2021年“羊城杯”网络安全大赛部分Writeup

原创

塞纳河畔的春水 于 2021-09-13 10:47:44 发布 4210 收藏 9
分类专栏： CTF_Writeup 文章标签： 网络安全 算法 python pycharm 密码学

本文链接：https://blog.csdn.net/qq_42815161/article/details/120260053

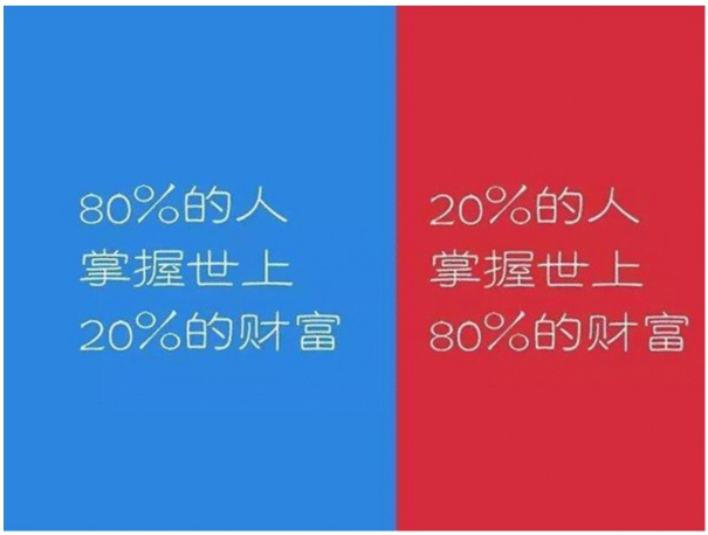
文章目录

MISC

签到

> 题目描述：猜数字01-30，数字序列以Sanfor{md5(**-**-**-**)}形式提交

附件为一张gif



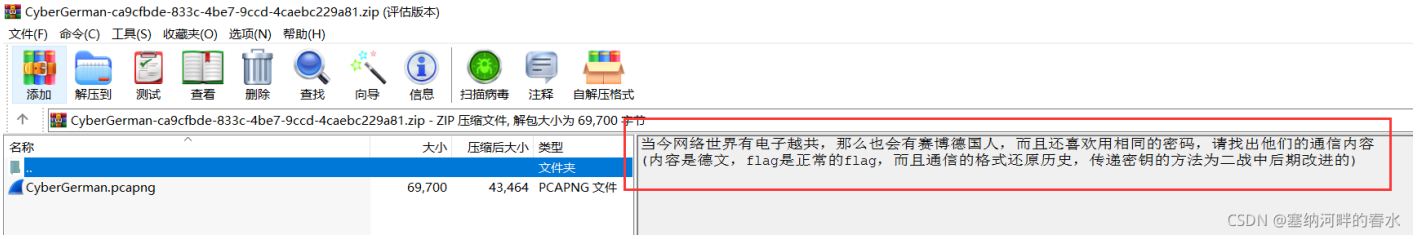
猛猜

图1 28准则
图2 8卦阵
图3 30而立之年
图4 北斗7星
图5 4大才子
图6 歼-20
图7 2只黄鹂鸣翠柳
图8 17来看流星雨
图9 23号乔丹
图10 1马当先
图11 12黄道
图12 新闻联播每晚19点首播

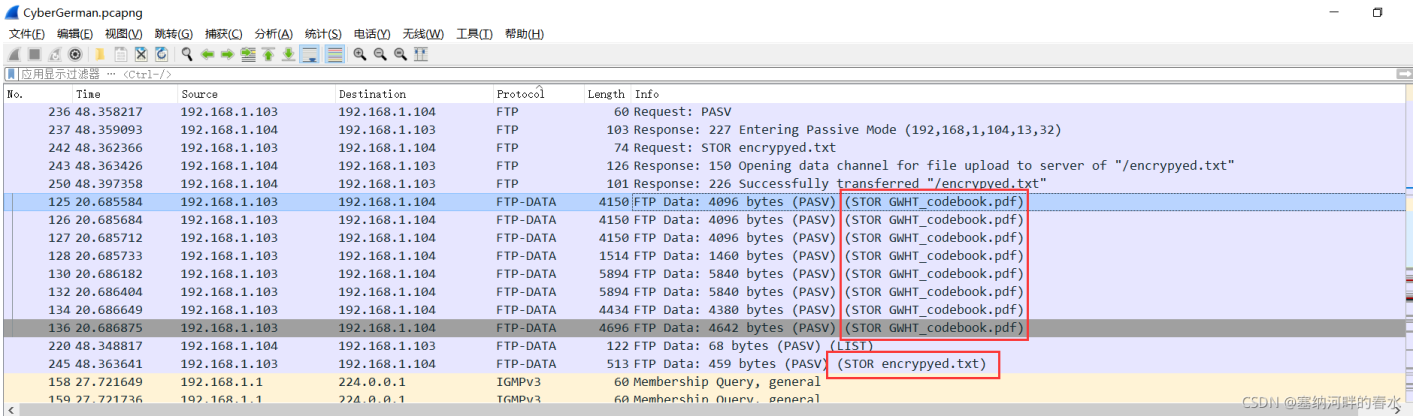
得到序列
md5(28-08-30-07-04-20-02-17-23-01-12-19)
#SangFor{d93b7da38d89c19f481e710ef1b3558b}

