

# WMCTF2021 WP misc

原创

是Mumuzi 于 2021-08-31 00:08:12 发布 917 收藏 2

分类专栏: [笔记](#)

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/qq\\_42880719/article/details/120000111](https://blog.csdn.net/qq_42880719/article/details/120000111)

版权



[笔记](#) 专栏收录该内容

23 篇文章 6 订阅

订阅专栏

唉因为太菜本来不想写的 (其实是因为都只能做一半导致太水了不好写)

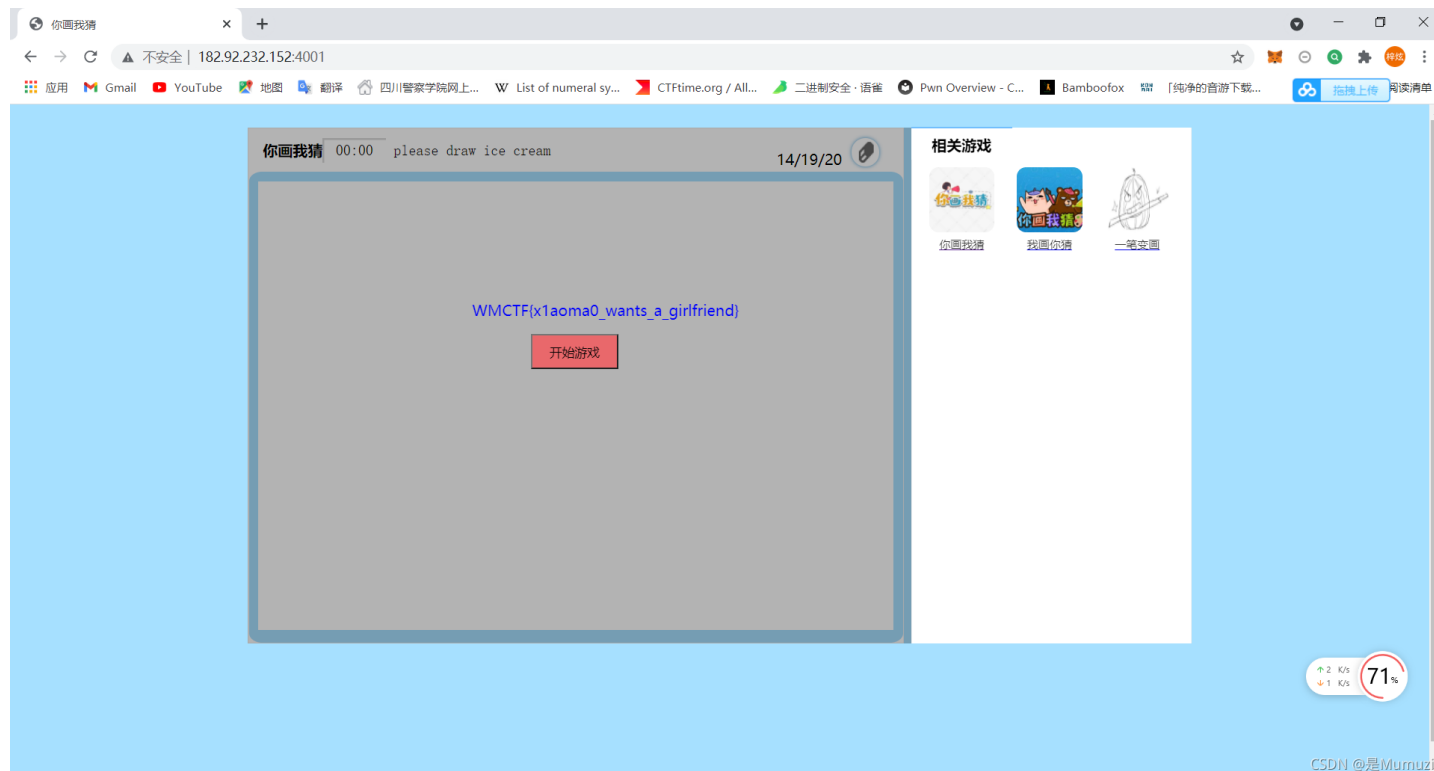
复现了一下取证还是写一下吧

## Checkin

ctrl+c ctrl+v enter

你画我猜

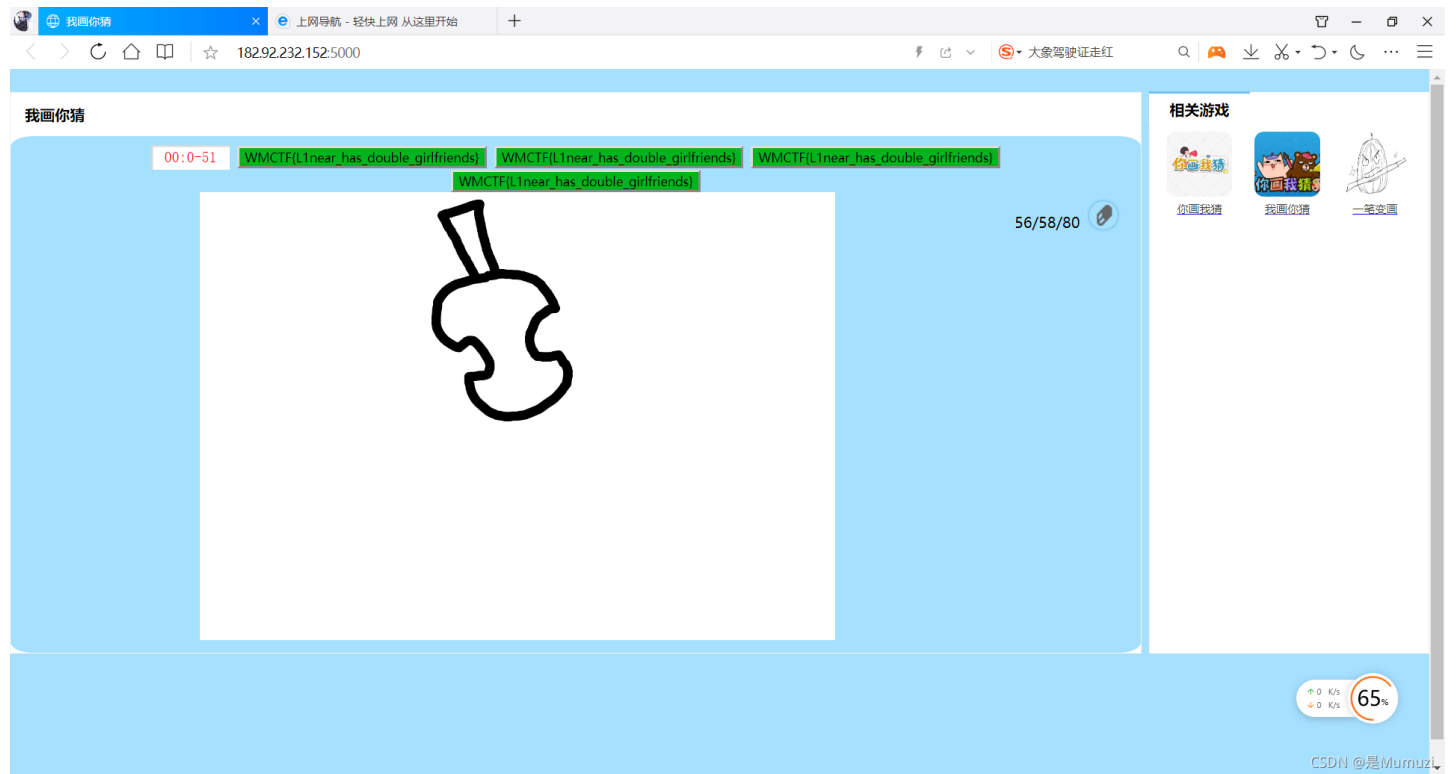
画! 都可以画!



```
WMCTF{x1aoma0_wants_a_girlfriend}
```

# 我画你猜

猜！都可以猜！



WMCTF{L1near\_has\_double\_girlfriends}

## LOGO

LOGO!都可以LOGO!

首先nc上去，看到WM标志



为了要把图给画下来，当然是重定向啦

```
nc 118.190.157.196 10001 > 1.txt
```

打开之后发现每次都是5个参数

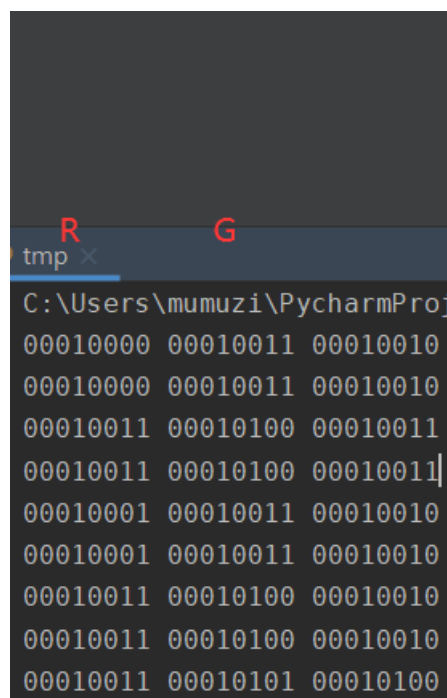
例如[48;2;23;21;24m

并发现前面两个都不变，因此推断出RGB为后三个参数，即(23,21,24)

因此就可以写一个脚本，将其画下来。

但是画下来之后发现，图片被拉宽了。并且有明显的LSB隐写，但是就是找不到flag

于是准备将值打印出来



```
C:\Users\mumuzi\PycharmPro
00010000 00010011 00010010
00010000 00010011 00010010
00010011 00010100 00010011
00010011 00010100 00010011
00010001 00010011 00010010
00010001 00010011 00010010
00010011 00010100 00010010
00010011 00010100 00010010
00010011 00010101 00010100
```

发现一个问题，每两行的值都是一样的，再结合nc上面输出的是正方形，由此可推断，原图是256256，而并非256512

于是每隔一个画图，最后查看LSB发现flag

一把梭脚本：

