目录

[Reverse 2](#_Toc14639)

[一. IDA 2](#_Toc4519)

[二. C&C++语言语法常见函数 5](#_Toc26642)

[三. 脱壳 7](#_Toc22798)

[1. 工具脱壳 7](#_Toc11858)

[2. Kali脱壳 7](#_Toc26375)

[四. APK 8](#_Toc16345)

[五. 常见函数 8](#_Toc26437)

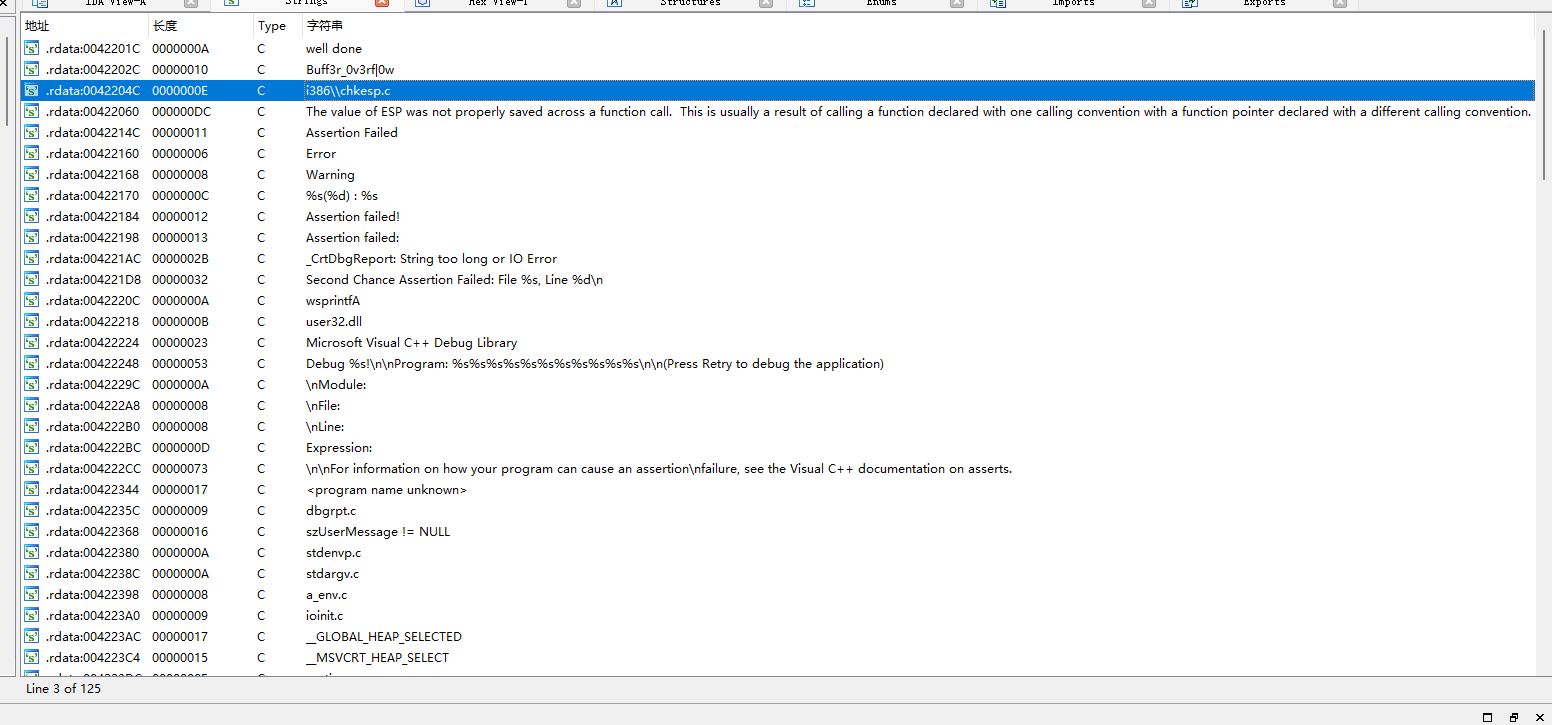
[六. Dnspy 11](#_Toc27462)

[七. Perl 12](#_Toc160)

