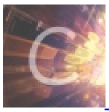
蓝帽杯2021初赛 writeup+赛后复现（misc123+pwn2+web1）

是Mumuzi 于 2021-04-30 14:33:50 发布 3118 收藏 16

分类专栏： ctf 文章标签： python 信息安全

本文链接：https://blog.csdn.net/qq_42880719/article/details/116278751

版权


ctf 专栏收录该内容

75 篇文章 28 订阅

订阅专栏

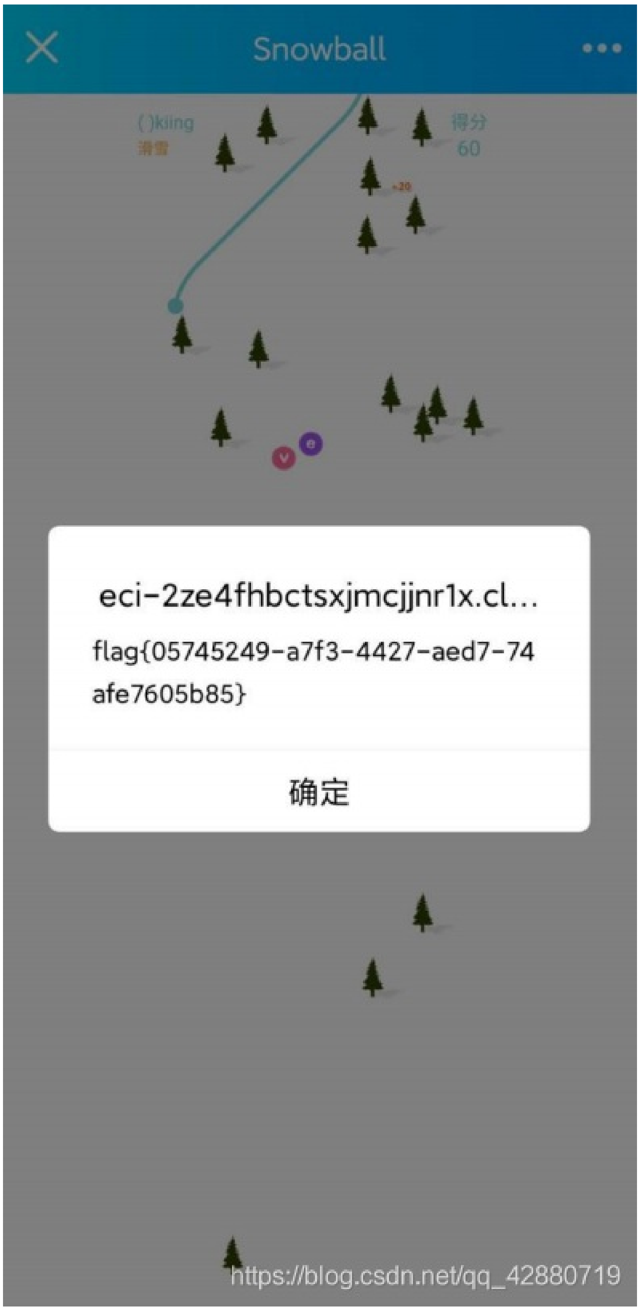
蓝帽越来越离谱

争分夺秒，101分排到前70

Ball_sigin + slient + 冬奥会_is_coming + 复现I_will_but_not_quite + 根据T佬复现嫌疑人x的硬盘整理

1.web：Ball_sigin

纯玩游戏，会出现3个单词缺一个字母，分别吃到对应字母即可出flag



eci-2ze4fhbctsxjmcjjnr1x.cl...

flag{05745249-a7f3-4427-aed7-74afe7605b85}

确定

https://blog.csdn.net/qq_42880719

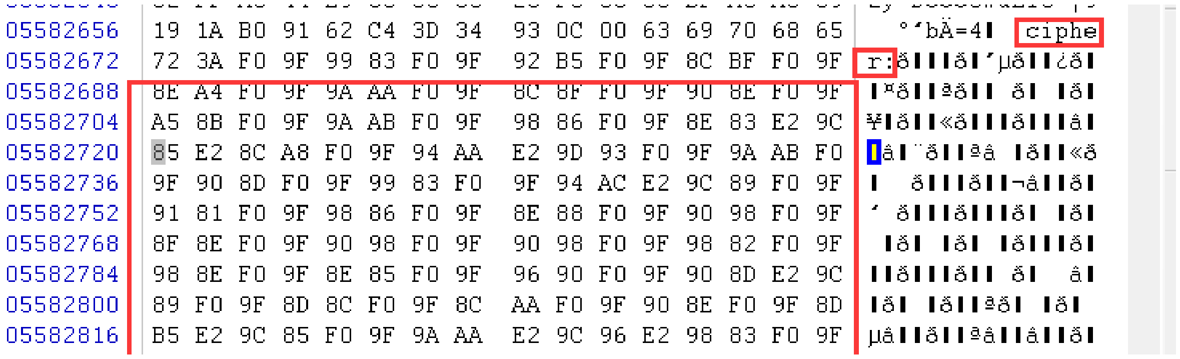
2.PWN:slient

既然是原题，就可以直接算杂项了吧(雾

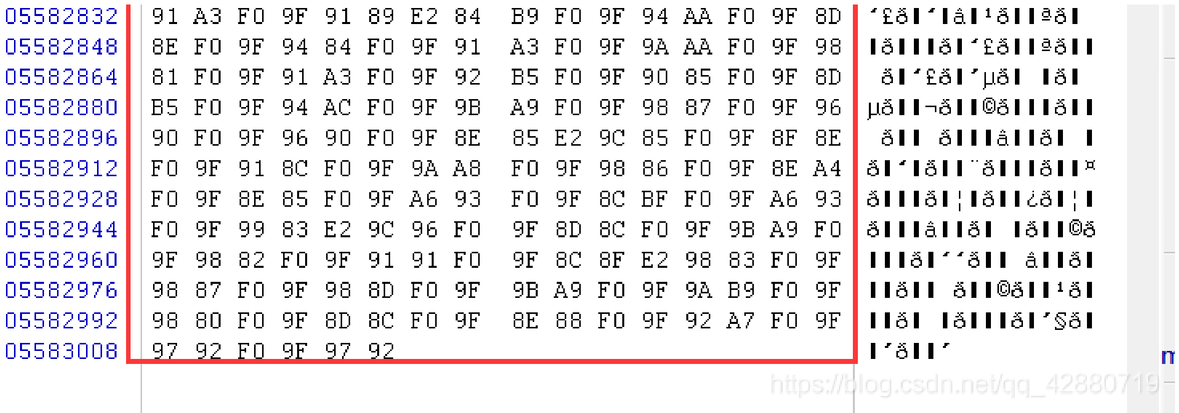
直接参考这个：https://www.lintstar.top/2020/12/784edd2e的slient，改掉端口和ip即可，flag都没变

3.冬奥会_is_coming

png文件尾有rar，foremost分离，分离出来个mp3，并且rar的注释里面提示8个数字

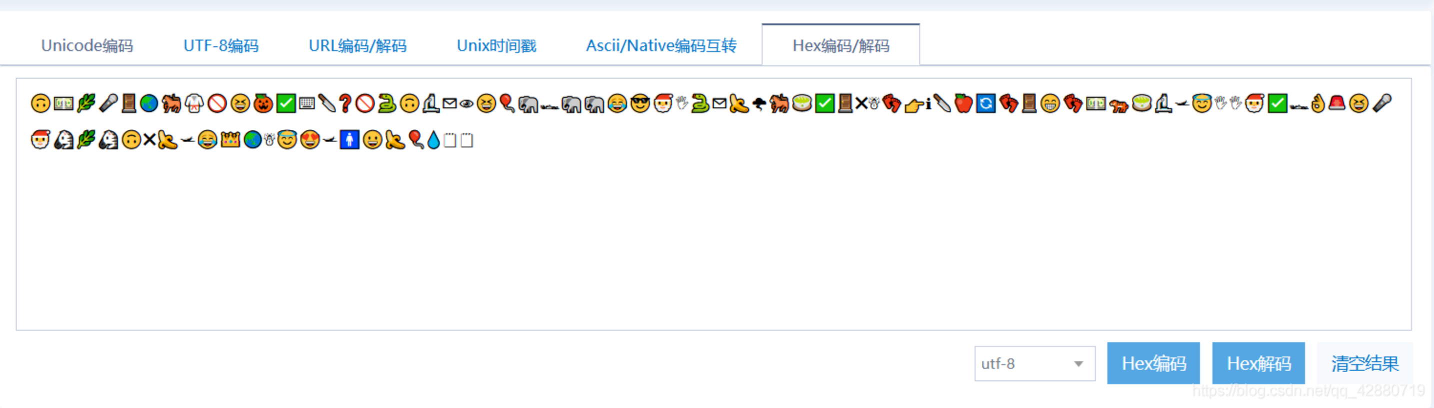


```
05582656  19 1A B0 91 62 C4 3D 34  93 0C 00 63 69 70 68 65    °‘bÄ=4▌ ciphe
05582672  72 3A F0 9F 99 83 F0 9F  92 B5 F0 9F 8C BF F0 9F  r:ð▌▌▌ð▌´µð▌▌¿ð▌
05582688  8E A4 F0 9F 9A AA F0 9F  8C 8F F0 9F 90 8E F0 9F  ▌¤ð▌▌ªð▌▌ ð▌ ▌ð▌
05582704  A5 8B F0 9F 9A AB F0 9F  98 86 F0 9F 8E 83 E2 9C  ¥▌ð▌▌«ð▌▌▌ð▌▌▌â▌
05582720  85 E2 8C A8 F0 9F 94 AA  E2 9D 93 F0 9F 9A AB F0  ▌â▌¨ð▌▌ªâ ▌ð▌▌«ð
05582736  9F 90 8D F0 9F 99 83 F0  9F 94 AC E2 9C 89 F0 9F  ▌  ð▌▌▌ð▌▌¬â▌▌ð▌
05582752  91 81 F0 9F 98 86 F0 9F  8E 88 F0 9F 90 98 F0 9F  ‘ ð▌▌▌ð▌▌▌ð▌ ▌ð▌
05582768  8F 8E F0 9F 90 98 F0 9F  90 98 F0 9F 98 82 F0 9F   ▌ð▌ ▌ð▌ ▌ð▌▌▌ð▌
05582784  98 8E F0 9F 8E 85 F0 9F  96 90 F0 9F 90 8D E2 9C  ▌▌ð▌▌▌ð▌▌ ð▌  â▌
05582800  89 F0 9F 8D 8C F0 9F 8C  AA F0 9F 90 8E F0 9F 8D  ▌ð▌ ▌ð▌▌ªð▌ ▌ð▌
05582816  B5 E2 9C 85 F0 9F 9A AA  E2 9C 96 E2 98 83 F0 9F  µâ▌▌ð▌▌ªâ▌▌â▌▌ð▌
```



