



# SANS

[www.sans.org](http://www.sans.org)

**FORENSICS 518**

MAC FORENSIC  
ANALYSIS

518.3

System and  
Local Domain File  
Analysis

*The right security training for your staff, at the right time, in the right location.*

Copyright © 2015, The SANS Institute. All rights reserved. The entire contents of this publication are the property of the SANS Institute.

**IMPORTANT-READ CAREFULLY:**

This Courseware License Agreement ("CLA") is a legal agreement between you (either an individual or a single entity; henceforth User) and the SANS Institute for the personal, non-transferable use of this courseware. User agrees that the CLA is the complete and exclusive statement of agreement between The SANS Institute and you and that this CLA supersedes any oral or written proposal, agreement or other communication relating to the subject matter of this CLA. If any provision of this CLA is declared unenforceable in any jurisdiction, then such provision shall be deemed to be severable from this CLA and shall not affect the remainder thereof. An amendment or addendum to this CLA may accompany this courseware. BY ACCEPTING THIS COURSEWARE YOU AGREE TO BE BOUND BY THE TERMS OF THIS CLA. IF YOU DO NOT AGREE YOU MAY RETURN IT TO THE SANS INSTITUTE FOR A FULL REFUND, IF APPLICABLE. The SANS Institute hereby grants User a non-exclusive license to use the material contained in this courseware subject to the terms of this agreement. User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of this publication in any medium whether printed, electronic or otherwise, for any purpose without the express written consent of the SANS Institute. Additionally, user may not sell, rent, lease, trade, or otherwise transfer the courseware in any way, shape, or form without the express written consent of the SANS Institute.

The SANS Institute reserves the right to terminate the above lease at any time. Upon termination of the lease, user is obligated to return all materials covered by the lease within a reasonable amount of time.

SANS acknowledges that any and all software and/or tools presented in this courseware are the sole property of their respective trademark/registered/copyright owners.

AirDrop, AirPort, AirPort Time Capsule, Apple, Apple Remote Desktop, Apple TV, App Nap, Back to My Mac, Boot Camp, Cocoa, FaceTime, FileVault, Finder, FireWire, FireWire logo, iCal, iChat, iLife, iMac, iMessage, iPad, iPad Air, iPad Mini, iPhone, iPhoto, iPod, iPod classic, iPod shuffle, iPod nano, iPod touch, iTunes, iTunes logo, iWork, Keychain, Keynote, Mac, Mac Logo, MacBook, MacBook Air, MacBook Pro, Macintosh, Mac OS, Mac Pro, Numbers, OS X, Pages, Passbook, Retina, Safari, Siri, Spaces, Spotlight, There's an app for that, Time Capsule, Time Machine, Touch ID, Xcode, Xserve, App Store, and iCloud are registered trademarks of Apple Inc.



**COMPUTER FORENSICS**  
and INCIDENT RESPONSE



## FOR518

### Section 3 – System & Local Domain File Analysis

The **SANS** Institute



Sarah Edwards  
oompa@csh.rit.edu  
@iamevl twin



@sansforensics

<http://computer-forensics.sans.org>

© SANS.  
All Rights Reserved.

Mac Forensic Analysis

Author: Sarah Edwards  
oompa@csh.rit.edu  
<http://twitter.com/iamevl twin>  
<http://twitter.com/sansforensics>

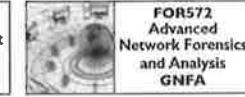


**D F I R C U R R I C U L U M**

C O R E

	<b>FOR408</b> Windows Forensics GCFE		<b>SEC504</b> Hacker Techniques, Exploits, and Incident Handling GCIH
---	---	---	---

I N - D E P T H I N C I D E N T R E S P O N S E

	<b>FOR508</b> Advanced Incident Response GCFA		<b>FOR572</b> Advanced Network Forensics and Analysis GNFA
---	--	---	--

	<b>FOR610</b> REM: Malware Analysis GREM
--	---

S P E C I A L I Z A T I O N

	<b>FOR518</b> Mac Forensics		<b>FOR526</b> Memory Forensics In-Depth
---	-----------------------------------	---	--

	<b>MGT535</b> Incident Response Team Management		<b>FOR585</b> Advanced Smartphone Forensics
---	--	---	--

**Join The SANS DFIR Community**

 Blog: <a href="http://dfir.to/DFIRBlog">dfir.to/DFIRBlog</a>	 Twitter: <a href="https://twitter.com/sansforensics">@sansforensics</a>	 Facebook: <a href="https://facebook.com/sansforensics">sansforensics</a>	 Google+: <a href="https://plus.google.com/sansforensics">gplus.to/sansforensics</a>
 Mailing list: <a href="https://dfir.to/MAIL-LIST">dfir.to/MAIL-LIST</a>	 YouTube: <a href="https://dfir.to/DFIRCast">dfir.to/DFIRCast</a>		

This page intentionally left blank.



# SANS DFIR

## DIGITAL FORENSICS & INCIDENT RESPONSE

### DFIR CURRICULUM

#### CORE

<b>FOR408</b> Windows Forensics GCFE	<b>SEC504</b> Hacker Techniques, Exploits, and Incident Handling GCIH
---	---

**Website**  
[digital-forensics.sans.org](http://digital-forensics.sans.org)

**SIFT Workstation**  
[dfir.to/SANS-SIFT](http://dfir.to/SANS-SIFT)

**Join The SANS DFIR Community**



Blog: [dfir.to/DFIRBlog](http://dfir.to/DFIRBlog)

Twitter: @sansforensics



Facebook: [sansforensics](#)



Google+: [gplus.to/sansforensics](#)



Mailing list: [dfir.to/MAIL-LIST](mailto:dfir.to/MAIL-LIST)



YouTube: [dfir.to/DFIRCast](http://dfir.to/DFIRCast)



#### IN-DEPTH INCIDENT RESPONSE

<b>FOR508</b> Advanced Incident Response GCFA	<b>FOR572</b> Advanced Network Forensics and Analysis GNFA
--	--

3

#### SPECIALIZATION

<b>FOR518</b> Mac Forensics	<b>FOR526</b> Memory Forensics In-Depth
-----------------------------------	--

<b>MGT535</b> Incident Response Team Management	<b>FOR585</b> Advanced Smartphone Forensics
--	--

# Course Agenda

---

Section 1 – Mac Essentials & the HFS+ File System

Section 2 – User Domain File Analysis

Section 3 – System & Local Domain File Analysis

Section 4 – Advanced Analysis Topics

Section 5 – iOS Analysis

Section 6 – Mac Forensic Challenge

This page intentionally left blank.



**COMPUTER FORENSICS**  
and INCIDENT RESPONSE



# System & Local Domain File Analysis

The SANS Institute  
Sarah Edwards

© SANS.  
All Rights Reserved

Mac Forensic Analysis

This page intentionally left blank.

## Section 3 - Agenda

---

Part 1 – System Information

Part 2 – System Preferences & Applications

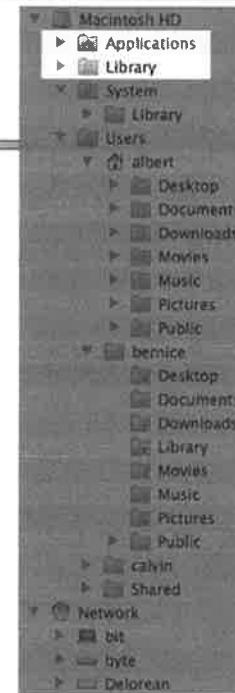
Part 3 – Log Analysis

Part 4 – Timeline Analysis & Data Correlation

This page intentionally left blank.

# Local Domain

- /Applications
  - Contains applications for all users
- /Library
  - Local Library - Contains application specific data
- /Developer
  - ...or /Library/Developer
  - ...or /Applications/Xcode.app/...



© SANS.  
All Rights Reserved

Mac Forensic Analysis

The local domain consists of the Applications directory, the (local) Library directory, and if installed, the Developer directory. This domain is used to store the files that may be shared amongst the users, such as applications.

The /Applications directory contains the programs available to all users of the system. It is not necessary to run an application from the /Applications directory. It may be run from a user directory if needed. Applications from the Mac App Store are installed in the /Applications directory.

The /Developer directory is created when Xcode is installed. The Developer directory may be located in either the root (/), under /Library or embedded in the Xcode.app application in the /Applications directory. The Developer folder contains Apple Developer (Mac, iOS,) related data and resources.

Reference:

File System Programming Guide – File System Basics

<https://developer.apple.com/library//mac/#/library/mac/documentation/FileManagement/Conceptual/FileSystemProgrammingGuide/FileSystemOverview/FileSystemOverview.html>

**System Domain**

- `/System`
  - Apple System Resources
- `/System/Library`
  - System Library

© SANS.  
All Rights Reserved.

