

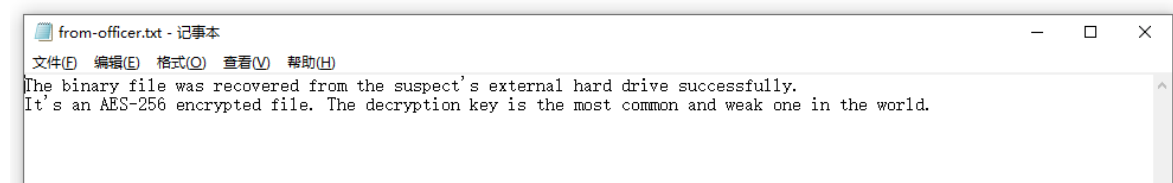
# de1ctf misc deepinreal writeup

压缩包解压得到三个文件。

名称	修改日期	类型	大小
from-officer.txt	2019/7/10 0:00	文本文档	1 KB
recovered.bin	2019/7/10 0:00	BIN 文件	29,725,505 KB
WinAES_0.2.4_x64.exe	2019/7/10 0:00	应用程序	221 KB

先看 from-officer.txt。

名称	修改日期	类型	大小
from-officer.txt	2019/7/10 0:00	文本文档	1 KB
recovered.bin	2019/7/10 0:00	BIN 文件	29,725,505 KB
WinAES_0.2.4_x64.exe	2019/7/10 0:00	应用程序	221 KB



大概意思是说，这个二进制文件是从嫌疑人的移动硬盘里恢复出来的，是一个 AES-256 加密文件，解密的密钥是世界上最常用和最弱的。

根据 officer 的提示，我们可以上网查一下世界上最常用和最弱的密码是什么。



[全部](#) [图片](#) [新闻](#) [视频](#) [购物](#) [更多](#) [设置](#) [工具](#)

找到约 82,300,000 条结果 (用时 0.37 秒)

This is a list of **the most common passwords**, according to various sources.

...  
National Cyber Security Centre.

Rank	2019
1	123456
2	123456789
3	qwerty
4	password

另外 16 行

[List of the most common passwords - Wikipedia](https://en.wikipedia.org/wiki/List_of_the_most_common_passwords)  
[https://en.wikipedia.org/wiki/List\\_of\\_the\\_most\\_common\\_passwords](https://en.wikipedia.org/wiki/List_of_the_most_common_passwords)

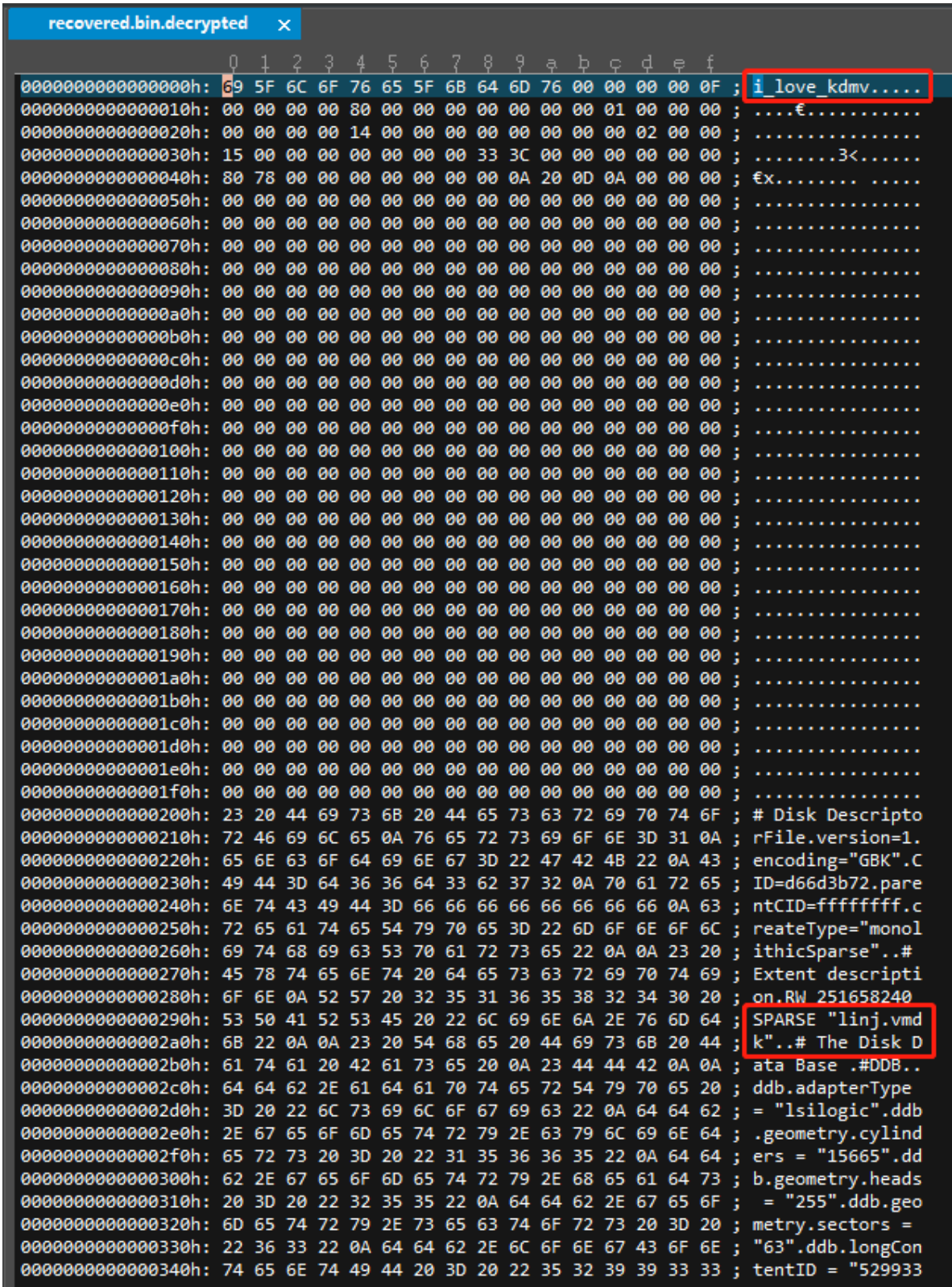
 关于这条结果的详细信息  反馈

根据维基百科的记录，2019年最常用的密码排在第一位的是 123456。

那么我们用题目所提供的加解密软件 WinAes 和密钥 123456 即可解密 recovered.bin 文件。

名称	修改日期	类型	大小
from-officer.txt	2019/7/10 0:00	文本文档	1 KB
recovered.bin	2019/7/10 0:00	BIN 文件	29,725,505 KB
recovered.bin.decrypted	2019/7/12 21:47	DECRYPTED 文件	29,725,504 KB
WinAes_0.2.4_x64.exe	2019/7/10 0:00	应用程序	221 KB