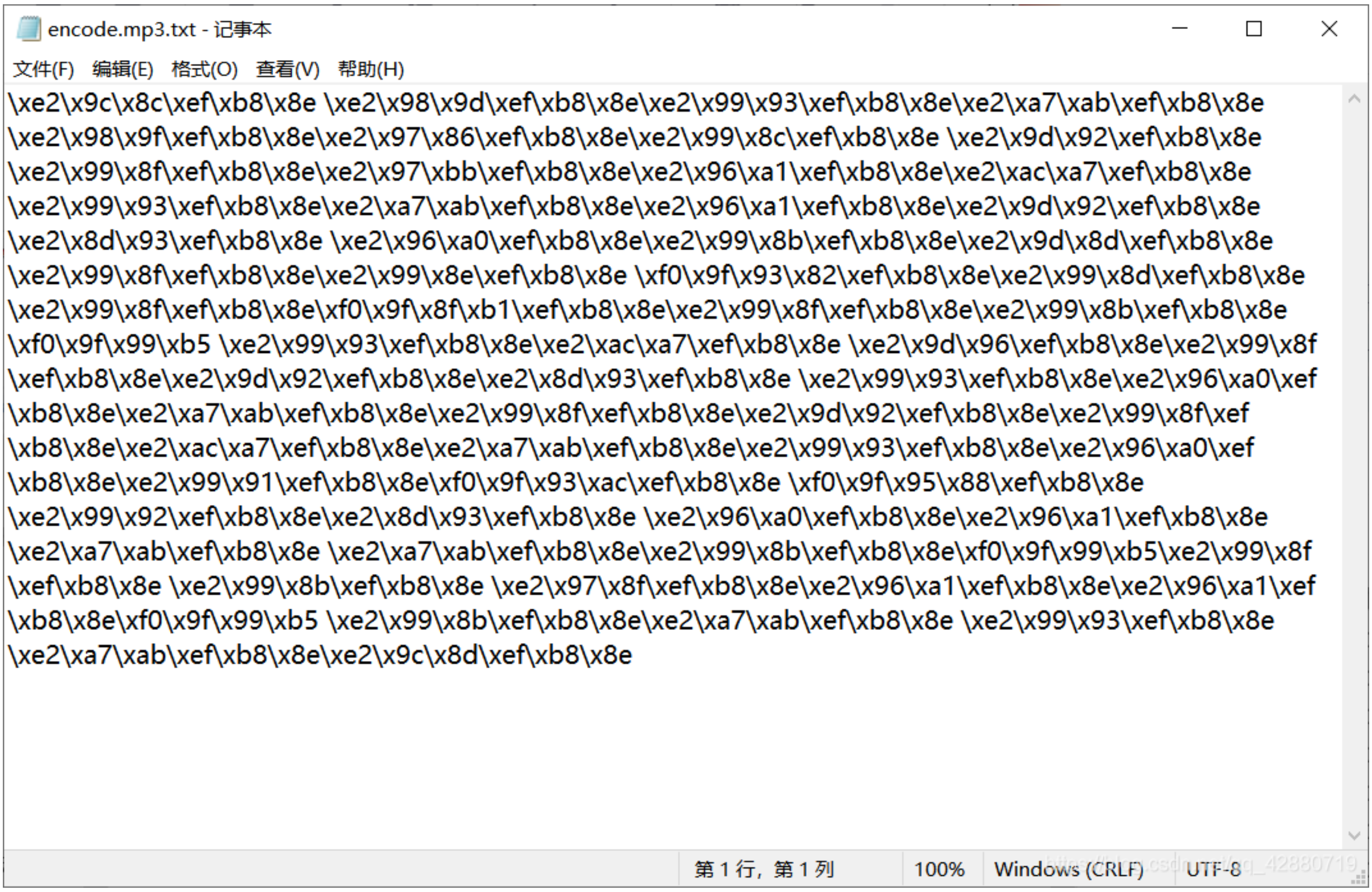
```
05582832  91 A3 F0 9F 91 89 E2 84  B9 F0 9F 94 AA F0 9F 8D  ´£ðI´IâI¹ðII ªðI
05582848  8E F0 9F 94 84 F0 9F 91  A3 F0 9F 9A AA F0 9F 98  IðIIðI´£ðIIªðII
05582864  81 F0 9F 91 A3 F0 9F 92  B5 F0 9F 90 85 F0 9F 8D   ðI´£ðI´µðI IðI
05582880  B5 F0 9F 94 AC F0 9F 9B  A9 F0 9F 98 87 F0 9F 96  µðII¬ðII©ðIIIðII
05582896  90 F0 9F 96 90 F0 9F 8E  85 E2 9C 85 F0 9F 8F 8E   ðII ðIIIâIIðI I
05582912  F0 9F 91 8C F0 9F 9A A8  F0 9F 98 86 F0 9F 8E A4  ðI´IðII¨ðIIIðII¤
05582928  F0 9F 8E 85 F0 9F A6 93  F0 9F 8C BF F0 9F A6 93  ðIIIðI¦IðII¿ðI¦I
05582944  F0 9F 99 83 E2 9C 96 F0  9F 8D 8C F0 9F 9B A9 F0  ðIIIâIIðI IðII©ð
05582960  9F 98 82 F0 9F 91 91 F0  9F 8C 8F E2 98 83 F0 9F  IIIðI´´ðII âIIðI
05582976  98 87 F0 9F 98 8D F0 9F  9B A9 F0 9F 9A B9 F0 9F  IIðII ðII©ðII¹ðI
05582992  98 80 F0 9F 8D 8C F0 9F  8E 88 F0 9F 92 A7 F0 9F  IIðI IðIIIðI´§ðI
05583008  97 92 F0 9F 97 92                                 I´ðII´
```

文件尾发现密文，将其16进制导出后转hex http://stool.chinaz.com/hex



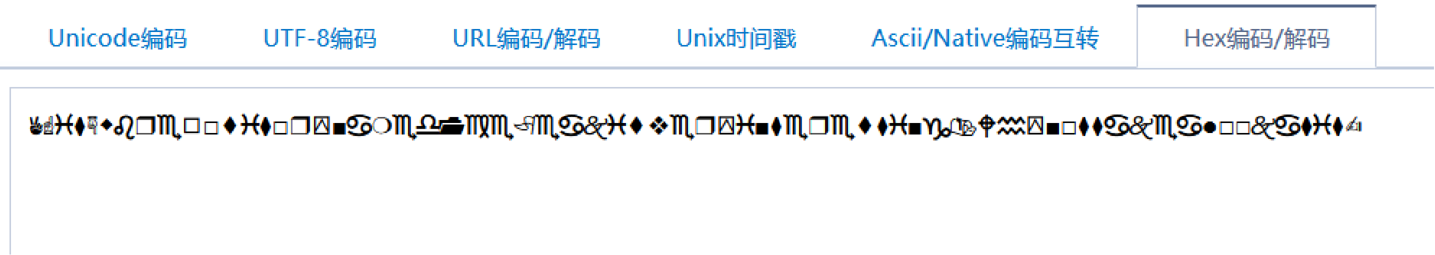
🙃💵🌿🎤🚪🌏🐎🥋🚫😆🎃✅⌨🔪❓🚫🐍🙃🔬✉👁😆🎈🐘🛩🐘🐘😂😎🎅🖐🐍✉🍌🌪🐎🍵✅🚪✖☃👣👉ℹ🔪🍎🔄👣🚪😁👣💵🐅🍵🔬🛩😇🖐🖐🎅✅🛩👌🚨😆🎤
🎅🦓🌿🦓🙃✖🍌🛩😂👑🌏☃😇😍🛩🚹😀🍌🎈💧□□

然后使用emoji-aeshttps://aghorler.github.io/emoji-aes/生成一个flag{}，发现与密文前几位相同，锁定emoji-aes，寻找秘钥有mp3并且提示八位数字，猜测与本次主题有关，尝试歌曲的发布时间和冬奥会开始时间，发现20220204这个数字拿去mp3stego能解出来

