CTF题记——取证小集合

原创

m0re 于 2020-11-07 17:08:47 发布 2676 收藏 28
分类专栏： kali linux CTF 文章标签： 取证
m0re


kali linux 同时被 2 个专栏收录
3 篇文章 5 订阅
订阅专栏


CTF
31 篇文章 3 订阅
订阅专栏

前言

最近接触到的取证类题目很多，所以就总结一下这类题。
不得不提的是Volatility这个神器，本次学习，结合几个题目总结一下命令使用。还有一些大师傅的博客学来的知识。

"食用"方法

判断镜像信息，获取操作系统类型

```
volatility -f ?.img/raw/... imageinfo
```

知道操作系统类型后，用–profile指定

```
volatility -f ?.img --profile=...
```

查看当前显示的notepad文本

```
volatility  -f file.raw --profile=WinXPSP2x86 notepad
```

查看当前运行的进程

```
volatility  -f file.raw --profile=WinXPSP2x86 psscan/pslist
```

扫描所有的文件列表(常常结合grep)

```
volatility  -f file.raw --profile=WinXPSP2x86 filescan
```

根据offset提取出文件

```
volatility  -f file.raw --profile=WinXPSP2x86 dumpfiles -D . -Q 0x.....
```

扫描 Windows 的服务

```
volatility -f file.raw --profile=WinXPSP2x86 svcscan
```

查看网络连接

```
volatility -f file.raw --profile=WinXPSP2x86 connscan
```

查看命令行上的操作

```
volatility -f file.raw --profile=WinXPSP2x86 cmdscan
```

根据pid dump出相应的进程

```
volatility -f easy_dump.img --profile=Win7SP1x64 memdump -p 2580 -D 目录
```

常用命令

命令	功能
cmdline/cmdscan	列出历史cmd命令
filescan	扫描文件，可配合grep使用
netscan	扫描建立的连接和套接字，类似于netstat
pslist/psscan	列出进程列表
svcscan	扫描windows服务列表
screenshot	显示GDI样式的截屏
memdump	从内存dump进程的内存
dumpfiles	从内存dump文件

题目

湖湘杯(取证)

一个G的raw文件，工具分析就行了。
使用

```
#使用imageinfo参数查看内存是什么系统的镜像
volatility -f men.raw imageinfo   #profile=Win7SP1x86_23418
#直接查看用户名和密码hash
volatility -f men.raw --profile=Win7SP1x86_23418 hashdump
```

然后看到三个用户

```
root@kali:~/Desktop# volatility -f men.raw --profile=Win7SP1x86_23418 hashdump
Volatility Foundation Volatility Framework 2.6
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
CTF:1000:aad3b435b51404eeaad3b435b51404ee:0a640404b5c386ab12092587fe19cd02:::
root@kali:~/Desktop#
```

使用彩虹表暴力猜解

hash猜解

Enter up to 20 non-salted hashes, one per line:

0a640404b5c386ab12092587fe19cd02

进行人机身份验证 reCAPTCHA 隐私权 - 使用条款

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
0a640404b5c386ab12092587fe19cd02	NTLM	qwer1234

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

密码是qwer1234

题目是说的sha1(password)

所以再进行加密一下就可以了

qwer1234

在线加密 在线解密

sha1（qwer1234）= db25f2fc14cd2d2b1e7af307241f548fb03c312a

BUU内存取证(VN)

首先查看镜像信息

```
root@kali:~/Desktop# volatility -f mem.raw imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win7SP1x86_23418, Win7SP0x86, Win7SP1x86_24000, Win7SP1x86
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (/root/Desktop/mem.raw)
                      PAE type : PAE
                           DTB : 0x185000L
                          KDBG : 0x8176bbe8L
          Number of Processors : 2
     Image Type (Service Pack) : 0
                KPCR for CPU 0 : 0x8176cc00L
                KPCR for CPU 1 : 0x807ec000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2020-02-18 19:56:24 UTC+0000
     Image local date and time : 2020-02-19 03:56:24 +0800
root@kali:~/Desktop#
```


