volatility食用方法

文章目录

_

- 九. 软件 与 导出
-
- IE
- 查看iexplore进程pid
- 提取某个进程
- 提取某个文件
- 复制、剪切版信息

- 其它

Volatility是开源的Windows，Linux，MaC，Android的内存取证分析工具，由python编写成，命令行操作，支持各种操作系统。

项目地址：
https://www.volatilityfoundation.org/releases
https://code.google.com/archive/p/volatility/
https://github.com/volatilityfoundation/volatility/wiki/Command-Reference#consoles
插件
https://github.com/ruokeqx/tool-for-CTF

安装：
kali-bt5自带此工具

volattity命令格式及常用参数：

```
volatility 使用:
        volatility -f <文件名> --profile=<配置文件> <插件> [插件参数]
通过volatility --info获取工具所支持的profile，Address Spaces，Scanner Checks，Plugins

常用插件:
imageinfo：显示目标镜像的摘要信息，知道镜像的操作系统后，就可以在 –profile 中带上对应的操作系统
pslist：该插件列举出系统进程，但它不能检测到隐藏或者解链的进程，psscan可以
psscan：可以找到先前已终止(不活动)的进程以及被rootkit隐藏或解链的进程
pstree：以树的形式查看进程列表，和pslist一样，也无法检测隐藏或解链的进程
mendump：提取出指定进程，常用foremost 来分离里面的文件
filescan：扫描所有的文件列表
hashdump：查看当前操作系统中的 password hash，例如 Windows 的 SAM 文件内容
svcscan：扫描 Windows 的服务
connscan：查看网络连接
```

参数大全（google翻译）

```
 Supported Plugin Commands:

amcache         Print AmCache information   //打印AmCache信息
apihooks        Detect API hooks in process and kernel memory   //检测进程和内核内存中的API挂钩
atoms           Print session and window station atom tables    //打印会话和窗口站atom表
atomscan        Pool scanner for atom tables    //用于atom表的池扫描程序
auditpol        Prints out the Audit Policies from HKLM\SECURITY\Policy\PolAdtEv   //auditpol从HKLM \ SECURITY \ Policy \ PolAdtEv中打印出审核策略
bigpools        Dump the big page pools using BigPagePoolScanner    // bigpools使用BigPagePoolScanner转储大页面池
bioskbd         Reads the keyboard buffer from Real Mode memory     // bioskbd从实模式内存中读取键盘缓冲区
cachedump       Dumps cached domain hashes from memory          //从内存中转储缓存的域哈希
callbacks       Print system-wide notification routines         //回调打印系统范围的通知例程
clipboard       Extract the contents of the windows clipboard     //剪贴板提取Windows剪贴板的内容
cmdline         Display process command-line arguments          //显示进程命令行参数
cmdscan         Extract command history by scanning for _COMMAND_HISTORY   //通过扫描_COMMAND_HISTORY提取命令历史记录
connections     Print list of open connections [Windows XP and 2003 Only]    //列印已开启的连接[只适用于windowsxp及2003]
connscan        Pool scanner for tcp connections    //用于tcp连接的池扫描程序
consoles        Extract command history by scanning for _CONSOLE_INFORMATION   通过扫描_CONSOLE_INFORMATION提取命令历史记录
crashinfo       Dump crash-dump information     //转储崩溃转储信息
deskscan        Poolscaner for tagDESKTOP (desktops)    //用于tagDESKTOP的Poolscaner（桌面）
devicetree      Show device tree    //显示设备树
dlldump         Dump DLLs from a process address space      //从进程地址空间中转储DLL
dlllist         Print list of loaded dlls for each process    //打印每个进程已加载的dll的列表
driverirp       Driver IRP hook detection     //驱动程序IRP挂钩检测
drivermodule    Associate driver objects to kernel modules      //将驱动程序对象与内核模块相关联
driverscan      Pool scanner for driver objects     //池扫描程序中的驱动程序对象
dumpcerts       Dump RSA private and public SSL keys      //转储RSA专用和公用SSL密钥
dumpfiles       Extract memory mapped and cached files      //取内存映射和缓存的文件
dumpregistry    Dumps registry files out to disk        //将注册表文件转储到磁盘
editbox         Displays information about Edit controls. (Listbox experimental.)      //显示有关“编辑”控件的信息。 （实验性的列表框。）
envars          Display process environment variables     //显示流程环境变量
eventhooks      Print details on windows event hooks      //在Windows事件挂钩上打印详细信息

