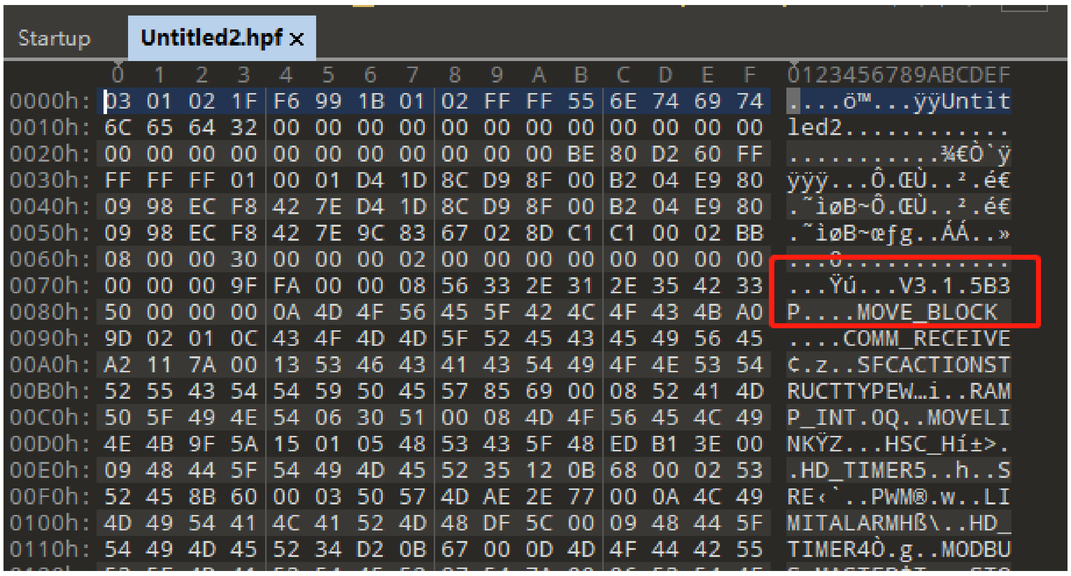
2021-06-工业信息安全技能大赛-线上第一场

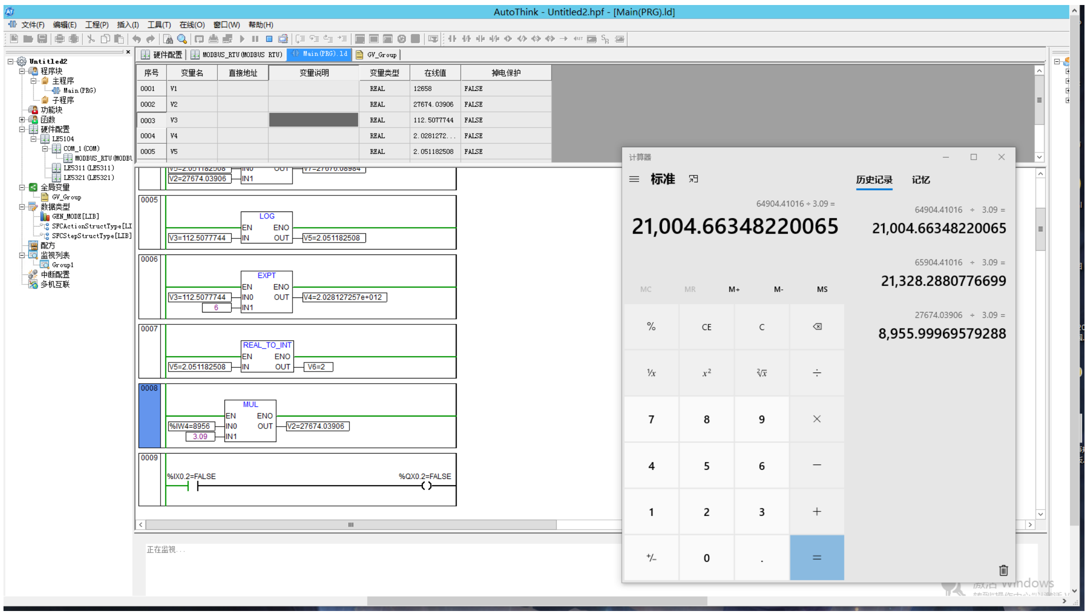
简单的梯形图

和利时V3.1.5B3及以上版本的组态软件AutoThink打开梯形图程序



```
Startup   Untitled2.hpf ×

       0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
0000h: 03 01 02 1F F6 99 1B 01 02 FF FF 55 6E 74 69 74   ....ö™...ÿÿUntit
0010h: 6C 65 64 32 00 00 00 00 00 00 00 00 00 00 00 00   led2............
0020h: 00 00 00 00 00 00 00 00 00 00 00 BE 80 D2 60 FF   ...........¾€Ò`ÿ
0030h: FF FF FF 01 00 01 D4 1D 8C D9 8F 00 B2 04 E9 80   ÿÿÿ...Ô.ŒÙ..².é€
0040h: 09 98 EC F8 42 7E D4 1D 8C D9 8F 00 B2 04 E9 80   .˜ìøB~Ô.ŒÙ..².é€
0050h: 09 98 EC F8 42 7E 9C 83 67 02 8D C1 C1 00 02 BB   .˜ìøB~œƒg..ÁÁ..»
0060h: 08 00 00 30 00 00 00 02 00 00 00 00 00 00 00 00   ...0............
0070h: 00 00 00 9F FA 00 00 08 56 33 2E 31 2E 35 42 33   ...Ÿú...V3.1.5B3
0080h: 50 00 00 00 0A 4D 4F 56 45 5F 42 4C 4F 43 4B A0   P....MOVE_BLOCK 
0090h: 9D 02 01 0C 43 4F 4D 4D 5F 52 45 43 45 49 56 45   ....COMM_RECEIVE
00A0h: A2 11 7A 00 13 53 46 43 41 43 54 49 4F 4E 53 54   ¢.z..SFCACTIONST
00B0h: 52 55 43 54 54 59 50 45 57 85 69 00 08 52 41 4D   RUCTTYPEW…i..RAM
00C0h: 50 5F 49 4E 54 06 30 51 00 08 4D 4F 56 45 4C 49   P_INT.0Q..MOVELI
00D0h: 4E 4B 9F 5A 15 01 05 48 53 43 5F 48 ED B1 3E 00   NKŸZ...HSC_Hí±>.
00E0h: 09 48 44 5F 54 49 4D 45 52 35 12 0B 68 00 02 53   .HD_TIMER5..h..S
00F0h: 52 45 8B 60 00 03 50 57 4D AE 2E 77 00 0A 4C 49   RE‹`..PWM®.w..LI
0100h: 4D 49 54 41 4C 41 52 4D 48 DF 5C 00 09 48 44 5F   MITALARMHß\..HD_
0110h: 54 49 4D 45 52 34 D2 0B 67 00 0D 4D 4F 44 42 55   TIMER4Ò.g..MODBU
```

解题思路



AutoThink - Untitled2.hpf - [Main(PRG).ld]

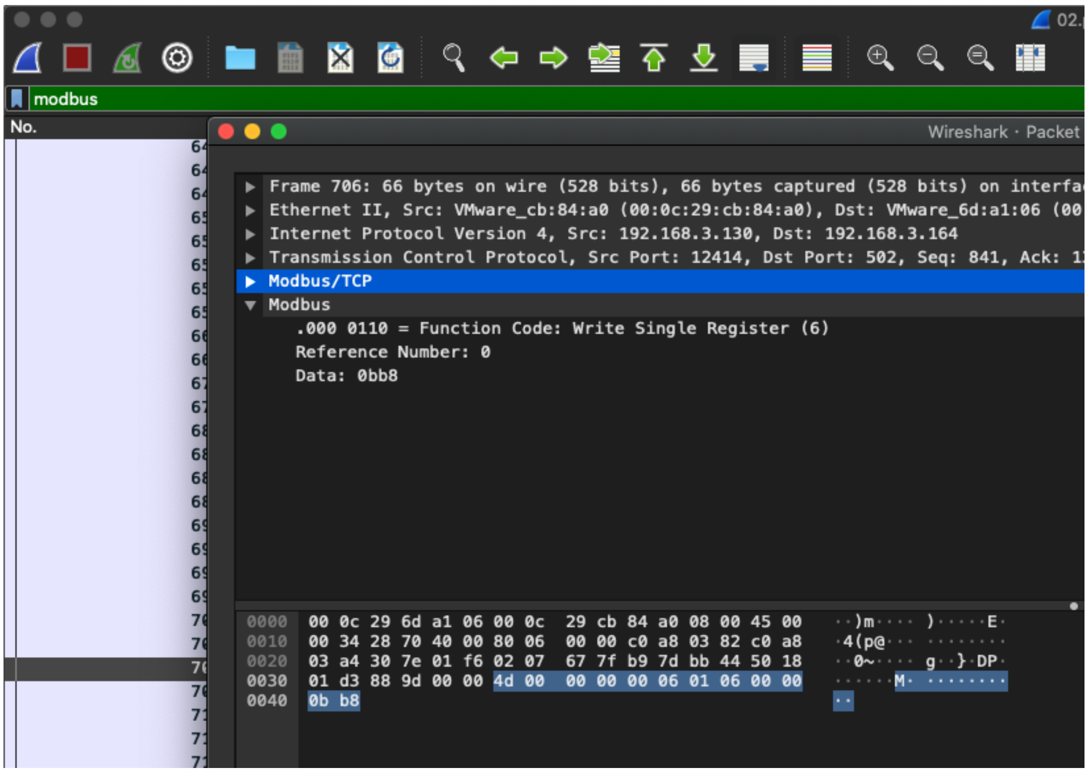
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护
0001	V1			REAL	12658	FALSE
0002	V2			REAL	27674.03906	FALSE
0003	V3			REAL	112.5077744	FALSE
0004	V4			REAL	2.0281272...	FALSE
0005	V5			REAL	2.051182508	FALSE

```
0005  LOG:         IN  V3=112.5077744              OUT V5=2.051182508
0006  EXPT:        IN0 V3=112.5077744  IN1 6       OUT V4=2.028127257e+012
0007  REAL_TO_INT: IN  V5=2.051182508              OUT V6=2
0008  MUL:         IN0 %IW4=8956       IN1 3.09    OUT V2=27674.03906
0009  %IX0.2=FALSE                                  %QX0.2=FALSE
正在监视...
```

```
计算器  标准
64904.41016 ÷ 3.09 =
21,004.66348220065

历史记录
64904.41016 ÷ 3.09 =
21,004.66348220065
65904.41016 ÷ 3.09 =
21,328.2880776699
27674.03906 ÷ 3.09 =
8,955.99969579288
```