赛博德国人

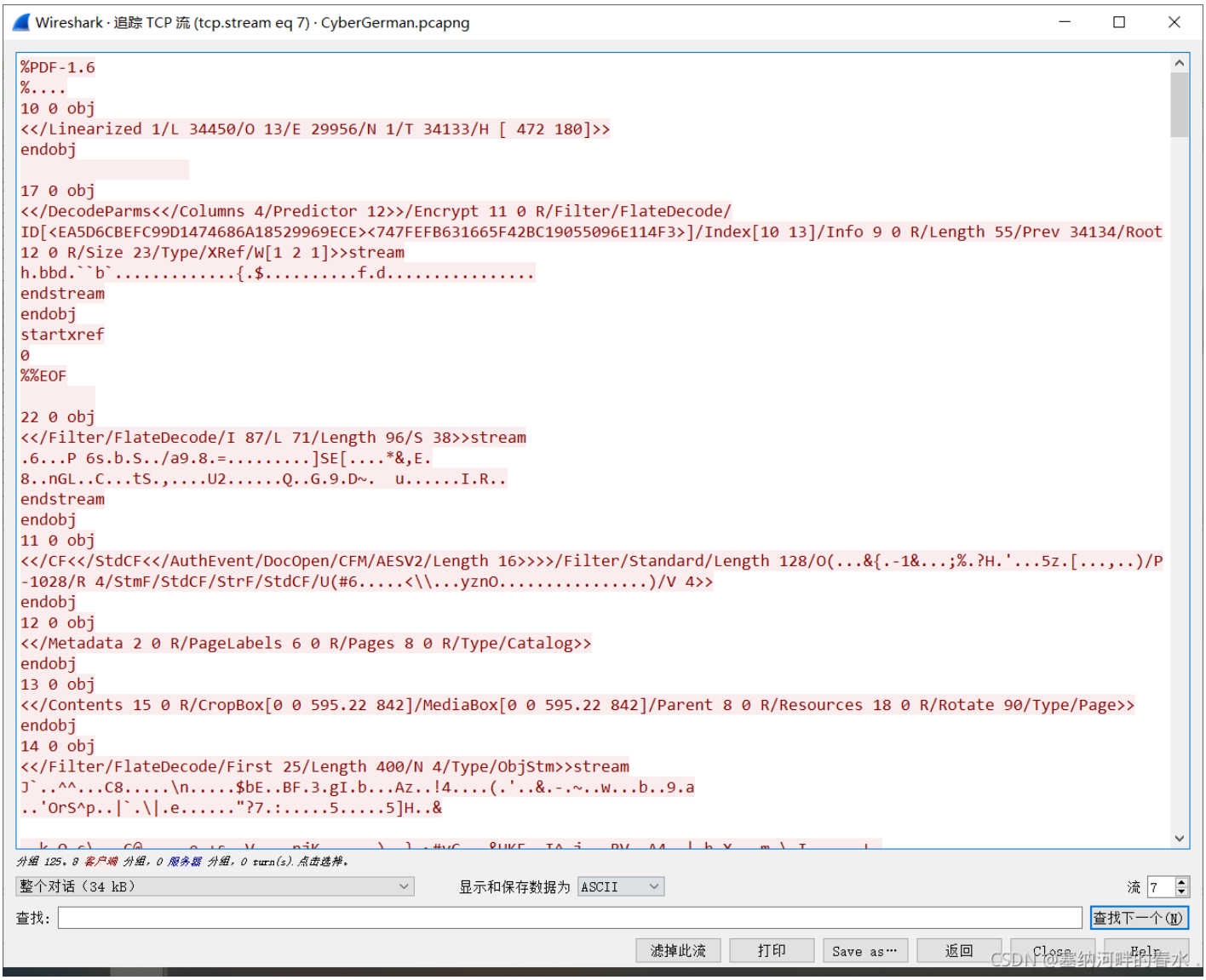
winrar打开题目附件，惊喜发现hint



解压之后为一个CyberGerman.pcapng流量包，打开之后发现里面有两文件



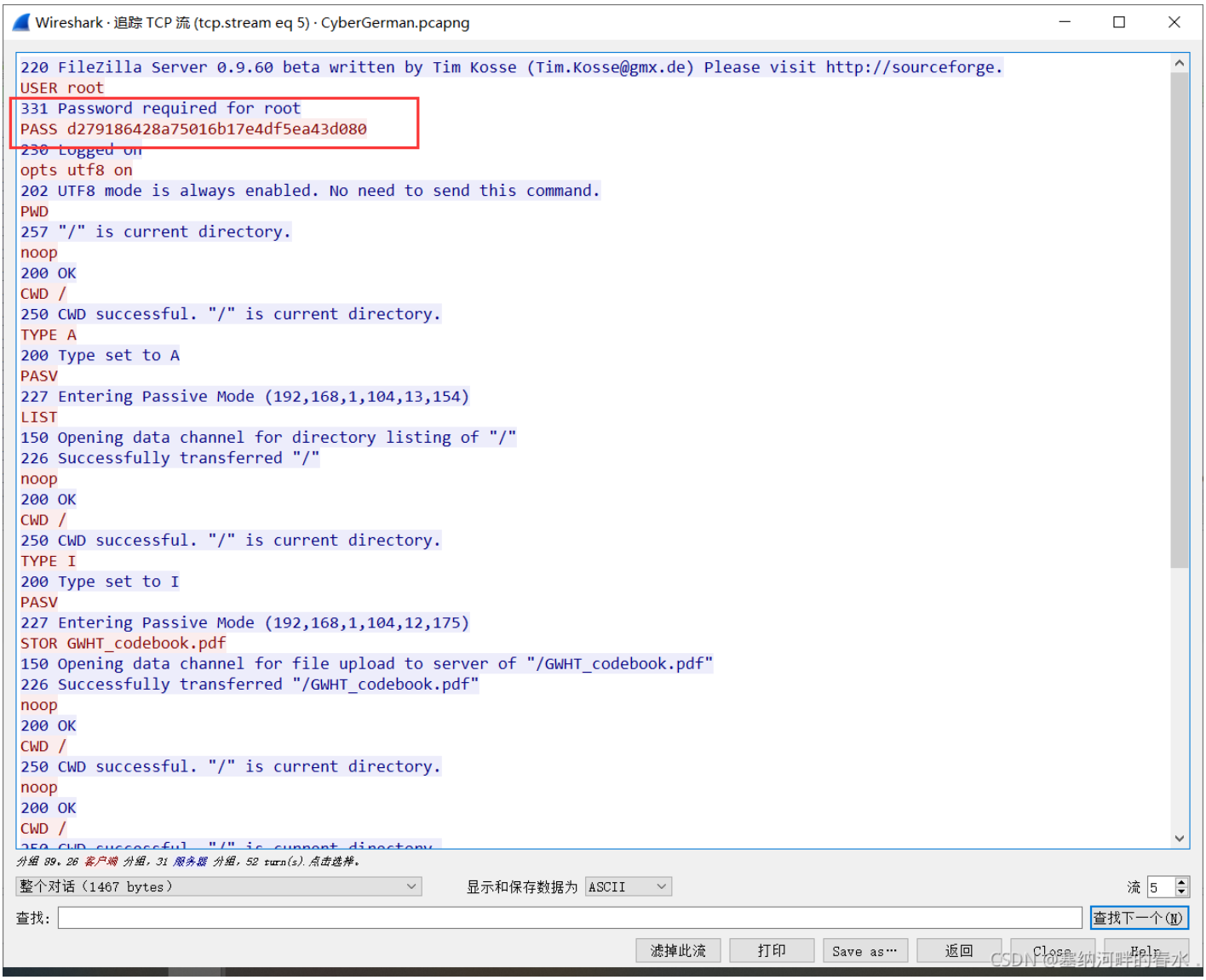
分别追踪TCP数据流导出



拿到一个codebook.pdf以及encrypyed.txt

尝试打开codebook.pdf，居然还是加密的

回去找密文，在流量包里发现疑似密文的数据



codebook.pdf输入密码后果然对了，打开发现像是一个密码表。

GEHEIM! GWHT SEPTEMBER 2021

Tag	Walzenlage	Ringstellung	Steckerverbindungen	Kenngruppen
30	III V I	02 16 17	AO CJ DZ EF HL IT MV NQ UX WY	SDQ MTP MEC RIT
29	II IV III	25 26 21	AG BL DR ES FI HY JQ KN PX TU	RHN IRX CQU PZJ
28	V IV III	25 01 19	AI CU DG EK FW HJ OQ PR TZ XY	PAX CLI TPC BOG
27	I IV II	02 23 23	AK BM CH DG FJ LR QW ST UV XZ	KNI OHK HBU FET
26	V II I	14 23 04	AW BX CH DM EF GN IO KY TU VZ	DBM DIN EIP PPH
25	V IV III	23 01 21	AV BI DL EP GK JS MX NT QU RY	CUR QQS OFG FMQ
24	V III I	15 10 24	BP CG EM FS HL JT NV OR QX UZ	FIC ZMP OGA EJE
23	III IV V	11 07 19	AS CI FU GZ HQ JW KR MO NV TY	DCU JRA UQS UXG
22	I IV III	19 05 25	AY BG CI FU HZ KS MR OT PW QV	YUV CHC BWT SOV
21	IV I V	09 12 21	AC BD FN HI JT KU OX PV RY SW	ATQ GKT PXF KMS
20	II III IV	06 24 26	AK CQ DJ EY GR HL NV OS PX TU	PGB MJY JUZ NJQ
19	V I IV	20 15 13	AD BN CY FL GU IK JP QS RT VW	TUQ RPV NSY SEU
18	III V I	07 11 15	AN BJ CG EP FM HS IR KW OU VX	ANG HRO QEN BHZ
17	IV II I	14 11 01	AU BH CK DP ET FI GN JO LZ XY	TBW WMR APU KZK
16	III II V	10 04 26	AQ CP DF EN GH IL JX KT RS VZ	TJV VJG LSA PRJ
15	III IV II	21 03 13	BT CW EV FR GN HU IS JZ KO LP	CHP XHM GPL OPZ
14	I V III	05 14 07	AY DX FP GM HQ JO LV NU RT SW	GMK QCU CZH VGY
13	II V III	14 03 07	AJ CT DH EI FY KO MX PW QS UV	VML HGV WUO QOT
12	I II III	03 12 02	AW CD EV GM IO KZ LT PS QX UY	GMF HDQ QSG SMJ
11	V IV II	08 18 10	AO BT CR DS EG HY IW MZ NQ PV	DCP IFH CVW EZV
10	II III I	05 21 25	AT BV CF EN GY HO IW LU MZ QX	SUW FGP OGA CRB
09	III II V	14 09 15	AP BC DX EK FR HN IM LQ TV WY	LFH GQQ HZO SUM
08	I V IV	21 23 05	AJ CW DH ES GL IX KQ MV NT PY	RRX QMT PEJ SSW
07	III V IV	16 02 06	AF CI DY ES GJ HR LW MO PZ UV	JYW QPG HVB CIG
06	II IV III	09 15 25	AG DZ ET FW HY JU LV MP NS OR	MCK CDA GVG ZNP