```

```
33  evtlogs         Extract Windows Event Logs (XP/2003 only)   //提取Windows
34  事件日志（仅适用于XP / 2003）
    filescan        Pool scanner for file objects     //池扫描程序中的文件对象
    gahti           Dump the USER handle type information     //转储USER句柄类
35  型信息               gditimers       Print installed GDI timers and callb
36  acks    //打印已安装的GDI计时器和回调
    gdt             Display Global Descriptor Table     //显示全局描述符表
37  getservicesids  Get the names of services in the Registry and return Calc
    ulated SID      //获取注册表中的服务名称，并返回计算出的SID
38  getsids         Print the SIDs owning each process        //打印拥有每个进
    程的SID
39  handles         Print list of open handles for each process   //打印每个进
    程的打开句柄列表
40  hashdump        Dumps passwords hashes (LM/NTLM) from memory    //从内存中
41  转储密码散列(LM/NTLM
42  hibinfo         Dump hibernation file information     //转储休眠文件信息
43  hivedump        Prints out a hive     //打印一个配置单元
    hivelist        Print list of registry hives.   //打印注册表配置单元列表。
44  hivescan        Pool scanner for registry hives     //注册表配置单元的池扫描
    程序
45  hpakextract     Extract physical memory from an HPAK file     //从HPAK文件
46  提取物理内存
47  hpakinfo        Info on an HPAK file      //有关HPAK文件的信息
    idt             Display Interrupt Descriptor Table      //显示中断描述符表
48  iehistory       Reconstruct Internet Explorer cache / history   //重建Int
    ernet Explorer缓存/历史记录
49  imagecopy       Copies a physical address space out as a raw DD image
50    //将物理地址空间复制为原始DD图像
    imageinfo       Identify information for the image      //标识图像信息
51  impscan         Scan for calls to imported functions    //扫描对导入功能的调
    用
52  joblinks        Print process job link information      //打印过程作业链接信
    息
53  kdbgscan        Search for and dump potential KDBG values     //搜索并转储
    潜在的KDBG值
54  kpcrscan        Search for and dump potential KPCR values     //搜索和转储
55  潜在的KPCR值
    ldrmodules      Detect unlinked DLLs      //检测链接dll
56  lsadump         Dump (decrypted) LSA secrets from the registry      //从
    注册表转储(解密的)LSA机密
57  machoinfo       Dump Mach-O file format information     //转储Mach-O文件格
58  式信息
    malfind         Find hidden and injected code     //找到隐藏的和注入的代码
59  mbrparser       Scans for and parses potential Master Boot Records (MBR
    s)      //扫描和解析潜在主引导记录(mbr)
60  memdump         Dump the addressable memory for a process     //转储进程的
61  可寻址内存
    memmap          Print the memory map      //打印内存映射
62  messagehooks    List desktop and thread window message hooks    //列出桌面
    和线程窗口消息挂钩
63  mftparser       Scans for and parses potential MFT entries      //扫描和解
    析潜在的MFT条目
64  moddump         Dump a kernel driver to an executable file sample     //
65  将内核驱动程序转储到可执行文件示例
66  modscan         Pool scanner for kernel modules       //内核模块的池扫描程序
67  modules         Print list of loaded modules        //打印加载模块的列表
68  multiscan       Scan for various objects at once      //一次扫描各种物体
```

```
mutantscan      Pool scanner for mutex objects        //池扫描互斥对象
notepad         List currently displayed notepad text   //列表当前显示的记事本文本
objtypescan     Scan for Windows object type objects    //扫描Windows对象类型的对象
patcher         Patches memory based on page scans      //基于页面扫描的内存补丁
poolpeek        Configurable pool scanner plugin      //可配置的池扫描器插件
printkey        Print a registry key, and its subkeys and values      //打印注册表项及其子项和值
privs           Display process privileges      //显示过程的特权
procdump        Dump a process to an executable file sample     //将进程转储到可执行文件示例
pslist          Print all running processes by following the EPROCESS lists      //按照EPROCESS列表打印所有正在运行的进程
psscan          Pool scanner for process objects      //进程对象的池扫描程序
pstree          Print process list as a tree      //以树的形式打印过程列表
psxview         Find hidden processes with various process listings      //使用各种进程列表查找隐藏的进程
qemuinfo        Dump Qemu information     //转储Qemu信息
raw2dmp         Converts a physical memory sample to a windbg crash dump     //将物理内存示例转换为windbg崩溃转储
screenshot      Save a pseudo-screenshot based on GDI windows     //保存一个基于GDI窗口的伪截图
servicediff     List Windows services (ala Plugx)     //列出Windows服务(ala Plugx)
sessions        List details on _MM_SESSION_SPACE (user logon sessions)      //列出关于_MM_SESSION_SPACE(用户登录会话)的详细信息
shellbags       Prints ShellBags info     //打印ShellBags信息
shimcache       Parses the Application Compatibility Shim Cache registry key    //解析应用程序兼容性垫片缓存注册表项
shutdowntime    Print ShutdownTime of machine from registry     //从注册表打印停机时间的机器
sockets         Print list of open sockets      //打印打开的套接字列表
sockscan        Pool scanner for tcp socket objects     //用于tcp套接字对象的池扫描程序
ssdt            Display SSDT entries      //SSDT条目显示
strings         Match physical offsets to virtual addresses (may take a while, VERY verbose)     //将物理偏移量匹配到虚拟地址(可能需要一段时间，非常冗长)
svcscan         Scan for Windows services     //扫描Windows服务
symlinkscan     Pool scanner for symlink objects      //符号链接对象的池扫描程序
thrdscan        Pool scanner for thread objects       //线程对象的池扫描程序
threads         Investigate _ETHREAD and _KTHREADs
timeliner       Creates a timeline from various artifacts in memory      //从内存中的各种工件创建时间线
timers          Print kernel timers and associated module DPCs      //打印内核计时器和相关模块DPCs
truecryptmaster Recover TrueCrypt 7.1a Master Keys      //恢复TrueCrypt 7.1a主密钥
truecryptpassphrase     TrueCrypt Cached Passphrase Finder      //TrueCrypt缓存了密码短语查找器
truecryptsummary        TrueCrypt Summary     //TrueCrypt总结
unloadedmodules Print list of unloaded modules      //打印已卸载模块列表
userassist      Print userassist registry keys and information      //打印userassist注册表项和信息
userhandles     Dump the USER handle tables     //转储用户句柄表
```

```
105 vaddump         Dumps out the vad sections to a file    //将vad节转储到一个
    文件中
106 vadinfo         Dump the VAD info     //转储VAD信息
107 vadtree         Walk the VAD tree and display in tree format      //遍历V
108 AD树并以树格式显示
    vadwalk         Walk the VAD tree     //走在树下
109 vboxinfo        Dump virtualbox information     //转储virtualbox信息
    verinfo         Prints out the version information from PE images     //
110 从PE图像打印出版本信息
111 vmwareinfo      Dump VMware VMSS/VMSN information       //转储VMware VMS
    S/VMSN信息
112 volshell        Shell in the memory image     //贝壳在记忆中的形象
    windows         Print Desktop Windows (verbose details)       //打印桌面窗
113 口(详细信息)
114 wintree         Print Z-Order Desktop Windows Tree      //打印z顺序桌面Wind
    ows树
    wndscan         Pool scanner for window stations      //池扫描窗口站
    yarascan        Scan process or kernel memory with Yara signatures
    //用Yara签名扫描进程或内核内存
```