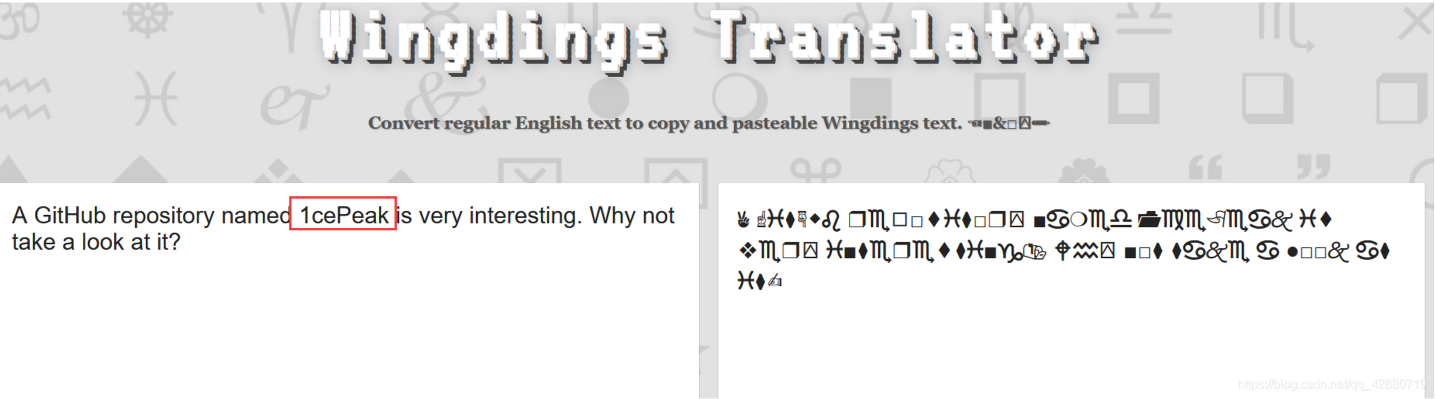


```
\xe2\x9c\x8c\xef\xb8\x8e \xe2\x98\x9d\xef\xb8\x8e\xe2\x99\x93\xef\xb8\x8e\xe2\xa7\xab\xef\xb8\x8e
\xe2\x98\x9f\xef\xb8\x8e\xe2\x97\x86\xef\xb8\x8e\xe2\x99\x8c\xef\xb8\x8e \xe2\x9d\x92\xef\xb8\x8e
\xe2\x99\x8f\xef\xb8\x8e\xe2\x97\xbb\xef\xb8\x8e\xe2\x96\xa1\xef\xb8\x8e\xe2\xac\xa7\xef\xb8\x8e
\xe2\x99\x93\xef\xb8\x8e\xe2\xa7\xab\xef\xb8\x8e\xe2\x96\xa1\xef\xb8\x8e\xe2\x9d\x92\xef\xb8\x8e
\xe2\x8d\x93\xef\xb8\x8e \xe2\x96\xa0\xef\xb8\x8e\xe2\x99\x8b\xef\xb8\x8e\xe2\x9d\x8d\xef\xb8\x8e
\xe2\x99\x8f\xef\xb8\x8e\xe2\x99\x8e\xef\xb8\x8e \xf0\x9f\x93\x82\xef\xb8\x8e\xe2\x99\x8d\xef\xb8\x8e
\xe2\x99\x8f\xef\xb8\x8e\xf0\x9f\x8f\xb1\xef\xb8\x8e\xe2\x99\x8f\xef\xb8\x8e\xe2\x99\x8b\xef\xb8\x8e
\xf0\x9f\x99\xb5 \xe2\x99\x93\xef\xb8\x8e\xe2\xac\xa7\xef\xb8\x8e \xe2\x9d\x96\xef\xb8\x8e\xe2\x99\x8f
\xef\xb8\x8e\xe2\x9d\x92\xef\xb8\x8e\xe2\x8d\x93\xef\xb8\x8e \xe2\x99\x93\xef\xb8\x8e\xe2\x96\xa0\xef
\xb8\x8e\xe2\xa7\xab\xef\xb8\x8e\xe2\x99\x8f\xef\xb8\x8e\xe2\x9d\x92\xef\xb8\x8e\xe2\x99\x8f\xef
\xb8\x8e\xe2\xac\xa7\xef\xb8\x8e\xe2\xa7\xab\xef\xb8\x8e\xe2\x99\x93\xef\xb8\x8e\xe2\x96\xa0\xef
\xb8\x8e\xe2\x99\x91\xef\xb8\x8e\xf0\x9f\x93\xac\xef\xb8\x8e \xf0\x9f\x95\x88\xef\xb8\x8e
\xe2\x99\x92\xef\xb8\x8e\xe2\x8d\x93\xef\xb8\x8e \xe2\x96\xa0\xef\xb8\x8e\xe2\x96\xa1\xef\xb8\x8e
\xe2\xa7\xab\xef\xb8\x8e \xe2\xa7\xab\xef\xb8\x8e\xe2\x99\x8b\xef\xb8\x8e\xf0\x9f\x99\xb5\xe2\x99\x8f
\xef\xb8\x8e \xe2\x99\x8b\xef\xb8\x8e \xe2\x97\x8f\xef\xb8\x8e\xe2\x96\xa1\xef\xb8\x8e\xe2\x96\xa1\xef
\xb8\x8e\xf0\x9f\x99\xb5 \xe2\x99\x8b\xef\xb8\x8e\xe2\xa7\xab\xef\xb8\x8e \xe2\x99\x93\xef\xb8\x8e
\xe2\xa7\xab\xef\xb8\x8e\xe2\x9c\x8d\xef\xb8\x8e
```

将\x替换为空，再去刚刚hex

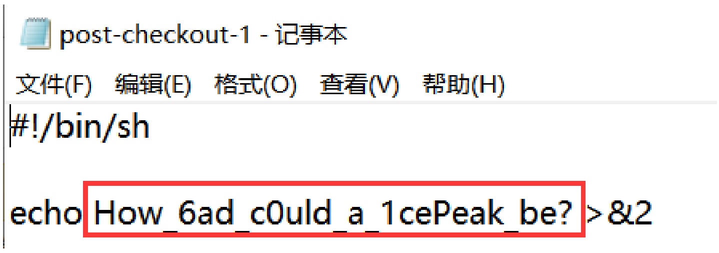


bugku做过，wingdings(闹酒狂欢)
使用https://lingojam.com/WingdingsTranslator


Wingdings Translator
Convert regular English text to copy and pasteable Wingdings text.

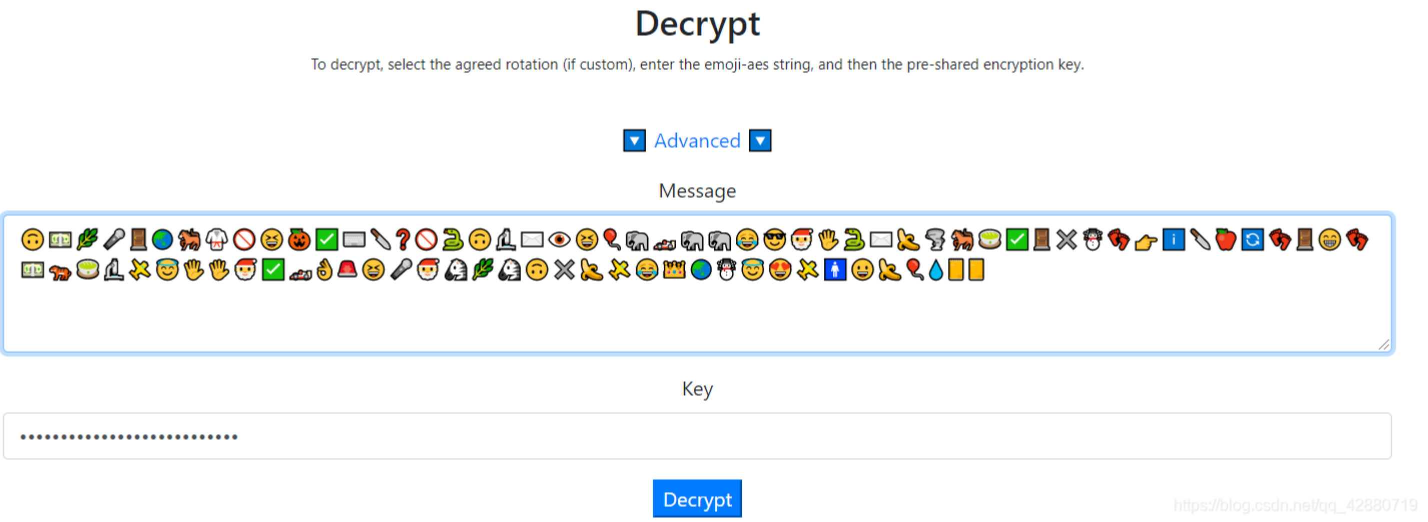
A GitHub repository named 1cePeak is very interesting. Why not take a look at it?

最后找到了这个https://github.com/Tr0jAnV1rU4/1cePeak/blob/main/A/post-checkout
下载下来记事本打开



```
#!/bin/sh

echo How_6ad_c0uld_a_1cePeak_be? >&2
```



Decrypt

To decrypt, select the agreed rotation (if custom), enter the emoji-aes string, and then the pre-shared encryption key.

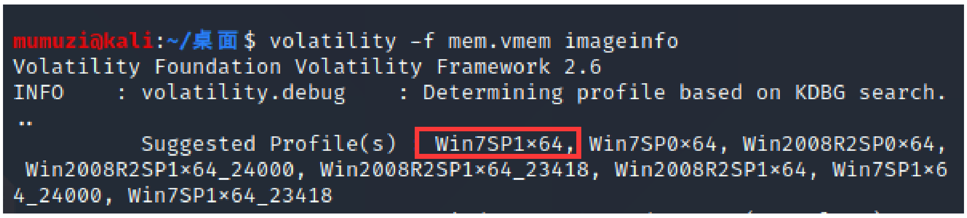
```
flag{e32f619b-dbcd-49bd-9126-5d841aa01767}
```