05	III V II	13 18 03	AB CV DN FS GH KP LO MT QR WX	IDX WHP NZG EUM
04	I V II	08 06 18	AM BT CG DW EZ HO IP LS NU QX	OQM KNX HNV NIY
03	I II V	02 22 24	AB CU DK EY FG IN JW LQ MV RS	UPD DHR KDH YXB
02	III II IV	02 01 24	AT BD ES FW HV IL JP KU MO RZ	BNY SVC RAN JYJ
01	III I IV	03 20 26	AL BC DM ER FZ HU JO KW NS QT	LYH UEB GKF NJZ

codebook.pdf

查看encrypyed.txt

```
0911 = 1tle = 1tl = 350(长度) = RZS NAJ（PKS） =
nkfgp roqad boprv yrdhy zwamf qsrhb owqvt jzotr ffcjq snpqh kpwzm fprru gufez xsuws aohyw xbreu pifbz kagxj
```

又是一堆看不懂的东西，这时候就要上百度谷歌了，查阅资料

> 二战德军 Enigma 密码机原理演示与破解
>
> Military Use of the Enigma
>
> Virtual 3 wheel Army/Air Force Enigma by Tony Sale
>
> [ENiGMA] - a pictured step-by-step-howto about encryption

得知加密为恩尼格玛密码机。

> 在密码学史中，**恩尼格玛密码机**（德语：Enigma，又译**哑谜机**，或**“谜”式密码机**）是一种用于加密与解密文件的密码机。确切地说，恩尼格玛是对二战时期纳粹德国使用的一系列相似的转子机械加解密机器的统称，它包括了许多不同的型号，为密码学对称加密算法的流加密。

开始破译

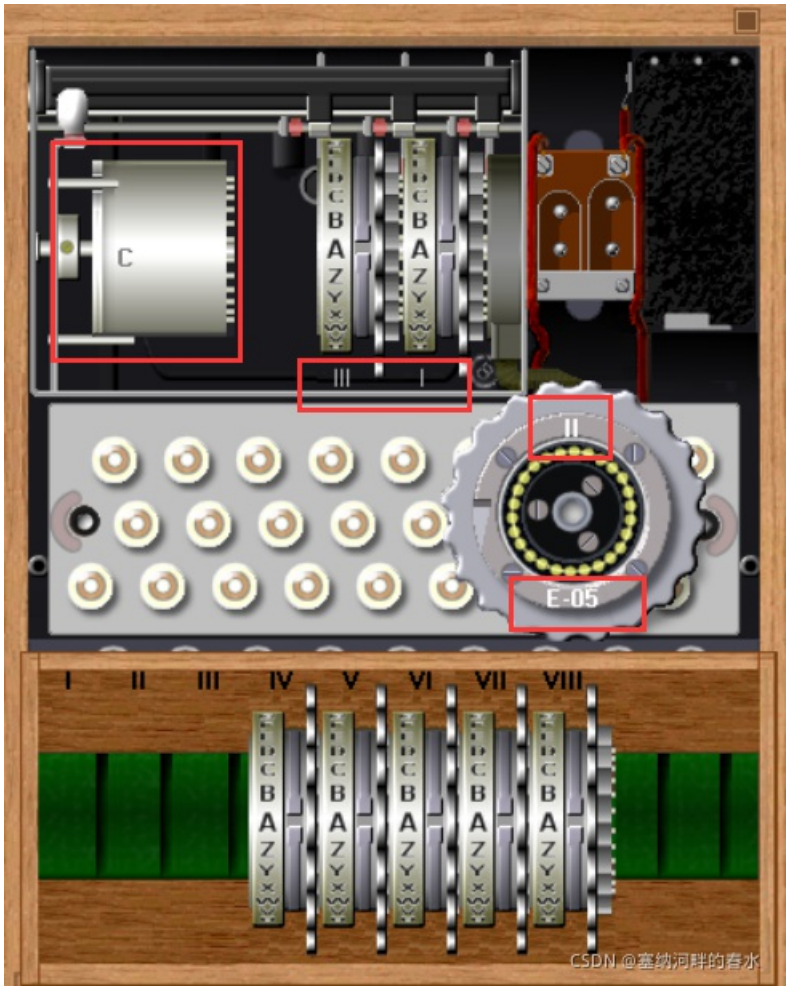
```
#encrypted.txt
0911 = 1tle = 1tl = 350 = RZS NAJ=
nkfgp roqad boprv yrdhy zwamf qsrhb owqvt jzotr ffcjq snpqh kpwzm fprru gufez xsuws aohyw xbreu pifbz kagxj

对于encrypted.txt结构分析
0911 猜测为接受到信息的日期
1tle 为发送方姓名
1tl 为接收方姓名
350 为密文长度为350位
RZS NAJ 为加密转子起始位置解密信号

首先确认信息发送的日期
密码段的前五位为标志位，通常为设定的三个字母加上任意的两个字母组成
前五个数据为nkfgp，去codebook.pdf中寻找

|Tag | Walzenlage |Ringstellung|      Steckerverbindungen       |   Kenngruppen   |
| 10 | II III I   | 05 21 25   | AT BV CF EN GY HO IW LU MZ QX | SUW FGP OGA CRB |

确认到10号的Kenngruppen一栏中含有FGP，锁定信息发送的日期，开始破译。
```

祭出模拟软件。

Enigma模拟软件

继续破译

```
|Tag | Walzenlage |Ringstellung|      Steckerverbindungen      |   Kenngruppen   |
| 10 | II III I   | 05 21 25   | AT BV CF EN GY HO IW LU MZ QX | SUW FGP OGA CRB |

调整密码机模式Kriegsmarine M3 - UKW = C（我也不知道为什么，乱试出来的）
选取转子序号II III I
将三个转子分别调整到05 21 25对应'E' 'U' 'Y'
随后按照Steckerverbindungen对字母进行连线
```



内部转子设置

按照Steckerverbindungen对字母进行连线

按照Steckerverbindungen设置

根据encrypted.txt的第一行信息 = RZS NAJ =确认破译转子起始位置，确认方法：配置好设备后将表面三个转盘调整至RZS，随后往密码机中键入NAJ（直接点模拟机中的键盘也可），密码机输出PKS，其中PKS代表译文转子起始位置。