列出进程，一般使用`pslist`当然还有其他的，`pstree`和`psscan`

```
0x829af8c0 System                    4      0    101      479 ------      0 2020-02-18 19:52:20
 UTC+0000
0x83c4d948 smss.exe                280      4      2       30 ------      0 2020-02-18 19:52:20
 UTC+0000
0x83cc8030 csrss.exe               376    352      9      453      0      0 2020-02-18 19:52:20
 UTC+0000
0x844e4d40 wininit.exe             432    352      3       79      0      0 2020-02-18 19:52:20
 UTC+0000
0x844d9608 csrss.exe               440    424     10      311      1      0 2020-02-18 19:52:20
 UTC+0000
0x84d59d40 services.exe            488    432      7      209      0      0 2020-02-18 19:52:21
 UTC+0000
0x8466f030 lsass.exe               520    432      7      595      0      0 2020-02-18 19:52:21
 UTC+0000
0x8466f600 lsm.exe                 528    432     10      146      0      0 2020-02-18 19:52:21
 UTC+0000
0x8465e3b0 winlogon.exe            536    424      4      115      1      0 2020-02-18 19:52:21
 UTC+0000
0x846af568 svchost.exe             668    488     10      356      0      0 2020-02-18 19:52:21
 UTC+0000
0x846c0728 svchost.exe             740    488      8      285      0      0 2020-02-18 19:52:21
 UTC+0000
0x8861ead8 svchost.exe             804    488     21      467      0      0 2020-02-18 19:52:21
 UTC+0000
0x846e6300 svchost.exe             860    488     23      487      0      0 2020-02-18 19:52:21
 UTC+0000
0x846edb38 svchost.exe             884    488     39      988      0      0 2020-02-18 19:52:21
 UTC+0000
0x847315c0 audiodg.exe             988    804      7      132      0      0 2020-02-18 19:52:21
```


记几个重要的地方，一般是`notepad.exe、TrueCrypt.exe 、mspaint.exe、iexplore.exe 、DumpIt.exe`

简单介绍：

`mspaint.exe`是一个画图软件

`notepad.exe`是记事本，一般记事本中会有内容hint或者在内存中(还未保存)

`DumpIt`是一款绿色免安装的 windows 内存镜像取证工具。利用它我们可以轻松地将一个系统的完整内存镜像下来，并用于后续的调查取证工作。

`TrueCrypt.exe`

TrueCrypt.exe 的信息

进程 TrueCrypt 是附属于软件 TrueCrypt 由 TrueCrypt Foundation (www.truecrypt.org) 发行。

注释: Windows不需要TrueCrypt.exe。 文件 TrueCrypt.exe 是存放在 "C:\Program Files" 下的子目录。 已知的 Windows 10/8/7/XP 文件大小为 1,516,496 字节 (占总出现比率 53%), 1,106,112 字节 及 6 种其它情况。
这已由可信任的公司发出证书。 这个不是 Windows 系统文件。 进程是不可见的。 这个进程在 Windows 载入程序中开启 (参看注册表项: Run)。 它可以改变其他程序的行为和操控其他程序。
TrueCrypt.exe 是有能力可以 纪录输入。 总结在技术上威胁的危险度是 46% 。
如果您遇到了TrueCrypt.exe出问题, 您可以询问开发者, www.truecrypt.org,, 或者在控制面板中的程序部分卸载它.

推荐: 识别和TrueCrypt.exe相关的问题

如果 TrueCrypt.exe 位于在 of C:\ 下的子目录下, 那么威胁的危险度是 28% 。文件大小是 1,516,496 字节 (占总出现比率 50%) 或 1,517,520 字节。 这文件是有数字证书。 这个不是 Windows 系统文件。 TrueCrypt.exe 是有能力可以 纪录输入。

切记: 有些病毒软件伪装成 TrueCrypt.exe, 尤其是处于这些目录下: c:\windows 或 c:\windows\system32 . 因此,您需要检查计算机中的TrueCrypt.exe进程,确保它不是病毒. 我们推荐使用 Security Task Manager来确保您的计算机安全.

先将这些可疑进程dump下来。进行进一步的分析

```
volatility -f mem.raw --profile=Win7SP0x86 memdump -p 2648 --dump-dir=./
```

