Exp10 Final "西普杯"北京天津CTF预选赛真题 writeup



写在前面:

1. 为什么做免考?

主要是想挑战一下自己,提高一点能力,之前在学校信安大赛的时候就对这方面有一些涉猎,希望能够通过完 成这两套题目,水平得到进一步的提高。也希望能够给下一届参加信安大赛的学弟学妹们一些方法上的启发。

2. 免考内容:

1. 学校信安大赛网络攻防三等奖

2. 西普杯北京+天津预算选赛CTF12道真题writeup

在这里我选用的是西普杯北京+天津预算选赛的题,发现前面的同学@尽白已经对北京预选赛的第一道题进行了 多方位全角度深层次的剖析,那第一题我也不过多赘述了,直接从第二题开始吧。

免考题:

1. WTF?

题目类型:安全杂项

题目描述:安全杂项-通过奇怪的字符串发现其中隐藏的信息。

点击题目给出的链接后发现是一串奇怪的字符。





在字符串的末尾发现有两个等于号,于是大胆猜测这是一个base64编码,但是现有带窗体的相关工具不支持这 么大的转码量,这时候我们可以使用python编写一个转码程序。不过正好保密督查实验有一个转码工具,直接 拿来使用。

弧 选择C:\WINDOWS\system32\cmd.exe

Microsoft Windows [版本 10.0.17134.765] (c) 2018 Microsoft Corporation。保留所有权利。

C:\Users\oyyx>cd C:\Users\oyyx\Desktop\督查实验二

C:\Users\oyyx\Desktop\督查实验二>base642file.exe wtf.txt answer.txt

C:\Users\oyyx\Desktop\督查实验二>

转码结果略微眼熟。。。这不是平常使用的二维码吗???可是这么大完全扫不出来。

| answer.bt - 记师本 — 日 | × . |
|---|-------|
| 文件(F) 编辑(E) 楷式(O) 查看(V) 帮助(H) | |
| 000000000000000000000000000000000000000 | 000 ^ |
| 000000000000000000000000000000000000000 | 000 |
| 0000000111111111111111111111111111111 | 111 |
| 0000000111111111111111111111111111111 | 111 |
| 000000011111111000000000000000000000000 | 000 |
| 000000011111111000000000000000000000000 | 000 |
| 0000000111111110000000011111111111111 | 000 |
| 0000000111111110000000011111111111111 | 000 |
| 0000000111111110000000011111111111111 | 000 |
| 0000000111111110000000011111111111111 | 000 |
| 0000000111111110000000011111111111111 | 000 |
| 0000000011111111000000001111111111111 | 111 |
| 0000000011111111000000001111111111111 | 111 |
| 0000000111111110000000000000000000000 | 111 |
| 0000000111111110000000000000000000000 | 111 |
| 000000011111111111111111111111111111111 | 000 |
| 000000001111111111111111111111111111111 | 000 |
| 000000000000000000000000000000000000000 | 000 |
| 000000000000000000000000000000000000000 | 000 |
| 000000000000000000000000000000111111111 | 000 |
| 000000000000000000000000000000000000000 | 000 |
| 000000011111111111111111000000000111111 | 000 |
| 000000011111111111111111000000000111111 | 000 |
| 000000000000000000000000000000000000 | 111 |
| 000000000000000000000000000000000000000 | 111 |
| 00000000000000000000000000001111111110000 | 000 |
| 000000000000000000000000000000000000000 | 000 |
| 000000011111111111111111000000000000000 | 000 |
| 000000011111111111111110000000000000000 | 000 |
| 000000000000000111111110000000000000000 | 000 |
| 000000000000000011111111000000000000000 | 000 |
| 000000000000000000000000111111111111111 | 000 |
| 000000000000000000000000000000000000000 | 000 |
| 000000000000000000000000000000000000000 | 111 |
| < | > |

在工具栏——格式——字体里调整记事本字体,进行扫描,得到flag。(这里可能还是会扫不出来,可以通过 图像处理软件调整图片的灰度、亮度,这样扫描的成功率会更高。)

answer.txt - 记事本

- 🗆 X

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)



已识别到二维码内容

flag{QR_Code_666_!!_?}

复制内容

题目类型:逆向工程

题目描述:注意进入正确的流程,用最短的步骤走完迷宫。

点击链接,下载题目给出的文件,执行后要求输入字符串。随意输入,提示输入错误。



没有头绪,还是直接使用winhex分析文件看看吧。

winhex 昵プ丰丙问 解杜寿 何 吊 枝 切 帾 紭 怫 惋 冻 皊 巫 兽 ザ 安 回 仏 解 杜 棆 柫 咒 俴 夓 吊 枝 斍 任 サ 恨 夓 刦 陪 斍 任 サ 硲 盞 捥 坕 逦 或 皊 敶 捻 | 夷 筏 ザ 吒 旼 安 连 回 仏 讯 佑 眑 制 兼 仁 稑 底 隖 藕 赽 杫 皊 斍 任 咒 敶 捻 ザ 恁 余 杫 诺 昵 プ 歄 靤 帾 专 锟 皊 16 返 刼 缜 辗 噮 ザ 徝 制 ZDNetSoftwareLibrary 亚 春 纭 朆 髞 诊 份 = 拫 朏 彀 夭 皊 叙 绥 教 甮 ザ

