ctf 内存取证的一些命令

原创

舞动的獾 于 2019-11-12 15:50:40 发布 5024 收藏 34

分类专栏： ctf 文章标签： ctf内存取证相关命令

本文链接：https://blog.csdn.net/Yu_csdnstory/article/details/103030570

版权

ctf 专栏收录该内容

3 篇文章 0 订阅

订阅专栏

内存取证

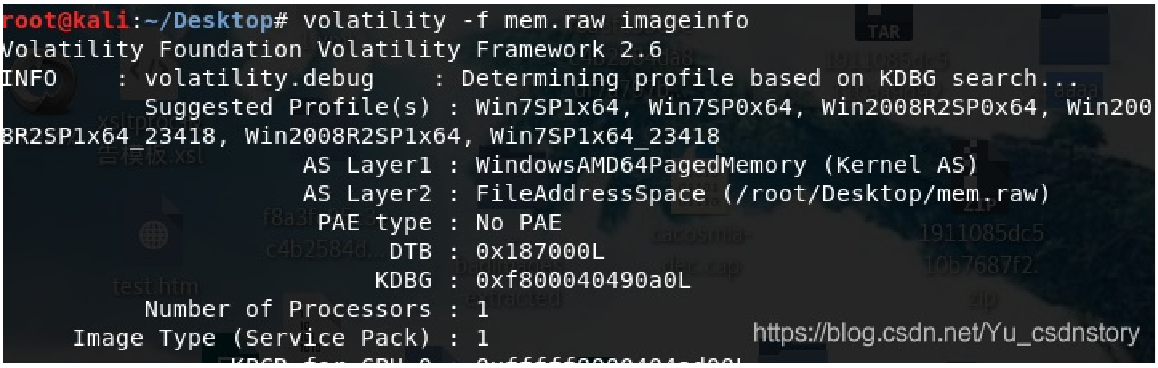
相当于给出镜像文件，一般为.raw文件，在这些文件中找出相应的文件

常用命令

在kali下用工具volatility，以湖湘杯的文件men.raw为例

查看系统信息

`volatility -f mem.raw imageinfo`

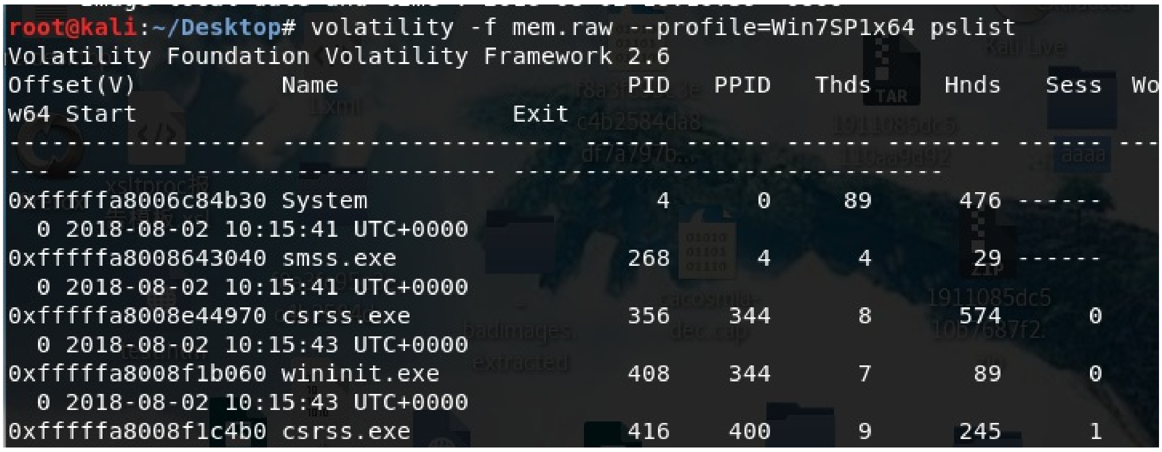


```
root@kali:~/Desktop# volatility -f mem.raw imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64, Win2008R2SP1x64_23418, Win2008R2SP1x64, Win7SP1x64_23418
                     AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/root/Desktop/mem.raw)
                      PAE type : No PAE
                           DTB : 0x187000L
                          KDBG : 0xf800040490a0L
          Number of Processors : 1
     Image Type (Service Pack) : 1
```