flag{21004}

损坏的风机

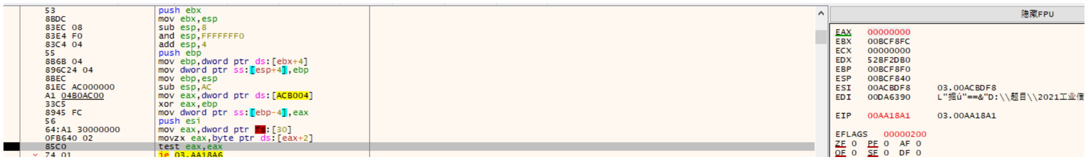


```
▶ Frame 706: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interfa
▶ Ethernet II, Src: VMware_cb:84:a0 (00:0c:29:cb:84:a0), Dst: VMware_6d:a1:06 (00
▶ Internet Protocol Version 4, Src: 192.168.3.130, Dst: 192.168.3.164
▶ Transmission Control Protocol, Src Port: 12414, Dst Port: 502, Seq: 841, Ack: 1
▶ Modbus/TCP
▼ Modbus
    .000 0110 = Function Code: Write Single Register (6)
    Reference Number: 0
    Data: 0bb8

0000  00 0c 29 6d a1 06 00 0c  29 cb 84 a0 08 00 45 00   ··)m···· )·····E·
0010  00 34 28 70 40 00 80 06  00 00 c0 a8 03 82 c0 a8   ·4(p@··· ········
0020  03 a4 30 7e 01 f6 02 07  67 7f b9 7d bb 44 50 18   ··0~···· g··}·DP·
0030  01 d3 88 9d 00 00 4d 00  00 00 00 06 01 06 00 00   ······M· ········
0040  0b b8                                              ··
```

flag{4d0000000006010600000bb8}

工控现场里异常的文件

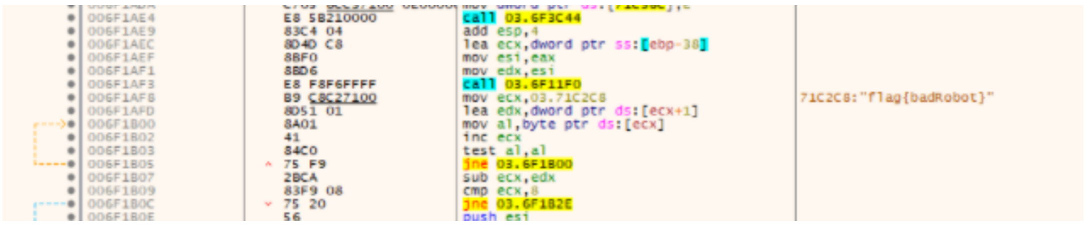
直接改EAX过掉反调试



```
53                  push ebx
8BDC                mov ebx,esp
83EC 08             sub esp,8
83E4 F0             and esp,FFFFFFF0
83C4 04             add esp,4
55                  push ebp
8B6B 04             mov ebp,dword ptr ds:[ebx+4]
896C24 04           mov dword ptr ss:[esp+4],ebp
8BEC                mov ebp,esp
81EC AC000000       sub esp,AC
A1 04B0AC00         mov eax,dword ptr ds:[ACB004]
33C5                xor eax,ebp
8945 FC             mov dword ptr ss:[ebp-4],eax
56                  push esi
64:A1 30000000      mov eax,dword ptr fs:[30]
0FB640 02           movzx eax,byte ptr ds:[eax+2]
85C0                test eax,eax
74 01               je 03.AA18A6
```

```
隐藏FPU
EAX     00000000
EBX     00BCF8FC
ECX     00000000
EDX     52BF2DB0
EBP     00BCF8F0
ESP     00BCF840
ESI     00ACBDF8      03.00ACBDF8
EDI     00DA6390      L"掴ú"==&"D:\\题目\\2021工业信

EIP     00AA18A1      03.00AA18A1

EFLAGS    00000200
ZF 0   PF 0   AF 0
OF 0   SF 0   DF 0
```

过掉检测以后直接在调试窗口里就能看到flag



```
006F1AE4   E8 5B210000    call 03.6F3C44
006F1AE9   83C4 04        add esp,4
006F1AEC   8D4D C8        lea ecx,dword ptr ss:[ebp-38]
006F1AEF   8BF0           mov esi,eax
006F1AF1   8BD6           mov edx,esi
006F1AF3   E8 F8F6FFFF    call 03.6F11F0
006F1AF8   B9 C8C27100    mov ecx,03.71C2C8                    71C2C8:"flag{badRobot}"
006F1AFD   8D51 01        lea edx,dword ptr ds:[ecx+1]
006F1B00   8A01           mov al,byte ptr ds:[ecx]
006F1B02   41             inc ecx
006F1B03   84C0           test al,al
006F1B05   75 F9          jne 03.6F1B00
006F1B07   2BCA           sub ecx,edx
006F1B09   83F9 08        cmp ecx,8
006F1B0C   75 20          jne 03.6F1B2E
006F1B0E   56             push esi
```

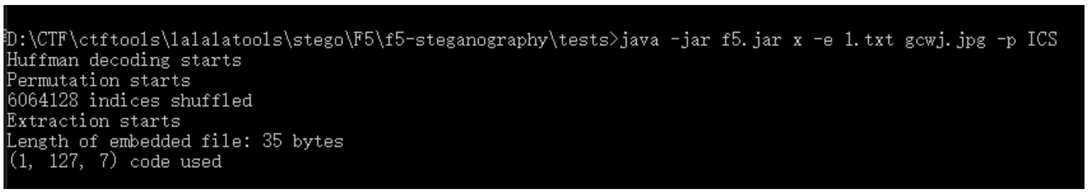
flag{badRobot}

隐藏的工程

F5隐写 https://github.com/matthewgao/F5-steganography

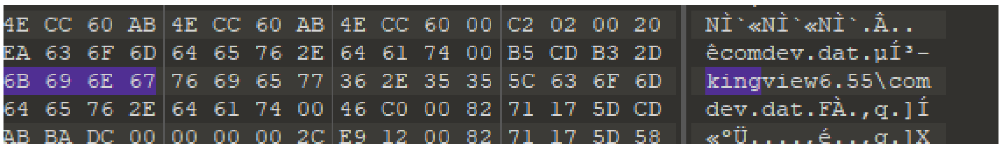
解题思路

密码：ICS

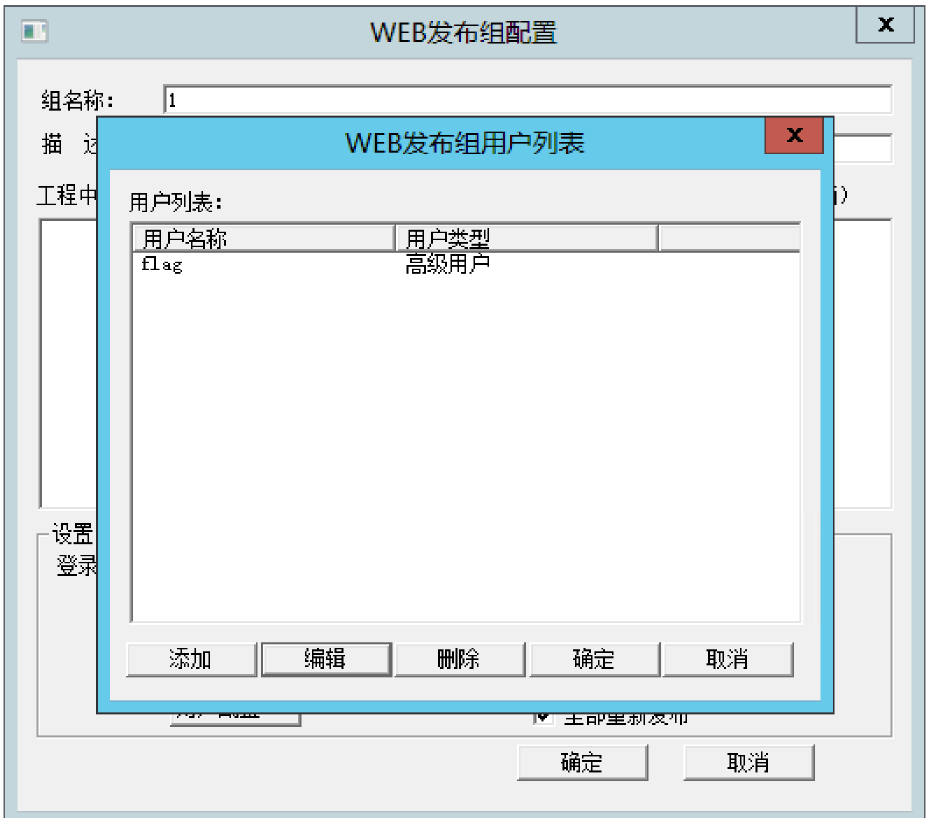


```
D:\CTF\ctftools\lalalatools\stego\F5\f5-steganography\tests>java -jar f5.jar x -e 1.txt gcwj.jpg -p ICS
Huffman decoding starts
Permutation starts
6064128 indices shuffled
Extraction starts
Length of embedded file: 35 bytes
(1, 127, 7) code used
```

