

ctf内存取证----easy_dump

原创

[MOLLMY](#) 于 2019-09-15 22:04:07 发布 2333 收藏 11

分类专栏: [网络攻防 CTF](#)

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/MOLLMY/article/details/100865618>

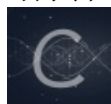
版权



[网络攻防](#) 同时被 2 个专栏收录

4 篇文章 0 订阅

订阅专栏



[CTF](#)

5 篇文章 0 订阅

订阅专栏

前一段时间又做了去年护网杯的内存取证题

第一次不参考任何wp提示, 自己做, 没想到做得挺顺利

Easy_dump writeup

解题步骤

Step 1

Volatility imageinfo

得知系统为Win7SP1x64

Setp 2

查看进程列表

```

0xfffffa800a1f8a10 TPAutoConnSvc. 2064 504 10 130 0 0 2018-10-01 14:27:07 UTC+0000
0xfffffa800a249b30 WmiPrvSE.exe 1516 628 10 202 0 0 2018-10-01 14:27:08 UTC+0000
0xfffffa800a1c7740 dllhost.exe 1700 504 17 200 0 0 2018-10-01 14:27:08 UTC+0000
0xfffffa800a2a67c0 msdtc.exe 1400 504 15 154 0 0 2018-10-01 14:27:09 UTC+0000
0xfffffa8009f13af0 SearchIndexer. 2312 504 13 605 0 0 2018-10-01 14:27:12 UTC+0000
0xfffffa8008f12b30 SearchProtocol 2380 2312 6 255 1 0 2018-10-01 14:27:12 UTC+0000
0xfffffa80088f35c0 TPAutoConnect. 2548 2004 4 110 1 0 2018-10-01 14:27:17 UTC+0000
0xfffffa800a388060 conhost.exe 2556 412 1 32 1 0 2018-10-01 14:27:17 UTC+0000
0xfffffa8009256b30 notepad.exe 2580 1264 2 57 1 0 2018-10-01 14:27:19 UTC+0000
0xfffffa8009bfb30 WmiPrvSE.exe 2780 628 12 296 0 0 2018-10-01 14:27:27 UTC+0000
0xfffffa8009005820 svchost.exe 608 504 8 114 0 0 2018-10-01 14:29:07 UTC+0000
0xfffffa8009d19b30 sppsvc.exe 1356 504 5 150 0 0 2018-10-01 14:29:07 UTC+0000
0xfffffa8008b088c0 svchost.exe 2472 504 13 369 0 0 2018-10-01 14:29:07 UTC+0000
0xfffffa8008a7a290 WmiApSrv.exe 2280 504 6 118 0 0 2018-10-01 14:29:30 UTC+0000
0xfffffa800a2d6b30 SearchProtocol 2576 2312 7 238 0 0 2018-10-01 14:30:11 UTC+0000
0xfffffa8009ee7660 SearchFilterHo 1580 2312 5 86 0 0 2018-10-01 14:30:11 UTC+0000
0xfffffa800a3864e0 dllhost.exe 1028 628 6 114 1 0 2018-10-01 14:30:47 UTC+0000
0xfffffa800902cb30 dllhost.exe 1056 628 10 199 1 0 2018-10-01 14:30:49 UTC+0000
0xfffffa8007ef92f0 DumpIt.exe 1724 1264 2 43 1 0 2018-10-01 14:30:51 UTC+0000
0xfffffa8007ee4230 conhost.exe 868 412 2 59 1 0 2018-10-01 14:30:51 UTC+0000

```

查看命令行历史

```

root@kali:~/Documents/easy_dump# volatility -f easy_dump.img --profile=Win7SP1x64 cmdscan
Volatility Foundation Volatility Framework 2.6
*****
CommandProcess: conhost.exe Pid: 2556
CommandHistory: 0xcd220 Application: TPAutoConnect.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x58
*****
CommandProcess: conhost.exe Pid: 868
CommandHistory: 0x44d9f0 Application: DumpIt.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x58

