RSACONFERENCE2014

FEBRUARY 24 – 28 | MOSCONE CENTER | SAN FRANCISCO

Share.
Learn.
Secure.

Capitalizing on
Collective Intelligence

Hunting Mac Malware with Memory Forensics

SESSION ID: HTA-F01

Andrew Case

Volatility
@attrc

Purpose of the Talk

- Show how real rootkits affect system security and stability
- Demonstrate how rootkits can be found with memory forensics
- Utilize the open source Volatility framework for deep analysis of system state

Agenda

- Why memory forensics?
- Introduction to Volatility
- Showcase Mac memory analysis capabilities
- Detect Mac kernel rootkit techniques with memory forensics

Why Memory Forensics?

- Memory forensics analyzes the entire operating system state
  - Processes
  - Network Data
  - Loaded kernel modules
  - Running processes
  - Much more..
- Nearly all of this information in memory is *never* written to disk

Why Memory Forensics? Cont.

- Advanced malware operates only in memory
  - Meterperter / CANVAS / Core Impact
  - Custom tools by real attackers
- “Pull the plug” and your best evidence disappears!

Volatility

- Open source memory analysis framework written in Python
- Provides an architecture and plugins for deep analysis of data structures in memory
- Contains many features not available in any other memory forensics tools
- One of the most used tools in forensics

Supported OSes

- Windows
  - XP through 7, including server operating systems
  - 32 & 64 bit
- Linux / Android
  - 2.6.11 through 3.x
- Mac

Supported Memory Capture Formats

- All
  - raw (dd), Encase (EWF), VMWare, Virtualbox
- Windows
  - crash dumps, hibernation files, Hpak
- Linux
  - LiME

Mac Memory Analysis

Acquisition

- Mac Memory Reader (ATC-NY)
  - Saves files to Macho-o format
  - Works from 10.5.x to 10.8.x, broken on 10.9
- OSXPmem (Michael Cohen)
  - Works on 10.9
- Mac Memoryze (Mandiant)
- 10.7+ guests in VMware Fusion
  - Fully supported by Apple

Previous Efforts before Volatility Support

- Matthieu Suiche - Mac OS X Physical Memory Analysis [1]
  - Finding page tables, processes, mounted file systems, and system call table
- Volafox
  - First real plugin based OS X analysis
  - Around 7 plugins for analysis
  - Brittle support for new versions and difficult to add

Volatility & Mac Memory Forensics

- 2.3 is the first official release with Mac support
- Has been in SVN for quite some time
  - 10.7.x support since summer 2012
  - Full support since early 2013
    - Many more OS versions supported
    - New plugins
    - Bug fixes

Supported Operating System Versions

- 32-bit 10.5.x Leopard (no 64 bit version)
- 32-bit & 64-bit 10.6.x Snow Leopard
- 32-bit & 64-bit 10.7.x Lion
- 64-bit 10.8.x Mountain Lion (no 32-bit version)
- 64-bit 10.9.x (no 32-bit version)

Process Enumeration

- mac_pslist*
  - Often hits an endless loop due to acquisition issues, plugin checks for the condition and bails
- mac_tasks
- mac_psaux
  - Command line arguments from userland
- mac_pstree
  - Parent/child relationship

mac_pslist

```
$ python vol.py --profile=MacMountainLion_10_8_3_AMDx64 -f 10.8.3.mmr.macho mac_pslist
Volatile Systems Volatility Framework 2.3
Offset              Name                 Pid      Uid      Gid      PGID     Bits         DTB                  Start Time
------------------  -------------------- -------- -------- -------- -------- ------------ ------------------   ----------
0xffffff8032be4ea0  image                4175     0        0        4167     64BIT        0x0000000317e7e000   2013-03-29 12:16:20
0xffffff803dfdea40  coresymbolicatio     4173     0        0        4173     64BIT        0x00000004114c0000   2013-03-29 12:16:18
0xffffff8032498d20  MacMemoryReader      4168     0        0        4167     64BIT        0x00000003f94a8000   2013-03-29 12:16:17
0xffffff803dfe0020  sudo                 4167     0        20       4167     64BIT        0x0000000414a34000   2013-03-29 12:16:15
0xffffff803dfe1a60  mdworker             4164     89       89       4164     64BIT        0x00000003f70cf000   2013-03-29 12:15:32
0xffffff80370af760  DashboardClient      4160     501      20       275      64BIT        0x00000003e5bd9000   2013-03-29 12:14:36
0xffffff803634ba60  CVMCompiler          4127     501      20       4127     64BIT        0x000000016692b000   2013-03-29 12:10:58
[snip]
```