提取出来一个蓝凑云连接：https://wwr.lanzoui.com/iIMaiqcpaxg
是kingview 6.55工程文件



```
4E CC 60 AB 4E CC 60 AB 4E CC 60 00 C2 02 00 20   NÌ`«NÌ`«NÌ`.Â..
EA 63 6F 6D 64 65 76 2E 64 61 74 00 B5 CD B3 2D   êcomdev.dat.µÍ³-
6B 69 6E 67 76 69 65 77 36 2E 35 35 5C 63 6F 6D   kingview6.55\com
64 65 76 2E 64 61 74 00 46 C0 00 82 71 17 5D CD   dev.dat.FÀ.,q.]Í
AB BA DC 00 00 00 00 2C E9 12 00 82 71 17 5D 58   «ºÜ....,é...q.]X
```



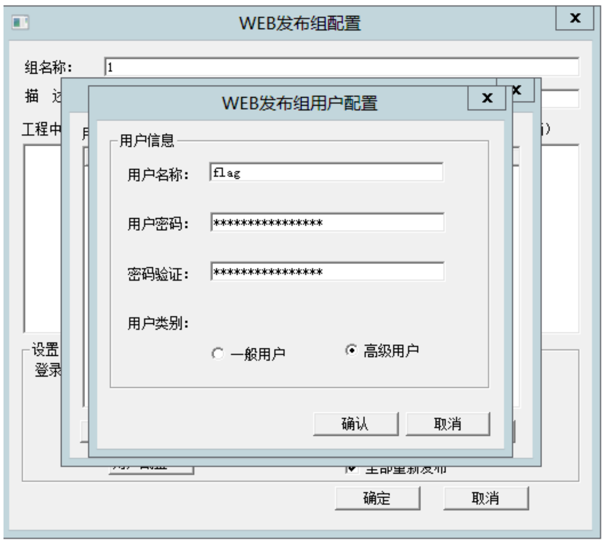
WEB发布组配置

组名称： 1

WEB发布组用户列表

用户列表：

用户名称	用户类型
flag	高级用户



WEB发布组用户配置

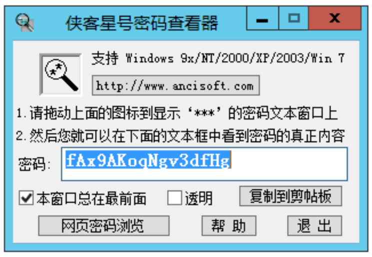
用户信息

用户名称： flag

用户密码： ****************

密码验证： ****************

用户类别： ( ) 一般用户 [✓ 高级用户]



侠客星号密码查看器

支持 Windows 9x/NT/2000/XP/2003/Win 7

http://www.ancisoft.com

1.请拖动上面的图标到显示 '***' 的密码文本窗口上

2.然后您就可以在下面的文本框中看到密码的真正内容

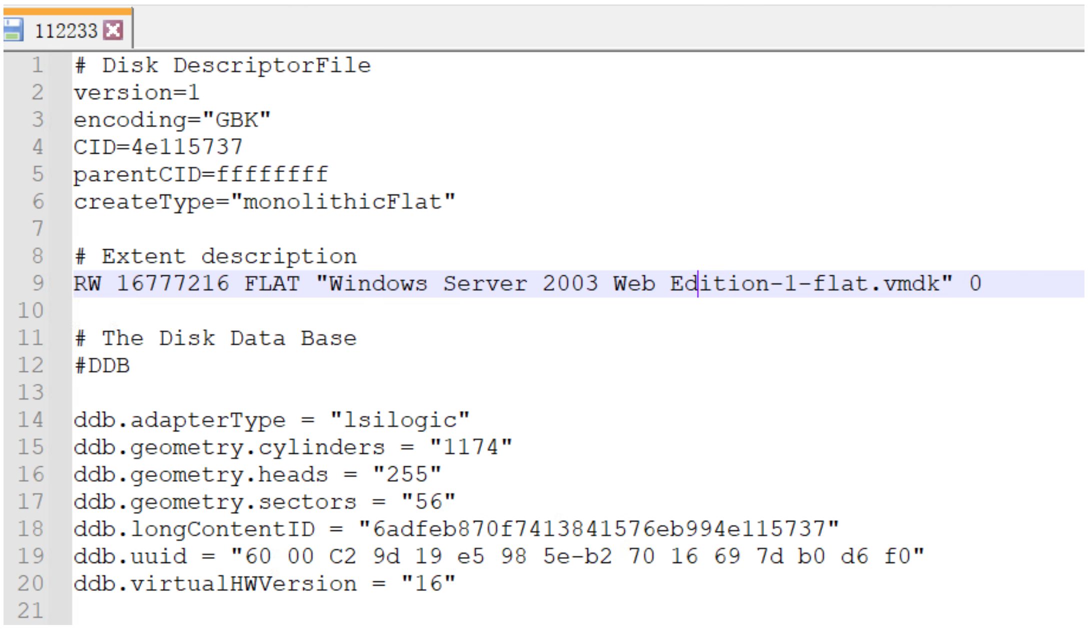
密码: fAx9AKoqNgv3dfHg

[✓] 本窗口总在最前面 [ ] 透明

flag{fAx9AKoqNgv3dfHg}

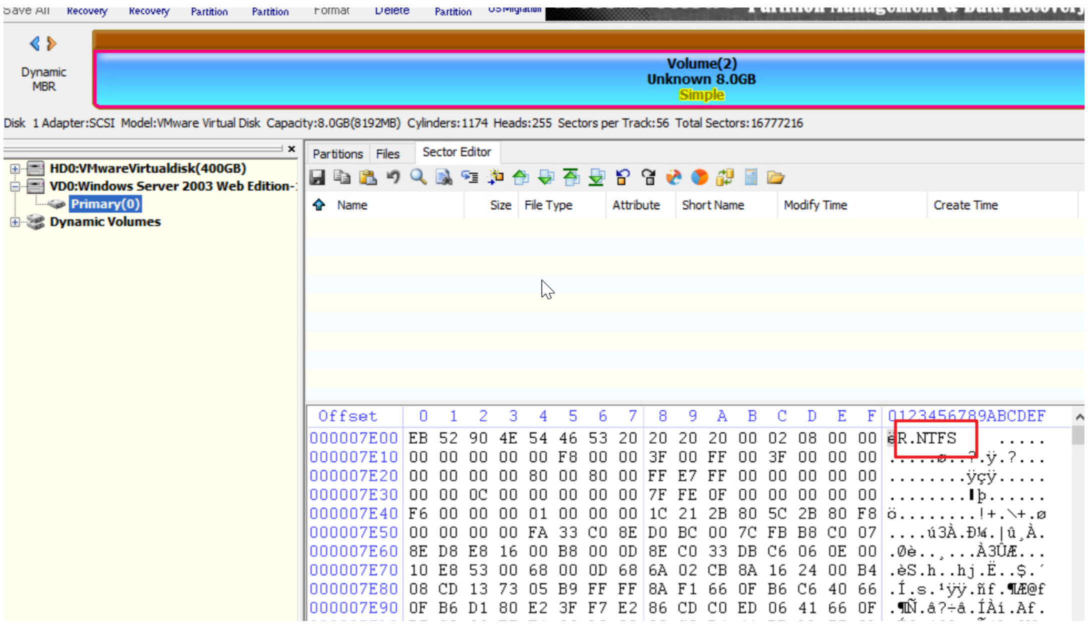
工控安全异常取证分析

解压出来两个文件，一个1122，文件挺大，另一个是 112233，1kb
一看就知道是个vmdk。



```
# Disk DescriptorFile
version=1
encoding="GBK"
CID=4e115737
parentCID=ffffffff
createType="monolithicFlat"

# Extent description
RW 16777216 FLAT "Windows Server 2003 Web Edition-1-flat.vmdk" 0

# The Disk Data Base
#DDB

ddb.adapterType = "lsilogic"
ddb.geometry.cylinders = "1174"
ddb.geometry.heads = "255"
ddb.geometry.sectors = "56"
ddb.longContentID = "6adfeb870f7413841576eb994e115737"
ddb.uuid = "60 00 C2 9d 19 e5 98 5e-b2 70 16 69 7d b0 d6 f0"
ddb.virtualHWVersion = "16"
```

