

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](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/mochu7777777/article/details/120553453>

版权



[CTF_MISC_Writeup](#) 专栏收录该内容

246 篇文章 46 订阅

订阅专栏

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

0300002402f080320100000003000e00050501120a10020001000083000000004000801。

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

工控流量分析

30
分值

未解答

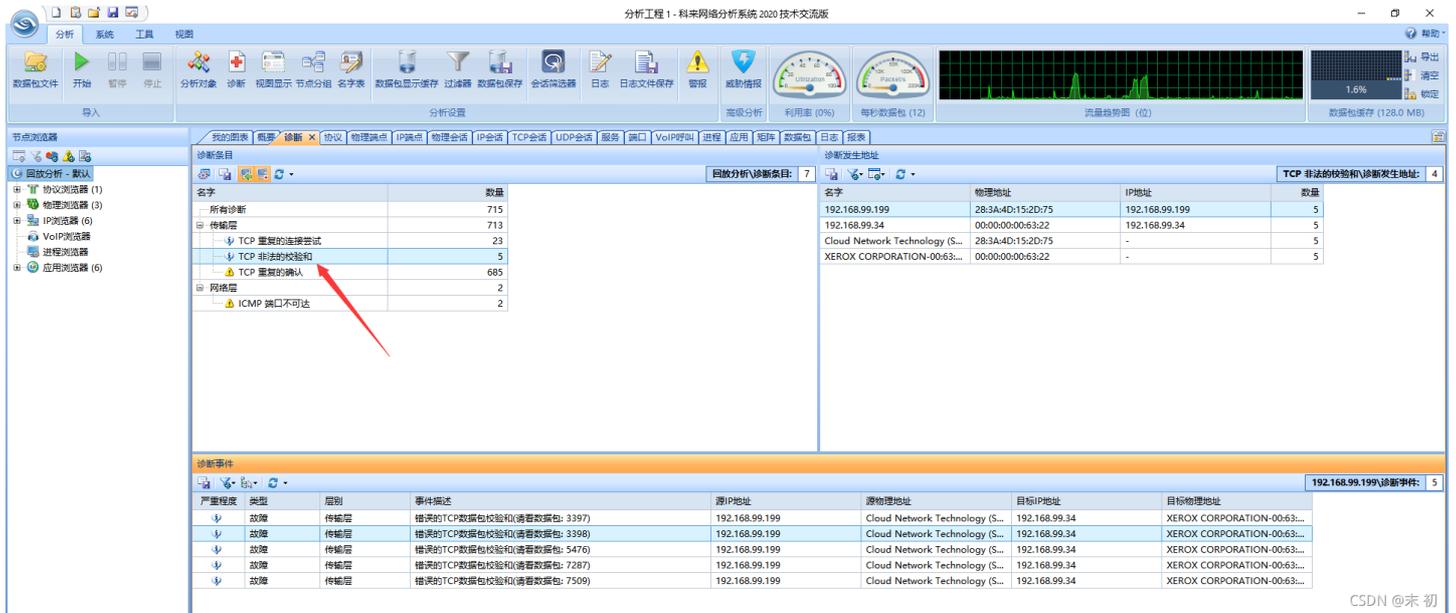
星火燎原队 Sword 赣州电信队

某企业车间PLC运行异常, 造成生产线无法正常运行。请您帮助改企业车间分析出PLC遭到异常的原因。flag格式为:flag{}

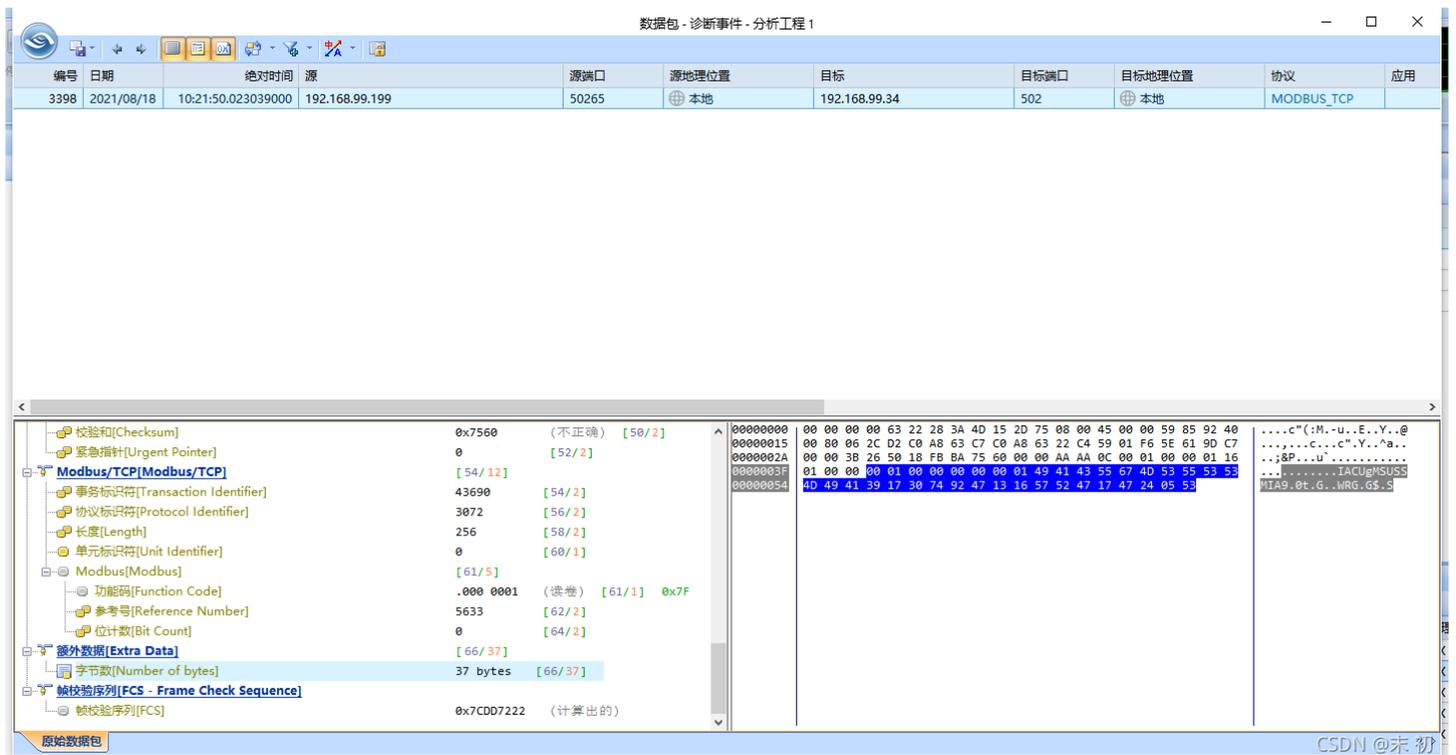
主下载地址

CSDN @末初

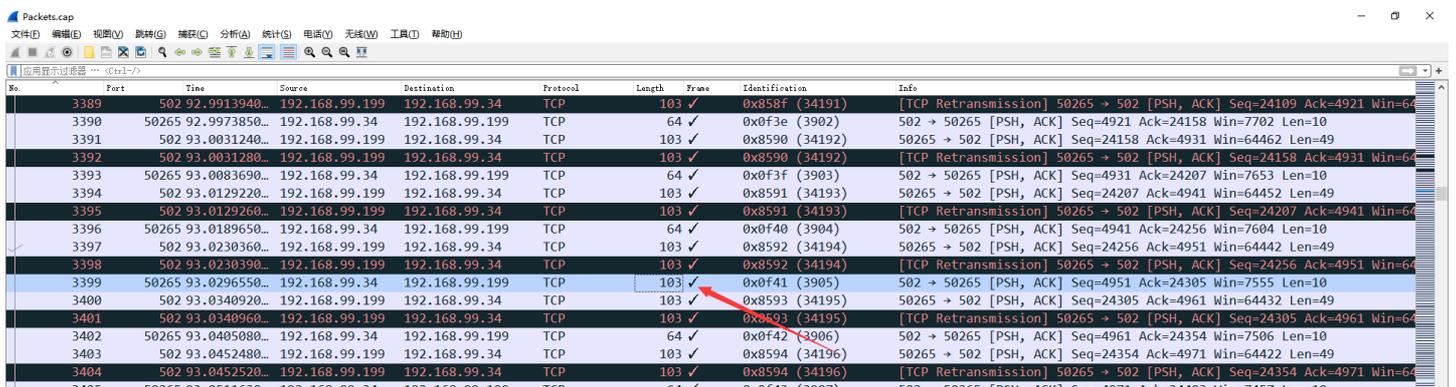
用科来诊断数据包时发现几个TCP非法校验的包



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



3405	50265	93.0557970...	192.168.99.34	192.168.99.199	TCP	64	✓	0x8595 (34197)	50265 → 50265 [PSH, ACK] Seq=4951 Ack=24403 Win=7457 Len=10
3406	502	93.0557970...	192.168.99.199	192.168.99.34	TCP	103	✓	0x8595 (34197)	50265 → 502 [PSH, ACK] Seq=24403 Ack=4981 Win=64412 Len=49

> Frame 3399: 103 bytes on wire (824 bits), 103 bytes captured (824 bits)
 > Ethernet II, Src: 00:00:00:00:63:22 (00:00:00:00:63:22), Dst: CloudNet_15:2d:75 (28:3a:4d:15:2d:75)
 > Internet Protocol Version 4, Src: 192.168.99.34, Dst: 192.168.99.199
 > Transmission Control Protocol, Src Port: 502, Dst Port: 50265, Seq: 4951, Ack: 24305, Len: 10
 > Data (10 bytes)
 Data: bbbb0200010000010000
 [Length: 10]

0000	28 3a 4d 15 2d 75 00 00	00 00 63 22 08 00 45 00	(:M--u...c"·E-
0010	00 32 0f 41 00 00 ff 06	64 4a c0 a8 63 22 c0 a8	·2·A...d]·c"··
0020	63 c7 01 f6 c4 59 00 00	3b 26 5e 61 9d f8 50 18	c...Y...;&^a·P·
0030	1d 83 8d 78 00 00 bb bb	02 00 01 00 00 01 00 00	··x·...·
0040	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	··
0050	00 00 00 64 48 45 79 65	58 4e 6b 63 7a 59 32 20	···dHEye·XNkczY2
0060	20 20 20 20 20 20 20 20		

