

171107 逆向-SWPU (Re300-解密)

原创

奈沙夜影 于 2017-11-07 23:41:52 发布 270 收藏

分类专栏: [CTF](#)

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/whklhyyy/article/details/78473965>

版权



[CTF 专栏收录该内容](#)

163 篇文章 4 订阅

订阅专栏

1625-5 王子昂 总结《2017年11月7日》【连续第403天总结】

A. 08067CTF-re300

B.

上文说到看到一个非常复杂的算法

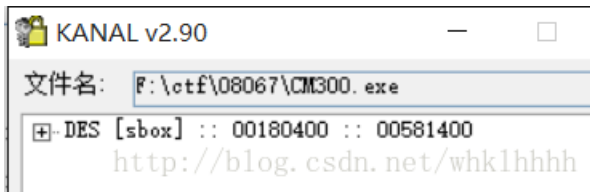
扒了一半想起来就算正向实现了这么复杂也没可能做出逆向的啊.....

果断放弃

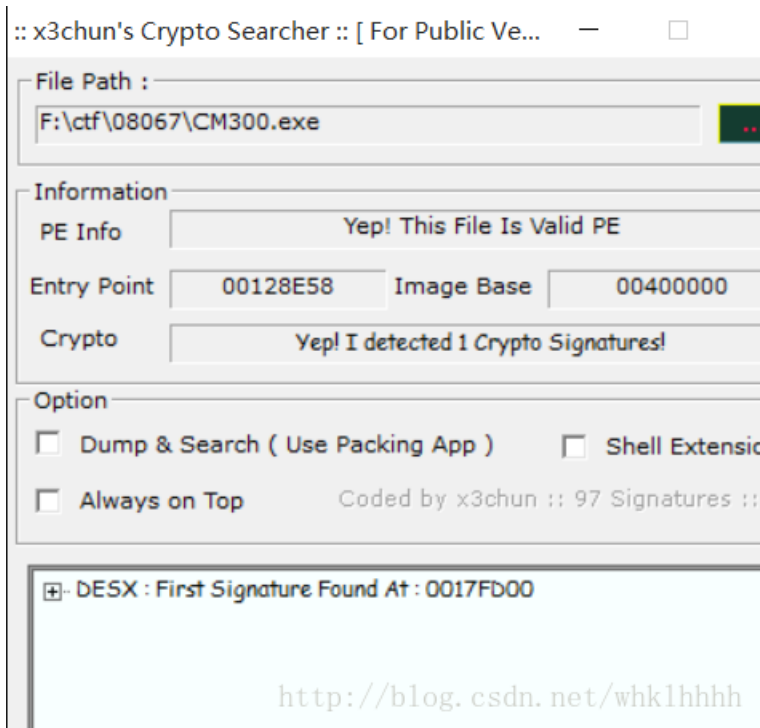
参考官方WriteUp得知是DES加密, 通过工具分析或者代码特征发现都可以
后者需要熟悉DES加密的算法, 我对现代密码学都很苦手.....

找了几个工具试下, 最后只有两个能查到:

PEiD中附带的, 最新版的kanal (v2.9):



CryptoSearcher:



确认是DES算法的话就好办了，DES的解密算法就是加密算法将轮密钥逆序
轮密钥是16个64位的值，合在一起长度为0x80，就是memcpy来的v9，值存在0x5A1308中

```
if ( v2 <= 8 )
{
    memset(&input + v2, 0, 8 - v2); // 用0补齐8个字节
    i58 = *( _DWORD *) (input + 4); // 后4个字符
    i04 = *( _DWORD *) input; // 前4个字符
    i58_ = i58;
    v10 = BYTE3(i04) | ((BYTE2(i04) | ((BYTE1(i04) | ((unsigned __int8)i04 << 8)) << 8)) << 8);
    v11 = BYTE3(i58) | ((BYTE2(i58) | ((BYTE1(i58_) | ((unsigned __int8)i58 << 8)) << 8)) << 8); // 字符串转整数
    memcpy(&v9, &unk_5A1308, 0x80u); // 密钥复制
    sub_401B20((int)&v10, (int)&v9); // 加密变换
    LOBYTE(i04) = BYTE3(v10);
    BYTE1(i04) = BYTE2(v10);
    BYTE2(i04) = BYTE1(v10);
    BYTE3(i04) = v10;
    LOBYTE(i58_) = BYTE3(v11);
    BYTE1(i58_) = BYTE2(v11);
    BYTE2(i58_) = BYTE1(v11);
    BYTE3(i58_) = v11;
    v10 = i04;
    v11 = i58_; // 整数转回字符串
    v4 = 0;
    while ( *((_BYTE *)&v10 + v4) == byte_581500[v4] ) // 校验结果
    {
        if ( ++v4 >= 8 )

```

<http://blog.csdn.net/whklhxxx>

<http://blog.csdn.net/whklhxxx>

密文为0x581500处存的，即

重新运行，在OD中将明文改为上述内容（要在字符串转整数之前，不然需要手动转换）

The screenshot shows the OllyDbg interface with the following components:

- Assembly Window:** Displays assembly code for the `GetWindowText` function. The instruction at address `011E3113` is `lea edx,dword ptr ds:[ecx*4]`, which is highlighted in purple. A red arrow points to this instruction.
- Registers Window:** Shows the state of registers. `ECX` contains the ASCII string "123".
- Memory Dump:** Located at the bottom, it shows the memory address `00A6F1F0` containing the ASCII string "123". A red arrow points to this memory location.

