XTU.apk.apk逆向writeup

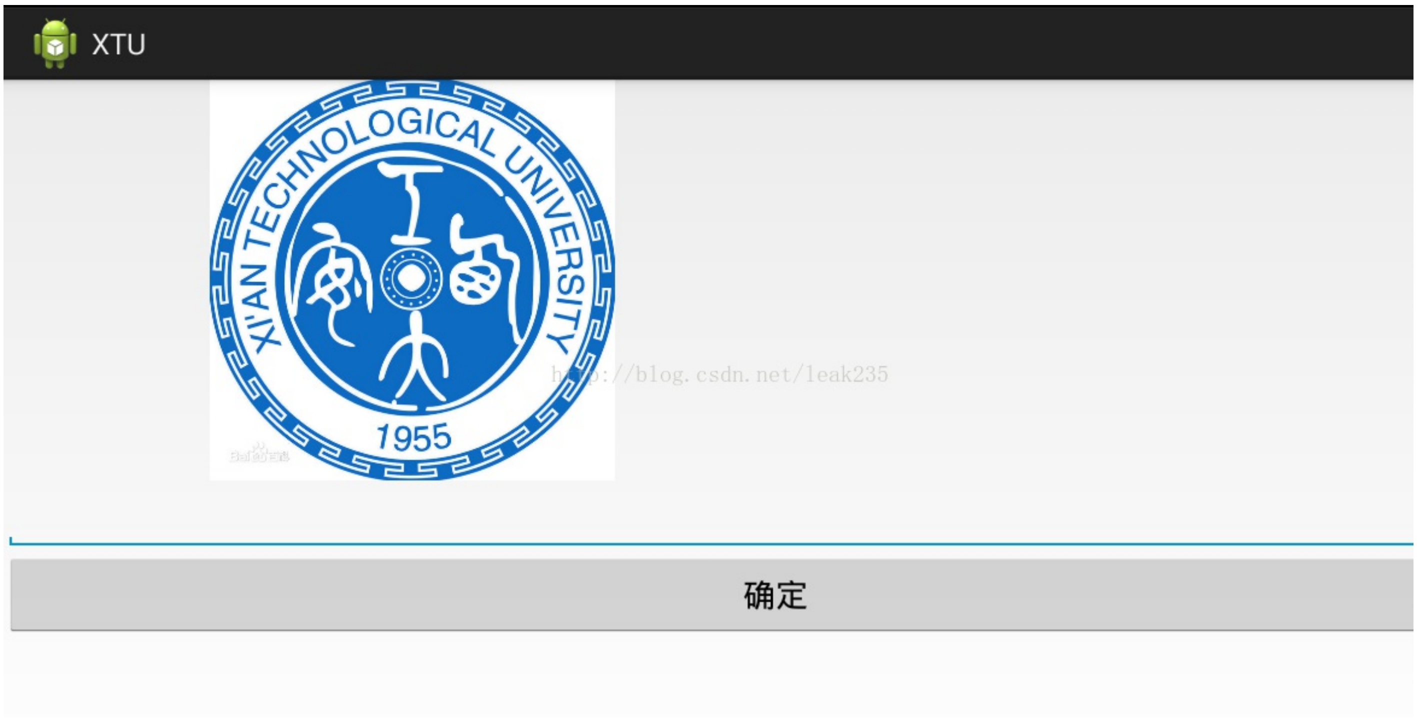
原创

leak235 于 2017-11-26 22:50:02 发布 449 收藏
分类专栏： 逆向 文章标签： 逆向 CTF Android

逆向 专栏收录该内容
1 篇文章 0 订阅
订阅专栏

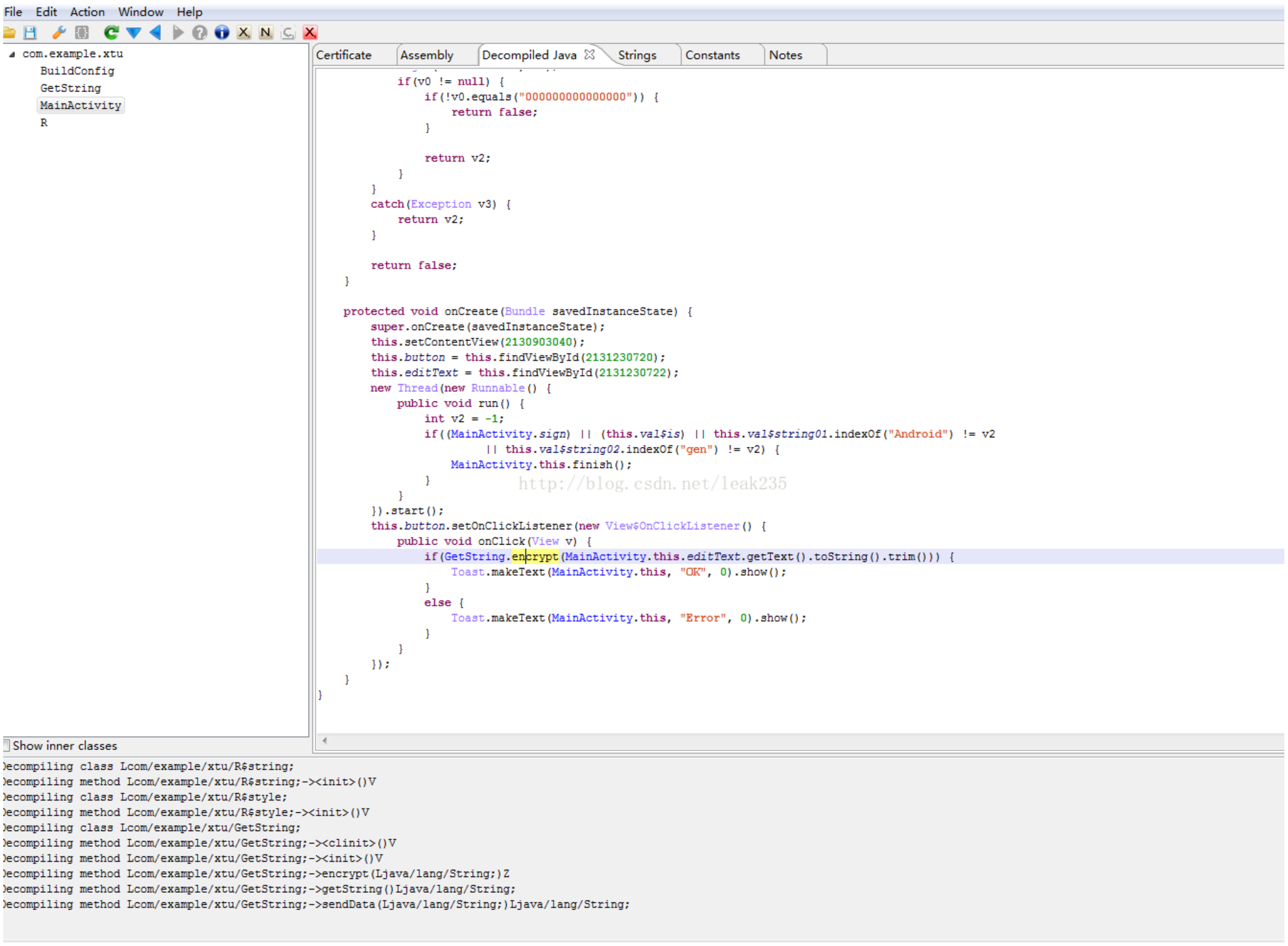
最近逆向了西安工业大学出的CTF题，XTU.apk.apk

打开后如图



首先放到jeb里，很强大，直接按Tab就可以反编译成Java代码

查找onclick事件函数的内容



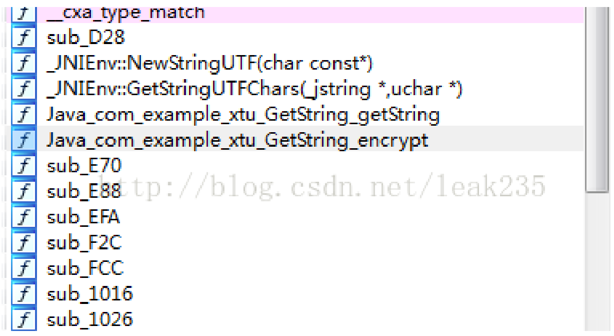
```java
            if(v0 != null) {
                if(!v0.equals("000000000000000")) {
                    return false;
                }

                return v2;
            }
        }
        catch(Exception v3) {
            return v2;
        }

        return false;
    }

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        this.setContentView(2130903040);
        this.button = this.findViewById(2131230720);