Mac Forensic Analysis

The System Domain is used to store Apple specific system software.

The System Domain contains the `system Library` directory which contains files associated with Apple system resources.

You may have noticed a common theme in the directory structure. There are three `Library` directories on Mac OS X, each with its own purpose.

- User Library - `/Users/<username>/Library/`
- Local Library - `/Library/`
- System Library - `/System/Library/`

It is easy to get confused about which Library directory contains what data. This class will discuss the difference between each Library directory and the contents located within.

Reference:

File System Programming Guide – File System Basics

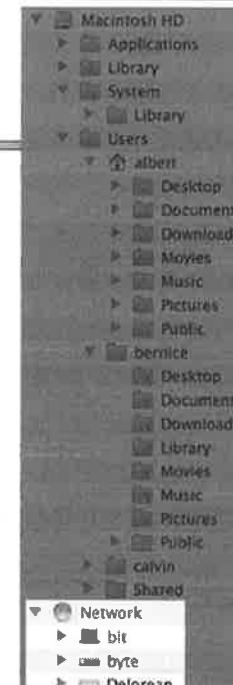
<https://developer.apple.com/library//mac/#/library/mac/documentation/FileManagement/Conceptual/FileSystemProgrammingGuide/FileSystemOverview/FileSystemOverview.html>

File System Program Guide – OS X Library Directory Details

<https://developer.apple.com/library//mac/#/library/mac/documentation/FileManagement/Conceptual/FileSystemProgrammingGuide/MacOSXDirectories/MacOSXDirectories.html>

# Network Domain

- Mac Computers
- PC Computers
- Time Capsules
- File Shares
- Network Attached Storage
- Etc.



© SANS.  
All Rights Reserved

Mac Forensic Analysis

The Network Domain contains the network resources such as network area storage, Time Capsules, and other systems on the network.

## Reference:

File System Programming Guide – File System Basics

<https://developer.apple.com/library//mac/#/library/mac/documentation/FileManagement/Conceptual/FileSystemProgrammingGuide/FileSystemOverview/FileSystemOverview.html>

## Standard Unix Directories [1]

/bin

- Contains binaries such as cat, echo, and mv.

/sbin

- Contains “system” binaries such as fsck, mount, and ping

/dev

- Contains “device” files such as disk0s2, stdout, and zero

/opt

- Contains “optional” software. Default install location for package management tools such as macports or fink

Mac OS X systems contains standard Unix directories.

- /bin contains various command utilities
- /sbin contains contains system binaries
- /dev contains “device” files
- /opt contains “optional” software

## Standard Unix Directories [2]

/private/var

- Contains “variable” directories. The contents changes often. Notable directories include /log, /db, and /audit

/private/etc

- Contains system configuration data such as passwd, hosts, and resolv.conf

/private/tmp

- Contains temporary files

© SANS.  
All Rights Reserved

Mac Forensic Analysis

Mac OS X systems contains standard Unix directories.

/private/var contains notable “variable” directory such as /log, /db, and /audit.

/private/etc contains system configuration data, with notable files such as hosts, passwd, and resolv.conf.

/private/tmp contains temporary files.

It is worth noting that the files var, etc, and tmp are all symbolic links to their /private equivalent.

# Agenda

---

Part 1 – System Information

Part 2 – System Preferences & Applications

Part 3 – Log Analysis

Part 4 – Timeline Analysis & Data Correlation

This page intentionally left blank.



## Section 3 – Part 1 System Information

This page intentionally left blank.

## System Information

System Version

Installation Information

Network Configuration

User Accounts

Time Zone Settings

© SANS.  
All Rights Reserved

Mac Forensic Analysis

The System and Local domains contain initial triage information that can be useful to an investigator, such as:

- System Version
- Installation Information
- Network Configuration
- User Accounts
- Time Zone Settings

## System Version

### /System/Library/CoreServices/SystemVersion.plist

▼ Root	Dictionary	(5 items)
ProductBuildVersion	String	12D78
ProductCopyright	String	1983-2013 Apple Inc.
ProductName	String	Mac OS X
ProductUserVisibleVersion	String	10.8.3
ProductVersion	String	10.8.3

© SANS.  
All Rights Reserved

Mac Forensic Analysis

The SystemVersion.plist property list located in the /System/Library/CoreServices/ directory contains the system version and build information.

In the example above, the system name and version is Mac OS X 10.8.3, while the build version is 12D78.

# System Installation

/private/var/db/.AppleSetupDone

/private/var/db/.AppleInstallType.plist

/private/var/log/install.log

```
Jul 15 19:34:25 localhost OSInstaller[375]: Installed "OS X" (10.9.4 (13E28))
Jul 15 19:34:25 localhost OSInstaller[375]: PackageKit: ----- End install -----
Jul 15 19:34:25 localhost OSInstaller[375]: PackageKit: 857.8s elapsed install time
Jul 15 19:34:26 localhost OSInstaller[375]: Running install actions
Jul 15 19:34:26 localhost OSInstaller[375]: Writing installation cookies
Jul 15 19:34:26 localhost OSInstaller[375]: InstallType cookie file was successfully
Jul 15 19:34:26 localhost OSInstaller[375]: Removing temporary directory "/Volumes/l
Jul 15 19:34:26 localhost OSInstaller[375]: Finalize disk "MBP" for OS Installation
Jul 15 19:34:26 localhost OSInstaller[375]: Finalizing Disk "MBP" for OS Install
nibble:db compa$ ls -la .Apple*
-rw-r--r-- 1 root wheel 233 Jul 15 22:34 .AppleInstallType.plist
-rw-r--r-- 1 root wheel 0 Jul 15 22:47 .AppleSetupDone
```

© SANS,  
All Rights Reserved

Mac Forensic Analysis

An original installation date may be found as the creation date for the .AppleSetupDone and .AppleInstallType.plist files located in the /private/var/db/ directory.

The OS X installation date may be able to be found in the install.log files located in the /var/log directory if the older files have not been turned over. These dates are when different versions of OS X were installed after the original installation.

# Time Zone Setting



- /etc/localtime

```
nibble:etc sledwards$ ls -l localtime
lrwxr-xr-x 1 root wheel 36 Apr 13 17:28 localtime -> /usr/share/zoneinfo/America/New_York
```

- /Library/Preferences/.GlobalPreferences.plist

com.apple.preferences.timezone.selected_city	Dictionary	(10 items)
RegionalCode	String	DC
Version	Number	1
TimeZoneName	String	America/New_York
Latitude	Number	38.89511
GeonameID	Number	4,140,963
Population	Number	601,723
Longitude	Number	-77.03637
CountryCode	String	US
Name	String	Washington D.C.

© SANS.  
All Rights Reserved

Mac Forensic Analysis

The link to the file located in /etc/localtime contains the current time zone value for the system. In the screenshot you can see the local time zone is set for New York or the Eastern Standard Time zone.

The property list located in the /Library/Preferences/ directory contains the time zone configuration data. The time zone is set for Washington, DC. This is for the same system configured for the New York time zone shown above. While the city may change, the time zone may be located in a more general location.

This configuration is likely chosen by the user when the system was configured during setup using the time zone map feature.

# Network Information - Interfaces

/Library/Preferences/SystemConfiguration/NetworkInterfaces.plist

Item	Dictionary
en0	Dictionary of 19 items
Active	Boolean: YES
BSD Name	String: en0
IOUnit	Boolean: YES
IOInterfaceType	Number: 6
IOInterfaceUnit	Number: 0
IONetworkAddress	Data: <2c1e3ef6:67>
IPPathMatch	String: /OSService:/AppleACPIPlatformExpert/PCI@0/AppleACPIPCI/EHC1@1D/AppleUSBEHCI/Apple USB Spine Adapter@1d120000/AppleUSBethernetZ
SCNetworkInterfaceInfo	Dictionary of 1 item
SCNetworkInterfaceType	String: Ethernet
en1	Dictionary of 18 items
BSD Name	String: en1
IOUnit	Boolean: NO
IOInterfaceType	Number: 6
IOInterfaceUnit	Number: 1
IONetworkAddress	Data: <2c1e3ef6:68>
IPPathMatch	String: /OSService:/OSResources:/OSUserEthernetResource/OSUserEthernetResourceClient/OSUserEthernetController/OSUserEthernetInterface
SCNetworkInterfaceInfo	Dictionary of 1 item
SCNetworkInterfaceType	String: Ethernet
en2	Dictionary of 18 items
BSD Name	String: en2
IOUnit	Boolean: NO
IOInterfaceType	Number: 5
IOInterfaceUnit	Number: 2
IONetworkAddress	Data: <c58d3255:c1:b>
IPPathMatch	String: /OSService:/AppleACPIPlatformExpert/PCI@0/AppleACPIPCI/EHC1@1D/AppleUSBEHCI/Apple USB Spine Adapter@1d120000/AppleUSBethernetZ
SCNetworkInterfaceInfo	Dictionary of 1 item
SCNetworkInterfaceType	String: Ethernet
en3	Dictionary of 18 items
BSD Name	String: en3
IOUnit	Boolean: NO
IOInterfaceType	Number: 6
IOInterfaceUnit	Number: 3
IONetworkAddress	Data: <c0460816:f9:3f>
IPPathMatch	String: /OSService:/AppleACPIPlatformExpert/PCI@0/AppleACPIPCI/EHC2@1A/ApplePSUHCI/A2BBv22A@1a120800/AppleUSBEthernetZ
SCNetworkInterfaceInfo	Dictionary of 1 item
SCNetworkInterfaceType	String: Ethernet

© SANS.  
All Rights Reserved.