mac_psaux

```
$ python vol.py --profile=MacMountainLion_10_8_3_AMDx64 -f 10.8.3.mmr.macho mac_psaux
Volatile Systems Volatility Framework 2.3
Pid      Name                     Bits      Stack                          Length    Argc   Arguments
-------- --------------------     --------- ------------------------------ --------  ------  ---------
       0 kernel_task              64BIT   0x0000000000000000   0          0
[snip]
      40 mDNSResponder     64BIT   0x00007fff54403000      384       2       /usr/sbin/mDNSResponder -launchd
      41 networkd                 64BIT   0x00007fff50d3f000       360       1       /usr/libexec/networkd
      60 usbmuxd                  64BIT   0x00007fff5fc00000       504       2
/System/Library/PrivateFrameworks/MobileDevice.framework/Versions/A/Resources/usbmuxd -launchd
      66 revisiond                64BIT    0x00007fff50d1b000      376       1
/System/Library/PrivateFrameworks/GenerationalStorage.framework/Versions/A/Support/revisiond
      72 mds                      64BIT   0x00007fff5713e000      376       1
/System/Library/Frameworks/CoreServices.framework/Frameworks/Metadata.framework/Support/mds
      75 loginwindow              64BIT  0x00007fff59635000       328       2
/System/Library/CoreServices/loginwindow.app/Contents/MacOS/loginwindow console
      77 KernelEventAgent    64BIT  0x00007fff5e5b7000      232       1          /usr/sbin/KernelEventAgent
      78 kdc                      64BIT  0x00007fff54a32000      304       1
/System/Library/PrivateFrameworks/Heimdal.framework/Helpers/kdc
      91 autofsd                  64BIT   0x00007fff577d9000      208       1       /usr/libexec/autofsd autofsd
      95 ntpd                     64BIT   0x00007fff5a494000      296       9      /usr/sbin/ntpd -c /private/etc/ntp-restrict.conf -n -g -p
/var/run/ntpd.pid -f /var/db/ntp.drift
[snip]
```

mac_pstree

```
$ python vol.py --profile=MacMountainLion_10_8_3_AMDx64 -f 10.8.3.mmr.macho mac_pstree
Volatile Systems Volatility Framework 2.3
Name                  Pid       Uid
kernel_task           0         0
.launchd              1         0
..coresymbolicatio    4173      0
..taskgated           4122      0
..ocspd               973       0
..launchd             561       89
...mdworker           4164      89
..VDCAssistant        558       0
..Dropbox             518       501
...dbfseventsd        545       0
....dbfseventsd       546       0
.....dbfseventsd      552       501
.....dbfseventsd      549       501
```

Process Memory

```
$ python vol.py --profile=MacMountainLion_10_8_3_AMDx64 -f 10.8.3.mmr.macho mac_proc_maps -p 1
Volatile Systems Volatility Framework 2.3
Pid   Name             Start                          End                            Perms     Map Name
----- ---------------- ------------------------------ ------------------------------ --------- ----------------
1     launchd          0x000000010630c000  0x0000000106333000 r-x      Macintosh HD/sbin/launchd
1     launchd          0x0000000106333000  0x0000000106335000 rw-      Macintosh HD/sbin/launchd
1     launchd          0x0000000106335000  0x000000010633b000 r--      Macintosh HD/sbin/launchd
[snip]
```

Opened File Handles

```
$ python vol.py --profile=MacMountainLion_10_8_1_AMDx64 -f 10.8.1.macho mac_lsof
Volatile Systems Volatility Framework 2.3
Pid     File Descriptor  File Path
-----   ---------------- ------------------------
1                      0    /Macintosh HD/dev/null
1                      1    /Macintosh HD/dev/null
1                      2   /Macintosh HD/dev/null
1                      4   /Macintosh HD/dev/console
1                      81 /Macintosh HD/dev/autofs_nowait
[snip]
1031                  19  /Macintosh HD/Users/vol/Desktop/volatility/volatility/plugins/mac/pstasks.py
1031                  20 /Macintosh HD/Users/vol/Desktop/volatility/volatility/plugins/mac/pstree.py
1031                  21 /Macintosh HD/Users/vol/Desktop/volatility/volatility/plugins/mac/pgrp_hash_table.py
1031                  22 /Macintosh HD/Users/vol/Desktop/volatility/volatility/plugins/mac/pslist.py
```