        this.editText = this.findViewById(2131230722);
        new Thread(new Runnable() {
            public void run() {
                int v2 = -1;
                if((MainActivity.sign) || (this.val$is) || this.val$string01.indexOf("Android") != v2
                         || this.val$string02.indexOf("gen") != v2) {
                    MainActivity.this.finish();
                }
            }
        }).start();
        this.button.setOnClickListener(new View$OnClickListener() {
            public void onClick(View v) {
                if(GetString.encrypt(MainActivity.this.editText.getText().toString().trim())) {
                    Toast.makeText(MainActivity.this, "OK", 0).show();
                }
                else {
                    Toast.makeText(MainActivity.this, "Error", 0).show();
                }
            }
        });
    }
}
```

```
)ecompiling class Lcom/example/xtu/R$string;
)ecompiling method Lcom/example/xtu/R$string;-><init>()V
)ecompiling class Lcom/example/xtu/R$style;
)ecompiling method Lcom/example/xtu/R$style;-><init>()V
)ecompiling class Lcom/example/xtu/GetString;
)ecompiling method Lcom/example/xtu/GetString;-><clinit>()V
)ecompiling method Lcom/example/xtu/GetString;-><init>()V
)ecompiling method Lcom/example/xtu/GetString;->encrypt(Ljava/lang/String;)Z
)ecompiling method Lcom/example/xtu/GetString;->getString()Ljava/lang/String;
)ecompiling method Lcom/example/xtu/GetString;->sendData(Ljava/lang/String;)Ljava/lang/String;
```