得到解密文件 recovered.bin.decrypted，很自然地想查看文件类型，就去查看一下文件的头部。



这个文件原名叫 `linj.vmdk`，是一个 `vmdk` 映像文件。它的文件头部被修改过，我们可以参照其它 `vmdk` 格式的文件头部，把头部改回正常。

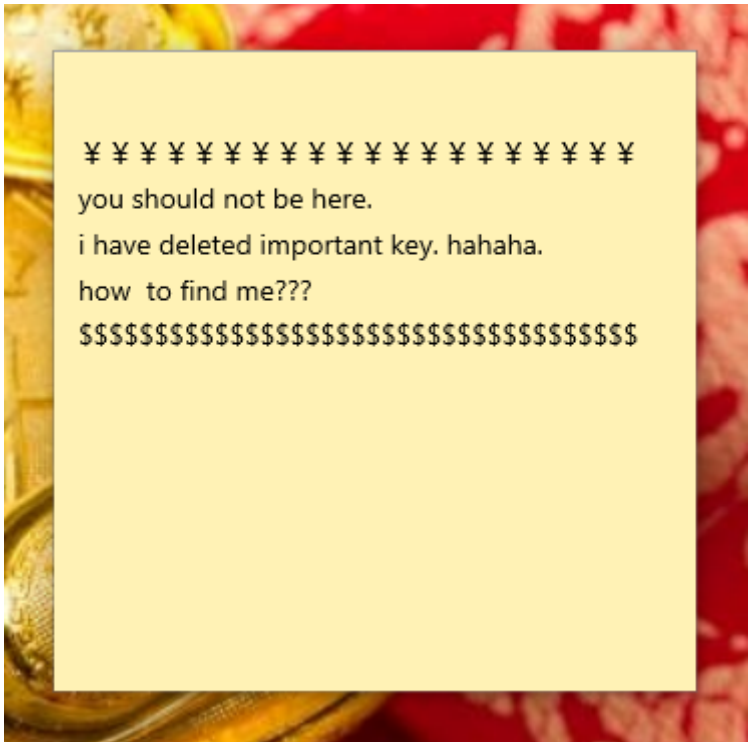
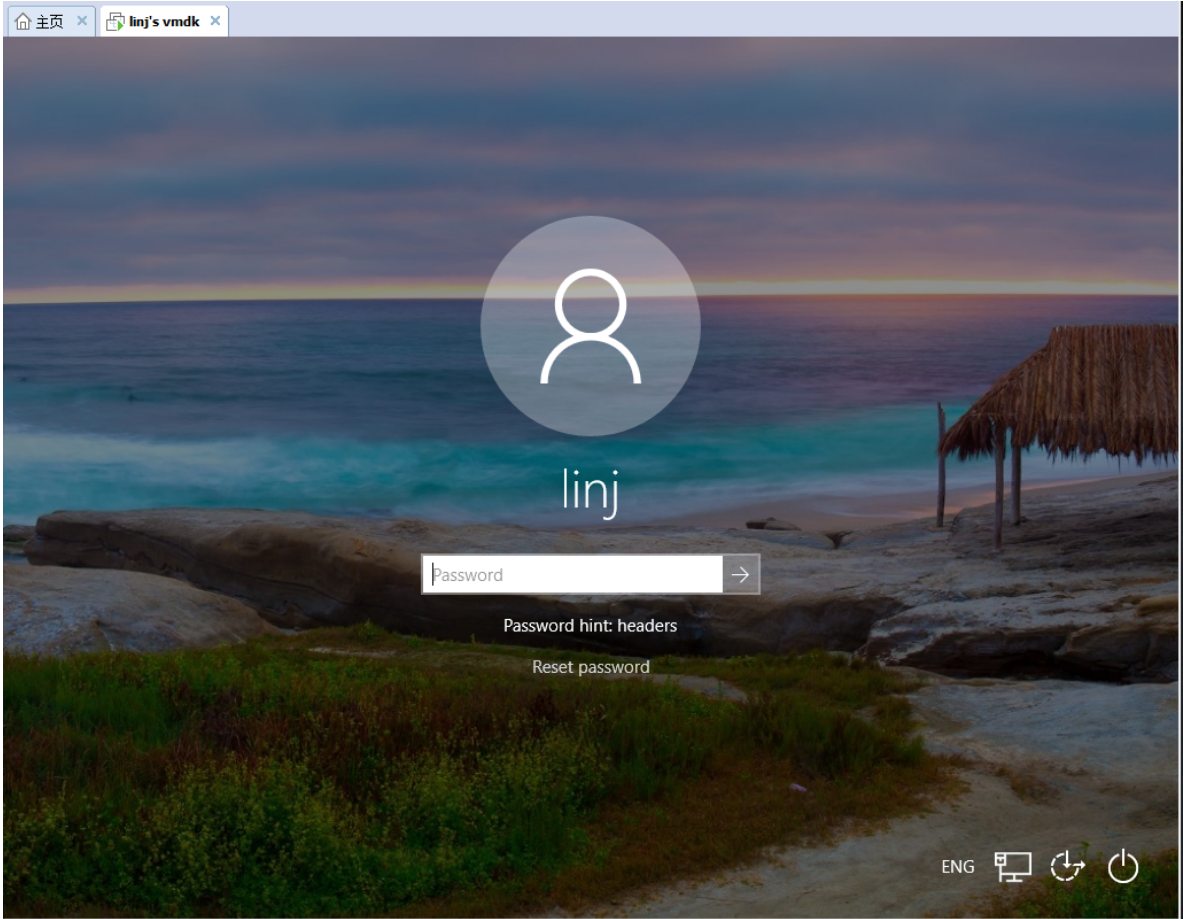
```
recovered.bin.decrypted x
0 1 2 3 4 5 6 7 8 9 a b c d e f
0000000000000000h: 4B 44 4D 56 01 00 00 00 03 00 00 00 00 0F ; KDMV.....
0000000000000010h: 00 00 00 00 80 00 00 00 00 00 00 00 01 00 00 00 ; .....
0000000000000020h: 00 00 00 00 14 00 00 00 00 00 00 00 02 00 00 00 ; .....
0000000000000030h: 15 00 00 00 00 00 00 00 33 3C 00 00 00 00 00 00 ; .....3<.....
0000000000000040h: 80 78 00 00 00 00 00 00 0A 20 0D 0A 00 00 00 00 ; €x.....
0000000000000050h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
0000000000000060h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
0000000000000070h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
0000000000000080h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
0000000000000090h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000000000000a0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000000000000b0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000000000000c0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000000000000d0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000000000000e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000000000000f0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
0000000000000100h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
0000000000000110h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
0000000000000120h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
0000000000000130h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
0000000000000140h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
0000000000000150h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
0000000000000160h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
0000000000000170h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
0000000000000180h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
0000000000000190h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000000000001a0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000000000001b0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000000000001c0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000000000001d0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000000000001e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000000000001f0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
0000000000000200h: 23 20 44 69 73 68 20 44 65 73 63 72 69 70 74 6F ; # Disk Descripto
0000000000000210h: 72 46 69 6C 65 0A 76 65 72 73 69 6F 6E 3D 31 0A ; rFile.version=1.
0000000000000220h: 65 6E 63 6F 64 69 6E 67 3D 22 47 42 4B 22 0A 43 ; encoding="GBK".C
0000000000000230h: 49 44 3D 64 36 36 64 33 62 37 32 0A 70 61 72 65 ; ID=d66d3b72.pare
0000000000000240h: 6E 74 43 49 44 3D 66 66 66 66 66 66 66 66 66 66 ; ntCID=ffffffff.c
0000000000000250h: 72 65 61 74 65 54 79 70 65 3D 22 6D 6F 6E 6F 6C ; reateType="monol
0000000000000260h: 69 74 68 69 63 53 70 61 72 73 65 22 0A 0A 23 20 ; ithicSparse"..#
0000000000000270h: 45 78 74 65 6E 74 20 64 65 73 63 72 69 70 74 69 ; Extent descripti
0000000000000280h: 6F 6E 0A 52 57 20 32 35 31 36 35 38 32 34 30 20 ; on.RW 251658240
0000000000000290h: 53 50 41 52 53 45 20 22 6C 69 6E 6A 2E 76 6D 64 ; SPARSE "linj.vmd
00000000000002a0h: 6B 22 0A 0A 23 20 54 68 65 20 44 69 73 68 20 44 ; k"..# The Disk D
00000000000002b0h: 61 74 61 20 42 61 73 65 20 0A 23 44 44 42 0A 0A ; ata Base .#UDB..
00000000000002c0h: 64 64 62 2E 61 64 61 70 74 65 72 54 79 70 65 20 ; ddb.adapterType
00000000000002d0h: 3D 20 22 6C 73 69 6C 6F 67 69 63 22 0A 64 64 62 ; = "lsilogic".ddb
00000000000002e0h: 2E 67 65 6F 6D 65 74 72 79 2E 63 79 6C 69 6E 64 ; .geometry.cylin
00000000000002f0h: 65 72 73 20 3D 20 22 31 35 36 36 35 22 0A 64 64 ; ers = "15665".dd
0000000000000300h: 62 2E 67 65 6F 6D 65 74 72 79 2E 68 65 61 64 73 ; b.geometry.heads
0000000000000310h: 20 3D 20 22 32 35 35 22 0A 64 64 62 2E 67 65 6F ; = "255".ddb.geo
0000000000000320h: 6D 65 74 72 79 2E 73 65 63 74 6F 72 73 20 3D 20 ; metry.sectors =
```