Mac Forensic Analysis

The NetworkInterfaces.plist file located in the /Library/Preferences/SystemConfiguration/ directory contains the network interfaces for the system.

Each network interface on the system will have an Item key.

In the screenshot, the example has four network interfaces:

- en0 – 802.11 Airport Card
- en1 – Physical Ethernet interface – This particular system does not have a physical Ethernet port as it is a MacBook Air. We can see this system is a MacBook Air from the “Model” key, shown at the bottom of the property list.
- en2 – USB Ethernet Adapter – The MacBook Air can use a USB Ethernet adapter dongle as a wired Ethernet port.
- en3 – USB Ethernet Adapter – A different Ethernet adapter used by this system.

You may also see a Thunderbolt Ethernet adapter in this list. Instead of using a USB port it will use the Thunderbolt port on the system.

		Dictionary (9 items)
▼ Item 0		
Active	Boolean	YES
BSD Name	String	en0
IOBuiltin	Boolean	YES
IOInterfaceType	Number	6
IOInterfaceUnit	Number	0
IOMACAddress	Data	<7cd1c3df 6467>
IOPathMatch	String	IOService:/AppleACPILPlatformExpert@FC0@0/AppleACPILPlatformExpert@IC,1/iOPCIPCIBridge/APT@0/AirPort_Brcm4331/e0
► SCNetworkInterfaceFaceInfo	Dictionary (1 item)	
SCNetworkInterfaceType	String	IEEE80211
▼ Item 1	Dictionary (8 items)	
BSD Name	String	en1
IOBuiltin	Boolean	NO
IOInterfaceType	Number	6
IOInterfaceUnit	Number	1
IOMACAddress	Data	<7cd1c3df 6468>
IOPathMatch	String	IOService:/IOResources/IOUserEthernetResource/IOUserEthernetResourceUseClient/IOUserEthernetController//OEthernetInterface
► SCNetworkInterfaceFaceInfo	Dictionary (1 item)	
SCNetworkInterfaceType	String	Ethernet
▼ Item 2	Dictionary (8 items)	
BSD Name	String	en2
IOBuiltin	Boolean	NO
IOInterfaceType	Number	6
IOInterfaceUnit	Number	2
IOMACAddress	Data	<b8d1255 e1b0>
IOPathMatch	String	IOService:/AppleACPILPlatformExpert/PCI@0/AppleACPICl/EHC1@1D/AppleUSBEHCI/Apple USB Ethernet Adapter@1d110000/AppleUSBEthernet/en2
► SCNetworkInterfaceFaceInfo	Dictionary (4 items)	
SCNetworkInterfaceType	String	Ethernet
► Item 3	Dictionary (8 items)	
BSD Name	String	en3
IOBuiltin	Boolean	NO
IOInterfaceType	Number	6
IOInterfaceUnit	Number	3
IOMACAddress	Data	<c080808f 93ff>
IOPathMatch	String	IOService:/AppleACPILPlatformExpert/PCI@0/AppleACPICl/EHC2@1A/AppleUSBEHCI/AX88x72A@1a120000/AppleUSBEthernet/en3
► SCNetworkInterfaceFaceInfo	Dictionary (4 items)	
SCNetworkInterfaceType	String	Ethernet
Mode	String	MacBookAir5,1

# Network Information – Configuration

/Library/Preferences/SystemConfiguration/preferences.plist

<pre>▼ 29A1FDC6-B462-4518-... Dictionary (7 items)   ▼ DNS Dictionary (1 item)     ▼ ServerAddresses Array (0 items)   ▼ IPv4 Dictionary (1 item)     ConfigMethod String DHCP   ▼ IPv6 Dictionary (2 items)     ConfigMethod String Automatic     ▀ INACTIVE Boolean YES   ▼ Interface Dictionary (4 items)     DeviceName String en0     Hardware String AirPort     Type String Ethernet     UserDefinedName String Wi-Fi   ▼ Proxies Dictionary (2 items)     ▼ ExceptionsList Array (2 items)       Item 0 String *.local       Item 1 String 169.254/16     FTPPassive Number 1   ▼ SMB Dictionary (1 item)     NetBIOSName String nibble     UserDefinedName String Wi-Fi</pre>	<pre>▼ CE4DF79D-2811-444D-... Dictionary (7 items)   ▼ DNS Dictionary (0 items)   ▼ IPv4 Dictionary (4 items)     ▼ Addresses Array (1 item)       Item 0 String 192.168.123.123     ConfigMethod String Manual     Router String 192.168.1.254     ▼ SubnetMasks Array (1 item)       Item 0 String 255.255.255.0   ▼ IPv6 Dictionary (1 item)     ConfigMethod String Automatic   ▼ Interface Dictionary (4 items)     DeviceName String en4     Hardware String Ethernet     Type String Ethernet     UserDefinedName String Thunderbolt Ethernet   ▼ Proxies Dictionary (2 items)     ▼ ExceptionsList Array (2 items)       Item 0 String *.local       Item 1 String 169.254/16     FTPPassive Number 1   ▼ SMB Dictionary (0 items)     UserDefinedName String Thunderbolt Ethernet</pre>
--	--

© SANS,  
All Rights Reserved

Mac Forensic Analysis

The NetworkServices Key contains the configuration for each network interface. Two examples are shown above.

The example on the left contains a Wi-Fi interface (UserDefinedName key). This interface uses DHCP (rather than a static IP) and it has a NetBIOSName Key that contains the NetBIOS name of the system. The network interface device is en0.

The example on the right contains a Thunderbolt Ethernet interface (UserDefinedName key). This interface has a static IP configured (as well as Router and Subnet Mask). The network interface device is en4.

## Network Information – DHCP Addresses

### /private/var/db/dhcpclient/leases/

```
bash-3.2# pwd
/private/var/db/dhcpclient/leases
bash-3.2# ls -l
total 16
-rw-r--r-- 1 root wheel 969 May 10 10:20 en0-1,b8:e8:56:37:ec:6
-rw-r--r-- 1 root wheel 927 Feb 18 20:48 en4-1,68:5b:35:91:1a:b5
bash-3.2# plutil -p en4-1\,68\:5b\:35\:91\:1a\:b5
{
    "LeaseStartDate" => 2014-02-19 01:39:52 +0000
    "RouterHardwareAddress" => <e0699550 4c06>
    "IPAddress" => "192.168.1.237"
    "LeaseLength" => 43200
    "RouterIPAddress" => "192.168.1.254"
    "PacketData" => <02010600 7a48b9f4 000d0000 00000000 c0a801ed c0a801fe 00000000 685b3591 1ab
50000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00
000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00
000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 63825363 35010
536 04c0a801 fe330400 00a8c03a 04000054 603b0400 0093a801 04fffffe 001c04c0 a801ff03 04c0a801
fe0604c0 a801feff 00000000 00000000>
}
```

© SANS.  
All Rights Reserved.

Mac Forensic Analysis

The XML-based files located in the /private/var/db/dhcpclient/leases/ directory contains files with configurations for DHCP network settings. These contain the settings for the latest connection on the specified interface.

Shown in the screenshot above, the example shows two DHCP configurations. One for the en0 adapter and another for the en4 adapter. Each adapter has an associated MAC address in the filename.

