New School Forensics

Latest Tools and Techniques in Memory Analysis
CHAD TILBURY

 Former: Special Agent with US Air Force Office of Special Investigations

 Current: Incident Response and Computer Forensics Consultant

 Over 12 years in the trenches

 SANS Digital Forensics and Incident Response Instructor & Author

chad@forensicmethods.com
http://ForensicMethods.com
@chadtilbury
The Year of Memory

Forensics?

64 bit support

Volatile Registry Analysis

Memory Timelining

Mac OS X Analysis

64 bit support

Linux Analysis

Volatile Registry Analysis

Whitelisting

Mac OS X Analysis

Live Memory Analysis

64 Bit

Whitelisting
Old School vs. New School

vs.

vs.
Mac and Linux Memory Forensics
Mac Memory Reader

- Runs on Mac OS X 10.4-10.8, PowerPC, Intel, x86, x64
- Generates a Mach-O file or raw dump of memory (-P)
- Optional image hashing (-H)
- Load kernel extension to fake /dev/mem only (-k)
- Simple and effective!

```
MacBook-Pro:MacMemoryReader Chad$ sudo ./MacMemoryReader -P /Volumes/Data/mem.dmp
Dumping memory regions:
available 0000000000000000 (572.00 KB)
ACPI_NVS 000000000008f000 (4.00 KB)
available 0000000000009000 (64.00 KB)
LoaderData 0000000000100000 (60.00 KB)
available 000000000010f000 (964.00 KB)
LoaderData 0000000000200000 (34.89 MB)
```
“There are currently very few tools to analyze physical memory dumps from Mac OS X machines. **Hex editors, string extraction tools, search tools, and file carvers** are all useful for extracting data.”

-Mac Memory Reader help file
Mac
Memoryze

- Dump memory
  - `sudo macmemoryze dump -f mem.dmp`

- Analysis (just the basics):
  - `proclist`
  - `proclist -w` (similar to `lsof`)
  - `proclist -c` (carve for processes)
  - `kextlist`
  - `kextlist -c` (carve for kernel extensions)
  - Enumerate System Call Table and Mach Trap Table

- Live analysis capable (do not include `-f` option)
## Mac Memoryze Procllist

```
INFO: [+] searching for lowGlo
INFO: [+] lowGlo [00000000002A3000]
INFO: [+] found os [Mac OS X 10.6 (Snow Leopard)] [32-bit]
INFO: [+] PA PML4 [0000000000100000]

*******************************************************************************
WALK PROCESS LIST
*******************************************************************************

<table>
<thead>
<tr>
<th>PADDR</th>
<th>VADDR</th>
<th>NAME</th>
<th>PID</th>
<th>PPID</th>
<th>BITS</th>
<th>STATE</th>
<th>STARTED</th>
<th>USERNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000085D6C0 00000000085D6C0</td>
<td>kernel_task</td>
<td>0</td>
<td>0</td>
<td>32</td>
<td>RUNNABLE</td>
<td>2013-06-12 18:44:57</td>
<td>_spotlight root</td>
<td></td>
</tr>
<tr>
<td>0000000005E18D20 000000000C266D20</td>
<td>launchd</td>
<td>1</td>
<td>0</td>
<td>64</td>
<td>RUNNABLE</td>
<td>2013-06-12 18:44:57</td>
<td>root</td>
<td></td>
</tr>
<tr>
<td>0000000005E187E0 000000000C2667E0</td>
<td>kextd</td>
<td>10</td>
<td>0</td>
<td>64</td>
<td>RUNNABLE</td>
<td>2013-06-12 18:45:02</td>
<td>root</td>
<td></td>
</tr>
<tr>
<td>0000000005E182A0 000000000C2662A0</td>
<td>notifyd</td>
<td>11</td>
<td>0</td>
<td>64</td>
<td>RUNNABLE</td>
<td>2013-06-12 18:45:03</td>
<td>root</td>
<td></td>
</tr>
<tr>
<td>0000000001E8540 000000000C266540</td>
<td>ntpd</td>
<td>13</td>
<td>0</td>
<td>64</td>
<td>RUNNABLE</td>
<td>2013-06-12 18:45:07</td>
<td>root</td>
<td></td>
</tr>
<tr>
<td>0000000001E0F07E0 000000000C6F77E0</td>
<td>usbmuxd</td>
<td>17</td>
<td>0</td>
<td>64</td>
<td>RUNNABLE</td>
<td>2013-06-12 18:45:07</td>
<td>root</td>
<td></td>
</tr>
<tr>
<td>0000000001E0F0540 000000000C6F7540</td>
<td>SystemStarter</td>
<td>18</td>
<td>0</td>
<td>64</td>
<td>RUNNABLE</td>
<td>2013-06-12 18:45:07</td>
<td>root</td>
<td></td>
</tr>
<tr>
<td>0000000001E0F0000 000000000C6F7000</td>
<td>syslogd</td>
<td>20</td>
<td>0</td>
<td>64</td>
<td>RUNNABLE</td>
<td>2013-06-12 18:45:07</td>
<td>root</td>
<td></td>
</tr>
<tr>
<td>0000000001E0F6A80 000000000C710540</td>
<td>securityd</td>
<td>22</td>
<td>0</td>
<td>64</td>
<td>RUNNABLE</td>
<td>2013-06-12 18:45:07</td>
<td>root</td>
<td></td>
</tr>
</tbody>
</table>
```