```

没有什么发现

Step 3

执行netscan、iehistory等命令，发现在iehistory中有一个jpg文件

```

0x00000030 00 00 00 bf 00 00 00 bf ff ff ff 00 00 00 00 .....
0x00000040 00 00 00 00 00 00 00 00 .....
root@kali:~/Documents/easy_dump# volatility -f easy_dump.img --profile=Win7SP1x64 iehistory
Volatility Foundation Volatility Framework 2.6
*****
Process: 1264 explorer.exe
Cache type "URL" at 0x2785000
Record length: 0x100
Location: :2018100120181002: n3k0@file:///C:/phos.jpg
Last modified: 2018-10-01 22:30:49 UTC+0000
Last accessed: 2018-10-01 14:30:49 UTC+0000
File Offset: 0x100, Data Offset: 0x0, Data Length: 0x0
*****
Process: 1264 explorer.exe
Cache type "URL" at 0x2785100
Record length: 0x100
Location: :2018100120181002: n3k0@Host: ??????????
Last modified: 2018-10-01 18:36:58 UTC+0000
Last accessed: 2018-10-01 10:36:58 UTC+0000
File Offset: 0x100, Data Offset: 0x0, Data Length: 0x0
*****
Process: 1264 explorer.exe
Cache type "URL" at 0x2785200
Record length: 0x100
Location: :2018100120181002: n3k0@file:///Users/phos.jpg
Last modified: 2018-10-01 21:57:40 UTC+0000
Last accessed: 2018-10-01 13:57:40 UTC+0000
File Offset: 0x100, Data Offset: 0x0, Data Length: 0x0
*****
Process: 1264 explorer.exe
Cache type "URL" at 0x2785300
Record length: 0x100
Location: :2018100120181002: n3k0@file:///C:/phos.jpg
Last modified: 2018-10-01 22:30:49 UTC+0000
Last accessed: 2018-10-01 14:27:30 UTC+0000
File Offset: 0x100, Data Offset: 0x0, Data Length: 0x0
*****

```

Filescan | grep "jpg" 找到这个jpg文件，

dumpfiles -Q physic_address -D directory_in_which_to_dump导出

```
*****
Process: 1264 explorer.exe
Cache type "URL" at 0x7c55200
Record length: 0x100
Location: Visited: n3k0@file:///C:/phos.jpg
Last modified: 2018-10-01 14:27:30 UTC+0000
Last accessed: 2018-10-01 14:27:30 UTC+0000
File Offset: 0x100, Data Offset: 0x0, Data Length: 0x8c
root@kali:~/Documents/easy_dump# volatility -f easy_dump.img --profile=Win7SP1x64 filescan | grep "jpg"
Volatility Foundation Volatility Framework 2.6
0x0000000023284f20 32 0 RW---- \Device\HarddiskVolume1\phos.jpg

root@kali:~/Documents/easy_dump# volatility -f easy_dump.img --profile=Win7SP1x64 dumpfiles -Q 0x0000000023284f20
r./
Volatility Foundation Volatility Framework 2.6
DataSectionObject 0x23284f20 None \Device\HarddiskVolume1\phos.jpg
SharedCacheMap 0x23284f20 None \Device\HarddiskVolume1\phos.jpg
```