Networking Information

- mac_ifconfig
  - Lists information on active network devices
- mac_netstat
  - Similar to netstat on a running system

mac_netstat

```
$ python vol.py --profile=MacMountainLion_10_8_3_AMDx64 -f 10.8.3.mmr.macho mac_netstat
Volatile Systems Volatility Framework 2.3
UNIX /var/tmp/launchd/sock
UNIX /var/tmp/com.barebones.authd.socket
UNIX /var/run/com.apple.ActivityMonitor.socket
TCP :::548 :::0 TIME_WAIT
TCP 0.0.0.0:548 0.0.0.0:0 TIME_WAIT
UDP 127.0.0.1:60762 0.0.0.0:0
UNIX /var/run/mDNSResponder
UNIX /var/rpc/ncacn_np/lsarpc
UNIX /var/rpc/ncalrpc/lsarpc
TCP 10.0.1.3:49179 173.194.76.125:5222 TIME_WAIT
TCP 10.0.1.3:49188 205.188.248.150:443 TIME_WAIT
TCP 10.0.1.3:49189 205.188.254.208:443 TIME_WAIT
TCP 10.0.1.3:50614 205.188.13.76:443 TIME_WAIT
UDP 0.0.0.0:137 0.0.0.0:0
UDP 0.0.0.0:138 0.0.0.0:0
[snip]
```

Routing Table & Arp Cache

- For each entry:
  - Src/Dest
  - # of packet sent/recv
  - Time route was created
  - Interface

mac_arp

```
$ python vol.py --profile=MacMountainLion_10_8_3_AMDx64 -f ~/10.8.3.mmr.macho mac_arp
Volatile Systems Volatility Framework 2.3
Source IP          Dest. IP           Name       Sent     Recv      Time                            Exp.     Delta
------------------ ------------------ ---------- -------- --------- ------------------------------- ------   -------
192.168.228.255    ff:ff:ff:ff:ff:ff  vmnet8     10       0         2013-03-29 12:13:59 UTC+0000    39913    0
172.16.244.255     ff:ff:ff:ff:ff:ff  vmnet1     10       0         2013-03-29 12:13:59 UTC+0000    39913    0
10.0.1.255         ff:ff:ff:ff:ff:ff  en1        12       0         2013-03-29 12:13:59 UTC+0000    39913    0
10.0.1.8           e8:8d:28:cb:67:07  en1        19       924       2013-03-29 11:56:30 UTC+0000    40065    1201
10.0.1.2           ac:16:2d:32:fc:d7  en1        1        47        2013-03-29 11:56:02 UTC+0000    40037    1201
10.0.1.1           00:26:bb:6c:8e:64  en1        4551     4517      2013-03-29 01:08:53 UTC+0000    40318    40310
```

Kernel Data

Loaded Kernel Modules

```
$ python vol.py --profile=MacMountainLion_10_8_3_AMDx64 -f 10.8.3.mmr.macho mac_lsmod
Volatile Systems Volatility Framework 2.3
Address              Size     Refs  Version     Name
------------------   ---------  -----  ---------    ------------------
0xffffff7f91847000   0x3000    0    3.0.2        com.atc-nycorp.devmem.kext
0xffffff7f91841000   0x6000    0    10.1.24      com.vmware.kext.vmioplug.10.1.24
0xffffff7f91834000   0xd000    0    0104.03.86   com.vmware.kext.vmx86
0xffffff7f9182a000   0xa000    0    0104.03.86   com.vmware.kext.vmnet
0xffffff7f9181a000   0x10000   0    90.4.23      com.vmware.kext.vsockets
0xffffff7f91808000   0x12000   1    90.4.18      com.vmware.kext.vmci
0xffffff7f916d2000   0xe000    0    75.19        com.apple.driver.AppleBluetoothMultitouch
```

Mounted Filesystems

```
$ python vol.py --profile=MacMountainLion_10_8_3_AMDx64 -f 10.8.3.mmr.macho mac_mount
Volatile Systems Volatility Framework 2.3
Device                 Mount Point                          Type
---------------------- ------------------------------------ ------
/                      /dev/disk3                           hfs
/dev                   devfs                                devfs
/net                   map -hosts                           autofs
/home                  map auto_home                        autofs
/Volumes/LaCie         /dev/disk2s2                         hfs
```

