CTF 内存取证 USB流量分析

原创

MOLLMY 于 2019-09-09 23:56:25 发布 4847 收藏 25
分类专栏： CTF 笔记 网络攻防 文章标签： CTF内存取证题


CTF 同时被 3 个专栏收录
5 篇文章 0 订阅
订阅专栏


笔记
3 篇文章 0 订阅
订阅专栏


网络攻防
4 篇文章 0 订阅
订阅专栏

护网杯2019预选赛第二题

内存取证弟弟题?

题目名叫：

Baby_forensic

题目附件地址

目录

6. 分析usb.pcapng文件

知识点：

内存取证工具volatility 的使用：

volatility -f <文件名> --profile=<配置文件> <插件> [插件参数]

使用imageinfo插件来猜测dump文件的profile值：WinXPSP2x86

root@kali:~/quzhen# volatility -f mem.vmem imageinfo

grep是用来搜索特定的字符串，bgrep是用来搜索非文本数据模式和hexdump

volatility –info　用于查看volatility已经添加的profile和插件信息

Volatility -f file.raw imageinfo　判断当前镜像信息，或kdbgscan，仅适合windows内存镜像

常见插件：

Volatility -f file.raw –profile=WinXPSP2x86 notepad　查看当前展示的notepad文本

Volatility -f file.raw –profile=WinXPSP2x86 pslist　列出运行的进程，如果Exit所在的一列显示了日期时间，则表明该进程已经结束了

Hivelist　列出缓存在内存中的注册表

Filescan　扫描内存中的文件

Dumpfiles　将内存中的缓存文件导出

Volatility -f file.raw –profile=WinXPSP2x86 Memdump -p 进程号 -D ./（导出目录）　将某个进程信息导出/根据pid dump出指定进程

Foremost 2888.dmp　分析dump出的内存文件

Svcscan　扫描windows的服务

Connscan 查看网络连接

Cmdscan 查看命令行上的操作

取证方法建议

RFC 3227提供了获取数字证据的许多做法，比如，收集数据的顺序可以决定调查的成败。

这个顺序称为波动顺序（Volatility Order），顾名思义，调查人员必须首先收集易消失的数据。易失性数据是系统关闭时可能丢失的任何数据，例如连接到仍然在RAM中注册的网站。调查人员必须将先从最不稳定的证据中开始收集数据：

（1）缓存

（2）路由表，进程表，内存

（3）临时系统文件

（4）硬盘

（5）远程日志，监控数据

（6）物理网络配置，网络拓扑

（7）媒体文件（CD，DVD）

Keyboard scan code：

USB_Interrupt in 表示该USB设备为键盘[孙1]，捕捉到的键盘数据位于Leftover Capture Data域;

键盘数据由8个字节组成：

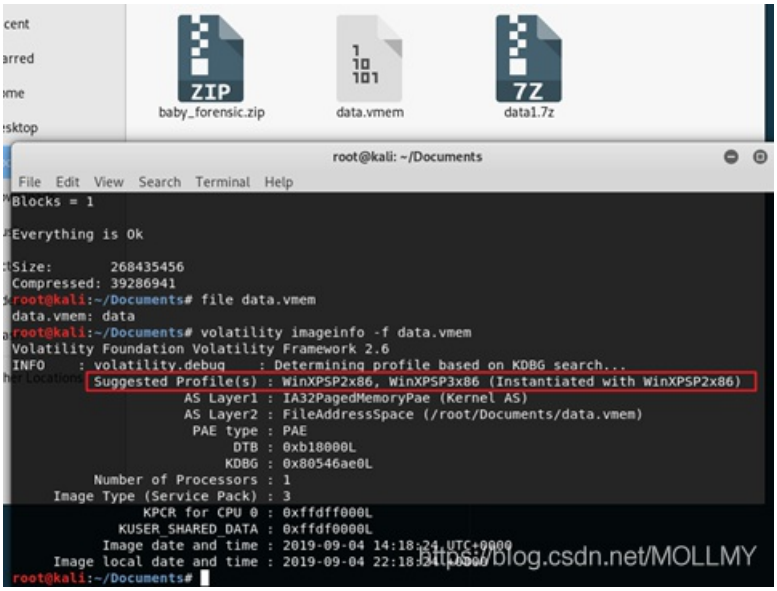
Byte 0	键盘修饰符位，SHIFT, ALT, CTRL等
Byte 1	保留位
Byte 2-7	当前按下的键，最多6个，顺序不重要（一个键只有两种状态：pressed or not pressed）