```

#import os
#os.system("nc 118.190.157.196 10001 > 1.txt") #Linux系统下取消注释，运行直接出

from PIL import Image
pic = Image.new('RGB', (256, 256))
f = open('1.txt', 'r').readlines()
for i in range(len(f)):
    f[i] = f[i].split(' ')
    for j in range(1, len(f[i])):
        f[i][j] = f[i][j].split(';')
for i in range(256):
    for j in range(1, 257):
        pic.putpixel((j-1, i), (int(f[i][j*2][2]), int(f[i][j*2][3]), int(f[i][j*2][4][:-3])))
pic.save('WM.png')

pic1 = Image.open("WM.png")
w = pic1.size[0]
flag = ''
for i in range(w):
    g = pic.getpixel((i, 0))
    R, G, B = bin(g[0])[2:].zfill(8), bin(g[1])[2:].zfill(8), bin(g[2])[2:].zfill(8)
    li = [R, G, B]
    for color in li:
        flag += color[7]
tmp = ''
for k in range(len(flag)):
    tmp += flag[k]
    if len(tmp) == 8:
        print(chr(int(tmp, 2)), end='')
    tmp = ''

```

```

mumuzi@kali:~/桌面$ python3 LOGO.py
WMCTF{dba5d43b6c0035d8559437ed34f2f8fb}'>ÉiηG³·PηKmZâr$(bI $ηÛbIØI;v

```

WMCTF{dba5d43b6c0035d8559437ed34f2f8fb}

## Questionnaire

WMCTF{s33\_u\_in\_2022}

## Flag Thief (复现!)

WP写的真的很详细了

题!好!考点:呜呜呜忘干净了

【卑微各群群欺】肘子炫之看守所分炫 2021/7/6 16:22:54

V3geD4g 2021/7/6 16:00:37  
<https://www.freebuf.com/articles/web/279561.html>  
好像又是一个出题思路