Linux

```
linux_apihooks              – 检查用户名apihooks
linux_arp                   – 打印ARP表
linux_aslr_shift            – 自动检测Linux aslr改变
linux_banner                – 打印Linux Banner信息
linux_bash                  – 从bash进程内存中恢复bash历史记录
linux_bash_env              – 恢复一个进程的动态环境变量
linux_bash_hash             – 从bash进程内存中恢复bash哈希表
linux_check_afinfo          – 验证网络协议的操作函数指针
linux_check_creds           – 检查是否有任何进程正在共享凭证结构
linux_check_evt_arm         – 检查异常向量表以查找系统调用表钩子
linux_check_fop             – 检查rootkit修改的文件操作结构
linux_check_idt             – 检查IDT是否被更改
linux_check_inline_kernel   – 检查内联内核挂钩
linux_check_modules         – 将模块列表与sysfs信息进行比较
linux_check_syscall         – 检查系统调用表是否已被更改
linux_check_tty             – 检查tty的钩子
linux_cpuinfo               – 打印有关每个活动处理器的信息
linux_dentry_cache          – 从dentry缓存收集文件
linux_dmesg                 – 收集dmesg buffer
linux_dump_map              – 将选定的内存映射写入到磁盘
linux_dynamic_env           – 恢复进程的动态环境变量
linux_elfs                  – 在进程映射中找ELF二进制文件
linux_enumerate_files       – 列出文件系统缓存引用的文件
linux_find_file             – 列出并从内存中恢复文件
linux_getcwd                – 列出每个进程的当前工作目录
linux_hidden_modules        – Carves内存寻找隐藏的内核模块
linux_ifconfig              – 收集活动接口
linux_info_regs             – GDB中的“info寄存器”。它打印出所有的输出
linux_iomem                 – 提供与/proc/iomem相似的输出
linux_kernel_opened_files   – 列出从内核中打开的文件
linux_keyboard_notifiers    – 解析键盘通知调用链
linux_ldrmodules            – 将proc映射的输出与libdl中的库列表进行比较
linux_library_list          – 将库加载到一个进程中
linux_librarydump           – 将进程内存中的共享库转储到磁盘
linux_list_raw              – 列出应用程序与混杂的套接字
linux_lsmod                 – 收集加载内核模块
linux_lsof                  – 列出文件描述符及其路径
linux_malfind               – 查找可疑的过程映射
linux_memmap                – 转储用于linux任务的内存映射
linux_moddump               – 提取加载内核模块
linux_mount                 – 收集挂载的fs/devices
linux_mount_cache           – 收集从kmem_cache安装的fs/设备。
linux_netfilter             – 列出Netfilter钩子
linux_netscan               – 刻画网络连接结构
linux_netstat               – 列表打开的套接字
linux_pidhashtable          – 通过PID哈希表枚举进程
linux_pkt_queues            – 将每个进程的数据包队列写入磁盘
linux_plthook               – 扫描ELF二进制文件' PLT hooks
linux_proc_maps             – 收集进程内存映射
linux_proc_maps_rb          – 通过映射红黑树收集linux的进程映射
linux_procdump              – 将进程的可执行映像转储到磁盘
linux_process_hollow        – 检查是否有进程被挖空的迹象
linux_psaux                 – 收集进程和完整的命令行和开始时间
linux_psenv                 – 收集进程及其静态环境变量
linux_pslist                – 收集活动任务通过task_struct->task list
linux_pslist_cache          – 从kmem_cache中收集计划任务
```

```
58   linux_psscan               - 扫描进程的物理内存
59   linux_pstree               - 显示进程之间的父/子关系
60   linux_psxview              - 查找隐藏进程与各种各样的进程列表
61   linux_recover_filesystem   - 从内存中恢复整个缓存的文件系统
62   linux_route_cache          - 从内存中恢复路由缓存
63   linux_sk_buff_cache        - 从sk_buff kmem_cache中恢复数据包
64   linux_slabinfo             - 在一台正在运行的机器上模拟/proc/slabinfo。
     linux_strings              - 将物理偏移量匹配到虚拟地址(可能需要一段时间，非常详
65   细)
66   linux_threads              - 打印进程的线程
67   linux_tmpfs                - 从内存中恢复tmpfs文件系统。
68   linux_truecrypt_passphrase - 恢复缓存Truecrypt口令
69   linux_vma_cache            - 从vm_area_struct 缓存中收集VMAs
70   linux_volshell             - 内存映像中的shell
     linux_yarascan             - Linux内存映像中的一个shell
```

一. 常见windows工具进程名

```
1   TrueCrypt.exe 磁盘加密工具
2   notepad.exe 自带记事本
3   mspaint.exe 自带画图工具
4   iexplore.exe IE浏览器
5   DumpIt.exe 内存镜像提取工具
```

二. 识别内存文件信息

```
1   $ volatility -f mem.dump imageinfo
```

```
root@kali-linux-wp:/home/dump# volatility -f mem.dump imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64, Win2008R2SP1x64_24000, W
in2008R2SP1x64_23418, Win2008R2SP1x64, Win7SP1x64_24000, Win7SP1x64_23418
                     AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/dump/mem.dump)
                      PAE type : No PAE
                           DTB : 0x187000L
                          KDBG : 0xf80003e02110L
          Number of Processors : 1
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0xfffff80003e03d00L
             KUSER_SHARED_DATA : 0xfffff78000000000L
           Image date and time : 2019-11-13 08:39:44 UTC+0000
     Image local date and time : 2019-11-13 16:39:44 +0800
root@kali-linux-wp:/home/dump#
```

