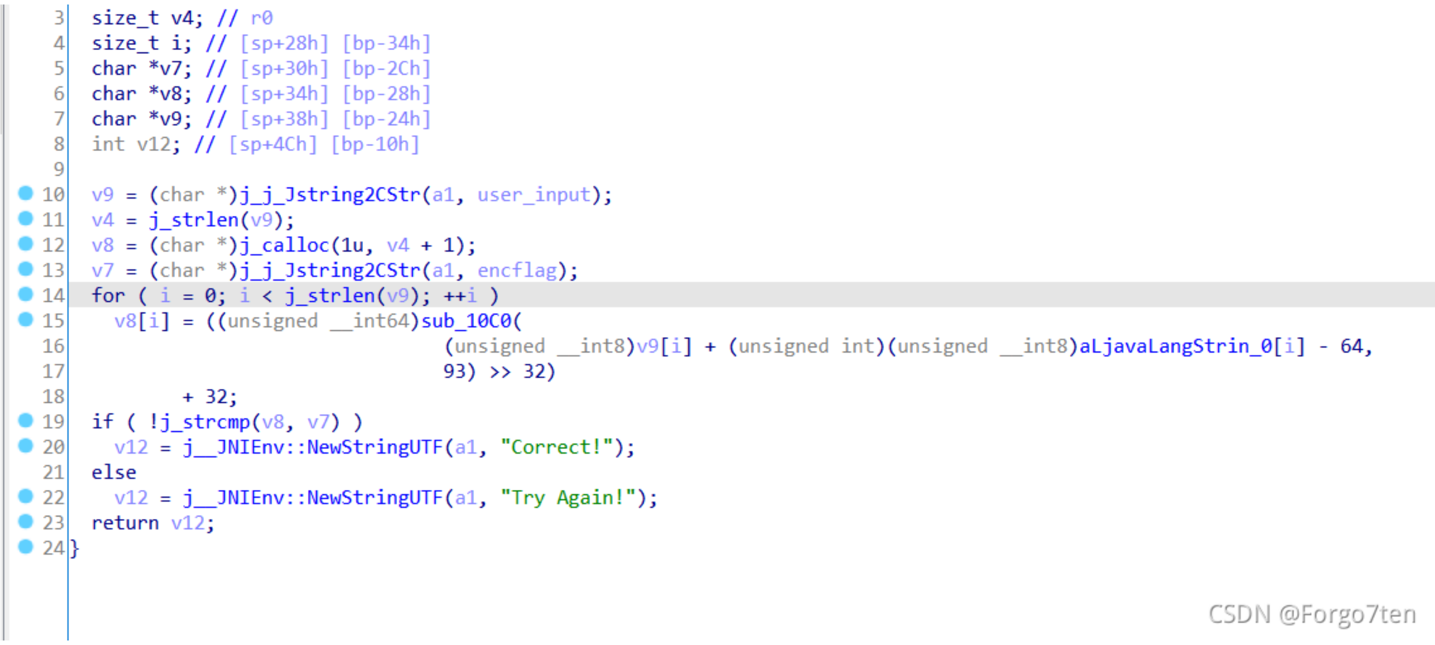


```
.data:00004003                 DCB    0
.data:00004004 off_4004        DCD aCheckflag            ; DATA XREF: JNI_OnLoad+76↑o
.data:00004004                                           ; JNI_OnLoad+78↑o ...
.data:00004004                                           ; "CheckFlag"
.data:00004008                 DCD aLjavaLangStrin_0     ; "(Ljava/lang/String;Ljava/lang/String;)L"...
.data:0000400C                 DCD sub_DC8+1
.data:0000400C ; .data         ends
.data:0000400C
```

查看逻辑
将input与字符串相加，再减去64，和93一并传入sub_10C0函数
之后将返回值加上32得到字符串结果。
将得到的字符串与参数二也就是encflag进行比较，返回比较结果