（复现）4.I_will_but_not_quite

vmem，明显内存取证题，还给了个加密python就离谱啊

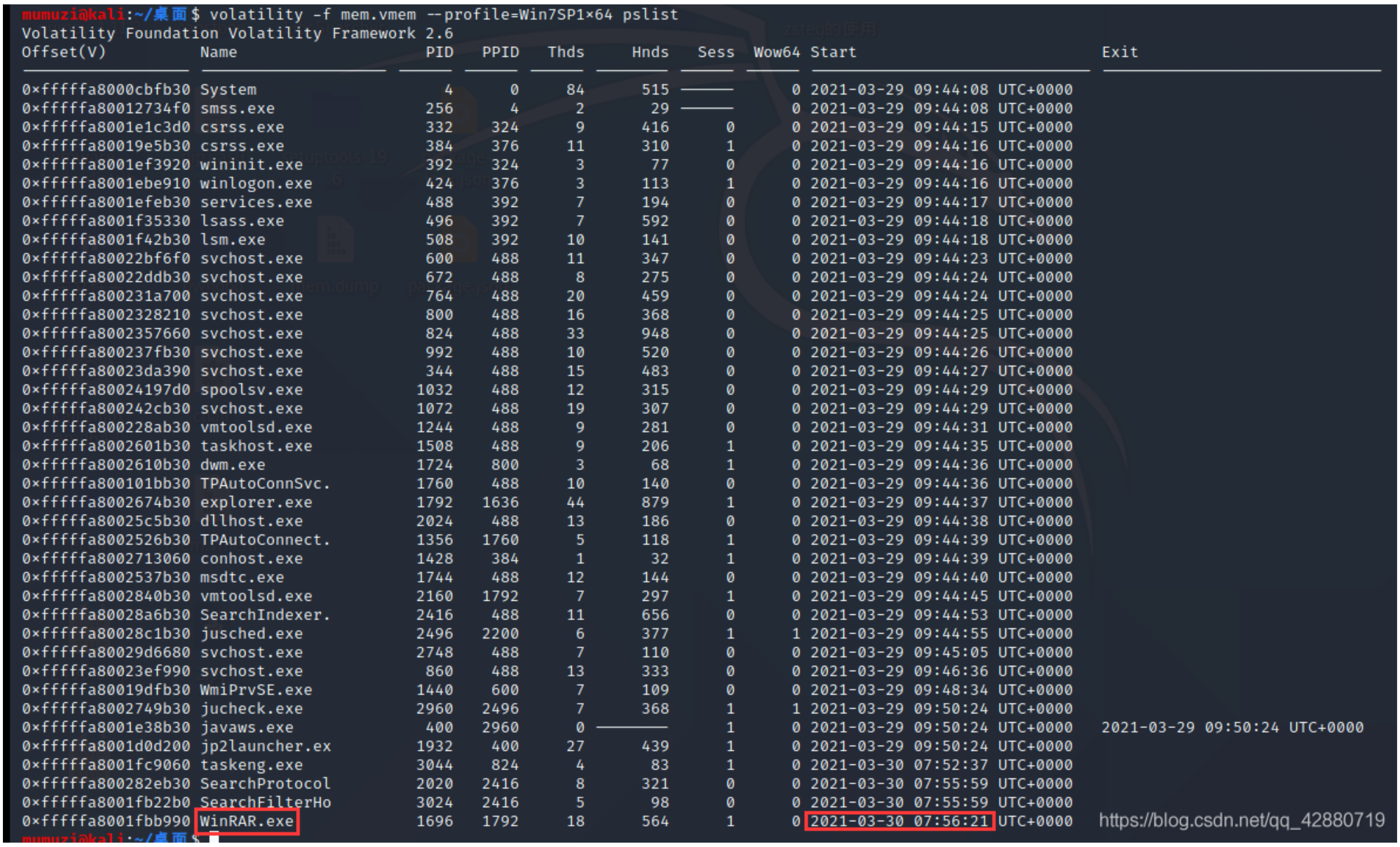
先进行常规操作

先查profile



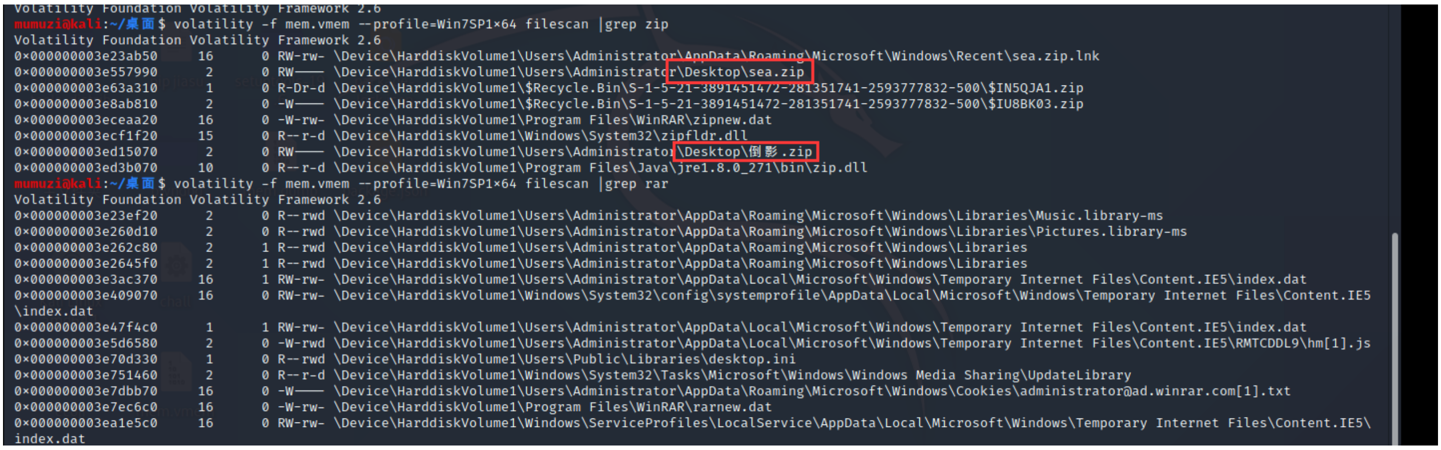
```
mumuzi@kali:~/桌面$ volatility -f mem.vmem imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search.
..
          Suggested Profile(s) : Win7SP1×64, Win7SP0×64, Win2008R2SP0×64,
 Win2008R2SP1×64_24000, Win2008R2SP1×64_23418, Win2008R2SP1×64, Win7SP1×6
4_24000, Win7SP1×64_23418
```

然后查一下进程

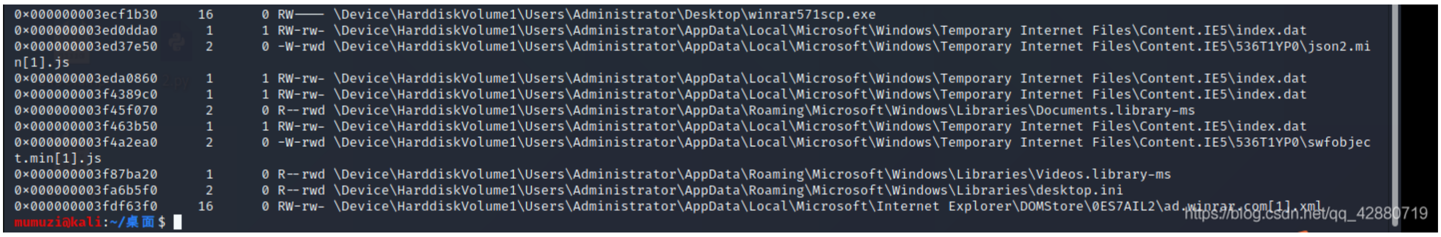


```
mumuzi@kali:~/桌面$ volatility -f mem.vmem --profile=Win7SP1×64 pslist
Volatility Foundation Volatility Framework 2.6
Offset(V)          Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
------------------ -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
0×fffffa8000cbfb30 System                    4      0     84      515 ------      0 2021-03-29 09:44:08 UTC+0000
0×fffffa80012734f0 smss.exe                256      4      2       29 ------      0 2021-03-29 09:44:08 UTC+0000
0×fffffa8001e1c3d0 csrss.exe               332    324      9      416      0      0 2021-03-29 09:44:15 UTC+0000
0×fffffa80019e5b30 csrss.exe               384    376     11      310      1      0 2021-03-29 09:44:16 UTC+0000
0×fffffa8001ef3920 wininit.exe             392    324      3       77      0      0 2021-03-29 09:44:16 UTC+0000
0×fffffa8001ebe910 winlogon.exe            424    376      3      113      1      0 2021-03-29 09:44:16 UTC+0000
0×fffffa8001efeb30 services.exe            488    392      7      194      0      0 2021-03-29 09:44:17 UTC+0000
0×fffffa8001f35330 lsass.exe               496    392      7      592      0      0 2021-03-29 09:44:18 UTC+0000
0×fffffa8001f42b30 lsm.exe                 508    392     10      141      0      0 2021-03-29 09:44:18 UTC+0000
0×fffffa80022bf6f0 svchost.exe             600    488     11      347      0      0 2021-03-29 09:44:23 UTC+0000
0×fffffa80022ddb30 svchost.exe             672    488      8      275      0      0 2021-03-29 09:44:24 UTC+0000
0×fffffa800231a700 svchost.exe             764    488     20      459      0      0 2021-03-29 09:44:24 UTC+0000
0×fffffa8002328210 svchost.exe             800    488     16      368      0      0 2021-03-29 09:44:25 UTC+0000
0×fffffa8002357660 svchost.exe             824    488     33      948      0      0 2021-03-29 09:44:25 UTC+0000
0×fffffa800237fb30 svchost.exe             992    488     10      520      0      0 2021-03-29 09:44:26 UTC+0000
0×fffffa80023da390 svchost.exe             344    488     15      483      0      0 2021-03-29 09:44:27 UTC+0000
0×fffffa80024197d0 spoolsv.exe            1032    488     12      315      0      0 2021-03-29 09:44:29 UTC+0000
0×fffffa800242cb30 svchost.exe            1072    488     19      307      0      0 2021-03-29 09:44:29 UTC+0000
0×fffffa800228ab30 vmtoolsd.exe           1244    488      9      281      0      0 2021-03-29 09:44:31 UTC+0000
0×fffffa8002601b30 taskhost.exe           1508    488      9      206      1      0 2021-03-29 09:44:35 UTC+0000
0×fffffa8002610b30 dwm.exe                1724    800      3       68      1      0 2021-03-29 09:44:36 UTC+0000
0×fffffa800101bb30 TPAutoConnSvc.         1760    488     10      140      0      0 2021-03-29 09:44:36 UTC+0000
0×fffffa8002674b30 explorer.exe           1792   1636     44      879      1      0 2021-03-29 09:44:37 UTC+0000
0×fffffa80025c5b30 dllhost.exe            2024    488     13      186      0      0 2021-03-29 09:44:38 UTC+0000
0×fffffa8002526b30 TPAutoConnect.         1356   1760      5      118      1      0 2021-03-29 09:44:39 UTC+0000
0×fffffa8002713060 conhost.exe            1428    384      1       32      1      0 2021-03-29 09:44:39 UTC+0000
0×fffffa8002537b30 msdtc.exe              1744    488     12      144      0      0 2021-03-29 09:44:40 UTC+0000
0×fffffa8002840b30 vmtoolsd.exe           2160   1792      7      297      1      0 2021-03-29 09:44:45 UTC+0000
0×fffffa80028a6b30 SearchIndexer.         2416    488     11      656      0      0 2021-03-29 09:44:53 UTC+0000
0×fffffa80028c1b30 jusched.exe            2496   2200      6      377      1      1 2021-03-29 09:44:55 UTC+0000
0×fffffa80029d6680 svchost.exe            2748    488      7      110      0      0 2021-03-29 09:45:05 UTC+0000
0×fffffa80023ef990 svchost.exe             860    488     13      333      0      0 2021-03-29 09:46:36 UTC+0000
0×fffffa80019dfb30 WmiPrvSE.exe           1440    600      7      109      0      0 2021-03-29 09:48:34 UTC+0000
0×fffffa8002749b30 jucheck.exe            2960   2496      7      368      1      1 2021-03-29 09:50:24 UTC+0000
0×fffffa8001e38b30 javaws.exe              400   2960      0 --------      1      0 2021-03-29 09:50:24 UTC+0000 2021-03-29 09:50:24 UTC+0000
0×fffffa8001d0d200 jp2launcher.ex         1932    400     27      439      1      0 2021-03-29 09:50:24 UTC+0000
0×fffffa8001fc9060 taskeng.exe            3044    824      4       83      1      0 2021-03-30 07:52:37 UTC+0000
0×fffffa80028 2eb30 SearchProtocol        2020   2416      8      321      0      0 2021-03-30 07:55:59 UTC+0000
0×fffffa8001fb22b0 SearchFilterHo         3024   2416      5       98      0      0 2021-03-30 07:55:59 UTC+0000
0×fffffa8001fbb990 WinRAR.exe             1696   1792     18      564      1      0 2021-03-30 07:56:21 UTC+0000
mumuzi@kali:~/桌面$
```

