攻防世界reverse新手之re1

原创

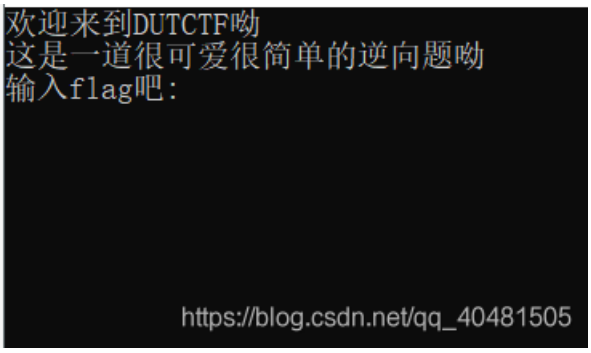
彬彬逊 于 2019-05-06 22:59:18 发布 4336 收藏 9

分类专栏： ctf总结

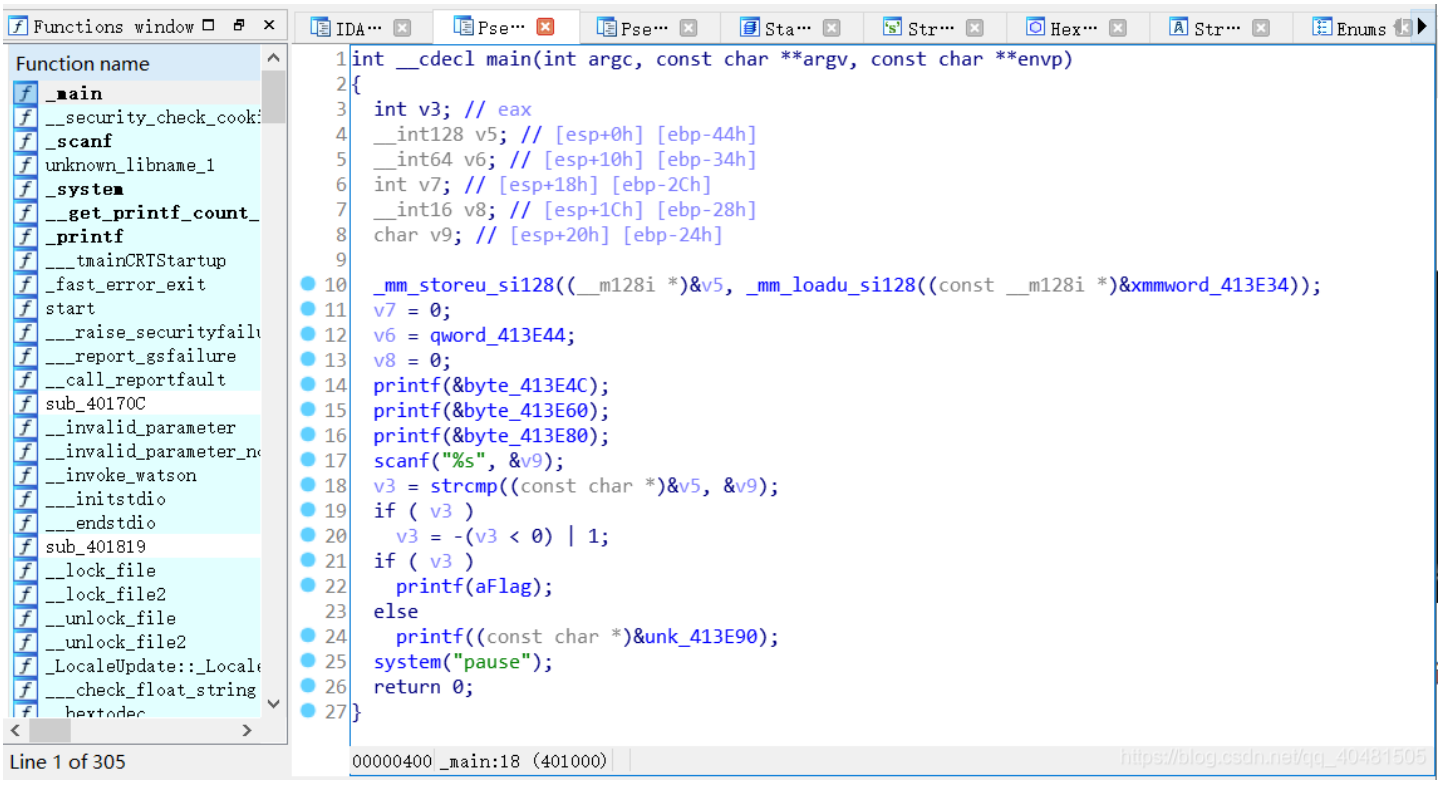
本文链接：https://blog.csdn.net/qq_40481505/article/details/89893552