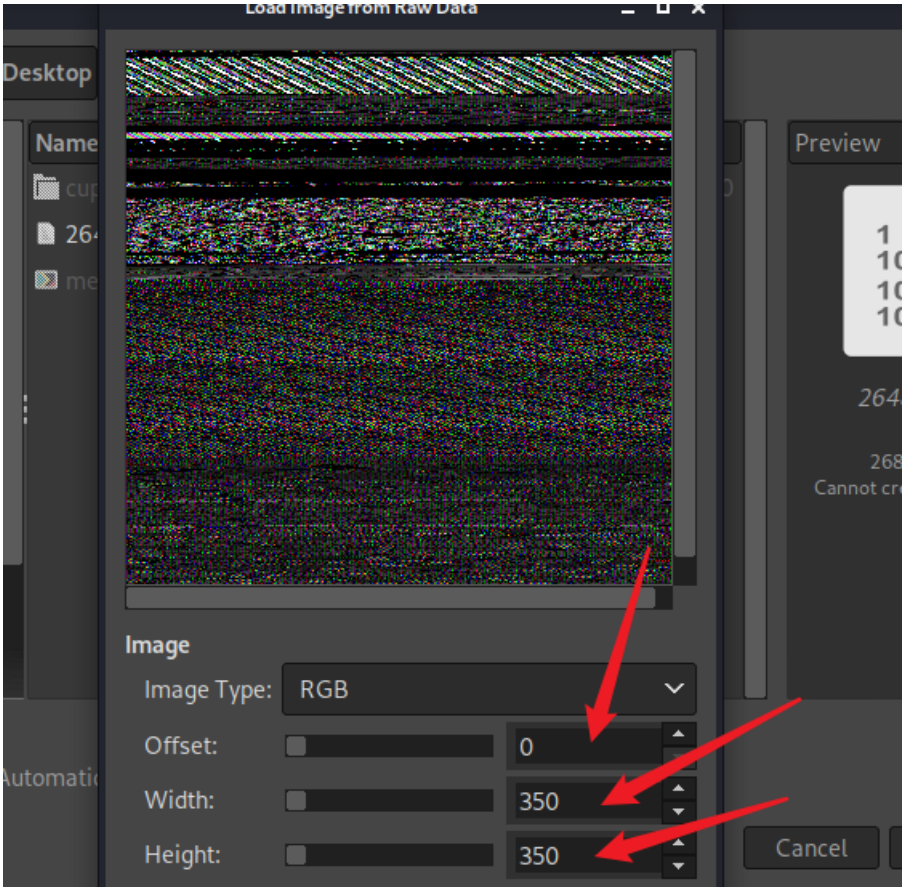
类似这样的，先分析画图的，需要使用一个工具`gimp`在linux中相当于photoshop一样的软件，直接安装就可以

```
#在root权限下
apt-get update
apt-get install gimp
```

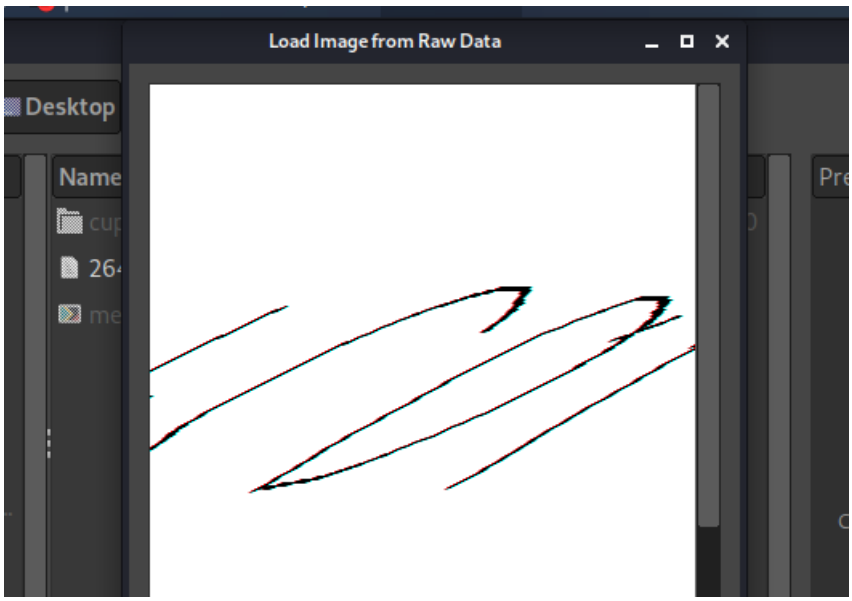
然后打开使用就可以了

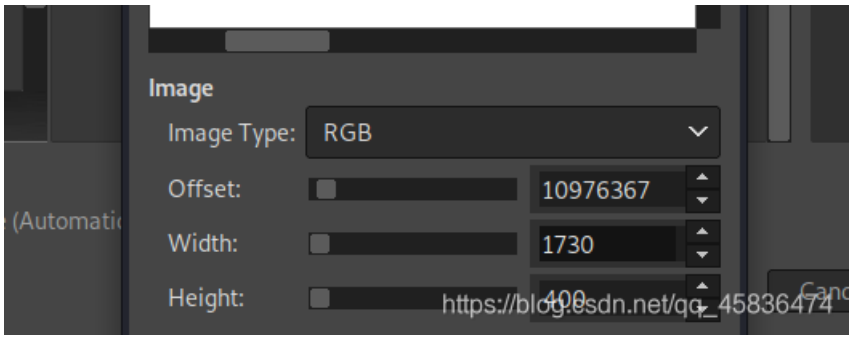
https://segmentfault.com/a/1190000018813033?utm_source=tag-newest