<table>
<thead>
<tr>
<th>PADDR</th>
<th>VADDR</th>
<th>NAME</th>
<th>PID</th>
<th>PPID</th>
<th>BITS</th>
<th>STATE</th>
<th>STARTED</th>
<th>USERNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000000001E0F6540 000000000C710540</td>
<td>WindowServer</td>
<td>90</td>
<td>1</td>
<td>64</td>
<td>RUNNABLE</td>
<td>2013-06-12 18:45:53</td>
<td>Chad root</td>
<td></td>
</tr>
</tbody>
</table>
```
<table>
<thead>
<tr>
<th>System Information</th>
<th>Process / Module Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>mac_print_boot_cmdline</td>
<td>mac_pslst</td>
</tr>
<tr>
<td>mac_dmesg</td>
<td>mac_pstree</td>
</tr>
<tr>
<td>mac_version</td>
<td>mac_proc_maps</td>
</tr>
<tr>
<td>mac_vfs_events</td>
<td>mac_psaux</td>
</tr>
<tr>
<td>mac_machine_info</td>
<td>mac_lsmod</td>
</tr>
<tr>
<td>mac_mount</td>
<td>mac_lsof</td>
</tr>
<tr>
<td>mac_list_sessions</td>
<td>mac_dead_procs</td>
</tr>
<tr>
<td>mac_list_zones</td>
<td>mac_pgrp_hash_table</td>
</tr>
<tr>
<td>mac_ls_logins</td>
<td>mac_pid_hash_table</td>
</tr>
<tr>
<td>mac_volshell</td>
<td>mac_dump_maps</td>
</tr>
<tr>
<td></td>
<td>mac_tasks</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Malware</td>
<td>Networking</td>
</tr>
<tr>
<td>mac_trustedbsd</td>
<td>mac_ifconfig</td>
</tr>
<tr>
<td>mac_check_syscalls</td>
<td>mac_netstat</td>
</tr>
<tr>
<td>mac_check_sysctl</td>
<td>mac_route</td>
</tr>
<tr>
<td>mac_check_trap_table</td>
<td>mac_arp</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="https://code.google.com/p/volatility/wiki/MacMemoryForensics">https://code.google.com/p/volatility/wiki/MacMemoryForensics</a></td>
<td></td>
</tr>
</tbody>
</table>
Volatility + Mac