可以发现最后使用的是winrar，猜测进行了压缩，|grep rar和zip试试

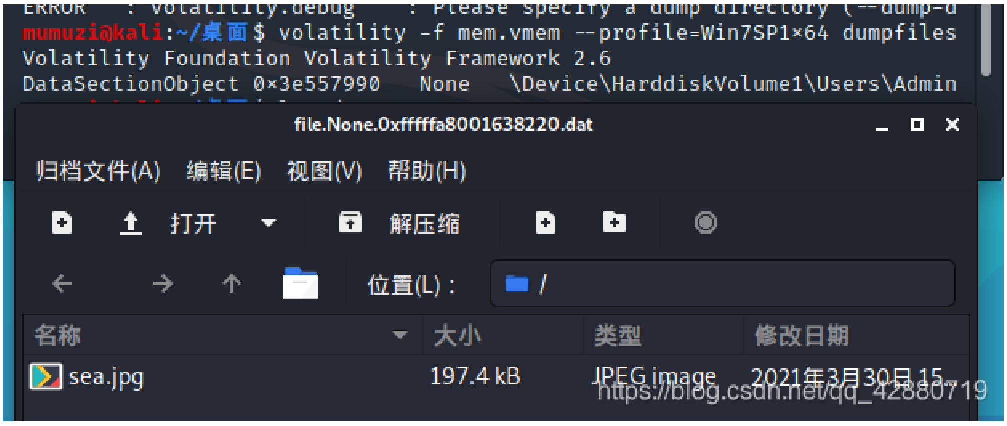


```
Volatility Foundation Volatility Framework 2.6
mumuzi@kali:~/桌面$ volatility -f mem.vmem --profile=Win7SP1×64 filescan |grep zip
Volatility Foundation Volatility Framework 2.6
0×000000003e23ab50     16      0 RW-rw- \Device\HarddiskVolume1\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\sea.zip.lnk
0×000000003e557990      2      0 RW---- \Device\HarddiskVolume1\Users\Administrator\Desktop\sea.zip
0×000000003e63a310      1      0 R-Dr-d \Device\HarddiskVolume1\$Recycle.Bin\S-1-5-21-3891451472-281351741-2593777832-500\$IN5QJA1.zip
0×000000003e8ab810      2      0 -W---- \Device\HarddiskVolume1\$Recycle.Bin\S-1-5-21-3891451472-281351741-2593777832-500\$IU8BK03.zip
0×000000003eceaa20     16      0 -W-rw- \Device\HarddiskVolume1\Program Files\WinRAR\zipnew.dat
0×000000003ecf1f20     15      0 R--r-d \Device\HarddiskVolume1\Windows\System32\zipfldr.dll
0×000000003ed15070      2      0 RW---- \Device\HarddiskVolume1\Users\Administrator\Desktop\倒影.zip
0×000000003ed3b070     10      0 R--r-d \Device\HarddiskVolume1\Program Files\Java\jre1.8.0_271\bin\zip.dll
mumuzi@kali:~/桌面$ volatility -f mem.vmem --profile=Win7SP1×64 filescan |grep rar
Volatility Foundation Volatility Framework 2.6
0×000000003e23ef20      2      0 R--rwd \Device\HarddiskVolume1\Users\Administrator\AppData\Roaming\Microsoft\Windows\Libraries\Music.library-ms
0×000000003e260d10      2      0 R--rwd \Device\HarddiskVolume1\Users\Administrator\AppData\Roaming\Microsoft\Windows\Libraries\Pictures.library-ms
0×000000003e262c80      2      1 R--rwd \Device\HarddiskVolume1\Users\Administrator\AppData\Roaming\Microsoft\Windows\Libraries
0×000000003e2645f0      2      1 R--rwd \Device\HarddiskVolume1\Users\Administrator\AppData\Roaming\Microsoft\Windows\Libraries
0×000000003e3ac370     16      1 RW-rw- \Device\HarddiskVolume1\Users\Administrator\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\index.dat
0×000000003e409070     16      0 RW-rw- \Device\HarddiskVolume1\Windows\System32\config\systemprofile\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5
\index.dat
0×000000003e47f4c0      1      1 RW-rw- \Device\HarddiskVolume1\Users\Administrator\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\index.dat
0×000000003e5d6580      2      0 -W-rwd \Device\HarddiskVolume1\Users\Administrator\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\RMTCDDL9\hm[1].js
0×000000003e70d330      1      0 R--rwd \Device\HarddiskVolume1\Users\Public\Libraries\desktop.ini
0×000000003e751460      2      0 R--r-d \Device\HarddiskVolume1\Windows\System32\Tasks\Microsoft\Windows\Windows Media Sharing\UpdateLibrary
0×000000003e7dbb70     16      0 -W---- \Device\HarddiskVolume1\Users\Administrator\AppData\Roaming\Microsoft\Windows\Cookies\administrator@ad.winrar.com[1].txt
0×000000003e7ec6c0     16      0 -W-rw- \Device\HarddiskVolume1\Program Files\WinRAR\rarnew.dat
0×000000003ea1e5c0     16      0 RW-rw- \Device\HarddiskVolume1\Windows\ServiceProfiles\LocalService\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\
index.dat
```



```
0×000000003ecf1b30     16      0 RW———— \Device\HarddiskVolume1\Users\Administrator\Desktop\winrar571scp.exe
0×000000003ed0dda0      1      1 RW-rw- \Device\HarddiskVolume1\Users\Administrator\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\index.dat
0×000000003ed37e50      2      0 -W-rwd \Device\HarddiskVolume1\Users\Administrator\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\536T1YP0\json2.mi
n[1].js
0×000000003eda0860      1      1 RW-rw- \Device\HarddiskVolume1\Users\Administrator\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\index.dat
0×000000003f4389c0      1      1 RW-rw- \Device\HarddiskVolume1\Users\Administrator\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\index.dat
0×000000003f45f070      2      0 R--rwd \Device\HarddiskVolume1\Users\Administrator\AppData\Roaming\Microsoft\Windows\Libraries\Documents.library-ms
0×000000003f463b50      1      1 RW-rw- \Device\HarddiskVolume1\Users\Administrator\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\index.dat
0×000000003f4a2ea0      2      0 -W-rwd \Device\HarddiskVolume1\Users\Administrator\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\536T1YP0\swfobjec
t.min[1].js
0×000000003f87ba20      1      0 R--rwd \Device\HarddiskVolume1\Users\Administrator\AppData\Roaming\Microsoft\Windows\Libraries\Videos.library-ms
0×000000003fa6b5f0      2      0 R--rwd \Device\HarddiskVolume1\Users\Administrator\AppData\Roaming\Microsoft\Windows\Libraries\desktop.ini
0×000000003fdf63f0     16      0 RW-rw- \Device\HarddiskVolume1\Users\Administrator\AppData\Local\Microsoft\Internet Explorer\DOMStore\0ES7AIL2\ad.winrar.com[1].xml
mumuzi@kali:~/桌面$ 
```

这两名字奇怪还出现在桌面上，必须得dump出来看看，(另一个没用

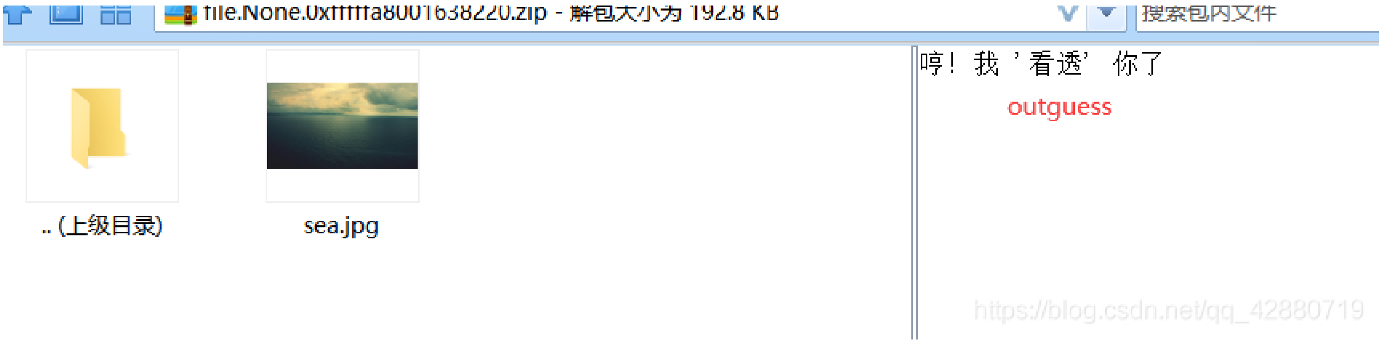


```
ERROR   : volatility.debug    : Please specify a dump directory (--dump-d
mumuzi@kali:~/桌面$ volatility -f mem.vmem --profile=Win7SP1x64 dumpfiles
Volatility Foundation Volatility Framework 2.6
DataSectionObject 0×3e557990   None   \Device\HarddiskVolume1\Users\Admin
```

file.None.0xfffffa8001638220.dat
归档文件(A) 编辑(E) 视图(V) 帮助(H)
打开
解压缩
位置(L)：
/

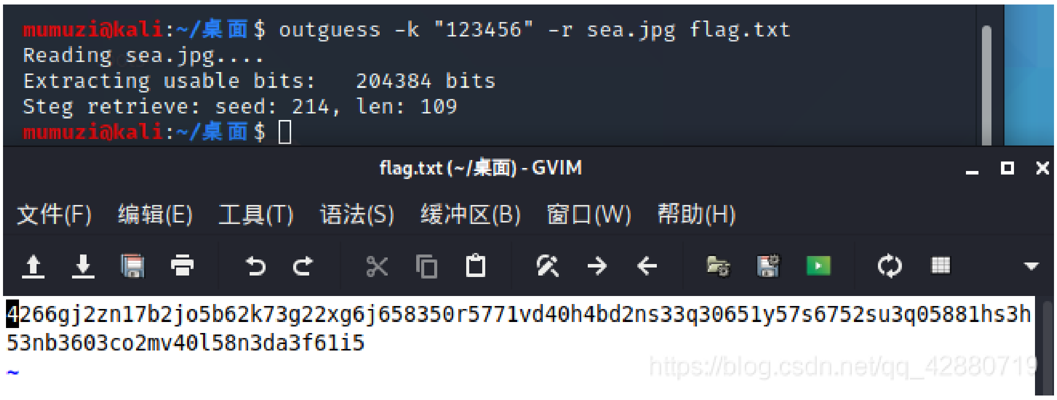
名称	大小	类型	修改日期
sea.jpg	197.4 kB	JPEG image	2021年3月30日 15