这个师傅讲的如何使用这个软件。

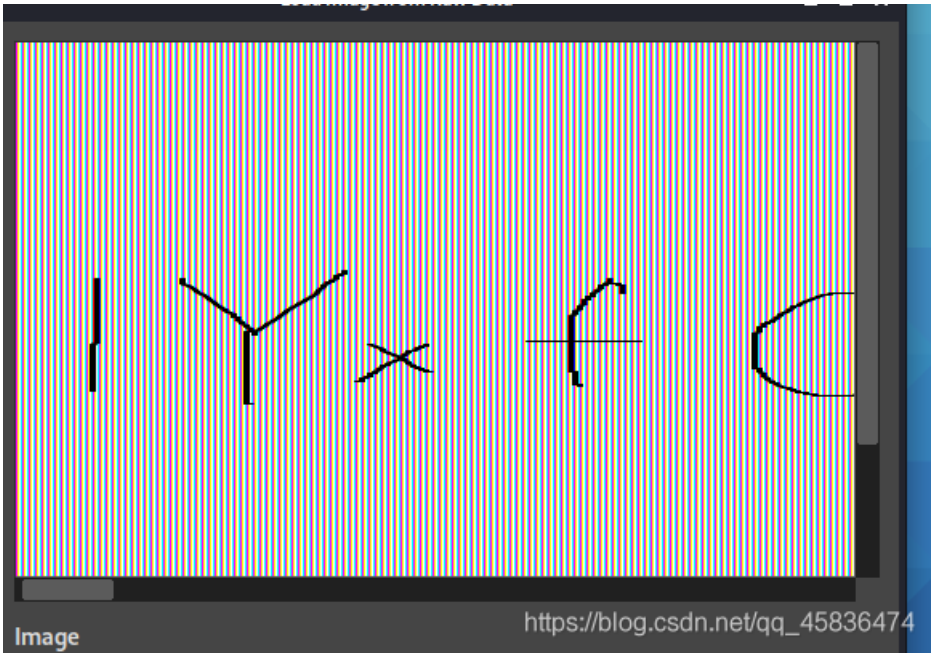


刚打开时，是默认分辨率是0，高度和宽度都是350，先随便调节一下，大概就是三个变量值都先调低一点，然后慢慢调高。调试很多次，慢慢调节出一些值，这个值可以参考windows系统自带画图软件的，我的是1628x440，修改一下，就开始改变分辨率就可以了。下图大概成型了