@V3geD4g 确定

【卑微各群群欺】肘子炫之看守所分炫 2021/7/6 16:22:56

确实

CSDN @是Mumuzi

<https://www.freebuf.com/articles/web/279561.html> (可恶居然押中题)

有关此类取证可能碰到的知识点(tokeii): <https://blog.csdn.net/u010418732/article/details/120009187>

根据提示,可以猜测是RDP,于是去仿真查看日志管理器-安全,id

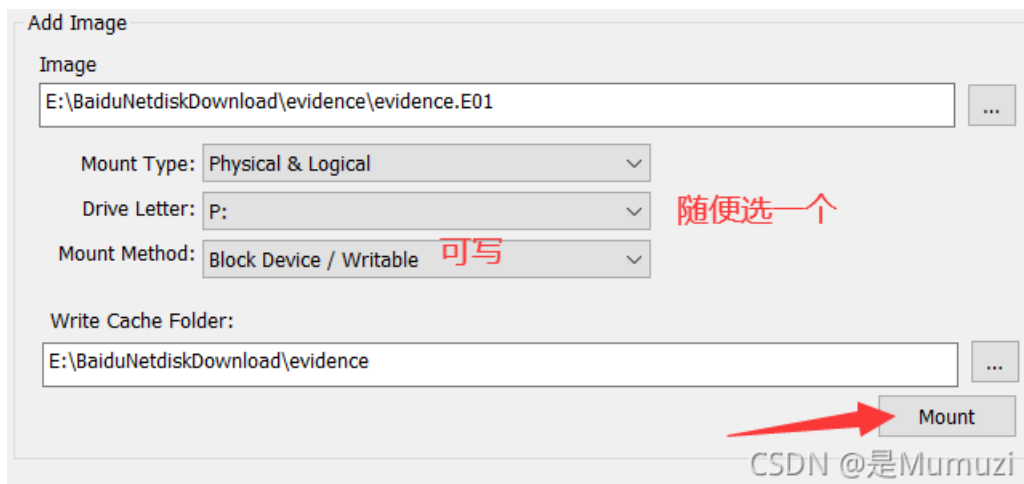
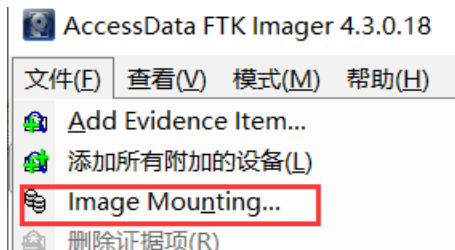
4624d的日志,但是啥都没翻到。然后中间阿巴阿巴做了一堆事情啥也没找到惹。

好!

复现就只知道是关于上面那个链接,然后就可以做完惹

路径%USERPROFILE%\AppData\Local\Microsoft\Terminal Server Client\Cache

用FTK挂载打开



本地磁盘 (Y:) > 用户 > WMCTF > AppData > Local > Microsoft > Terminal Server Client > Cache

名称	修改日期	类型	大小
----	------	----	----

bcache24.bmc	2021/8/17 18:50	BMC 文件	0 KB
Cache0000.bin	2021/8/17 18:51	BIN 文件	32,471 KB
Cache0001.bin	2021/8/17 17:37	BIN 文件	53,994 KB
Cache0002.bin	2021/8/17 17:39	BIN 文件	74,243 KB

CSDN @是Mumuzi

这三个bin，弄出来。

你跟着那个链接，一个个去弄，那肯定不现实，那好，那咱就去百度

## RDP bmc

[Q 网页](#)
[资讯](#)
[视频](#)
[图片](#)
[知道](#)
[文库](#)
[贴贴吧](#)
[地址](#)

百度为您找到相关结果约1,070,000个

搜索工

### [GitHub - ANSSI-FR/bmc-tools: RDP Bitmap Cache parser](#)

查看此网页的中文翻译, 请点击 [翻译此页](#)

RDP Bitmap Cache parser.Input**bmc-tools** processes **bcache\*.bmc** and **cache???.bin** files f  
 nside Windows user profiles.Usage:**/bmc-tools.py** [-h] -s SRC -d DEST [-c]