kali其实看不到注释，这里当时是师兄dump出来然后发qq，windows看到了注释



哼！我＇看透＇你了
outguess

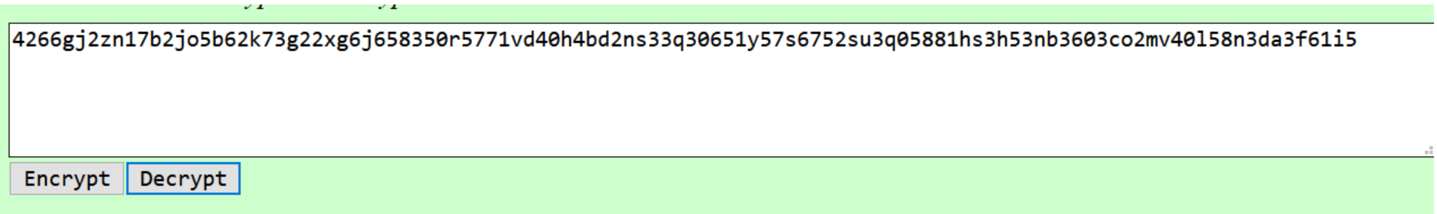
密码猜测成功弱密码123456



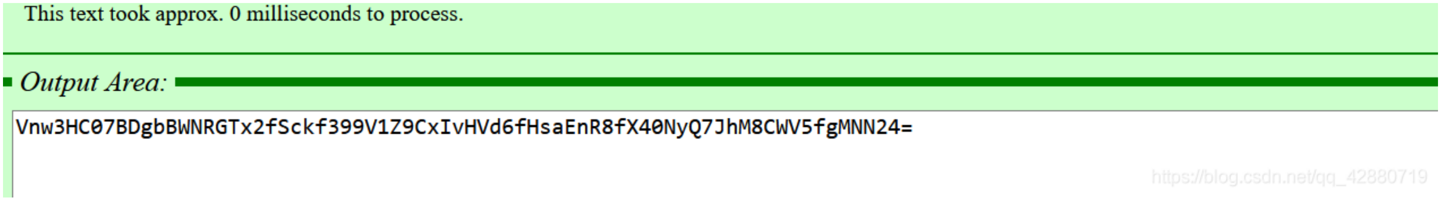
```
mumuzi@kali:~/桌面$ outguess -k "123456" -r sea.jpg flag.txt
Reading sea.jpg....
Extracting usable bits:   204384 bits
Steg retrieve: seed: 214, len: 109
mumuzi@kali:~/桌面$ 
```

```
4266gj2zn17b2jo5b62k73g22xg6j658350r5771vd40h4bd2ns33q30651y57s6752su3q05881hs3h
53nb3603co2mv40l58n3da3f61i5
~
```

然后这里盲区其实没碰到过，是双□六进制编码https://www.calcresult.com/misc/cyphers/twin-hex.html



```
4266gj2zn17b2jo5b62k73g22xg6j658350r5771vd40h4bd2ns33q30651y57s6752su3q05881hs3h53nb3603co2mv40l58n3da3f61i5
```



This text took approx. 0 milliseconds to process.

Vnw3HC07BDgbBWNRGTx2fSckf399V1Z9CxIvHVd6fHsaEnR8fX40NyQ7JhM8CWV5fgMNN24=

> Vnw3HC07BDgbBWNRGTx2fSckf399V1Z9CxIvHVd6fHsaEnR8fX40NyQ7JhM8CWV5fgMNN24=

然后看那个加密函数

```python
#!/user/bin/python2
import random
def r(s, num):
 l=""
 for i in s:
  if(ord(i) in range(97,97+26)):
   l+=chr((ord(i)-97+num)%26+97)
  else:
   l+=i
 return l

def x(a, b):
 return chr(ord(a)^ord(b))

def encrypt(c):
 secret = c
 n=random.randint(1,1000)
 for i in range(n):
  secret = r(secret, random.randint(1,26))
 secret = secret.encode('base64')

 l = ""
 for i in range(len(secret)):
  l += x(secret[i], secret[(i+1)%len(secret)])
 return l.encode('base64')

flag = "#################"
print "secret =", encrypt(flag)

#secret = The key you got
```

encrypt相当于主函数，是随机一个n然后进入r凯撒加密，加密之后将其base64编码，然后对编码后的字符串每两位进行异或，最后得到的值再base64编码就得到了刚刚解出来的函数。

其中，虽然r里面获取了随机数n，还有对小写字母+num，但是由于是一起增加，所以最后只需要将0-26全部遍历一遍即可，所以暂时可忽略掉此等会再来写

问题出在x(secret[i], secret[(i+1)%len(secret)])
这里因为将最后一位也与第一位异或了，所以不能倒过来异或回去(因为极大可能会损失高位数据，事实也证明的确如此)

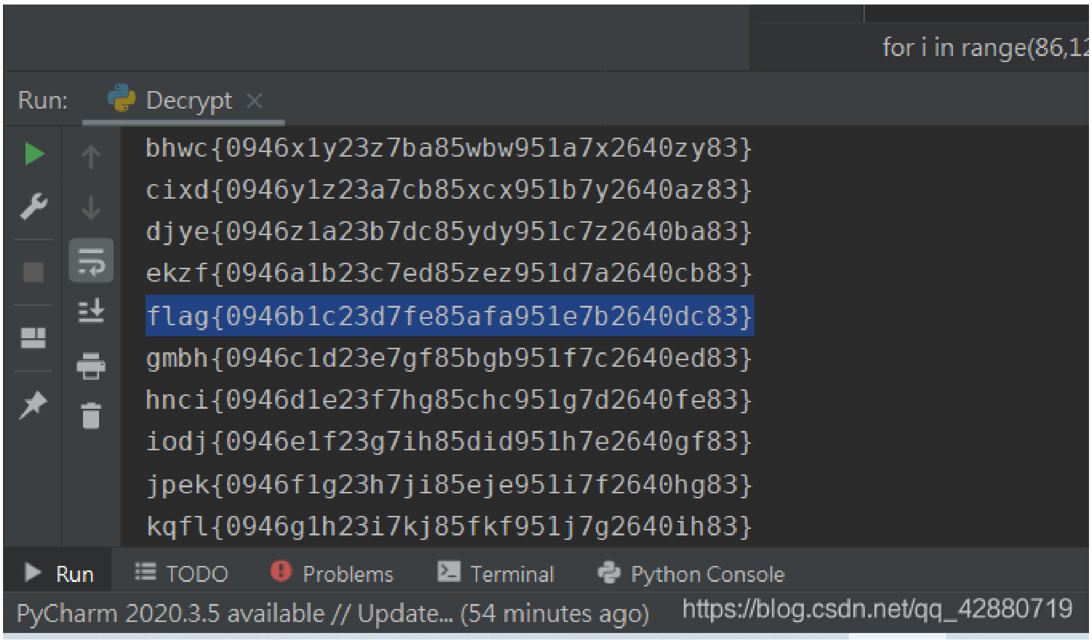
那么现在只需要得到第一次加密的base串，那串base解密用try-except,并循环1,26即可。但是如何得到那串base？

其实可以知道，虽然我之前说过了，因为每一位都进行了异或，所以不能逆回去，但是可以爆破啊。其实这里先将那串base解码之后看第一位和最后一位，一个110一个86，很容易将范围缩小，暂且尝试遍历(86,128)，这样我们就相当于得到了最后一位的ascii码十进制值，再逆回去的时候，当长度达到我们解出来的base64串时，停止，并try base64.b64decode(s).decode("utf-8)，如果能解码，则再执行r函数(或者上bugku去凯撒解密遍历，即可找到flag。

写脚本：

```python
import base64
import random
secret = "Vnw3HC07BDgbBWNRGTx2fSckf399V1Z9CxIvHVd6fHsaEnR8fX40NyQ7JhM8CWV5fgMNN24="
dec = base64.b64decode(secret).decode("utf-8")
# for i in range(len(dec)):
#  print(ord(dec[i]))
def r(s, num): #凯撒
 l=""
 for i in s:
  if(ord(i) in range(97,97+26)):
   l+=chr((ord(i)-97+num)%26+97)
  else:
   l+=i
 return l

for i in range(86,128):
 j = 1
 tmp = [""]*len(dec)
 tmp[-1] = chr(i)#爆破恢复最后一位，即可恢复所有
 while j != len(dec):
  tmp[-j-1] = chr(ord(dec[-j])^ord(tmp[-j])) #反着进行异或
  j += 1
 s = tmp[-1] #因为最后一位是最后一位和第一位异或，所以刚开始异或的其实是最后一位
 for i in range(len(tmp)-1):
  s += tmp[i]#这里即是将第2位至最后一位拼接起来加在第一位后面
 try:
  s = base64.b64decode(s).decode("utf-8")
  for i in range(1,26):#遍历凯撒
   flag = r(s,i)
   print(flag)
 except:
  pass
```