然后微调一下宽度就行,不过调过之后是反的，所以可以再调大一些，多调动尝试最佳角度。



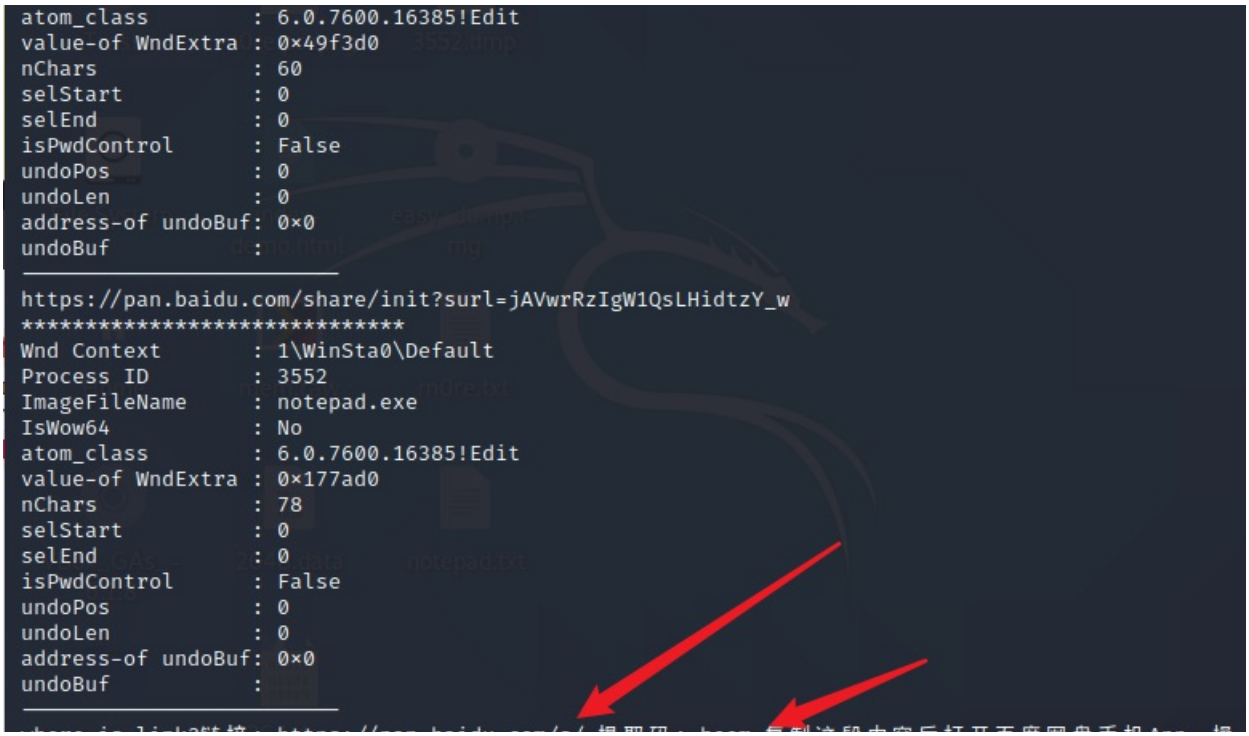
拿到了一个字符串`1YxfCQ6goYBD6Q`

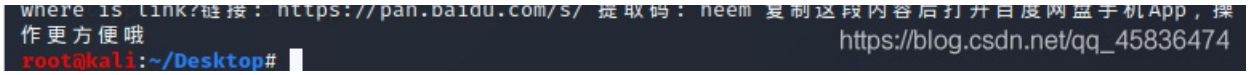
然后看下一步，提取记事本中的内容

这里介绍一个插件`editbox`可以显示有关编辑控件的信息。

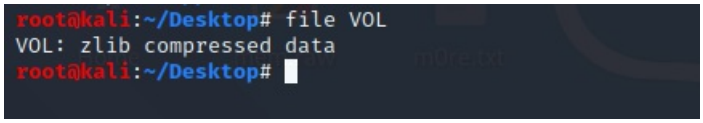
使用命令

```
volatility -f men.raw --profile=Win7SP1x86_23418 editbox
```