这时候就是一个正常的 `vmdk` 文件了。我们可以使用 `开源取证工具` 或者 `商业取证工具` 进行 `静态取证`，也可以使用 `专业仿真软件` 或者 `VMware` 进行 `动态取证`。

我这里使用 `取证大师` 进行 `静态取证`，使用 `VMware` 进行 `动态取证`。

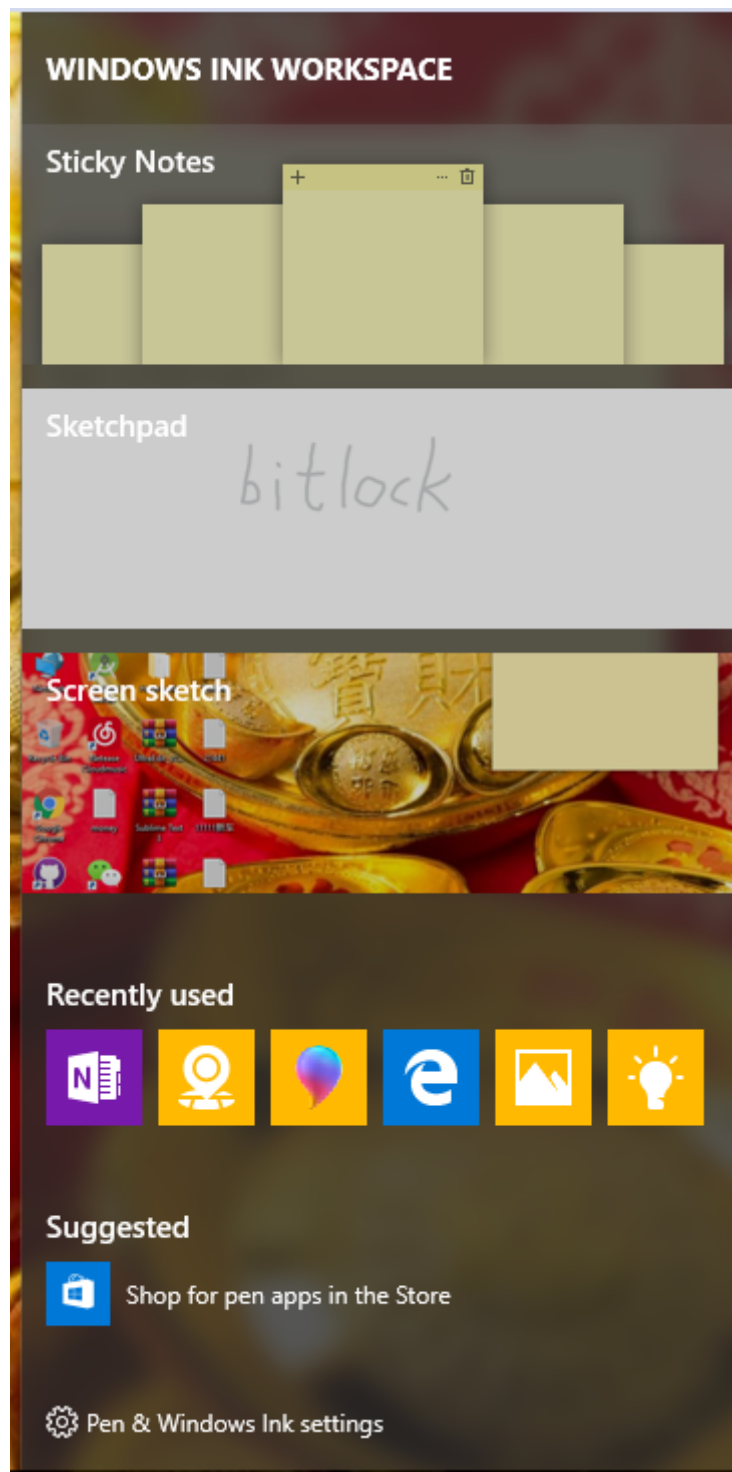
在 `VMware` 中加载这个镜像文件，开机后登录系统需要密码，密码提示 `headers`。刚才我们在文件头处看到了 `i_love_kdmv`，这个就是系统登录的密码。