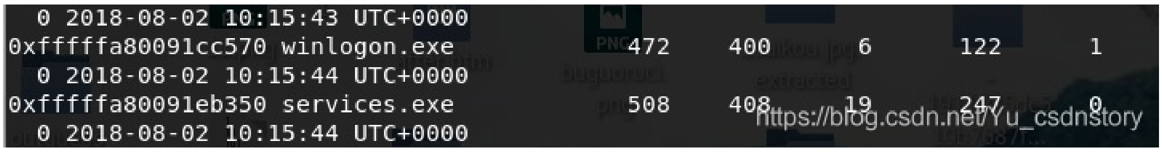
如下显示的系统都有可能，可以一个个的试试

查看运行程序列表

`volatility -f mem.raw --profile=Win7SP1x64 pslist`



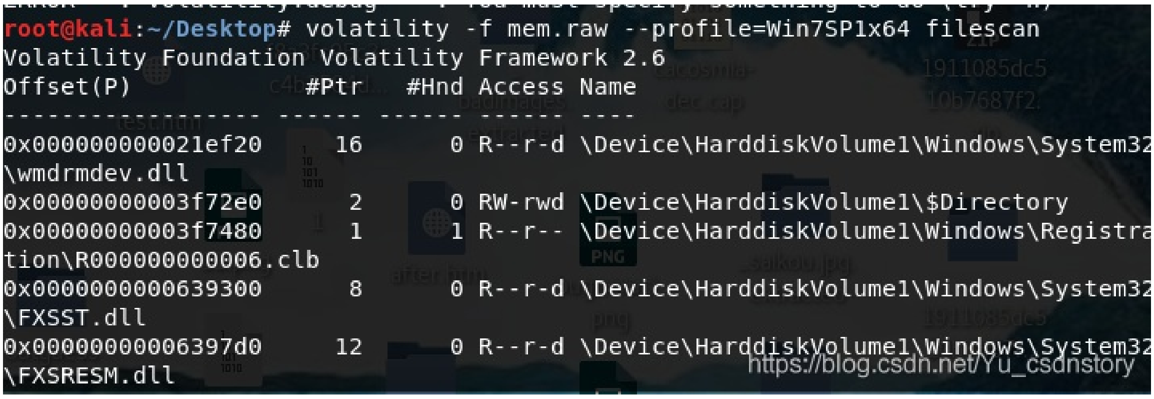
```
root@kali:~/Desktop# volatility -f mem.raw --profile=Win7SP1x64 pslist
Volatility Foundation Volatility Framework 2.6
Offset(V)          Name                    PID   PPID   Thds     Hnds   Sess  Wo
w64 Start                          Exit
------------------ -------------------- ------ ------ ------ -------- ------ ---
--- ------------------------------ ------------------------------
0xfffffa8006c84b30 System                    4      0     89      476 ------
  0 2018-08-02 10:15:41 UTC+0000
0xfffffa8008643040 smss.exe                268      4      4       29 ------
  0 2018-08-02 10:15:41 UTC+0000
0xfffffa8008e44970 csrss.exe               356    344      8      574      0
  0 2018-08-02 10:15:43 UTC+0000
0xfffffa8008f1b060 wininit.exe             408    344      7       89      0
  0 2018-08-02 10:15:43 UTC+0000
0xfffffa8008f1c4b0 csrss.exe               416    400      9      245      1
```



```
 0 2018-08-02 10:15:43 UTC+0000
0xfffffa80091cc570 winlogon.exe          472    400      6      122      1
 0 2018-08-02 10:15:44 UTC+0000
0xfffffa80091eb350 services.exe          508    408     19      247      0
 0 2018-08-02 10:15:44 UTC+0000
```

查看文件

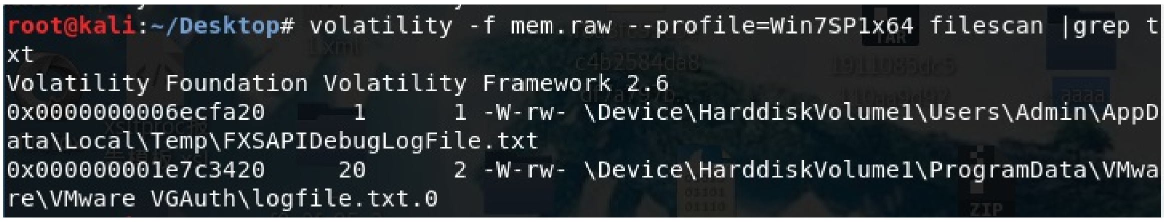
`volatility -f mem.raw --profile=Win7SP1x64 filescan`



```
root@kali:~/Desktop# volatility -f mem.raw --profile=Win7SP1x64 filescan
Volatility Foundation Volatility Framework 2.6
Offset(P)            #Ptr   #Hnd Access Name
------------------ ------ ------ ------ ----
0x000000000021ef20     16      0 R--r-d \Device\HarddiskVolume1\Windows\System32
\wmdrmdev.dll
0x000000000003f72e0     2      0 RW-rwd \Device\HarddiskVolume1\$Directory
0x000000000003f7480     1      1 R--r-- \Device\HarddiskVolume1\Windows\Registra
tion\R000000000006.clb
0x0000000000639300      8      0 R--r-d \Device\HarddiskVolume1\Windows\System32
\FXSST.dll
0x00000000006397d0     12      0 R--r-d \Device\HarddiskVolume1\Windows\System32
\FXSRESM.dll
```