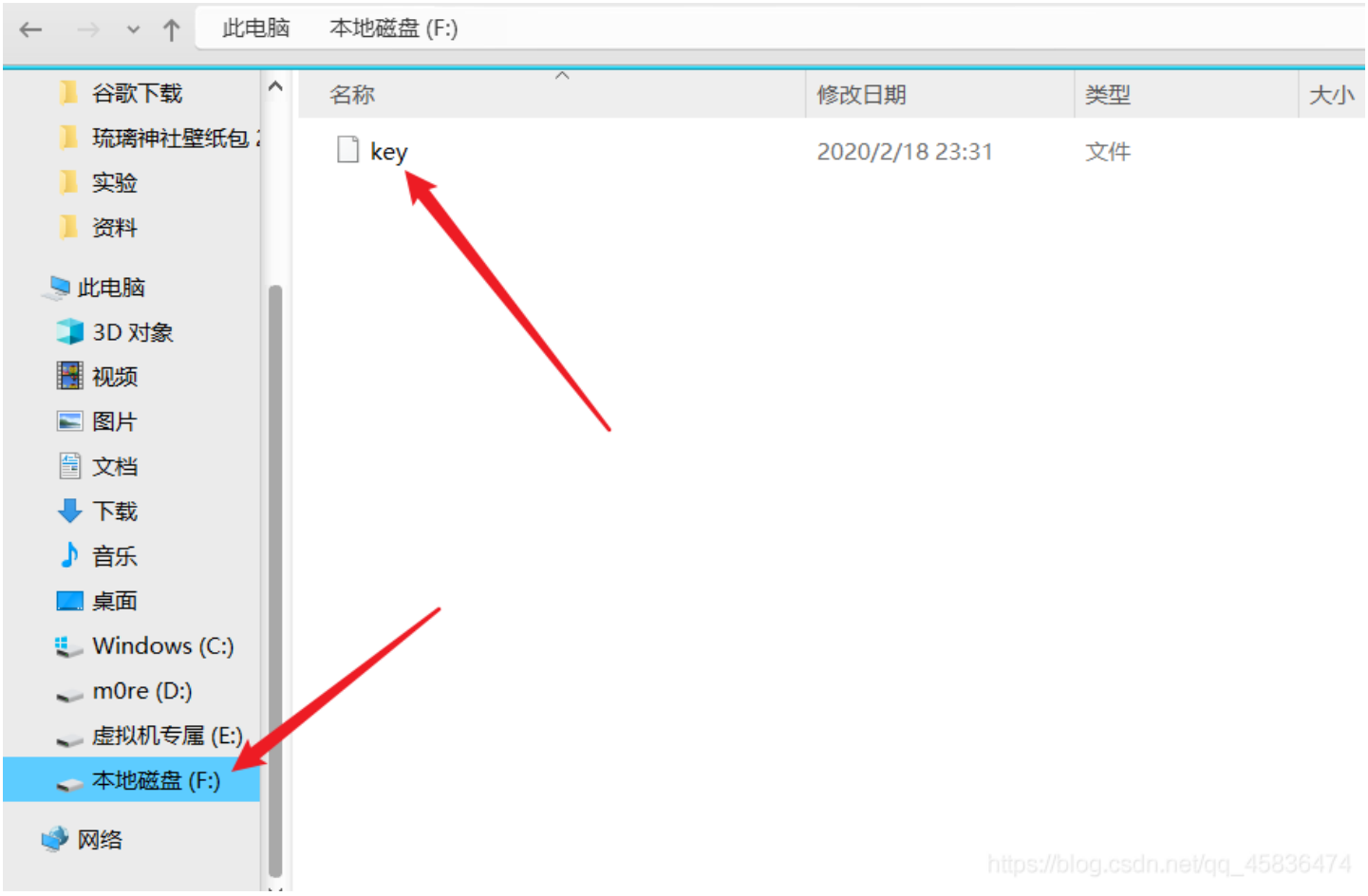




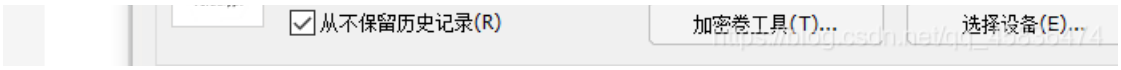
网盘链接得到了，提取码也得到了，就进行下一步
下载得到一个VOL文件



然后就是TrueCrypt，这个dump下来不用进行其他操作，成功挂载到F盘，里面看到key 文件



打开获得`uOjFdKu1jsbWI8N51jsbWI8N5`
这个就是VOL挂载加密的key，然后使用TrueCrypt对VOL进行正常解密，
下载安装TrueCrypt，我安装到物理机上面了，直接使用。
密钥填写一下，选择TrueCrypto模式



成功挂载

挂载后，可以在Z盘直接看到一个压缩包

打开需要密码，这个时候用到了画图时得到的密码。

文件(F)　编辑(E)　查看(V)　帮助(H)

RoarCTF{wm_D0uB1e_TC-cRypt}

easydump

来源：护网杯2018-MISC-easydump

`.img`文件，也是一种内存镜像

使用取证神器直接跑

```
root@kali:~/Desktop# volatility -f easy_dump.img imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64, Win2008R2SP1x64_24000
, Win2008R2SP1x64_23418, Win2008R2SP1x64, Win7SP1x64_24000, Win7SP1x64_23418
                     AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/root/Desktop/easy_dump.img)
                      PAE type : No PAE
                           DTB : 0x187000L
                          KDBG : 0xf80003ff4070L
          Number of Processors : 1
     Image Type (Service Pack) : 0
                KPCR for CPU 0 : 0xfffff80003ff5d00L
             KUSER_SHARED_DATA : 0xfffff78000000000L
           Image date and time : 2018-10-01 14:30:52 UTC+0000
     Image local date and time : 2018-10-01 22:30:52 +0800
root@kali:~/Desktop#
```


然后就是列出进程

pslist

最显眼的还是这个notepad.exe了

```
0xfffffa80088f35c0 TPAutoConnect.          2548   2004      4      110      1
14:27:17 UTC+0000
0xfffffa800a388060 conhost.exe             2556    412      1       32      1
14:27:17 UTC+0000
0xfffffa8009256b30 notepad.exe             2580   1264      2       57      1
14:27:19 UTC+0000
0xfffffa8009bf8b30 WmiPrvSE.exe            2780    628     12      296      0
14:27:27 UTC+0000
0xfffffa8009005820 svchost.exe              608    504      8      114      0
14:29:07 UTC+0000
0xfffffa8009d19b30 sppsvc.exe              1356    504      5      150      0
14:29:07 UTC+0000
0xfffffa8008b088c0 svchost.exe             2472    504     13      369      0
14:29:07 UTC+0000
0xfffffa8008a7a290 WmiApSrv.exe            2280    504      6      118      0
14:29:30 UTC+0000
0xfffffa800a2d6b30 SearchProtocol          2576   2312      7      238      0
14:30:11 UTC+0000
0xfffffa8009ee7660 SearchFilterHo          1580   2312      5       86      0
14:30:11 UTC+0000
0xfffffa800a3864e0 dllhost.exe             1028    628      6      114      1
14:30:47 UTC+0000
0xfffffa800902cb30 dllhost.exe             1056    628     10      199      1
14:30:49 UTC+0000
0xfffffa8007ef92f0 DumpIt.exe              1724   1264      2       43      1
14:30:51 UTC+0000
0xfffffa8007ee4230 conhost.exe              868    412      2       59      1
14:30:51 UTC+0000
```