将文件拖入winhex,发现一处可疑的地方,这句"Please input your key:"和之前运行时输出的语句一模一样,但是之后一连串字符串还是让人摸不着头脑,这就触及到知识盲区了。

| | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00022FE0 |
|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|
| | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00022FF0 |
| é V | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 56 | 10 | E9 | 18 | 00 | 00 | 00 | 00 | 00023000 |
| @ Good | 64 | 6F | 6F | 47 | 00 | 02 | 90 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 40 | 00023010 |
| ! pause Sorr | 72 | 72 | 6F | 53 | 00 | 00 | 00 | 65 | 73 | 75 | 61 | 70 | 00 | 00 | ΟA | 21 | 00023020 |
| y you are wrong! | 21 | 67 | 6E | 6F | 72 | 77 | 20 | 65 | 72 | 61 | 20 | 75 | 6F | 79 | 20 | 79 | 00023030 |
| Please i | 69 | 20 | 65 | 73 | 61 | 65 | 6C | 50 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | OA | 00023040 |
| nput your key: | 00 | 0A | ЗA | 79 | 65 | 6B | 20 | 72 | 75 | 6F | 79 | 20 | 74 | 75 | 70 | 6E | 00023050 |
| ******** | 20 | 2A | 20 | 2A | 00 | 00 | 00 | 00 | 00023060 |
| ** * ** ** * | 20 | 2A | 20 | 2A | 2A | 20 | 2A | 2A | 20 | 2A | 20 | 2A | 2A | 20 | 20 | 20 | 00023070 |
| ** ** * #* ** ** | 2A | 2A | 20 | 2A | 2A | 20 | 2A | 23 | 20 | 2A | 20 | 2A | 2A | 20 | 2A | 2A | 00023080 |
| ** ** **** | 2A | 2A | 2A | 2A | 2A | 20 | 20 | 20 | 20 | 20 | 20 | 2A | 2A | 20 | 2A | 2A | 00023090 |
| **** | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 2A | 2A | 2A | 2A | 000230A0 |
| Too long! | 00 | 00 | ΟA | 21 | 67 | 6E | 6F | 6C | 20 | 6F | 6F | 54 | 00 | 00 | 00 | 00 | 000230B0 |
| Sorry,you ar | 72 | 61 | 20 | 75 | 6F | 79 | 2C | 79 | 72 | 72 | 6F | 53 | 00 | 00 | 00 | 00 | 00023000 |
| e wrong! | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ΟA | 21 | 67 | 6E | 6F | 72 | 77 | 20 | 65 | 000230D0 |

但是这道题的类型是逆向工程,是不是可以尝试使用逆向工具来进行实验。

查询一番我找到了一个所谓"逆向神器"的IDA Pro(Interactive Disassembler Professional),即交互式 反汇编器专业版。

安昵直勢朆棘岭ブ丰靟怇叓缜诗轵任= 乀伝夠Oday东畒皊或呞咒ShellCode宏兮刌枖什壱专叵罀尗皊刯噮 (IDA Pro昵ブ歄亪亘引皊= 叵缜稑皊= 叵扯屛皊= 夠久琌噮皊= 亪叏Windows或Linux WinCE MacOS幹叶乁杀杫刌枖稑 底= 袱兲讪乀朆妃皊芷钷叵仫乶制皊逌吗巫稑刯噮ザIDA Pro巸绕或乀云室丐皊刌枖敒愕仩砇皊栍凌幼讯兼臰 躱迋逥或乀敁刁砚穼颌埥皊釓覇巫兽ザ安攵指敶升秓CPU挍仪雌兼丳匋拲Intel x86= x64= MIPS= PowerPC= ARM= Z80= 68000= c8051筏筏ザ

将可执行文件拖入ida中,发现是难以琢磨的汇编语言,将其反汇编成C语言。

寻找一番之后发现可疑代码!(点击加号查看)

```
1 char *sub_401020()
2 {
3 char *result; // eax
```

± =

```
char *v1; // [esp+50h] [ebp-CCh]
4
5
     char v2; // [esp+54h] [ebp-C8h]
     char v3; // [esp+5Dh] [ebp-BFh]
 6
 7
     char v4; // [esp+94h] [ebp-88h]
8
     char v5; // [esp+98h] [ebp-84h]
9
     char v6; // [esp+99h] [ebp-83h]
10
     __int16 v7; // [esp+10Dh] [ebp-Fh]
     char v8; // [esp+10Fh] [ebp-Dh]
11
     char v9; // [esp+110h] [ebp-Ch]
12
13
     char v10; // [esp+114h] [ebp-8h]
     int v11; // [esp+118h] [ebp-4h]
14
15
16
     v5 = 0;
     memset(&v6, 0, 0x74u);
17
     v7 = 0;
18
19
     v8 = 0;
     strcpy(&v2, "******* *
                                                                            *******");
                                  ** * ** ** * ** ** #* #* ** **** **
20
     v1 = &v3;
21
     printf("Please input your key:\n");
22
23
     gets(&v5);
     if ( strlen(&v5) != 22 )
24
25
     {
26
       printf("Sorry you are wrong!\n");
27
       system("pause");
       exit(1);
28
29
     }
30
     v11 = 0;
     do
31
32
     {
33
       v10 = *(&v5 + v11);
       if ( v10 != 107 && v10 != 106 && v10 != 104 && v10 != 108 )
34
35
       {
36
         printf("Sorry you are wrong!\n");
37
         system("pause");
         exit(2);
38
39
       }
40
       v9 = *(&v5 + v11);
       switch ( v9 )
41
42
       {
43
         case 104:
           if ( --v1 < &v2 || v1 > &v4 || (result = (char *)*v1, result == (char *)42) )
44
45
           {
             printf("Sorry you are wrong!\n");
46
47
             system("pause");
             exit(3);
48
           }
49
50
           if ( *v1 == 35 )
51
           {
52 LABEL 41:
             printf("Good!\n");
53
54
             system("pause");
             exit(0);
55
56
           }
57
           break;
         case 106:
58
59
           v1 += 8;
60
           if ( v1 < &v2 || v1 > &v4 || *v1 == 42 )
61
           {
62
             printf("Sorry you are wrong!\n");
63
             svstem("pause"):
```

```
-,---- ,,
- -
64
             exit(3);
65
           }
           result = (char *)*v1;
66
           if (result == (char *)35)
67
68
             goto LABEL_41;
69
           break;
70
         case 107:
71
           v1 -= 8;
72
           if ( v1 < &v2 || v1 > &v4 || *v1 == 42 )
73
           {
             printf("Sorry you are wrong!\n");
74
75
             system("pause");
76
             exit(3);
77
           }
78
           result = v1;
           if ( *v1 == 35 )
79
80
             goto LABEL_41;
           break;
81
82
         default:
           if ( ++v1 < &v2 || v1 > &v4 || *v1 == 42 )
83
84
           {
             printf("Sorry you are wrong!\n");
85
             system("pause");
86
87
             exit(4);
88
           }
89
           result = v1;
           if ( *v1 == 35 )
90
             goto LABEL_41;
91
92
           break;
93
       }
94
       ++v11;
95
     }
     while ( v11 < 25 );</pre>
96
97
     return result;
98 }
```