```
root@SIFT-Workstation:~/# voldev.py -f mem.0 --profile=MacSnowLeopard_10_6_8_Intelx86 mac_pslst
Volatile Systems Volatility Framework 2.3_beta
Offset   Name          Pid  Uid  Gid          Start Time
--------  --------------  -----  ---  ---------    --------------
0x0e75f2a0 image 1023  0   0             2013-06-12 21:29:04 UTC+0000
0x0c6f7a80 taskgated 1017  0   0             2013-06-12 21:29:02 UTC+0000
0x0e75f7e0 MacMemoryReader 1016  0   0             2013-06-12 21:29:02 UTC+0000
0x0c6f72a0 mdworker 999   89  89             2013-06-12 21:25:33 UTC+0000
```
Linux Memory Acquisition

Old School:
- `dd if=/dev/kmem`
- Fmem kernel module
- Redhat Crash Dump Utilities

New School

http://code.google.com/p/lime-forensics/

```
root@SIFT-Workstation:/lime/src# tar xvzf lime-forensics-1.1-r14.tar.gz
root@SIFT-Workstation:/lime/src# cd src
root@SIFT-Workstation:/lime/src# make -f Makefile
root@SIFT-Workstation:/lime/src# insmod lime.ko "path=/lime/ram.lime format=lime"
root@SIFT-Workstation:/lime/src# ll /lime
```

```
-r--r--r-- 1 root root 2146957408 2012-10-18 17:38 ram.lime
drwxrwxrwx 2 root root 4096 2012-10-18 17:54 src
```
Volatility + Linux =

System Information
- linux_dmesg
- linux_bash
- linux_cpuinfo
- linux_dentry_cache
- linux_tmpfs
- linux_find_file
- linux_memmap
- linux_mount
- linux_mount_cache
- linux_slabinfo
- linux_vma_cache
- linux_volshell

Malware
- linux_yarascan
- linux_check_syscall
- linux_check_idt
- linux_check_afinfo
- linux_check_creds
- linux_check_evt_arm
- linux_check_fop
- linux_check_tty
- linux_check_modules
- linux_keyboard_notifier

Networking
- linux_arp
- linux_ifconfig
- linux_netstat
- linux_route_cache
- linux_pkt_queues
- linux_lsof

Process / Module Info
- linux_proc_maps
- linux_dump_map
- linux_psaux
- linux_pstree
- linux_pstree_cache
- linux_pstree
- linux_psxview
- linux_pidhashtable
- linux_lsmmod
- linux_moddump
- linux_lsof

https://code.google.com/p/volatility/wiki/LinuxMemoryForensics
```
root@SIFT-Workstation:/# voldev.py -f dfrws.mem --profile=Linuxdfrws-profilex86 linux_yaraskan -Y "tar -z"

Volatile Systems Volatility Framework 2.3_beta

<table>
<thead>
<tr>
<th>Task</th>
<th>bash</th>
<th>pid</th>
<th>2585 rule r1 addr 0x8b4c500</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x8b4c500</td>
<td>74 61 72 20 2d 7a 70 78 76 66 20 78 6d 6f 64 75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x8b4c510</td>
<td>6c 65 70 61 74 68 2e 74 67 7a 00 00 d1 00 00 00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x8b4c520</td>
<td>18 22 9e 00 18 22 9e 00 b5 e5 b4 08 b3 e5 b4 08</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x8b4c530</td>
<td>b4 e5 b4 08 b5 e5 b4 08 b6 e5 b4 08 b7 e5 b4 08</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Task: bash pid 2585 rule r1 addr 0x8b59dd6
| 0x8b59dd6 | 74 61 72 20 2d 7a 70 78 76 66 20 78 6d 6f 64 75 |
| 0x8b59de6 | 6c 65 70 61 74 68 2e 74 67 7a 00 63 64 20 78 6d |
| 0x8b59df6 | 6f 64 75 6c 65 70 61 74 68 00 6c 6c 00 75 6e 73 |
| 0x8b59e06 | 65 74 20 48 49 53 54 4f 52 59 00 2e 2f 72 6f 6f |
```