https://blog.csdn.net/qq_38626043

三. 进程信息

查看进程信息

熟悉常见进程，查看启动时间，大小等信息

```
$ volatility -f mem.dump --profile=Win7SP1x64 pslist
```

```
root@kali-linux-wp:/home/dump# volatility -f mem.dump --profile=Win7SP1x64 pslist
Volatility Foundation Volatility Framework 2.6
Offset(V)          Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
------------------ -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
0xfffffa800ccc1b10 System                    4      0     88      534 ------      0 2019-11-13 08:31:48 UTC+0000
0xfffffa800d2fbb10 smss.exe                252      4      2       29 ------      0 2019-11-13 08:31:48 UTC+0000
0xfffffa800e2227e0 csrss.exe               344    328      9      400      0      0 2019-11-13 08:31:49 UTC+0000
0xfffffa800e3f3340 wininit.exe             396    328      3       79      0      0 2019-11-13 08:31:49 UTC+0000
0xfffffa800e3f77d0 csrss.exe               404    388     10      225      1      0 2019-11-13 08:31:49 UTC+0000
0xfffffa800e41fb10 winlogon.exe            444    388      3      111      1      0 2019-11-13 08:31:49 UTC+0000
0xfffffa800e457060 services.exe            500    396      8      210      0      0 2019-11-13 08:31:49 UTC+0000
0xfffffa800e426b10 lsass.exe               508    396      6      554      0      0 2019-11-13 08:31:49 UTC+0000
0xfffffa800e464060 lsm.exe                 516    396      9      145      0      0 2019-11-13 08:31:49 UTC+0000
0xfffffa800e4f8b10 svchost.exe             608    500     10      351      0      0 2019-11-13 08:31:50 UTC+0000
0xfffffa800e52bb10 svchost.exe             684    500      8      273      0      0 2019-11-13 08:31:50 UTC+0000
0xfffffa800e570b10 svchost.exe             768    500     21      443      0      0 2019-11-13 08:31:50 UTC+0000
0xfffffa800e5b5b10 svchost.exe             816    500     16      381      0      0 2019-11-13 08:31:50 UTC+0000
0xfffffa800e5d7870 svchost.exe             860    500     18      666      0      0 2019-11-13 08:31:50 UTC+0000
0xfffffa800e5f8b10 svchost.exe             888    500     37      919      0      0 2019-11-13 08:31:50 UTC+0000
0xfffffa800e66c870 svchost.exe            1016    500      5      114      0      0 2019-11-13 08:31:50 UTC+0000
0xfffffa800e74fb10 svchost.exe            1032    500     15      364      0      0 2019-11-13 08:31:51 UTC+0000
0xfffffa800e510320 spoolsv.exe            1156    500     13      273      0      0 2019-11-13 08:31:51 UTC+0000
0xfffffa800e5b0060 svchost.exe            1184    500     11      194      0      0 2019-11-13 08:31:51 UTC+0000
0xfffffa800e56e060 svchost.exe            1276    500     10      155      0      0 2019-11-13 08:31:52 UTC+0000
0xfffffa800e685060 svchost.exe            1308    500     12      228      0      0 2019-11-13 08:31:52 UTC+0000
0xfffffa800e632060 svchost.exe            1380    500      4      167      0      0 2019-11-13 08:31:52 UTC+0000
0xfffffa800e692060 VGAuthService.         1480    500      4       94      0      0 2019-11-13 08:31:52 UTC+0000
0xfffffa800e7dab10 vmtoolsd.exe           1592    500     11      287      0      0 2019-11-13 08:31:52 UTC+0000
0xfffffa800e8a7720 svchost.exe            1824    500      6       92      0      0 2019-11-13 08:31:53 UTC+0000
0xfffffa800e898300 WmiPrvSE.exe           1980    608     10      203      0      0 2019-11-13 08:31:53 UTC+0000
0xfffffa800e8e9b10 dllhost.exe            2044    500     15      197      0      0 2019-11-13 08:31:53 UTC+0000
                                                                                      https://blog.csdn.net/qq_38626043
```

四. cmd历史命令

通过扫描_COMMAND_HISTORY提取命令历史记录

```
$ volatility -f mem.dump --profile=Win7SP1x64 cmdscan
```

```
root@kali-linux-wp:/home/dump# volatility -f mem.dump --profile=Win7SP1x64 cmdscan
Volatility Foundation Volatility Framework 2.6
**************************************************
CommandProcess: conhost.exe Pid: 2632
CommandHistory: 0x242350 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 1 LastAdded: 0 LastDisplayed: 0
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x60
Cmd #0 @ 0x2229d0: flag.ccx_password_is_same_with_Administrator
**************************************************
CommandProcess: conhost.exe Pid: 2748
CommandHistory: 0x2926d0 Application: DumpIt.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x60
root@kali-linux-wp:/home/dump#
```

https://blog.csdn.net/qq_38626043

五. 文件

5.1 搜索地址池里的文件 filescan

```
$ volatility -f mem.dump --profile=Win7SP1x64 filescan | grep flag.cc
```

```
root@kali-linux-wp:/home/dump# volatility -f mem.dump --profile=Win7SP1x64 filescan | grep flag.cc
x
Volatility Foundation Volatility Framework 2.6
0x000000003e435890     15      0 R--rw- \Device\HarddiskVolume2\Users\Administrator\Desktop\flag.ccx
root@kali-linux-wp:/home/dump#
```