但里面用到了内置函数encrypt

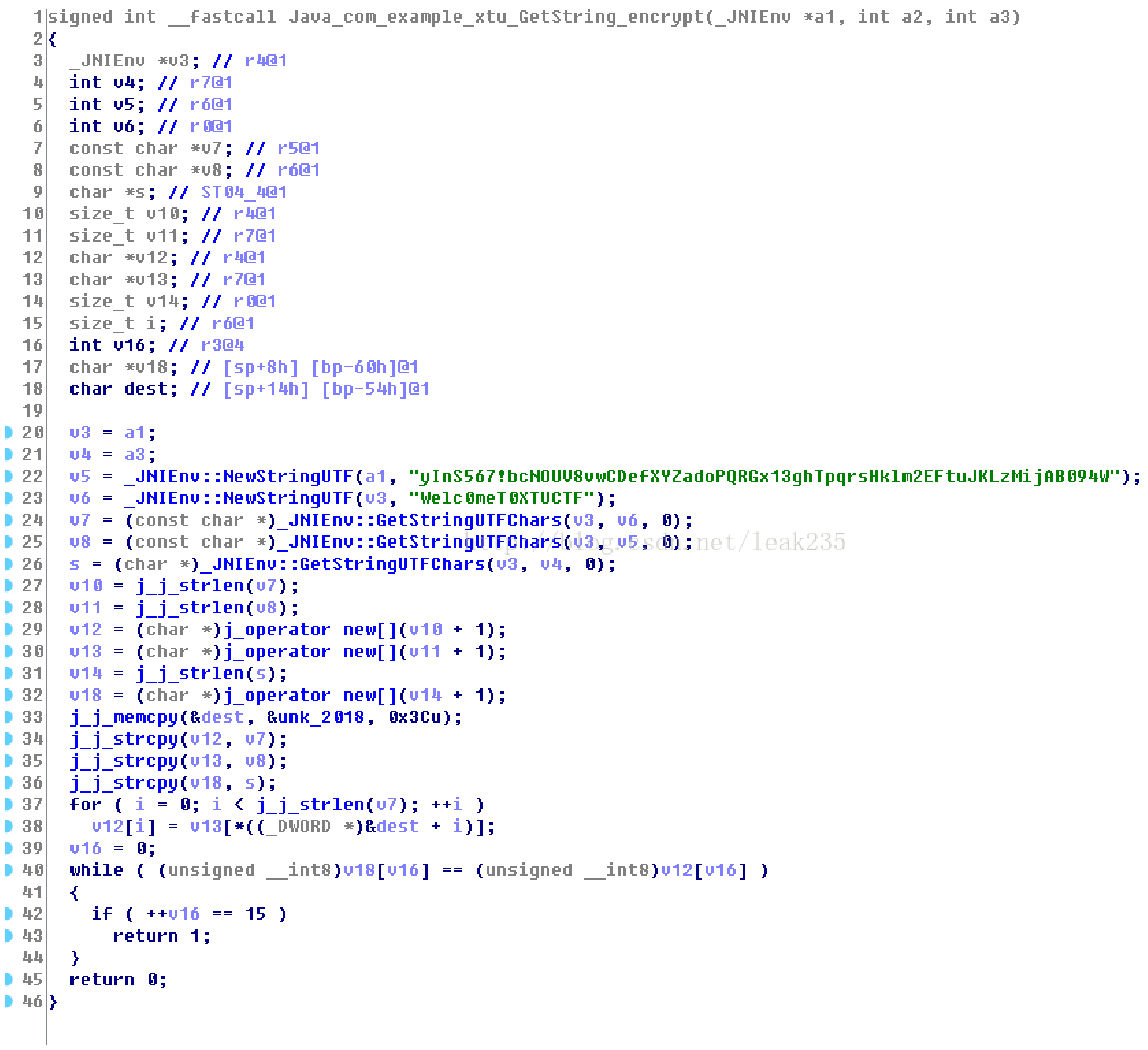
用AndroidKiller解包后，果断使用ida，打开工程里面的lib\armeabi\libXTU.so文件

找到java内置函数



```
__cxa_type_match
sub_D28
_JNIEnv::NewStringUTF(char const*)
_JNIEnv::GetStringUTFChars(_jstring *,uchar *)
Java_com_example_xtu_GetString_getString
Java_com_example_xtu_GetString_encrypt
sub_E70
sub_E88
sub_EFA
sub_F2C
sub_FCC
sub_1016
sub_1026
```