CSDN @末初

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

Recipe

From Base64

Alphabet
A-Za-z0-9+/=

Remove non-alphabet chars

Input

dHEyeXNkczY2

Output

tq2ysds66

CSDN @末初

flag{tq2ysds66}

异常流量分析

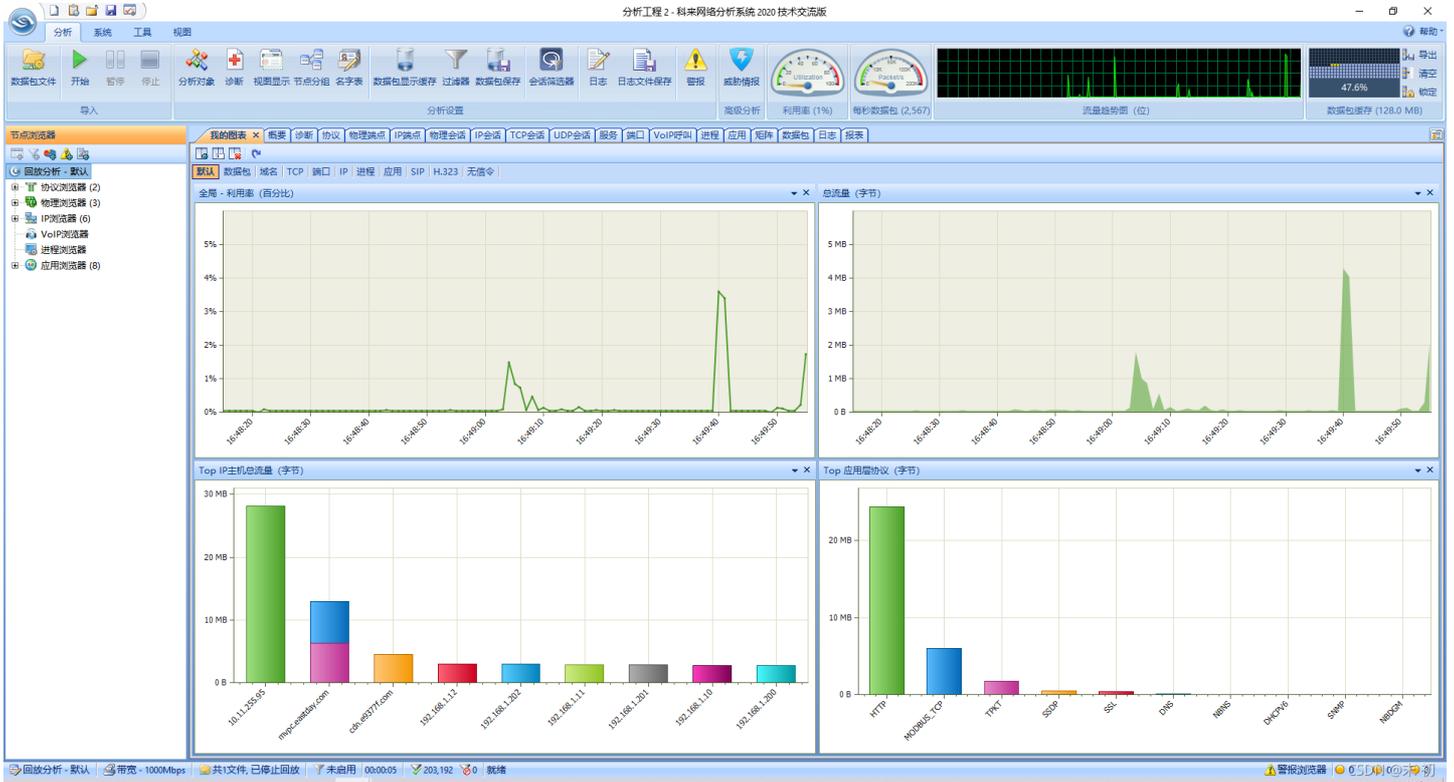
异常流量分析

30

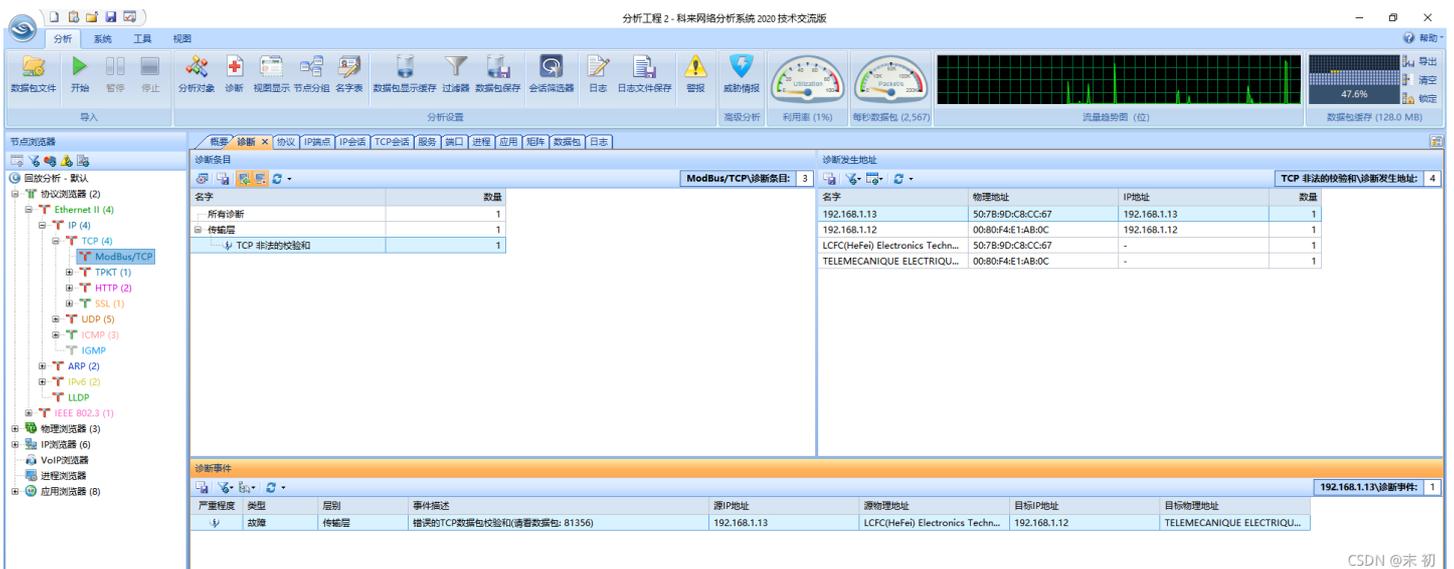
分值

已解答

某企业的运维工程师发现网络中出现流量异常，于是从场内一交换机抓取了数据包，请协助找出流量中针对正常的业务的异常数据内容，flag提交形式为flag{xxxx}。

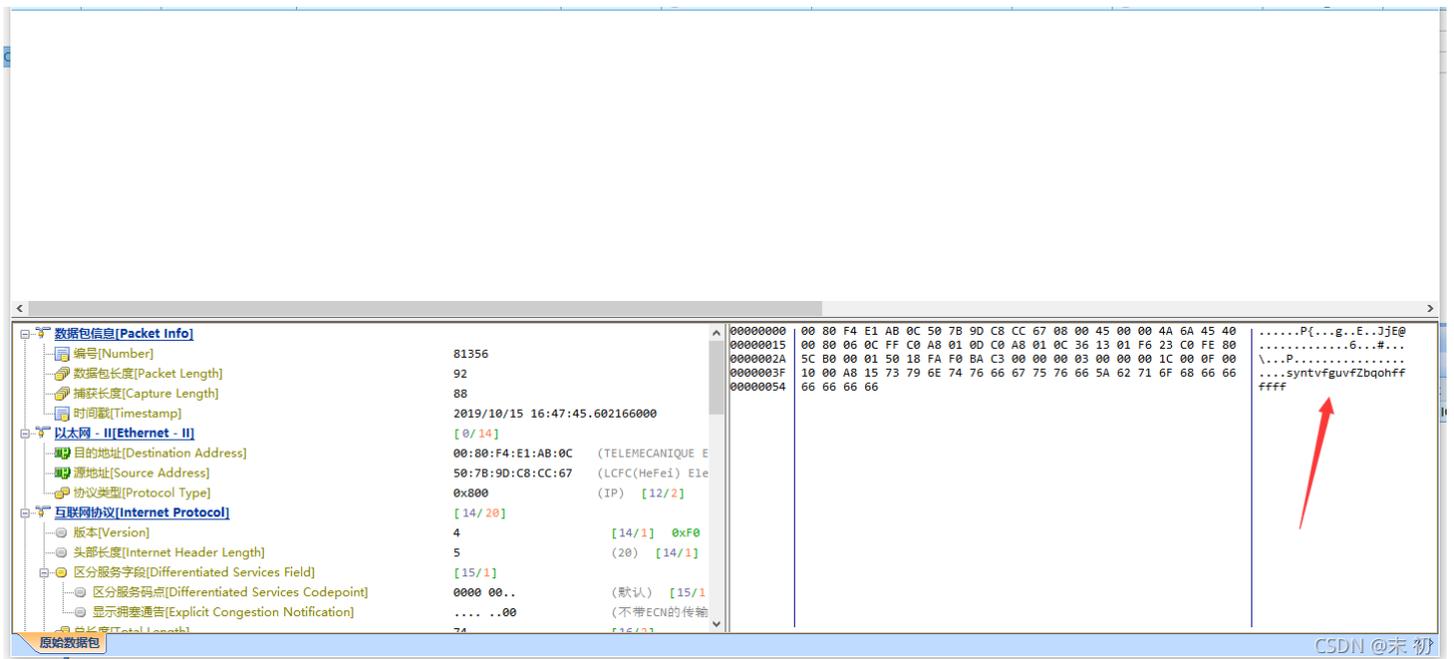


大部分为 HTTP 协议流量，但是查看http流和导出文件都没有发现flag的线索。继续查看少部分的 Modbus 协议



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

编号	日期	绝对时间	源	源端口	源地理位置	目标	目标端口	目标地理位置	协议	应用
81356	2019/10/15	16:47:45.602166000	192.168.1.13	13843	本地	192.168.1.12	502	本地	MODBUS_TCP	



```
syntvfguvfZbqohffff
```

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

rot13.com
[About ROT13](#)

```
syntvfguvfZbqohffff|
```



ROT13 ▾



```
flagisthisModbussssss
```

CSDN @末初

```
flag{flagisthisModbussssss}
```

OPC流量分析

50
分值

未解答

1 赣电东西 2 F421战队 3 jx.sgcc

OPC是微软公司的对象连接和嵌入技术在过程控制方面的应用，OPC标准定义了基于PC的客户机之间进行自动化数据实时交换的方法，因此OPC协议在工业控制现场使用非常多。请对提供的OPC通信流量进行分析，尝试找出流量中的flag。

