攻防世界reverse新手之re1

原创

彬 地 于 2019-05-06 22:59:18 发布 ● 4336 ☆ 收藏 9

 分类专栏: ctf总结

 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/qq_40481505/article/details/89893552



ctf总结 专栏收录该内容

43 篇文章 1 订阅 订阅专栏

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

欢迎来到DUTCTF呦 这是一道很可爱很简单的逆向题呦 输入flag吧:
https://blog.csdn.net/qq_40481505

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 🛛	/iew-1 🗵	🖪 Structures 🗵	🔃 Enums 🗵	🛅 Imports 🗵	🛃 Exports 🛽	3
Address Length			Туре	String					^
's'	.rdata:0…	00000010	С	GetActive	e₩indow				
's'	.rdata:0…	00000013	С	GetLastA	ctivePopup				
's'	.rdata:0…	0000001A	С	GetUserOł	ojectInformation₩				
's'	.rdata:0…	00000018	С	GetProces	ss₩indowStation				
ʻs'	.rdata:0…	00000005	С	\a\b\t\n	\v				
's'	.rdata:0…	0000005F	С	!\″#\$%&'	()*+,/0123456789:	<=>?@ABCDEFGHI	JKLMNOPQRSTUVWXY:	z[…	
's'	.rdata:0…	00000005	С	\a\b\t\n	\ v				
's'	.rdata:0…	0000005F	С	!\″#\$%&'	0*+,/0123456789:	<=>?@abcdefghi	jklmnopqrstuvwxy:	z[…	
's'	.rdata:0…	00000005	С	\a\b\t\n	\ v				
's'	.rdata:0…	0000005F	С	!\″#\$%&'	'() ∗+,−. /0123456789:	<=>?@ABCDEFGHI	JKLMNOPQRSTUVWXY:	z[…	
's'	.rdata:0…	00000006	С	e+000					
's'	.rdata:0…	00000007	С	1#SNAN					
's'	.rdata:0…	00000006	С	1#IND					
's'	.rdata:0…	00000006	С	1#INF					
's'	.rdata:0…	00000007	С	1 #QNAN					
's'	.rdata:0…	00000006	С	DUTCTF					
's'	.rdata:0…	00000008	С	flag get					
's'	.rdata:0…	00000006	С	pause					
's'	.rdata:0…	000000D	С	KERNEL32.	.dll				
's'	.data:00…	0000001B	С						
's'	.data:00…	0000001B	С	abcdefghi	ijklmnopqrstuvwxyz				
's'	.data:00…	0000001B	С	ABCDEFGHI	IJKLMNOPQRSTUVWXYZ				
's'	.data:00…	0000001B	С						
's'	.data:00…	0000001B	С	abcdefghi	ijklmnopqrstuvwxyz				
's'	.data:00…	0000001B	С	ABCDEFGHI	IJKLMNOPQRSTUVWXYZ				
's'	.data:00…	00000005	С	z?aUY					
's'	.data:00…	00000005	С	zc%C1					481505
's'	.data:00…	00000005	С	-640S					······································

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

's	Strings	window 🗵	🖸 Hex	View-1 🗵	🖪 Structures 🗵	🛅 Enums 🗵	🛅 Imports 🗵	🛃 Exports 🗵	
A	ddress	Length	Тур	e String					^
's'	seg000:0	00000012	С	h`hhh\b\	b\axppwpp\b\b				
's'	seg000:0	··· 0000000C	С	MessageBo	хΨ				
's'	seg000:0	00000010	С	GetActive	Window				
's'	seg000:0	00000013	С	GetLastAc	tivePopup				
's'		··· 0000001A	С	GetUserOb	jectInformationW				
's'	seg000:0	00000018	С	GetProces	sWindowStation				
's'	seg000:0	00000005	С	\a\b\t\n\	v				
's'	seg000:0	0000005F	С	!\″#\$%&'	()*+,/0123456789:	;<=>?@ABCDEFGHI	IJKLMNOPQRSTUVWXY	z[…	
's'	seg000:0	00000005	С	\a\b\t\n\	V				
's'	seg000:0	0000005F	С	!\″#\$%&'	()*+,/0123456789:	;<=>?@abcdefghi	ijklmnopqrstuvwxy	z[…	
's'	seg000:0	00000005	С	\a\b\t\n\	v				
's'	seg000:0	··· 0000005F	С	!\″#\$%&'	()*+,/0123456789:	;<=>?@ABCDEFGHI	IJKLMNOPQRSTUVWXY	z[…	
's'	seg000:0	00000006	С	e+000					
's'	seg000:0	00000007	С	1#SNAN					
's'	seg000:0	00000006	С	1#IND					
's'	seg000:0	00000006	С	1#INF					
's'	seg000:0	00000007	С	1 #QNAN					
's'	seg000:0	00000018	С	DUTCTF {	}				
's'	seg000:0	00000006	С	DUTCTF					
's'	seg000:0	00000008	С	flag get					
's'	seg000:0	00000006	С	pause					
's'	seg000:0	00000005	С	RSDSk					
's'	seg000:0	00000039	С	$E: \ Co$	nsoleApplication2\\	Release\\Consol	leApplication2.pd	Ь	
's'	seg000:0	00000010	С	GetComman	dLineA				
's'	seg000:0	00000012	С	IsDebugge	rPresent				
's'	seg000:0	··· 0000001A	С	IsProcess	orFeaturePresent				
's'	seg000:0	••• 0000000E	С	EncodePoi	nter				
's'	seg000:0	··· 0000000E	С	DecodePoi	nter				
's'	seg000:0	00000015	С	EnterCrit	icalSection				
's'	seg000:0	00000015	С	LeaveCrit	icalSection				
's'	seg000:0	0000000D	С	GetLastEr	ror				
's'	seg000:0	00000009	С	HeapFree					
's'	seg000:0	000000D	С	SetLastEr	ror				
's'	seg000:0	00000013	С	GetCurren	tThreadId				
's'	seg000:0	··· 0000000C	С	ExitProce	SS				
's'	seg000:0	00000013	С	GetModule	HandleEx₩		nttps://blog	.csan.net/qq_404	81500

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。



创作打卡挑战赛 赢取流量/现金/CSDN周边激励大奖