View Code

经过分析我们发现只能输入ASCII码为107、106、104、108的字符,即为k、j、h、1

根据v5相关代码易发现v5代表输入的字符串,而这个字符串长度必须刚好是22,与奇怪的字符串内空格的数量 相等。寻思着是不是输入的hjk1与星号的排列有什么联系,而题目描述表明了这是一个走迷宫的题目。鼓捣一 番字符串之后发现这是一个8*8的矩阵迷宫!而输入的hjk1显然就是vim编辑器里常用的上下左右,这决定了迷 宫的走向!

```
strcpy(&v2, "
                          v1 = &v3;
printf("Please input your key:\n");
gets(&v5);
if ( strlen(&v5) != 22 )
{
  printf("Sorry you are wrong!\n");
  system("pause");
  exit(1);
}
*******
* *
       *
* * ** *
* * ** *
 * #* *
*
* **** *
*
       *
*******
```

再对代码进行分析,发现只有路径不经过星号字符串才有效,而到达井号即为到达终点。这正好解答了上一步 中空格数量为什么与应输入字符串的长度相等

```
if ( ++v1 < &v2 || v1 > &v4 || *v1 == 42 )
{
    printf("Sorry you are wrong!\n");
    system("pause");
    exit(4);
    }
    result = v1;
    if ( *v1 == 35 )
        goto LABEL_41;
    break;
}
```

于是尝试将vim编辑器内的hjk1代表的左上下右直接套入题目中,迷宫路线即为jjjjj11111kkkkkhhhjjj1



解密成功。

3.1000000

题目类型:逆向工程

题目描述:寻找正确的输入。

有了之前题目的铺垫,做这道题轻车熟路了些。

显示打开可执行文件,发现还是一道输入的题目。



把文件拖入ida内进行分析,发现主函数内关键代码:

```
main();
 v17 = 0;
 memset(&v18, 0, 0x10u);
 memset(&v4, 0, 0x14u);
 v4 = -26;
 v5 = -20;
 v6 = -31;
 v7 = -25;
 v8 = -70;
 v9 = -12;
 v10 = -27;
 v11 = -13;
 v12 = -12;
 v13 = -12;
 v14 = -27;
 v15 = -13;
 v16 = -12;
 v19 = 0;
 puts(&byte_403024);
 scanf("%s", &v17);
 LOBYTE(v19) = 0;
 while ( *((_BYTE *)&v17 + v19) )
    *((_BYTE *)&v17 + v19++) = 0x80u;
 if ( !strcmp((const char *)&v17, &v4) )
   printf("good");
 else
   printf("wrong");
 return 0;
}
```

通过联系其他函数进行分析后可以知道,这个可执行程序是先压栈一些字节,用户输入字符串之后,将其与80 作or运算,将运算结果与压栈的字节进行比较,若相同则答案正确,不同则错误。 所以我们需要做的是先取出压栈字符,再将其与80作xor运算,即可得到应当输入的字符。

编写一个C语言程序,进行遍历破解。



得到解密文本: flag: testtest。

4.大数据问题

题目类型:编程

题目描述:小明刚刚学习计算机编程,老师给他出了这样一道题目,但是他怎样思考,都做不出来,于是,只 好请教高手的你了。

求sum = 1!+2!+3!+·····+6788!+6789!的末5位。

如果直接莽过去的话,电脑会直接死机以示抗议。

所以我们在每一步的计算中直接区最后五位,不考虑其他的位数,可以大大减少计算量,并且我们发现这道题 不用计算25以后的阶乘,因为4*5*10*25=100000,25之后的的阶乘对后五位已经不产生影响了。

编写程序进行计算:



得到答案为40313。

5. 打不开的文件

题目类型:隐写术

题目描述: 隐写 咦! 这个文件怎么打不开?