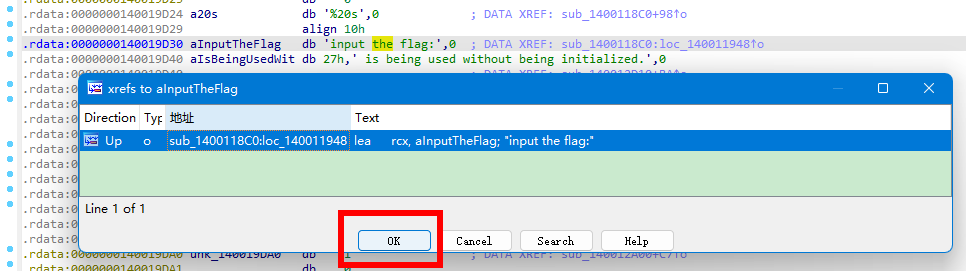
# Reverse

## IDA

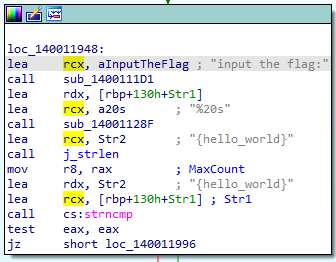
SHIFT+F12查看所有IDA程序中字符串，双击字符串进入内部查看



Ctrl+X查看当前引用位置

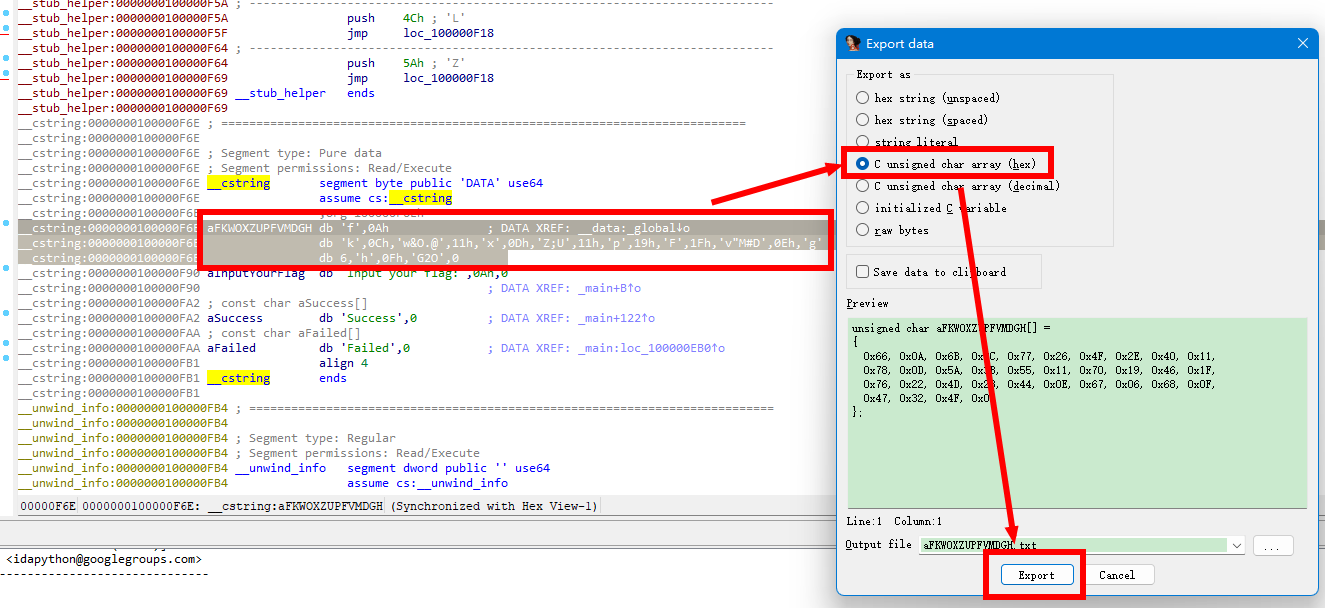


点击OK查看

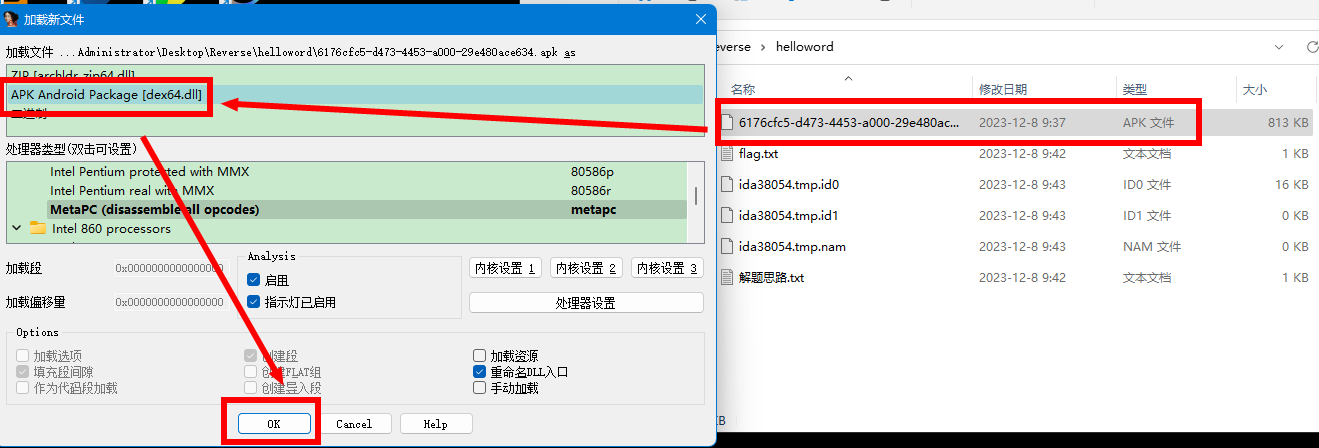


Esc回退

选中数据Shift+E可以将数据导出为任意格式

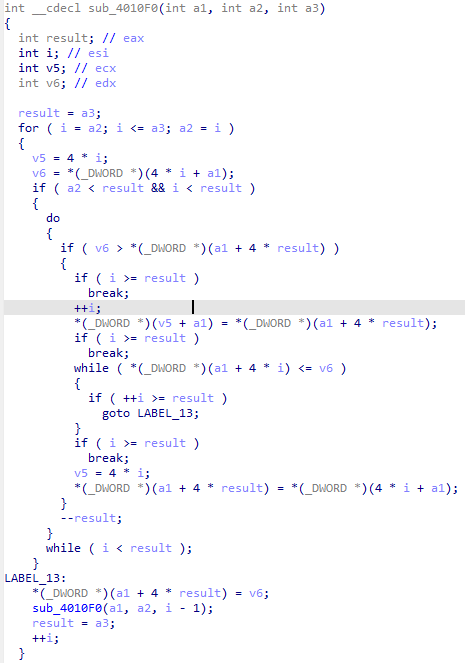


IDA查看APK文件



IDA字符串格式转换：右键字符串选择格式即可

伪代码转换为C语言代码，按F5同时双击



在数字变量按R键即可显示数值



使用Angr脚本模板解答exe文件得到结果，注意修改文件名、避免走到的分支地址，需要找到的结果地址



| **ida快捷键** | **功能** |
| --- | --- |
| F7 | 单步步进 |
| F8 | 单步步过 |
| F9 | 继续运行程序 |
| F4 | 运行到光标所在行 |
| Ctrl + F7 | 直到该函数返回时才停止 |
| Ctrl + F2 | 终止一个正在运行的进程 |
| F2 | 设置断点 |

调试快捷键

打断点之后，左键点击两次变量查看其值，选中数据范围按Shift+K可以得到数组格式数据并复制