```
tar.-zpxvf.xmodul lepath.tgz.. tar.-zpxvf.xmodul lepath.tgz.cd.xm odulepath.ll.uns et.HISTORY../roo
```
### Volatile Systems Volatility Framework 2.3_beta

<table>
<thead>
<tr>
<th>Pid</th>
<th>Name</th>
<th>Command Time</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>2042</td>
<td>bash</td>
<td>2011-02-06 14:04:39 UTC+0000</td>
<td>apt-get remove exim4</td>
</tr>
<tr>
<td>2042</td>
<td>bash</td>
<td>2011-02-06 14:04:39 UTC+0000</td>
<td>apt-get remove exim4-base</td>
</tr>
<tr>
<td>2042</td>
<td>bash</td>
<td>2011-02-06 14:04:39 UTC+0000</td>
<td>apt-get remove exim4-daemon-light</td>
</tr>
<tr>
<td>2042</td>
<td>bash</td>
<td>2011-02-06 14:04:39 UTC+0000</td>
<td>dpkg -l</td>
</tr>
<tr>
<td>2042</td>
<td>bash</td>
<td>2011-02-06 14:04:39 UTC+0000</td>
<td>apt-get remove exim4-config</td>
</tr>
<tr>
<td>2042</td>
<td>bash</td>
<td>2011-02-06 14:04:39 UTC+0000</td>
<td>ls -a</td>
</tr>
<tr>
<td>2042</td>
<td>bash</td>
<td>2011-02-06 14:04:39 UTC+0000</td>
<td>dpkg --purge</td>
</tr>
<tr>
<td>2042</td>
<td>bash</td>
<td>2011-02-06 14:04:39 UTC+0000</td>
<td>pwd</td>
</tr>
<tr>
<td>2042</td>
<td>bash</td>
<td>2011-02-06 14:04:39 UTC+0000</td>
<td>apt-get remove exim</td>
</tr>
<tr>
<td>2042</td>
<td>bash</td>
<td>2011-02-06 14:04:39 UTC+0000</td>
<td>dpkg -l</td>
</tr>
<tr>
<td>2042</td>
<td>bash</td>
<td>2011-02-06 14:04:39 UTC+0000</td>
<td>mkdir exim4</td>
</tr>
<tr>
<td>2042</td>
<td>bash</td>
<td>2011-02-06 14:04:39 UTC+0000</td>
<td>dpkg -i exim4-config_4.69-9_all.deb</td>
</tr>
<tr>
<td>2042</td>
<td>bash</td>
<td>2011-02-06 14:04:39 UTC+0000</td>
<td>cd exim4/</td>
</tr>
<tr>
<td>2042</td>
<td>bash</td>
<td>2011-02-06 14:04:39 UTC+0000</td>
<td>scp yom@192.168.56.1:/home/yom/temporary/exmi4/* .</td>
</tr>
<tr>
<td>2042</td>
<td>bash</td>
<td>2011-02-06 14:04:39 UTC+0000</td>
<td>dpkg -i exim4-base 4.69-9 i386.deb</td>
</tr>
</tbody>
</table>
What is Timeliner?

- Set of Volatility plugins to collect time information from memory artifacts
- Many memory artifacts have embedded timestamps:
  - Processes
  - Threads
  - Portable Executable Files
    - Process EXEs, DLLs, and Drivers
  - Network Sockets
  - Registry Keys
  - Event Logs
- **Timeliner** consolidates artifacts into a delimited file that can be easily converted to a timeline
  - Volatility 2.3 now capable of body file format!
  - David Nides submitted recent patch for Log2Timeline format
Memory Timelining
timeliner

**Purpose**
- Timeliner collects timestamps from memory artifacts and outputs them in a timeline format

**Important Parameters**
- Send output to a delimited file (--output-file=\textit{file\_name}) v2.1
- Create output in body file format (--output=body) v2.3
- Log2Timeline output format (pending) v2.4??

**Investigative Notes**
- Compatible with XP and Win7: automatically adjusts helper plugins
- Output can voluminous; best practice is to use “--output-file”
- The output is not currently compatible with other timeline formats
- Timeliner can take \textit{hours} to run – be patient!
- The “-h” help information currently lists many incorrect options
## Example Output: Timeliner Processes

<table>
<thead>
<tr>
<th>Column</th>
<th>Header</th>
<th>Column</th>
<th>Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Creation Time</td>
<td>5</td>
<td>Parent Process ID</td>
</tr>
<tr>
<td>2</td>
<td>Artifact Type (PROCESS)</td>
<td>6</td>
<td>Exit Time</td>
</tr>
<tr>
<td>3</td>