一看汇编有点懵，幸亏ida可以转化为c代码（F5键）

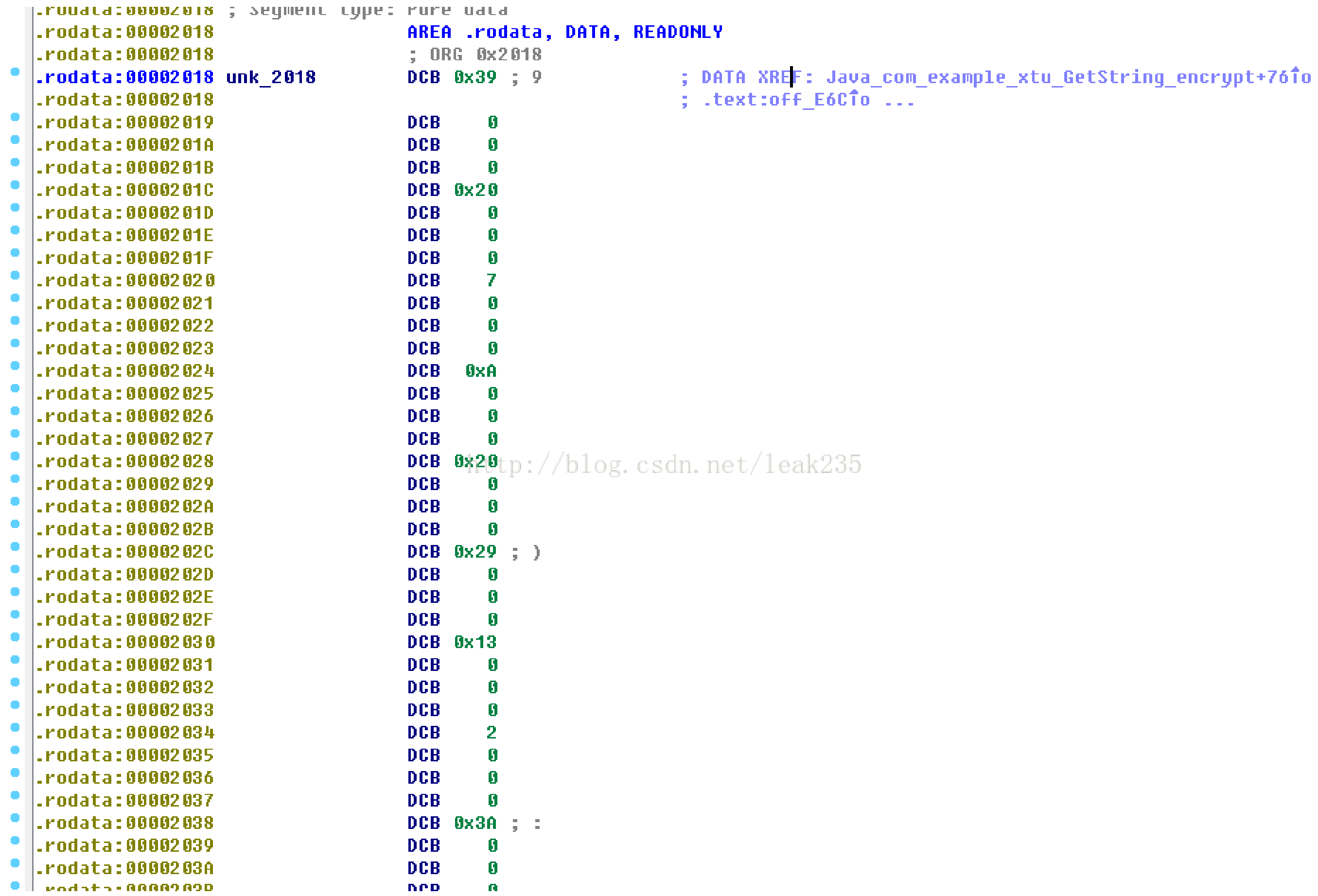


```c
signed int __fastcall Java_com_example_xtu_GetString_encrypt(_JNIEnv *a1, int a2, int a3)
{
  _JNIEnv *v3; // r4@1
  int v4; // r7@1
  int v5; // r6@1
  int v6; // r0@1
  const char *v7; // r5@1
  const char *v8; // r6@1
  char *s; // ST04_4@1
  size_t v10; // r4@1
  size_t v11; // r7@1
  char *v12; // r4@1
  char *v13; // r7@1
  size_t v14; // r0@1
  size_t i; // r6@1
  int v16; // r3@4
  char *v18; // [sp+8h] [bp-60h]@1
  char dest; // [sp+14h] [bp-54h]@1

  v3 = a1;
  v4 = a3;
  v5 = _JNIEnv::NewStringUTF(a1, "yInS567!bcNOUV8vwCDefXYZadoPQRGx13ghTpqrsHklm2EFtuJKLzMijAB094W");
  v6 = _JNIEnv::NewStringUTF(v3, "Welc0meT0XTUCTF");
  v7 = (const char *)_JNIEnv::GetStringUTFChars(v3, v6, 0);
  v8 = (const char *)_JNIEnv::GetStringUTFChars(v3, v5, 0);
  s = (char *)_JNIEnv::GetStringUTFChars(v3, v4, 0);
  v10 = j_j_strlen(v7);
  v11 = j_j_strlen(v8);
  v12 = (char *)j_operator new[](v10 + 1);
  v13 = (char *)j_operator new[](v11 + 1);
  v14 = j_j_strlen(s);
  v18 = (char *)j_operator new[](v14 + 1);
  j_j_memcpy(&dest, &unk_2018, 0x3Cu);
  j_j_strcpy(v12, v7);
  j_j_strcpy(v13, v8);
  j_j_strcpy(v18, s);
  for ( i = 0; i < j_j_strlen(v7); ++i )
    v12[i] = v13[*((_DWORD *)&dest + i)];
  v16 = 0;
  while ( (unsigned __int8)v18[v16] == (unsigned __int8)v12[v16] )
  {
    if ( ++v16 == 15 )
      return 1;
  }
  return 0;
}
```