5.2 扫描桌面文件

```
$ volatility -f attachment.vmem --profile=Win7SP1x64 filescan | grep "Desktop"
```

```
root@kali-linux-wp:/home/13# volatility -f attachment.vmem --profile=Win7SP1x64 filescan | grep "Desktop"
Volatility Foundation Volatility Framework 2.6
0x000000007e5c6c80      2      1 R--rwd \Device\HarddiskVolume1\Users\Public\Desktop
0x000000007e5c73d0      2      1 R--rwd \Device\HarddiskVolume1\Users\PC\Desktop
0x000000007e5c8070      2      1 R--rwd \Device\HarddiskVolume1\Users\Public\Desktop
0x000000007e5cb440      2      1 R--rwd \Device\HarddiskVolume1\Users\PC\Desktop
0x000000007e5d7960     16      0 R--rwd \Device\HarddiskVolume1\Users\PC\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Accessories\Desktop.ini
0x000000007e5da790     16      0 R--rwd \Device\HarddiskVolume1\Users\PC\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Accessories\System Tools\Desktop.ini
0x000000007e5db070     16      0 R--rwd \Device\HarddiskVolume1\Users\PC\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Maintenance\Desktop.ini
0x000000007e5dcac0     16      0 R--rwd \Device\HarddiskVolume1\Users\PC\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Accessories\Accessibility\Desktop.ini
0x000000007e5dcc10     16      0 R--rwd \Device\HarddiskVolume1\ProgramData\Microsoft\Windows\Start Menu\Programs\Accessories\Accessibility\Desktop.ini
0x000000007e5de900     16      0 R--rwd \Device\HarddiskVolume1\ProgramData\Microsoft\Windows\Start Menu\Programs\Accessories\Desktop.ini
0x000000007e5e0a40     16      0 R--rwd \Device\HarddiskVolume1\ProgramData\Microsoft\Windows\Start Menu\Programs\Maintenance\Desktop.ini
0x000000007e5e1200     16      0 R--rwd \Device\HarddiskVolume1\ProgramData\Microsoft\Windows\Start Menu\Programs\Accessories\Tablet PC\Desktop.ini
0x000000007e5e2a40     16      0 R--rwd \Device\HarddiskVolume1\ProgramData\Microsoft\Windows\Start Menu\Programs\Accessories\System Tools\Desktop.ini
0x000000007e708dd0     16      0 R--rwd \Device\HarddiskVolume1\Users\Public\Desktop\desktop.ini
0x000000007eae6f20      2      0 R--rwd \Device\HarddiskVolume1\Users\PC\Desktop\desktop.ini
root@kali-linux-wp:/home/13#
```

https://blog.csdn.net/qq_38626043

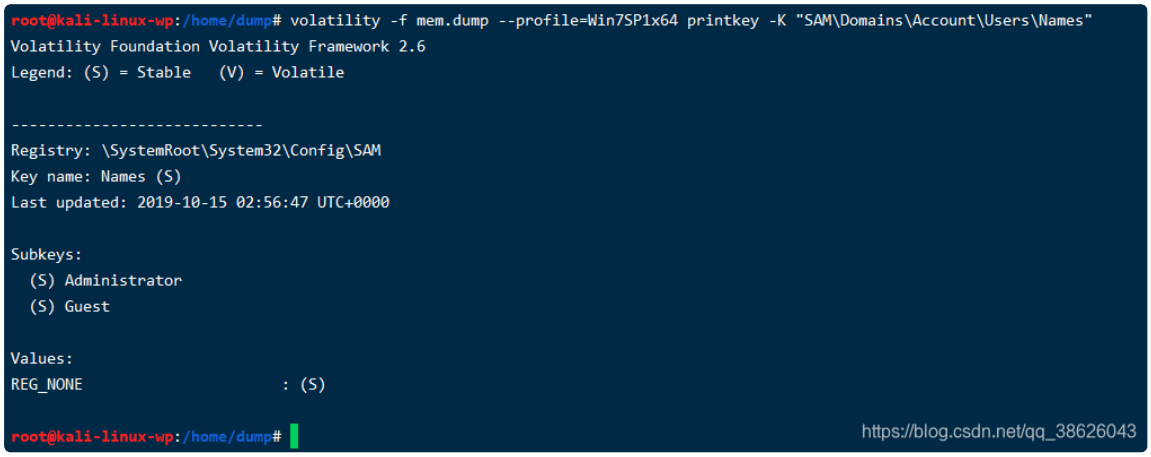
5.3 查找图片

```
$ volatility -f name--profile=Win7SP1x64 filescan | grep -E 'jpg|png|jpeg|bmp|gif
```

六. 注册表

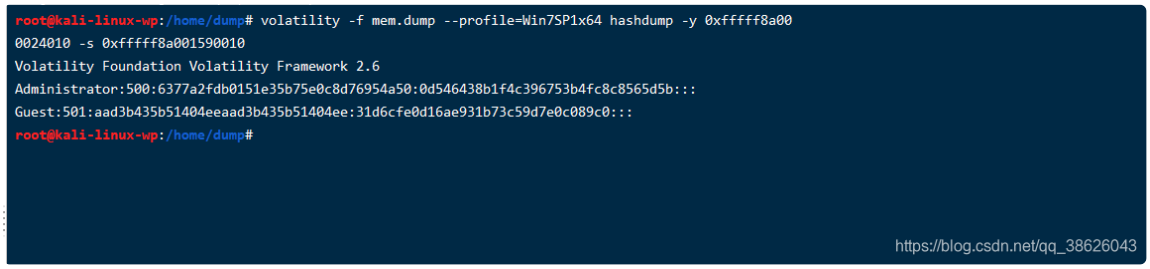
输出SAM\Domains\Account\Users\Names注册表子项目

```
$ volatility -f mem.dump --profile=Win7SP1x64 printkey -K "SAM\Domains\Account\Users\Names"
```



6.2 获取指定地址sha内容

```
$ volatility -f mem.dump --profile=Win7SP1x64 hashdump -y 0xfffff8a00
```



6.3 打印HKEY_LOCAL_MACHINE \ Microsoft \ Security Center \ Svc密钥

```
$ volatility -f mem.dump --profile=Win7SP1x64 printkey -K "Microsoft\Security Center\Svc"
```

```
root@kali-linux-wp:/home/dump# volatility -f mem.dump --profile=Win7SP1x64 printkey -K "Microsoft\Security Center\Svc"
Volatility Foundation Volatility Framework 2.6
Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: \SystemRoot\System32\Config\SOFTWARE
Key name: Svc (S)
Last updated: 2016-09-21 12:15:34 UTC+0000

Subkeys:

Values:
REG_QWORD     VistaSp1        : (S) 128920218544262440
REG_DWORD     AntiVirusOverride : (S) 0
REG_DWORD     AntiSpywareOverride : (S) 0
REG_DWORD     FirewallOverride : (S) 0
root@kali-linux-wp:/home/dump#
```
https://blog.csdn.net/qq_38626043