<td>Process Name</td>
<td>7</td>
<td>EPROCESS Offset</td>
</tr>
<tr>
<td>4</td>
<td>Process ID</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

|            |                             |        |                               |
| 2009-12-11 | 19:04:40 | [PROCESS]| svchost.exe | 1456 | 628 | 0x7e1e3bf8 | |
| 2009-12-11 | 19:21:36 | [PROCESS]| wuauc1t.exe | 3420 | 1296 | 0x7ec408f8 | |
| 2009-12-11 | 19:21:26 | [PROCESS]| Eraser.exe   | 596  | 2840 | 0x7ecd49c0 | |
| 2009-12-11 | 19:21:23 | [PROCESS]| avgtray.exe  | 2792 | 2840 | 0x7ecd8330 | |
| 2009-12-11 | 19:21:23 | [PROCESS]| sidebar.exe  | 2932 | 2840 | 0x7ed3580  | |
| 2009-12-11 | 19:21:23 | [PROCESS]| jusched.exe  | 1120 | 2840 | 0x7ed39020 | |
| 2009-12-11 | 19:52:24 | [PROCESS]| mdd_1.3.exe  | 3412 | 4012 | 0x7f143ce8 | |
| 2009-12-11 | 19:21:21 | [PROCESS]| dwm.exe      | 156  | 1284 | 0x7f38e020 | |
| 2009-12-11 | 19:21:21 | [PROCESS]| taskeng.exe  | 2588 | 1296 | 0x7f3b0d90 | |
| 2009-12-11 | 19:21:28 | [PROCESS]| BRNIPMON.exe | 3136 | 1752 | 0x7f39300  | |
| 2009-12-11 | 19:51:25 | [PROCESS]| cmd.exe      | 4012 | 2840 | 0x7f3d9598 | |
Timeliner Example

12/10/2009 18:49 [USER ASSIST] \Users\terry\NTUSER.DAT UEME_RUNPIDL:%csidl23%\Eraser\Eraser.Ink
12/10/2009 18:49 [USER ASSIST] \Users\terry\NTUSER.DAT UEME_RUNPATH:C:\Program Files\Eraser\Eraser.exe
12/10/2009 18:53 [REGISTRY] \Users\terry\NTUSER.DAT \Software\Microsoft\Windows\Mail\Columns
12/10/2009 18:56 [REGISTRY] \Users\terry\NTUSER.DAT \Software\Microsoft\Windows\CurrentVersion\Run

root@SIFT-Workstation: # vol.py -f mem.dmp --profile=VistaSP2x86 printkey
-K "Software\Microsoft\Windows\CurrentVersion\Run"

Volatile Systems Volatility Framework 2.2
Legend: (S) = Stable (V) = Volatile
-----------------------------
Registry: \Device\HarddiskVolume1\Users\terry\NTUSER.DAT
Key name: Run (S)
Last updated: 2009-12-10 18:56:07

Subkeys:

Values:
REG_SZ   Sidebar : (S) C:\Program Files\Windows Sidebar\sidebar.exe /autoRun
REG_SZ   Google Update : (S) "C:\Users\terry\AppData\Local\Google\Update\GoogleUpdate.exe" /c
REG_SZ   Eraser : (S) C:\Program Files\Eraser\Eraser.exe -hide

C:\Program Files\Eraser\Eraser.exe -hide
Live Response & Live Memory Analysis
echo Starting 12 of 28 - psloggedon
echo.>> "%RESULTS_FILE%"
echo ***** psloggedon *****>> "%RESULTS_FILE%"
"%IRSHARED%\sysinternals\t_psloggedon" -accepteula>> "%RESULTS_FILE%"

echo Starting 13 of 28 - handle
echo.>> "%RESULTS_FILE%"
echo ***** handles *****>> "%RESULTS_FILE%"
"%IRSHARED%\sysinternals\t_handle" -
Mandiant Redline
“Collector”

**Create a Standard Collector**

- **Review Script Configuration**
  - You have chosen to create a Standard Collector
  - The standard collector will collect the data necessary to perform a Redline that Redline does not directly visualize, which are recommended for matching.
  - **Edit your script**
  - **Acquire Memory Image**
    - Checking this option will acquire an image of memory that can be used for analysis in Redline.

- **Specify Collector Location**
  - Save Your Collector To: E:\RedlineCollector

**View and Edit Your Script**

- **Memory**
  - Process Listing
    - Handles
    - Sections
    - Imports
    - Exports
    - Md5
    - Sha256
  - Drivers Enumeration
    - Imports
    - Exports
    - Md5
    - Sha256
  - Hook Detection
    - IDT
    - SSDT Inline
    - Verify Digital Signatures
  - Acquire Memory Image