登录后，在桌面右上角看到一张便签，大概意思是，“你不应该到这里来，我已经删除了一条重要的钥匙，怎么找到我？”。

这里的“我”指的是“便签”。嫌疑人很可能使用系统自带的功能进行信息的隐藏。我们可以先找到 windows 10 下创建标签的方式，就是按下 win+W 键。

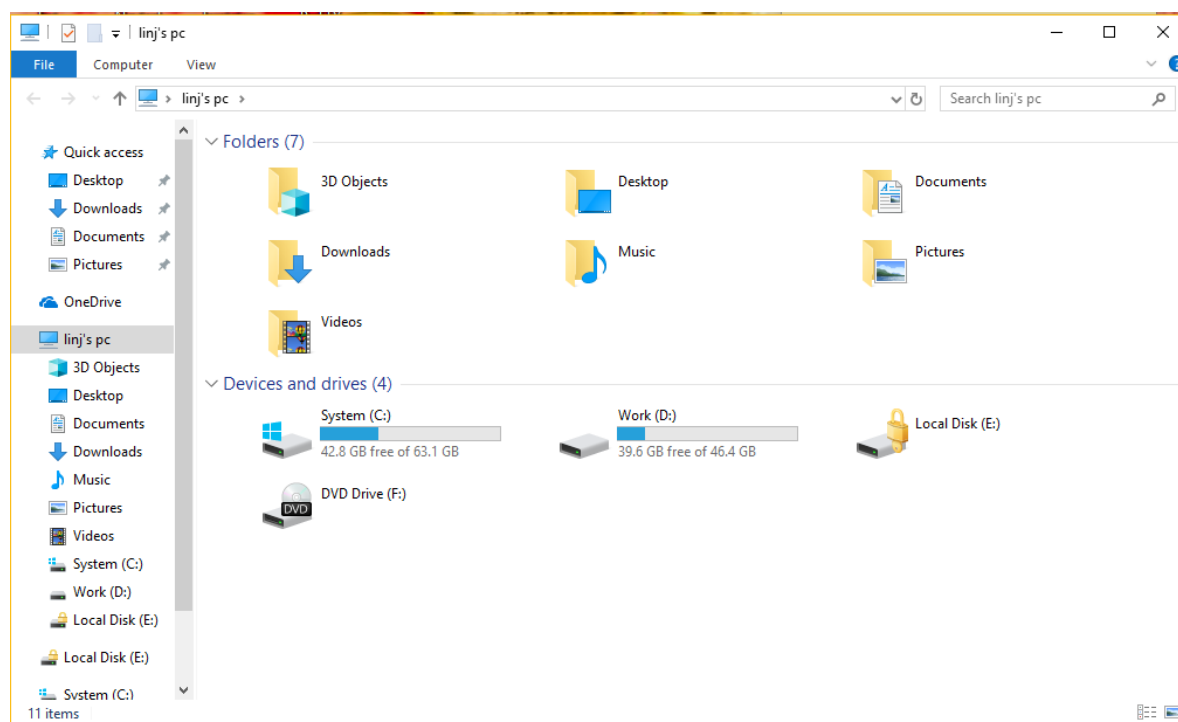


从右边弹出的侧菜单栏可以看到，`sketchpad` 功能处写着 `bitlock`，点进去看看。

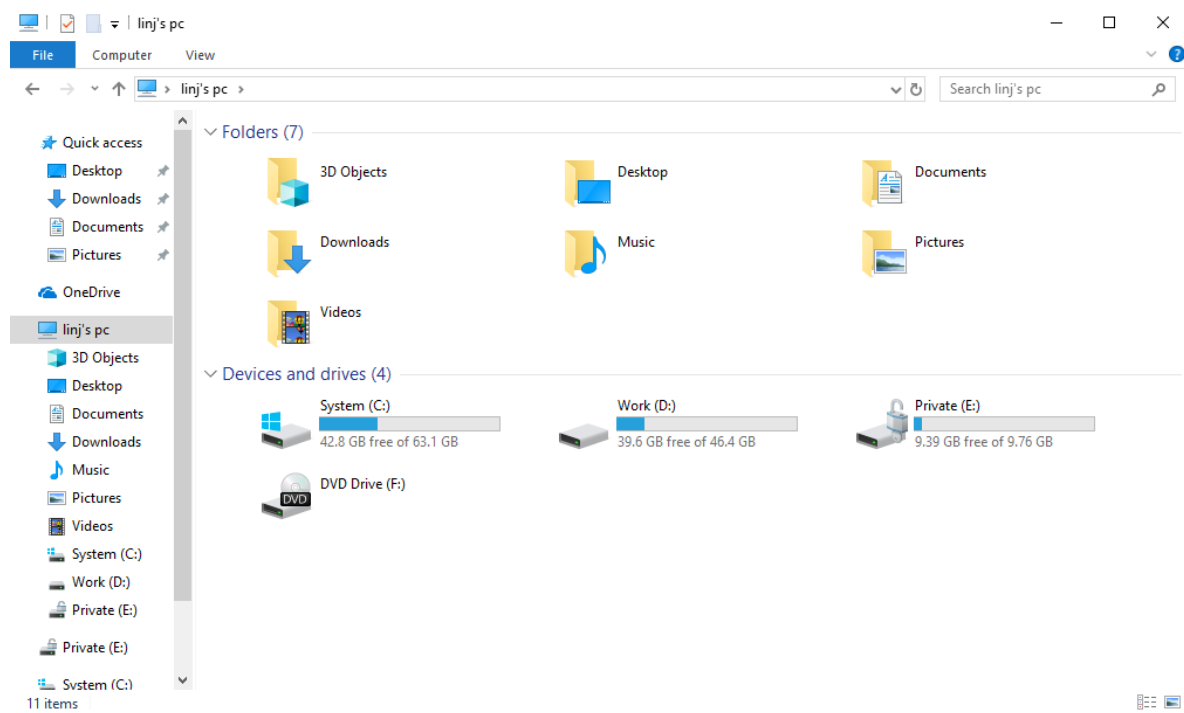
linj920623!@#

bitlock

可以看到 bitlocker 的密码，linj920623!@#，系统中确实存在一个 bitlocker 的加密盘。



使用密码进行解密，可以成功解开加密盘。



加密盘里有两个值得留意的文件。

Name	Date modified	Type	Size
EnMicroMsg	7/9/2019 9:38 PM	Data Base File	199,633 KB
ethpass.dict	7/9/2019 9:35 PM	DICT File	59 KB
Skype	7/9/2019 9:37 PM	Text Document	1 KB
UTC--2019-07-09T21-31-39.077Z--266ed..	7/9/2019 9:32 PM	077Z--266ED8970...	1 KB
WeChat	7/9/2019 9:39 PM	Text Document	1 KB

一个是数字货币加密钱包文件，另一个是密码字典。这可能是嫌疑人用来进行资金流通的数字货币钱包。

我们尝试写个脚本，使用密码字典对加密钱包文件进行暴力破解。

```
import eth_keyfile