## C&C++语言语法常见函数

atoi(x)：x为string，将字符串转为整数，注意”123xyz”会变成123，”-12.34”会变成-12，”123w45”会变成123，” 123w456”会变成123，”123 456”会变成123

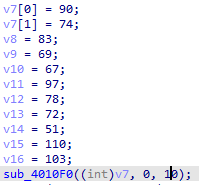
strncmp(a, b, 0x21uLL)代表：将字符串a和字符串b的前0x21个字符进行比较

strcat(a, b)代表：得到字符串为a+b

int占四个字节，所以需要\*4。如果是char类型或者系统作为下标，就不需要，例如a1 + result \* 4的值就是a1[result]，而v5+a1的值就是a1[v5]



IDA伪代码中，例如v开头接数字的，表示为一个字符串，如果函数调用(int)v7，则代表从7开始后的字符串的int数组，例如下面的v实际上是：ZJSECaNH3ng



函数地址重置并覆盖：例如：off\_201028 = sub\_6EA;即将off\_201028地址的函数赋值为函数sub\_6EA

arr中按照数值数目得到键(数值)值(数值个数)对：sumerizeArr = Counter(arr)

UnDecorateSymbolName(a, b, c, d)函数中根据b值(假设是：private: char \* \_\_thiscall R0Pxx::My\_Aut0\_PWN(unsigned char \*))确定a的值确定方式：