代码有点不太明晰，拷贝到notepad里优化一下

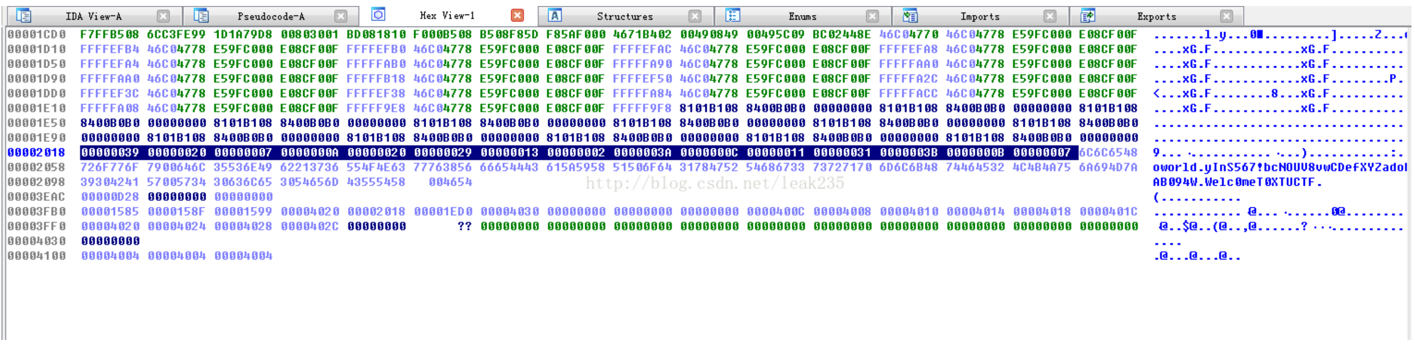
```
signed int __fastcall Java_com_example_xtu_GetString_encrypt(_JNIEnv *a1, int a2, int a3)
{
  _JNIEnv *v3; // r4@1
  int v4; // r7@1
  int v5; // r6@1
  int v6; // r0@1
  const char *v7_Welc; // r5@1
  const char *v8_yInS; // r6@1
  char *s; // ST04_4@1
  size_t v10_len_Welc; // r4@1
  size_t v11_len_yInS; // r7@1
  char *v12_Welc; // r4@1
  char *v13_yInS; // r7@1
  size_t len_s; // r0@1
  size_t i; // r6@1
  int v16; // r3@4
  char *v18_s; // [sp+8h] [bp-60h]@1
  char dest; // [sp+14h] [bp-54h]@1

  v3 = a1;
  v4 = a3;
  v5 = _JNIEnv::NewStringUTF(a1, "yInS567!bcNOUv8_yInSvwCDefXYZadoPQRGx13ghTpqrsHklm2EFtuJKLzMijAB094W");
  v6 = _JNIEnv::NewStringUTF(v3, "Welc0meT0XTUCTF");
  v7_Welc = _JNIEnv::GetStringUTFChars(v3, v6, 0);
  v8_yInS = _JNIEnv::GetStringUTFChars(v3, v5, 0);
  s = _JNIEnv::GetStringUTFChars(v3, v4, 0);
  v10_len_Welc = j_j_strlen(v7_Welc);
  v11_len_yInS = j_j_strlen(v8_yInS);
  v12_Welc = j_operator new[](v10_len_Welc + 1);
  v13_yInS = j_operator new[](v11_len_yInS + 1);
  len_s = j_j_strlen(s);
  v18_s = j_operator new[](len_s + 1);
  j_j_memcpy(&dest, &unk_2018, 0x3Cu);
  j_j_strcpy(v12_Welc, v7_Welc);
  j_j_strcpy(v13_yInS, v8_yInS);
  j_j_strcpy(v18_s, s);
  for ( i = 0; i < j_j_strlen(v7_Welc); ++i )//长度15
    v12_Welc[i] = v13_yInS[*(&dest + i)];
  v16 = 0;
  while ( v18_s[v16] == v12_Welc[v16] )
  {
    if ( ++v16 == 15 )
      return 1;
  }
  return 0;
}
```