    - Acquire an image of memory that can be used to accurately acquire process memory and drivers during analysis in Redline.
Redline Portable Collector
Live Memory Analysis
Who Cares?

- **Digital Signature Checks**
  - Digital signatures stripped when loaded into memory
  - Verification done using file certificates stored on-disk

- **MD5 Whitelisting**
  - MD5 hashes of on-disk copies of memory mapped files
  - Must have access to file system

- **MemD5 Whitelisting**
  - Hashing of in memory copy of binaries
  - Requires access to Page File
Narrowing Your Focus with Live Analysis

<table>
<thead>
<tr>
<th>Trust Status</th>
<th>MD5</th>
<th>MemD5</th>
<th>Count</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digitally signed and verified</td>
<td>✔</td>
<td>✔</td>
<td>9</td>
<td>C:\WINDOWS\system32\apphelp.dll</td>
</tr>
<tr>
<td>Digitally signed and verified</td>
<td>✔</td>
<td>✔</td>
<td>31</td>
<td>C:\WINDOWS\system32\comctl32.dll</td>
</tr>
<tr>
<td>Digitally signed and verified</td>
<td>✔</td>
<td>✔</td>
<td>8</td>
<td>C:\WINDOWS\system32\cryptdll.dll</td>
</tr>
<tr>
<td>Digitally signed and verified</td>
<td></td>
<td></td>
<td>40</td>
<td>C:\WINDOWS\system32\ctype.nls</td>
</tr>
<tr>
<td>Not signed and verified</td>
<td></td>
<td></td>
<td>1</td>
<td>C:\WINDOWS\system32\dllhost\svchost.exe</td>
</tr>
<tr>
<td>Digitally signed and verified</td>
<td>✔</td>
<td>✔</td>
<td>40</td>
<td>C:\WINDOWS\system32\gdi32.dll</td>
</tr>
<tr>
<td>Digitally signed and verified</td>
<td>✔</td>
<td>✔</td>
<td>16</td>
<td>C:\WINDOWS\system32\hnetcfg.dll</td>
</tr>
<tr>
<td>Digitally signed and verified</td>
<td>✔</td>
<td>✔</td>
<td>13</td>
<td>C:\WINDOWS\system32\iervtutil.dll</td>
</tr>
<tr>
<td>Digitally signed and verified</td>
<td>✔</td>
<td>✔</td>
<td>18</td>
<td>C:\WINDOWS\system32\iphlpapi.dll</td>
</tr>
<tr>
<td>Digitally signed and verified</td>
<td>✔</td>
<td>✔</td>
<td>40</td>
<td>C:\WINDOWS\system32\kernel32.dll</td>
</tr>
</tbody>
</table>

Hide Whitelisted Items

205 Items
### Whitelist Filtering

#### Whitelist Management

<table>
<thead>
<tr>
<th>Trust Status</th>
<th>MD5</th>
<th>MemD5</th>
<th>Count</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digitally signed and verified</td>
<td>1</td>
<td></td>
<td></td>
<td>C:\WINDOWS\system32\unimdm.tsp</td>
</tr>
<tr>
<td>Digitally signed and verified</td>
<td>1</td>
<td></td>
<td></td>
<td>C:\WINDOWS\system32\ndptsp.tsp</td>
</tr>
<tr>
<td>Digitally signed and verified</td>
<td>1</td>
<td></td>
<td></td>
<td>C:\WINDOWS\system32\kmdssp.tsp</td>
</tr>
<tr>
<td>Digitally signed and verified</td>
<td>1</td>
<td></td>
<td></td>
<td>C:\WINDOWS\system32\ipconf.tsp</td>
</tr>
<tr>
<td>Digitally signed and verified</td>
<td>8</td>
<td></td>
<td></td>
<td>C:\WINDOWS\system32\winspool.drv</td>
</tr>
</tbody>
</table>

**169 vs. 12 Items**

Include Whitelisted Items
Live Memory Analysis with Volatility

- **Winpmem**
  - Raw, crash dump, and output to stdout
  - Direct analysis of running kernel (-l switch)
  - Optional write support!
  - [https://code.google.com/p/volatility/downloads/list](https://code.google.com/p/volatility/downloads/list)

- **Volatility Technology Preview Branch**
  - [https://code.google.com/p/volatility/wiki/TechPreviewBranch](https://code.google.com/p/volatility/wiki/TechPreviewBranch)
  - Includes interactive shell (similar to volshell) -> the future of Volatility?
Live Analysis with **winpmem**