Kernel Debug Buffer

```
$ python vol.py --profile=MacMountainLion_10_8_3_AMDx64 -f 10.8.3.mmr.macho mac_dmesg
Volatile Systems Volatility Framework 2.3
deny mach-lookup com.apple.coresymbolicationd
MacAuthEvent en1   Auth result for: 00:26:bb:77:d2:a7  MAC AUTH succeeded
wlEvent: en1 en1 Link UP virtIf = 0
AirPort: RSN handshake complete on en1
wl0: Roamed or switched channel, reason #8, bssid 00:26:bb:77:d2:a7
en1: BSSID changed to 00:26:bb:77:d2:a7
en1::IO80211Interface::postMessage bssid changed
MacAuthEvent en1   Auth result for: 00:26:bb:77:d2:a7  MAC AUTH succeeded
wlEvent: en1 en1 Link UP virtIf = 0
AirPort: RSN handshake complete on en1
[snip]
```

Allocator Zones

- Important kernel data structures are created using the zone allocator
- The allocator keeps track of both active and previously freed objects
- The free lists can be used to find historical objects in a structured manner

Allocator Zones

```
$ python vol.py --profile=MacMountainLion_10_8_3_AMDx64 -f 10.8.3.mmr.macho mac_list_zones
Volatile Systems Volatility Framework 2.3_alpha
Name                          Active Count   Free Count   Element Size
----------------------------- ----------------  ----------   ------------
zones                           182             0              592
vm.objects                      153401          8832498        224
vm.object.hash.entries          135206          882875         40
maps                            149             34033          232
VM.map.entries                  26463           24372727       80
Reserved.VM.map.entries         35              13164          80
VM.map.copies                   0               220097         80
pmap                            139             7962           256
pagetable.anchors               139             7962           4096
proc                            133             4042           1120
```

mac_dead_procs

```
$ python vol.py --profile=MacMountainLion_10_8_3_AMDx64 -f ~/10.8.3.mmr.macho mac_dead_procs
Volatile Systems Volatility Framework 2.3_alpha
Offset                   Name                 Pid      Uid     Gid      PGID    Bits        DTB              Start Time
------------------       --------------------  -------- -------- -------- -------- ------------ ------------------  ----------
0xffffff8036349760 diskmanagementd     4158      -         -          -55...11            ------------------  2013-03-29 12:14:31 UTC+0000
0xffffff8036349760 diskmanagementd     4158      -         -          -55...11            ------------------  2013-03-29 12:14:31 UTC+0000
0xffffff8032c60d20  lssave                  4161    -         -          -55...11            ------------------ 2013-03-29 12:14:43 UTC+0000
0xffffff803dfe08e0  com.apple.audio.     4146      -         -          -55...11            ------------------ 2013-03-29 12:12:59 UTC+0000
0xffffff803dfe0d40  com.apple.audio.     4145      -         -          -55...11            ------------------ 2013-03-29 12:12:59 UTC+0000
0xffffff8032c62300 com.apple.qtkits      4147      -         -          -55...11            ------------------ 2013-03-29 12:12:59 UTC+0000
[snip]
```

Kernel Rootkit Detection

- Volatility provides the most comprehensive kernel-rootkit detection available
- We will now walkthrough analyzing a memory sample infected with the Rubilyn rootkit
- Other kernel rootkits employ similar or the same techniques as Rubilyn

mac_psxview

```
$ python vol.py -f rubilyn.vmem --profile=MacLion_10_7_5_AMDx64 mac_psxview
Volatile Systems Volatility Framework 2.3
 Offset(P)                 Name        PID      pslist  parents pid_hash pgrp_hash_table session leaders task processes
------------------         -------             ------  ------  -------  --------  ---------------      ---------------    ------------
0xffffff80008d8d40 kernel_task         0       True    True   False     True                True              True
0xffffff8005ee4b80 launchd             1       False  True    True      True                True              True
0xffffff8005ee4300 kextd               10      True  True     True      True                True              True
0xffffff8005ee3ec0 UserEventAgent  11      True  False     True      True                True              True
0xffffff8005ee3640 notifyd             12      True  False     True      True                True              True
0xffffff8005ee3200 mDNSResponder 13      True  False     True      True                True              True
0xffffff8005ee2dc0 opendirectoryd      14      True  False     True      True                True              True
0xffffff8005ee2980 diskarbitrationd    15      True  False     True      True                True              True
```