第38行，分析后发现就是个查表替换工作，查的是&unk_2018这个地址，双击后发现在.rodata段，参照以前的汇编知识应该是数据段的意思，也就是c里面最前面定义的全局变量



```
.rodata:00002018 ; Segment type: Pure data
.rodata:00002018                 AREA .rodata, DATA, READONLY
.rodata:00002018                 ; ORG 0x2018
.rodata:00002018 unk_2018        DCB 0x39 ; 9            ; DATA XREF: Java_com_example_xtu_GetString_encrypt+76↑o
.rodata:00002018                                         ; .text:off_E6C↑o ...
.rodata:00002019                 DCB    0
.rodata:0000201A                 DCB    0
.rodata:0000201B                 DCB    0
.rodata:0000201C                 DCB 0x20
.rodata:0000201D                 DCB    0
.rodata:0000201E                 DCB    0
.rodata:0000201F                 DCB    0
.rodata:00002020                 DCB    7
.rodata:00002021                 DCB    0
.rodata:00002022                 DCB    0
.rodata:00002023                 DCB    0
.rodata:00002024                 DCB  0xA
.rodata:00002025                 DCB    0
.rodata:00002026                 DCB    0
.rodata:00002027                 DCB    0
.rodata:00002028                 DCB 0x20
.rodata:00002029                 DCB    0
.rodata:0000202A                 DCB    0
.rodata:0000202B                 DCB    0
.rodata:0000202C                 DCB 0x29 ; )
.rodata:0000202D                 DCB    0
.rodata:0000202E                 DCB    0
.rodata:0000202F                 DCB    0
.rodata:00002030                 DCB 0x13
.rodata:00002031                 DCB    0
.rodata:00002032                 DCB    0
.rodata:00002033                 DCB    0
.rodata:00002034                 DCB    2
.rodata:00002035                 DCB    0
.rodata:00002036                 DCB    0
.rodata:00002037                 DCB    0
.rodata:00002038                 DCB 0x3A ; :
.rodata:00002039                 DCB    0
.rodata:0000203A                 DCB    0
.rodata:0000203B                 DCB    0
```