攻防世界reverse新手之re1

下载附件后发现是exe文件，运行后显示



看来没给什么提示，于是用IDA反编译，按F5能够查看反编译C代码结果



阅读代码发现，程序的功能是将用户输入的flag存入v9，然后将v9和v5比较，如果值相同输出aFlag
既然需要比较flag,那正确的flag应该已经作为一个常量保存在程序内部，可以尝试直接查找flag

首先尝试查看IDA的string界面，view -> open subview -> string 打开

Strings window | Hex View-1 | Structures | Enums | Imports | Exports

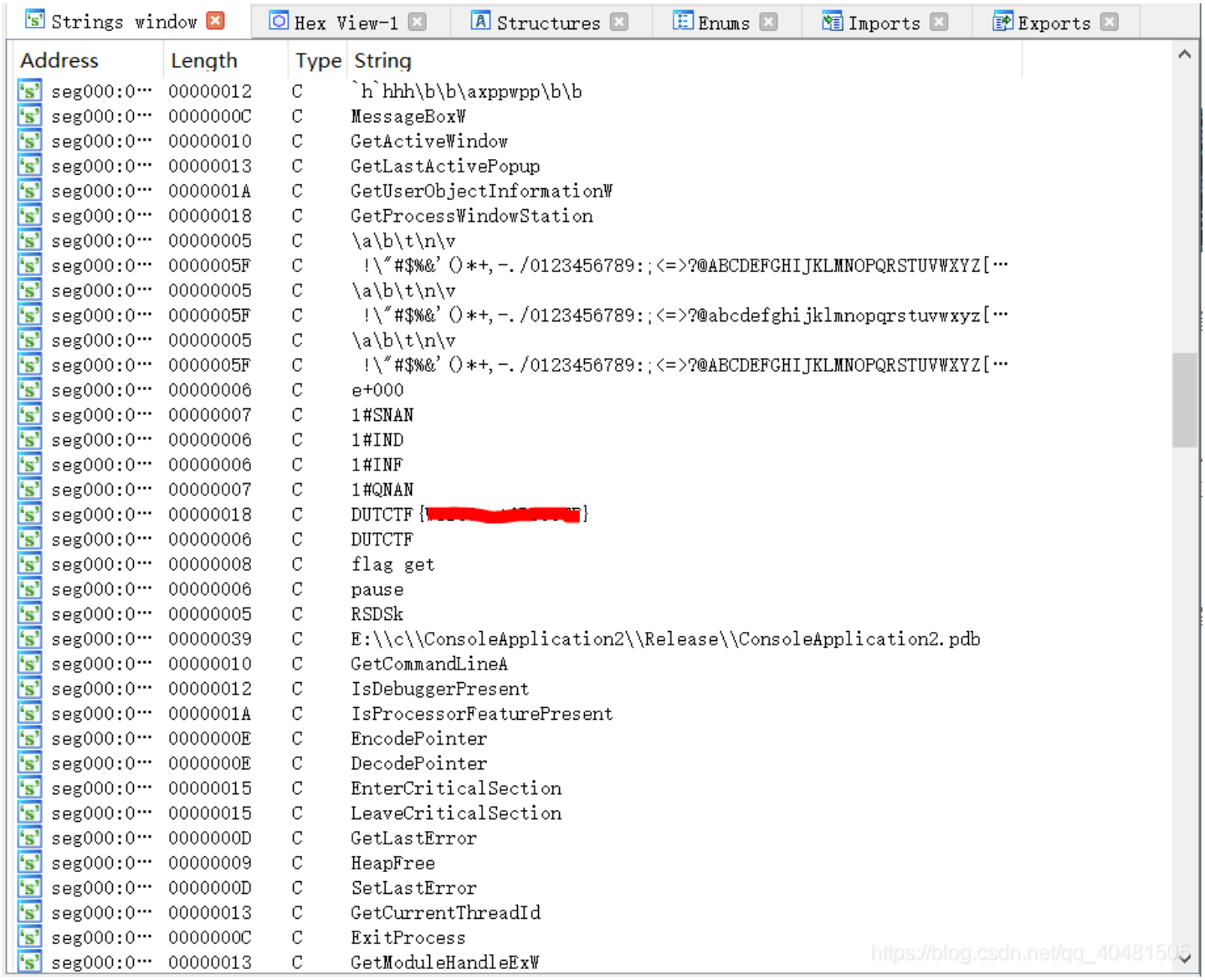
Address	Length	Type	String
.rdata:0…	00000010	C	GetActiveWindow
.rdata:0…	00000013	C	GetLastActivePopup
.rdata:0…	0000001A	C	GetUserObjectInformationW
.rdata:0…	00000018	C	GetProcessWindowStation
.rdata:0…	00000005	C	\a\b\t\n\v
.rdata:0…	0000005F	C	!\"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[…
.rdata:0…	00000005	C	\a\b\t\n\v
.rdata:0…	0000005F	C	!\"#$%&'()*+,-./0123456789:;<=>?@abcdefghijklmnopqrstuvwxyz[…
.rdata:0…	00000005	C	\a\b\t\n\v
.rdata:0…	0000005F	C	!\"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[…
.rdata:0…	00000006	C	e+000