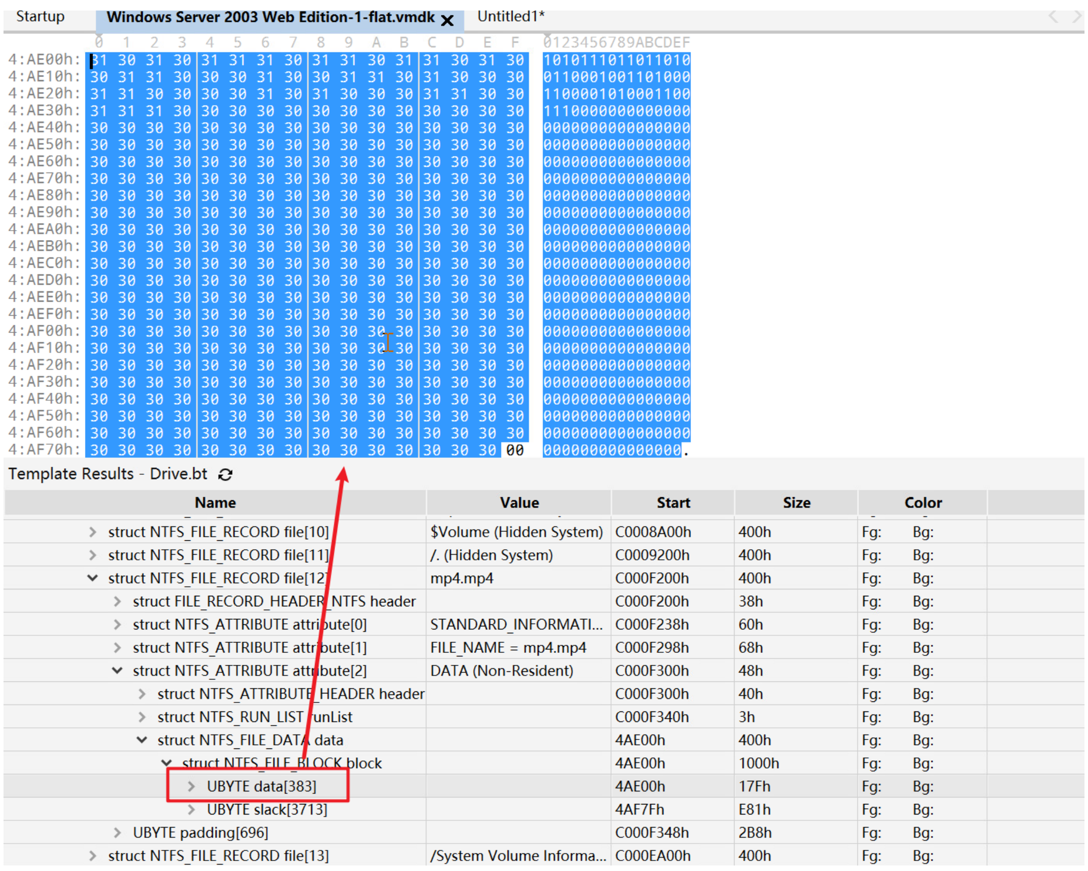
将 112233 改名为 Windows Server 2003 Web Edition-1.vmdk
将 1122 改名为 Windows Server 2003 Web Edition-1-flat.vmdk
用 DiskGenius 打开，应该是个NTFS分区FDD，没有分区表的那种。



Disk 1 Adapter:SCSI Model:VMware Virtual Disk Capacity:8.0GB(8192MB) Cylinders:1174 Heads:255 Sectors per Track:56 Total Sectors:16777216

```
Offset     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
000007E00 EB 52 90 4E 54 46 53 20 20 20 20 00 02 08 00 00 ëR.NTFS    .....
000007E10 00 00 00 00 00 F8 00 00 3F 00 FF 00 3F 00 00 00 .....ø..?.ÿ.?...
000007E20 00 00 00 00 80 00 80 00 FF E7 FF 00 00 00 00 00 ........ÿçÿ.....
000007E30 00 00 0C 00 00 00 00 00 7F FE 0F 00 00 00 00 00 ........þ......
000007E40 F6 00 00 00 01 00 00 00 1C 21 2B 80 5C 2B 80 F8 ö........!+.\+.ø
000007E50 00 00 00 00 FA 33 C0 8E D0 BC 00 7C FB B8 C0 07 ....ú3À.Ð¼.|û¸À.
000007E60 8E D8 E8 16 00 B8 00 0D 8E C0 33 DB C6 06 0E 00 .Øè..¸...À3ÛÆ...
000007E70 10 E8 53 00 68 00 0D 68 6A 02 CB 8A 16 24 00 B4 .èS.h..hj.Ë..$.´
000007E80 08 CD 13 73 05 B9 FF FF 8A F1 66 0F B6 C6 40 66 .Í.s.¹ÿÿ.ñf.¶Æ@f
000007E90 0F B6 D1 80 E2 3F F7 E2 86 CD C0 ED 06 41 66 0F .¶Ñ.â?÷â.ÍÀí.Af.
```

从名为 mp4.mp4 的文件中提取出一段 ascii 字符串。



```
Startup    Windows Server 2003 Web Edition-1-flat.vmdk    Untitled1*

           0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
4:AE00h:  31 30 31 30 31 31 31 30 31 31 30 31 31 30 31 30   1010111011011010
4:AE10h:  30 31 31 30 30 30 31 30 30 31 31 30 31 30 30 30   0110001001101000
4:AE20h:  31 31 30 30 30 30 31 30 31 30 30 30 31 31 30 30   1100001010001100
4:AE30h:  31 31 31 30 30 30 30 30 30 30 30 30 30 30 30 30   1110000000000000
4:AE40h:  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30   0000000000000000
4:AE50h:  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30   0000000000000000
4:AE60h:  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30   0000000000000000
4:AE70h:  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30   0000000000000000
4:AE80h:  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30   0000000000000000
4:AE90h:  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30   0000000000000000
4:AEA0h:  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30   0000000000000000
4:AEB0h:  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30   0000000000000000
4:AEC0h:  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30   0000000000000000
4:AED0h:  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30   0000000000000000
4:AEE0h:  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30   0000000000000000
4:AEF0h:  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30   0000000000000000
4:AF00h:  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30   0000000000000000
4:AF10h:  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30   0000000000000000
4:AF20h:  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30   0000000000000000
4:AF30h:  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30   0000000000000000
4:AF40h:  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30   0000000000000000
4:AF50h:  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30   0000000000000000
4:AF60h:  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30   0000000000000000
4:AF70h:  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 00   000000000000000.
```

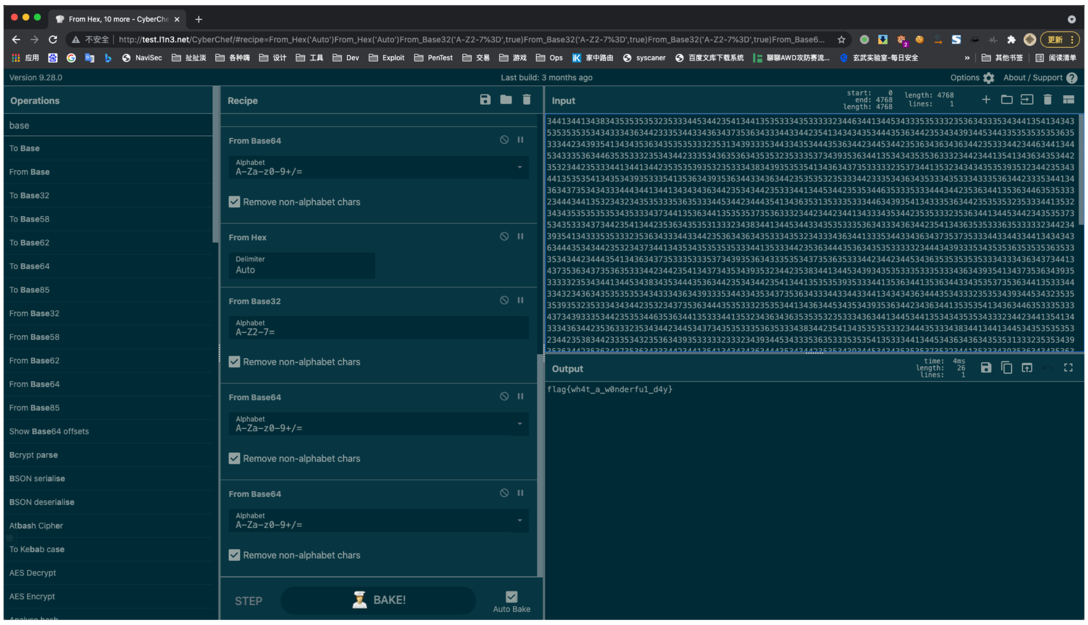
Template Results - Drive.bt

Name	Value	Start	Size	Color	
struct NTFS_FILE_RECORD file[10]	$Volume (Hidden System)	C0008A00h	400h	Fg:	Bg:
struct NTFS_FILE_RECORD file[11]	/. (Hidden System)	C0009200h	400h	Fg:	Bg:
struct NTFS_FILE_RECORD file[12]	mp4.mp4	C000F200h	400h	Fg:	Bg:
struct FILE_RECORD_HEADER NTFS header		C000F200h	38h	Fg:	Bg:
struct NTFS_ATTRIBUTE attribute[0]	STANDARD_INFORMATI...	C000F238h	60h	Fg:	Bg:
struct NTFS_ATTRIBUTE attribute[1]	FILE_NAME = mp4.mp4	C000F298h	68h	Fg:	Bg:
struct NTFS_ATTRIBUTE attribute[2]	DATA (Non-Resident)	C000F300h	48h	Fg:	Bg:
struct NTFS_ATTRIBUTE_HEADER header		C000F300h	40h	Fg:	Bg:
struct NTFS_RUN_LIST runList		C000F340h	3h	Fg:	Bg:
struct NTFS_FILE_DATA data		4AE00h	400h	Fg:	Bg:
struct NTFS_FILE_BLOCK block		4AE00h	1000h	Fg:	Bg:
UBYTE data[383]		4AE00h	17Fh	Fg:	Bg:
UBYTE slack[3713]		4AF7Fh	E81h	Fg:	Bg:
UBYTE padding[696]		C000F348h	2B8h	Fg:	Bg:
struct NTFS_FILE_RECORD file[13]	/System Volume Informa...	C000EA00h	400h	Fg:	Bg:

提取出字符串

```
344134413438343535353532353334453442354134413535333435333332344634413445343
3335353332353634333534344135413434353535353534343334363442333534433436343
7353634333443344235413434343534443536344235343439344534433535353535363533
3442343935413434353634353535333235313439333534433435344435363442344534442353
6343634363442353334423446344134453433353634463535333235343442333534363536
3435353235333537343935363441353434353536333234423441354134363435344235323442
3533344134413442353535393532353334383439353535413436343735333332353734413
5323434343535393532344235343441353535413435343935333541353634393536344334
3634423535353235333442333534363435333435333433353634423335344134363437353434
3333444344134413434343436344235343442353334413445344235353446353333533344434
4235363441353634463535333234443441353234323435353333536353334453442344435413
4363531353335333446343935413433353634423535353235333441353234343535353535
3435333437344135363441353535373536333234423442344134333435344235353332353634
4134453442343535373534353334373442354134423536343535313332343834413445344
3343535333536343334363442354134363535333635333332344234393541343335353332
3536343333443344235363436343533343532343333436344133353443343634373537353334
4334433441343434363444353434423532343734413435343535353533344135333442353634
4435363435353333323444343933353435353635353536353335343442344435413436343
7353335333537343935363433353534373536353334423442344534363535353535353334
3334363437344134373536343735363533344234423541343734353439353234423538344134
4534393435353335333533343634393541343735363439353333323534344134453438343534
4435363442353434423541344135353539353334413536344135363443343535373536344
1353334433434323436343535353534343334363439333534433435343735363433344334
4334413434343634443534333235353439344534323535353935323533343434423532343735
3634443535333235353441343634453435343935363442343634413535354134363446353
3353334373439333534423535344635363441353334413532343634363535353235333436
3441
```

```
34453441353434353534333234423441354134333436344235363332353434423445343734
```


Recipe

From Base64
Alphabet A-Za-z0-9+/=
Remove non-alphabet chars

From Hex
Delimiter Auto

From Base32
Alphabet A-Z2-7=
Remove non-alphabet chars

From Base64
Alphabet A-Za-z0-9+/=
Remove non-alphabet chars

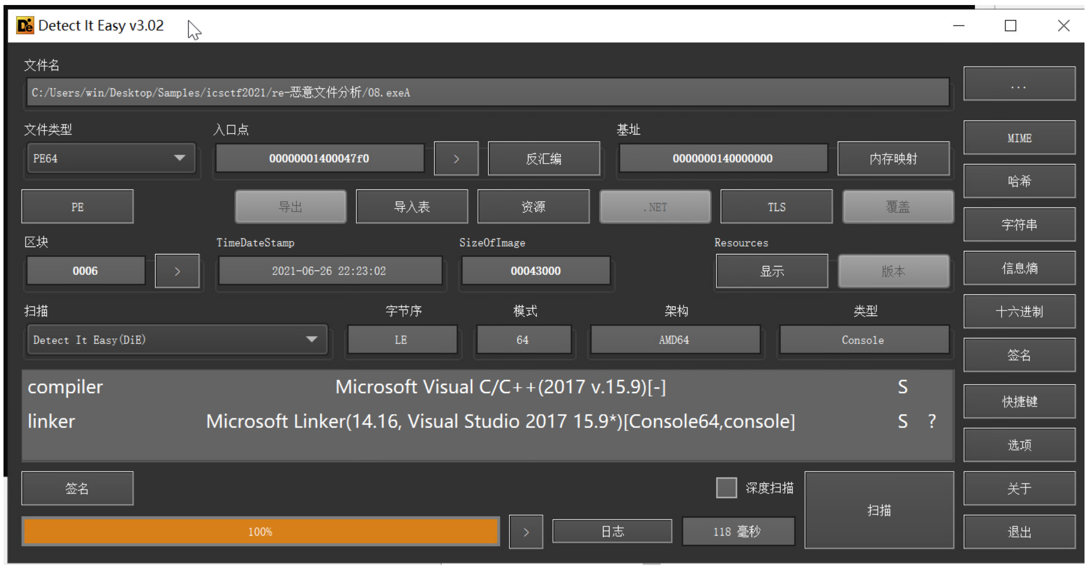
From Base64
Alphabet A-Za-z0-9+/=
Remove non-alphabet chars

Output

```
flag{wh4t_a_w0nderful_d4y}
```

flag{wh4t_a_w0nderfu1_d4y}

恶意文件分析


Detect It Easy v3.02

文件名: C:/Users/win/Desktop/Samples/icsctf2021/re-恶意文件分析/08.exeA

文件类型: PE64　入口点: 00000001400047f0　基址: 0000000140000000

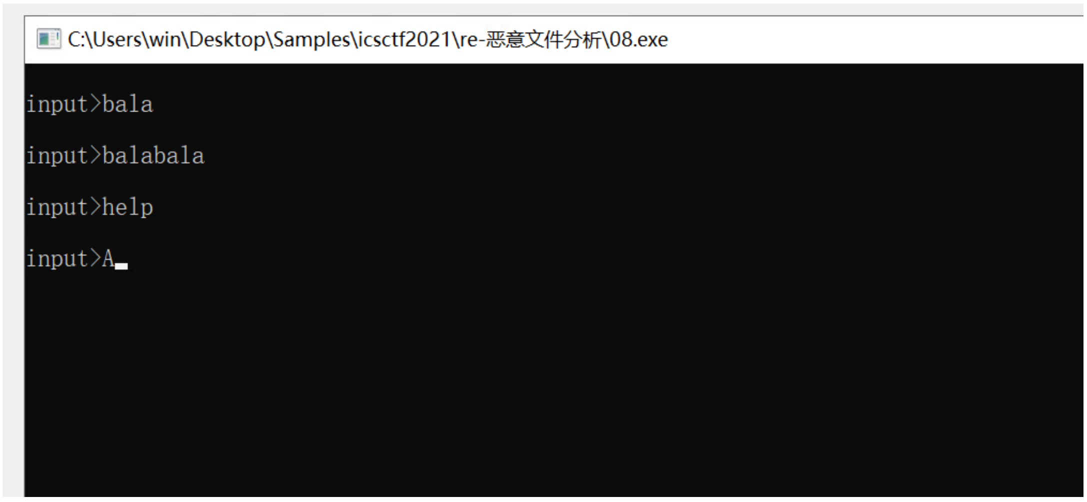
区块: 0006　TimeDateStamp: 2021-06-26 22:23:02　SizeOfImage: 00043000

扫描: Detect It Easy(DiE)　字节序: LE　模式: 64　架构: AMD64　类型: Console

```
compiler    Microsoft Visual C/C++(2017 v.15.9)[-]                                  S
linker      Microsoft Linker(14.16, Visual Studio 2017 15.9*)[Console64,console]    S ?
```