无论 \_\_cdecl，\_\_fastcall还是\_\_stdcall调用方式，函数修饰都是以一个“?”開始，后面紧跟函数的名字。 再后面是參数表的開始标识和 依照參数类型代号拼出的參数表。即：a = ?My\_Aut0\_PWN

对于C++的类成员函数（其调用方式是thiscall）。 函数的名字修饰与非成员的C++函数稍有不同，首先就是在函数名字和參数表之间插入以“@”字 符引导的类名。即：a = ?My\_Aut0\_PWN@R0Pxx

其次是參数表的開始标识不同，公有（public）成员函数的标识是“@@QAE”,保护（protected）成员函数的标识是 “@@IAE”,私有（private）成员函数的标识是“@@AAE”，假设函数声明使用了constkeyword，则对应的标识应分别为 “@@QBE”，“@@IBE”和“@@ABE”。即：a = ?My\_Aut0\_PWN@R0Pxx@@AAE

參数表的拼写代号例如以下所看到的：

X--void

D--char

E--unsigned char

F--short

H--int

I--unsigned int

J--long

K--unsigned long（DWORD）

M--float

N--double

\_N--bool

U--struct

....

指针的方式有些特别。用PA表示指针，用PB表示const类型的指针。即如果函数返回值类型为 char \* ：a = ?My\_Aut0\_PWN@R0Pxx@@AAE

如果形参类型为unsigned char \* 对应PAE。即v2 = ?My\_Aut0\_PWN@R0Pxx@@AAEPADPAE

參数表后以“@Z”标识整个名字的结束。假设该函数无參数，则 以“Z”标识结束。最终v2 = ?My\_Aut0\_PWN@R0Pxx@@AAEPADPAE@Z

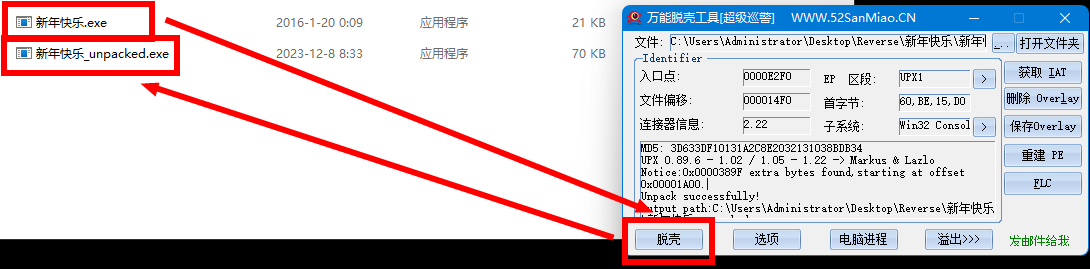
在正向遍历爆破时，如果涉及到移位，可以在向左移位时余上遍历最大值保证数据不会溢出

Strcpy如果存在将多位字符串赋值给少位字符串的情况，为栈溢出，可以通过追加指定函数位置的逆序，实现调用指定函数，例如sub\_40233D，就要在字符串后面加上chr（0x3D）chr（0x23）chr（0x40）