解题过程：

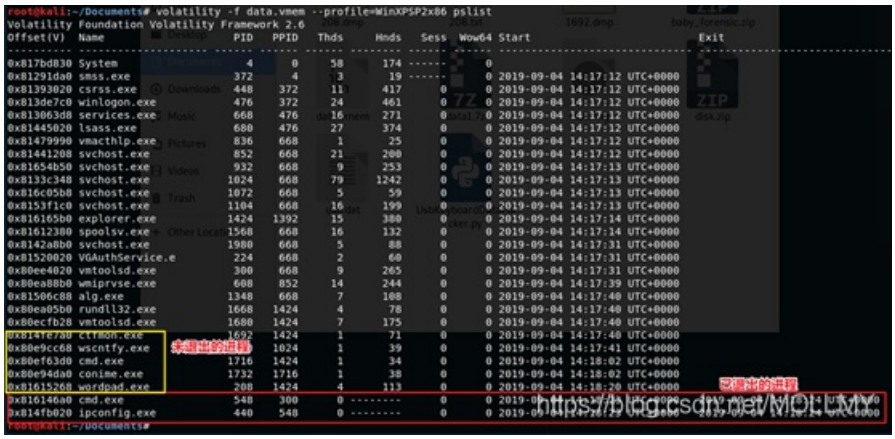
1. Kali中解压文件

由题目可知是内存取证，利用volatility工具分析解压的vmem文件

判断当前镜像信息，分析操作系统



2. pslist查看进程



进程分析：

wscntfy.exe，Windows系统关键进程，负责检查计算机的安全状态，包括防火墙、病毒防护软件、自动更新三个安全要素，如果这些服务状态不正常，系统就会在状态栏进行告警提示。这个进程也可能会被病毒软件和黑客程序伪装

ctfmon.exe，Microsoft Office产品套装的一部分，是有关输入法的一个可执行程序。它可以选择用户文字输入程序，和微软Office XP语言条。这不是纯粹的系统程序，但是如果终止它，可能会导致不可知的问题。另外，ctfmon.exe可能被感染上木马而成为病毒程序

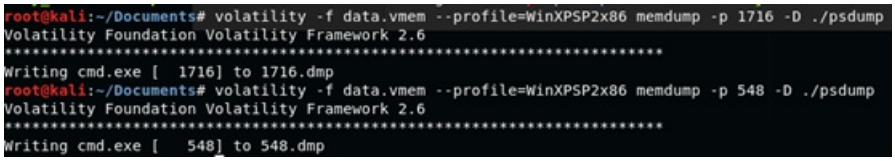
wordpad.exe，是微软Microsoft Windows自带的免费字处理工具。

Conime.exe，输入法编辑器

Cmd.exe，windows系统的命令行程序

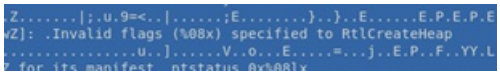
根据进程列表可以推断出当前用户可能正在使用写字板和命令行程序，可以从这两个方面入手

查看进程wordpad，保存为dmp文件



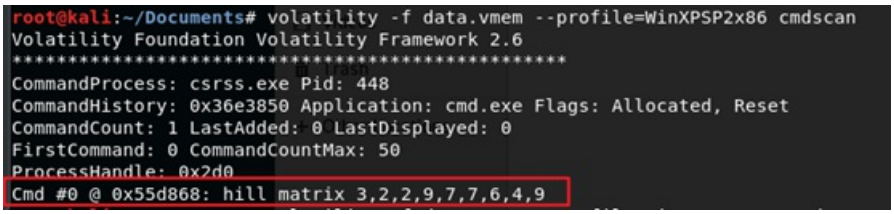
用hexeditor查看导出的dmp文件，检索flag，这几个进程文件里都有提示：