随后将表面三个转盘调整至PKS就可以开始快乐打印了，记得把前五个标志位nkfgp去除

```
#打印得到信息
VIERSIEBENFUENFSIEBENVIERACHTFUENFVIERSIEBENBERTADREISECHSSECHSZWEIDREINEUNDREISECHSDREISIEBENDREIZWEIDREIN
```

猜测为德语，直接百度德语数字

个位数：

0: null	1: eins	2: zwei	3: drei	4: vier
5: fünf	6: sechs	7: sieben	8: acht	9: neun

使用Notepad++文本替换

```
475748547BERTA3662393637323035666537353739356631303438353731636634636662373033 7DORA
```

其中BERTA为b，DORA为d，显而易见16进制转文本

进制转文本 | 16进制转文本 | 二进制转换 | 四进制转换 | 八进制转换 | 十进制转换 | 十六进制转换 | 三十二进制转换 | 六十四进制转换

16进制转换文本 / 文本转16进制

475748547b3662393637323035666537353739356631303438353731636634366623730337d

字符串转16进制 >>

16进制转字符串 >>

结果互换

全部清空

GWHT{6b967205fe75795f1048571cf4cfb703}

CSDN @塞纳河畔的春水

MISC520

题目描述：有一天，zip爱上了pcap，zip为了能与pcap创造更多机会，不断地将自己的能力表现出来。可是，LSBSteg却突然杀了出来，将pcap吞并于png中，不放出来
。zip看到了png，多喝热水少做梦。zip异常的愤怒，不断地用自己的能力去报复png，不让png逃走。至今，zip仍未释怀。

好家伙，一看就是套娃题，什么zip、pcap、lsb、png...一个个来

首先给了一个压缩包520.zip，解压后得到519.zip和story，story打开和题目描述一样，519.zip解压后是518.zip和一样的story（套娃）。

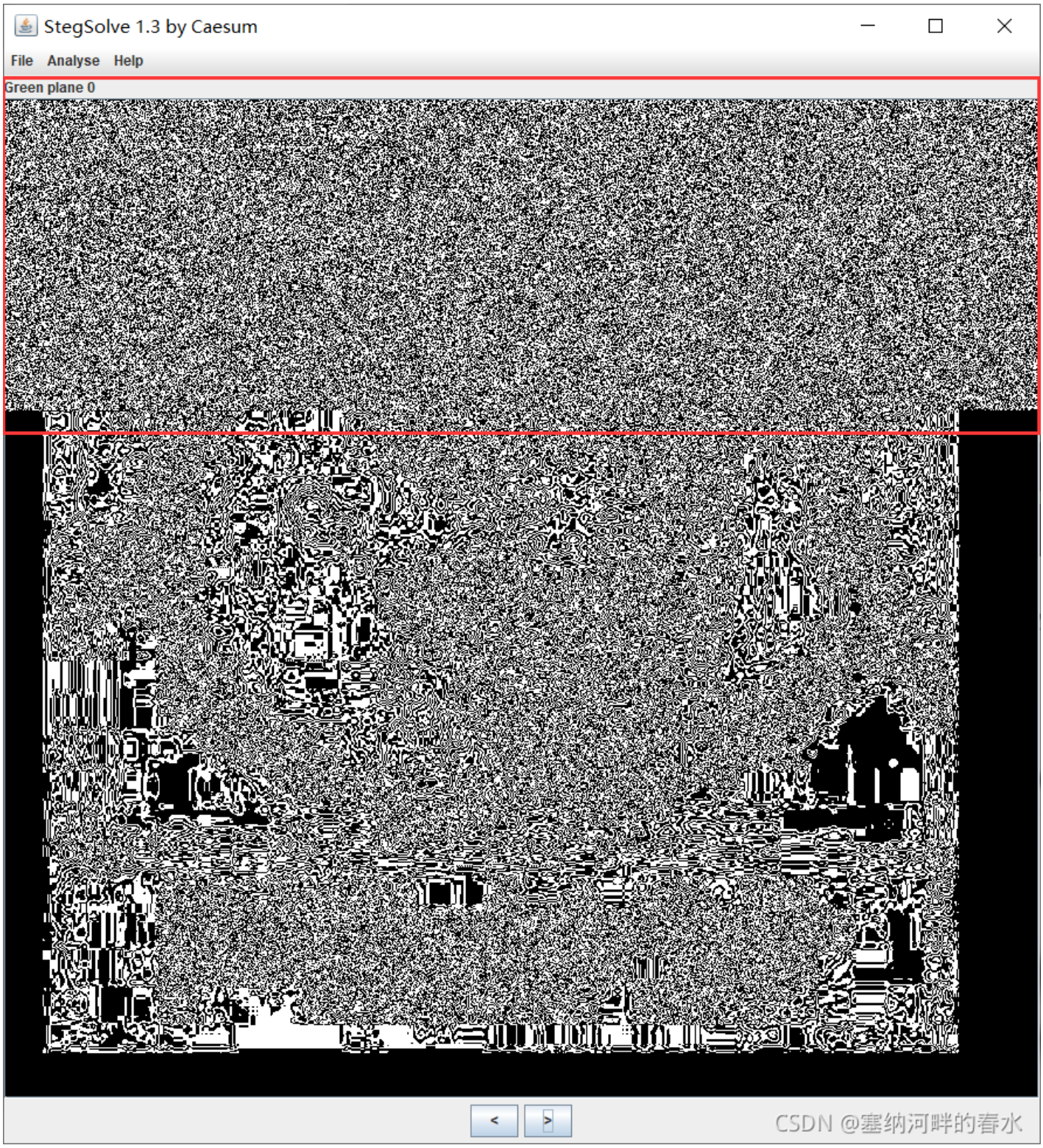
写脚本解压

```
import zipfile

# 解压文件夹
file_input = 'D:/2021.9羊城杯/misc520-2fee7e6e-92ea-46f9-b382-e8ca5e7534ba/1/'
# 解压目标文件夹
file_output = 'D:/2021.9羊城杯/misc520-2fee7e6e-92ea-46f9-b382-e8ca5e7534ba/1/'


def zip_file(zip_name):
    # print(zip_name)
    r = zipfile.is_zipfile(zip_name)
    if r:
        fz = zipfile.ZipFile(zip_name, 'r')
        for file in fz.namelist():
            fz.extract(file, file_output)


for i in range(519, 0, -1):
    zip_file(file_input + "{}.zip".format(i))
    story = open(file_input + "story", 'r', encoding='UTF-8').read()
    if 'png' in story:
        pass
    else:
        print(story)

#print
"""
这都被你发现了？
我这故事不错吧，嘻嘻嘻
那就把flag给你吧
oh，不，还有一半藏在了pcap的心里，快去找找吧
左心房右心房，扑通扑通的心，咿呀咿呀的❤
72, 89, 75, 88, 128, 93, 58, 116, 76, 121, 120, 63, 108,
"""
```