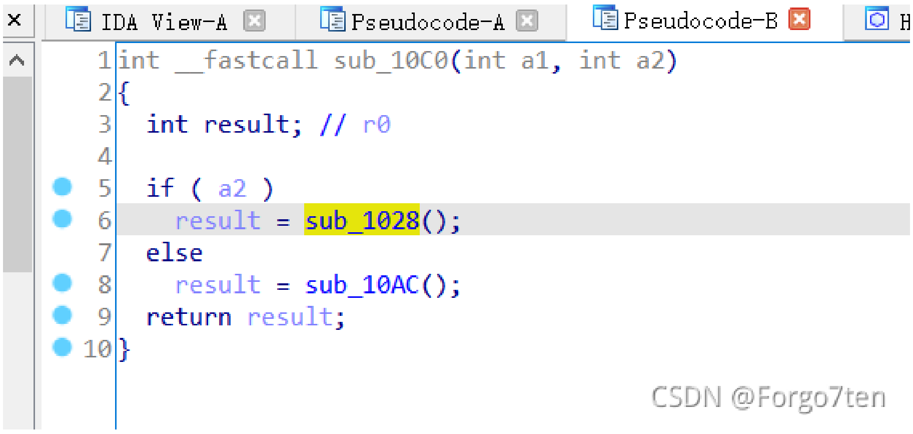


```
int __fastcall sub_DC8(_JNIEnv *a1, int a2, jstring user_input, jstring encflag)
{
```



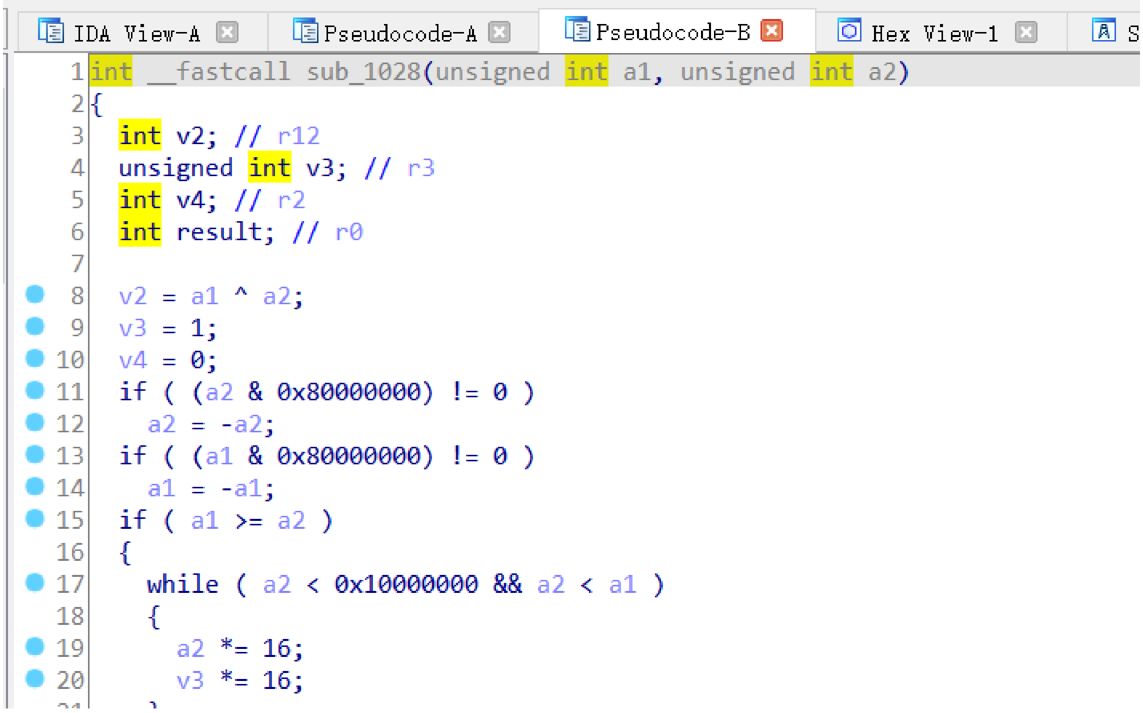
```
  size_t v4; // r0
  size_t i; // [sp+28h] [bp-34h]
  char *v7; // [sp+30h] [bp-2Ch]
  char *v8; // [sp+34h] [bp-28h]
  char *v9; // [sp+38h] [bp-24h]
  int v12; // [sp+4Ch] [bp-10h]

  v9 = (char *)j_j_Jstring2CStr(a1, user_input);
  v4 = j_strlen(v9);
  v8 = (char *)j_calloc(1u, v4 + 1);
  v7 = (char *)j_j_Jstring2CStr(a1, encflag);
  for ( i = 0; i < j_strlen(v9); ++i )
    v8[i] = ((unsigned __int64)sub_10C0(
                                 (unsigned __int8)v9[i] + (unsigned int)(unsigned __int8)aLjavaLangStrin_0[i] - 64,
                                 93) >> 32)
          + 32;
  if ( !j_strcmp(v8, v7) )
    v12 = j__JNIEnv::NewStringUTF(a1, "Correct!");
  else
    v12 = j__JNIEnv::NewStringUTF(a1, "Try Again!");
  return v12;
}
```