6.3 注册表解析

```
1   $ volatility -f name --profile=Win7SP1x64 hivelist
```

七. 密码

7.1 输出lsa解码信息

http://moyix.blogspot.com/2008/02/decrypting-lsa-secrets.html

```
1   $ volatility -f attachment.vmem --profile=Win7SP1x64 lsadump
```

$MACHINE.ACC$：域身份验证$Microsoft$。$DefaultPassword$：启用自动登录后用于登录$Windows$的密
KM：用于加密缓存的域密码的密钥解密LSA密钥。
L $RTMTIMEBOMB_*$：时间戳记提供未激活的$Windows$副本停止工作的日期。L
HYDRAENCKEY_ *：用于远程桌面协议（RDP）的私钥。如果您还从受RDP攻击的系统中捕获了数据包，则可以从数据包捕获中提取客户端的公钥，并从内存中提取服务器的私钥。然后解密流量。

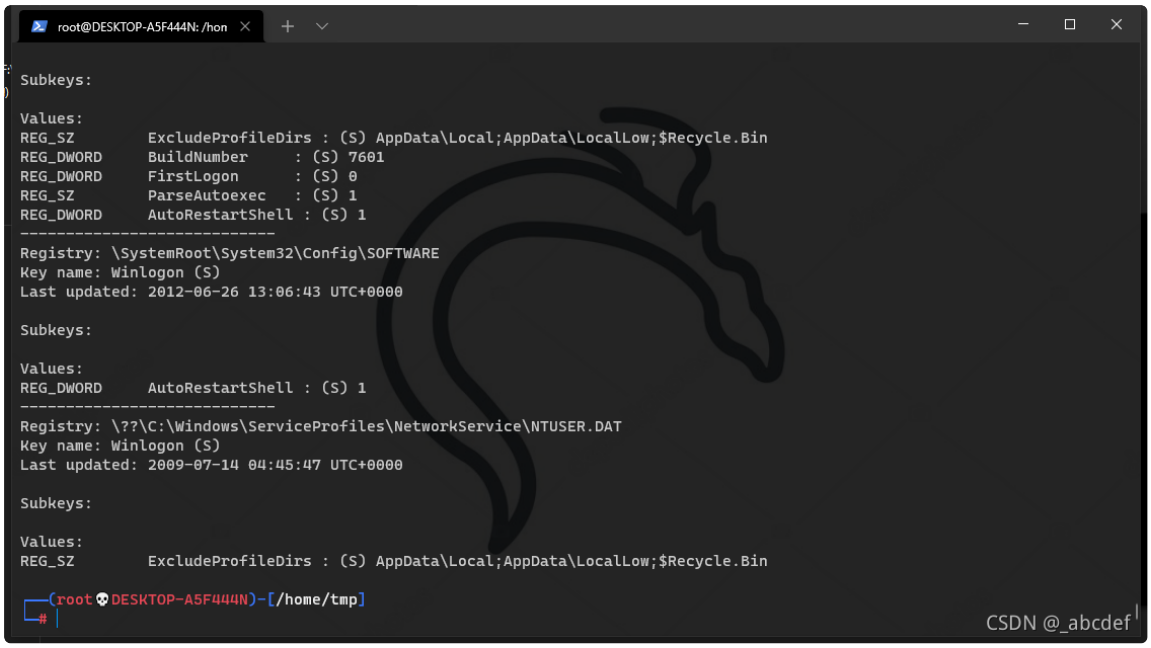
```
root@kali-linux-wp:/home/dump# volatility -f /home/13/attachment.vmem --profi
le=Win7SP1x64 lsadump
Volatility Foundation Volatility Framework 2.6
DefaultPassword
0x00000000  20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00000010  54 00 68 00 69 00 73 00 69 00 73 00 6e 00 6f 00   T.h.i.s.i.s.n.o.
0x00000020  74 00 74 00 68 00 65 00 66 00 6c 00 61 00 67 00   t.t.h.e.f.l.a.g.
0x00000030  fe 80 78 69 0a 7b fd 6b 50 eb 87 4d da ff d8 27   ..xi.{.kP..M...'

DPAPI_SYSTEM
0x00000000  2c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ,...............
0x00000010  01 00 00 00 ae d1 21 d4 9f e3 2b 26 0b bc 0d 96   ......!...+&....
0x00000020  b1 76 3f fb c5 ed a0 d7 7e a9 ab c1 d9 95 be ed   .v?.....~.......
0x00000030  0c 39 81 30 0c d0 5a a3 17 11 3b c3 00 00 00 00   .9.0..Z...;.....

root@kali-linux-wp:/home/dump#
```
https://blog.csdn.net/qq_38626043