invalid flags (%08x) specified to RtlCreateHeap，



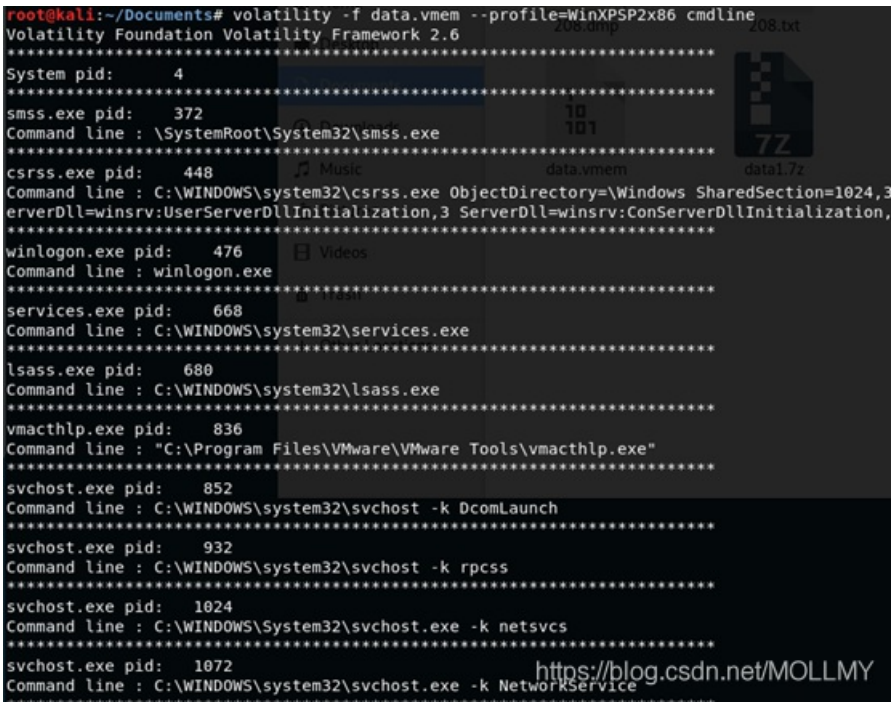
无其他发现

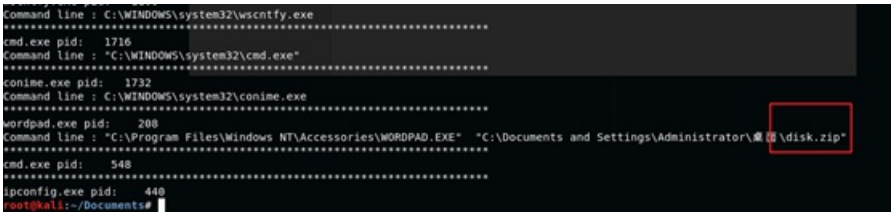
3. 通过cmdscan[孙2] 提取命令历史记录，结果如下



有用的信息：hill_matrix，应该是用到了hill cipher

通过cmdline查看命令行用到的参数：



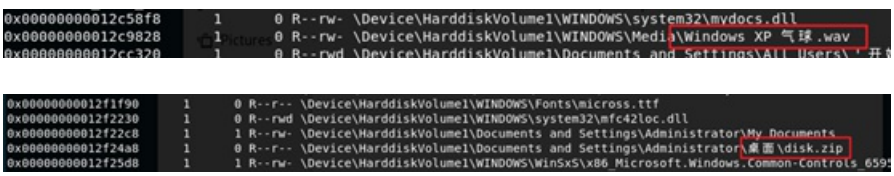


发现可能通过写字板程序产生了一个文件：disk.zip

4. 通过filescan查看文件目录，检查disk.zip:

Filescan没有发现直接的flag文件，媒体文件气球.wav也没有什么有用的信息

最后在disk.zip中寻找



Dump出媒体文件气球.wav和之前发现的disk.zip

将dump出的文件重命名为气球.wav 和disk.zip

```
root@kali:~/Documents# unzip disk.zip
Archive:  disk.zip
  inflating: disk.img
root@kali:~/Documents#
```

```
root@kali:~/Documents/filedump# ls
balloon.wav  disk.img  disk.zip
root@kali:~/Documents/filedump# ls -l
total 2052
-rw-r--r-- 1 root root    8192 Sep  9 03:27 balloon.wav
-rw-r--r-- 1 root root 2048000 Sep  4 10:11 disk.img
-rwxr-xr-x 1 root root   45056 Sep  9 03:28 disk.zip
root@kali:~/Documents/filedump# strings balloon.wav
RIFF
WAVEfmt
data
f       l
```