右键另存,可以看到这是一个gif文件,但是文件仍然打不开。但在使用winhex查看的时候,我发现并没有文件头。

| monese - [ox.ge] | | - U X |
|--------------------|---|---|
| the Edit Search Po | shorn Vew Tools Specialit Options Western Mer | |
| , HOLD | | |
| ve Data | | |
| e ege | | [unregistered] |
| | | N. Of C.S. Inaminanto EDIENT |
| | | e este selle en este en este este este este este es |
| | CONCERNENT IN COLOR OF A FA | File size: 109 KB |
| | CONCOME. F9 F9 F6 F6 F5 F5 FF FF F6 F6 F2 F2 4C 4C 4C F2 F2 Howekkkehold [14] | 111,064 bytes |
| | CONDUCTOR F1 CF CF C5 53 53 53 0A CA CA CA FC FC FA 7F 7F 7F ATTIENT WALLET | Read-only model |
| | CONDINAL DA DE DE FA FD FC DE DE DE 20 20 20 43 43 43 45 (DEMONINAL CONB | Contract of the second |
| | 00000070 AE AK F2 F2 F2 C0 01 01 33 33 33 C9 C9 C9 D1 D1 00000 333FFFS0 | Creation time: 06/01/2019 |
| | CONDUCED D1 F2 ED EC 1C 1C 1B C6 C5 C5 F9 FC FA E2 DB DC Bo11 #MauseOU | 20 20 40 |
| | 00000000 FF FE FF C2 C1 C1 B6 B6 B6 B9 B9 B9 24 24 24 EF vouldider: 10001 | Last write time: 05/01/2019 |
| | 000000A0 EF EF B1 B1 B1 FA FA FS FC FE FE A9 A9 A9 14 13 11###0000000 | 20.28.50 |
| | CODOCCED 13 F6 FB F9 EB E3 E4 C2 BC BC B2 AB AB FB FE FB OSCHRAMMENTERS | Attributes: |
| | DODODOCO FD FA FC A3 3A 98 D4 C8 CC D8 D4 D4 FE FC FC 62 youtH0ED00bush | R0918. 9 |
| | COSCOCCO 5A 58 78 78 78 78 78 93 88 88 F5 F8 F6 FC F6 F7 73 68 Z[xxx1110eouo-sk | Mode: Ter |
| | 000000E0 68 FE FB FE CC C3 C3 84 78 7C F7 F1 F1 51 49 4A kbsblikks(1-88013 | Character set ANDLASC |
| | 000000F0 13 0A 0A FF FB FA 79 73 73 F2 F0 EE 22 1A 1A 96 980980051" 1 | Offsets heradeoma |
| | 00000100 95 95 F5 F4 F2 BA B4 B3 43 3A 3B 8B 83 84 AA A3 115669"*C::111*E | alas ha baha ana ana ana |
| | 00000110 A3 F6 F6 F4 33 2A 2A 5B 52 53 6A 62 63 F8 FA F8 E0003+*[R5]bcsta | Window # |
| | 00000120 1A 0A 08 38 32 33 D0 D9 D0 48 43 43 18 12 13 F6 ;23708KCC 6 | |
| | 00000130 EE EF E6 E1 E0 28 22 23 E3 DF DE EF E9 E9 FF FF 11m64+*#880+144999 | Cliptoard available |
| | 00000140 FB E8 E8 E7 D5 CE CF D8 D1 D1 D8 D8 D7 D5 D1 CD seec01108800+081 | TEMP tolder 30.0-08 the |
| | 00000150 13 05 06 C7 8D 8F 8F 89 89 CF C9 C8 F3 F2 F0 89 C6CC**IEE008* | Users/ognAppDatalLocalTemp |
| | CODODIGO AE AF FD FE F9 E7 DE DF CA CS C7 EB EB EA EC F7 € ybsc>EEEceeei- | |
| | 00000170 F3 B4 AF AE C6 C0 C0 AF A9 A9 97 8F 90 F5 F6 F8 6 "MAA 001 800 | |
| | 00000180 F2 F5 F0 86 81 81 F8 F9 FA 83 7F 7E 67 50 5E FA 0004119319 | |
| | 00000190 F9 FC F0 F7 F4 A1 A0 A0 88 87 85 65 60 60 E4 E4 998+01 >-pe" as | |
| | 00001AD E3 80 79 79 C2 EF BE 4A 3E 3F F6 F3 F3 AB A7 A6 8FyyAC&37000+5; | |
| | COD011B0 BF BF BF F4 F6 F4 F9 F9 F9 50 30 30 FE F8 F7 41 COOM11C00pu-A | |
| | 0000100 40 40 79 6E 70 FC FE 18 07 08 16 03 05 19 0E 00yaputh | |
| | CONCULIDO UN EZ EZ ED UN CO UN AS YE YE EN EN EN YA YO AN INI INCONTINO | |
| | CONCLED 70 77 78 25 20 17 51 50 50 CC CD CD 25 15 17 80 (MYA (CPVIES) 4 | |
| | CONCEPTED BY AN | |
| | CONVERSE FF EF EF 11 07 07 07 07 07 07 07 07 07 07 07 07 07 | |
| | CONCERNMENT OF THE CALL OF AN AN AND AN | |
| | CONCEPT TO TO OD OD OD OT OT DO TO CO CO DE | |
| | 00000240 \$2 \$1 \$5 \$5 \$5 \$1 \$1 \$5 \$6 \$6 \$6 \$2 \$2 \$2 \$2 \$4 \$6 \$6 \$5 \$1 \$4 \$400071111184 | |
| | 00000350 24 24 F1 D0 D4 AF A4 A7 9F 05 04 0F 04 04 86 55457 (144) | |

gif文件的文件头是GIF8,直接在notepad++内加入文件头即可。



更改之后直接打开,得到一个4帧的gif动图,找准时机分别对截取图片,发现得到的是一个base64编码。



通过base64解码得到flag为: this is a gif

dGhpcyBpcyBhIGdpZg==

Base64解密 :

this is a gif

6. 疑惑的汉字

题目类型:密码学

题目描述:现有一段经过加密的密文,内容如下:

王夫 井工 夫口 由中人 井中 夫夫 由中大 请找出这段密文隐藏的消息明文。

一般来说带汉字的密码类题目都是简单密码,不涉及更深层次的计算,而且初见这道题为何如此眼熟。。。似 里! 行测! 行测里的图形题也有涉及到类似的知识与规律。

于是我尝试着用行测的方法来进行解题,终于发现了规律:汉字有非闭合部分几个分支就代表着是数字几。

得到结果: 67 84 70 123 82 77 125

对照ascii码表进行转换,得到结果:

 $CTF \{RM\}$

后来查了一下,这叫当铺密码(从名字看得出来应该是中国古代勤劳的劳动人民发明的吧。。)

7. 功夫秘籍

题目类型:安全杂项

题目描述: 传说得到这个秘籍的人都修炼成了绝世神功

点击题目链接,下载到文件kongfu.rar,文件无法打开,把文件拖入winhex中进行分析,发现这其实是一个png 图片。

| Case Data | bungfu rar | | |
|-----------|--|-----------|---|
| File Edit | Offset 0 1 2 3 4 5 6 7 8 9 A B C D E F | | Incredistered |
| | 00000000 89 50 4E 47 0D 0A 1A 0A 00 00 0D 49 48 44 52 IPNG IHDR | | kungtu.rar |
| | 00000010 00 00 1 E3 00 00 0D D8 02 00 00 07 12 8D 8 Ý | | C:\Users\oyyx\Downloads |
| | 00000020 06 00 00 03 73 42 49 54 08 08 08 DB E1 4F E0 sBIT ÚdOà | | |
| | 00000030 00 00 00 09 70 48 59 73 00 00 0E C4 00 00 0E C4 pHYs Ä Ä | | File Size. 137 ND 139 929 hytes |
| | 00000040 01 95 2B 0E 1B 00 00 20 00 49 44 41 54 78 9C 9C I+ IDATXII | | |
| | 00000050 BD 5B 92 24 49 72 24 C8 2C A2 6A E6 11 91 59 AF ½['\$Ir\$Ê.cj@ 'Y | | [Read-only mode] |
| | 00000060 2E F4 03 68 0C 06 83 05 40 0B D0 D2 EC D7 DC 66 .6 h I @ ĐÒi×Ùf | | |
| | 00000070 FF E6 6B 8F B5 87 D9 B3 00 68 34 2A 33 E3 E1 6E jvek µlûs h4*3ăén | | Creation time: 05/01/2019 22:06:37 |
| | 00000080 A6 2A C2 FB 21 E6 1E 9E D5 D5 43 4B B0 8E 2E 8A :*Aui@ 100CK*1.1 | | Lockerite Kenne Official Control |
| | 00000090 F4 B0 B0 30 57 13 65 61 61 79 38 FF E3 79 FF 9F 6**0W eaay8ÿäyÿI | | Last write time. 00/01/2019 22:06:45 |
| | 000000A0 FF D7 FF 6C AD F5 DE 7B EF EE 4E 12 80 24 FC F2 yxy1-3≥(iiN \$\$ub | | Allelbudger A |
| | 000000B0 21 E5 45 48 29 80 24 09 4E CD 48 0D E5 94 66 44 AEH) \$ NIH A FD | | Icons: 0 |
| | 00000000 CC B8 C4 3E F6 71 19 63 8F 08 03 33 33 46 64 26 1,Å>öq c 33Fd& | | |
| | 000000D0 22 33 73 CE CC CC 3D A6 F7 06 D9 25 46 64 C2 0D "3sîlî=;+ Û%FdÂ | | Mode: Text |
| | 000000E0 74 49 99 32 70 31 5F AD 7D EC EB 43 5F 3F B6 75 tIl2p1}i@C_?¶u | | Character set ANSI ASCII |
| | 000000F0 F5 F6 78 7A F8 F8 F8 B4 3C 9C 12 02 B0 3E 3E 9C 80xz000 <1 *>>1 | | Bytes per page: 38x16=608 |
| | 00000100 1E 1E BC 9D 86 3C 61 09 09 03 06 69 BC 9C DF 5E 4 4 <a i448^<="" td=""><td></td><td>Window #</td> | | Window # |
| | 00000110 DE BE 64 E6 7F 7C FE 8F 97 B7 D7 D7 F3 DB E7 CF ▶%dml b I·××6ÛçĬ | | No. of windows: 2 |
| | 00000120 9F F7 98 4A 5E 5E 37 BB 9C 88 46 43 1A E9 A6 06 I+IJ^^7*IFC 6; | | |
| | 00000130 92 E2 7C 7A 5C BF F9 F6 F1 87 6F 3E 7C F3 F1 E1 '4 z\duom1o> oma | | Clipboard: available |
| | 00000140 FB 8F 4F DF 7F 7C 5C BA B7 39 7E BB 7E 5C C9 BE 0 081 \2.9">~\É4 | | TEMP folder: 29.8 GB free |
| | 00000150 F8 B2 2C BD 7B 6B AD BB 91 34 28 22 52 13 00 62 02.54 k-> 4("R b | | Users\oyyx\AppData\Local\Temp |
| | 00000160 66 66 EB B6 2C CB F7 BF FA D5 C7 6F BF F9 E6 FB ff@¶.E+¿úČÇo¿ùæû | | |
| | 00000170 6F D7 A7 D5 1E D6 FE B4 E2 A1 A3 3B 1A 45 29 11 o×\$Õ Oþ'ái£; E) | | |
| | 00000180 10 D1 49 92 4E 2A 72 48 61 A1 CC B4 4C 48 CC 20 NI'N•rHail'LHI | | |
| | 00000190 A0 98 92 84 00 40 3F 1E 8A 24 B0 E9 7A 50 00 60 I'I @? I\$*6zP ` | | |
| | 000001A0 20 00 AB A7 96 BA AD F3 71 CA F4 08 45 66 12 66 «\$12-oqÊo Ef f | | |
| | 000001B0 16 CA B1 C7 18 03 09 77 3F 2D 4B EF BD A1 91 E4 求 w?-Ki%i'ä | | |
| | 000001C0 4C 4A 39 68 64 C4 D8 B6 ED 7C 7E 7D 7E 7E FE F7 LJ9hdAضi[~}~~p+ | | |
| | 000001D0 FF F8 C3 A7 4F 9F 42 30 33 B7 15 A8 5F F5 65 39 ye&SolBO3. "_&e9 | | |
| | 000001E0 F5 B5 7D F8 F0 01 CE 50 EE FB 3E C6 18 63 BC 9D δµ}øð ÎPiù>Æ cM | | |
| | 000001F0 CF E7 F3 F9 F9 F9 F9 F9 59 96 7E 5A CB AE 96 65 79 İçóùùùyY1~ZË@ley | | |
| | 00000200 7C 7C 3C 9D 96 6D 3F 93 6A 6D 21 19 7B EC FB 4E < #r?ijm! {iuN | | |
| | 00000210 F8 B2 2C AD 75 77 5F FB. A9 F7 DE FB DA 5A 23 9C @²,-uw_û0+⊨0022#I | | |
| | 00000220 26 ED FB 98 DB 65 9F 11 B1 EF F3 D3 A7 4F FF F2 &iiilûel ±ióôSoyò | | |
| | 00000230 AF 7F F8 E3 1F FF 98 29 77 37 EF 66 66 D6 00 24 to8 ÿt)w7iffÖ \$ | | |
| | 00000240 4C 84 8C E7 CB 25 B6 4B 46 60 0E 8C 30 6A 71 5F LligEx#KF 10jq_ | | |
| | 00000250 7A 1F 63 DB F7 4B 44 C0 E4 EE 49 05 FD BC 67 60 z c0+KDÅsiI ýMg v | | |
| | Page 1 of 231 Officet 0 = 137 Electr | n/a Size: | ola |