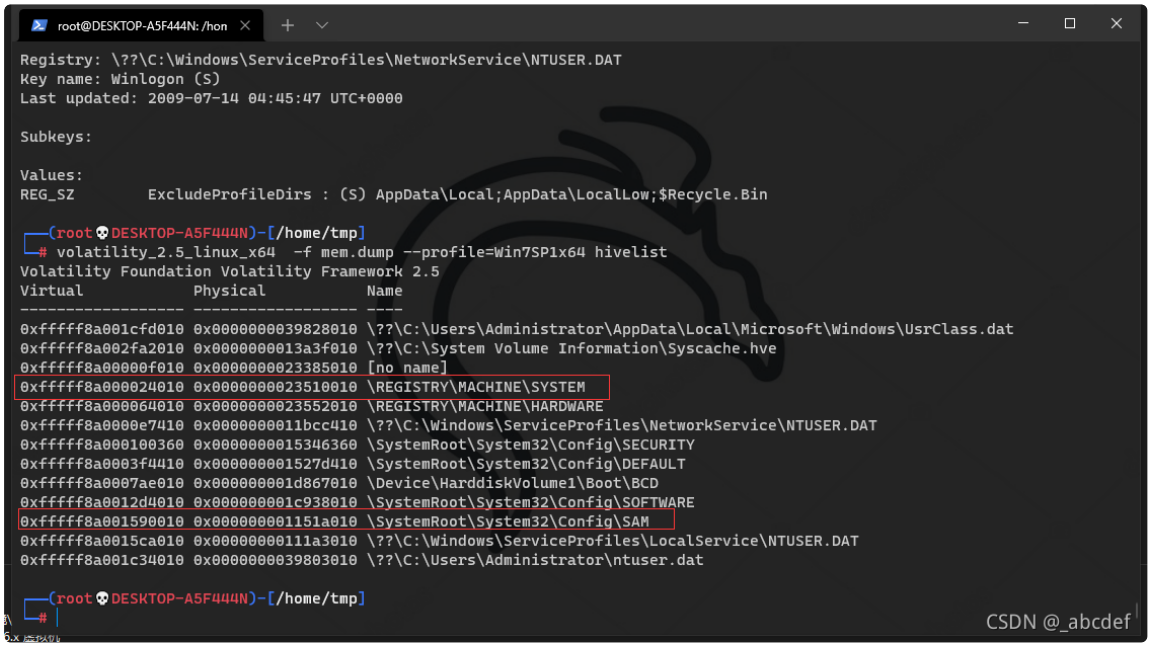
7.2 最后登录用户

```
volatility -f 1.raw  --profile=Win7SP1x86 printkey -K "SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon"
```

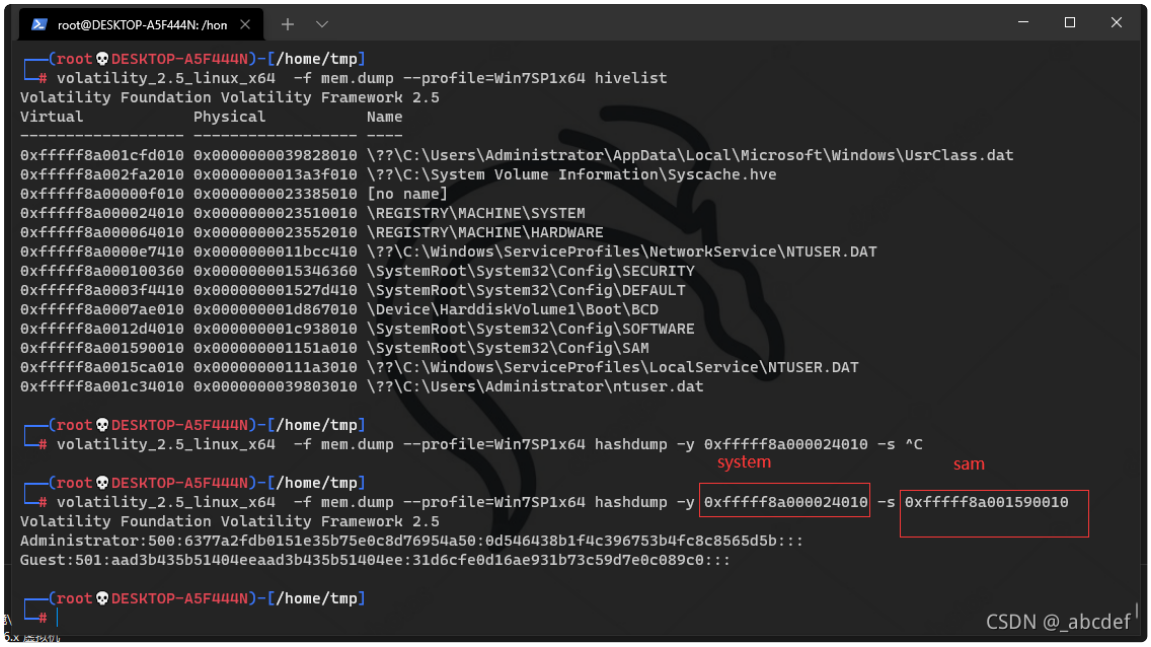


7.3 通过SAM提取密码

```
volatility_2.5_linux_x64  -f mem.dump --profile=Win7SP1x64 hivelist
```




```
volatility_2.5_linux_x64  -f mem.dump --profile=Win7SP1x64 hashdump -y (system virtual) -s (sam virtual)
```