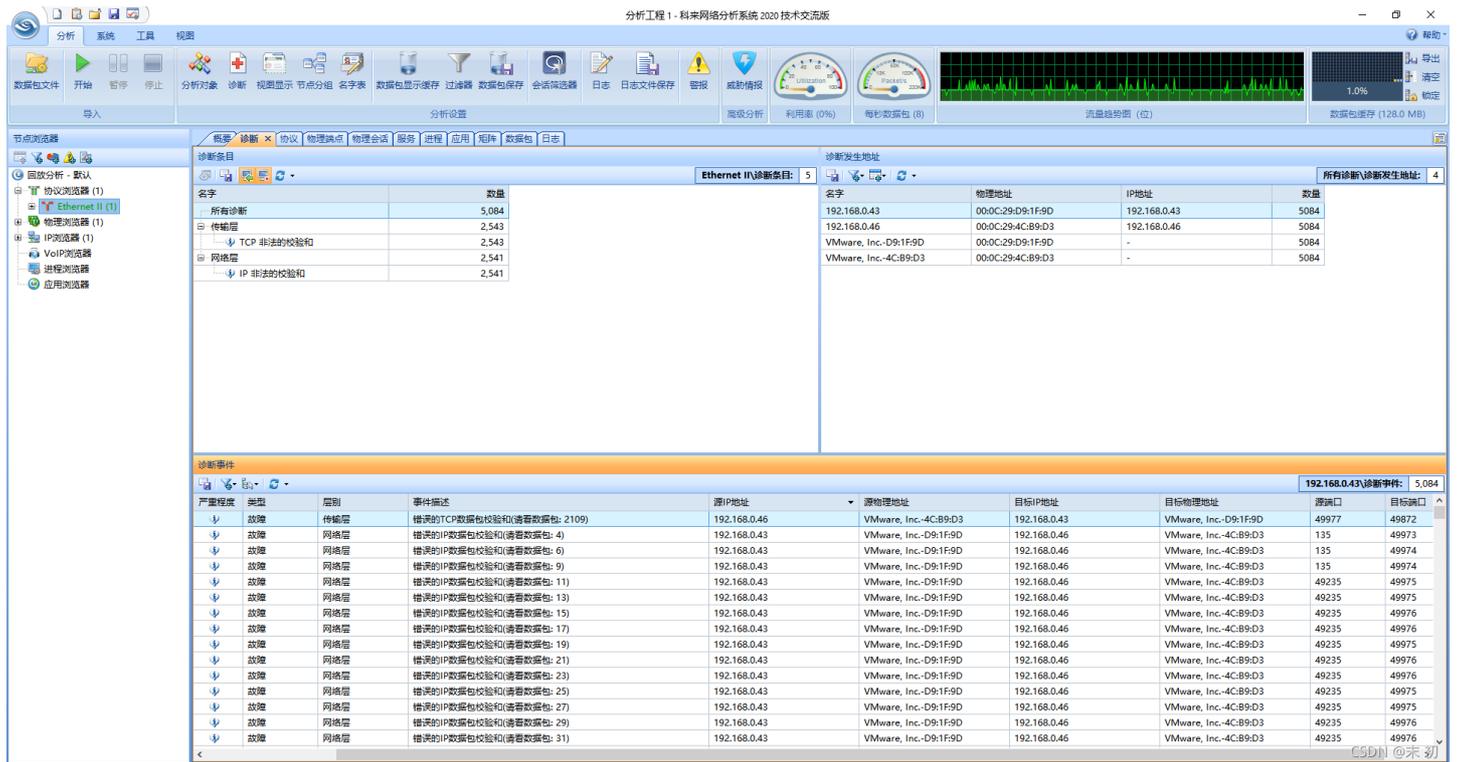
CSDN @末初

使用 [科来网络分析系统](#) 分析流量包

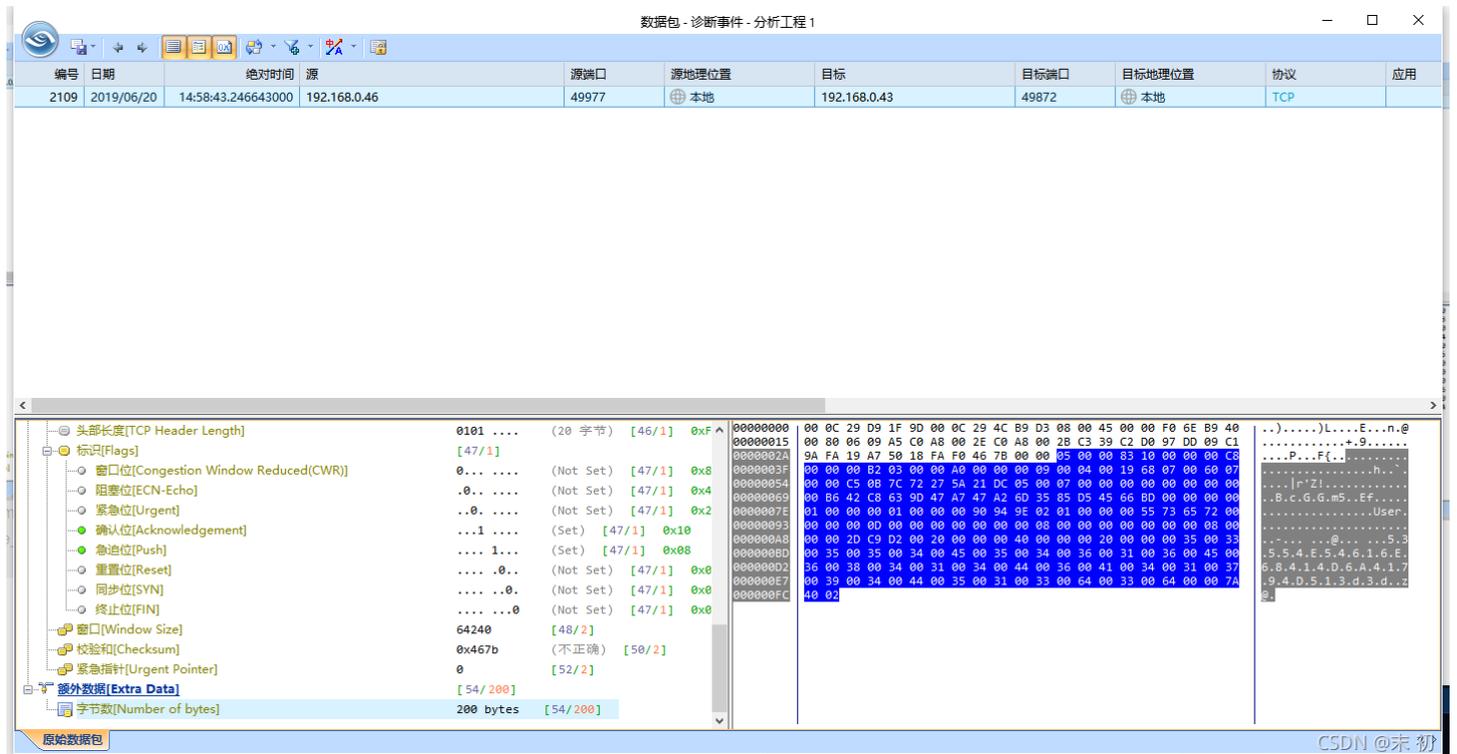
- [科来网络分析系统下载地址](#)

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'
>>>
```

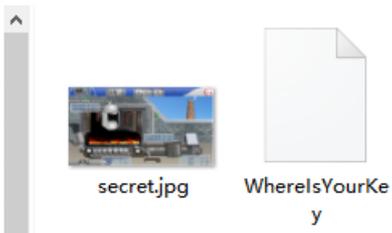
flag{ICSjx@2021}

应急处置

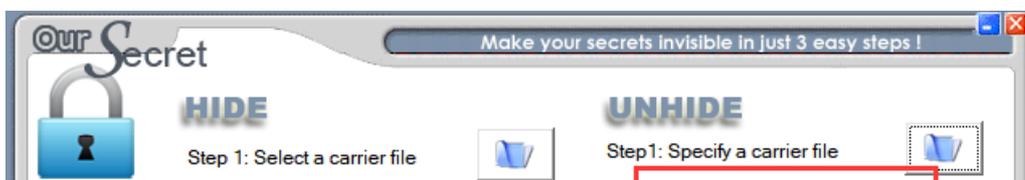
图片的秘密

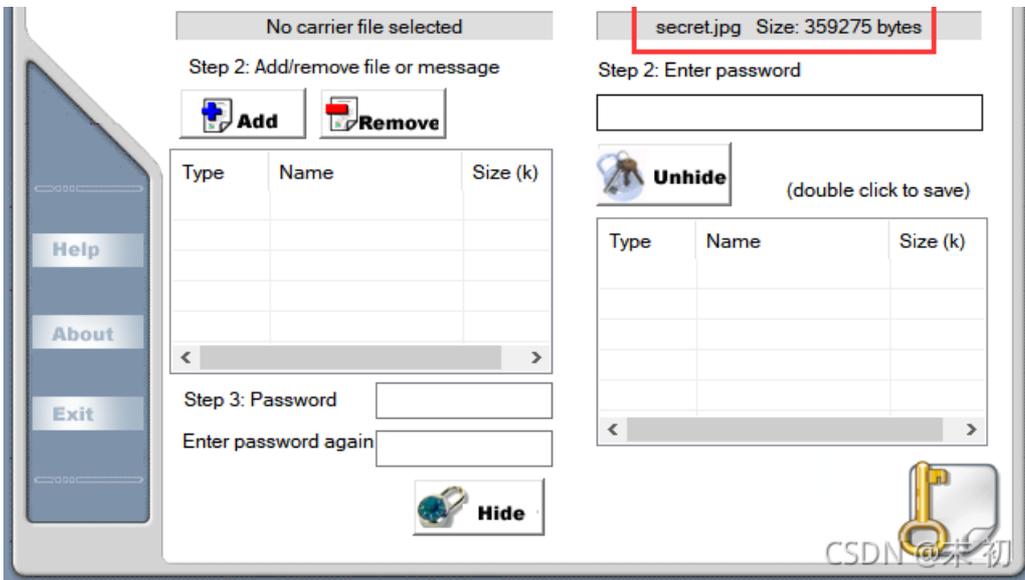


此电脑 > 下载 > ee8ff29a4e084466041e239070440e670068efb0



secret.jpg 根据文件名猜测为 OurSecret 隐写





下一步就是获取密码

```

PowerShell kali-linux
root@mochu7-pc: /mnt/c/Users/Administrator/Downloads/ee8ff29a4e084466041e239070440e670868efb0# ls
secret.jpg WhereIsYourKey
root@mochu7-pc: /mnt/c/Users/Administrator/Downloads/ee8ff29a4e084466041e239070440e670868efb0# file WhereIsYourKey
WhereIsYourKey: ELF 64-bit LSB executable, x86_64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.24, BuildID[sha1]=862ee37793af334043b423ba50ec91cfa132260a, n
ot stripped
root@mochu7-pc: /mnt/c/Users/Administrator/Downloads/ee8ff29a4e084466041e239070440e670868efb0#
root@mochu7-pc: /mnt/c/Users/Administrator/Downloads/ee8ff29a4e084466041e239070440e670868efb0#
root@mochu7-pc: /mnt/c/Users/Administrator/Downloads/ee8ff29a4e084466041e239070440e670868efb0# ./WhereIsYourKey
Usage: ./WhereIsYourKey password
This time the string is hidden and we used strcmp
root@mochu7-pc: /mnt/c/Users/Administrator/Downloads/ee8ff29a4e084466041e239070440e670868efb0# ./WhereIsYourKey password
password "password" not OK
root@mochu7-pc: /mnt/c/Users/Administrator/Downloads/ee8ff29a4e084466041e239070440e670868efb0# ./WhereIsYourKey 123456
password "123456" not OK
root@mochu7-pc: /mnt/c/Users/Administrator/Downloads/ee8ff29a4e084466041e239070440e670868efb0# ./WhereIsYourKey mochu777
password "mochu777" not OK
root@mochu7-pc: /mnt/c/Users/Administrator/Downloads/ee8ff29a4e084466041e239070440e670868efb0#

```

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

```

[ DISASM ]
> 0x4006d5 <compare_pwd+91> call strcmp@plt <strcmp@plt>
    s1: 0x7fffffffde50 ← 'my_m0r3_secur3_pwd'
    s2: 0x7fffffff2ef7 ← 0x5800363534333231 /* '123456' */

0x4006da <compare_pwd+96> test    eax, eax
0x4006dc <compare_pwd+98> jne    compare_pwd+112 <compare_pwd+112>

0x4006de <compare_pwd+100> mov    edi, 0x4007e8
0x4006e3 <compare_pwd+105> call   puts@plt <puts@plt>

0x4006e8 <compare_pwd+110> jmp    compare_pwd+134 <compare_pwd+134>

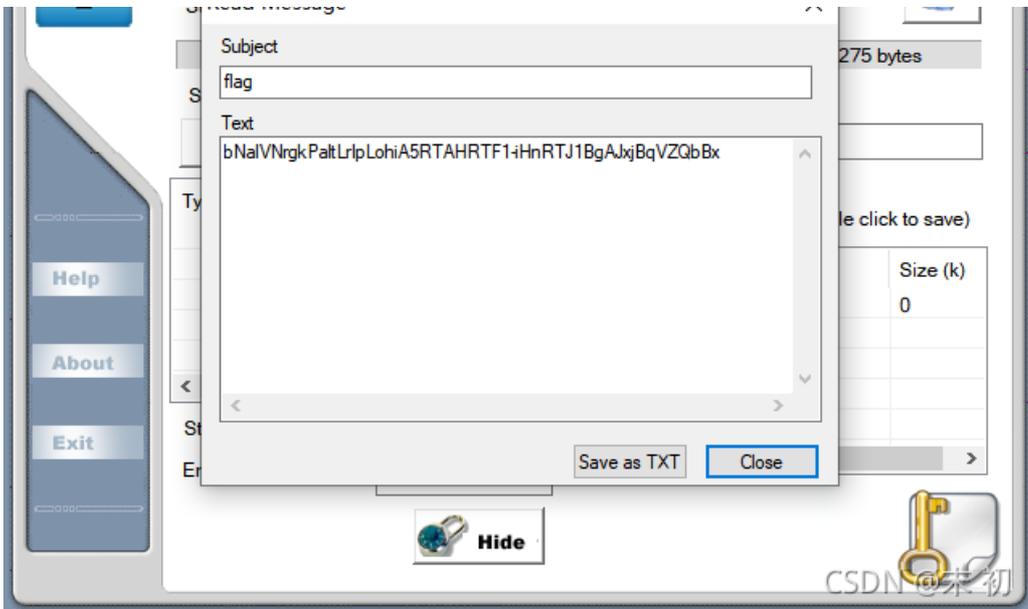
0x4006ea <compare_pwd+112> mov    rax, qword ptr [rbp - 0x28]
0x4006ee <compare_pwd+116> mov    rsi, rax
0x4006f1 <compare_pwd+119> mov    edi, 0x4007f4
0x4006f6 <compare_pwd+124> mov    eax, 0
0x4006fb <compare_pwd+129> call   printf@plt <printf@plt>

[ STACK ]

```

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**

XXencode

XXencode