Each file contains:

- Lease Start Date
- Router MAC Address
- Assigned IP Address
- SSID of Access Point (Wi-Fi only)
- DHCP Lease Length (in minutes)
- Router IP Address
- Packet Data

**Network Information – Wi-Fi (10.9)**

/Library/Preferences/SystemConfiguration/com.apple.airport.preferences.plist

The screenshot shows two windows side-by-side. On the left is a property list editor titled "Network Information – Wi-Fi (10.9)" showing the contents of the file "/Library/Preferences/SystemConfiguration/com.apple.airport.preferences.plist". The "RememberedNetworks" key is expanded, showing five items (Item 0 to Item 4) and one additional item (Item 5). Item 5 contains several attributes: AutoLogin (Boolean, NO), CachedScanRecord (Dictionary, 13 items), Captive (Boolean, YES), Closed (Boolean, NO), Disabled (Boolean, NO), LastConnected (Date, Jun 12, 2012 1:07:23 PM), SSID (Data, <50414e45 5241>), SSIDString (String, PANERA), SecurityType (String, Open), SystemMode (Boolean, YES), TemporarilyDisabled (Boolean, NO), and Version (Number, 10). On the right is a "Wi-Fi" settings dialog from the Mac OS X interface, showing a list of "Preferred Networks" with names like Washington Dulles Intl, Manliest Coast, Marlett Conference, and CLYNET, all set to "None" security. Below the list are checkboxes for "Remember networks this computer has joined", "Require administrator authorization to change", "Create computer-to-computer networks", "Change networks", and "Turn Wi-Fi on or off". The "Wi-Fi Address" is listed as 90:27:b4.

Mac Forensic Analysis

© SANS.  
All Rights Reserved

The `com.apple.airport.preferences.plist` property list file located in the `/Library/Preferences/SystemConfiguration/` directory contains network information of the “remembered” or “saved” networks. Each network is stored in its own `Item` key.

The “remembered” networks are a list of networks the system has previously established a connection; These items do not appear to be purged unless performed by the user. If removed using the GUI shown, the items will be removed from the property list.

Some of the attributes available for each network include:

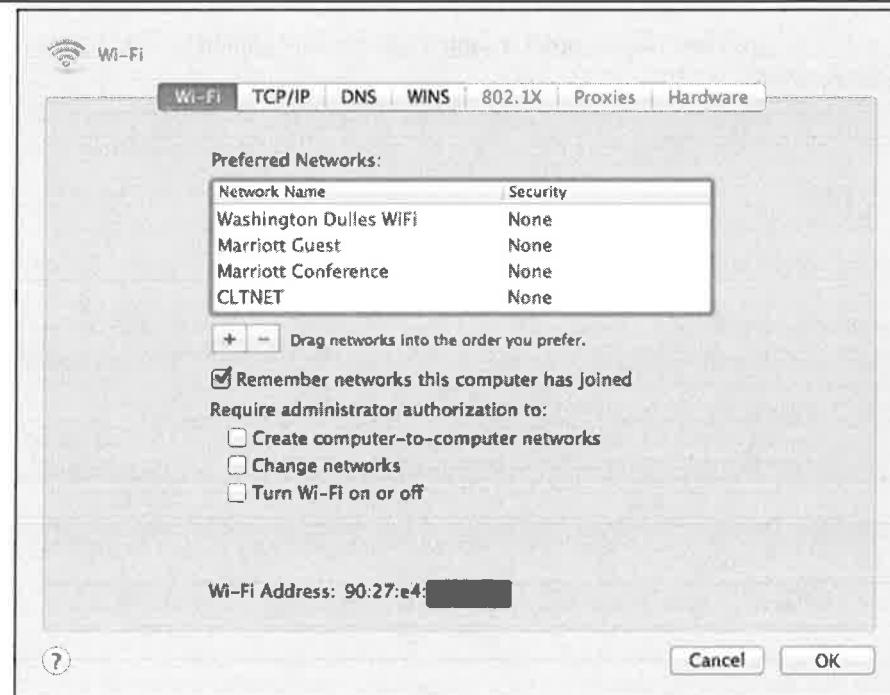
- Captive (that popup screen you get in hotels and restaurants)
- When the system last connected to the network, stored in local system time.
- Network SSID
- Automatic logon

These networks can help an investigator determine where a system might have travelled to if the network BSSIDs are unique such as a restaurant or hotel name.

Note: This file will only be available if the system has a wireless network card available.

By default, these are stored in a chronological format (Item 0 is the oldest), however the user can use the GUI to change the order in which they attempt to connect which will change the order in the property list file.

Key	Type	Value
▼ RememberedNetworks	Array	(6 items)
► Item 0	Dictionary	(11 items)
► Item 1	Dictionary	(11 items)
► Item 2	Dictionary	(11 items)
► Item 3	Dictionary	(11 items)
► Item 4	Dictionary	(11 items)
▼ Item 5	Dictionary	(11 items)
AutoLogin	Boolean	NO
► CachedScanRecord	Dictionary	(13 items)
Captive	Boolean	YES
Closed	Boolean	NO
Disabled	Boolean	NO
LastConnected	Date	Jun 12, 2012 1:07:23 PM
SSID	Data	<50414e45 5241>
SSIDString	String	PANERA
SecurityType	String	Open
SystemMode	Boolean	YES
TemporarilyDisabled	Boolean	NO
Version	Number	10



# Network Information – Wi-Fi (10.10)

/Library/Preferences/SystemConfiguration/com.apple.airport.preferences.plist

Key	Type	Value
Root	Dictionary	{5 items}
Counter	Number	2
KnownNetworks	Dictionary	{3 items}
> wifi.ssid.<48790174 74204775 65737472 0f0f0d>	Dictionary	{15 items}
> wifi.ssid.<0d6f0e289 6c652d77 6972656c 657373>	Dictionary	{15 items}
> wifi.ssid.<76657972 0f0e>	Dictionary	{16 items}
Autologin	Boolean	NO
Captive	Boolean	NO
> ChannelHistory	Array	{1 item}
Closed	Boolean	YES
> CollocatedGroup	Array	{0 items}
Disabled	Boolean	NO
LastConnected	Date	Dec 10, 2014, 5:43:48 PM
PaePoint	Boolean	NO
PossiblyHiddenNetwork	Boolean	NO
RoamingProfileType	String	Single
SPRoaming	Boolean	NO
SSID	Date	<76657972 0f0e>
SSIDString	String	verizon
SecurityType	String	WPA2 Personal
SystemMode	Boolean	YES
TemporarilyDisabled	Boolean	NO
> PreferredOrder	Array	{3 items}
Item 0	String	wifi.ssid.<76657972 0f0e>
Item 1	String	wifi.ssid.<0d6f0e289 6c652d77 6972656c 657373>
Item 2	String	wifi.ssid.<48790174 74204775 65737472 0f0f0d>
> UpdateHistory	Array	{1 item}
Version	Number	2,200

© SANS,  
All Rights Reserved

Mac Forensic Analysis

On 10.10 systems the com.apple.airport.preferences.plist property list file contains similar information, however is organized a bit differently.

The keys under KnownNetworks contain keys named wifi.ssid.<hex>. The <hex> is the hex representation of the network's SSID name.

The PreferredOrder key contains the order in which access points should be used in order of preference. Item 0 is the most preferred.

Key	Type	Value
▼ Root	Dictionary	(5 items)
Counter	Number	2
▼ KnownNetworks	Dictionary	(3 items)
► wifi.ssid.<48796174 74204775 65737472 6f6f6d>	Dictionary	(16 items)
► wifi.ssid.<6d6f6269 6c652d77 6972656c 657373>	Dictionary	(15 items)
▼ wifi.ssid.<76657972 6f6e>	Dictionary	(16 items)
AutoLogin	Boolean	NO
Captive	Boolean	NO
► ChannelHistory	Array	(1 item)
Closed	Boolean	YES
► CollocatedGroup	Array	(0 items)
Disabled	Boolean	NO
LastConnected	Date	Dec 16, 2014, 5:43:48 PM
Passpoint	Boolean	NO
PossiblyHiddenNetwork	Boolean	NO
RoamingProfileType	String	Single
SPRoaming	Boolean	NO
SSID	Data	<76657972 6f6e>
SSIDString	String	veyron
SecurityType	String	WPA2 Personal
SystemMode	Boolean	YES
TemporarilyDisabled	Boolean	NO
▼ PreferredOrder	Array	(3 items)
Item 0	String	wifi.ssid.<76657972 6f6e>
Item 1	String	wifi.ssid.<6d6f6269 6c652d77 6972656c 657373>
Item 2	String	wifi.ssid.<48796174 74204775 65737472 6f6f6d>
► UpdateHistory	Array	(1 item)
Version	Number	2,200

## Determine “Home” (or “Work”) Wi-Fi Network

### com.apple.airport.preferences.plist

- “Item 0”
- SecurityType != OPEN
  - “OPEN” generally seen at Wi-Fi hotspots
  - Example: My home network is “WPA2 Personal”

### system.log

- More entries than most others when “airportd” searched for

© SANS.  
All Rights Reserved.