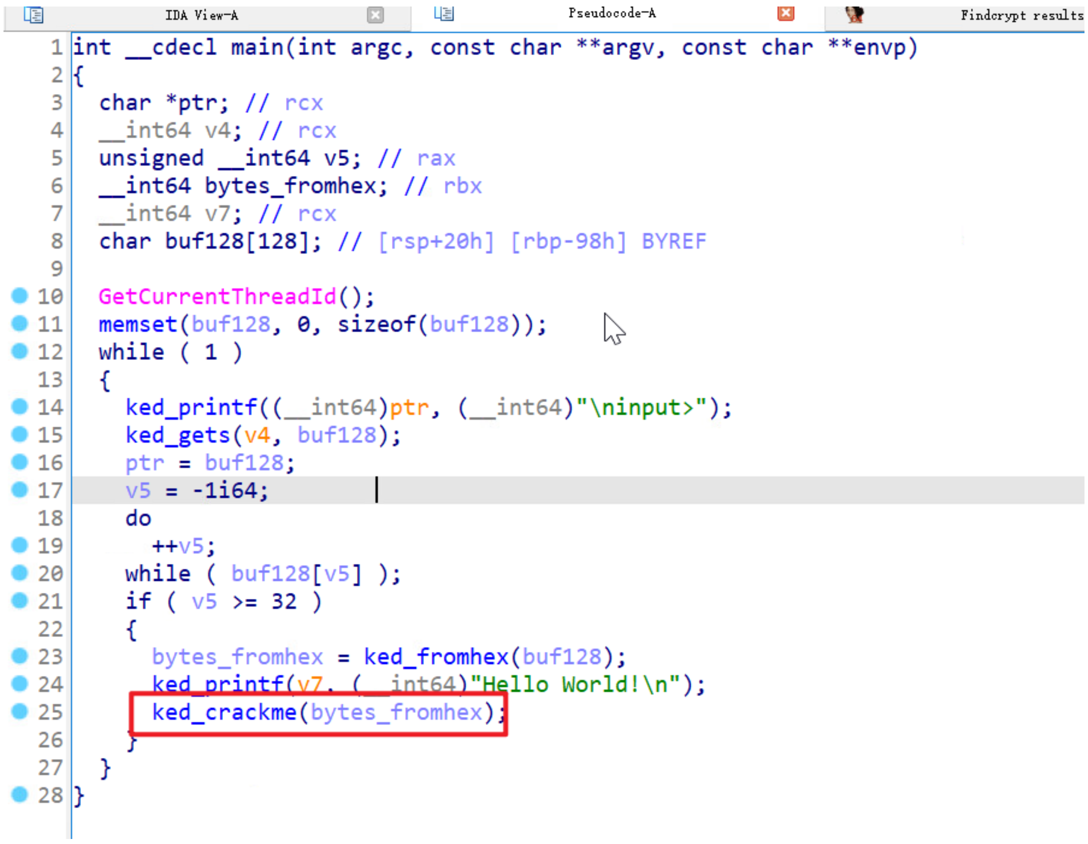
100%　118 毫秒

无壳。进去类似一个shell


C:\Users\win\Desktop\Samples\icsctf2021\re-恶意文件分析\08.exe

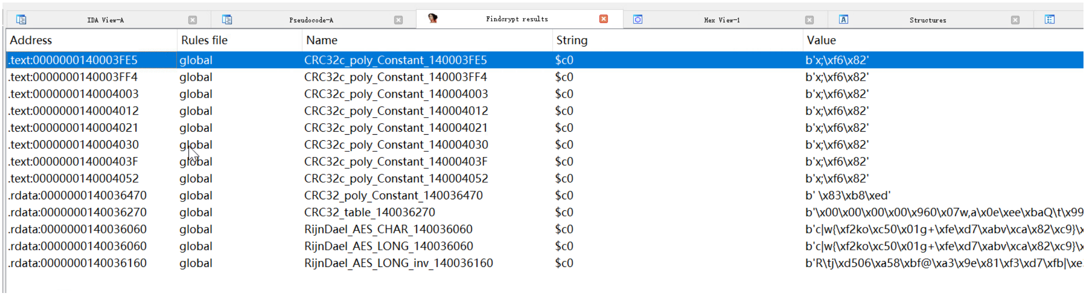
```
input>bala
input>balabala
input>help
input>A
```

程序流程，输入32长度的哈希字符串，bytes.fromhex然后运行关键 crackme 函数。



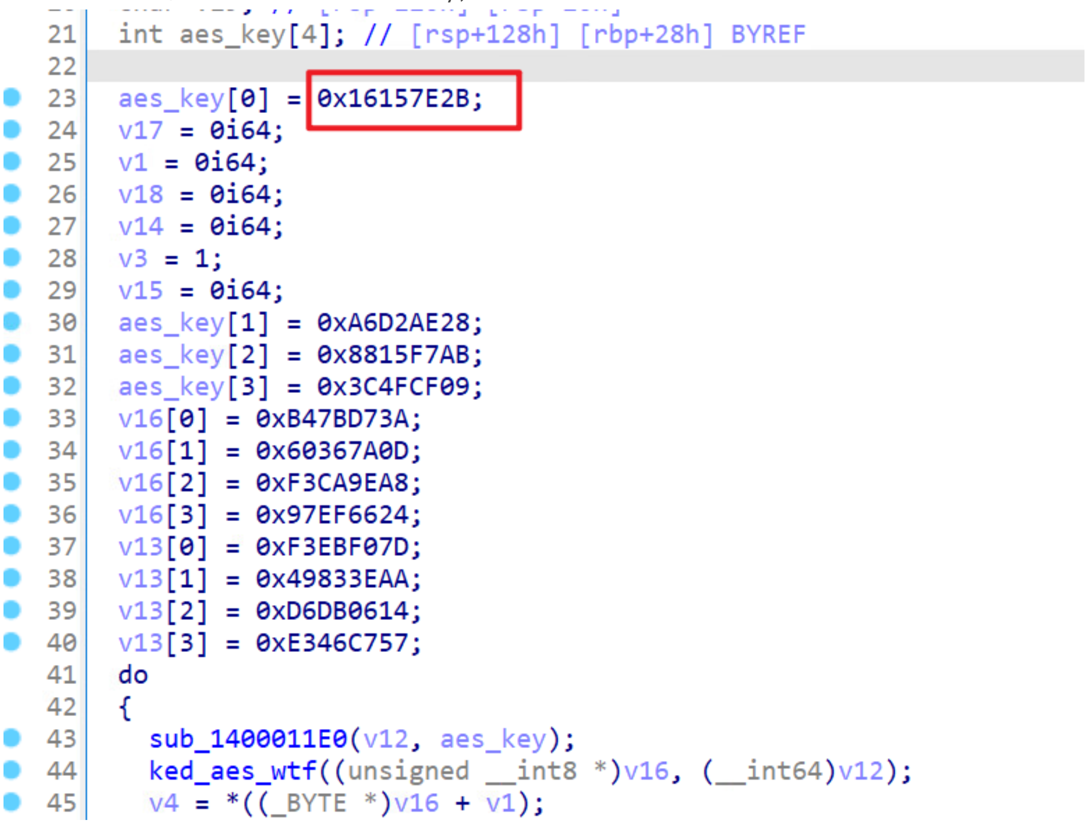
```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char *ptr; // rcx
  __int64 v4; // rcx
  unsigned __int64 v5; // rax
  __int64 bytes_fromhex; // rbx
  __int64 v7; // rcx
  char buf128[128]; // [rsp+20h] [rbp-98h] BYREF

  GetCurrentThreadId();
  memset(buf128, 0, sizeof(buf128));
  while ( 1 )
  {
    ked_printf((__int64)ptr, (__int64)"\ninput>");
    ked_gets(v4, buf128);
    ptr = buf128;
    v5 = -1i64;
    do
      ++v5;
    while ( buf128[v5] );
    if ( v5 >= 32 )
    {
      bytes_fromhex = ked_fromhex(buf128);
      ked_printf(v7, (__int64)"Hello World!\n");
      ked_crackme(bytes_fromhex);
    }
  }
}
```

findcrypt发现文件中有CRC32常量。



Address	Rules file	Name	String	Value
.text:0000000140003FE5	global	CRC32c_poly_Constant_140003FE5	$c0	b'x;\xf6\x82'
.text:0000000140003FF4	global	CRC32c_poly_Constant_140003FF4	$c0	b'x;\xf6\x82'
.text:0000000140004003	global	CRC32c_poly_Constant_140004003	$c0	b'x;\xf6\x82'
.text:0000000140004012	global	CRC32c_poly_Constant_140004012	$c0	b'x;\xf6\x82'
.text:0000000140004021	global	CRC32c_poly_Constant_140004021	$c0	b'x;\xf6\x82'
.text:0000000140004030	global	CRC32c_poly_Constant_140004030	$c0	b'x;\xf6\x82'
.text:000000014000403F	global	CRC32c_poly_Constant_14000403F	$c0	b'x;\xf6\x82'
.text:0000000140004052	global	CRC32c_poly_Constant_140004052	$c0	b'x;\xf6\x82'
.rdata:0000000140036470	global	CRC32_poly_Constant_140036470	$c0	b' \x83\xb8\xed'
.rdata:0000000140036270	global	CRC32_table_140036270	$c0	b'\x00\x00\x00\x00\x960\x07w,a\x0e\xee\xbaQ\t\x99
.rdata:0000000140036060	global	RijnDael_AES_CHAR_140036060	$c0	b'c|w{\xf2ko\xc50\x01g+\xfe\xd7\xabv\xca\x82\xc9}\x
.rdata:0000000140036060	global	RijnDael_AES_LONG_140036060	$c0	b'c|w{\xf2ko\xc50\x01g+\xfe\xd7\xabv\xca\x82\xc9}\x
.rdata:0000000140036160	global	RijnDael_AES_LONG_inv_140036160	$c0	b'R\tj\xd506\xa58\xbf@\xa3\x9e\x81\xf3\xd7\xfb|\xe

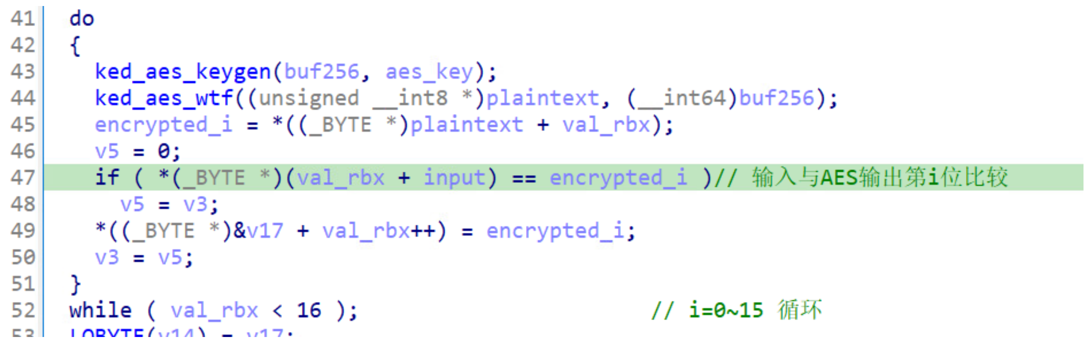
crackme函数中包含了一个硬编码的key，



```
  int aes_key[4]; // [rsp+128h] [rbp+28h] BYREF

  aes_key[0] = 0x16157E2B;
  v17 = 0i64;
  v1 = 0i64;
  v18 = 0i64;
  v14 = 0i64;
  v3 = 1;
  v15 = 0i64;
  aes_key[1] = 0xA6D2AE28;
  aes_key[2] = 0x8815F7AB;
  aes_key[3] = 0x3C4FCF09;
  v16[0] = 0xB47BD73A;
  v16[1] = 0x60367A0D;
  v16[2] = 0xF3CA9EA8;
  v16[3] = 0x97EF6624;
  v13[0] = 0xF3EBF07D;
  v13[1] = 0x49833EAA;
  v13[2] = 0xD6DB0614;
  v13[3] = 0xE346C757;
  do
  {
    sub_1400011E0(v12, aes_key);
    ked_aes_wtf((unsigned __int8 *)v16, (__int64)v12);
    v4 = *((_BYTE *)v16 + v1);
```