```
bNalVNrgkPaltLrIpLohiA5RTAHRTF1-iHnRTJ1BgAJxjBqVZQbBx
```

字符集 utf8(unicode编码)

编码

解码

```
flag{0nly_u5_Kn0w_17_D0n07_T311_o7hers}
```

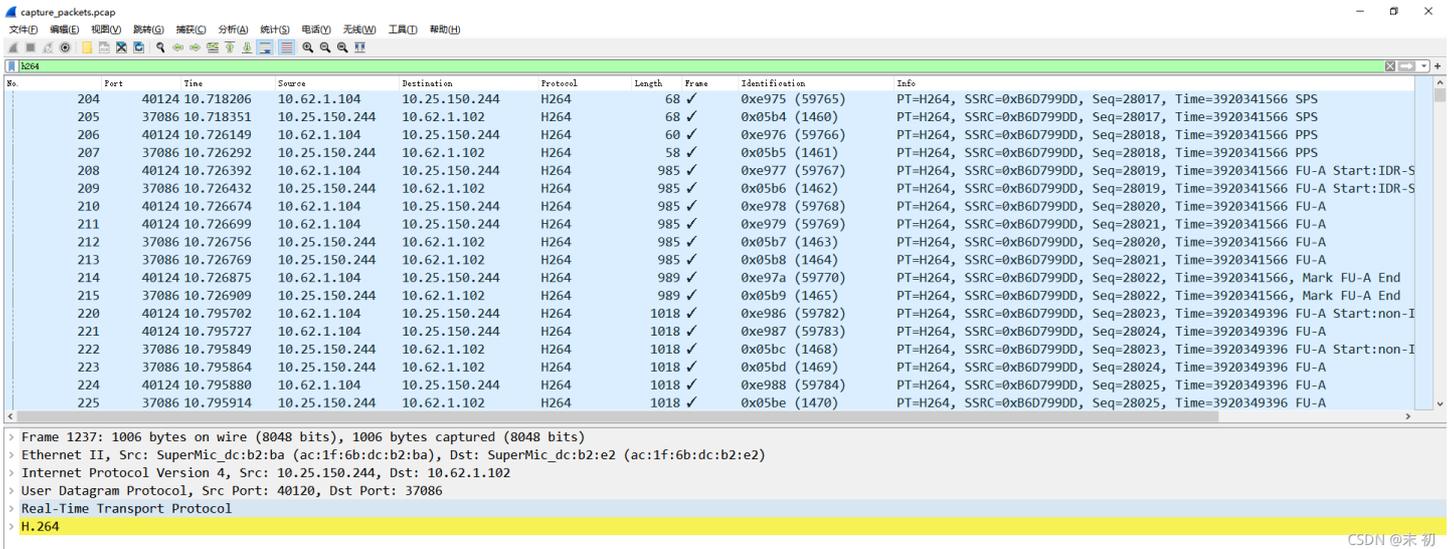
CSDN @末初

```
flag{0nly_u5_Kn0w_17_D0n07_T311_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



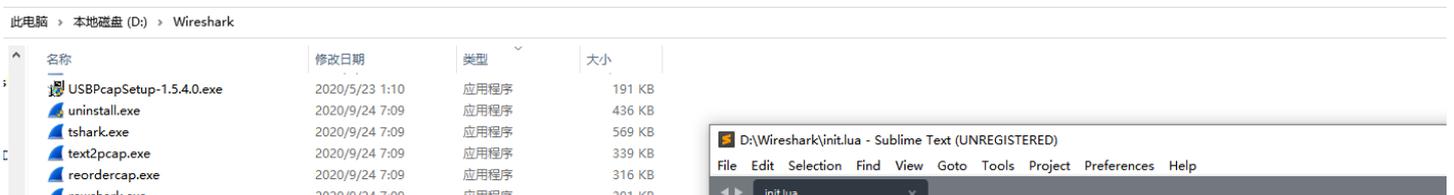
- H264 编解码协议详解

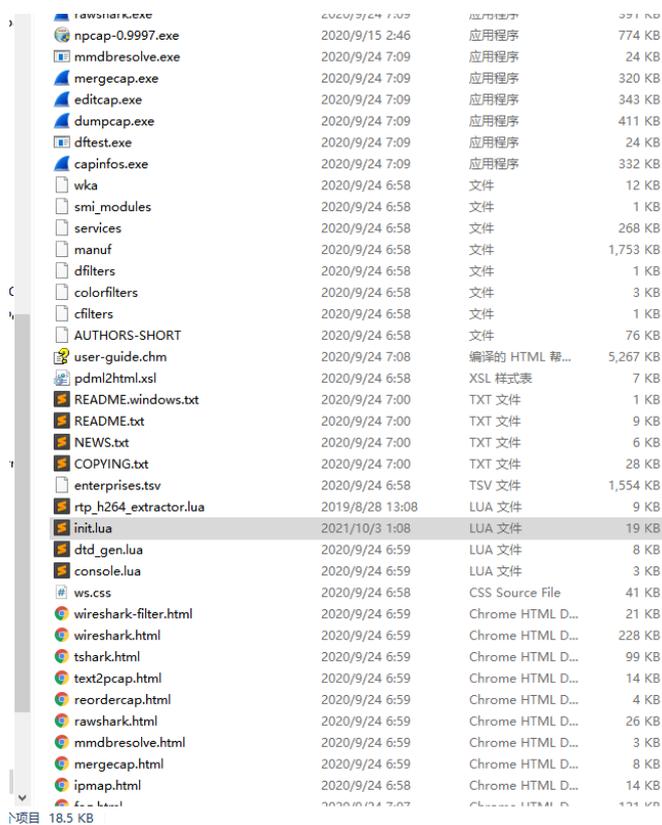
搜索引擎查阅发现可以Wireshark加载lua脚本提取出 h264 数据流, 然后利用 Eleccard StreamEye 分析

<https://github.com/volvet/h264extractor>

Eleccard StreamEye Basic 4.4

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





```

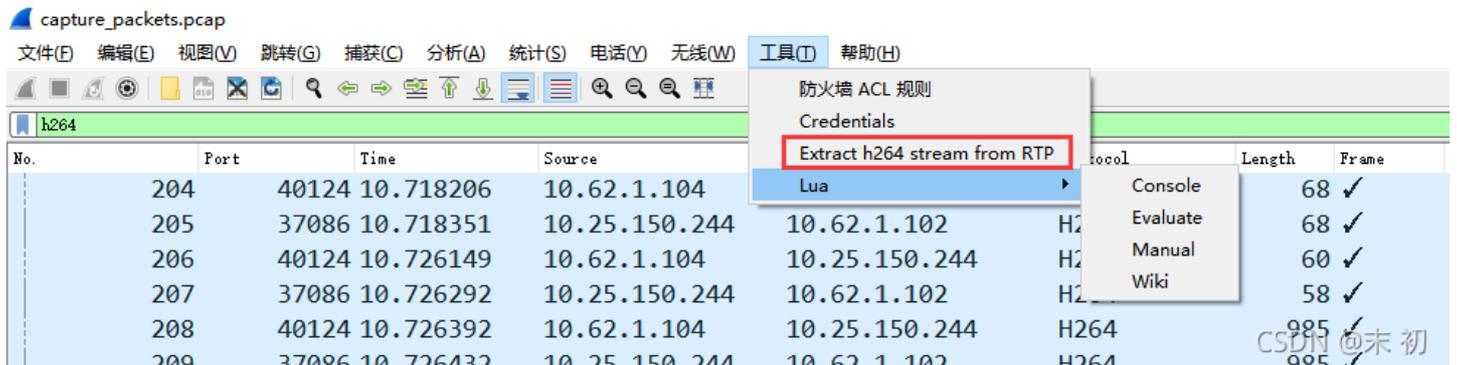
699
700 -- other useful constants
701 -- DATA_DIR and USER_DIR have a trailing directory separator.
702 GUI_ENABLED = gui_enabled()
703 DATA_DIR = Dir.global_config_path()..package.config:sub(1,1)
704 USER_DIR = Dir.personal_config_path()..package.config:sub(1,1)
705
706 -- deprecated function names
707 datafile_path = Dir.global_config_path
708 persconffile_path = Dir.personal_config_path
709
710
711 if not running_superuser or run_user_scripts_when_superuser then
712     dofile(DATA_DIR.."console.lua")
713 end
714 --dofile(DATA_DIR.."dtd_gen.lua")
715 dofile(DATA_DIR.."rtp_h264_extractor.lua")

```

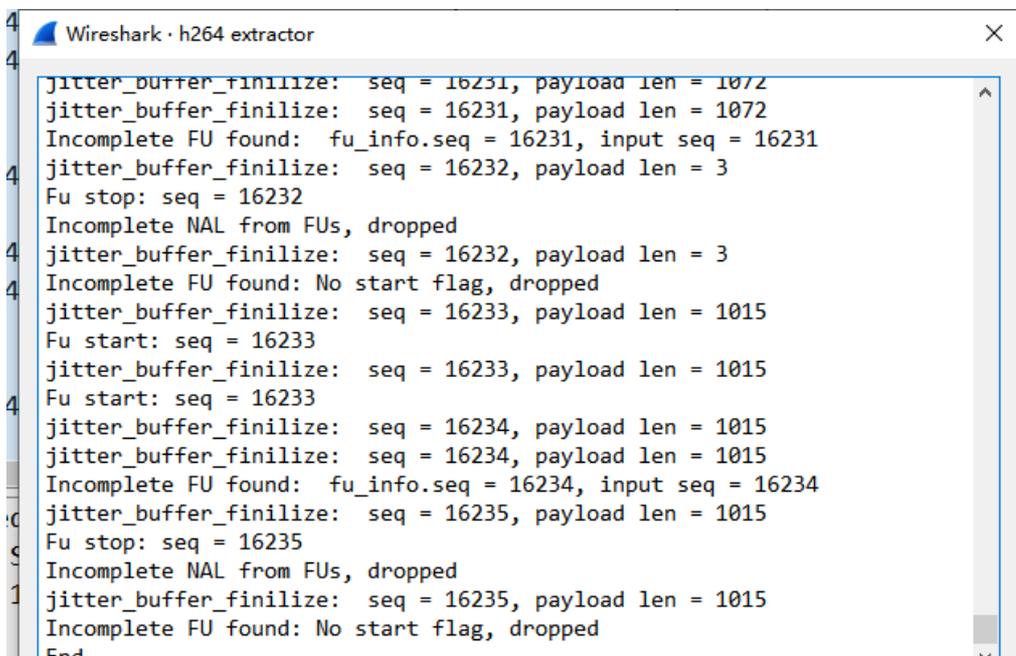
CSDN @末初

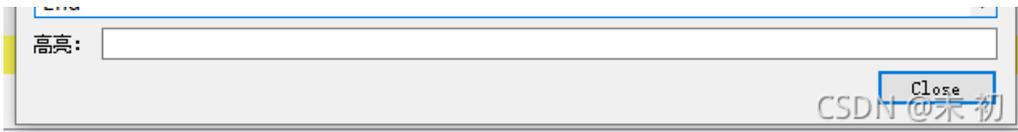
修改 `init.lua`，在最后面添加一行：`dofile(DATA_DIR.."rtp_h264_extractor.lua")`

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

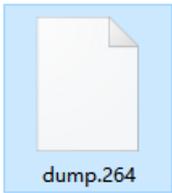


CSDN @末初





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

The screenshot shows the Elecard StreamEye 2021 Basic interface. The main window displays a video stream with a watermark that reads "FLAG (GOOD FOR YOU)". The interface includes several panels:

- BarChart:** Shows bit allocation and metric data for the stream.
- Stream:** Displays technical details for the video stream, including resolution (480 x 640), frame rate (25.00), and bit rate (602.780).
- Block Info:** Shows metadata for the current block, including mb location (80x176), slice_id (0), and mb_type (P_Skip).