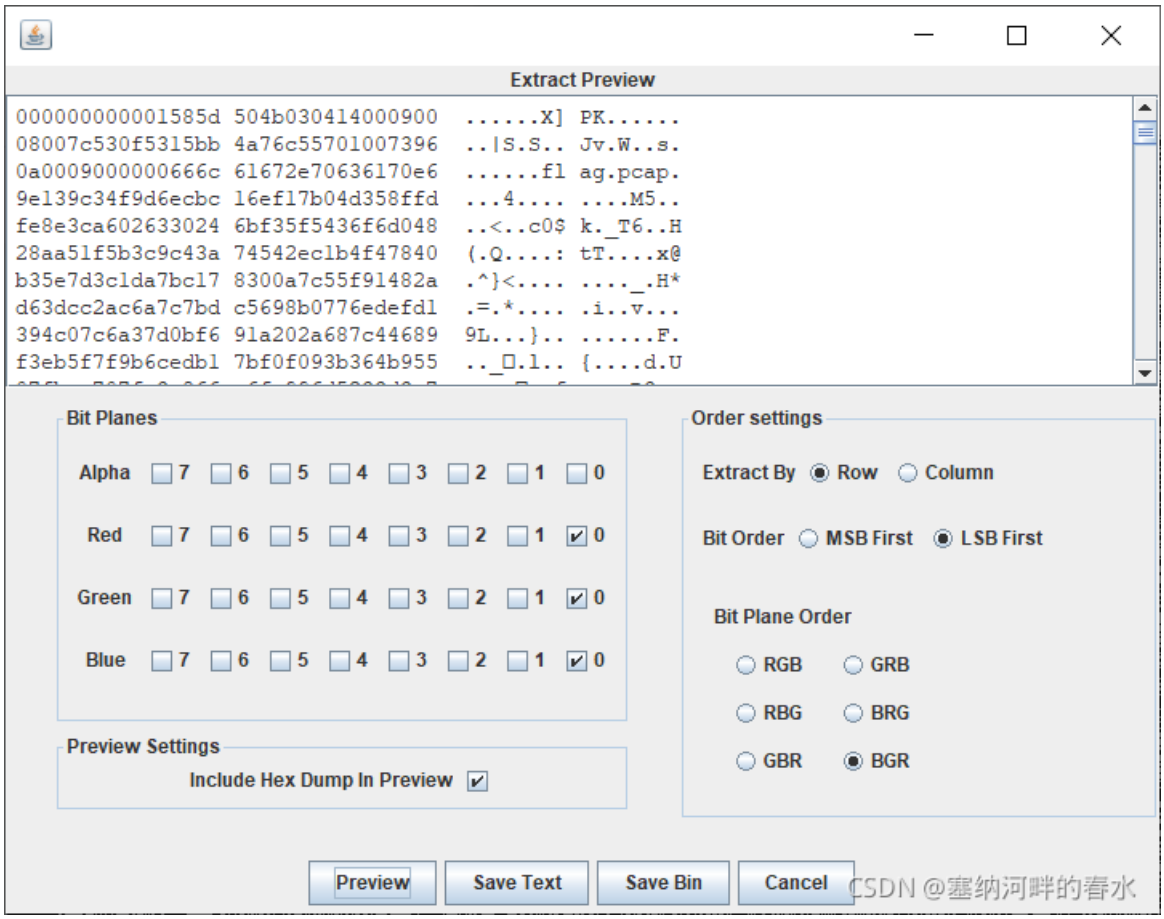
拿到一串数字，提示去找另一串，解压完所有文件后拿到张图flag.png，打开Stegsolve

flag.png

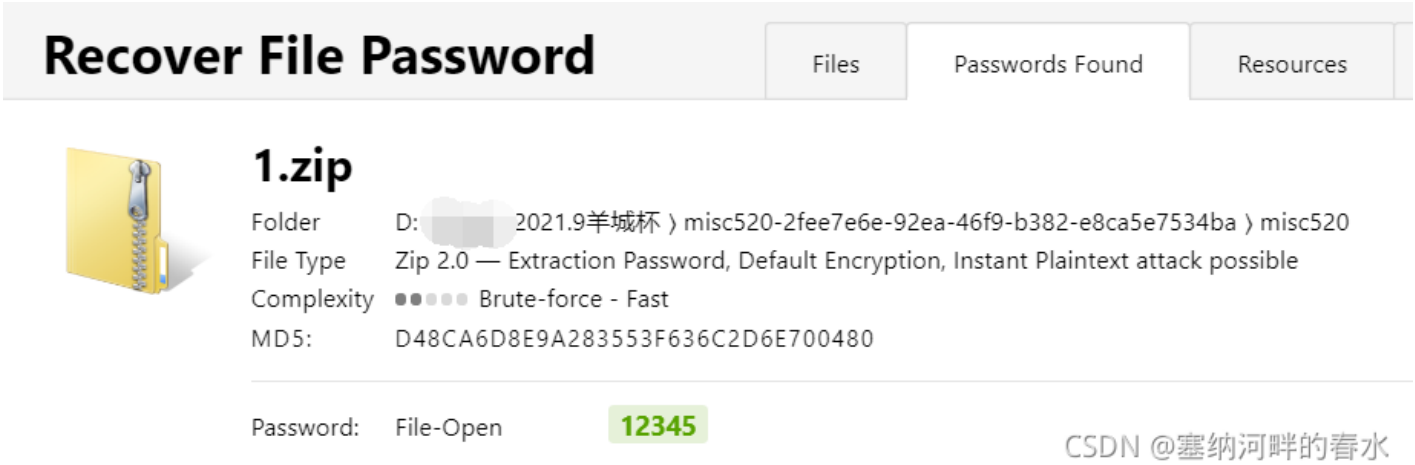
0通道有东西



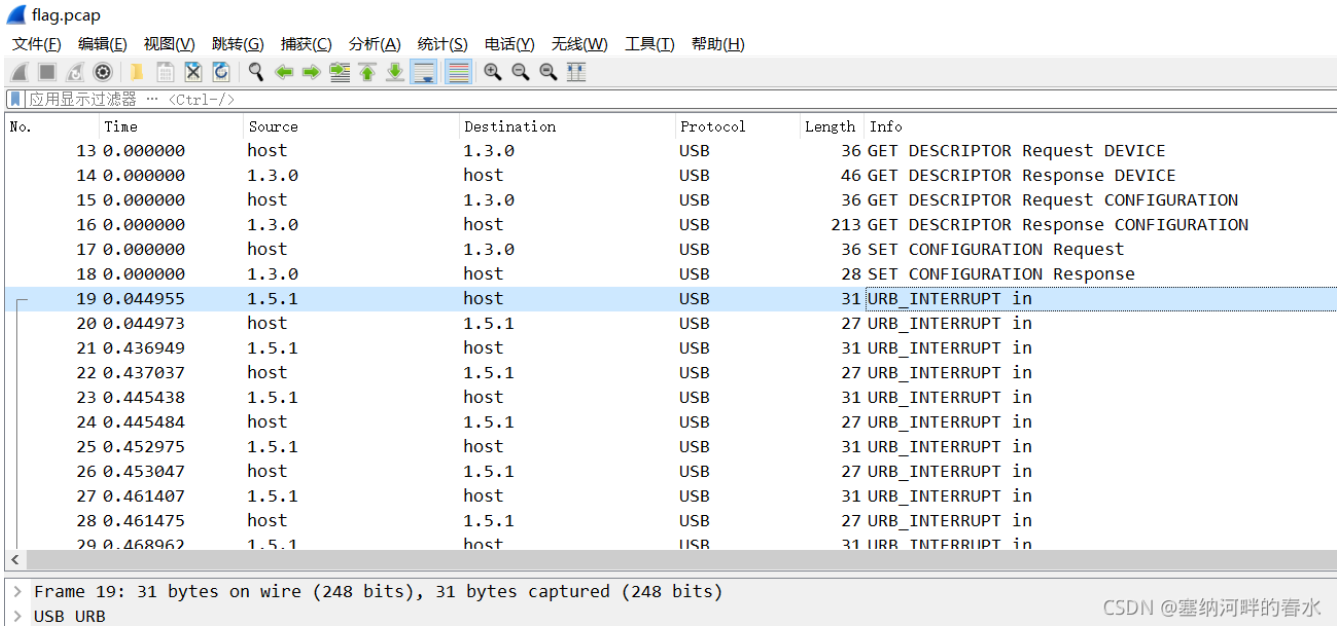
Data Extract尝试后找到PK头



导出压缩包数据，发现是个加密的压缩包，里面有flag.pcap，没找到密码尝试弱密码爆破。



打开flag.pcap，认定为USB流量隐写，Leftover Capture Data为4字节，锁定鼠标流量。



flag.pcap

使用tshark命令把该流量分析包输出为一个名为usbdata的文本

```
tshark -r flag.pcap -T fields -e usb.capdata  >  usbdata.txt
```