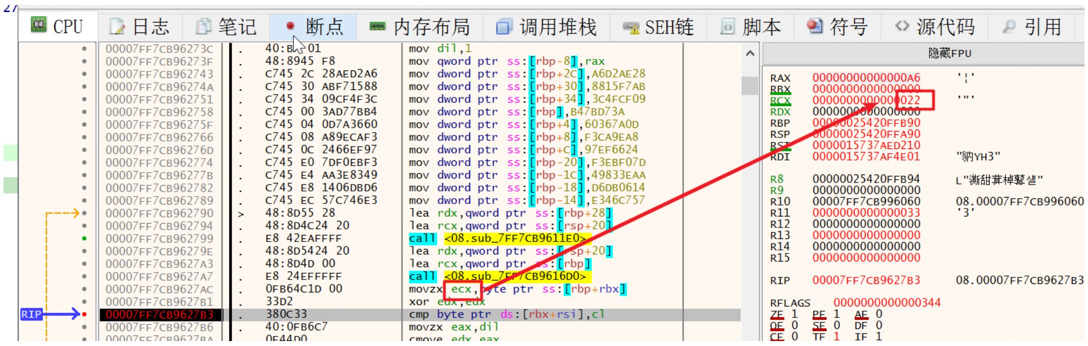
从开源项目中可以检索到。
https://github.com/TurboPack/LockBox3/blob/master/run/ciphers/uTPLb_AES.pas
https://chromium.googlesource.com/chromiumos/platform/ec/+/refs/heads/stabilize-7797.B/test/tpm_test/crypto_test.xml

输入与硬编码的Key、加密结果作比较。16轮。



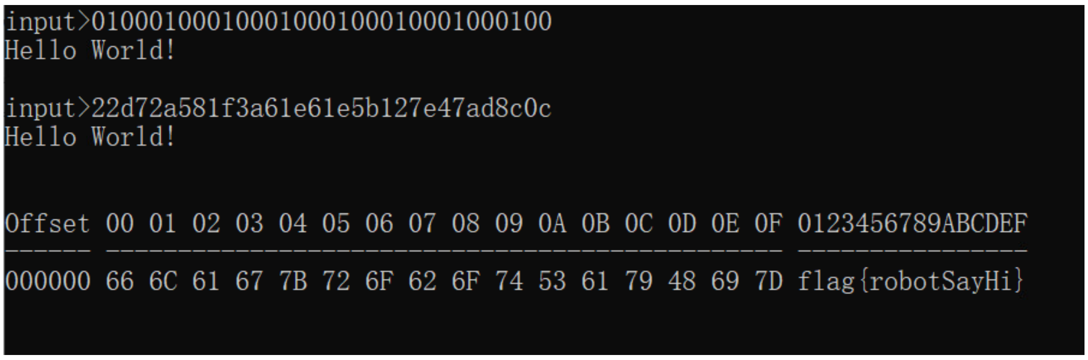
```
do
{
  ked_aes_keygen(buf256, aes_key);
  ked_aes_wtf((unsigned __int8 *)plaintext, (__int64)buf256);
  encrypted_i = *((_BYTE *)plaintext + val_rbx);
  v5 = 0;
  if ( *(_BYTE *)(val_rbx + input) == encrypted_i )// 输入与AES输出第i位比较
    v5 = v3;
  *((_BYTE *)&v17 + val_rbx++) = encrypted_i;
  v3 = v5;
}
while ( val_rbx < 16 );                    // i=0~15 循环
```

循环16次看ECX值即可dump出预期输入。拼起来可以得到：
22d72a581f3a61e61e5b127e47ad8c0c



```
00007FF7CB96273C | 40:B_01           | mov dil,1
00007FF7CB96273F | 48:8945 F8         | mov qword ptr ss:[rbp-8],rax
00007FF7CB962743 | C745 2C 28AED2A6   | mov dword ptr ss:[rbp+2C],A6D2AE28
00007FF7CB96274A | C745 30 ABF71588   | mov dword ptr ss:[rbp+30],8815F7AB
00007FF7CB962751 | C745 34 09CF4F3C   | mov dword ptr ss:[rbp+34],3C4FCF09
00007FF7CB962758 | C745 00 3AD77BB4   | mov dword ptr ss:[rbp],B47BD73A
00007FF7CB96275F | C745 04 0D7A3660   | mov dword ptr ss:[rbp+4],60367A0D
00007FF7CB962766 | C745 08 A89ECAF3   | mov dword ptr ss:[rbp+8],F3CA9EA8
00007FF7CB96276D | C745 0C 2466EF97   | mov dword ptr ss:[rbp+C],97EF6624
00007FF7CB962774 | C745 E0 7DF0EBF3   | mov dword ptr ss:[rbp-20],F3EBF07D
00007FF7CB96277B | C745 E4 AA3E8349   | mov dword ptr ss:[rbp-1C],49833EAA
00007FF7CB962782 | C745 E8 1406DBD6   | mov dword ptr ss:[rbp-18],D6DB0614
00007FF7CB962789 | C745 EC 57C746E3   | mov dword ptr ss:[rbp-14],E346C757
00007FF7CB962790 | 48:8D55 28         | lea rdx,qword ptr ss:[rbp+28]
00007FF7CB962794 | 48:8D4C24 20       | lea rcx,qword ptr ss:[rsp+20]
00007FF7CB962799 | E8 42EAFFFF        | call <08.sub_7FF7CB9611E0>
00007FF7CB96279E | 48:8D5424 20       | lea rdx,qword ptr ss:[rsp+20]
00007FF7CB9627A3 | 48:8D4D 00         | lea rcx,qword ptr ss:[rbp]
00007FF7CB9627A7 | E8 24EFFFFF        | call <08.sub_7FF7CB9616D0>
00007FF7CB9627AC | 0FB64C1D 00        | movzx ecx,byte ptr ss:[rbp+rbx]
00007FF7CB9627B1 | 33D2               | xor edx,edx
00007FF7CB9627B3 | 380C33             | cmp byte ptr ds:[rbx+rsi],cl
00007FF7CB9627B6 | 40:0FB6C7          | movzx eax,dil
00007FF7CB9627BA | 0F44D0             | cmove edx,eax

RAX     00000000000000A6      '¦'
RBX     0000000000000000
RCX     0000000000000022      '"'
RDX     0000000000000000
RBP     00000025420FFB90
RSP     00000025420FFA90
RSI     0000015737AED210
RDI     0000015737AF4E01      "豹YH3"

R8      00000025420FFB94      L"澌甜萁棹鼞섵"
R9      0000000000000000
R10     00007FF7CB996060      08.00007FF7CB996060
R11     0000000000000033      '3'
R12     0000000000000000
R13     0000000000000000
R14     0000000000000000
R15     0000000000000000

RIP     00007FF7CB9627B3      08.00007FF7CB9627B3

RFLAGS  0000000000000344
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 1  IF 1
```

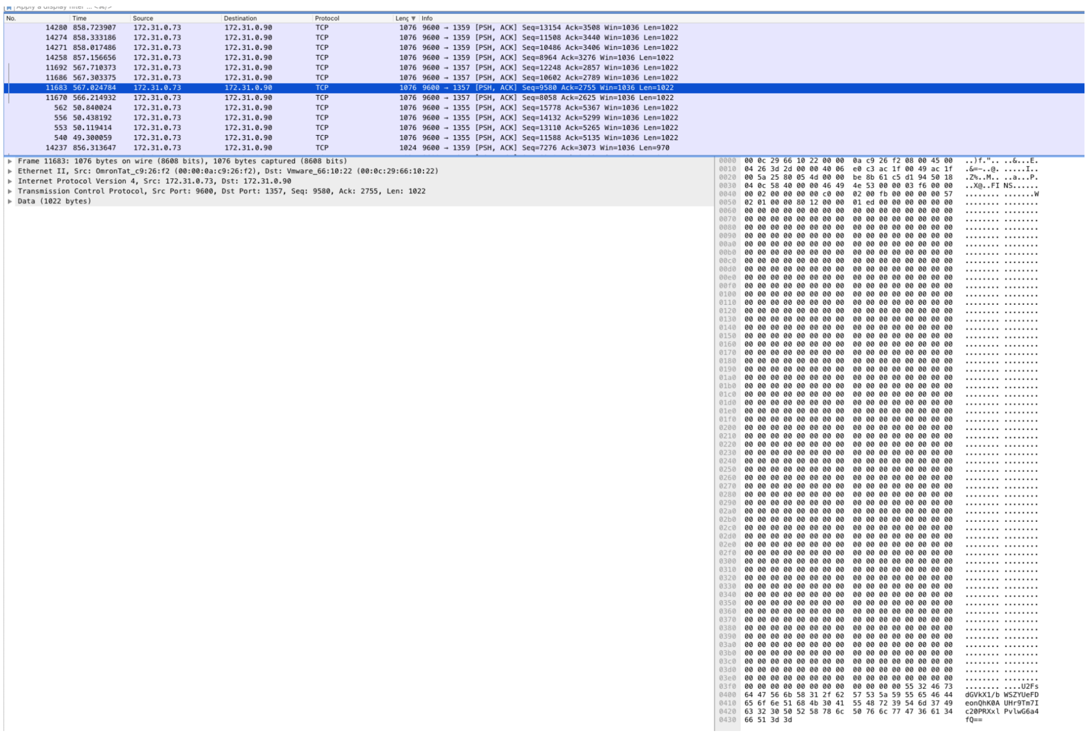
将这个值输入程序，得到flag



```
input>01000100010001000100010001000100
Hello World!

input>22d72a581f3a61e61e5b127e47ad8c0c
Hello World!


Offset 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 0123456789ABCDEF
------ ----------------------------------------------- ----------------
000000 66 6C 61 67 7B 72 6F 62 6F 74 53 61 79 48 69 7D flag{robotSayHi}
```