Mac Forensic Analysis

To determine which one of the listed networks is the “home” network, such as the system owner’s home access point or the company enterprise Wi-Fi, analysis can be done on the airport property list file com.apple.airport.preferences.plist or the system.log. This is assuming the user is using Wi-Fi versus a wired connection.

The airport preferences file should at least have one Item key in the list if the wireless network adapter was used. When a system is setup, the first item or Item 0 is populated. Most users set up their new laptop or device in their own homes and offices. Security of these access points should (hopefully) be secured and not listed as “OPEN” as most Wi-Fi hotspots such as the local coffee shop.

The system.log file can be searched for the “airportd” term to determine where a device has established the most connections. Many users tend to use the device in one location more than any other such as their home or office.

This is not exactly a scientific method to determine a home network, but can help in determining the user profile of a system user.

# Deleted User Accounts [1]

/Library/Preferences/com.apple.preferences.accounts.plist

Key	Type	Value
▼ deletedUsers	Array	(2 items)
► Item 0	Diction...	(4 items)
▼ Item 1	Diction...	(4 items)
dsAttrTypeStandard:RealName	String	testuser
dsAttrTypeStandard:UniqueID	Number	502
name	String	testuser
date	Date	Jun 13, 2012 8:41:58 PM

© SANS  
All Rights Reserved

Mac Forensic Analysis

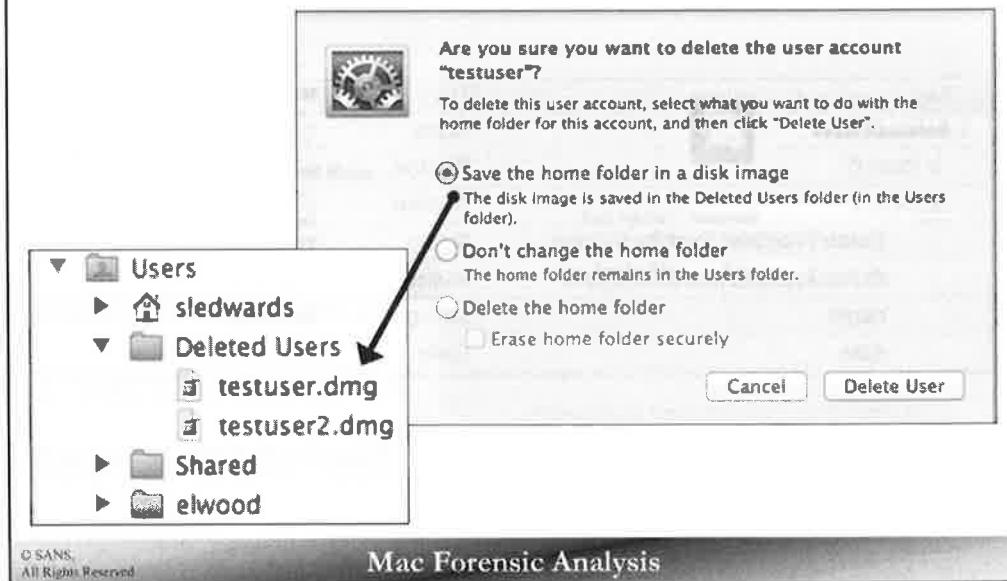
The deleted users are shown in the com.apple.preferences.accounts.plist file under the deletedUsers key. This property list is located in the /Library/Preferences/ directory.

This key contains:

- Deleted user's "Real Name"
- UID
- Username
- Deletion date (local system time)

## Deleted User Accounts [2]

/Library/Preferences/com.apple.preferences.accounts.plist



There are three options available when a user account is deleted:

- “Save the home folder in a disk image” – This default option archives the user’s home directory and saves it in a disk image file (DMG). The inset screenshot shows a couple of deleted user accounts saved to DMG files which are then moved to the /Users/Deleted Users/ directory.
- “Don’t change the home folder” – The home folder does not get archived, it stays in place.
- “Delete the home folder” – The user’s home directory is removed, and if selected can be removed securely (wiped).

When the user accounts are deleted, the user’s plist in the  
/private/var/db/dslocal/nodes/Default/users/ directory is also removed.

## Last User Logged in & Auto Login

/Library/Preferences/com.apple.loginwindow.plist

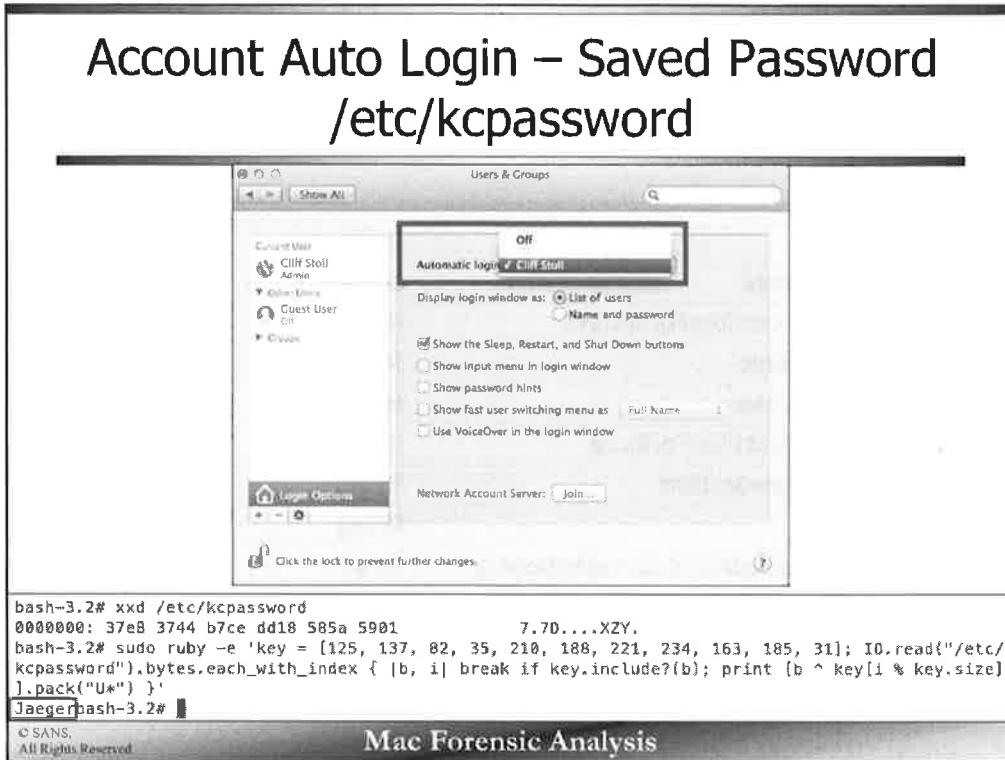
Root	Dictionary	(9 items)
GuestEnabled	Boolean	YES
OptimizerLastRunForSystem	Number	168,297,472
lastUserName	String	sledwards
autoLoginUser	String	sledwards
OptimizerLastRunForBuild	Number	25,429,728
MasterPasswordHint	String	
lastUser	String	loggedin
► AutoLaunchedApplicationDictionary	Array	(1 item)
RetriesUntilHint	Number	3

© SANS.  
All Rights Reserved

Mac Forensic Analysis

The com.apple.loginwindow.plist property list located in the /Library/Preferences/ directory contains:

- lastUser – If the a user is currently logged in (assuming the system was imaged live)
- autoLoginUser - The Auto Login user (if configured)
- lastUserName - Last logged in user
- RetriesUntilHint - The number of times before a password hint is given
- GuestEnabled - Guest Account Status
- MasterPasswordHint – If a master password has been configured, a hint may have also been configured.



A user may choose the ability to have the system automatically log them on using the “Automatic Login” selection window in the Users & Groups preferences pane.

The user’s password is then OR’d with a multi-byte key and stored in /etc/kpassword. To decode this password use the Ruby script below, provided by Lauri Ranta on [apple.stackexchange.com](http://apple.stackexchange.com).

```
sudo ruby -e 'key = [125, 137, 82, 35, 210, 188, 221, 234, 163, 185, 31];
IO.read("/etc/kpassword").bytes.each_with_index { |b, i| break if
key.include?(b); print [b ^ key[i % key.size]].pack("U*") }'
```

#### References:

<http://www.brock-family.org/gavin/perl/kpassword.html>

<http://apple.stackexchange.com/questions/50652/does-activating-auto-login-compromise-secure-password-storage>

# Agenda

---

Part 1 – System Information

Part 2 – System Preferences & Applications

Part 3 – Log Analysis

Part 4 – Timeline Analysis & Data Correlation

This page intentionally left blank.



## Section 3 – Part 2

### System Preferences & Applications

This page intentionally left blank.