```plaintext
C:\Python27>\Volatility\winpmem_1.4.exe -l
Driver Unloaded.
Loaded Driver C:\Users\SANSFO\~2\AppData\Local\Temp\pmeFD2F.tmp.
Setting acquisition mode to 1
3 memory ranges:
Start 0x0000185000 - Length 0x0009E000
Start 0x00100000 - Length 0x7FDE0000
Start 0x7FF00000 - Length 0x00100000

C:\Python27>python \vol-tech-preview\vol.py -f \.\\pmem -p Win7SP1x86_psscan

<table>
<thead>
<tr>
<th>Offset</th>
<th>Offset(V)</th>
<th>Name</th>
<th>PID</th>
<th>PPID</th>
<th>PDB</th>
<th>Stat</th>
<th>Time created</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x7d4274d0</td>
<td>17:04:23+0000</td>
<td>NisSrv.exe</td>
<td>2364</td>
<td>556</td>
<td>0x7ecc55e0</td>
<td></td>
<td>2013-04-30</td>
</tr>
<tr>
<td>0x7d42b5e0</td>
<td>17:04:26+0000</td>
<td>TPAutoConnSvc.</td>
<td>2504</td>
<td>556</td>
<td>0x7ecc5600</td>
<td></td>
<td>2013-04-30</td>
</tr>
<tr>
<td>0x7d4319c8</td>
<td>17:04:27+0000</td>
<td>svchost.exe</td>
<td>2584</td>
<td>556</td>
<td>0x7ecc5620</td>
<td></td>
<td>2013-04-30</td>
</tr>
<tr>
<td>0x7d492298</td>
<td>17:04:33+0000</td>
<td>mstdc.exe</td>
<td>2836</td>
<td>556</td>
<td>0x7ecc5660</td>
<td></td>
<td>2013-04-30</td>
</tr>
<tr>
<td>0x7d470030</td>
<td>17:06:26+0000</td>
<td>SearchIndexer.</td>
<td>3396</td>
<td>556</td>
<td>0x7ecc5520</td>
<td></td>
<td>2013-04-30</td>
</tr>
<tr>
<td>0x7d85a5d8</td>
<td>17:03:52+0000</td>
<td>sesvc.exe</td>
<td>1772</td>
<td>556</td>
<td>0x7ecc5360</td>
<td></td>
<td>2013-04-30</td>
</tr>
<tr>
<td>0x7d876d40</td>
<td>17:03:53+0000</td>
<td>postgres.exe</td>
<td>1820</td>
<td>1728</td>
<td>0x7ecc53c0</td>
<td></td>
<td>2013-04-30</td>
</tr>
<tr>
<td>0x7d87e640</td>
<td>17:03:53+0000</td>
<td>conhost.exe</td>
<td>1832</td>
<td>396</td>
<td>0x7ecc53a0</td>
<td></td>
<td>2013-04-30</td>
</tr>
<tr>
<td>0x7d8d5718</td>
<td>17:03:56+0000</td>
<td>svchost.exe</td>
<td>1888</td>
<td>556</td>
<td>0x7ecc5380</td>
<td></td>
<td>2013-04-30</td>
</tr>
</tbody>
</table>
```
Good Day or Bad Day?
Old School → doskey

C:\> doskey /history
csvde -f out.csv
gsecdump -s
doskey /history
Memory Carving

Extracting Windows command line details from physical memory

Richard M. Stevens a,*, Eoghan Casey b

a Information Security Institute, Johns Hopkins University, Baltimore, MD 21218, USA
b cmdLabs, Baltimore, MD 21218, USA
Typed Commands: cmdscan & consoles

Purpose
- Scan csrss.exe (XP) and conhost.exe (Win7) for Command_History and Console_Information residue

Important Parameters
- None

Investigative Notes
- Gathering command history and console output can give insight into user / attacker activities
- `cmdscan` provides information from the command history buffer