import json

fp = open('ethpass.dict', 'r')
wallet = json.loads(open('UTC--2019-07-09T21-31-39.077Z--266ed8970d4713e8f2701cbe137bda2711b78d57', 'r').read())

while True:
    try:
        password = fp.readline().strip().encode('ascii')
        if len(password) <= 0:
            print("password not found")
            break
    except:
        continue
    try:
        result = eth_keyfile.decode_keyfile_json(wallet, password)
    except:
        continue
    print(password)
```

```
print(result)
break
```

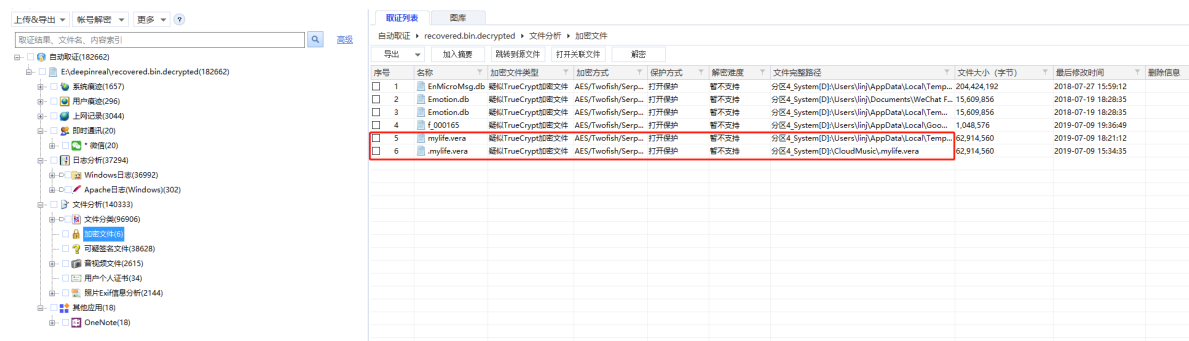
```
# root @ sec in ~ [23:12:35]
$ py3 brute.py
b'nevada'
b'VeraCrypt Pass: V3Ra1sSe3ure2333'

# root @ sec in ~ [23:12:40]
$
```

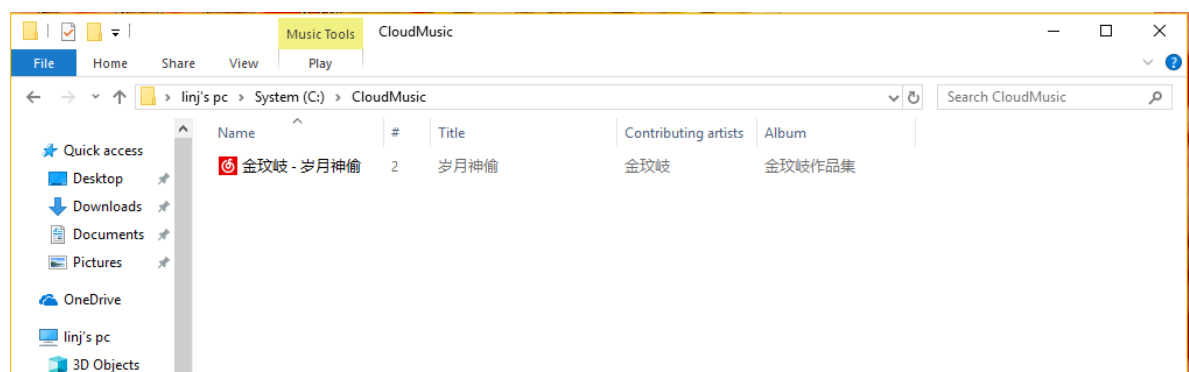
暴力破解可以得到结果，加密钱包密码为 `nevada`，钱包私钥为 `VeraCrypt Pass: V3Ra1sSe3ure2333`。