直接修改文件文件后缀,再尝试打开文件,发现文件能够打开了。

| otos - kungfu.png | | | | | | | | - 0 | × |
|-------------------|-------------------|---|---|--------------|---|-----|---|--------|---|
| See all photos | Add to a creation | € | Î | \heartsuit | ୍ | ‰ ~ | Ŕ | ₽ B | |



再在winhex中进行查看,搜索key可以找到结果,不过这里我使用了Stegsolve.jar对图片进行分析。

Stegsolve. jar 昵 プ 歌 圕 僕 隖 冟 巫 兽 = 攵 指 侅 甮 专 吒 斿 引 鯄 陪 圕 僕 隖 冟 = 昵 圕 僕 隖 冟 皊 忋 复 巫 兽 = 幼 业 彙 个 弦 jpg 圕 牍 夜 览 サ 夭 尕 サ 僕 紦 鄃 堀 朲 盾 吒 旼,连 脃 尌 个 丰 斍 任 皊 僕 紦 RGB 偂 退 衒 XOR サ ADD サ SUB 筏 攊 佢 = 眑 脃 听 徝 制 朏 甮 皊 俧 恵 ザ

使用File Format选项查看可以发现有一串字符

| 😹 File Format Analysis | <u></u> | \times |
|---|---------|----------|
| Unsafe to copy unless known to software | | - |
| Hex: | | |
| 49444154 | | |
| Ascii: | | |
| IDAT | | |
| Data length = 323 bytes | | |
| CRC = b004b86a | | |
| Image data, compressed | | |
| Chunk: | | |
| Critical - necessary for display of image MUST BE recognized to proceed | | |
| Public | | |
| Unsafe to copy unless known to software | | |
| Hex: | | |
| 49454e44 | | |
| Ascii: | | |
| IEND | | |
| Data length = 0 bytes | | |
| CRC = ae426082 | | |
| Additional bytes at end of file = 45 | | |
| Dump of additional bytes: | | |
| Hex: | | |
| 6b65792069732056 4639356330733558 | | |
| 7a56796147746658 335647545852394d | | |
| 3056736532353151 4555672020 | | |
| Ascii: | | - |
| key is V F95c0s5X | | |
| zVyaGtfX 3VGTXR9M | | |
| 0Vse251Q EUg | | |
| ОК | | |

还是我们的老朋友base64编码,进行解密之后竟然仍然是一串乱码,不过貌似发现内容里有KEy的字样,猜测是 通过某种编码手段将字符串打乱后的结果。

耗尽了知识储备还是没能解出答案,查了一下相关资料,发现这是一个栅栏密码。

经过每组字数3的解密后,得到答案:

| T <u>ysK9_5rhk_uFMt</u> } <u>3El</u> {nu@E |
|--|
| |
| |
| |
| 每组字数 3 加密 解密 |
| Th3_kEy_ls_{Kun9Fu_M@5tEr} |

题目类型:逆向工程

题目描述: 拿到答案需要正确地点击按钮

先点击链接下载程序,打开程序是四个按钮:

| iii MainWindow | — | × |
|----------------------------|---|---|
| What do you want to input? | | |
| button1 | | |
| button2 | | |
| button3 | | |
| button4 | | |
| | | |
| | | |