而sub_10C0函数，则是判断a2是否等于0，然后来判断执行哪个函数
由于a2的值恒定为93，所以只执行sub_1028()函数

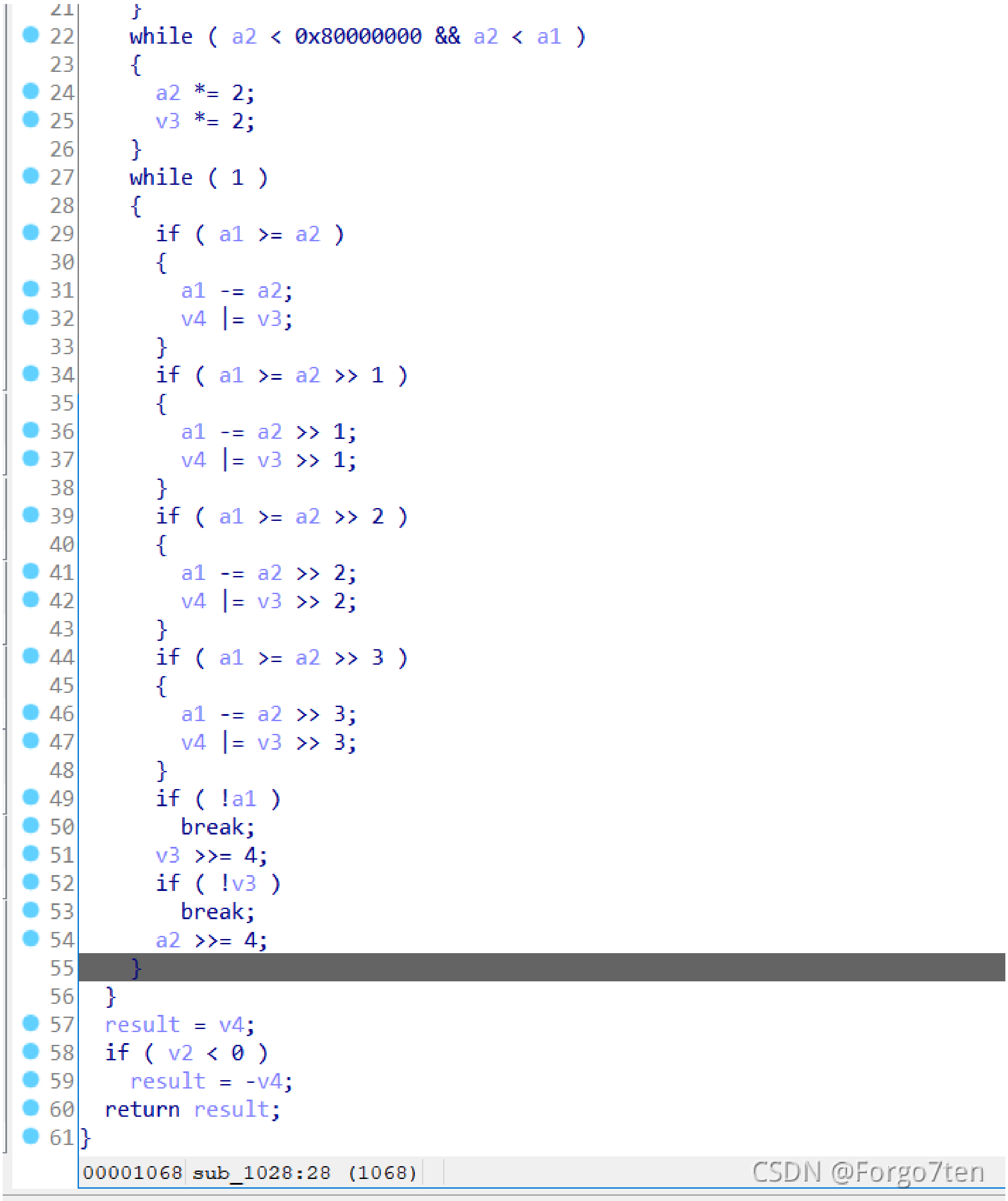


```
int __fastcall sub_10C0(int a1, int a2)
{
  int result; // r0

  if ( a2 )
    result = sub_1028();
  else
    result = sub_10AC();
  return result;
}
```