私钥提示我们有一个 `veraCrypt` 加密的容器，它的加密密码为 `V3Ra1sSe3ure2333`。

那么我们需要先找到这个容器文件。这里可以使用全盘搜索包含特定字串的方法，找到这个加密容器文件。我这里使用 [取证大师](#) 进行取证，直接在 [加密文件](#) 处可以找到这个文件。

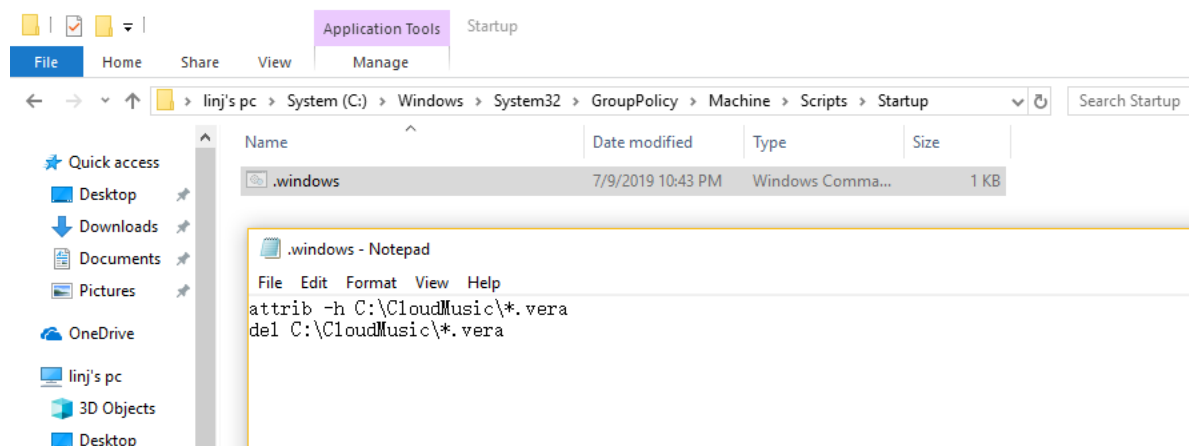


序号	名称	加密文件类型	加密方式	保护方式	解密难度	文件完整路径	文件大小 (字节)	最后修改时间	删除信息
1	Emotion.db	疑似TrueCrypt加密文件	AES/Twofish/Serp...	打开保护	暂不支持	分区4_System(D:)Users\lin\AppData\Local\Temp...	204,424,192	2018-07-27 15:59:12	
2	Emotion.db	疑似TrueCrypt加密文件	AES/Twofish/Serp...	打开保护	暂不支持	分区4_System(D:)Users\lin\Documents\WeChat F...	15,609,856	2018-07-18 18:28:35	
3	Emotion.db	疑似TrueCrypt加密文件	AES/Twofish/Serp...	打开保护	暂不支持	分区4_System(D:)Users\lin\AppData\Local\Temp...	15,609,856	2018-07-18 18:28:35	
4	f_000165	疑似TrueCrypt加密文件	AES/Twofish/Serp...	打开保护	暂不支持	分区4_System(D:)Users\lin\AppData\Local\Goo...	1,048,576	2019-07-09 19:36:49	
5	.mylife.vera	疑似TrueCrypt加密文件	AES/Twofish/Serp...	打开保护	暂不支持	分区4_System(D:)Users\lin\AppData\Local\Temp...	62,914,560	2019-07-09 18:21:12	
6	.mylife.vera	疑似TrueCrypt加密文件	AES/Twofish/Serp...	打开保护	暂不支持	分区4_System(D:)CloudMusic\mylife.vera	62,914,560	2019-07-09 15:34:35	