## 脱壳

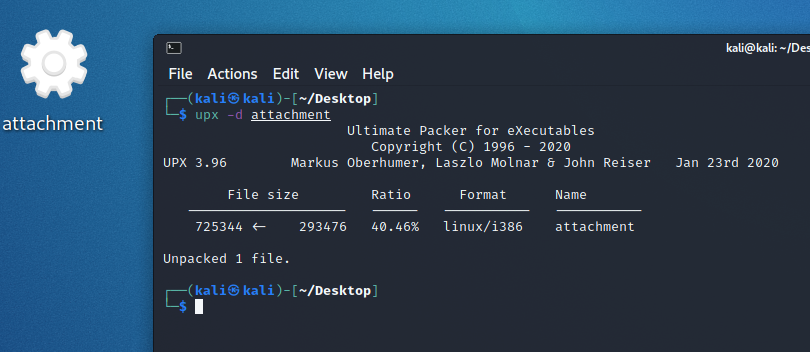
### 工具脱壳

将文件拖入万能脱壳助手后点击脱壳得到脱壳后的文件



### **Kali脱壳**

在文件所在文件夹打开cmd输入：upx -d 文件名

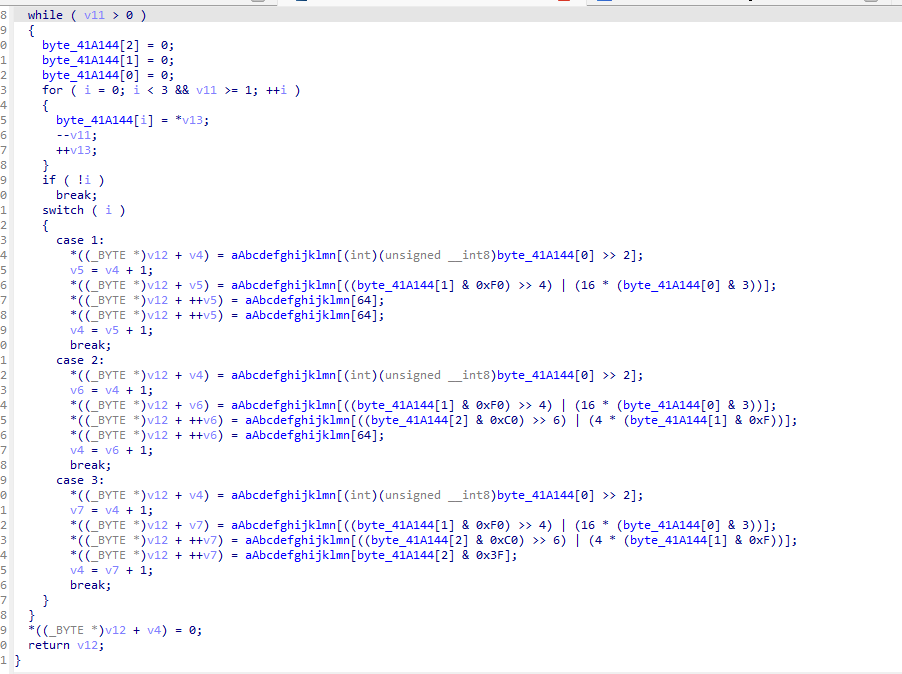


## APK

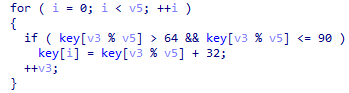
jadx-gui打开apk文件，之后保存到文件夹下，内部如果存在libcore.so文件就用IDA打开查看