sub1028虽然有些长，但是逻辑很简单，返回的结果就是a1除以a2的商



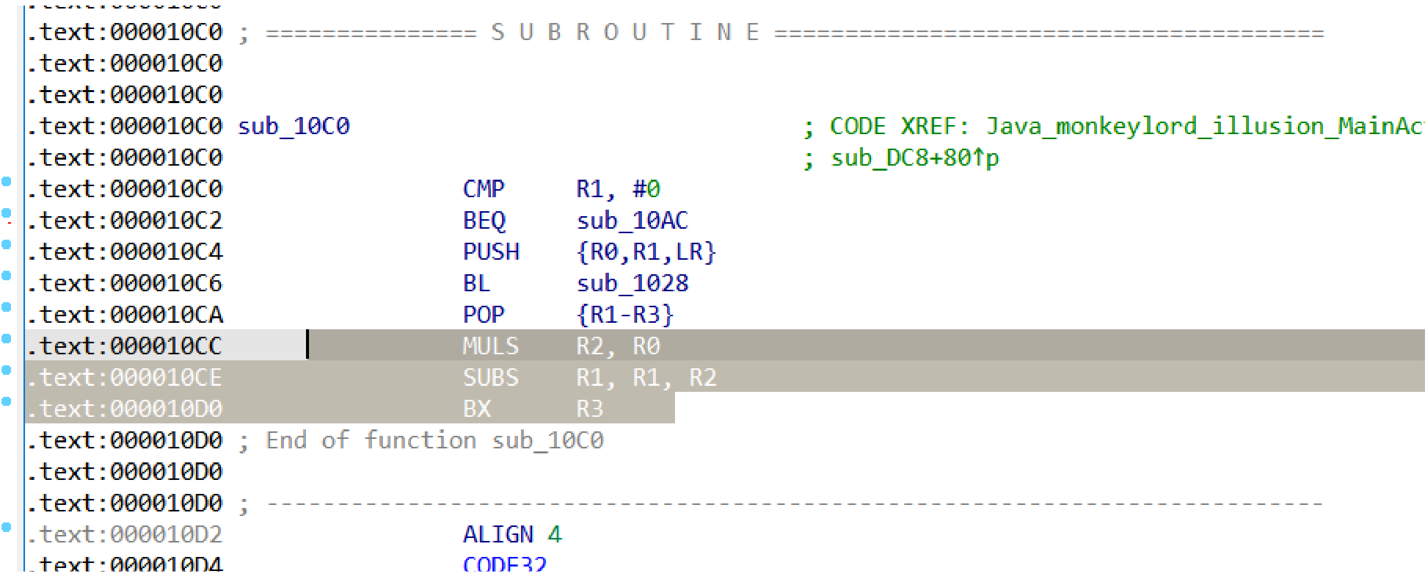
```
int __fastcall sub_1028(unsigned int a1, unsigned int a2)
{
  int v2; // r12
  unsigned int v3; // r3
  int v4; // r2
  int result; // r0

  v2 = a1 ^ a2;
  v3 = 1;
  v4 = 0;
  if ( (a2 & 0x80000000) != 0 )
    a2 = -a2;
  if ( (a1 & 0x80000000) != 0 )
    a1 = -a1;
  if ( a1 >= a2 )
  {
    while ( a2 < 0x10000000 && a2 < a1 )
    {
      a2 *= 16;
      v3 *= 16;
    }
```



```
    }
    while ( a2 < 0x80000000 && a2 < a1 )
    {
      a2 *= 2;
      v3 *= 2;
    }
    while ( 1 )
    {
      if ( a1 >= a2 )
      {
        a1 -= a2;
        v4 |= v3;
      }
      if ( a1 >= a2 >> 1 )
      {
        a1 -= a2 >> 1;
        v4 |= v3 >> 1;
      }
      if ( a1 >= a2 >> 2 )
      {
        a1 -= a2 >> 2;
        v4 |= v3 >> 2;
      }
      if ( a1 >= a2 >> 3 )
      {
        a1 -= a2 >> 3;
        v4 |= v3 >> 3;
      }
      if ( !a1 )
        break;
      v3 >>= 4;
      if ( !v3 )
        break;
      a2 >>= 4;
    }
  }
  result = v4;
  if ( v2 < 0 )
    result = -v4;
  return result;
}
```