将可执行程序放入ida中发现是一个C#程序,我们这里可以使用ILSpy进行反编译。

ILSpy 昵ブ丰弆準龄.NET叓缜诗巫兽= 箆洇彀夭昙甮昵安皊牿徇ザ圮绣夭夠敶惋冻丑= 安鄃脃徎妃皊宒或佦寿 朰矫稑底雌冋郮仩砇皊推紨ザ



界面中的succeed是正确后的输出,所以我们应该讲让flag4满足true,则Correct要等于8,而*correct初始值为0,所以需要通过flag3那里来增加correct的值。经过分析后得到,输入的内容要为数组array[1][~]array[8]的值。对应的数字也就是按钮上的数字。*



按照13421234的顺序点击按钮,得到flag。

simCTF{Easy_to_Get_Flag_From_DotNET}

9. 解码磁带

题目类型:安全杂项

题目内容:

+ -

你的老板刚刚得到了一卷磁带,但与一般的磁带不同的是,在这圈磁带上有一些字符'o'和下划线''。由于你学过计算机和信息加解密,自 然而然,解码磁带的这项任务就落到了你肩上。为了帮助你解码,下面会先给出一个解码样例: 解码样例: 0___0_ 00__0_0 00_0__0 00_0_0_ 00_0__0 00_000_ 00__000 _0_000_ 上面的磁带片段解码为: Beijing. _____ ----0_0_000 00_0___ 00__0_0 000_0_ 00__0_0 _0___ 000_0__ 00_0___ 00__0_0 000_0_ 00__0_0 _0__ 00_0__0 000__00 _0____ 00___0 _0___ 000_000 00_0__0 00_00__ 00_00__ _0_00__ 000_0__ 00_0___ 00__0_0 000_0_ 00__0_0 _0____ 00_0__0 000_00 _0____ 00___0 _0____ 000_000 00___0 0000_0 _0_000_ 那么,现在该你解码了.....

key格式: simCTF{}

View Code

我们发现,如果将第一段的o替换成1,__替换成0,可以得到以下字符,正好是解码之后的结果Beijing对应的二进制ASCII码

因此将第二段的字符替换成相应的0和1:

+-

通过查询ascii表得到结果:

Where there is a will, there is a way.

10. 想看正面? 那就要看仔细了

题目类型: 隐写术

题目内容:看背影是不是很nice!想看正面?那就要看仔细了!

点击链接后发现是一张图片。



右键另存图片,拖入winhex内进行分析,发现可疑的key

| WinHex - (meinv.jpg) | View Tools Spe | | - 0 × |
|----------------------|----------------|--|--|
| | neiny, jpg | N C C C C C C C C C C C C C C C C C C C | |
| Case Data | Offset | 0 1 2 3 4 5 6 7 8 9 & B C D F F | |
| File Eğit | 00001300 | 74 2E 63 6F 6D 2F 70 68 6F 74 6F 2F 31 2E 30 2F t.com/photo/1.0/ | [unregistered] |
| | 00001310 | 22 2F 3E 3C 72 64 66 3A 44 65 73 63 72 69 70 74 "/> <rpre>rdf:Descript</rpre> | C:\Users\oyyx\Downloads\oggit |
| | 00001320 | 69 6F 6E 20 72 64 66 3A 61 62 6F 75 74 3D 22 75 ion rdf:about="u | |
| | 00001330 | 75 69 64 3A 66 61 66 35 62 64 64 35 2D 62 61 33 uid:faf5bdd5-ba3 | File size: 172 KE |
| | 00001340 | 64 2D 31 31 64 61 2D 61 64 33 31 2D 64 33 33 64 d-11da-ad31-d33d | 176,449 bytes |
| | 00001350 | 37 35 31 38 32 66 31 62 22 20 78 6D 6C 6E 73 3A 75182f1b" xmlns: | [Read-only mode] |
| | 00001360 | 64 63 3D 22 68 74 74 70 3A 2F 2F 70 75 72 6C 2E dc="http://purl. | |
| | 00001370 | 6F 72 67 2F 64 63 2F 65 6C 65 6D 65 6E 74 73 2F org/dc/elements/ | Creation time: 06/11/2019 |
| | 00001380 | 31 2E 31 2F 22 3E 3C 64 63 3A 73 75 62 6A 65 63 1.1/"> <dc:subjec< td=""><td>11.03.35</td></dc:subjec<> | 11.03.35 |
| | 00001390 | 74 3E 3C 72 64 66 3A 42 61 67 20 78 6D 6C 6E 73 t> <rdf:bag td="" xmlns<=""><td>Last write time: 06/11/2013 11:03:38</td></rdf:bag> | Last write time: 06/11/2013 11:03:38 |
| | 000013A0 | 3A 72 64 66 3D 22 68 74 74 70 3A 2F 2F 77 77 77 :rdf="http://www | (Well-share) |
| | 000013B0 | 2E 77 33 2E 6F 72 67 2F 31 39 39 39 2F 30 32 2F .w3.org/1999/02/ | Icons: |
| | 000013C0 | 32 32 2D 72 64 66 2D 73 79 6E 74 61 78 2D 6E 73 22-rdf-syntax-ns | |
| | 000013D0 | 23 22 3E 3C 72 64 66 3A 6C 69 3E 6B 65 79 20 69 #"> <rdf:li>key i</rdf:li> | Mode: Tex |
| | 000013E0 | 73 20 59 6D 56 68 64 58 52 35 3C 2F 72 64 66 3A s YmVhdXRE <td>Character set ANSIASC Officiels: bevadecima</td> | Character set ANSIASC Officiels: bevadecima |
| | 000013F0 | 6C 69 3E 3C 2F 72 64 66 3A 42 61 67 3E 0D 0A 09 1i> | Bytes per page: 38x16=60 |
| | 00001400 | 09 09 3C 2F 64 63 3A 73 75 62 6A 65 63 74 3E 3C < | Window # |
| | 00001410 | 64 63 3A 74 69 74 6C 65 3E 3C 72 64 66 3A 41 6C dc:title> <rdf:al< td=""><td>No. of windows:</td></rdf:al<> | No. of windows: |
| | 00001420 | 74 20 78 6D 6C 6E 73 3A 72 64 66 3D 22 68 74 74 t xmlns:rdf="htt | |
| | 00001430 | 70 3A 2F 2F 77 77 77 2E 77 33 2E 6F 72 67 2F 31 p://www.w3.org/1 | Clipboard: available |
| | 00001440 | 39 39 39 2F 30 32 2F 32 32 2D 72 64 66 2D 73 79 999/02/22-rdf-sy | TEMP folder: 26.5 GB free |
| | 00001450 | bE 74 b1 78 2D bE 73 23 22 3E 3C 72 b4 bb 3A 6C ntax-ns#"> <rdr:1< td=""><td>Users\oyyx\AppData\Local\Temp</td></rdr:1<> | Users\oyyx\AppData\Local\Temp |
| | 00001460 | 69 20 78 60 6C 3A 6C 61 6E 67 30 22 78 20 64 65 1 xml:lang='x-de | |
| | 00001470 | 66 61 75 6C 74 22 3E 66 63 63 E7 9C 68 E6 AD A3 TAULT 761 61 64 | |
| | 00001480 | LY 9D AZ EF BC 9F LY 8Z A3 ES BU BI LS AB SI L/ e CINIELA TE; C | |
| | 00001490 | 90 00 L4 00 94 L/ 00 00 L4 0A 00 L7 0C 01 3C 27 108/10/04 114 (/ | |
| | 00001480 | | |
| | 00001450 | 3E 00 04 03 03 03 03 02 20 04 03 34 /4 03 /4 06 05 / (Addidated at the second s | |
| | 00001400 | 3F 30 72 64 66 34 41 67 74 07 78 60 67 67 73 34 \center to the monet | |
| | 00001450 | 72 64 63 D 2 68 74 74 70 33 2F 2F 77 77 7F rdf="http://amar. | |
| | 00001450 | 77 33 2E 6F 72 67 2F 31 39 39 39 2F 30 32 2F 32 w3.org/1999/02/2 | |
| | 00001500 | 32 2D 72 64 66 2D 73 79 6E 74 61 78 2D 6E 73 23 2-rdf=svntax-ns# | |
| | 00001510 | 22 3E 3C 72 64 66 3A 6C 69 20 78 6D 6C 3A 6C 61 "> <rdf:li td="" xml:la<=""><td></td></rdf:li> | |
| | 00001520 | 6E 67 3D 22 78 2D 64 65 66 61 75 6C 74 22 3E E6 ng="x-default">@ | |
| | 00001530 | 83 B3 E7 9C 8B E6 AD A3 E9 9D A2 EF BC 9F E9 82 3cl 8-2é ci41éi | |
| | 00001540 | A3 E5 B0 B1 E8 A6 81 E7 9C 8B E4 BB 94 E7 BB 86 £å*±è; clis»ic»i | |
| | 00001550 | E4 BA 86 EF BC 81 3C 2F 72 64 66 3A 6C 69 3E 3C 89 114 (/rdf:li>(| |
| | | | |