CSDN @是Mumuzi

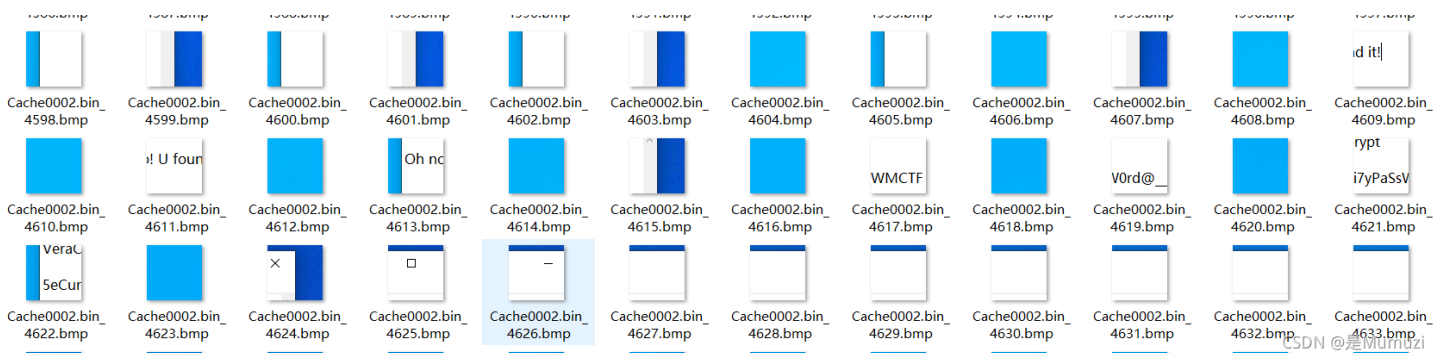
工具一用，完美

首先把那3个bin放在一个文件夹下，比如abc

然后

```
python3 bmc-tools.py -s abc -d abc (-v)
```

然后得到10000多张图片，其中！



啊，我一看，原来是密码

仔细一看，是VeraCrypt的密码

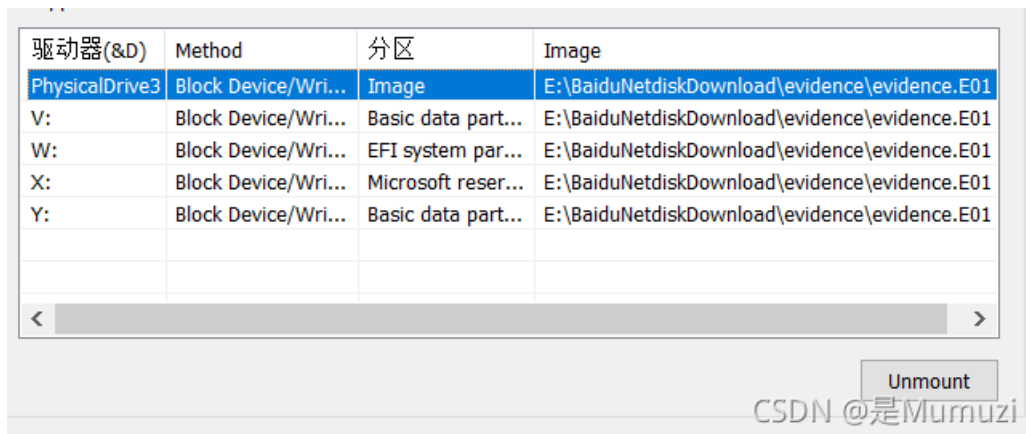
再仔细一看，密码是

```
5eCuri7yPaSsW0rd@_WMCTF
```

好！那VeraCrypt的东西在哪。当然是用取证大师来找了

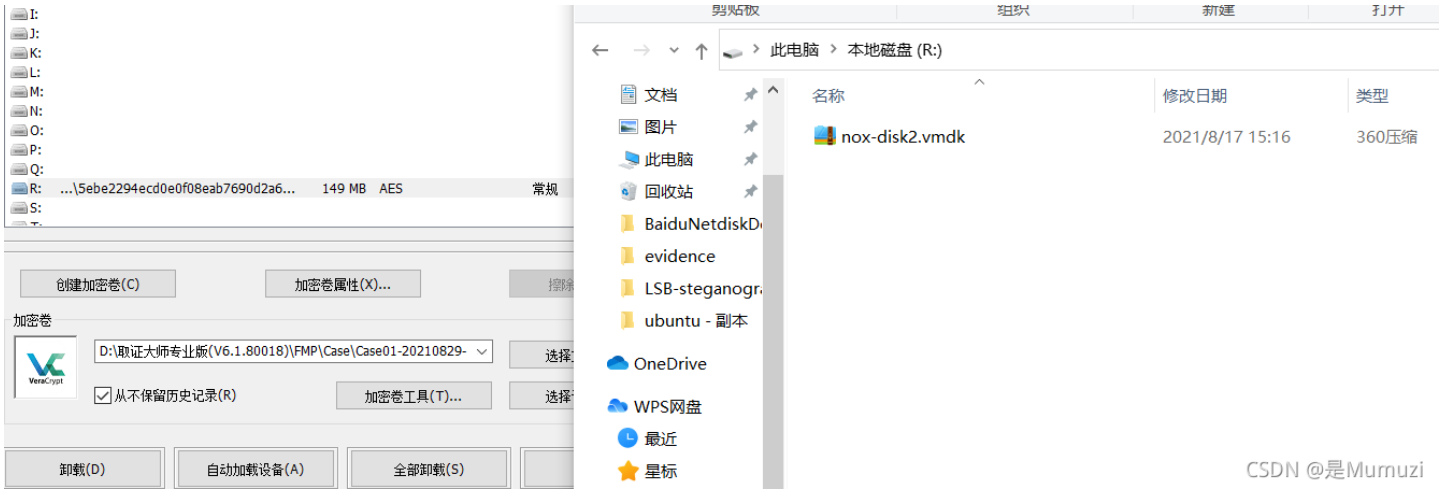
<input type="checkbox"/>	1	edbres00002.jrs	疑似TrueCrypt加密文件	AES/TwoFish/Serp...	打开保护	暂不支持	分区4_本地磁盘[D]:\Users\WMCTF\AppData\Loca...	524,288
<input checked="" type="checkbox"/>	2	5ebe2294ecd...	疑似TrueCrypt加密文件	AES/TwoFish/Serp...	打开保护	暂不支持	分区4_本地磁盘[D]:\Users\WMCTF\AppData\Loca...	157,286,400
<input type="checkbox"/>	3	startscreen.vid	疑似TrueCrypt加密文件	AES/TwoFish/Serp...	打开保护	暂不支持	分区4_本地磁盘[D]:\Program Files\WindowsApps\...	2,457,600
<input type="checkbox"/>	4	edbres00001.jrs	疑似TrueCrypt加密文件	AES/TwoFish/Serp...	打开保护	暂不支持	分区4_本地磁盘[D]:\Users\Administrater\AppData...	524,288
<input type="checkbox"/>	5	edbres00002.jrs	疑似TrueCrypt加密文件	AES/TwoFish/Serp...	打开保护	暂不支持	分区4_本地磁盘[D]:\Users\Administrater\AppData...	524,288
<input type="checkbox"/>	6	V010001A.log	疑似TrueCrypt加密文件	AES/TwoFish/Serp...	打开保护	暂不支持	分区4_本地磁盘[D]:\Users\WMCTF\AppData\Loca...	524,288
<input type="checkbox"/>	7	V01res00001.jrs	疑似TrueCrypt加密文件	AES/TwoFish/Serp...	打开保护	暂不支持	分区4_本地磁盘[D]:\Users\Administrater\AppData...	524,288
<input type="checkbox"/>	8	V01res00002.jrs	疑似TrueCrypt加密文件	AES/TwoFish/Serp...	打开保护	暂不支持	分区4_本地磁盘[D]:\Users\Administrater\AppData...	524,288
<input type="checkbox"/>	9	USSres00001.jrs	疑似TrueCrypt加密文件	AES/TwoFish/Serp...	打开保护	暂不支持	分区4_本地磁盘[D]:\Users\Administrater\AppData...	3,145,728
<input type="checkbox"/>	10	USSres00002.jrs	疑似TrueCrypt加密文件	AES/TwoFish/Serp...	打开保护	暂不支持	分区4_本地磁盘[D]:\Users\Administrater\AppData...	3,145,728
<input type="checkbox"/>	11	edbres00002.jrs	疑似TrueCrypt加密文件	AES/TwoFish/Serp...	打开保护	暂不支持	分区4_本地磁盘[D]:\Users\Administrater\AppData...	524,288
<input type="checkbox"/>	12	edbres00001.jrs	疑似TrueCrypt加密文件	AES/TwoFish/Serp...	打开保护	暂不支持	分区4_本地磁盘[D]:\Users\Administrater\AppData...	524,288