- `consoles` prints commands (inputs) + screen buffer (outputs)
- Plugins can identify data from active and closed sessions
Typed Commands: cmdscan & consoles

CommandProcess: csrss.exe Pid: 600
CommandHistory: 0x1460a98 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 15 LastAdded: 14 LastDisplayed: 14
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x488
Cmd #0 @ 0x4e2f18: cd Desktop
Cmd #1 @ 0x1458fa8: duser.exe
Cmd #2 @ 0x1447b58: duser.exe -i 832 -c
Cmd #3 @ 0x1460a50: duser.exe -i 832 -c "cmd.exe"

Screen 0x1454d48 X:80 Y:300
Dump:
C:\Documents and Settings\qauser>cd Desktop
C:\Documents and Settings\qauser\Desktop>duser.exe Read usage
C:\Documents and Settings\qauser\Desktop>duser.exe -i 832 -c duser.exe: option requires an argument -- c
Read usage
C:\Documents and Settings\qauser\Desktop>duser.exe -i 832 -c "cmd.exe"
Code is injected
Old School → pclip

Find pcclip.exe at http://unxutils.sourceforge.net/ (or just get infected with Zeus)
Clipboard Contents:

Purpose

- Extract contents of windows clipboard

Important Parameters

- Verbose mode (-v) shows hex view of data (necessary if binary data stored in clipboard)

Investigative Notes

- Recovers clipboard data for each Windows Station (i.e. console, RDP, Fast User Switching, etc.)
- Works on both XP/2003 and Windows 7/2008 systems
- In some cases, the clipboard only holds a pointer to the "clipped" content – (i.e. the full path for a copied file)
Clipboard Contents:
clip\(\text{board}\)
Additional References

- http://volatility-labs.blogspot.com/2013/05/movp-ii-23-creating-timelines-with.html
- https://www.mandiant.com/blog/unibody-memory-analysis-introducing-mac-memoryze/
- http://volatility-labs.blogspot.com/2013/05/movp-ii-32-linuxandroid-memory.html
- http://holisticinfosec.blogspot.com/2013/03/toolsmith-redline-apt1-and-you-were-all.html
- DFIROnline Memory Forensics with Michael Cohen:
  http://www.youtube.com/watch?v=9aC7yIYwvAY
Thank You!

chad@forensicmethods.com
computer-forensics.sans.org/blog
ForensicMethods.com
@chadtilbury
Digital Forensics and Incident Response

Website:
http://computer-forensics.sans.org

Blogs:
http://computer-forensics.sans.org/blog

SIFT Workstation:
http://computer-forensics.sans.org/community/downloads

Digital Forensics Challenge:
http://computer-forensics.sans.org/challenges

Twitter:
www.twitter.com/sansforensics

FOR408
Computer Forensic Investigations – Windows In-Depth
GCFE

FOR508
Advanced Computer Forensic Analysis & Incident Response
GCFA

FOR558
Network Forensics

FOR610
REM: Malware Analysis Tools & Techniques
GREM

FOR563
Mobile Device Forensics

FOR526
Windows Memory Forensics In-Depth