`flag{GOODFORYOU}`

应急恢复



30

分值

已解答



YYDS这个flag真的绝绝子



江西外语外贸ctfk



JXCFS

文件为硬盘镜像副本，请恢复该硬盘中的文件，寻找Flag。

CSDN @末初

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

DiskGenius V5.4.0.1124 x64 发现新版本 V5.4.2.1239 主要更新内容如下: 26、纠正预览某些heif格式照片时画面显示不正常的问题。

文件(F) 磁盘(D) 分区(P) 工具(T) 查看(V) 帮助(H)

保存更改 搜索分区 恢复文件 快速分区 新建分区 格式化 删除分区 备份分区 系统迁移

数据丢失怎么办

DiskGenius 团队为您服务 致电: 400-008-9958 或点击此处选择QQ咨询

基本 MBR

data(0) NTFS 1021.0MB

磁盘2 接口:File 型号:Disk Image 容量:1.0GB(1024MB) 柱面数:130 磁头数:255 每道扇区数:63 总扇区数:2097153

- HD0:ST1000LM035-1RK172(932GB)
 - 本地磁盘 (D:)
 - HD1:TOSHIBATHNSFJ256GDNVA(231GB)
 - 恢复(0)
 - ESP(1)
 - MSR(2)
 - 本地磁盘 (C:)
 - 分区(4)
 - VD0:Data.img(1GB)
 - 分区(恢复文件)
 - data(已识别)(0)
 - \$Extend

分区参数 浏览文件 扇区编辑

名称

修改时间 创建时间

恢复文件 - VD0:Data.img(1GB)

恢复选项:

- 恢复已删除的文件
- 完整恢复 高级选项
- 额外扫描已知文件类型 选择文件类型
- 加载扫描进度
- 扫描时阻止系统睡眠 开始

CSDN @末初

基本 MBR

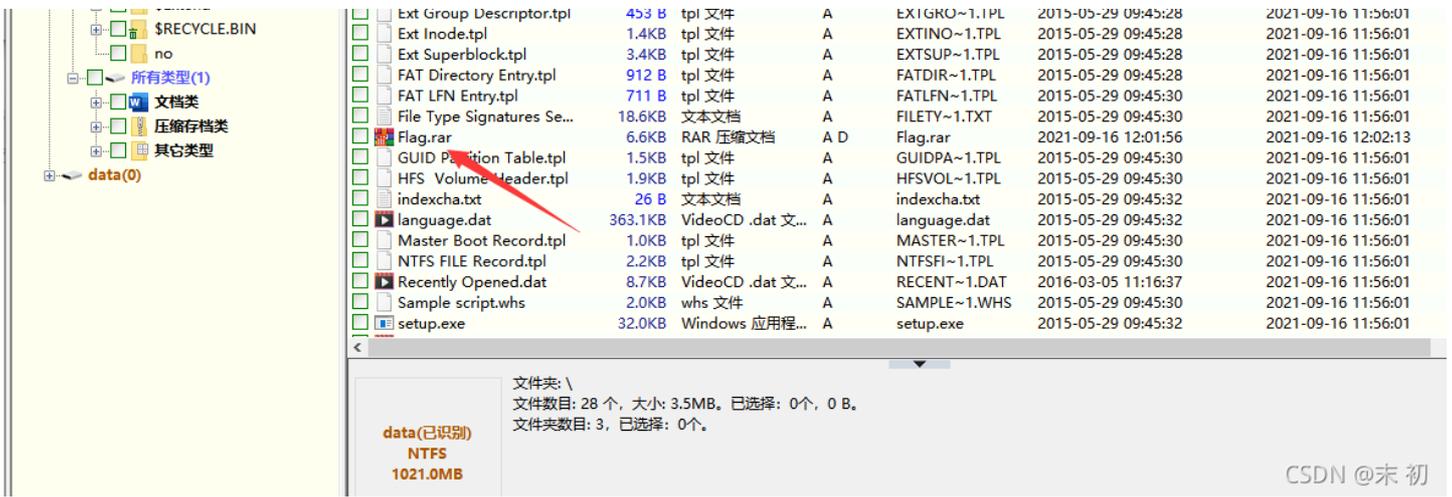
data(0) NTFS 1021.0MB

磁盘2 接口:File 型号:Disk Image 容量:1.0GB(1024MB) 柱面数:130 磁头数:255 每道扇区数:63 总扇区数:2097153

- HD0:ST1000LM035-1RK172(932GB)
 - 本地磁盘 (D:)
 - HD1:TOSHIBATHNSFJ256GDNVA(231GB)
 - 恢复(0)
 - ESP(1)
 - MSR(2)
 - 本地磁盘 (C:)
 - 分区(4)
 - VD0:Data.img(1GB)
 - 分区(恢复文件)
 - data(已识别)(0)
 - \$Extend

名称: *.* (*.jpg;*.bmp) 已删除 正常文件 系统文件 重复文件 过滤 更多 >>

名称	大小	文件类型	属性	短文件名	修改时间	创建时间
\$Extend		文件夹	HS	\$Extend	2021-09-16 11:54:06	2021-09-16 11:54:06
\$RECYCLE.BIN		文件夹	HS	\$RECYCLE.BIN	2021-09-16 12:04:08	2021-09-16 12:04:08
no		文件夹		no	2021-09-16 11:56:25	2021-09-16 11:56:09
Boot Sector FAT.tpl	1.2KB	tpl 文件	A	BOOTSE~1.TPL	2015-05-29 09:45:24	2021-09-16 11:56:01
Boot Sector FAT32.tpl	1.4KB	tpl 文件	A	BOOTSE~2.TPL	2015-05-29 09:45:26	2021-09-16 11:56:01
Boot Sector NTFS.tpl	1.6KB	tpl 文件	A	BOOTSE~3.TPL	2015-05-29 09:45:26	2021-09-16 11:56:01
Chinese.dat	21.0KB	VideoCD .dat 文...	A	Chinese.dat	2015-05-29 09:45:26	2021-09-16 11:56:01
Chinese.txt	41.8KB	文本文档	A	Chinese.txt	2015-05-29 09:45:26	2021-09-16 11:56:01
Ext Directory Entry.tpl	586 B	tpl 文件	A	EXTDIR~1.TPL	2015-05-29 09:45:26	2021-09-16 11:56:01



CSDN @末初

flag{73D3DA963F7505E9}

CSDN @末初

flag{73D3DA963F7505E9}

文件分析

文件分析

30

分值

已解答

1 一道题都做不队

2 Stalker戴戴我

3 星火燎原队

这是工艺监控流程文件被人破坏，写入了某些特别的内容，请根据文件，找出其中的flag。提交格式：flag{xxx}。

CSDN @末初

```

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

```

起始页 what x
编辑方式: 十六进制(H) 运行脚本 运行模板
0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789A B C D E F
0000h: 00 00 03 FD 00 00 00 20 08 06 00 00 00 17 96 D7 ...ý... ..-x
0010h: 41 00 00 00 01 73 52 47 42 00 AE CE 1C E9 00 00 A....sRGB.@Í.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...Ă...Ă.Ç
0040h: 6F A8 64 00 00 DC A5 49 44 41 54 78 5E ED BD 0D o`d..Û¥IDATx^í%#
0050h: F0 55 D5 7D A8 CD 6D 7A D3 A4 93 1B FB F1 D6 DB ŸUŌ} ``ímzÓα``.ûñŌŪ
0060h: B4 62 FB 8E 4D 9A 26 D1 E9 5B 92 B9 5E 6F CC FB `búžMš&Ñé[' ^°iû
0070h: 4E 6B 2D 6F 12 A7 49 AF F7 B6 4D E8 9D 0C F5 05 Nk-o.ŠI~÷ŹMè..ö.
0080h: 41 44 09 05 09 12 45 87 82 14 B9 22 21 20 84 10 AD...E‡,.!"! ".
0090h: 94 52 11 4D 50 4B 09 91 52 F1 AB 51 8B C4 94 72 "R.MPK.`Rñ«Q<Ă"r
00A0h: FD 20 69 A4 64 12 6F 6D 32 69 66 D2 4E 6F EF CC ý i=d.GSDN@木初
00B0h: FF 2D DF 7D CF 23 67 0D 7D F6 DF 6D FD FF 8F F2

```

```

DC90h: 80 00 04 20 00 01 08 40 00 02 10 F0 27 80 F4 23 €.. ...@...ø'èó#
DCA0h: FD 10 80 00 04 20 00 01 08 40 00 02 10 80 00 04 ý.ε.. ...@...é..
DCB0h: 20 00 81 8E 12 40 FA 3B 5A B0 FE ED 3E 6C 09 01 ..ž.Ōú;Z°pí>1..
DCC0h: 08 40 00 02 10 80 00 04 20 00 01 08 40 00 02 5D .@...ε... ..@..]
DCD0h: 25 80 F4 23 FD 10 80 00 04 20 00 01 08 40 00 02 %èó#ý.ε... ..@..
DCE0h: 10 80 00 04 20 00 81 8E 12 F8 FF 01 3D 4F F5 1A .ε... ..ž.øÿ.=Oö.
DCF0h: 4C A4 9D 0C 00 00 00 00 49 45 4E 44 AE 42 60 82 Lα.....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

```

起始页 what x
编辑方式: 十六进制(H) 运行脚本 运行模板
0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789A B C D E F
0000h: 00 00 03 FD 00 00 00 20 08 06 00 00 00 17 96 D7 ...ý... ..-x
0010h: 41 00 00 00 01 73 52 47 42 00 AE CE 1C E9 00 00 A....sRGB.@Í.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...Ă...Ă.Ç
0040h: 6F A8 64 00 00 DC A5 49 44 41 54 78 5E ED BD 0D o`d..Û¥IDATx^í%#
0050h: F0 55 D5 7D A8 CD 6D 7A D3 A4 93 1B FB F1 D6 DB ŸUŌ} ``ímzÓα``.ûñŌŪ

```

```

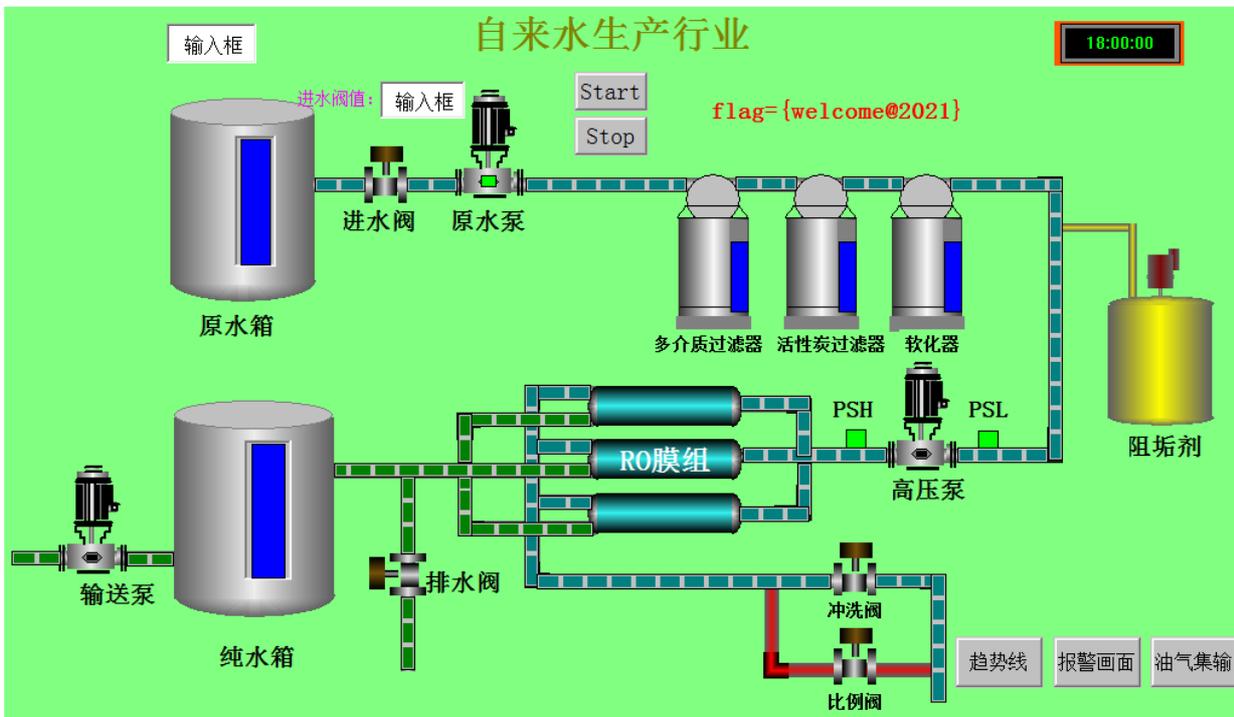
起始页 what 1212.png x
编辑方式: 十六进制(H) 运行脚本 运行模板: PNG.bt
0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789A B C D E F
0000h: 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 %PNG.....IHDR
0010h: 00 00 04 E4 00 00 08 00 08 06 00 00 00 86 B4 EC ...ä.....t`i
0020h: FC 00 00 00 01 73 52 47 42 00 AE CE 1C E9 00 00 ü....sRGB.@Í.é..
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...Ă...Ă.Ç
0050h: 6F A8 64 00 00 FF A5 49 44 41 54 78 5E EC FD F5 o`d..Û¥IDATx^íÿö