# System Preferences & Application

Application Formats

Autoruns

Firewall Settings

Sharing Configuration

Printing Configuration

Bluetooth Configuration

Software & Install History

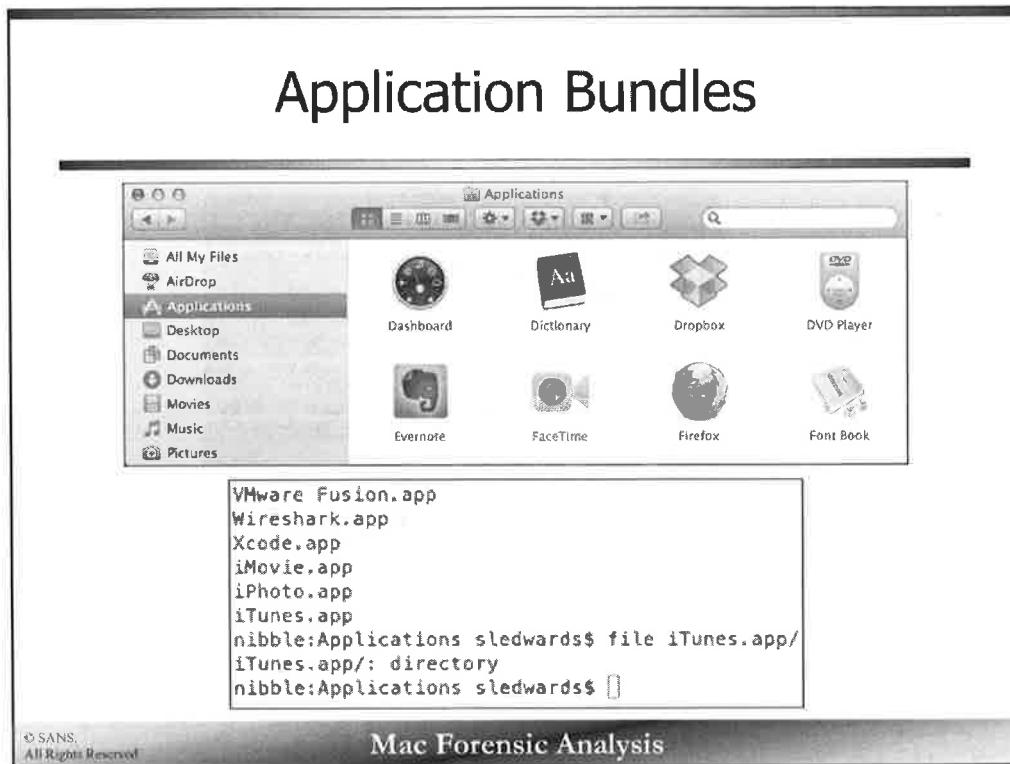
Kernel Extensions

Unix Periodic Scripts

© SANS,  
All Rights Reserved

Mac Forensic Analysis

This page intentionally left blank.



Applications on OS X systems come in a packaged format that appears as one file to the user in the Finder window.

The top screenshot shows some of the Applications available in the `/Applications` directory, each is accessible by double-clicking the application icon. Each one of these icons is actually the application bundle. It is worth noting that applications are not required to be run from the Applications directory.

The lower screenshot shows what these applications look like in the Terminal window. Each application has the `.app` file extension and is a directory containing other files.

#### References:

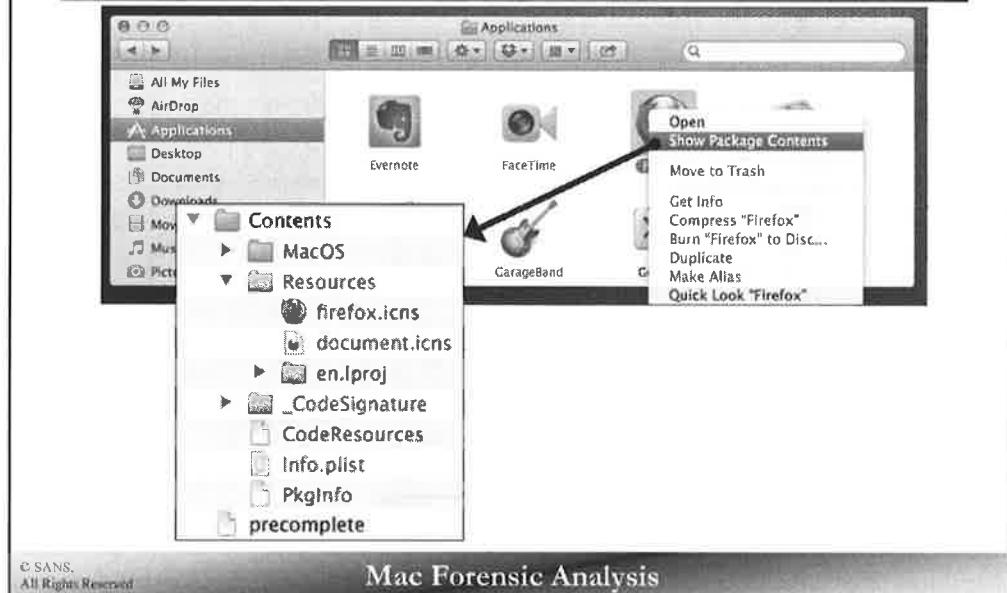
Bundle Programming Guide – Apple Developer Documentation

<http://developer.apple.com/library/mac/#documentation/CoreFoundation/Conceptual/CFBundles/AboutBundles/AboutBundles.html>

Application Bundles – Apple Developer Documentation

[http://developer.apple.com/library/mac/#documentation/CoreFoundation/Conceptual/CFBundles/BundleTypes/BundleTypes.html#/apple\\_ref/doc/uid/10000123i-CH101-SW13](http://developer.apple.com/library/mac/#documentation/CoreFoundation/Conceptual/CFBundles/BundleTypes/BundleTypes.html#/apple_ref/doc/uid/10000123i-CH101-SW13)

# Application Bundles Package Contents



Application bundles are directories that contain everything an application needs to execute such as:

- Executable Code
- Resource Files
- Support Files

These files can be accessed via the Finder application by “right-clicking” (Control+clicking) the “Show Package Contents” as shown in the screenshot above.

Each application bundle contains the same basic directory and file structure contained in the “Contents” directory.

- Info.plist File – Information Property List
- MacOS Directory – Contains executable code
- Resources Directory – Contains resource files

## References:

Bundle Programming Guide – Apple Developer Documentation

<http://developer.apple.com/library/mac/#documentation/CoreFoundation/Conceptual/CFBundles/AboutBundles/AboutBundles.html>

Application Bundles – Apple Developer Documentation

[http://developer.apple.com/library/mac/#documentation/CoreFoundation/Conceptual/CFBundles/BundleTypes/BundleTypes.html#/apple\\_ref/doc/uid/10000123i-CH101-SW13](http://developer.apple.com/library/mac/#documentation/CoreFoundation/Conceptual/CFBundles/BundleTypes/BundleTypes.html#/apple_ref/doc/uid/10000123i-CH101-SW13)

# Application Bundles

## Information Property List - Info.plist

▼ Information Property List		Dictionary	(19 items)
Localization native development region	String	English	
► Document types	Array	(7 items)	
Executable file	String	firefox	
Get Info string	String	Firefox 19.0.2	
Icon file	String	firefox	
Bundle identifier	String	org.mozilla.firefox	
InfoDictionary version	String	6.0	
Bundle name	String	Firefox	
Bundle OS Type code	String	APPL	
Bundle versions string, short	String	19.0.2	
Bundle creator OS Type code	String	MOZB	
► URL types	Array	(4 items)	
Bundle version	String	1913.3.7	
Scriptable	Boolean	YES	
Application Category	String	Productivity	
Minimum system version	String	10.6	
► Minimum system versions, per-architecture	Dictionary	(2 items)	
NSSupportsAutomaticGraphicsSwitching	Boolean	YES	
Principal class	String	GeckoNSApplication	

© SANS.  
All Rights Reserved.

Mac Forensic Analysis

The Information Property List (`Info.plist`) contains XML formatted data that describes the application.

- Bundle Name
- Bundle Version
- Bundle Identifier (Reverse DNS format)
- Executable Filename

The example shows an `Info.plist` file for the Firefox application, version 19.0.2. Other items in the file may be optional but still useful to an investigator to determine the purpose of a specific application. The “Application Category” is how the developer determines what type of application it is, while the “Document Types” and “URL Types” determine what files or URLs this program supports. The application may also state what versions of OS X it will run on, listing minimum and system versions and architectures types.