这个很大，我忍一下，直接导出，然后解密。顺便把FTK关了



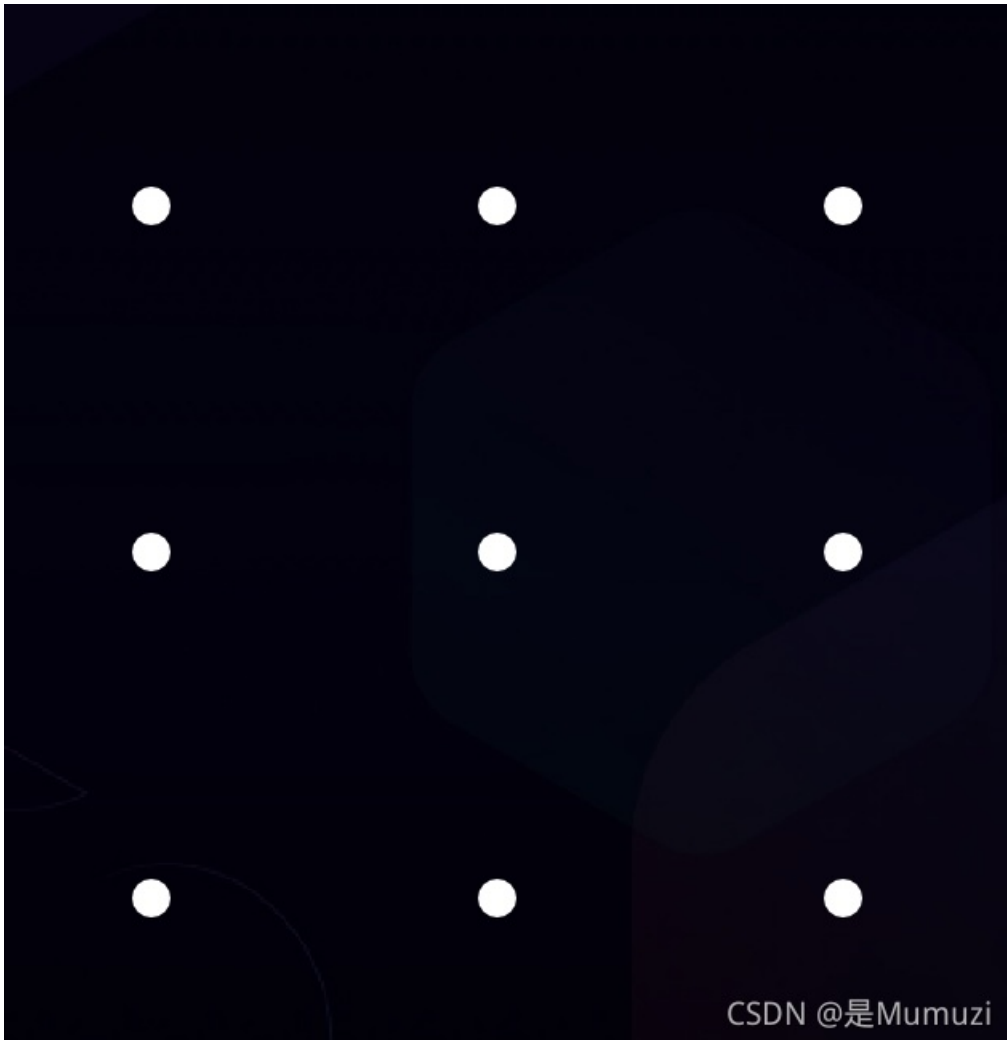
记得unmount!