经查看，wav文件中没有有用的信息

disk.zip解压后得到一个镜像文件disk.img，查看它的imageinfo，无果

File命令发现是linux下的ext2文件系统：

```
root@kali:~/Documents/filedump# volatility -f disk.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/Documents/filedump/disk.img)
                      PAE type : No PAE
root@kali:~/Documents/filedump# volatility -f disk.img kdbgscan
Volatility Foundation Volatility Framework 2.6
root@kali:~/Documents/filedump# file disk.img
disk.img: Linux rev 1.0 ext2 filesystem data, UUID=63f426ba-3c01-4149-8b3e-639ff00156b9 (large files)
root@kali:~/Documents/filedump#
```

5. 使用binwalk -e命令提取文件

```
root@kali:~/Documents/filedump# ls
balloon.wav  disk.img  disk.zip
root@kali:~/Documents/filedump# binwalk -e disk.zip

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             Zip archive data, at least v2.0 to extract, compr
sk.img
42835         0xA753          End of Zip archive, footer length: 22

root@kali:~/Documents/filedump# ls
balloon.wav  disk.img  disk.zip  _disk.zip.extracted

root@kali:~/Documents/filedump# cd _disk.zip.extracted/
root@kali:~/Documents/filedump/_disk.zip.extracted# ls
0.zip  disk.img   提取出的文件
root@kali:~/Documents/filedump/_disk.zip.extracted# file 0.zip
0.zip: Zip archive data, at least v2.0 to extract
root@kali:~/Documents/filedump/_disk.zip.extracted# file disk.img
disk.img: Linux rev 1.0 ext2 filesystem data, UUID=63f426ba-3c01-4149-8b3e-639f
root@kali:~/Documents/filedump/_disk.zip.extracted# unzip 0.zip
Archive:  0.zip
replace disk.img? [y]es, [n]o, [A]ll, [N]one, [r]ename: r
new name: from_0.zip_disk.img
  inflating: from_0.zip_disk.img
root@kali:~/Documents/filedump/_disk.zip.extracted# ls
0.zip  disk.img  from_0.zip_disk.img
root@kali:~/Documents/filedump/_disk.zip.extracted# file from_0.zip_disk.img
from_0.zip_disk.img: Linux rev 1.0 ext2 filesystem data, UUID=63f426ba-3c01-414
root@kali:~/Documents/filedump/_disk.zip.extracted# binwalk -e disk.img

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             Linux EXT filesystem, rev 1.0, ext2 filesystem da
```

解压0.zip并重命名

0.zip解压出来与当前的disk.img相同，所以可以直接binwalk -e disk.img 进一步提取这个文件

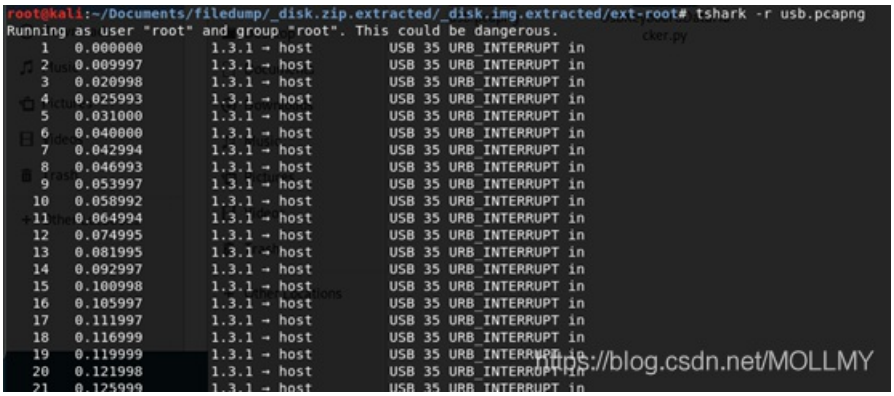

```
root@kali:~/Documents/filedump/_disk.zip.extracted# cd _disk.img.extracted/
root@kali:~/Documents/filedump/_disk.zip.extracted/_disk.img.extracted# ls
0.ext  ext-root
root@kali:~/Documents/filedump/_disk.zip.extracted/_disk.img.extracted# cd ext-root
root@kali:~/Documents/filedump/_disk.zip.extracted/_disk.img.extracted/ext-root# ls
usb.pcapng
```