一般文件会很多，不易查看，用grep命令过滤

`volatility -f mem.raw --profile=Win7SP1x64 filescan |grep txt`



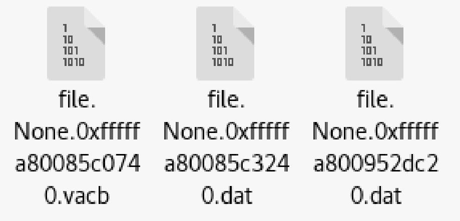
```
root@kali:~/Desktop# volatility -f mem.raw --profile=Win7SP1x64 filescan |grep t
xt
Volatility Foundation Volatility Framework 2.6
0x0000000006ecfa20      1      1 -W-rw- \Device\HarddiskVolume1\Users\Admin\AppD
ata\Local\Temp\FXSAPIDebugLogFile.txt
0x000000001e7c3420     20      2 -W-rw- \Device\HarddiskVolume1\ProgramData\VMwa
re\VMware VGAuth\logfile.txt.0
```

这里显示出了txt文件，下面将他们分离出来

提取文件

`volatility -f mem.raw --profile=Win7SP1x64 dumpfiles -Q 0x000000001e7c3420 -D aaa`

-Q是偏移量，-D是存储的文件夹



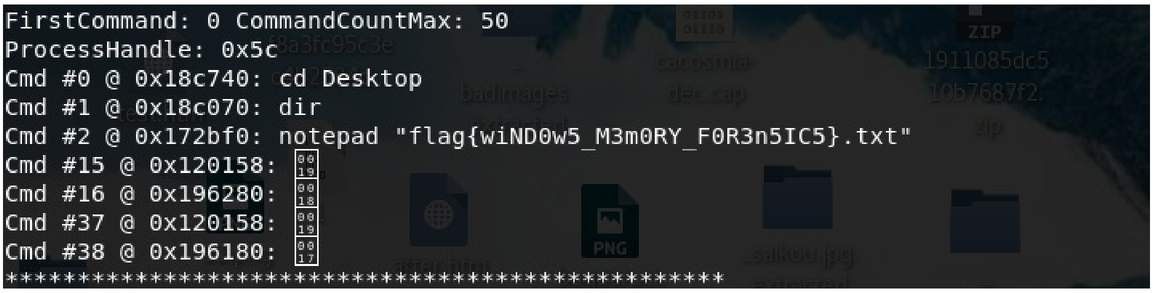
file.None.0xfffffa80085c0740.vacb

file.None.0xfffffa80085c3240.dat

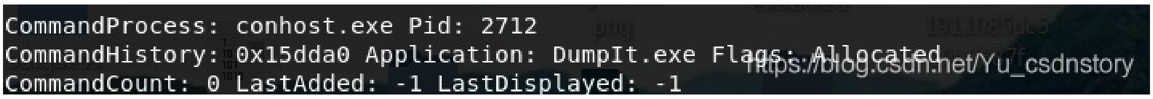
file.None.0xfffffa800952dc20.dat

查看cmd下执行的文件

`volatility -f mem.raw --profile=Win7SP1x64 cmdscan`



```
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x5c
Cmd #0 @ 0x18c740: cd Desktop
Cmd #1 @ 0x18c070: dir
Cmd #2 @ 0x172bf0: notepad "flag{wiND0w5_M3m0RY_F0R3n5IC5}.txt"
Cmd #15 @ 0x120158: □
Cmd #16 @ 0x196280: □
Cmd #37 @ 0x120158: □
Cmd #38 @ 0x196180: □
**************************************************
```



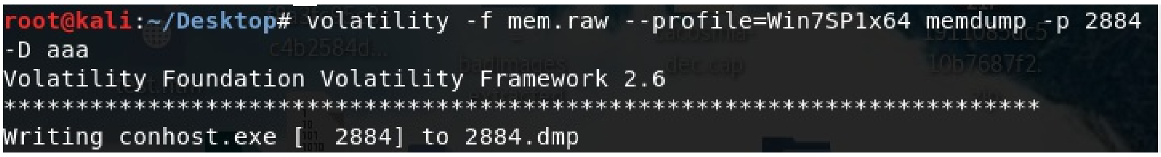
```
CommandProcess: conhost.exe Pid: 2712
CommandHistory: 0x15dda0 Application: DumpIt.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
```