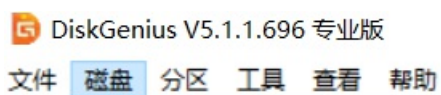
CSDN @是Mumuzi

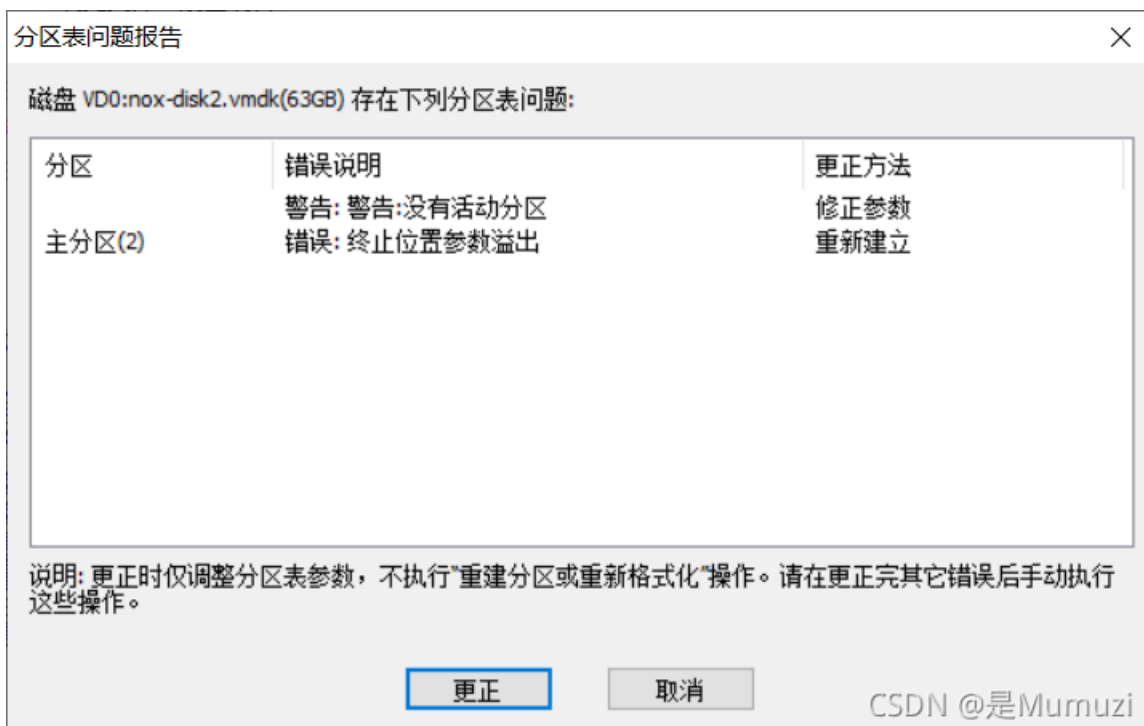
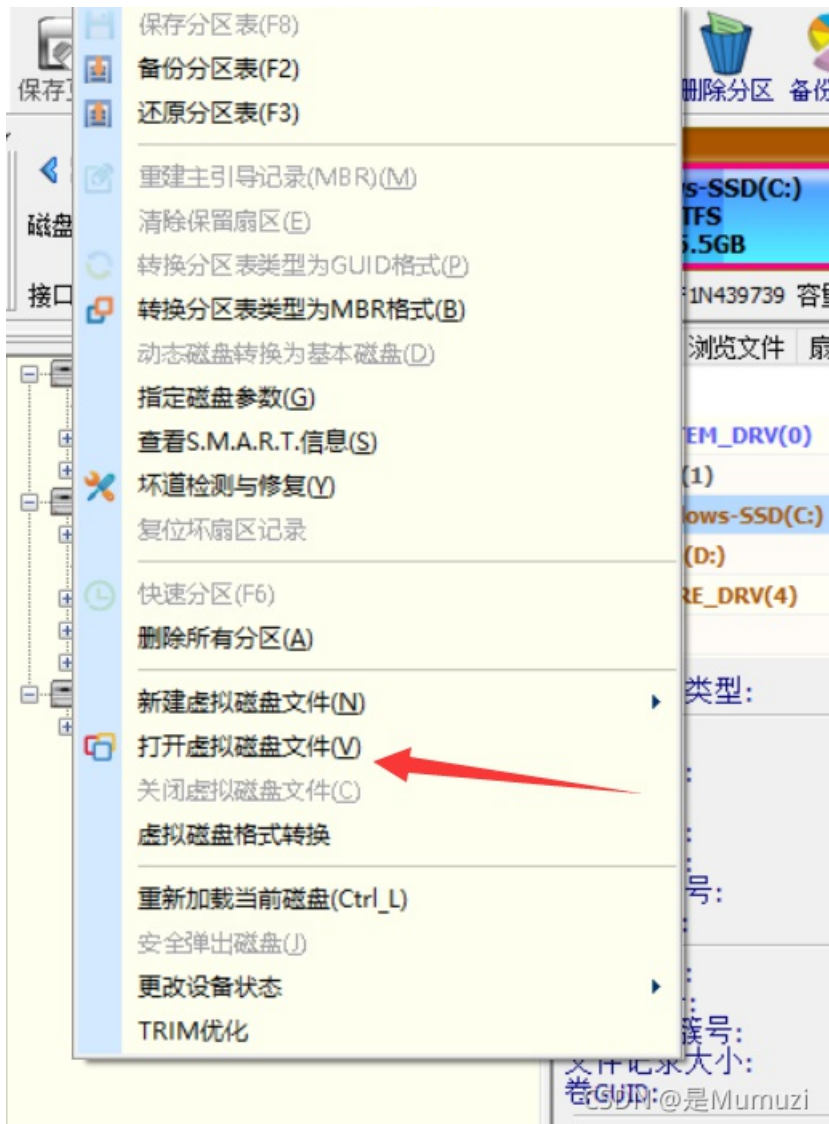
啊，是nox-disk，搜一下，是夜神模拟器，于是可以去下载一个  
 (做完了表示，直接diskgenius恢复文件然后去找到通讯录就可以了)  
 导入一开，妈的还有密码，绝了。