经过两次binwalk 提取到以上文件，发现一个usb.pcapng文件（usb流量分析）

6. 分析usb.pcapng文件

（虽然有现成的脚本跑USB流量的pacp文件，但是不看怎么知道是键盘流量呢?（命名里写了usb），所以还是要一步一步来呀）

Tshark -r 查看，信息量有点大：



USB_Interrupt in 表示该USB设备为键盘[孙3]，捕捉到的键盘数据位于Leftover Capture Data域，用tshark -T fields -e usb.capdata 查看键盘数据

linux命令后加 ” > file.path.name” 可以输出存为文件

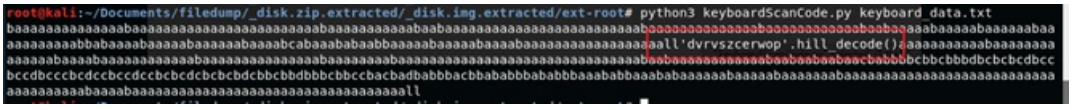
根据keyboard scan code表（参考

https://www.shsu.edu/csc_tjm/fall2000/cs272/scan_codes.html）写脚本提取之。

```
'''
2019/9/9
@author：sunmx
USB流量分析
    keyboard scan code转为键盘字符
参数：
    导出的usb流量信息
'''
import sys
import os
```

```
usb_codes = {
   0x04:"aA", 0x05:"bB", 0x06:"cC", 0x07:"dD", 0x08:"eE", 0x09:"fF",
   0x0A:"gG", 0x0B:"hH", 0x0C:"iI", 0x0D:"jJ", 0x0E:"kK", 0x0F:"lL",
   0x10:"mM", 0x11:"nN", 0x12:"oO", 0x13:"pP", 0x14:"qQ", 0x15:"rR",
   0x16:"sS", 0x17:"tT", 0x18:"uU", 0x19:"vV", 0x1A:"wW", 0x1B:"xX",
   0x1C:"yY", 0x1D:"zZ", 0x1E:"1!", 0x1F:"2@", 0x20:"3#", 0x21:"4$",
   0x22:"5%", 0x23:"6^", 0x24:"7&", 0x25:"8*", 0x26:"9(", 0x27:"0)",
   0x2C:"  ", 0x2D:"-_", 0x2E:"=+", 0x2F:"[{", 0x30:"]}",  0x32:"#~",
   0x33:";:", 0x34:"'\"",  0x36:",<",  0x37:".>", 0x4f:">", 0x50:"<"
   }

def code2chr(filepath):
    lines = []
    pos = 0
    for x in open(filepath,"r").readlines():
        code = int(x[6:8],16)   # 即第三个字节
        if code == 0:
            continue
        # newline or down arrow - move down
        if code == 0x51 or code == 0x28:
            pos += 1
            continue
        # up arrow - move up
        if code == 0x52:
            pos -= 1
            continue

        # select the character based on the Shift key
        while len(lines) <= pos:
            lines.append("")
        if code in range(4,81):
            if int(x[0:2],16) == 2:
                lines[pos] += usb_codes[code][1]
            else:
                lines[pos] += usb_codes[code][0]

    for x in lines:
        print(x)


if __name__ == "__main__":
    # check argv
    if len(sys.argv) != 2:
        print("Usage:\n\tpython keyboardScanCode.py datafile.txt\nhow to get datafile:\t tshark -r file.usb
        exit(1)
    else:
        filepath = sys.argv[1]
        code2chr(filepath)
```

运行结果：



在线解密：

得到LHBDSLUDRIDT

转为小写 即为flag

别问我为什么转小写，我也不知道

提交大写的不对，小写显示正确

[孙1]https://bitvijays.github.io/LFC-Forensics.html#usb-keyboard

[孙2]The cmdscan plugin searches the memory of csrss.exe on XP/2003/Vista/2008 and conhost.exe on Windows 7 for commands that attackers entered through a console shell (cmd.exe). This is one of the most powerful commands you can use to gain visibility into an attackers actions on a victim system, whether they opened cmd.exe through an RDP session or proxied input/output to a command shell from a networked backdoor.

[孙3]https://bitvijays.github.io/LFC-Forensics.html#usb-keyboard