```

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

```
起始页 what x 1212.png
编辑方式: 十六进制(H) 运行脚本 运行模板
0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0000h: 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 %PNG.....IHDR
0010h: 00 00 03 FD 00 00 00 20 08 06 00 00 00 17 96 D7 ...ý... -x
0020h: 41 00 00 00 01 73 52 47 42 00 AE CE 1C E9 00 00 A....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...Ä...Ç
0050h: 6F A8 64 00 00 DC A5 49 44 41 54 78 5E ED BD 0D o`d..Û¥IDATx^î%.
0060h: F0 55 D5 7D A8 CD 6D 7A D3 A4 93 1B FB F1 D6 DB 8UÖ}~ímzÖα".ûñÖÜ
0070h: F4 62 FB 8F 4D 8A 26 D1 F8 5B 82 B8 5E 6E CC FB (hâžMšřÑóL/1âöîâ
```

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}

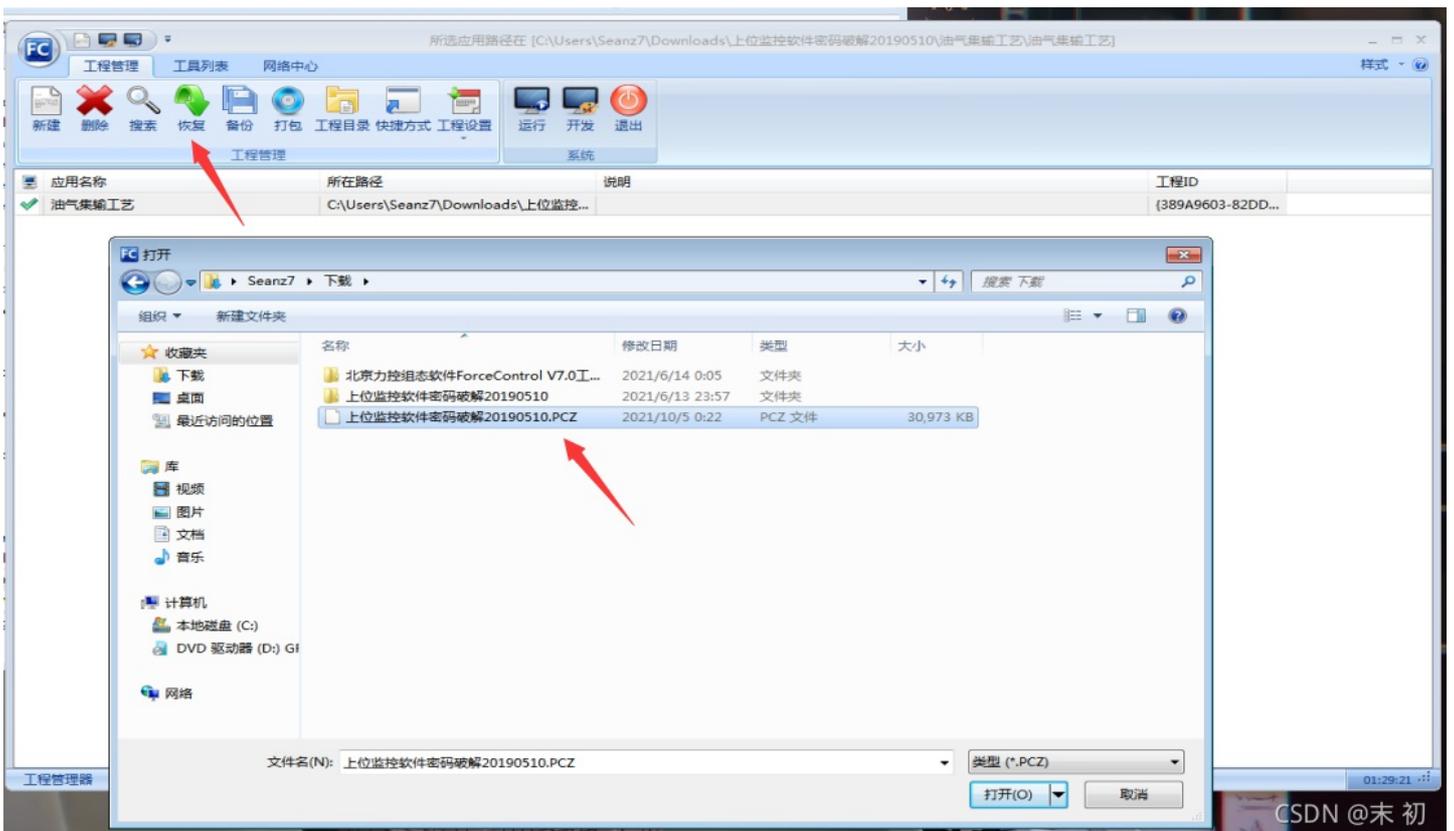
组态编程

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

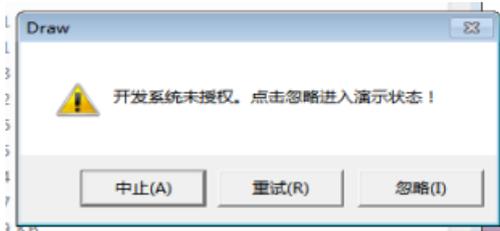
- 北京力控组态软件ForceControl V7.0(低版本只适配Windows7)



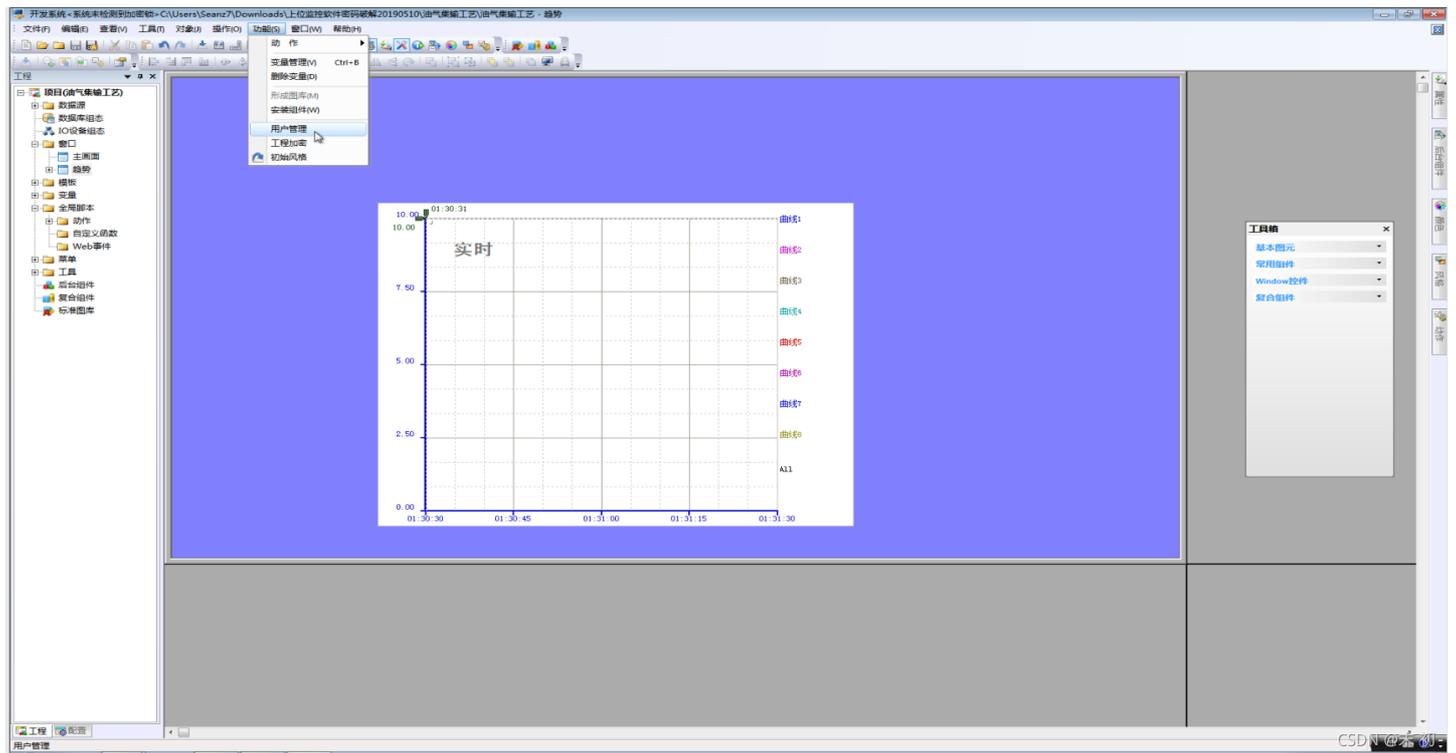
然后点击 开发



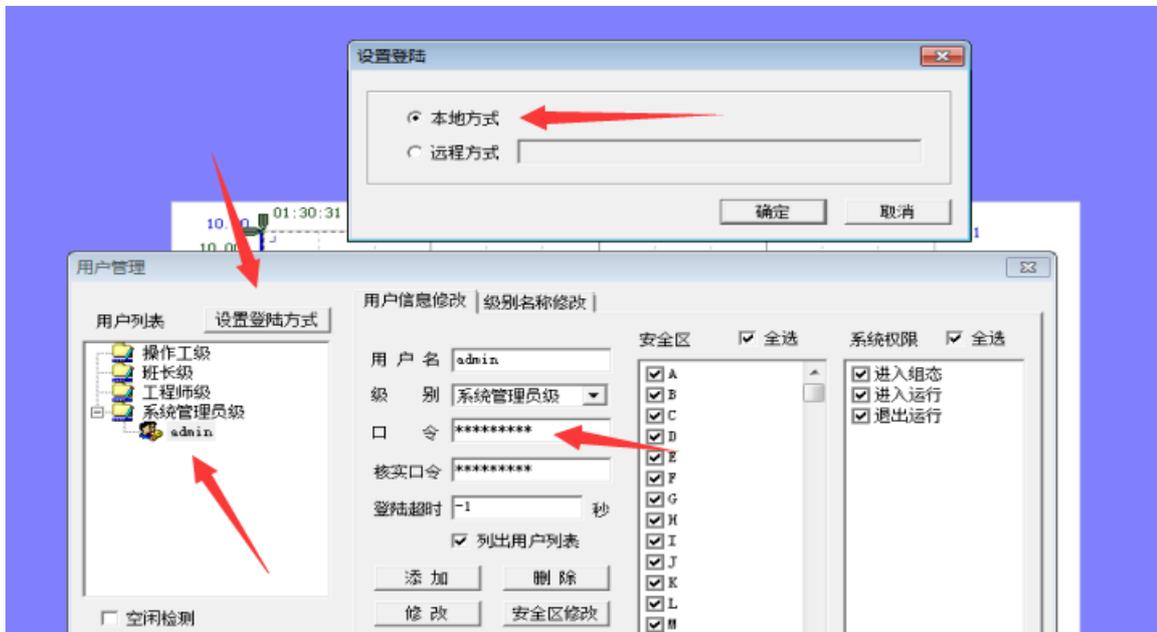
选择 忽略



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 classes.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