.rdata:0…	00000007	C	1#SNAN
.rdata:0…	00000006	C	1#IND
.rdata:0…	00000006	C	1#INF
.rdata:0…	00000007	C	1#QNAN
.rdata:0…	00000006	C	DUTCTF
.rdata:0…	00000008	C	flag get
.rdata:0…	00000006	C	pause
.rdata:0…	0000000D	C	KERNEL32.dll
.data:00…	0000001B	C	
.data:00…	0000001B	C	abcdefghijklmnopqrstuvwxyz
.data:00…	0000001B	C	ABCDEFGHIJKLMNOPQRSTUVWXYZ
.data:00…	0000001B	C	
.data:00…	0000001B	C	abcdefghijklmnopqrstuvwxyz
.data:00…	0000001B	C	ABCDEFGHIJKLMNOPQRSTUVWXYZ
.data:00…	00000005	C	z?aUY
.data:00…	00000005	C	zc%C1
.data:00…	00000005	C	-64OS

然而没有找到flag，那就换条路试试
使用vim打开，输入‘/’可进入查找模式

emmmm…绕了一圈发现flag直接是明文出现的，按理说是个编辑器打开就能找到，但为什么在IDA里却又找不到了？

查阅资料了解到，反编译的结果不是一定正确的，IDA采用递归下降法进行反编译，它的优点在于很少会在反编译时把数据当作代码来处理，不过这次IDA很明显把flag当成代码，进行了反编译，因此在string界面无法找到flag

解决方法是在IDA打开文件时选择binary file,在此模式下IDA不会进行反编译，此时再打开strings界面就可发现flag



由于在程序代码编译产生可执行文件过程中信息量是减少的，导致反编译器在反编译过程中需要做很多“猜”的工作，因此反编译结果也是可能存在错误。