打开十六进制视图，由于指针是DWORD，所以显示使用4-byte



```
00001CD0  F7FFB508 6CC3FE99 1D1A79D8 00803001 BD081810 F000B508 B508F85D F85AF000 4671B402 00490849 00495C09 BC02448E 46C04770 46C04778 E59FC000 E08CF00F  .......l.y...0█.........]......Z...
00001D10  FFFFEFB4 46C04778 E59FC000 E08CF00F FFFFEFB0 46C04778 E59FC000 E08CF00F FFFFEFAC 46C04778 E59FC000 E08CF00F FFFFEFA8 46C04778 E59FC000 E08CF00F  ....xG.F............xG.F..........
00001D50  FFFFEFA4 46C04778 E59FC000 E08CF00F FFFFFAB0 46C04778 E59FC000 E08CF00F FFFFFA90 46C04778 E59FC000 E08CF00F FFFFFAA0 46C04778 E59FC000 E08CF00F  ....xG.F............xG.F..........
00001D90  FFFFFAA0 46C04778 E59FC000 E08CF00F FFFFFB18 46C04778 E59FC000 E08CF00F FFFFEF50 46C04778 E59FC000 E08CF00F FFFFFA2C 46C04778 E59FC000 E08CF00F  ....xG.F............xG.F........P.
00001DD0  FFFFEF3C 46C04778 E59FC000 E08CF00F FFFFEF38 46C04778 E59FC000 E08CF00F FFFFFA84 46C04778 E59FC000 E08CF00F FFFFFACC 46C04778 E59FC000 E08CF00F  <...xG.F........8...xG.F..........
00001E10  FFFFFA08 46C04778 E59FC000 E08CF00F FFFFF9E8 46C04778 E59FC000 E08CF00F FFFFF9F8 8101B108 8400B0B0 00000000 8101B108 8400B0B0 00000000 8101B108  ....xG.F............xG.F..........
00001E50  8400B0B0 00000000 8101B108 8400B0B0 00000000 8101B108 8400B0B0 00000000 8101B108 8400B0B0 00000000 8101B108 8400B0B0 00000000 8101B108 8400B0B0  ..................................
00001E90  00000000 8101B108 8400B0B0 00000000 8101B108 8400B0B0 00000000 8101B108 8400B0B0 00000000 8101B108 8400B0B0 00000000 8101B108 8400B0B0 00000000  ..................................
00002018  00000039 00000020 00000007 0000000A 00000020 00000029 00000013 00000002 0000003A 0000000C 00000011 00000031 0000003B 0000000B 00000007 6C6C6548  9... .............. ...)...........:.
00002058  726F776F 7900646C 35536E49 62213736 554F4E63 77763856 66654443 615A5958 51506F64 31784752 54686733 73727170 6D6C6B48 74464532 4C4B4A75 6A694D7A  oworld.yInS567!bcNOUV8vwCDefXYZado
00002098  39304241 57005734 30636C65 3054656D 43555458   004654                                                                                       AB094W.Welc0meT0XTUCTF.
00003EAC  00000D28 00000000 00000000                                                                                                                 (...........
00003FB0  00001585 0000158F 00001599 00004020 00002018 00001ED0 00004030 00000000 00000000 00000000 00000000 0000400C 00004008 00004010 00004014 00004018 0000401C  ............. @... .......0@........
00003FF0  00004020 00004024 00004028 0000402C 00000000       ?? 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   @..$@..(@..,@.......? ............
00004030  00000000                                                                                                                                   ....
00004100  00004004 00004004 00004004                                                                                                                 .@...@...@..
```

把表考出来，编写python代码获得flag

```
table = (0x39,0x20,0x07,0x0A,0x20,0x29,0x13,0x02,0x3A,0x0C,0x11,0x31,0x3B,0x0B,0x07)

a = "yInS567!bcNOUV8vwCDefXYZadoPQRGx13ghTpqrsHklm2EFtuJKLzMijAB094W"
b = "Welc0meT0XTUCTF"
c = ''

for i in range(len(b)):
 c+=(a[table[i]])
 print(i)
 print(table[i])
 print(a[table[i]])
 print("----")

print c
```

得到flag：A1!N1HenBUCu0O!