Binwalk phos.jpg发现图片后面附了一个zip文件，

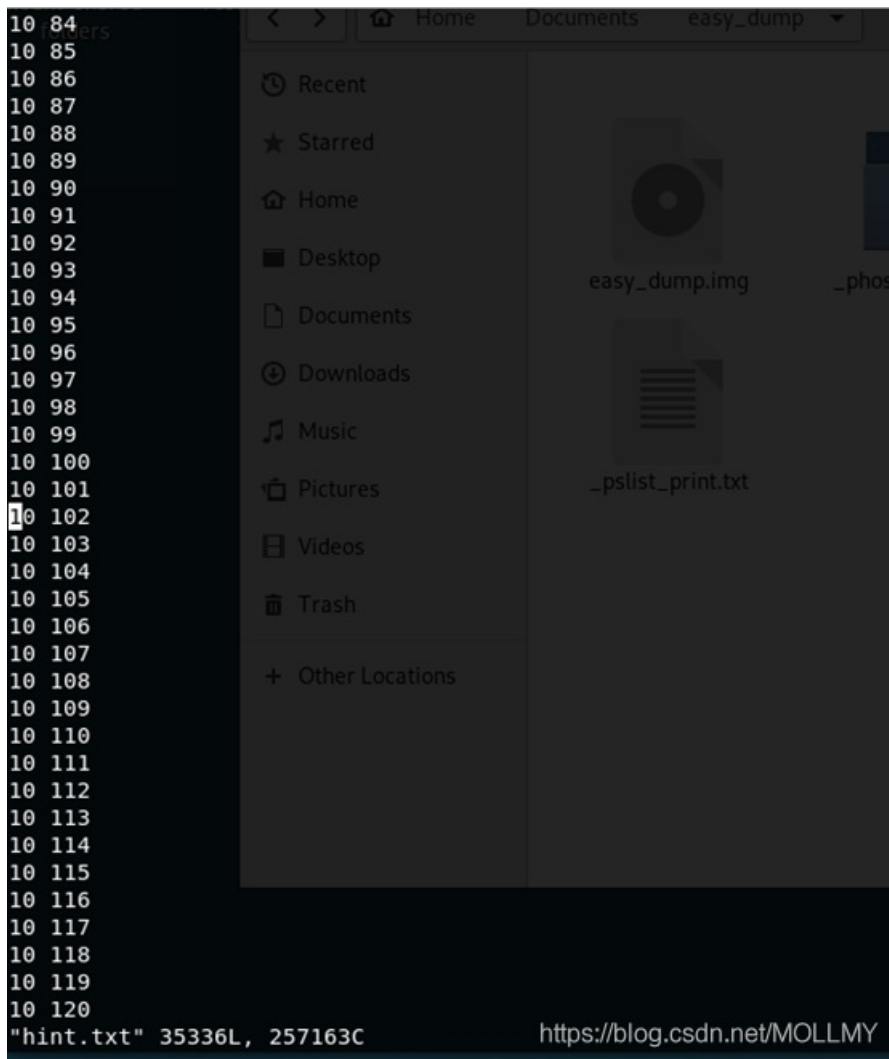
```
root@kali:~/Documents/easy_dump# binwalk phos.jpg
DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x0          JPEG image data, EXIF standard
12          0xc          TIFF image data, little-endian offset of first image directory: 8
2238922     0x2229CA    Zip archive data, at least v2.0 to extract, compressed size: 87799, uncompressed size: 1048576,
name: message.img
2326871     0x238157    End of Zip archive, footer length: 22

root@kali:~/Documents/easy_dump# binwalk -e phos.jpg
DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x0          JPEG image data, EXIF standard
12          0xc          TIFF image data, little-endian offset of first image directory: 8
2238922     0x2229CA    Zip archive data, at least v2.0 to extract, compressed size: 87799, uncompressed size: 1048576,
name: message.img
2326871     0x238157    End of Zip archive, footer length: 22
```

binwalk -e phos.jpg提取文件，解压后得到一个message.img文件，binwalk -e message.img提取到一个hint.txt

Step 4

查看hint.txt



疑似坐标点

将这些数字作为坐标画出来

```
import matplotlib.pyplot as plt
import numpy as np
...
@author:sunmx
根据二位坐标点绘图
...
x = []
y = []
with open('hint.txt','r') as f:
    datas = f.readlines()
    for data in datas:
        arr = data.split(' ')
        x.append(int(arr[0]))
        y.append(int(arr[1]))

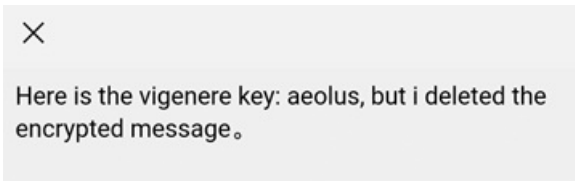
plt.plot(x,y,'ks',ms=1)
plt.show()
```



```
hint.txt  draw.py x
draw.py > ...
1 import matplotlib.pyplot as plt
2 import numpy as np
3 x = []
4 y = []
5 with open('hint.txt','r') as f:
6     datas = f.readlines()
7     for data in datas:
8         arr = data.split(' ')
9         x.append(int(arr[0]))
10        y.append(int(arr[1]))
11
12 plt.plot(x,y,'ks',ms=1)
13 plt.show()
14
15
```

<https://blog.csdn.net/MOLLMY>

二维码内容为“Here is the vigenere key: aeolus, but I deleted the encrypted message.”