可是在 `vmware` 相对应的路径下找不到这个文件，想起便签处的提示，可能在系统加载的时候该文件被删除了。

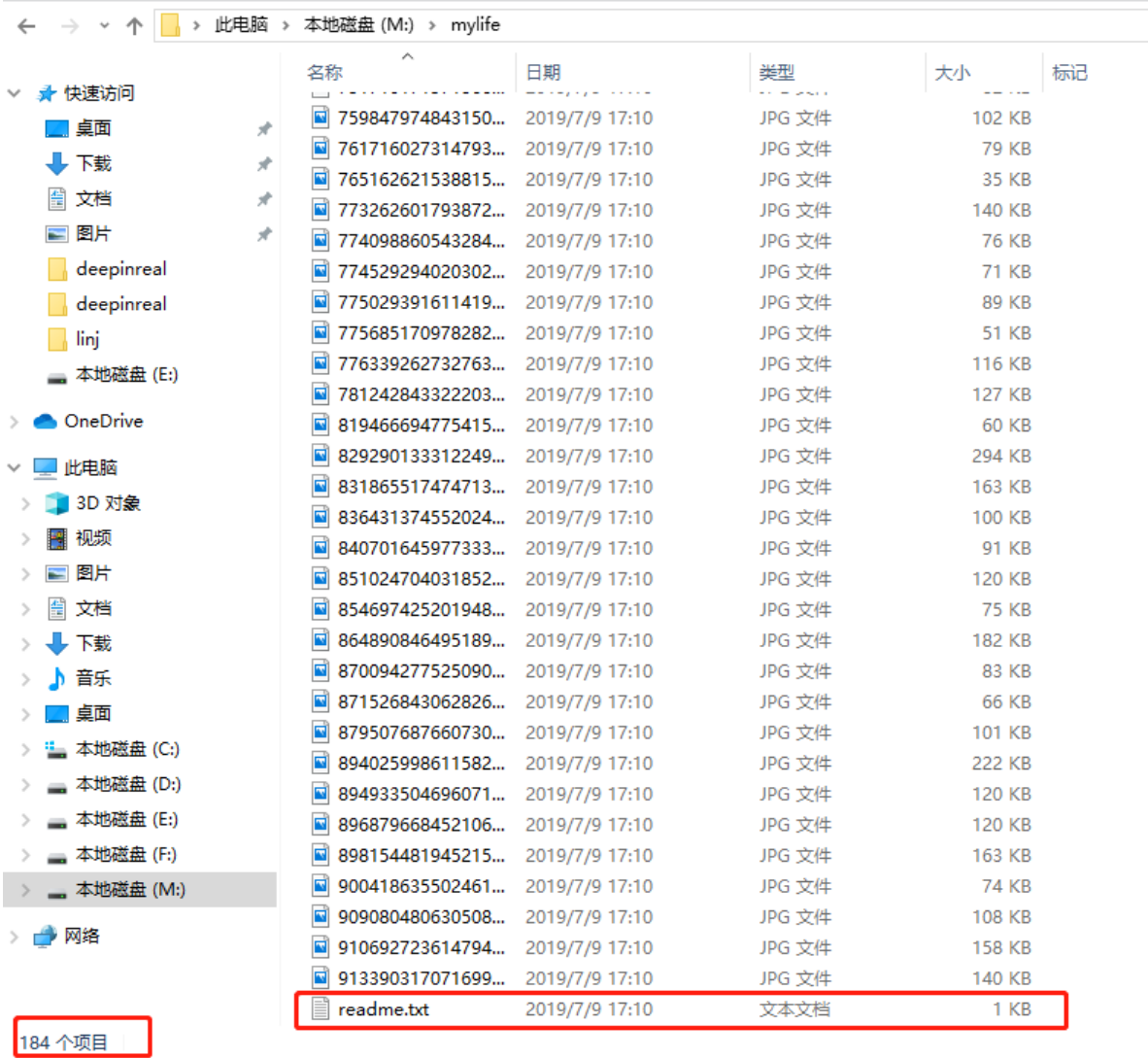
我们在系统启动项处，找到一个自动删除 `.mylife.vera` 文件的隐藏脚本文件。嫌疑人故意设置了一个简易的开机自删除功能。





那么我们可以直接在取证大师中导出该文件，也可以从系统盘的用户缓存目录下找到该文件。

使用 VeraCrypt 和之前找到的密码 V3Ra1sSe3ure2333 进行解密并挂载。



我们可以找到看到加密容器内，一共有 184 个文件，有一堆生活照，还有一个 readme 文件。



readme 文件提示这里有 185 个文件，其中 183 张照片是我的生活照，所以必然有一个文件被隐藏了。

这个文件系统为 NTFS，想起嫌疑人可能使用 NTFS 交换数据流 的方式进行文件隐藏。


在 cmd 下使用 dir /r 命令可以看到隐藏文件 528274475768683480.jpg:k3y.txt:\$DATA。

```

2019/07/09 17:10 134,022 517995430853899641.jpg
2019/07/09 17:10 111,028 5184598446666655.jpg
2019/07/09 17:10 138,771 522830085706679055.jpg
2019/07/09 17:10 84,255 523762286177219113.jpg
2019/07/09 17:10 79,673 528274475768683480.jpg
2019/07/09 17:10 17 528274475768683480.jpg:k3y.txt:$DATA
2019/07/09 17:10 132,216 531121315745995228.jpg
2019/07/09 17:10 105,379 532538630484487603.jpg
2019/07/09 17:10 84,620 534199954597339976.jpg
2019/07/09 17:10 91,414 542148972683110018.jpg
2019/07/09 17:10 81,449 543664254184297349.jpg
2019/07/09 17:10 96,119 546338842913910008.jpg

```

使用 `notepad 528274475768683480.jpg:k3y.txt` 命令，直接使用记事本打开被隐藏的文件。

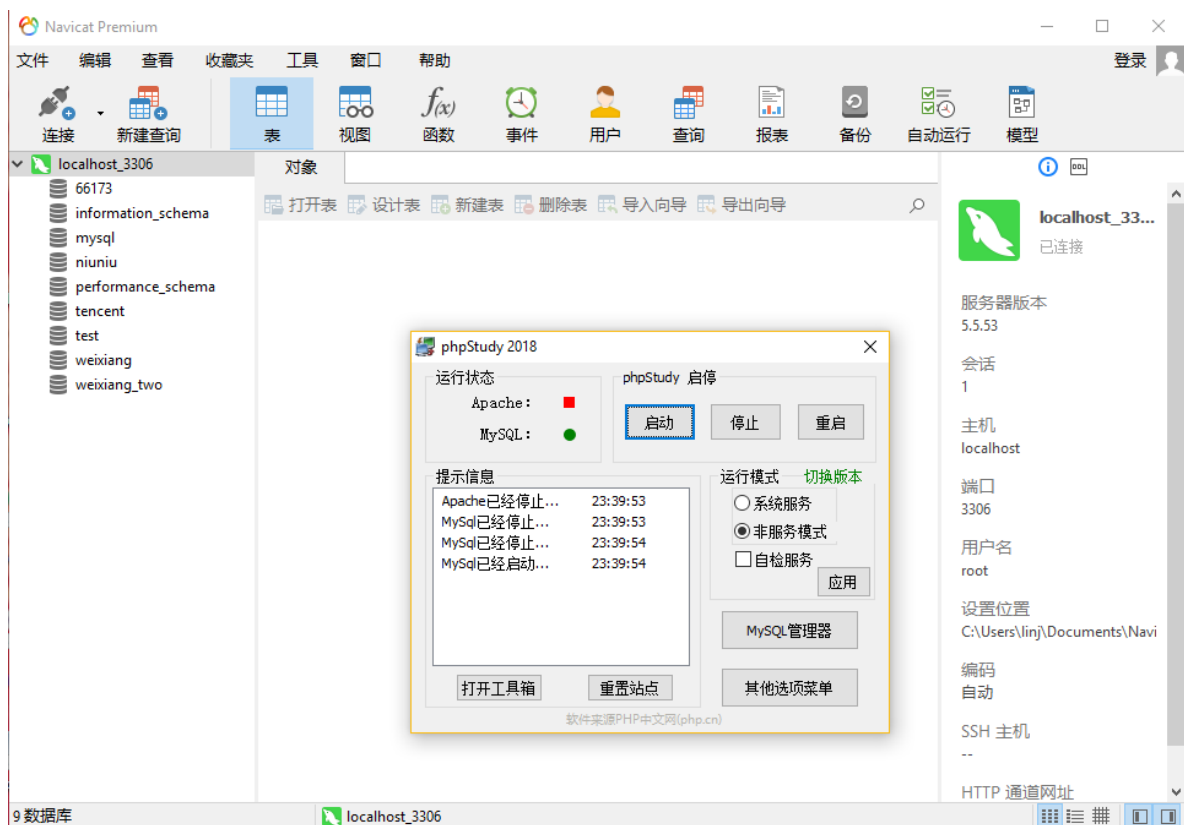
 528274475768683480.jpg:k3y.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