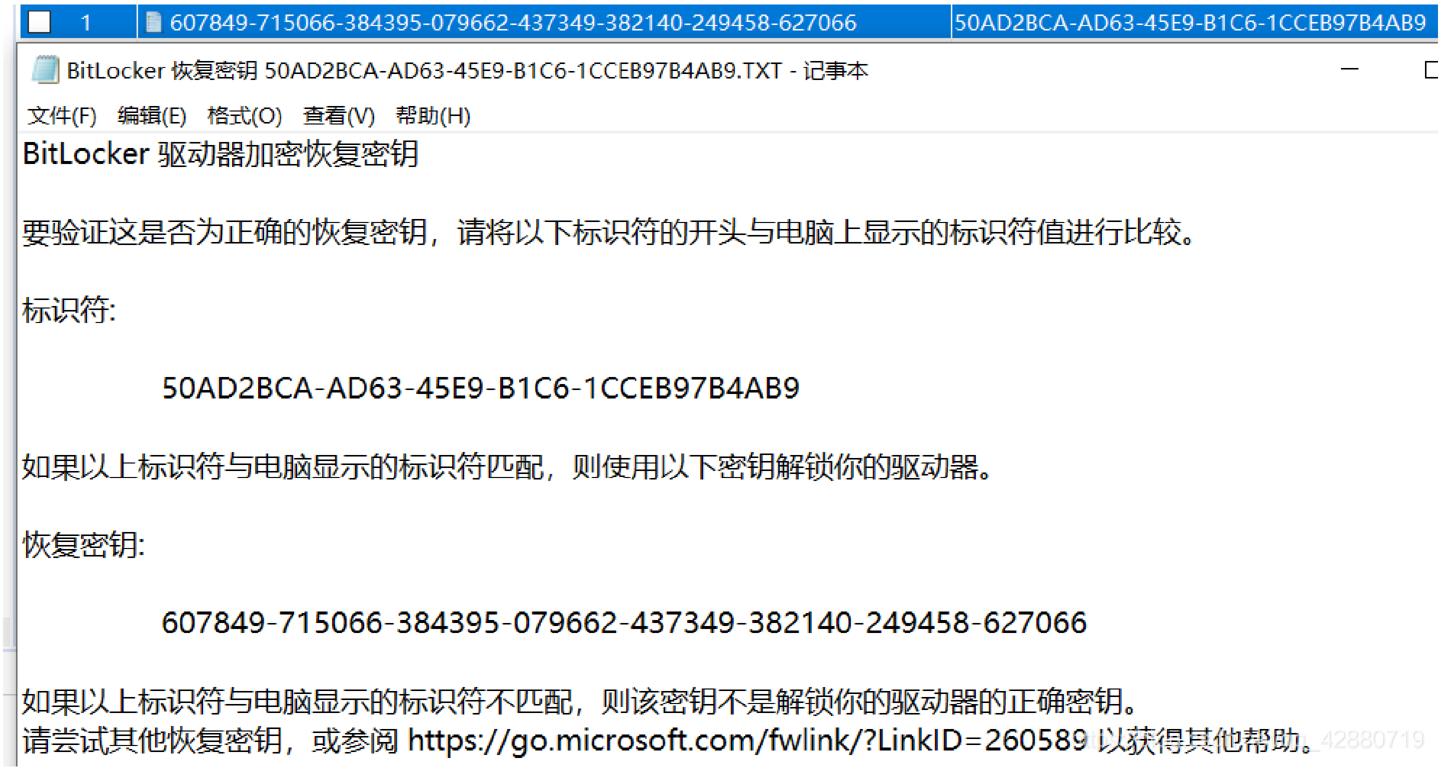


for i in range(86,12
Run:
Decrypt
Run
TODO
Problems
Terminal
Python Console
PyCharm 2020.3.5 available // Update... (54 minutes ago)
https://blog.csdn.net/qq_42880719

```
bhwc{0946x1y23z7ba85wbw951a7x2640zy83}
cixd{0946y1z23a7cb85xcx951b7y2640az83}
djye{0946z1a23b7dc85ydy951c7z2640ba83}
ekzf{0946a1b23c7ed85zez951d7a2640cb83}
flag{0946b1c23d7fe85afa951e7b2640dc83}
gmbh{0946c1d23e7gf85bgb951f7c2640ed83}
hnci{0946d1e23f7hg85chc951g7d2640fe83}
iodj{0946e1f23g7ih85did951h7e2640gf83}
jpek{0946f1g23h7ji85eje951i7f2640hg83}
kqfl{0946g1h23i7kj85fkf951j7g2640ih83}
```

5.嫌疑人x的硬盘整理
不会，稍微写一点再引一点最后T佬解出来的解题过程

将x.vmdk放进取证大师，提示需要bitlocker，取证完后查看取证结果



607849-715066-384395-079662-437349-382140-249458-627066 50AD2BCA-AD63-45E9-B1C6-1CCEB97B4AB9

BitLocker 恢复密钥 50AD2BCA-AD63-45E9-B1C6-1CCEB97B4AB9.TXT - 记事本

BitLocker 驱动器加密恢复密钥

要验证这是否为正确的恢复密钥，请将以下标识符的开头与电脑上显示的标识符值进行比较。

标识符:

    50AD2BCA-AD63-45E9-B1C6-1CCEB97B4AB9

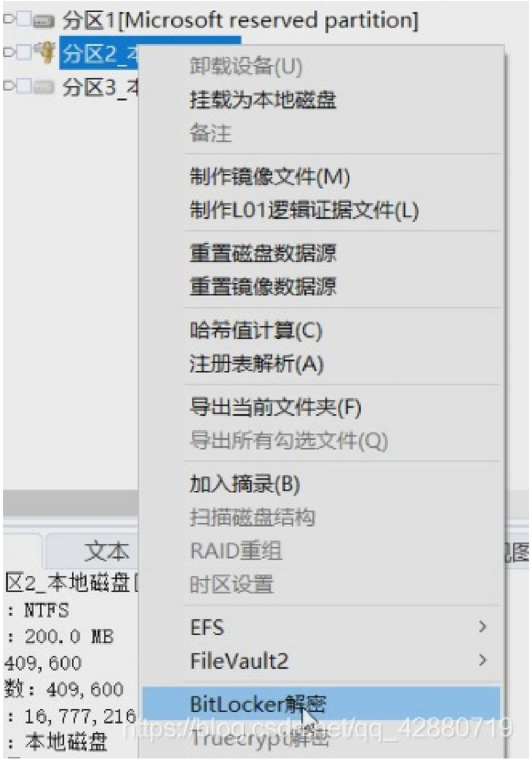
如果以上标识符与电脑显示的标识符匹配，则使用以下密钥解锁你的驱动器。

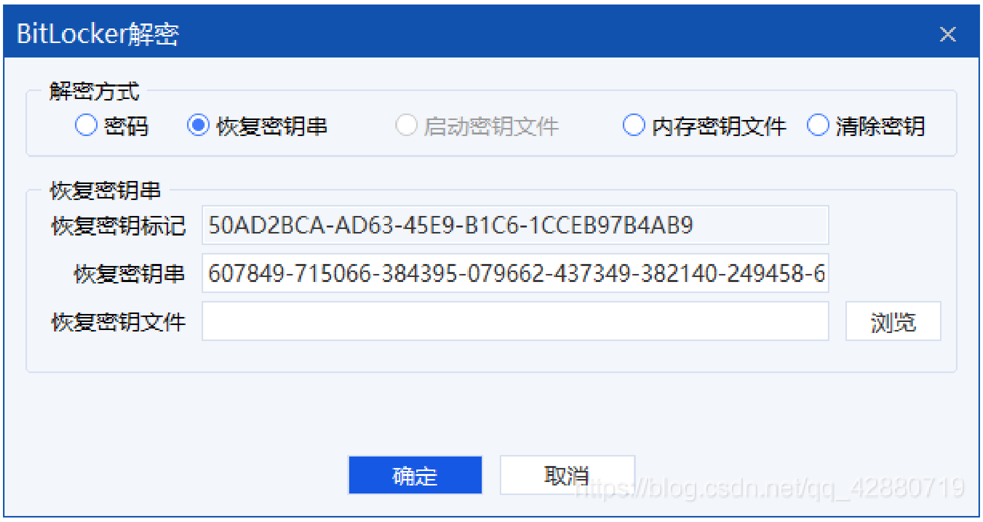
恢复密钥:

    607849-715066-384395-079662-437349-382140-249458-627066

如果以上标识符与电脑显示的标识符不匹配，则该密钥不是解锁你的驱动器的正确密钥。
请尝试其他恢复密钥，或参阅 https://go.microsoft.com/fwlink/?LinkID=260589 以获得其他帮助。

然后右击c盘，点击bitlocker解密





解密方式

○ 密码 ● 恢复密钥串 ○ 启动密钥文件 ○ 内存密钥文件 ○ 清除密钥

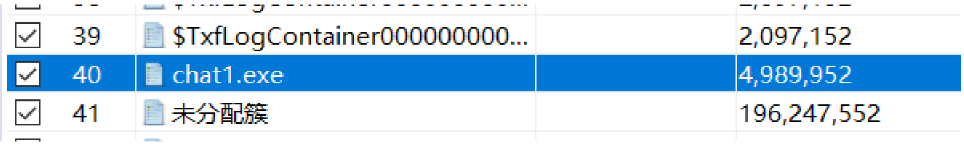
恢复密钥串

恢复密钥标记 50AD2BCA-AD63-45E9-B1C6-1CCEB97B4AB9

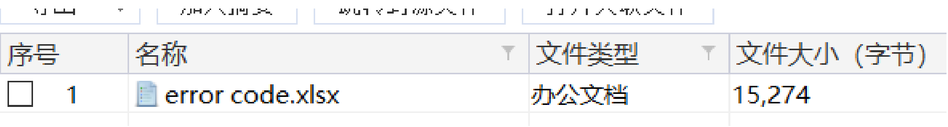
恢复密钥串 607849-715066-384395-079662-437349-382140-249458-6

恢复密钥文件 浏览

解开C盘之后，重新取证



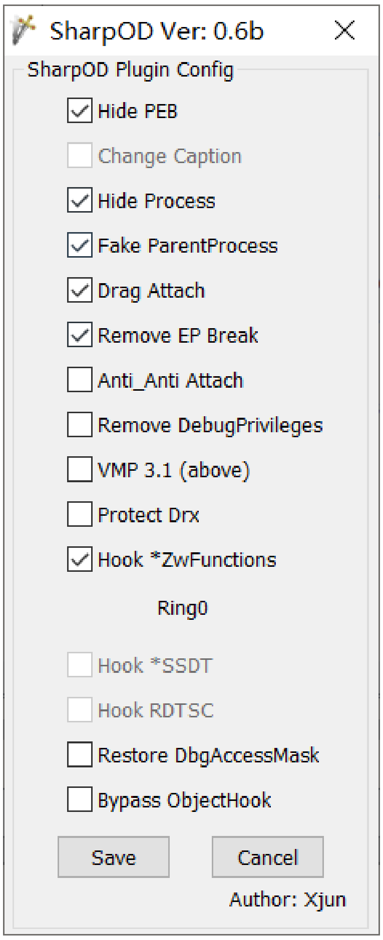
☑	39	$TxfLogContainer000000000...	2,097,152
☑	40	chat1.exe	4,989,952
☑	41	未分配簇	196,247,552



序号	名称	文件类型	文件大小（字节）
□ 1	error code.xlsx	办公文档	15,274

取出这两个文件，xlsx未发现宏，chat1.exe为关键，并且最后提示不要逆chat1.exe，更加锁定了在chat1.exe里，而且调试后发现有反调试，其实猜测flag在内存中，根据T佬说用sharpOD反反调试操作一波