祭脚本

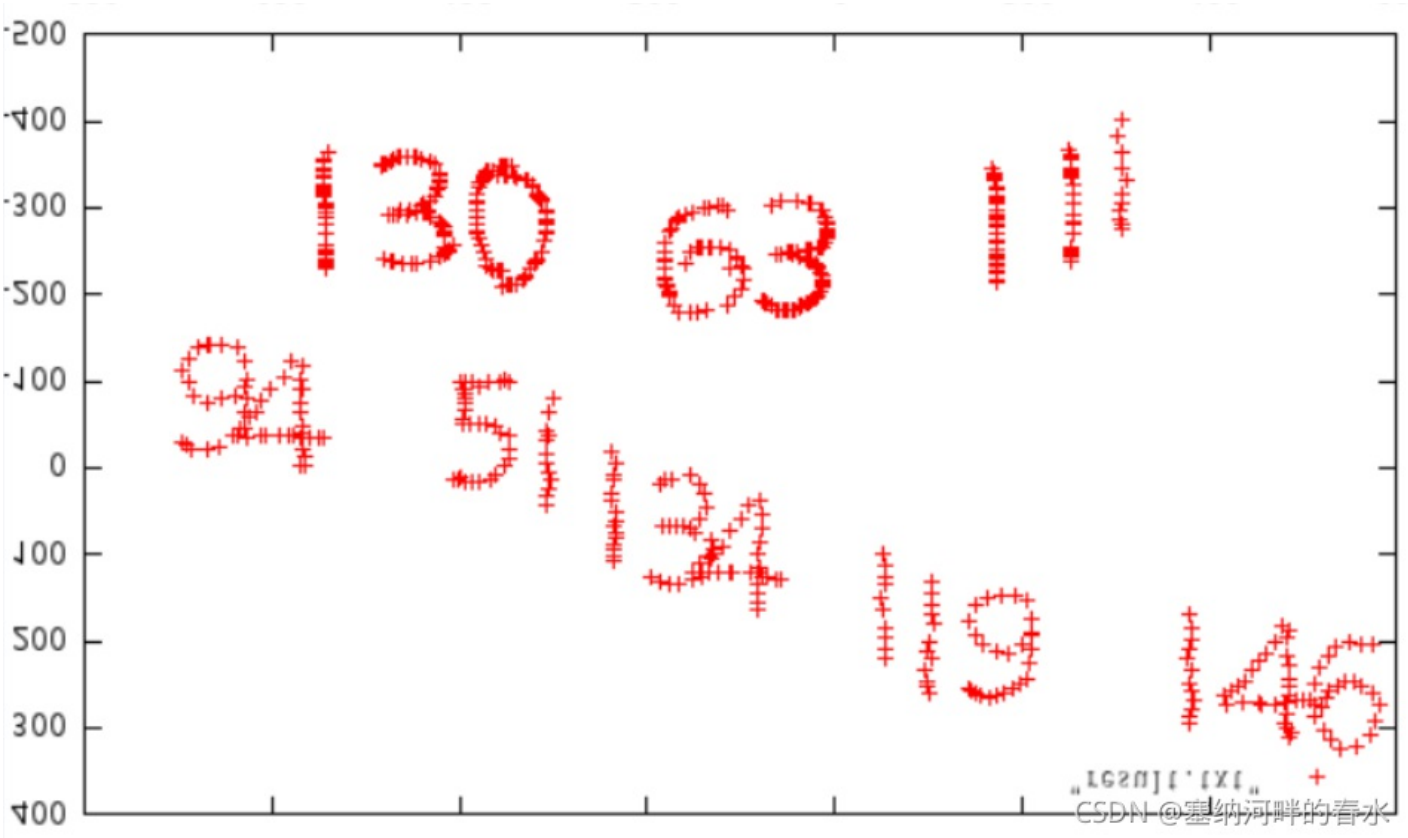
```
#将数据0005fa00转化为00:05:fa:00四字节格式
#第一字节为00代表鼠标按键未按下，01代表鼠标左键按下，02代表鼠标右键按下
#第二字节为鼠标水平移动距离
#第三字节为鼠标垂直移动距离

while 1:
    a=f.readline().strip()
    if 1:
        if len(a)==8: # 鼠标流量的话len改为8
            out=''
            for i in range(0,len(a),2):
                if i+2 != len(a):
                    out+=a[i]+a[i+1]+":"
                else:
                    out+=a[i]+a[i+1]
            fi.write(out)
            fi.write('\n')
    else:
        break

fi.close()
```

将16进制的usbdata.txt转为坐标

```
nums = []
keys = open('usbdata.txt','r')
f = open('xy.txt','w')
posx = 0
posy = 0
for line in keys:
    if len(line) != 12 :
        continue
    x = int(line[3:5],16)
    y = int(line[6:8],16)
    if x > 127 :
        x -= 256
    if y > 127 :
        y -= 256
    posx += x
    posy += y
    btn_flag = int(line[0:2],16)  # 1 for left , 2 for right , 0 for nothing

    if btn_flag != 0 : # 输出鼠标有按键按下后的坐标
        f.write(str(posx))
        f.write(' ')
        f.write(str(posy))
        f.write('\n')

f.close()
```

上gnuplot，把xy.txt文本里的坐标转化为图片

```
gnuplot.exe plot "xy.txt"
```



最后拿到数字串

```
72, 89, 75, 88, 128, 93, 58, 116, 76, 121, 120, 63, 108, 130, 63, 111, 94, 51, 134, 119, 146
```

好了，开始和出题人对脑电波，题目flag格式GWHT{}，对比猜测为ASCII码加密，每位减去相应位数。如72为第一位chr(72-1)='G'，89为第二位chr(89-2)='W'等等。

```
a = [72, 89, 75, 88, 128, 93, 58, 116, 76, 121, 120, 63, 108, 130, 63, 111, 94, 51, 134, 119, 146]
flag = ''
for i in range(len(a)):
    flag = flag + chr(a[i] - i - 1)
print(flag)

#GWHT{W3lCom3_t0_M!sc}
```

Baby_Forenisc

内存取证题目，题目给了一个raw文件，上kali打开volatility常规检测。

```
python vol.py -f 1.raw --profile=WinXPSP2x86 cmdscan

Volatility Foundation Volatility Framework 2.6.1
CommandProcess: csrss.exe Pid: 580
CommandHistory: 0x566bb8 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 5 LastAdded: 4 LastDisplayed: 4
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x4cc
Cmd #0 @ 0x3689ed8: git push -u origin master
Cmd #1 @ 0x566148: ok....
Cmd #2 @ 0x56aa08: then delete .git and flagfile
Cmd #3 @ 0x368a798: You can never find my account
Cmd #4 @ 0x56a580: hahaha!
```