然后dump下来2580.dmp,binwalk查看，信息太多，直接foremost分离文件，里面发现有用的就是两个压缩包，解压是img文件。但是这个文件同样使用volatility去跑，确没有信息，所以判断这个不是内存镜像文件。

```
root@kali:~/Desktop# volatility -f m
m0re.pcapng  message.img
root@kali:~/Desktop# volatility -f message.img imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : No suggestion (Instantiated with no profile)
                     AS Layer1 : FileAddressSpace (/root/Desktop/message.img)
                      PAE type : No PAE
root@kali:~/Desktop#
```


先strings看一下

```
root@kali:~/Desktop# strings message.img > message.txt
root@kali:~/Desktop# ls
2580.dmp  easy_dump.img  index-demo.html  m0re.pcapng  message.img  message.txt  output
root@kali:~/Desktop# more message.txt
/root/ctf/message
lost+found
hint.txt
.message.swp
.Trash-0
10 10
10 11
10 12
10 13
10 14
10 15
10 16
10 17
10 18
10 19
10 20
10 21
```


发现好多信息，然后将img文件挂载在linux系统中，

```
mount -o loop message.img /root/Desktop/m0re
```

挂载后，可以切换到该目录进行查看信息

```
cd m0re/
ls -all  #查看所有文件，一般隐藏信息较多
```

然后在`.Trash-0/file`下看到一个`.message.swp`
转存一下

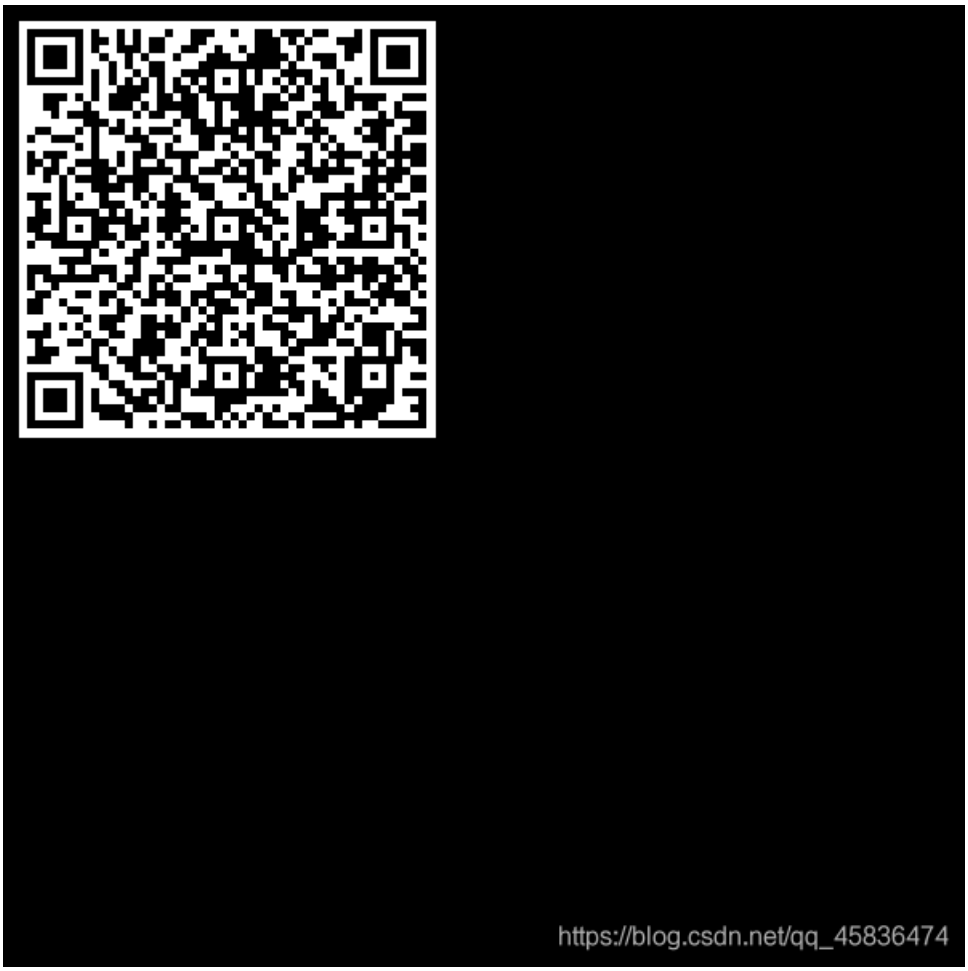
```
cat .message.swp > m0re.txt
strings m0re.txt
```

```
root@kali:~/Desktop/m0re/.Trash-0/files# cat .message.swp > m0re.txt
root@kali:~/Desktop/m0re/.Trash-0/files# strings m0re.txt
b0VIM 8.0
n3k0
shiki-2.local
~n3k0/work/HuWangBei/5/message
U3210
#"!
yispywex!xfml_fmbo_ufd_wcwpw
root@kali:~/Desktop/m0re/.Trash-0/files#
```