首先x64dbg安装这个插件，勾选如下



SharpOD Ver: 0.6b

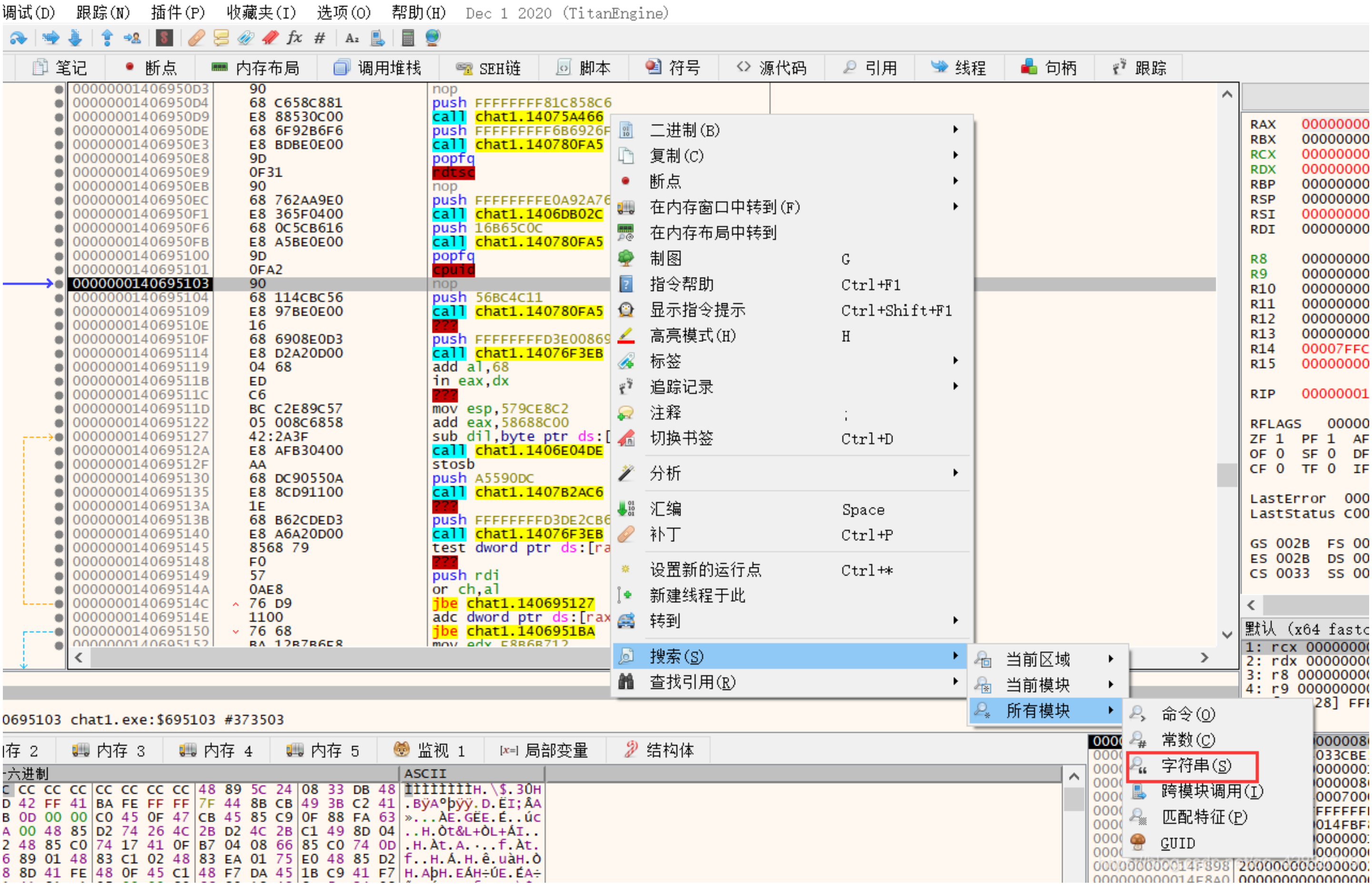
SharpOD Plugin Config

☑ Hide PEB
□ Change Caption
☑ Hide Process
☑ Fake ParentProcess
☑ Drag Attach
☑ Remove EP Break
□ Anti_Anti Attach
□ Remove DebugPrivileges
□ VMP 3.1 (above)
□ Protect Drx
☑ Hook *ZwFunctions

Ring0

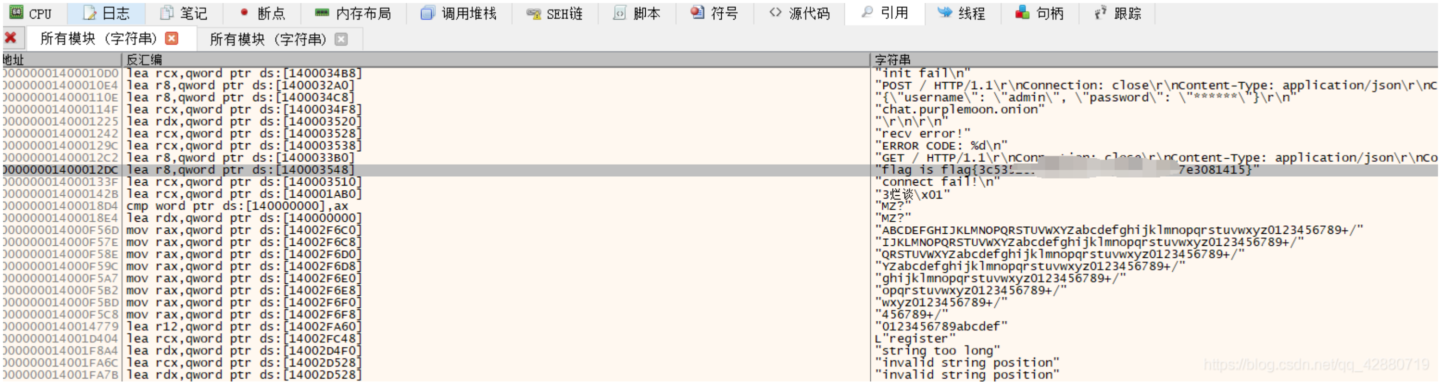
□ Hook *SSDT
□ Hook RDTSC
□ Restore DbgAccessMask
□ Bypass ObjectHook

然后运行程序，F9到达第一个断点处，此时在内存中找不到东西，发现再F9达到几处断点后程序关闭，根据正常运行的时候弹出connect fail!可以知道那4处断点时有connect fail!弹出，在最后一个断点处查看内存中的字符串（虽然之后发现第二个断点处已经有值了）



```
00000001406950D3  90                nop
00000001406950D4  68 C658C881       push FFFFFFFF81C858C6
00000001406950D9  E8 88530C00       call chat1.14075A466
00000001406950DE  68 6F92B6F6       push FFFFFFFFF6B6926F
00000001406950E3  E8 BDBE0E00       call chat1.140780FA5
00000001406950E8  9D                popfq
00000001406950E9  0F31              rdtsc
00000001406950EB  90                nop
00000001406950EC  68 762AA9E0       push FFFFFFFFE0A92A76
00000001406950F1  E8 365F0400       call chat1.1406DB02C
00000001406950F6  68 0C5CB616       push 16B65C0C
00000001406950FB  E8 A5BE0E00       call chat1.140780FA5
0000000140695100  9D                popfq
0000000140695101  0FA2              cpuid
0000000140695103  90                nop
0000000140695104  68 114CBC56       push 56BC4C11
0000000140695109  E8 97BE0E00       call chat1.140780FA5
000000014069510E  16                ???
000000014069510F  68 6908E0D3       push FFFFFFFFD3E00869
0000000140695114  E8 D2A20D00       call chat1.14076F3EB
0000000140695119  04 68             add al,68
000000014069511B  ED                in eax,dx
000000014069511C  C6                ???
000000014069511D  BC C2E89C57       mov esp,579CE8C2
0000000140695122  05 008C6858       add eax,58688C00
0000000140695127  42:2A3F           sub dil,byte ptr ds:[
000000014069512A  E8 AFB30400       call chat1.1406E04DE
000000014069512F  AA                stosb
0000000140695130  68 DC90550A       push A5590DC
0000000140695135  E8 8CD91100       call chat1.1407B2AC6
000000014069513A  1E                ???
000000014069513B  68 B62CDED3       push FFFFFFFFD3DE2CB6
0000000140695140  E8 A6A20D00       call chat1.14076F3EB
0000000140695145  8568 79           test dword ptr ds:[ra
0000000140695148  F0                ???
0000000140695149  57                push rdi
000000014069514A  0AE8              or ch,al
000000014069514C  76 D9             jbe chat1.140695127
000000014069514E  1100              adc dword ptr ds:[rax
0000000140695150  76 68             jbe chat1.1406951BA
```

```
0695103 chat1.exe:$695103 #373503
```



CPU 日志 笔记 断点 内存布局 调用堆栈 SEH链 脚本 符号 源代码 引用 线程 句柄 跟踪
所有模块 (字符串) 所有模块 (字符串)

```
地址              反汇编                                  字符串
00000001400010D0  lea rcx,qword ptr ds:[1400034B8]        "init fail\n"
00000001400010E4  lea r8,qword ptr ds:[1400032A0]         "POST / HTTP/1.1\r\nConnection: close\r\nContent-Type: application/json\r\nC
000000014000110E  lea r8,qword ptr ds:[1400034C8]         "{\"username\": \"admin\", \"password\": \"******\"}\r\n"
000000014000114F  lea rcx,qword ptr ds:[1400034F8]        "chat.purplemoon.onion"
0000000140001225  lea rdx,qword ptr ds:[140003520]        "\r\n\r\n"
0000000140001242  lea rcx,qword ptr ds:[140003528]        "recv error!"
000000014000129C  lea rcx,qword ptr ds:[140003538]        "ERROR CODE: %d\n"
00000001400012C2  lea r8,qword ptr ds:[1400033B0]         "GET / HTTP/1.1\r\nCon[redacted]ontent-Type: application/json\r\nCo
00000001400012DC  lea r8,qword ptr ds:[140003548]         "flag is flag{3c53[redacted]7e3081415}"
000000014000133F  lea rcx,qword ptr ds:[140003510]        "connect fail!\n"
000000014000142B  lea rcx,qword ptr ds:[140001AB0]        "3烂谈\x01"
00000001400018D4  cmp word ptr ds:[140000000],ax          "MZ?"
00000001400018E4  lea rdx,qword ptr ds:[140000000]        "MZ?"
000000014000F56D  mov rax,qword ptr ds:[14002F6C0]        "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
000000014000F57E  mov rax,qword ptr ds:[14002F6C8]        "IJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
000000014000F58E  mov rax,qword ptr ds:[14002F6D0]        "QRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
000000014000F59C  mov rax,qword ptr ds:[14002F6D8]        "YZabcdefghijklmnopqrstuvwxyz0123456789+/"
000000014000F5A7  mov rax,qword ptr ds:[14002F6E0]        "ghijklmnopqrstuvwxyz0123456789+/"
000000014000F5B2  mov rax,qword ptr ds:[14002F6E8]        "opqrstuvwxyz0123456789+/"
000000014000F5BD  mov rax,qword ptr ds:[14002F6F0]        "wxyz0123456789+/"
000000014000F5C8  mov rax,qword ptr ds:[14002F6F8]        "456789+/"
0000000140014779  lea r12,qword ptr ds:[14002FA60]        "0123456789abcdef"
000000014001D404  lea rcx,qword ptr ds:[14002FC48]        L"register"
000000014001F8A4  lea rdx,qword ptr ds:[14002D4F0]        "string too long"
000000014001FA6C  lea rcx,qword ptr ds:[14002D528]        "invalid string position"
000000014001FA7B  lea rdx,qword ptr ds:[14002D528]        "invalid string position"
```

最后找到flag，这样解可能是作者留下的后门解出来的