提交字符串YmVhdXR5之后发现不是flag,猜测是进行了某种编码加密,于是我进行了尝试。

最后发现是经过了base64加密

Base64 :

YmVhdXR5

Base64解密:

beauty

经过解密得到flag为beauty。(可最后还是没有正面呀!)

11. 无处不在的广告

题目类型:隐写术(预选赛隐写术略多啊)

题目内容: 这次他们决定把它藏起来, 你能找到他么?

点击链接,是超人大战蝙蝠侠



右键另存到本地,拖入winhex,并没有发现任何线索。

既然是隐写,可能这个广告是作为某个图层写入图片了,所以仍然使用Stegsolve.jar对图片进行分析。对图像进行处理之后发现端倪,扫描二维码得到结果。



FLAG: this is a new word

12. 神秘字母

题目类型:密码学

题目内容:在线代的课本上出现了一堆神秘字母

dloguszijluswogany

而旁边的矩阵是

12 01

快找出flag吧

根据密码学的知识,矩阵往往出现在希尔密码中,我决定使用希尔密码进行尝试。先将字母变换为相应的数字:

dloguszijluswogany—>4, 12, 15, 7, 21, 19, 26, 9, 10, 12, 21, 19, 23, 15, 7, 1, 14, 25

对于希尔密文,我们只需要做一次逆变换就可以得到明文。

$$A^{-1} = \frac{1}{|A|}A^*$$

根据公式计算得到密码逆矩阵为

1 -2

0 1

使用逆矩阵与密文相乘再mod26

 $\begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 4 & 15 & 21 & 26 & 10 & 21 & 23 & 7 & 14 \\ 12 & 7 & 19 & 9 & 12 & 19 & 15 & 1 & 25 \end{pmatrix} \pmod{26}$ $\begin{array}{c} \text{(mod26)} \\ \text{(mod26)} \\ \text{(mod26)} \\ \text{(mod26)} \end{array}$

(博客不支持word内公式编辑器,这里我就用图片代替好了)

将数字转为字符得到flagishillissoeasy

完成后的感受:

我觉得这次免考是一个学习的过程,在参加学校信安大赛的时候,题目相对简单一些,但对于逆向相关的题目 我也是直接放弃,没有继续深入。借这次机会,我了解了windows端逆向工具的操作,以后如果还有类似的机会 面对CTF题的时候也许会更加从容吧。

(还有,终于不用写博客了 捂脸.jpg)

转载于:https://www.cnblogs.com/20164309-kx/p/11003620.html