可能是密码什么的，保存一下
hint.txt文件里面都是坐标，猜测是要画图
之前保存过画图的python脚本

```
from PIL import Image

with open('hint.txt','r') as f:
 points = f.readlines()

pic=Image.new('RGB',(600,600),'black')
pix=pic.load()

for i in points:
 i=i.strip().split(' ')
 pix[int(i[0]),int(i[1])]=(255,255,255)

pic.save('out.png','png')
```

得到二维码

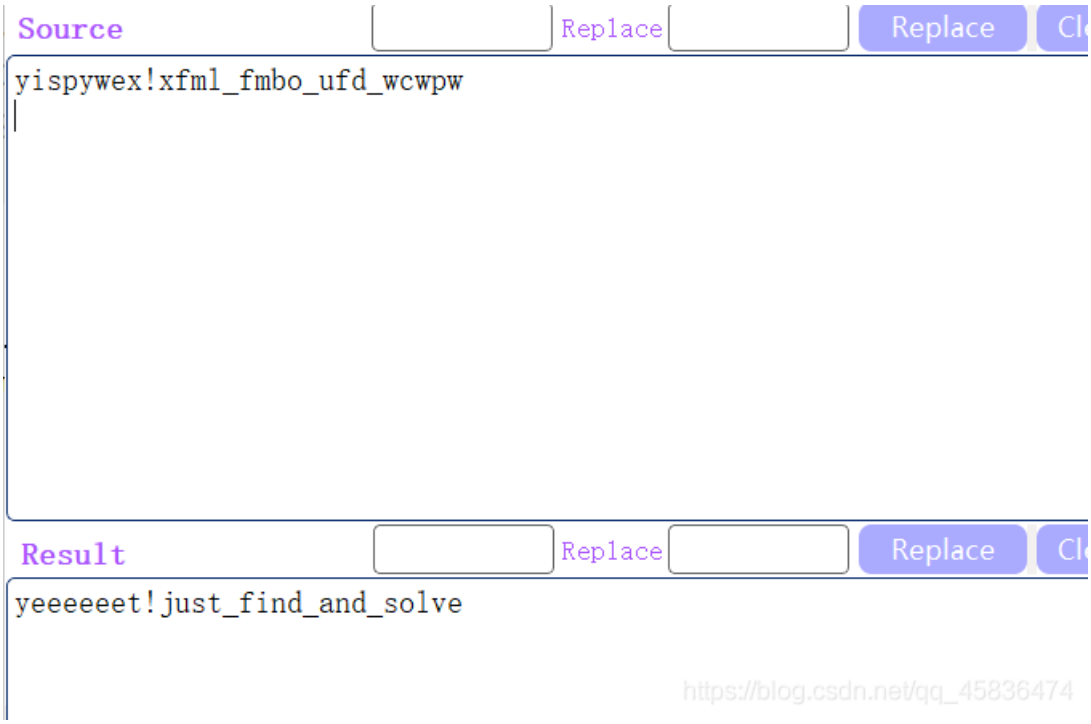


识别后得到

```
Here is the vigenere key: aeolus, but i deleted the encrypted message。
```

维吉尼亚密码，密钥是`aeolus`

密文不知道，前面的可疑字符串有可能是密文



得到结果。这个题就有点杂了。

总结

学到这里，内存取证的基础题已经可以应付了，这是第二次学习取证的知识，感觉很有意思。

参考博客

https://blog.xiafeng2333.top/ctf-25/
https://blog.xiafeng2333.top/ctf-11/
https://segmentfault.com/a/1190000018813033?utm_source=tag-newest