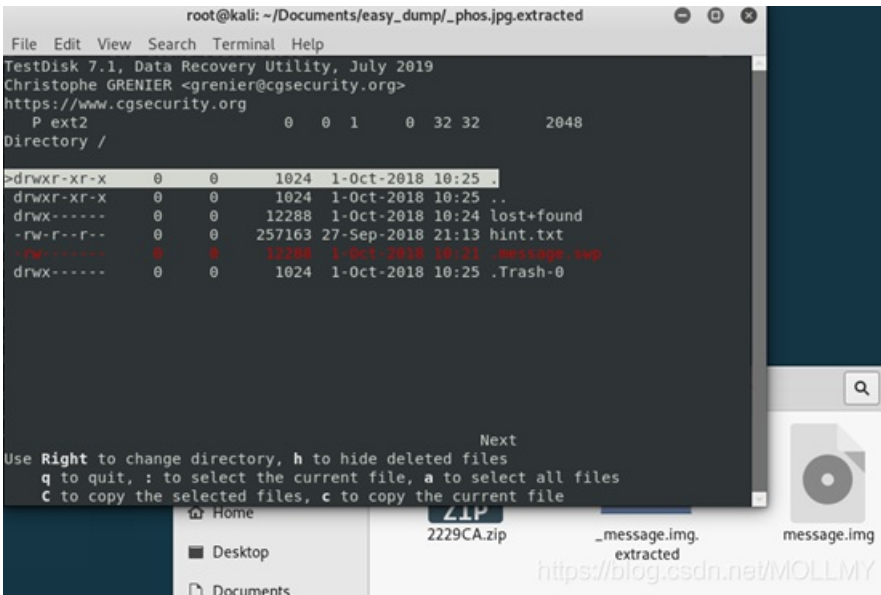


Step 5

恢复删掉的信息

Testdisk message.img[孙1]

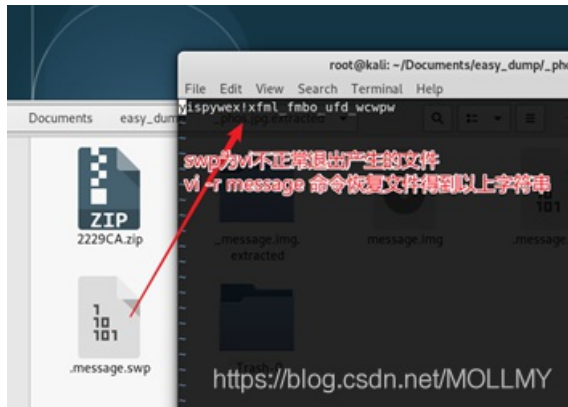
分析，然后list



<https://blog.csdn.net/MOLLMY>

恢复到当前文件夹，ls -a查看所有的文件包括隐藏文件

```
root@kali:~/Documents/easy_dump/_phos.jpg_extracted# ls -la
. 2229CA.zip message.img .message.swp
.. message _message.img.extracted .Trash-0
root@kali:~/Documents/easy_dump/_phos.jpg_extracted#
```



恢复这个message文件得到字符串yispywex!xfml_fmbo_ufd_wcwpw

这应该是vigenere加密的密文

Step 6

Vigenere 解密

Key: aeolus

密文: yispywex!xfml_fmbo_ufd_wcwpw

```

...
@author:sunmx
维吉尼亚密码解密
...

key = 'aeolus'
cip = 'yispywex!xfml_fmbo_ufd_wcwpw'
ascii = 'abcdefghijklmnopqrstuvwxyz'
# 忽略符号，符号仍位于原来的位置
syb = dict()
cip_l = list(cip)
for i in range(len(cip)):
    if cip[i] not in ascii:
        syb[i] = cip[i]
        cip_l.pop(cip_l.index(syb[i]))
print(syb)

cipher = ''.join(cip_l)
k_len = len(key)
c_len = len(cipher)
plaintext = ''
i = 0
while i < c_len:
    j = i % k_len
    k = ascii.index(key[j])
    m = ascii.index(cipher[i])
    if m < k:
        m += 26
    plaintext += ascii[m-k]
    i += 1

syb_i = syb.keys()
for i in syb_i:
    plaintext = plaintext[:i]+syb[i]+plaintext[i:]

print(plaintext)

```

或者在线解密[\[孙2\]](#)

维吉尼亚密码加密解密

yispywex!xfml_fmbo_ufd_wcwpw

密钥

yeeeeet!just_find_and_solve

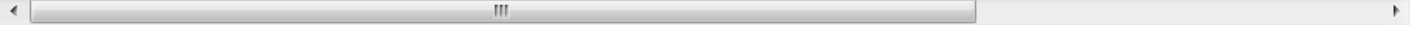
<https://blog.csdn.net/MOLLMY>

yeeeeet!just_find_and_solve

应该是flag

[孙1]参考https://www.cgsecurity.org/wiki/TestDisk:_undelete_file_for_ext2

https://www.cgsecurity.org/wiki/Testdisk_%E6%93%8D%E4%BD%9C%E6%8C%87%E5%8D%97#.E8.BF.90.



[孙2]<https://www.qqxiuzi.cn/bianma/weijiniyamima.php>