00001068 sub_1028:28 (1068)

动调后发现结果不一致，找到sub_10C0函数的最后，将商R0再乘以R2=93，然后让R1减去相乘后的结果
也就是说10C0函数实际上返回的值是除以93的余数



```
.text:000010C0 ; =============== S U B R O U T I N E =======================================
.text:000010C0
.text:000010C0
.text:000010C0 sub_10C0                                ; CODE XREF: Java_monkeylord_illusion_MainAc
.text:000010C0                                         ; sub_DC8+80↑p
.text:000010C0                 CMP     R1, #0
.text:000010C2                 BEQ     sub_10AC
.text:000010C4                 PUSH    {R0,R1,LR}
.text:000010C6                 BL      sub_1028
.text:000010CA                 POP     {R1-R3}
.text:000010CC                 MULS    R2, R0
.text:000010CE                 SUBS    R1, R1, R2
.text:000010D0                 BX      R3
.text:000010D0 ; End of function sub_10C0
.text:000010D0
.text:000010D0 ; ---------------------------------------------------------------------------
.text:000010D2                 ALIGN 4
.text:000010D4                 CODE32
```

```
.text:000010D4
```

所以程序的流程就是
将用户输入的每个字符，加上内置data字符串对应的下标的字符，然后再减去64；
得到的结果，取余93后，进行比较

exp为

```python
enc_flag = "Ku@'G_V9v(yGS"
data = "(Ljava/lang/String;Ljava/lang/String;)Ljava/lang/String;"
data = [ord(i) for i in data]


def main():
    enc = [ord(i) for i in enc_flag]
    flag = [0]*len(enc)
    for i in range(len(enc)):
        flag[i] = (enc[i]-32)+64-data[i]
        while flag[i] < 0x32:
            flag[i] += 93
        while flag[i] > 125:
            flag[i] -= 93
    print("".join([chr(i) for i in flag]))


if __name__ == '__main__':
    main()
# CISCN{GJ5728}
```

得到flag为 `CISCN{GJ5728}`