mac_check_sysctl

```
# python vol.py --profile=MacLion_10_7_5_AMDx64 -f rubilyn.vmem mac_check_sysctl
<snip>
pid2           102 RW-   0xffffff7f807ff14b UNKNOWN   0
pid3           103 RW-   0xffffff7f807ff1ed UNKNOWN   0
dir            104 RW-   0xffffff7f807ff2aa UNKNOWN
cmd            105 RW-   0xffffff7f807ff2bb UNKNOWN
user           106 RW-   0xffffff7f807ff2cc UNKNOWN
port           107 RW-   0xffffff7f807ff2dd UNKNOWN
```

mac_check_syscalls / mac_check_trap_table

```
$ python vol.py -f rubilyn.vmem --profile=MacLion_10_7_5_AMDx64 mac_check_syscalls | grep HOOK
Volatile Systems Volatility Framework 2.3
SyscallTable 222 0xffffff7f807ff41d HOOKED
SyscallTable 344 0xffffff7f807ff2ee HOOKED
SyscallTable 397 0xffffff7f807ffa7e HOOKED
```

The hooked entries allow the rootkit to hide files and file data from the file system

mac_ip_filters

```
$ python vol.py -f rubilyn.vmem --profile=MacLion_10_7_5_AMDx64 mac_ip_filters
Volatile Systems Volatility Framework 2.3
Context   Filter          Pointer              Status
---------- -------------- ------------------ ------
INPUT     rubilyn         0xffffff7f807ff577  OK
OUTPUT    rubilyn         0xffffff7f807ff5ff  OK
DETACH    rubilyn         0xffffff7f807ff607  OK
```

mac_notifiers

```
$ python vol.py --profile=MacMountainLion_10_8_3_AMDx64 -f ~/10.8.3.mmr.macho mac_notifiers
Volatile Systems Volatility Framework 2.3_alpha
Status     Key                       Handler              Matches
---------- ------------------------  ---------------      -------
OK         IOServicePublish          0xffffff7f8fa878e8   IODisplayConnect
OK         IOServicePublish          0xffffff7f91206ab6   IOResources,AppleClamshellState
OK         IOServicePublish          0xffffff7f8fa94188   IOResources,AppleClamshellState
OK         IOServicePublish          0xffffff800f872d50   IODisplayWrangler
OK         IOServicePublish          0xffffff7f902ff732   IOHIDevice
OK         IOServicePublish          0xffffff7f902ff732   IOHIDEventService
OK         IOServicePublish          0xffffff7f902ff732   IODisplayWrangler
OK         IOServicePublish           0xffffff7f902ffe74  AppleKeyswitch
[snip]
```

Work from @osxreverser & Friends

- Their initial releases led to mac_trustedbsd
- Their second round of rootkit techniques led to Cem Gurkok’s submission to the Volatility plugin contest [4]

mac_volshell & mac_yarascan

- MHL ported Volatility's yarascan infrastructure and volshell plugin to work with both Linux & Mac
- yarascan:
  - Search yara rules or simple strings across processes or kernel memory
- volshell:
  - Fully interactive Python shell inside Volatility environment

Mac Analysis

- Mac memory forensics has come a long way in the last year
  - Still some work to be done to reach the level of Windows & Linux, but that will be fixed soon
- 10.9.x has some interesting new research areas
  - Particularly the compressed free pages
  - Dr. Golden Richard of the University of New Orleans has implemented compressed page support into Volatility

Want to Learn Memory Forensics?

- Community Documentation [5]
  - Links to all memory forensics research published by entire forensics community
- Blog [6]
  - “Solving the GrrCon Network Forensics Challenge with Volatility ” [7]

Questions/Comments?

- Contact info:
  - andrew@memoryanalysis.net
  - @attrc

References

[1] https://www.blackhat.com/presentations/bh-dc-10/Suiche_Matthieu/Blackhat-DC-2010-Advanced-Mac-OS-X-Physical-Memory-Analysis-slides.pdf

[2] http://code.google.com/p/volafox/

[3] https://code.google.com/p/volatility/wiki/MacMemoryForensics#Download_pre-built_profiles

[4] http://www.volatilityfoundation.org/contest/2013/CemGurkok_OSXDetect.zip

[5] http://code.google.com/p/volatility/wiki/VolatilityDocumentationProject

[6] http://volatility-labs.blogspot.com/

[7] http://volatility-labs.blogspot.com/2012/10/solving-grrcon-network-forensics.html