提示东西在git上，找找能不能找到涉及到git具体仓库的信息。

```
python vol.py -f 1.raw --profile=WinXPSP2x86 filescan|grep "txt"

Volatility Foundation Volatility Framework 2.6.1
0x00000000020bf6a0      1      0 RW-r-- \Device\HarddiskVolume1\Documents and Settings\Owner\桌面\ssh.txt
0x00000000021c01b0      1      0 R--rwd \Device\HarddiskVolume1\Documents and Settings\All Users\ssh.txt
0x000000000231d6b0      4      2 -W-rw- \Device\HarddiskVolume1\Documents and Settings\All Users\Applicatio
```

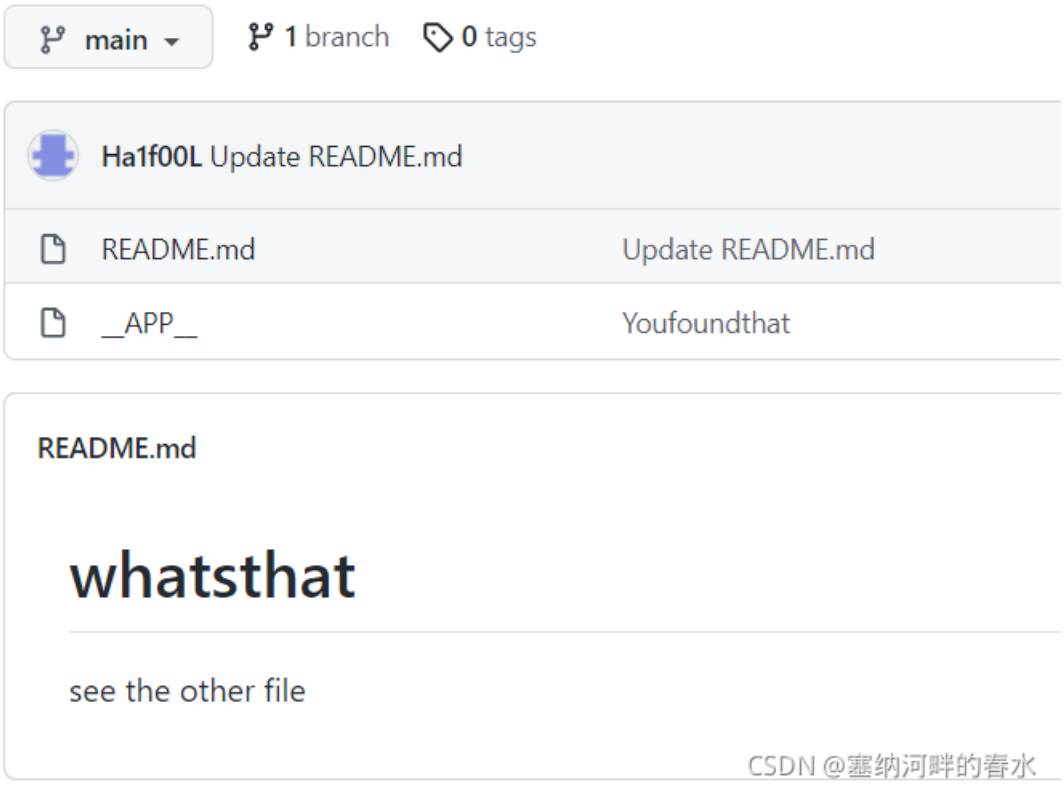
找ssh.txt文件，dump出来改文件后缀为txt。

```
python vol.py -f 1.raw --profile=WinXPSP2x86 dumpfiles -Q 0x00000000020bf6a0 -D ./
```

```
#ssh.txt
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAmw8eqi/h23ABuRhhmx83LuRhw6m8C8k76Me0s7MNdvDP2ZB5hJUU
fZ4HxR5sEoQf6NyIcCDeznb8FAYAktm3cBlgof847aL661F0R5FtIfOJC/MwklRmXjYr46
6HNjQ0Ouu12znqBPJAaMkAaZXknqlEAxCRvyOQhg0bPSR3xxCM39TxpXRKd3tzhlBUQHZi
upgt6CF3TkBuIcKUPgZ7OgJ/7ES3FaiUOlpZdUYf/H3VwwQumuXPPwvT5QdRA9Myv/zbee
R9ddLJL84raHK6unuHjngGvWjhXUUQulta49HH55pyrFUViIvH1tfns/6BglTrYWRlFX3A
TNOVy2igHkhZI8M9GK5VUBwEo3kXcWRiK85vAWwmddBd9+c0NERahRg+SNbodsd1JFu0C9
kqJ8/HlOnDfPBsUpD0EY/EbzW5PKbkksp2Vp3z+S0y1aVpX2EJRhq2S5kEEU+V4LLN6uqu
CJzVLeG5Lpnn4V/Ekf/ZpJmmk1Pp9KGFw3tlOqTLAAAFkNMuPgLTLj4CAAAAB3NzaC1yc2
EAAAGBAJsPHqov4dtwAbkYYZsfNy7kYcOpvAvJO+jHtLOzDXbwz9mQeYSVFH2eB8UebBKE
H+jciHAg3s52/BQGAJLZt3AZYKH/OO2i+utRdEeRbSHziQvzMJJUZl42K+OuhzY0NDrrtd
s56gTyQGjJAGmV5J6pRAMQkb8jkIYNGz0kd8cQjN/U8aV0Snd7c4ZQVEB2YrqYLeghd05A
biHClD4GezoCf+xEtxWolDpaWXVGH/x91cMELprlzz8L0+UHUQPTMr/823nkfXXSyS/OK2
hyurp7h454Br1o4V1FELpbWuPRx+eacqxVFYiLx9bX57P+gYJU62FkZRV9wEzTlctooB5I
WSPDPRiuVVAcBKN5F3FkYivObwFsJnXQXffnNDREWoUYPkjW6HbHdSRbtAvZKifPx5Tpw3
zwbFKQ9BGPxG81uTym5JLKdlad8/ktMtWlaV9hCUYatkuZBBFPleCyzerqrgic1S3huS6Z
5+FfxJH/2aSZppNT6fShhcN7ZTqkywAAAAMBAAEAAAGAdfojEsorxpKKPRLX8PbnPb52xD
C46x7Jfmu0iaWKcRz4iEjsrHvhg1JiBxEGmW/992cUSHw6Ck1trq6CcTlF4PzuEVPnNKf0
0ma/WlTD/DkX5Qe7xRqCaNw+uJVqO0utEceWLp7595l6eD+3GJ77u9x96vcIba3ZoKUIPJ
UqrUNibEvRMFoy7oX3eBJWiFWK+P4gr6YG6HsNUJKDyE2WJKUSP+pogwoo/d0Qg7I/VBVK
N39PFnwUG5wcNP5EHezqWQVVln/dltDgOc5IldknTRt4Q3NDrSyNsRpv0EYI2gz+yRu/IE
RR9PHYjH5l6uYwowW34iGi/xloSxG5bDEWOe0eEANCjowiYYrmTLffIQ/AU9w4te/+eWd2
WV56LUuC6k4mEdNhtljMZR/0A+C5EkPzgsTEJEmYLYvqrNejM7Y1UKz3+YZ8m8rT4XcNmf
j5wfJd1TbCu0hB5kZC1DkybYQaMRNnZ3+PjwU2hZBTuh02F787nG5NFkpI96qkWxTBAAAA
wBdaxLNzl/7Dig/neTUAQLa/C1F2cpQt6RcJbzHodgxm8n75a/wdRI4/oCvGJkRgyAnyCE
tgfMnTQ4opmHf5k0U0R/wmCGivcGhg5KIBSSnp9mWt6qclJ8O6vZ5L3rKIgreWzGUDk8IT
W3Lcl5EO0sskpVvp65xncEdv3CefxXVTlkgp4PXgXcxPao633hWA6TAm2zZx7R6fJt0Ex4
x3lVG68ghRE/ZFbF48s8Gy+zRDyA5JEGPWxWddO623IVgG6AAAAMEAyX4CJKSxE5gvJdrw
lhx8dBbVQxw06fPoVlu/z/JTwkPdliuAdp30SV8WbmXUhLvv457WdqAMCwlGs/7xrCW21U
84+VeD9aGM61nSsT7kUzGjdvbjQiHCmys7dwuy/thCrpWFTxI4fjOEYHc3N8S+hBHQRJKk
mEYyBoI3eJ3NhUsGHr1V4LONBKkoUZyC+LjKev06m9qM6R0/0k4cB09pkDVinuFuGk5iDy
YKyjAGiAxFI9ACiZ5NLKTsdaEqtCPfAAAAwQDFAXbSxwbLYWDacBNUm4E7FZsYKkqoIAWQ
3uEQP5Sp7GrCU5dWraGB2wOkX+irMYGDfTk5qG8NLyYoSKVIZwA6ijDliWekL6XdPGJfKK
7xw64Nx6syc7oD7scSzTGNH0m1z+T2rjP3dMDDVhYMHksYcSxikyHNzLR9Z51hCOHeKb1O
8LNW4IrC6AYeXt8sHizSLIagncOuPtSkKiGdR5fn65fHomMzaVQsSJYvwNeSrKXu36NSJm
27AuL6DDE2vJUAAAAUc29uZzU1MjA4NTEwN0BxcS5jb20BAgMEBQYH
```