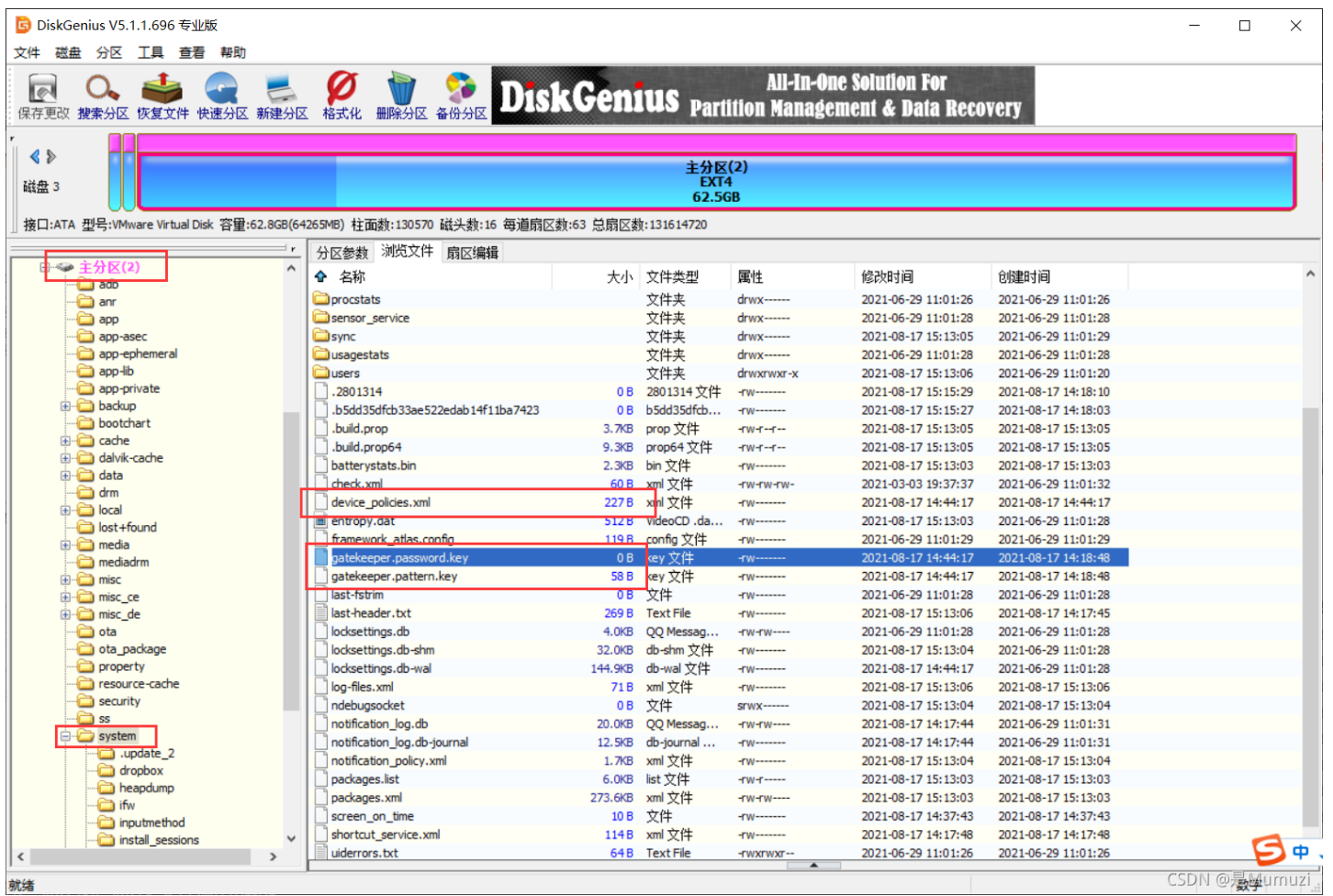
点我，一定要看

diskgenius挂载（想用winhex也行的）

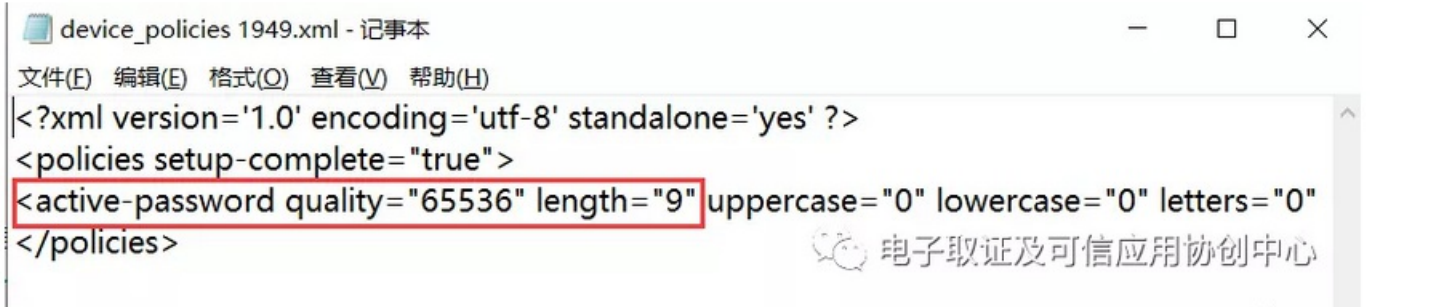




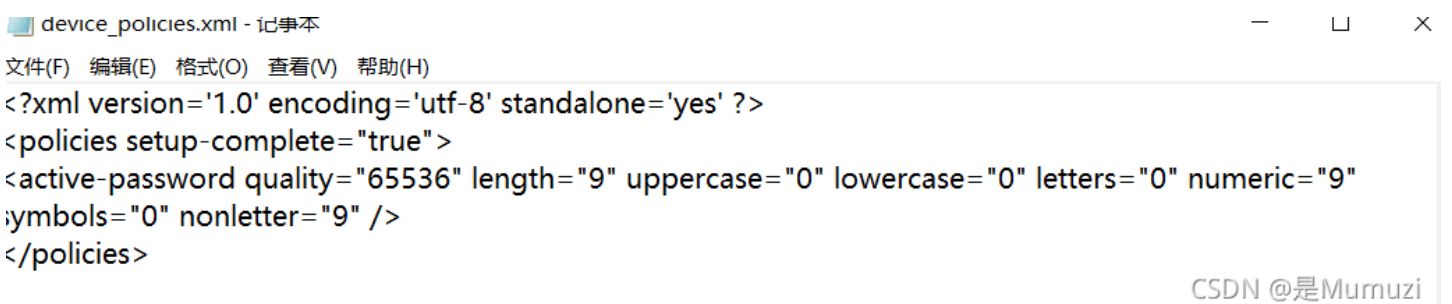
上图不用管



3.根据device\_policies.xml文件中，我们可以得到该解锁图案的长度为9.