## 常见函数

BASE64方法代码

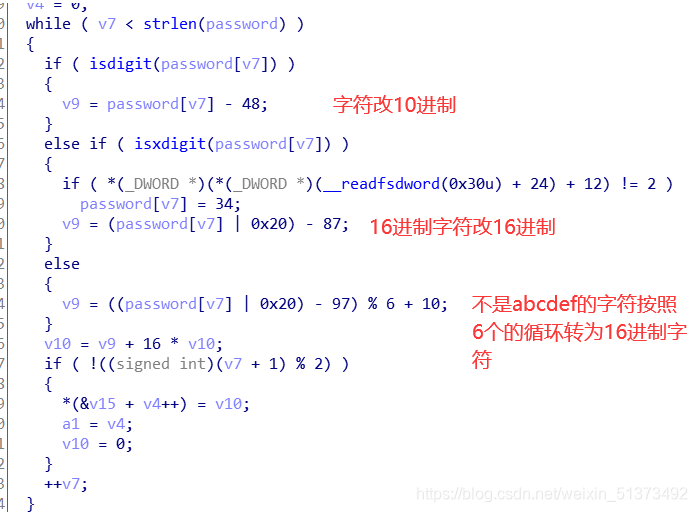


大小写转换方法代码

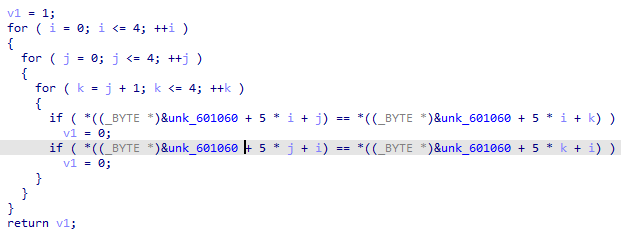


存在两个字符串并且输出存在\*left和\*right可以考虑二叉树两种遍历方式去复原另一种遍历方式

进制转换方法代码



数独算法，外部是矩阵，内部同行同列不允许有相同的值。



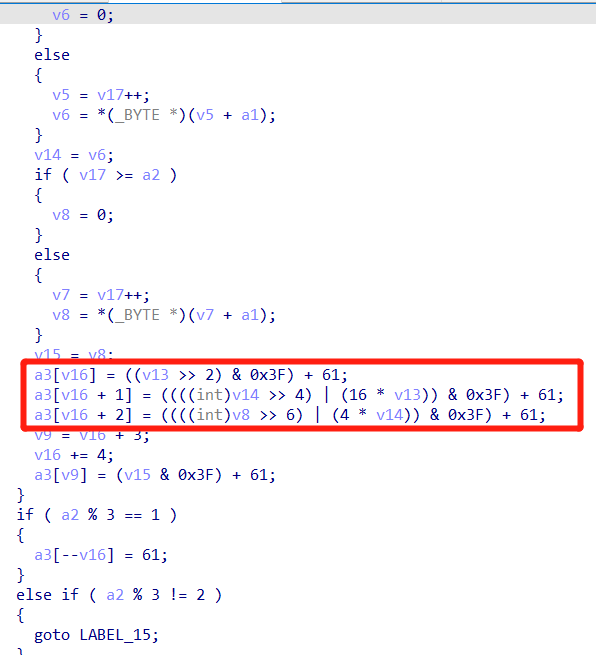
二叉树中序遍历算法



各种base64解码打法



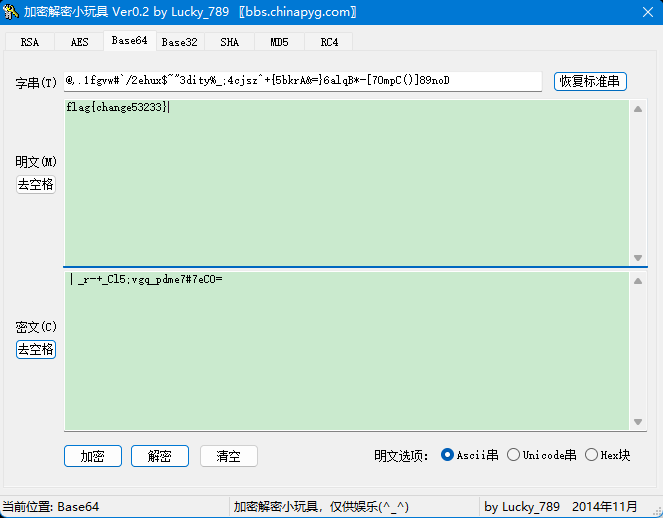
三个字符一组的base64



解码脚本

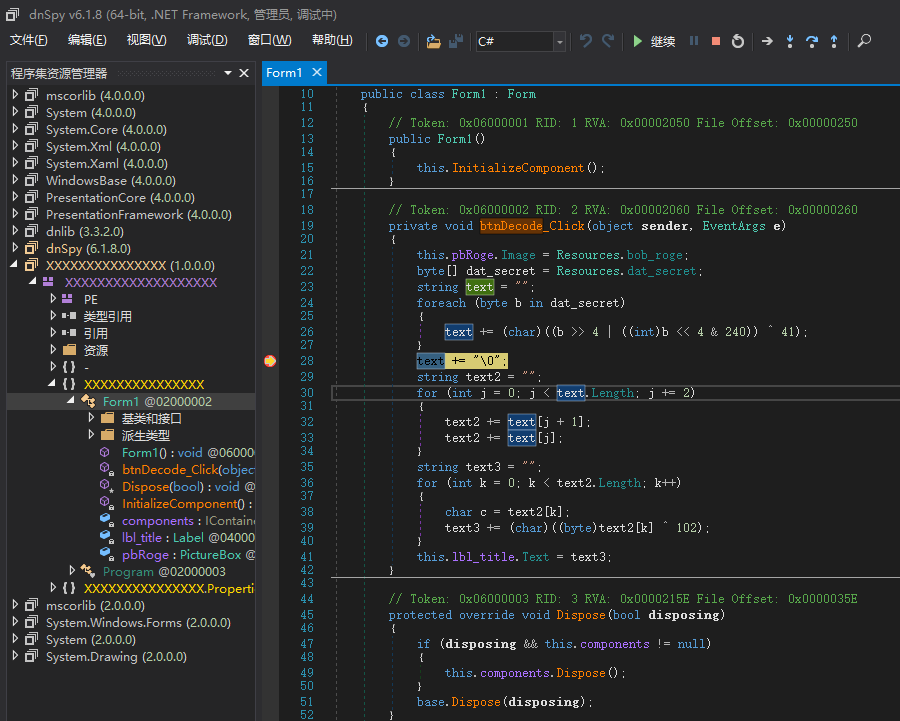


使用加密解密小玩具实现base64码表字串替换解密，注意存在/的时候可能需要删除(作为转义字符)



## Dnspy

反编译.net项目，在程序里可以打断点后点击运行按钮并触发断点操作



## Perl

搜索script相关信息