# Mach-O Executables

```
nibble:MacOS sledwards$ pwd  
/Applications/Firefox.app/Contents/MacOS  
nibble:MacOS sledwards$ file firefox  
firefox: Mach-O universal binary with 2 architectures  
firefox (for architecture x86_64):      Mach-O 64-bit executable x86_64  
firefox (for architecture i386):        Mach-O executable i386
```

00000	CA	FE	BA	BE	00	00	00	02
00053	00	00	00	00	00	00	00	00
00106	00	00	00	00	00	00	00	00
00159	00	00	00	00	00	00	00	00
00212	00	00	00	00	00	00	00	00
00265	00	00	00	00	00	00	00	00
00318	00	00	00	00	00	00	00	00
00371	00	00	00	00	00	00	00	00

© SANS,  
All Rights Reserved

## Mac Forensic Analysis

Mach-O (Mach Object) executable files are the main type of binary used for Mac applications.

Mach-O binaries can support multiple architectures. The top screenshot shows the file command used on the Firefox executable. These are called universal binaries, or “fat” binaries.

- x86\_64 executables can be used on Macs with the 64-bit Intel processor.
  - It is worth noting that some 64-bit executables can run on 32-bit kernels - visit the AppleInsider.com article listed in the references section for more information.
- i386 executables can be used on Macs with the with 32 and 64-bit Intel processors.

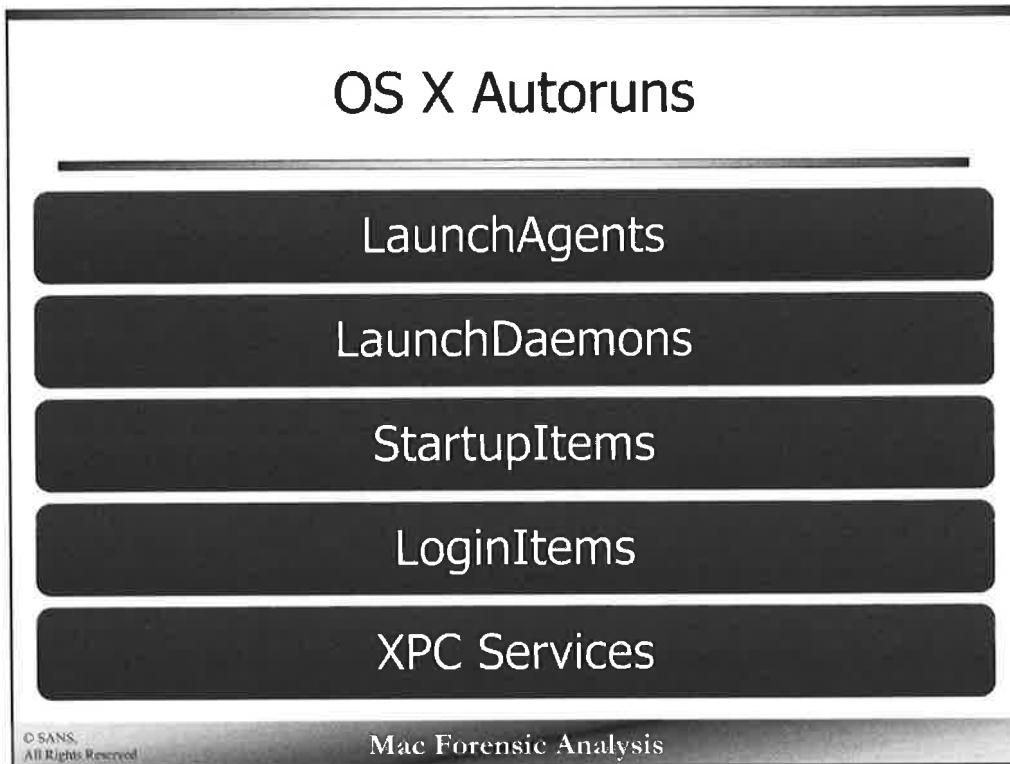
Fat binaries include multiple architectures such as PowerPC and Intel (these are sometimes called Universal binaries), or Intel 32-bit and 64-bit as shown in the screenshot above.

The Mach-O binary uses many signatures depending on the architecture:

- 0xCAFEBAE – Fat binary
- 0xFEEDFACE – 32-bit
- 0xFEEDFACE – 64-bit
- 0xCEFAEDFE – 32-bit, little-endian
- 0xCFFAEDFE – 64-bit, little-endian

### References:

- OS X Mach-O File Format Reference – Apple Developer Documentation  
<https://developer.apple.com/library/mac/#documentation/DeveloperTools/Conceptual/MachORuntime/Reference/Reference.html>
- [http://appleinsider.com/articles/08/10/28/road\\_to\\_mac\\_os\\_x\\_snow\\_leopard\\_64\\_bit\\_to\\_the\\_kernel.html](http://appleinsider.com/articles/08/10/28/road_to_mac_os_x_snow_leopard_64_bit_to_the_kernel.html)



Mac OS X systems have a variety of locations that applications can auto start from; however, not nearly as many as Windows systems.

While these locations can be used legitimately, malware authors also tend to use them. A python script called OS X Autoruns is available from Malicious Streams [<http://www.malicious-streams.com/Downloads/files/6bad73d5437c84b23fde9ebad97c7cff-6.html>] that enumerates many of these locations. It is worth noting it does not enumerate the XPC Services.

Some tools you might want to look at to find autoruns and other auditing information:

- Knockknock - [github.com/synack/knockknock](https://github.com/synack/knockknock)
- OSXAuditor - [github.com/jipegit/OSXAuditor](https://github.com/jipegit/OSXAuditor)
- checkout4mac - [code.google.com/p/checkout4mac/](https://code.google.com/p/checkout4mac/)
- pac4mac - [code.google.com/p/pac4mac/](https://code.google.com/p/pac4mac/)

#### References:

Mac OS X Malware Analysis - Joel Yonts, Malicious Streams

[http://www.malicious-streams.com//article/Mac OSX\\_Malware\\_Analysis.pdf](http://www.malicious-streams.com//article/Mac OSX_Malware_Analysis.pdf)

## Autoruns Launch Agents & Daemons

Introduced in 10.4 (w/launchd)

Property List File

Popular with Current Mac Malware

© SANS.  
All Rights Reserved

Mac Forensic Analysis

The preferred method of creating persistence is by using Launch Agents and Daemons. These methods were introduced in Mac OS X 10.4 with the introduction of `launchd`, the system agent and daemon launch manager.

The agents and daemons are implemented as information in a property list file. Most modern Mac malware uses these methods to keep their malicious files persistent across system shutdown and reboots.

Reference:

Daemons and Agents - TechNote 2083 (TN2083)

[http://developer.apple.com/library/mac/#technotes/tn2083/\\_index.html](http://developer.apple.com/library/mac/#technotes/tn2083/_index.html)

# Autoruns

## Launch Agents

### Launch Agent

- Background User Process
- Can access user home directory
- May have GUI (limited, if at all)

/System/Library/LaunchAgents/

/Library/LaunchAgents/

~/Library/LaunchAgents

© SANS.  
All Rights Reserved

Mac Forensic Analysis

Launch Agents are a background user process while daemons are background system processes.

These processes can access user home directories and have limited GUI interfaces while daemons cannot. The typical locations where LaunchAgents are found is the LaunchAgents directory located in the System Library, Local Library, and User Library directories.

Those agents launched from the /System/Library or /Library/ directories are launched for all users, while those located in the Users Library directory are available only to that user.