这里断到GetWindowTextA的缓冲区，直接修改它的值就可以了

（注意明文长度为8，后一个比特要改为00，作为字符串结束标志，否则复制时将认为字符串超长而直接终止）

然后修改轮密钥，k1~k16改为k16~k1

```

31 1E 0B 16 05 07 2F 20 26 03 1A 39 2E 32 20 27
2C 35 31 28 14 28 0A 1F 19 3F 1A 30 04 16 16 39
19 34 37 1E 3B 12 10 0B 35 08 13 2A 36 37 25 18
26 29 06 0D 3F 00 34 1D 2D 3D 1A 0E 05 29 1B 18
33 0A 3E 10 1C 3E 2B 20 00 2B 25 19 3F 28 00 2E
2D 2E 2F 20 05 25 0E 1C 03 25 06 1C 0B 17 16 27
0F 14 35 3E 1A 2F 03 14 16 0A 01 11 1E 3A 37 15
04 3E 15 17 31 28 1E 0C 3D 04 1E 22 2A 3D 0B 0B
    
```

将它覆盖轮密钥，密文覆盖明文：

这里是变换的两个参数：
轮密钥和明文

这是轮密钥，需要将它逆序放置。
下一行就是明文转成的数值（前8字节为所用的值，后8字节为明文的副本，变换中不会使用）

地址	HEX 数据	ASCII
00A6F200	3D 04 1E 22 2A 3D 0B 0B 04 3E 15 17 31 28 1E 0C	= " * = > 1 (
00A6F214	16 0A 01 11 1E 3A 37 15 0F 14 35 3E 1A 2F 03 14	. : 7 5 > /
00A6F224	03 25 06 1C 0B 17 16 27 2D 2E 2F 20 05 25 0E 1C	% ' - / %
00A6F234	00 2B 25 19 3F 28 00 2E 33 0A 3E 10 1C 3E 2B 20	. + % ? (. 3 . > > +
00A6F244	2D 3D 1A 0E 05 29 1B 18 26 29 06 0D 3F 00 34 1D	- = & ? &) . ? 4
00A6F254	35 08 13 2A 36 37 25 18 19 34 37 1E 3B 12 10 0B	5 * 6 % 4 7 ; ;
00A6F264	19 3F 1A 30 04 16 16 39 2C 35 31 28 14 28 0A 1F	? ? 0 9 . 5 1 ((.
00A6F274	26 03 1A 39 2E 32 20 27 31 1E 00 16 05 07 2F 20	& 9 . 2 ' 1 ; ; ; /
00A6F284	0B 9C 13 A8 6C 9C A5 0E A8 13 9C 08 0E A5 9C 6C	跳 上 源 ? 湖 1
00A6F294	65 8F D9 4E 18 F4 A6 00 5B 1B 32 01 00 00 00 00	e 德 N 瑞 . [2 去 ...
00A6F2A4	B4 F2 A6 00 2E AE 1E 01 00 00 00 00 F0 FC A6 00	打 ? . ? 去 ... 瘦 ?

覆盖的双击弹窗修改窗口，取消保持大小的勾以后直接粘贴就行了

Edit data beginning from address: 00A6F204

ASCII: 1 Δ δ _ α = / & ♡ → 9 . 2 ' , 5 1 < 9 (< . ▼ 1 ? → 0 ♣ _ = 9 1 4 7 ▲

UNICODE: 竹 真 巾 悱 楚 ? 滑 光

HEX +00: 31 1E 0B 16 05 07 2F 20 26 03 1A 39 28 14 28 0A 1F 17 31 1A 30 04 16 16 39 39 19 34 37 1E

保持大小

确定 取消

运行DES函数以后，数据窗口就出现了flag，不过这时候还是数值顺序的，要再往下走到比较结果的地方：

011E31F4	>	8A4C05 E0	mov cl,byte ptr ss:[ebp+eax-0x20]	结果比较
011E31F8	.	3A88 0015360	cmp cl,byte ptr ds:[eax+0x1361500]	
011E31FE	~	75 0E	jnz short CM300.011E320E	
011E3200	.	40	inc eax	
011E3201	.	83F8 08	cmp eax,0x8	
011E3204	^	7C EE	jil short CM300.011E31F4	
011E3206	.	FF35 1CBA380	push dword ptr ds:[0x138BA1C]	

堆栈 ss:[00A6F284]=61 ('a')
 cl=70 ('p')
 跳转来自 011E3204

地址	HEX 数据	ASCII
00A6F204	31 1E 0B 16 05 07 2F 20 26 03 1A 39 2E 32 20 27	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
00A6F214	2C 35 31 28 14 28 0A 1F 19 3F 1A 30 04 16 16 39	,51((. . ? # 0 ; # 9
00A6F224	19 34 37 1E 3B 12 10 0B 35 08 13 2A 36 37 25 18	#47# ; # # 5 # # *67%#
00A6F234	26 29 06 0D 3F 00 34 1D 2D 3D 1A 0E 05 29 1B 18	&) # . ? . 4 # - = # # # # #
00A6F244	33 0A 3E 10 1C 3E 2B 20 00 2B 25 19 3F 28 00 2E	3 . > # # # > + . + % # ? (. .
00A6F254	2D 2E 2F 20 05 25 0E 1C 03 25 06 1C 0B 17 16 27	- . / % # # # % # # # # # # # #
00A6F264	0F 14 35 3E 1A 2F 03 14 16 0A 01 11 1E 3A 37 15	# # 5 > # # / # # . # # : 7 #
00A6F274	04 3E 15 17 31 28 1E 0C 3D 04 1E 22 2A 3D 0B 0B	> # # 1 (# . = # # " * = # # #
00A6F284	61 48 61 5F 53 74 30 70 61 48 61 5F 53 74 30 70	1 a H a _ S t o p a H a _ S t o p a H a _ S t o p

轮密钥后面就出现了flag: aHa_Stop

C. 明日计划
 Re200和400