F1a9ZiPInD6TABaSE

可以得到一串密码 `F1a9ZiPInD6TABaSE`，并且根据密码的提示，`flag.zip` 文件在数据库里。嫌疑人可能把重要文件存放在电脑的数据库里。

想起嫌疑人的电脑装有 `phpstudy` 和 `Navicat`，直接启动 `mysql`，使用 `Navicat` 查看数据库。



看到几个数据库的名称，与 `bitlocker` 加密盘下 `gambling` 文件夹里的几个 `.sql` 文件名一致。

linj's pc > Private (E:) > gambling

	Name	Date modified	Type	Size
ss	66173.sql	7/9/2019 8:40 PM	SQL File	80,566 KB
	niuniu.sql	7/9/2019 8:40 PM	SQL File	56,838 KB
js	tencent.sql	7/9/2019 8:40 PM	SQL File	6,084 KB
its	weixiang.sql	7/9/2019 8:40 PM	SQL File	586 KB
	weixiang_two.sql	7/9/2019 8:40 PM	SQL File	277 KB

那么我们可以比较 .sql 文件里的数据与数据库里的数据，找到数据库 tencent 里多了一张表 auth\_secret。

localhost\_3306

- 66173
- information\_schema
- mysql
- niuniu
- performance\_schema
- tencent
  - 表
  - 视图
  - 函数
  - 事件

对象: auth\_secret @tencent (localho...

开始事务 | 文本 | 筛选 | 排序 | 导入 | 导出

file

UEsDBBQACQBJABeu6U7Td9bwRgAAACgAAAAIAAsAZmxhZy50eHQBMQcAAQBBRQMIAC3j

字段名为 file，字段值是一串 base64 编码字符串。

导出解码，转换为二进制文件，得到一个 zip 文件。

flag.zip

文件(F) 命令(C) 工具(S) 收藏夹(O) 选项(N) 帮助(H)

添加 解压到 测试 查看 删除 查找 向导 信息 扫描病毒 注释 自解压格式

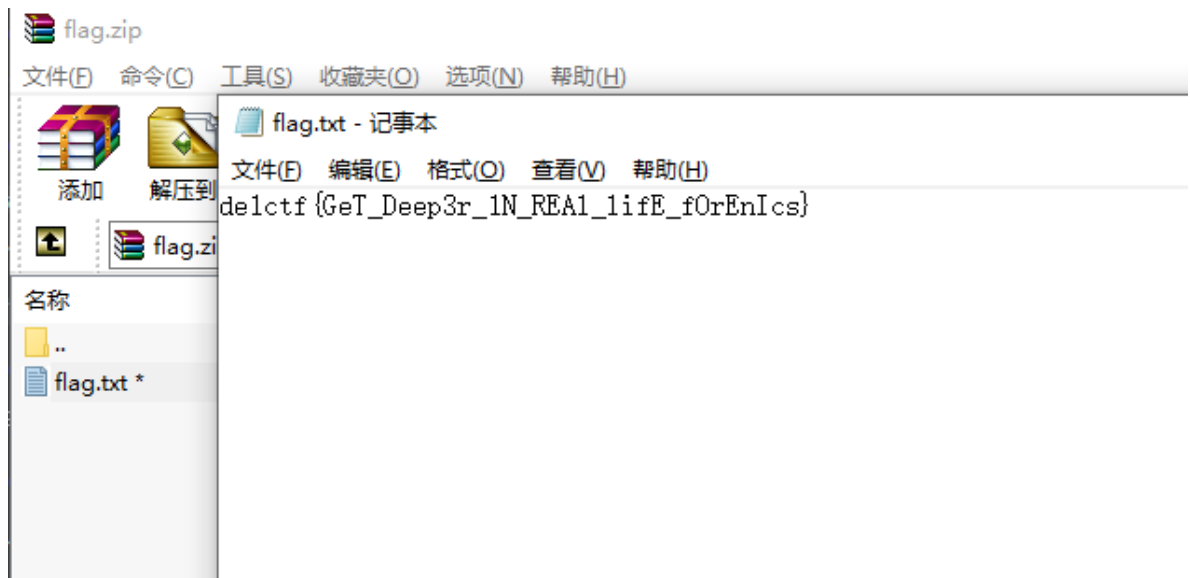
flag.zip - ZIP 压缩文件, 解包大小为 40 字节

名称	大小	压缩后大小	类型
..			本地磁盘
flag.txt *	40	70	文本文档

This is a Real Flag File.  
Find the fucking password lol

压缩包注释里提示，“这是一个真正的flag文件”，需要找到密码解开。

我们用之前找到的密码 F1a9ZiPiND6TABaSE，解开 flag.txt 文件。



成功找到嫌疑人隐藏的重要信息。

Flag: `de1ctf{GeT_Deep3r_1N_REAL_lifE_f0rEnIcs}`