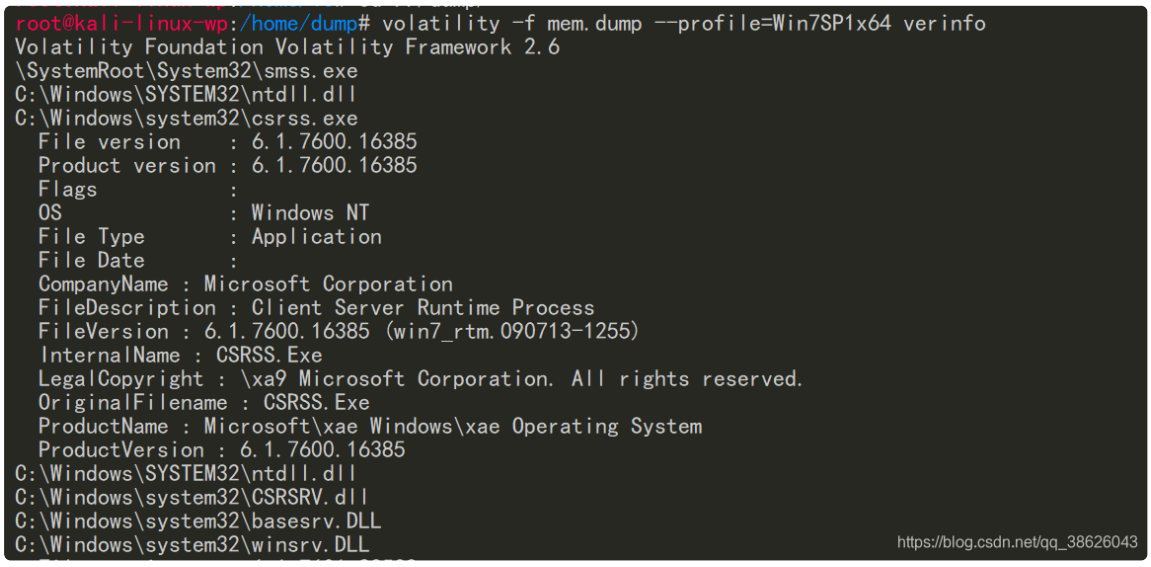


八. 系统

8.1 扫描系统信息

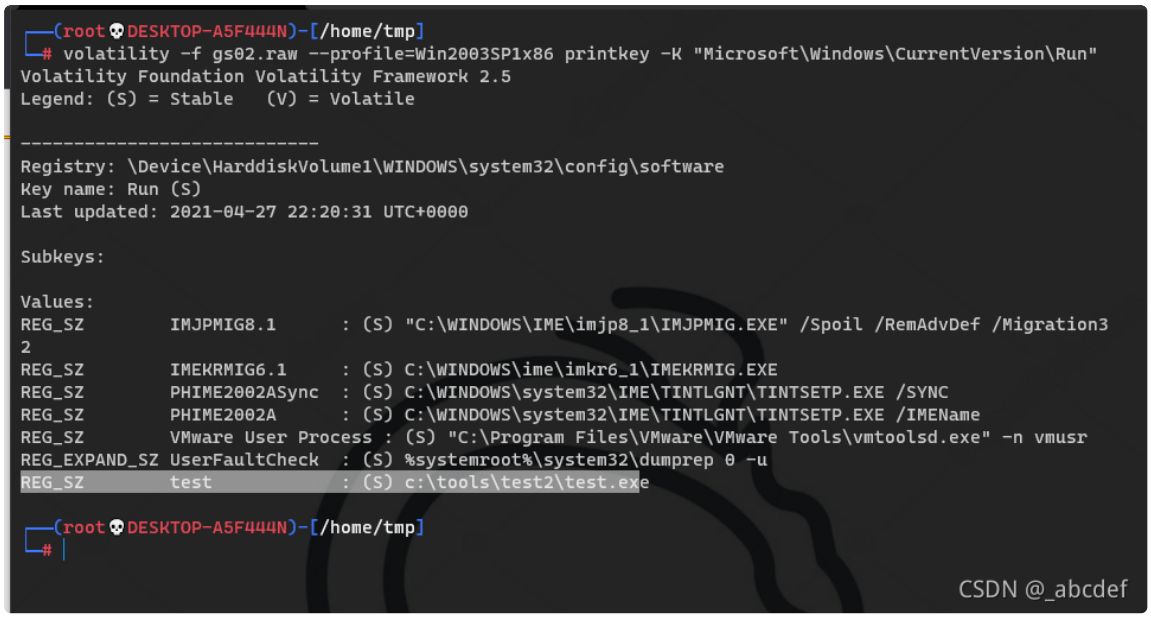
个人觉得不太准确

```
1   $ volatility -f mem.dump --profile=Win7SP1x64 verinfo
```



8.2 启动项

```
volatility -f gs02.raw --profile=Win2003SP1x86 printkey -K "Microsoft\Windows\CurrentVersion\Run"
```



8.3 隐藏进程注入

```
volatility -f gs02.raw --profile=Win2003SP1x86 malfind
```

8.4 查看内核驱动模块

```
$ volatility -f 232828.raw --profile=Win7SP1x86 modules
$ volatility -f 232828.raw --profile=Win7SP1x86 modscan
$ volatility -f 232828.raw --profile=Win7SP1x86 driverscan
```

8.5 userassist信息

userassist键值包含系统或桌面执行文件的信息，如名称、路径、执行次数、最后一次执行时间等。

```
$ volatility -f 042003.raw --profile=Win7SP1x86 userassist
```

8.6 网络连接

```
$ volatility -f name --profile=WinXPSP2x86 netscan
```

8.7 安全进程

```
$ volatility -f name --profile=Win7SP1x64 psscan
```

8.8 服务

svcscan查看

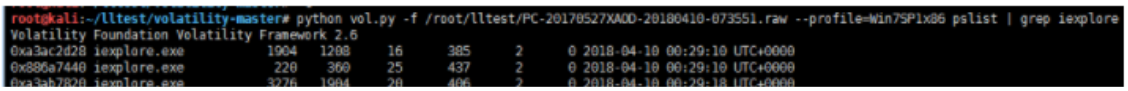
```
$ volatility -f name --profile=Win7SP1x86 svcscan
```

九. 软件 与 导出

IE

```
$ volatility -f name --profile=WinXPSP2x86 iehistory
```

查看iexplore进程pid



提取某个进程

将内存中的某个进程数据以 dmp 的格式保存出来

```
$ volatility -f name --profile=WinXPSP2x86 -p [PID] -D [dump 出的文件保存的目录]
```

获取内存中的系统密码

```
$ volatility -f name --profile=WinXPSP2x86 hashdump -y （注册表 system 的 virtual 地址 ) -s （SAM 的 virtual 地址)
$ volatility -f name --profile=WinXPSP2x86 hashdump -y 0xe1035b60 -s 0xe16aab60
```

提取某个文件

```
$ volatility -f Target.vmem --profile=Win7SP1x64 dumpfiles -Q 0x000000007feabbc0 -D ./
```

复制、剪切版信息

```
$ volatility -f name --profile=Win7SP1x64 clipboard
$ volatility -f name --profile=Win7SP1x64 dlllist -p 3820
```

其它

利用stringes 扫描 Flag字符串扫描

```
strings -e l 2616.dmp | grep flag
```