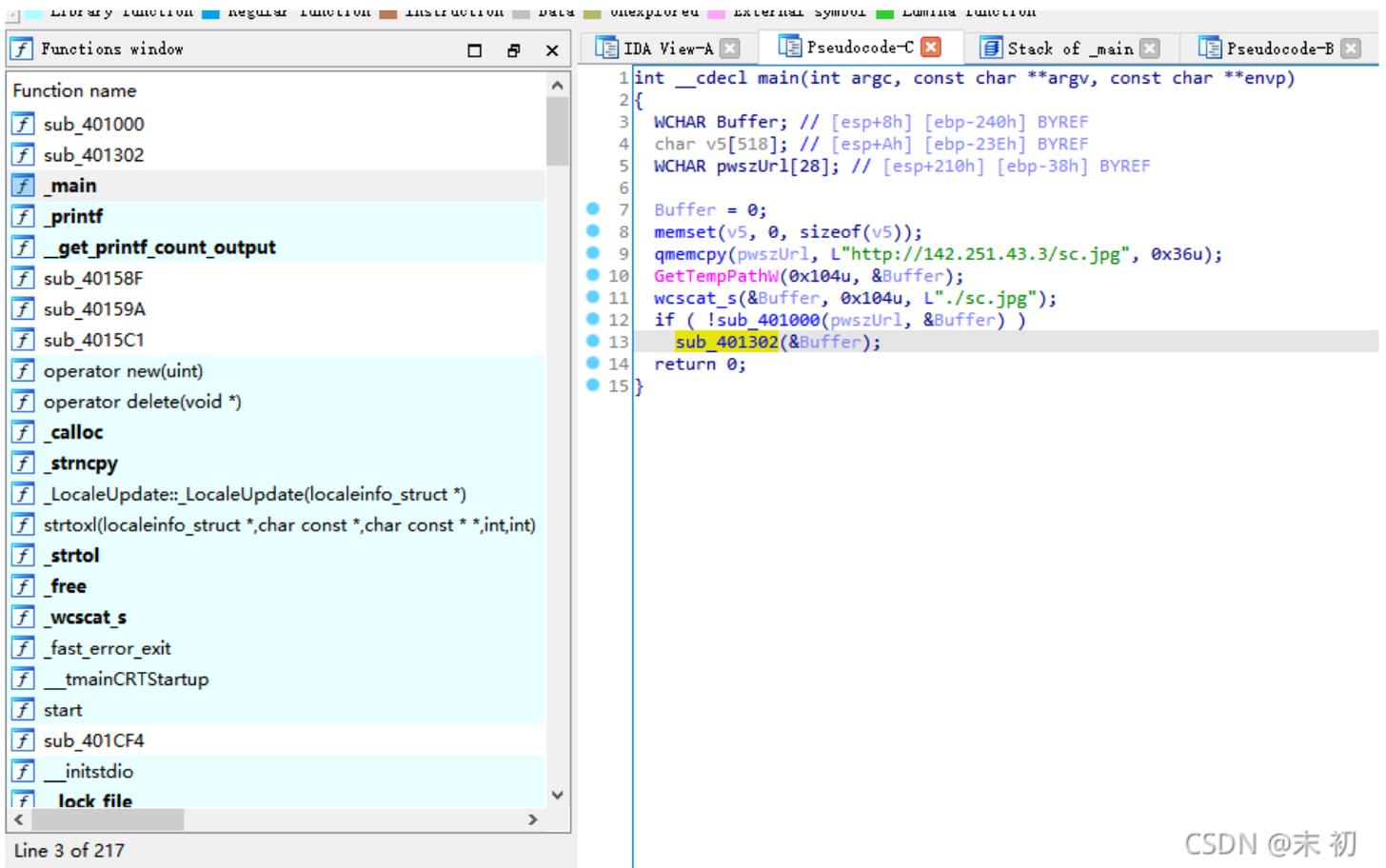
jpg 用 010 Editor 打开啥也看不出来，猜测应该是被 exe 文件处理过的

```

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#

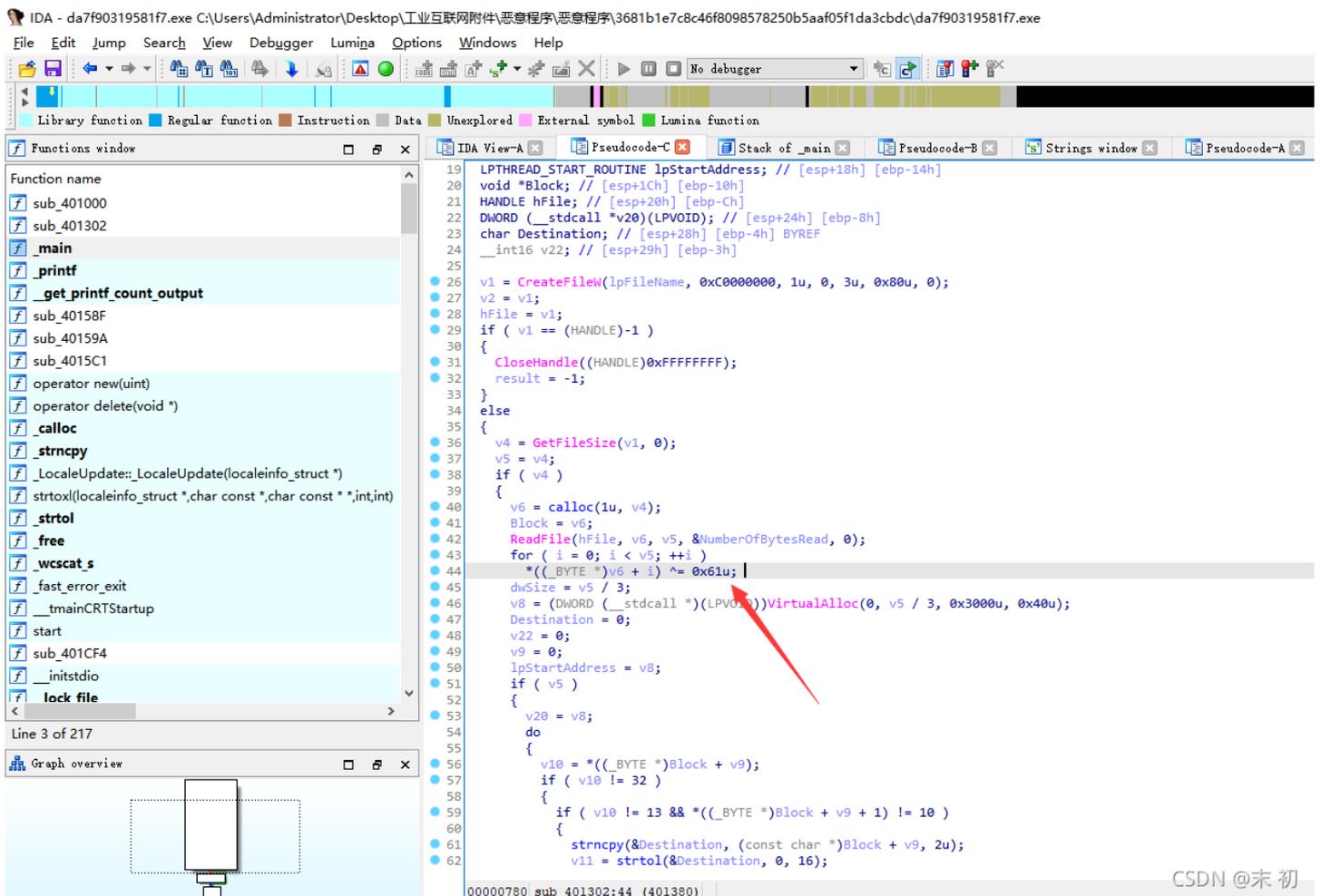
```

ida 打开 exe 文件



CSDN @末初

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



CSDN @末初

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

十六进制运算

2 进制异或:

将数据视为(T): 带符号字节

操作数(O): 31

十进制(D) 十六进制(X)

描述

用操作数对每个值按位进行“异或”运算。
X[i] = 操作数

选项(O)

确定(O) 取消(C) 帮助(H)

Offset	Hex	ASCII
0170h	30 20 46 42 20 36 31 20 38 44 20 34 31 20 45 30	0 FB 61 8D 41 E0
0180h	20 30 46 20 34 43 0D 0A 43 31 20 30 33 20 44 30	0F 4C ..C1 03 D0
0190h	20 34 36 20 38 41 20 31 45 20 38 34 20 44 42 20	46 8A 1E 84 DB
01A0h	37 35 20 45 38 20 35 45 20 38 42 20 43 32 20 35	75 E8 5E 8B C2 5
01B0h	42 20 43 33 20 38 44 0D 0A 34 31 20 46 38 20 43	B C3 8D ..41 F8 C
01C0h	33 20 35 35 20 38 42 20 45 43 20 38 33 20 45 43	3 55 8B EC 83 EC
01D0h	20 31 34 20 35 33 20 35 36 20 35 37 20 38 39 20	14 53 56 57 89
01E0h	34 44 20 46 34 20 36 34 0D 0A 41 31 20 33 30 20	4D F4 64 ..A1 30
01F0h	30 32 20 30 32 20 30 32 20 38 39 20 34 35 20 46	02 02 02 89 45 F
0200h	43 20 38 42 20 34 35 20 46 43 20 38 42 20 34 30	C 8B 45 FC 8B 40
0210h	20 30 43 20 38 42 20 34 30 0D 0A 31 34 20 38 42	0C 8B 40 ..14 8B
0220h	20 46 38 20 38 39 20 34 35 20 45 43 20 38 42 20	F8 89 45 EC 8B
0230h	43 46 20 45 38 20 44 32 20 46 46 20 46 46 20 46	CF E8 D2 FF FF F
0240h	46 20 38 42 20 33 46 20 38 42 0D 0A 37 30 20 31	F 8B 3F 8B ..70 1
0250h	38 20 38 35 20 46 36 20 37 34 20 34 46 20 38 42	8 85 F6 74 4F 8B
0260h	20 34 36 20 33 43 20 38 42 20 35 43 20 33 30 20	46 3C 8B 5C 30
0270h	37 38 20 38 35 20 44 42 20 37 34 0D 0A 34 34 20	78 85 DB 74 ..44
0280h	38 42 20 34 43 20 33 33 20 30 43 20 30 33 20 43	8B 4C 33 0C 03 C
0290h	45 20 45 38 20 38 45 20 46 46 20 46 46 20 46 46	E E8 8E FF FF FF
02A0h	20 38 42 20 34 43 20 33 33 20 32 30 0D 0A 38 39	8B 4C 33 20 ..89
02B0h	20 34 35 20 46 38 20 30 33 20 43 45 20 33 33 20	45 F8 03 CE 33
02C0h	43 30 20 38 39 20 34 44 20 46 30 20 38 39 20 34	C0 89 4D F0 89 4
02D0h	35 20 46 43 20 33 39 20 34 34 20 33 33 0D 0A 31	5 FC 39 44 33 ..1
02E0h	38 20 37 36 20 32 32 20 38 42 20 30 43 20 38 31	8 76 22 8B 0C 81
02F0h	20 30 33 20 43 45 20 45 38 20 36 44 20 46 46 20	03 CE E8 6D FF
0300h	46 46 20 46 46 20 30 33 20 34 35 20 46 38 0D 0A	FF FF 03 45 F8 ..
0310h	33 39 20 34 35 20 46 34 20 37 34 20 31 43 20 38	39 45 F4 74 1C 8
0320h	42 20 34 35 20 46 43 20 38 42 20 34 44 20 46 30	B 45 FC 8B 4D F0
0330h	20 34 30 20 38 39 20 34 35 20 46 43 20 33 42 0D	40 89 45 FC 3B .
0340h	0A 34 34 20 33 33 20 31 38 20 37 32 20 44 45 20	.44 33 18 72 DE
0350h	33 42 20 37 44 20 45 43 20 37 35 20 39 43 20 33	3B 7D EC 75 9C 3
0360h	33 20 43 30 20 35 46 20 35 45 20 35 42 20 43 39	3 C0 5F 5E 5B C9
0370h	0D 0A 43 33 20 38 42 20 34 44 20 46 43 20 38 42	..C3 8B 4D FC 8B
0380h	20 34 34 20 33 33 20 32 34 20 38 44 20 30 34 20	44 33 24 8D 04
0390h	34 38 20 30 46 20 42 37 20 30 43 20 33 30 20 38	48 0F B7 0C 30 8
03A0h	42 0D 0A 34 34 20 33 33 20 31 43 20 38 44 20 30	B ..44 33 1C 8D 0
03B0h	34 20 38 38 20 38 42 20 30 34 20 33 30 20 30 33	4 88 8B 04 30 03
03C0h	20 43 36 20 45 42 20 44 46 20 32 30 20 30 32 20	C6 EB DF 20 02
03D0h	30 32 0D 0A 30 32 20 33 33 20 30 32 20 30 32 20	02 ..02 33 02 02
03E0h	30 32 20 33 38 20 30 32 20 30 32 20 30 32 20 37	02 38 02 02 02 7
03F0h	35 20 37 33 20 36 35 20 37 32 20 33 33 20 33 32	5 73 65 72 33 32
0400h	20 32 45 0D 0A 36 34 20 36 43 20 36 43 20 30 32	2E ..64 6C 6C 02
0410h	20 36 36 20 36 43 20 36 31 20 36 37 20 37 42 20	66 6C 61 67 7B
0420h	33 34 20 33 39 20 36 32 20 36 31 20 33 35 20 33	34 39 62 61 35 3
0430h	39 20 36 31 0D 0A 36 32 20 36 32 20 36 35 20 33	9 61 ..62 62 65 3
0440h	35 20 33 36 20 36 35 20 33 30 20 33 35 20 33 37	5 36 65 30 35 37
0450h	20 37 44 20 30 32 20 37 39 20 36 46 20 37 35 20	7D 02 79 6F 75
0460h	32 30 20 36 37 0D 0A 36 46 20 37 34 20 32 30 20	20 67 ..6F 74 20
0470h	36 39 20 37 34 20 32 31	69 74 21