Fins协议通讯

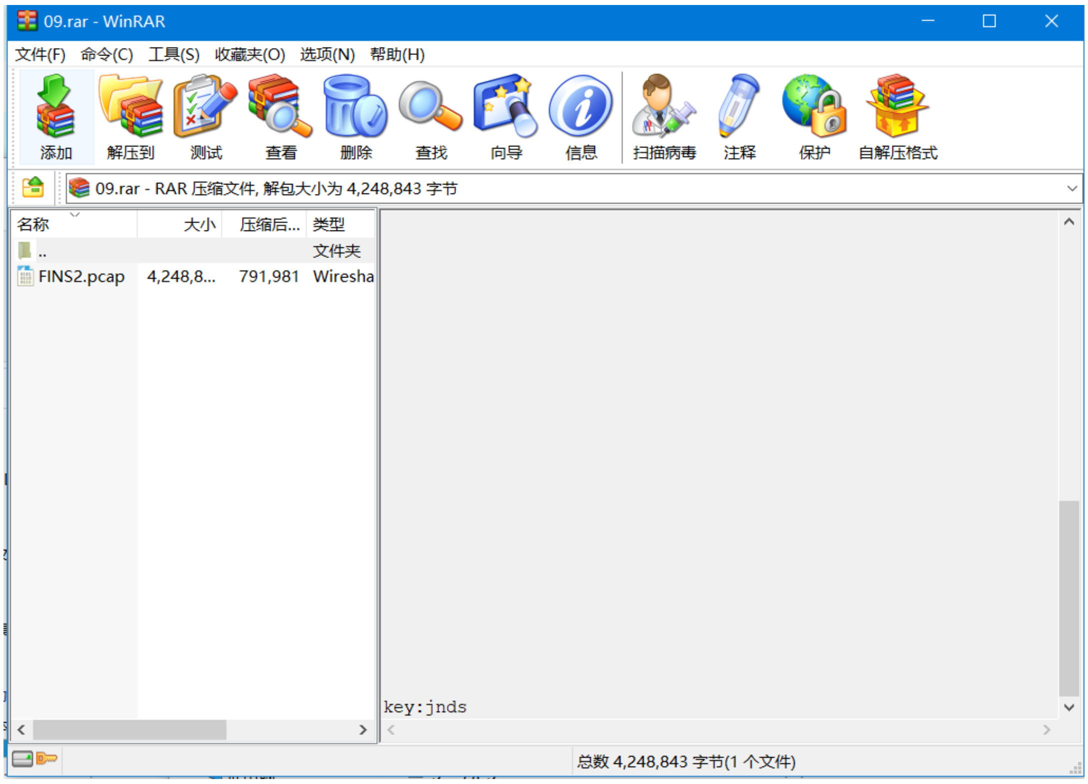
11683 号数据包



```
No.        Time        Source        Destination   Protocol  Leng▼ Info
     14280 858.723907  172.31.0.73   172.31.0.90   TCP       1076 9600 → 1359 [PSH, ACK] Seq=13154 Ack=3508 Win=1036 Len=1022
     14274 858.333186  172.31.0.73   172.31.0.90   TCP       1076 9600 → 1359 [PSH, ACK] Seq=11508 Ack=3440 Win=1036 Len=1022
     14271 858.017486  172.31.0.73   172.31.0.90   TCP       1076 9600 → 1359 [PSH, ACK] Seq=10486 Ack=3406 Win=1036 Len=1022
     14258 857.156656  172.31.0.73   172.31.0.90   TCP       1076 9600 → 1359 [PSH, ACK] Seq=8964 Ack=3276 Win=1036 Len=1022
     11692 567.710373  172.31.0.73   172.31.0.90   TCP       1076 9600 → 1357 [PSH, ACK] Seq=12248 Ack=2857 Win=1036 Len=1022
     11686 567.303375  172.31.0.73   172.31.0.90   TCP       1076 9600 → 1357 [PSH, ACK] Seq=10602 Ack=2789 Win=1036 Len=1022
     11683 567.024784  172.31.0.73   172.31.0.90   TCP       1076 9600 → 1357 [PSH, ACK] Seq=9580 Ack=2755 Win=1036 Len=1022
     11670 566.214932  172.31.0.73   172.31.0.90   TCP       1076 9600 → 1357 [PSH, ACK] Seq=8058 Ack=2625 Win=1036 Len=1022
       562 50.840024   172.31.0.73   172.31.0.90   TCP       1076 9600 → 1355 [PSH, ACK] Seq=15778 Ack=5367 Win=1036 Len=1022
       556 50.438192   172.31.0.73   172.31.0.90   TCP       1076 9600 → 1355 [PSH, ACK] Seq=14132 Ack=5299 Win=1036 Len=1022
       553 50.119414   172.31.0.73   172.31.0.90   TCP       1076 9600 → 1355 [PSH, ACK] Seq=13110 Ack=5265 Win=1036 Len=1022
       540 49.300059   172.31.0.73   172.31.0.90   TCP       1076 9600 → 1355 [PSH, ACK] Seq=11588 Ack=5135 Win=1036 Len=1022
     14237 856.313647  172.31.0.73   172.31.0.90   TCP       1024 9600 → 1359 [PSH, ACK] Seq=7276 Ack=3073 Win=1036 Len=970

▶ Frame 11683: 1076 bytes on wire (8608 bits), 1076 bytes captured (8608 bits)
▶ Ethernet II, Src: OmronTat_c9:26:f2 (00:00:0a:c9:26:f2), Dst: Vmware_66:10:22 (00:0c:29:66:10:22)
▶ Internet Protocol Version 4, Src: 172.31.0.73, Dst: 172.31.0.90
▶ Transmission Control Protocol, Src Port: 9600, Dst Port: 1357, Seq: 9580, Ack: 2755, Len: 1022
▶ Data (1022 bytes)
```

```
0000  00 0c 29 66 10 22 00 00  0a c9 26 f2 08 00 45 00   ..)f.".. ..&...E.
0010  04 26 3d 2d 00 00 40 06  e0 c3 ac 1f 00 49 ac 1f   .&=-..@. .....I..
0020  00 5a 25 80 05 4d 00 00  be 8b 61 c5 d1 94 50 18   .Z%..M.. ..a...P.
0030  04 0c 58 40 00 00 46 49  4e 53 00 00 03 f6 00 00   ..X@..FI NS......
0040  00 02 00 00 00 00 c0 00  02 00 fb 00 00 00 00 57   ........ .......W
0050  02 01 00 00 80 12 00 00  01 ed 00 00 00 00 00 00   ........ ........
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0100  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0110  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0120  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0130  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0140  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0160  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0170  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0180  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0190  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0200  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0210  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0220  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0230  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0240  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0250  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0260  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0270  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0280  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0290  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
02a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
02b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
02c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
02d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
02e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
02f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0300  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0310  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0320  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0330  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0340  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0350  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0360  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0370  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0380  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0390  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
03a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
03b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
03c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
03d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
03e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
03f0  00 00 00 00 00 00 00 00  00 00 00 00 55 32 46 73   ........ ....U2Fs
0400  64 47 56 6b 58 31 2f 62  57 53 5a 59 55 65 46 44   dGVkX1/b WSZYUeFD
0410  65 6f 6e 51 68 4b 30 41  55 48 72 39 54 6d 37 49   eonQhK0A UHr9Tm7I
0420  63 32 30 50 52 58 78 6c  50 76 6c 77 47 36 61 34   c20PRXxl PvlwG6a4
0430  66 51 3d 3d                                        fQ==
```

```
U2FsdGVkX1/bWSZYUeFDeonQhK0AUHr9Tm7Ic20PRXxlPvlwG6a4fQ==
```



09.rar - RAR 压缩文件, 解包大小为 4,248,843 字节

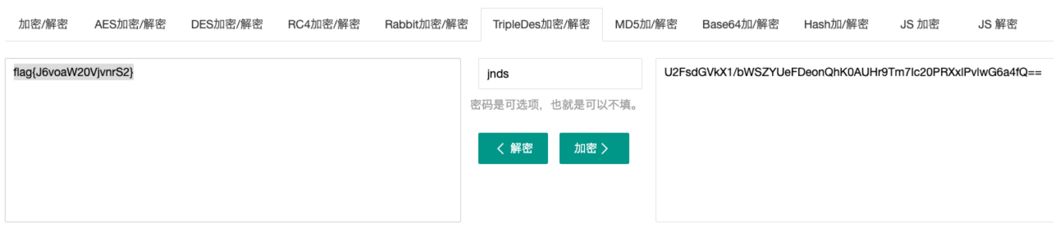
名称	大小	压缩后...	类型
..			文件夹
FINS2.pcap	4,248,8...	791,981	Wiresha

key:jnds

总数 4,248,843 字节(1 个文件)

3DES解密

https://www.sojson.com/encrypt_triple_des.html



flag{J6voaW20VjvnrS2}

jnds

密码是可选项，也就是可以不填。

U2FsdGVkX1/bWSZYUeFDeonQhK0AUHr9Tm7Ic20PRXxlPvlwG6a4fQ==