CSDN @是Mumuzi



芽，咱也是9

然后跑他的脚本，具体看那篇博客，记得最后加一个print password

```
f1 = open('password123.txt', 'r')
lines = f1.readlines()
for data in lines:
    password = data.strip()
    to_hash = meta
    to_hash += password
    hash = scrypt.hash(to_hash, salt, N, r, p)
    print password
    print 'Equal:    %s' % (hash[0:32] == signature)

    if hash[0:32] == signature:
        print password
        print "OK"
        exit()
```

CSDN @是Mumuzi

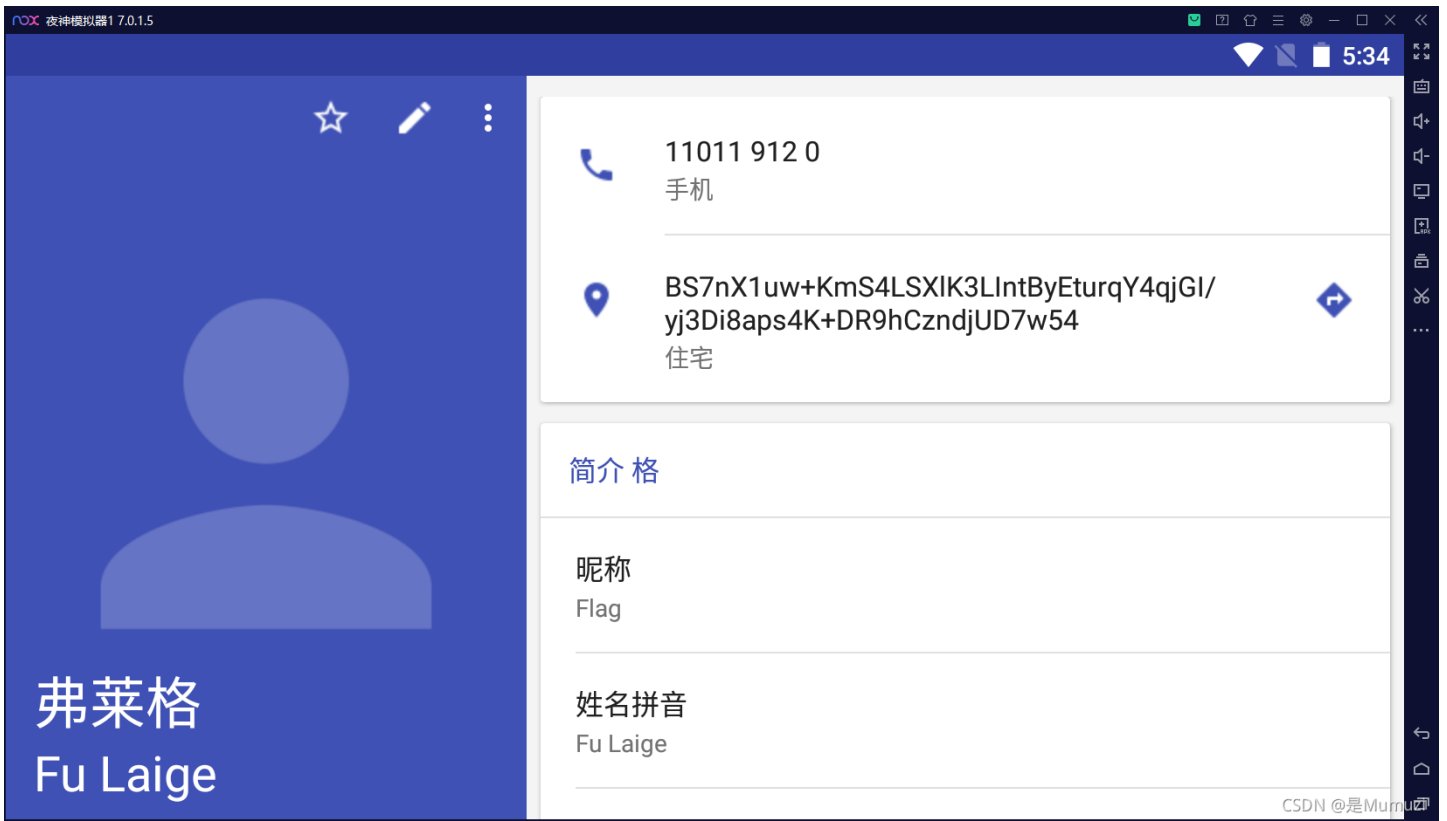
然后。。。跑了我一个多小时

不过是有其他方法的捏，但是复现玩玩捏，就懒得去搞其他方法捏，比如hashcat捏。

```
183492765
Equal:    False
183492756
Equal:    False
183492765
Equal:    True
183492765
OK
mumuzi@kali:~/桌面$
```

183492765





## 备注

锁屏密码

BS7nX1uw+KmS4LSXIK3LIntByEturqY4qjGI/yj3Di8aps4K+DR9hCzndjUD7w54

AES加密模式: ECB ▾ 填充: zeropadding ▾ 数据块: 128位 ▾ 密码: 183492765 偏移量: iv偏移量, ▾


待加密、解密的文本:  

```
BS7nX1uw+KmS4LSXlK3LIntByEturqY4qjGI/yj3Di8aps4K+DR9hCzndjUD7w54
```

↑ 将你电脑文件直接拖入试试^-^

AES加密

AES解密

AES加密、解密转换结果(base64了):   

```
wmctf{dc4fc81e0aedc4692a7e312ce503e3ef}
```

CSDN @是Mumuzi

```
wmctf{dc4fc81e0aedc4692a7e312ce503e3ef}
```