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

固件分析

丢失的密码

小明作为工厂运维人员，对路由器固件进行升级操作。升级后尝试使用默认帐户信息以admin / admin身份登录以设置路由器，但无法连接。分析原因并获取帐户信息。密码即为flag。

```
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  工业互联网分析
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 已正式上线，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)

CSDN @末初

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
```

CSDN @末初

```
PowerShell x kali-linux x + v
create_inode: failed to create character device squashfs-root/squashfs-root/dev/tty0, because Operation not permitted
create_inode: failed to create character device squashfs-root/squashfs-root/dev/tty1, because Operation not permitted
create_inode: failed to create character device squashfs-root/squashfs-root/dev/tty2, because Operation not permitted
create_inode: failed to create character device squashfs-root/squashfs-root/dev/tty3, because Operation not permitted
create_inode: failed to create character device squashfs-root/squashfs-root/dev/tty4, 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
6d1c8915f6ccce78785f8f4c97d469d.png 82283ae0624def7ff597d75ac4ca705303625165.7z 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          lux_download.cgi              menu_list_right.png          private                    trafficconf
192.168.0.1          etcsystem_setup.png           lux_get.cgi                   mentor.gif                   product_db                 trafficconf_connadvanced_help.html
192.168.255.1       expertconf                    luxpe.png                     mgmtport.html                prsig                      trafficconf_connctrl_help.html
22860                expertconf_advertise_help.html iwconfig                       mgr_icon.png                 ps                           trafficconf_connctrl.js
22860.7z             expertconf_advertise.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
```

```
addnac.html      expertconf_hostscan.Lang.js  jquery.mobile-1.4.5.min.css  minipnpd.conf              qos                    trafficconf_qos_help.html
addmanual.html  expertconf_ipvtv.js          jquery.mobile-1.4.5.min.js  minus_icon.gif             qos.png               trafficconf_qos.js
addroute.html   expertconf_ipvtv.Lang.js    jquery.mobile.css           minus_icon_green.png      qos_rule.html         trafficconf_qos.Lang.js
addsearch.html  expertconf.js                jquery.selectlist.js        minus_icon.png             radio_off.png          trafficconf_switch_help.html
adduser         expertconf_pptpvpn_help.html js                            mipsel-linux-uclibc       radio_on.png           trafficconf_switch.js
advancesetup    expertconf_pptpvpn.js        kill                          misc                        ram                    trafficconf_switch.Lang.js
advertise        expertconf_pptpvpn.Lang.js  klog                          misc.html                  ram0                   trigger_off.png
afp.html        expertconf_remotepc_help.html klogd                          mount                       ram1                   trigger_on.png
ajax-loader.gif expertconf_remotepc.js       klogd                          mknod                       ram2                   triggerstatus.html
alert.png       expertconf_remotepc.Lang.js  kr                              m_login.cgi                ram3                   true
apache.html     explorerbg2.gif              kr.js                          mobile.css                  rc                      tty0
apcpd           exp                           lang                            mobile.css.backup          rcs                     tty1
apeclan.html    extendssetup.html           lang.js                        modify.html                 reboot                  tty2
apply_ani.gif   external.png                 laninfo                        modifyroute.html           refresh_0.png          tty3
apply.html      extmgr.html                  LAN_ON.png                    modules                      register.gif            tty4
apscan          extra.png                    last_line.gif                  mount                         register.gif            tty50
apscan.gif      ez-ipupdate                  last_minus.gif                 mtd                          remove.gif              tty51
arp_protection  fakedns
```

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=WldOb2J5NWll1V1YwZER4
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{WldOb2J5NWll1V1YwZER4}
```

工业固件分析

工业固件分析

70
分值

已解答

1 新起点 2 F421战队 3 SRDX

某PLC设备的固件已被攻击者提取并打包，请对固件进行分析，获取固件中被硬编码的ftp账户用户名密码信息。flag格式为：flag{ftp username+ftp password}，例如，用户名为admin，密码为123，则flag为flag{admin+123}。

CSDN @末初

```
PowerShell kali-linux
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/固件分析/工业固件分析# ls -lha
total 9.7M
drwxrwxrwx 1 1000 root 512 Oct 5 02:43 .
drwxrwxrwx 1 1000 root 512 Sep 29 10:45 ..
-rwxrwxrwx 1 1000 root 129K Sep 29 16:26 3807d31d451aaa04b042710aaa5a0a.png
-rwxrwxrwx 1 1000 root 3.2M Sep 29 10:44 9dedeb49ec7f30a87588aa6e15e10bb5dad65be4.zip
drwxrwxrwx 1 1000 root 512 Oct 5 02:43 firm
-rwxrwxrwx 1 1000 root 3.2M Apr 21 2020 firm.ldx
-rwxrwxrwx 1 1000 root 3.2M Apr 21 2020 firm.zip
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/固件分析/工业固件分析# file firm.ldx
firm.ldx: Zip archive data, at least v2.0 to extract
root@mochu7-pc:/mnt/c/Users/Administrator/Desktop/工业互联网附件/固件分析/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

此电脑 > 桌面 > 工业互联网附件 > 固件分析 > 工业固件分析 > firm > Web > wwwroot > classes

名称	修改日期	类型	大小
JL.jar	2015/4/7 22:35	Executable Jar File	13 KB
RDE.jar	2015/4/6 22:48	Executable Jar File	154 KB
rdelite.jar	2015/4/6 22:48	Executable Jar File	115 KB
SAComm.jar	2015/4/7 22:35	Executable Jar File	298 KB
SysDiag.jar	2015/4/7 22:35	Executable Jar File	595 KB
sysSetup.jar	2015/4/7 22:35	Executable Jar File	135 KB
webdiag.jar	2015/4/6 22:48	Executable Jar File	59 KB
XMLParser.jar	2015/4/3 23:10	Executable Jar File	52 KB

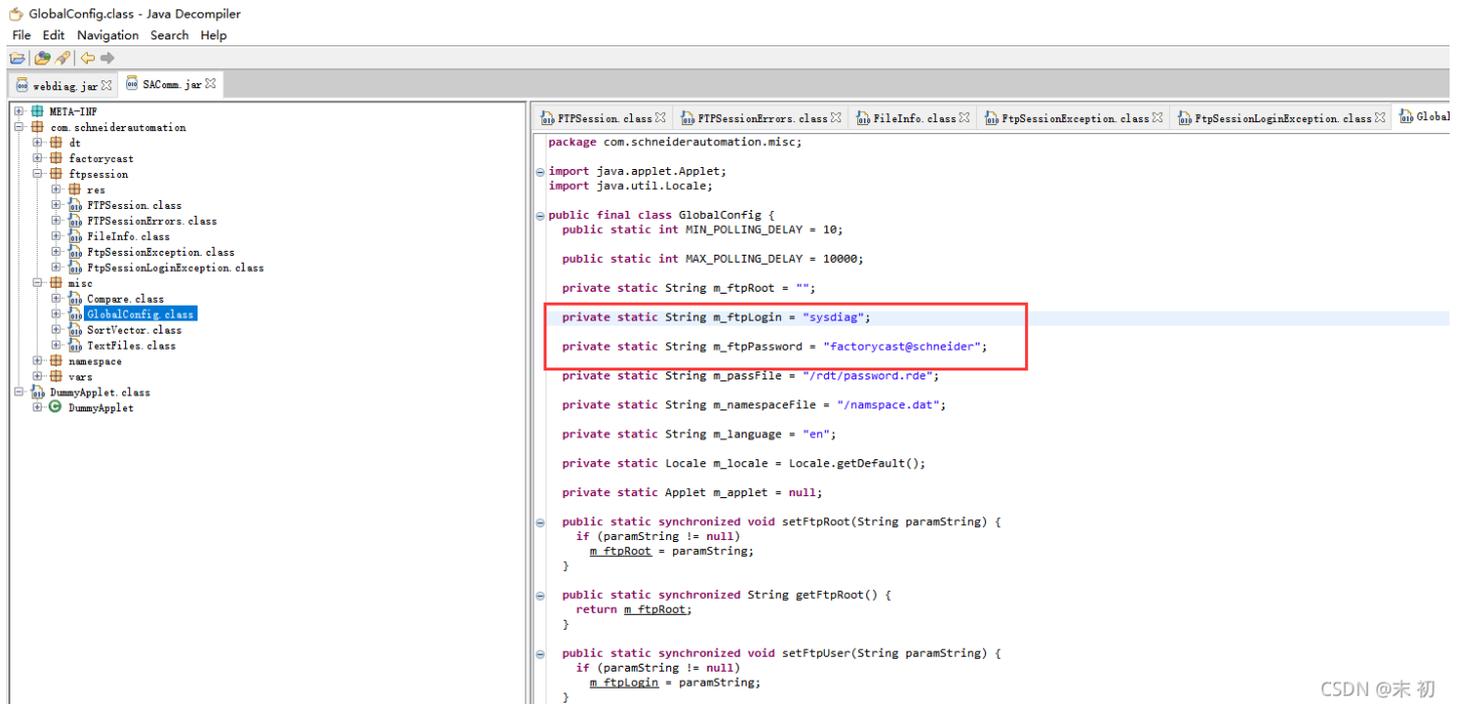
CSDN @末初

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



The screenshot shows a Java decompiler interface with the following components:

- Left Panel (Class Hierarchy):** A tree view showing the package structure. The package `com.schneiderautomation.misc` is expanded, and the class `GlobalConfig.class` is selected and highlighted in blue.
- Right Panel (Source Code):** The decompiled Java code for `GlobalConfig.class`. The code is as follows:

```
package com.schneiderautomation.misc;

import java.applet.Applet;
import java.util.Locale;

public final class GlobalConfig {
    public static int MIN_POLLING_DELAY = 10;

    public static int MAX_POLLING_DELAY = 10000;

    private static String m_ftpRoot = "";

    private static String m_ftpLogin = "sysdiag";
    private static String m_ftpPassword = "factorycast@schneider";

    private static String m_passFile = "/rdt/password.rde";

    private static String m_namespaceFile = "/namespace.dat";

    private static String m_language = "en";

    private static Locale m_locale = Locale.getDefault();

    private static Applet m_applet = null;

    public static synchronized void setFtpRoot(String paramString) {
        if (paramString != null)
            m_ftpRoot = paramString;
    }

    public static synchronized String getFtpRoot() {
        return m_ftpRoot;
    }

    public static synchronized void setFtpUser(String paramString) {
        if (paramString != null)
            m_ftpLogin = paramString;
    }
}
```

The lines `private static String m_ftpLogin = "sysdiag";` and `private static String m_ftpPassword = "factorycast@schneider";` are highlighted with a red rectangular box.
- Bottom Right:** The text "CSDN @未初" is visible.

`flag{sysdiag+factorycast@schneider}`