可以看到，flag已经被找到，不过出题人并没有将flag放在一个文件里,只是个文件 名，若是放在某个文件里，会加大我们的难度

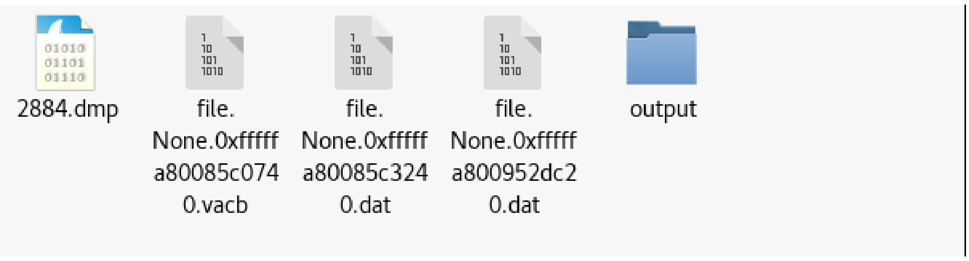
分离出cmd下执行的某个文件

`volatility -f mem.raw --profile=Win7SP1x64 memdump -p 2884 -D aaa`

-p是进程号，flag的文件在进程号为2884，分离出的文件为流量包



```
root@kali:~/Desktop# volatility -f mem.raw --profile=Win7SP1x64 memdump -p 2884
-D aaa
Volatility Foundation Volatility Framework 2.6
************************************************************************
Writing conhost.exe [  2884] to 2884.dmp
```


2884.dmp
file.None.0xfffffa80085c0740.vacb
file.None.0xfffffa80085c3240.dat
file.None.0xfffffa800952dc20.dat
output

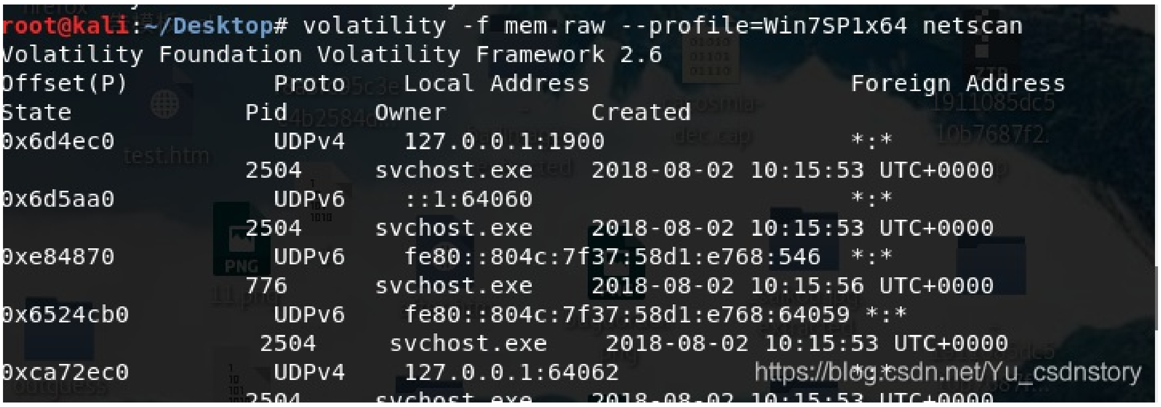
好像用wireshark打不开，可以直接foremost一波，得到文件

提取账户密码

`volatility -f mem.raw --profile=Win7SP0x64 hashpump`

查看网络连接

`volatility -f mem.raw --profile=Win7SP1x64 netscan`



```
root@kali:~/Desktop# volatility -f mem.raw --profile=Win7SP1x64 netscan
Volatility Foundation Volatility Framework 2.6
Offset(P)          Proto    Local Address                  Foreign Address
State            Pid      Owner          Created
0x6d4ec0           UDPv4    127.0.0.1:1900                 *:*
                 2504     svchost.exe    2018-08-02 10:15:53 UTC+0000
0x6d5aa0           UDPv6    ::1:64060                      *:*
                 2504     svchost.exe    2018-08-02 10:15:53 UTC+0000
0xe84870           UDPv6    fe80::804c:7f37:58d1:e768:546  *:*
                 776      svchost.exe    2018-08-02 10:15:56 UTC+0000
0x6524cb0          UDPv6    fe80::804c:7f37:58d1:e768:64059 *:*
                  2504     svchost.exe    2018-08-02 10:15:53 UTC+0000
0xca72ec0          UDPv4    127.0.0.1:64062
                 2504     svchost.exe    2018-08-02 10:15:53 UTC+0000
```

查看已经建立的网络连接

`volatility -f mem.raw --profile=Win7SP1x64 netscan|grep ESTABLISHED`