base64解密，看到最后包含一个邮箱song552085107@qq.com，接下来直接上Github。

```
openssh-key-v1nonenone�ssh-rsa���/��p�a�7.�aé��;�Ǵ��
v��◇y��}��l��·p ��v��□p`��8����QtG�m!��0�Tf^6+桥644:��9��@h�i�趧D��#��
=$w������tJw{s�PT@vb��-�!wN@n!>{:�D���:ZYuF�}��.���?��Q�2���y�}u��/�+hr��{��x�h�]E
v��◇y��}��l��·p ��v��□p`��8����QtG�m!��0�Tf^6+桥644:��9��@h�i�趧D��#��
=$w������tJw{s�PT@vb��-�!wN@n!>{:�D���:ZYuF�}��.���?��Q�2���y�}u��/�+hr��{��x�h�]E
Mm���q9E���S�4���f�ZT��9����h�>��j;K�ũ.����zx?�����}��m·.<*CblK0Z2xV U
P���M$?h
(����?TJ7O|4D YU9□H'MxCsC,o F DQ ~e0!hK l19B"a L
]6»HA FB92m1gg66S6;nMJH[Zijs Ê y5@Qvr- o1vf~k D?+&D`
0E#M,u*#v a`pT;*J ?j S V_1}99o
/&(H Hg:0g/<b_(î
ǫ2s1I3t VKq,b!g a)N V^,,,,îy*!G Æc3iT,H/ ■ ɹR&m
```



Notepad++打开_APP_，开始翻文件找找找，找到熟悉的。

```
if( __WXML_GLOBAL__.ops_cached.$gwx_1)return __WXML_GLOBAL__.ops_cached.$gwx_1
__WXML_GLOBAL__.ops_cached.$gwx_1=[];
(function(z){var a=11;function Z(ops,debugLine){z.push(['11182016',ops,debugLine])}
Z([3,'这是地图组件测试'],['./pages/index/index.wxml',2,7])
Z([3,'地图'],['./pages/index/index.wxml',4,7])
Z([3,'这也是测试'],['./pages/index/index.wxml',7,7])
Z([3,'什么，你想要f\x5cl\x5ca\x5cg?'],['./pages/index/index.wxml',9,7])
Z([3,'U2FuZ0ZvcntTMF8zYXp5XzJfY3JhY2tfbm9vYl9wbGF5ZXJ9'],['./pages/index/index.wxml',10,7])
})(__WXML_GLOBAL__.ops_cached.$gwx_1);return __WXML_GLOBAL__.ops_cached.$gwx_1
}
```


继续Base64解密

```
U2FuZ0ZvcntTMF8zYXp5XzJfY3JhY2tfbm9vYl9wbGF5ZXJ9
#base64:SangFor{S0_3azy_2_crack_noob_player}
```

Crypto

Bigrsa

题目：

```
from Crypto.Util.number import *
from flag import *

n1 = 10383529640908175186077053551474658681539589842726033432568031364836913266105784068082329551223694895 3
n2 = 11538319858467714748755601433644831072185384116875801244563418281418031448050182892716007101519708945 6
e = 65537
m = bytes_to_long(flag)
c = pow(m, e, n1)
c = pow(c, e, n2)

print("c = %d" % c)

# output
# c = 6040616830276886080421122005570855181623881606177246455795698569940078216359725186167596790924618783 3
```

尝试发现n1与n2存在公因数，分别计算分解出各自pq，常规RSA解密。

```
import binascii
import gmpy2

n1 = 10383529640908175186077053551474658681539589842726033432568031364836913266105784068082329551223694895 3
n2 = 11538319858467714748755601433644831072185384116875801244563418281418031448050182892716007101519708945 6
e = 65537
c = 60406168302768860804211220055708551816238816061772464557956985699400782163597251861675967909246187833 32
p = gmpy2.gcd(n1, n2)
q1 = n1 // p
q2 = n2 // p
phi1 = (p - 1) * (q1 - 1)
phi2 = (p - 1) * (q2 - 1)

d1 = gmpy2.invert(e, phi1)
d2 = gmpy2.invert(e, phi2)

m = pow(pow(c, d2, n2), d1, n1)
print(binascii.unhexlify(hex(m)[2:].strip("L")))

#SangFor{qSccmm1WrgvIg2Uq_cZhmqNfEGTz2GV8}
```

Ring Ring Ring

VPN连接服务器（这vpn卡了半天进进不去），后过hash认证。要求输入100组abcde满足

$$a^4+b^4+c^4+d^4=e^2$$

乍一看，abcde没限制条件，妙哇，直接爆破，令a=b=c=d，则

$$2a^2=e$$

上exp

```
from pwn import *
import string
from hashlib import *
context.log_level='debug'
io=remote('ip',port)
str1=string.digits+string.ascii_letters
io.recvuntil('Please find a string that md5(str + ')
end=io.recvuntil(')')[:-1].decode()
io.recvuntil(' == ')
sha=io.recv(5).decode()
print(end,sha)
def pow(end,sha):
    for i in str1:
        for j in str1:
            for k in str1:
                for l in str1:
                    str2=(i+j+k+l+end).encode()
                    if md5(str2).hexdigest()[:5]==sha:
                        return i+j+k+l
v=pow(end,sha)
io.recvuntil('[>] Give me xxxxx:')
io.sendline(v)
for i in range(1,101):
    io.recvuntil('[>] a:')
    io.sendline(str(i))
    io.recvuntil('[>] b:')
    io.sendline(str(i))
    io.recvuntil('[>] c:')
    io.sendline(str(i))
    io.recvuntil('[>] d:')
    io.sendline(str(i))
    io.recvuntil('[>] e:')
    io.sendline(str(2*i*i))
io.recvall()
```

> flag：GWHT{a_funny_equation}