# Autoruns

## Launch Agent Examples [1]

```
com.apple.AOSNotification OSX.plist
com.apple.AddressBook.SourceSync.plist
com.apple.AddressBook.abd.plist
com.apple.AirPortBaseStationAgent.plist
com.apple.AppStoreUpdateAgent.plist
com.apple.AppleGraphicsWarning.plist
com.apple.BezelUI.plist
com.apple.CoreLocationAgent.plist
com.apple.DictionaryPanelHelper.plist
com.apple.DiskArbitrationAgent.plist
com.apple.Dock.plist
com.apple.FTCleanup.plist
com.apple.FileSyncAgent.PHD.plist
com.apple.FileSyncagent.IDisk.plist
com.apple.Finder.plist
com.apple.FontRegistryUIAgent.plist
com.apple.FontValidator.plist
com.apple.FontValidatorConduit.plist
com.apple.FontWorker.plist
com.apple.KerberosHelper.LKDCHelper.plist
com.apple.LaunchServices.tsboxd.plist
com.apple.NetworkDiagnostics.plist
com.apple.PCIELockCheck.plist
com.apple.PreferenceSyncAgent.plist
com.apple.PubSub.Agent.plist
com.apple.ReclaimSpaceAgent.plist
com.apple.RemoteDesktop.plist
com.apple.ReportCrash.Self.plist
com.apple.ReportCrash.plist
com.apple.ReportGPURestart.plist
com.apple.ReportPanic.plist
com.apple.ScreenReaderUIServer.plist
com.apple.ServiceManagement.LogInItems.plist
com.apple.SubmitDiagInfo.plist
com.apple.SystemUIServer.plist
com.apple.TMLaunchAgent.plist
com.apple.TrustEvaluationAgent.plist
com.apple.UserEventAgent-Aqua.plist
com.apple.UserEventAgent-LoginWindow.plist
com.apple.UserNotificationCenterAgent-LoginWindow.plist
com.apple.UserNotificationCenterAgent.plist
com.apple.VoiceOver.plist
com.apple.WebKit.PluginAgent.plist
com.apple.ZoomWindow.plist
com.apple.alf.useragent.plist
com.apple.hos.migrate.plist
com.apple.bluetoothHIDServer.plist
com.apple.btsz.plist
com.apple.cfnetwork.AuthBrokerAgent.plist
com.apple.cookie.plist
com.apple.coredata.externalrecordswriter.plist
com.apple.coreservices.apleid.authentication.plist
com.apple.coreservices.uingent.plist
com.apple.courseragent.plist
com.apple.cvmsCompAgent_i386.plist
com.apple.cvmsCompAgent_x86_64.plist
com.apple.distnoted.xpc.agent.plist
com.apple.familycontrols.useragent.plist
com.apple.findmymacmessenger.plist
com.apple.fontd.useragent.plist
com.apple.gssd-agent.plist
com.apple.helpd.plist
com.apple.icalPush.plist
com.apple.icloud.Theater.plist
com.apple.imageon.plist
com.apple.iok-launchagent.plist
com.apple.imtranscoderagent.plist
com.apple.imtransferagent.plist
```

© SANS,  
All Rights Reserved

Mac Forensic Analysis

The screenshot above shows a typical example of the LaunchAgents directory in /System/Library/.

Each launch agent property list is named in the reverse-DNS format.

com.apple.AOSNotification OSX.plist	com.apple.SystemUIServer.plist
com.apple.AddressBook.SourceSync.plist	com.apple.ThLaunchAgent.plist
com.apple.AddressBook.abd.plist	com.apple.TrustEvaluationAgent.plist
com.apple.AirPortBaseStationAgent.plist	com.apple.UserEventAgent-Aqua.plist
com.apple.AppStoreUpdateAgent.plist	com.apple.UserEventAgent-LoginWindow.plist
com.apple.AppleGraphicsWarning.plist	com.apple.UserNotificationCenterAgent-LoginWindow.plist
com.apple.BezelUI.plist	com.apple.UserNotificationCenterAgent.plist
com.apple.CoreLocationAgent.plist	com.apple.VoiceOver.plist
com.apple.DictionaryPanelHelper.plist	com.apple.WebKit.PluginAgent.plist
com.apple.DiskArbitrationAgent.plist	com.apple.ZoomWindow.plist
com.apple.Dock.plist	com.apple.alf.useragent.plist
com.apple.FTCleanup.plist	com.apple.abs.migrate.plist
com.appleFileSyncAgent.PHD.plist	com.apple.bluetoothUIServer.plist
com.apple.Finder.plist	com.apple.cftnetwork.AuthBrokerAgent.plist
com.apple.FontRegistryUIAgent.plist	com.apple.cookie.d.plist
com.apple.FontValidator.plist	com.apple.coredata.externalrecordswriter.plist
com.apple.FontValidatorConduit.plist	com.apple.coreervices.appleid.authentication.plist
com.apple.FontWorker.plist	com.apple.coreservices.viaagent.plist
com.apple.KerberosHelper.LKDCHelper.plist	com.apple.csuseragent.plist
com.apple.LaunchServices.lsboxd.plist	com.apple.cvmsCompAgent_i386.plist
com.apple.NetworkDiagnostics.plist	com.apple.cvmsCompAgent_x86_64.plist
com.apple.PCIESlotCheck.plist	com.apple.distnoted.xpc.agent.plist
com.apple.PreferenceSyncAgent.plist	com.apple.familycontrols.useragent.plist
com.apple.pubSub.Agent.plist	com.apple.findmymacmessenger.plist
com.apple.ReclaimSpaceAgent.plist	com.apple.fontd.useragent.plist
com.apple.RemoteDesktop.plist	com.apple.gssd-agent.plist
com.apple.ReportCrash.Self.plist	com.apple.help.plist
com.apple.ReportCrash.plist	com.apple.icalpush.plist
com.apple.ReportGPURestart.plist	com.apple.ichat.Theater.plist
com.apple.ReportPanic.plist	com.apple.imaget.plist
com.apple.ScreenReaderUIServer.plist	com.apple.inklaunchagent.plist
com.apple.ServiceManagement.LoginItems.plist	com.apple.intranscoderagent.plist
com.apple.SubmitDiagInfo.plist	com.apple.intransferagent.plist

# Autoruns Launch Agent Examples [2]

The image shows two screenshots of launchd.plist files. The top screenshot displays a plist for the 'imagent' process. It includes keys for ProgramArguments (containing Item 0 with a String value of '/System/Library/PrivateFrameworks/IMCore.framework/imagent.app/Contents/MacOS/imagent'), KeepAlive (with a subkey SuccessfulExit set to Boolean NO), Label (set to com.apple.imagent), MachServices (containing com.apple.imagent.desktop.auth with a subkey ResetAtClose set to Boolean YES), and EnvironmentVariables (containing NSRunningFromLaunchd with a String value of 1). The bottom screenshot shows a plist for the 'org.openbsd.ssh-agent' process. It includes a Label key (set to org.openbsd.ssh-agent) and a ProgramArguments key (containing Item 0 with a String value of '/usr/bin/ssh-agent' and Item 1 with a String value of '-l'). Other keys shown include ServiceIPC (Boolean YES), Sockets (Dictionary containing Listeners with a SecureSocketWithKey String value of SSH\_AUTH\_SOCK and EnableTransactions Boolean value of YES).

Three examples of launch agents are shown above. Each launch agent contains two basic keys:

- Program or ProgramArguments – File path of the program to launch
- Label – Reverse DNS name for the process. This string should be unique across launch agents.

Other keys may be used to customize the launch agent. Many of these keys are explained in the man page for `launchd.plist`.

The top example shows a launch agent for the “imagent” process. The `KeepAlive` key contains the subkey “`SucessfulExit`”. According to TechNote 2083 this means to “run on demand as long as you exit successfully.” This example also shows the `EnvironmentVariables` which is used to specify environment variables.

The middle screenshot shows a good example of the `ProgramArguments` key. This key can have multiple items; the first is always the program to execute, while the others are program arguments. This example runs the command `/usr/bin/ssh-agent` with the `-l` flag.

## References:

launchd.plist Man Page:

<https://developer.apple.com/library/mac/documentation/Darwin/Reference/ManPages/man5/launchd.plist.5.html>

Key	Type	Value
▼ ProgramArguments	Array	{1 item}
Item 0	String	/System/Library/PrivateFrameworks/IMCore.framework/IMagent.app/Contents/MacOS/IMagent
▼ KeepAlive	Diction...	(1 item)
SuccessfulExit	Boolean	NO
Label	String	com.apple.IMagent
▼ MachServices	Diction...	(1 item)
▼ com.apple.imagent.desktop.auth	Diction...	(1 item)
ResetAtClose	Boolean	YES
▼ EnvironmentVariables	Diction...	(1 item)
NSRunningFromLaunchd	String	1

Key	Type	Value
Label	String	org.openbsd.ssh-agent
▼ ProgramArguments	Array	(2 items)
Item 0	String	/usr/bin/ssh-agent
Item 1	String	-l
ServiceIPC	Boolean	YES
▼ Sockets	Diction...	(1 item)
▼ Listeners	Diction...	(1 item)
SecureSocketWithKey	String	SSH_AUTH_SOCK
EnableTransactions	Boolean	YES

## Autoruns

## Launch Daemons

### Launch Daemon

- Background System Process

/System/Library/LaunchDaemons

/Library/LaunchDaemons

© SANS.  
All Rights Reserved

Mac Forensic Analysis

Launch Daemons are a background system process while agents are background user processes.

The typical locations where Launch Daemons are found is the LaunchDaemons directory located in the System Library and Local Library directories.

**Autoruns**  
**Launch Daemon Example**

The screenshot shows the output of the daily launch daemon and its configuration plist.

**Output Log:**

```

Fri Apr 13 03:15:00 EDT 2012
Removing old temporary files:
Cleaning out old system announcements:
Removing stale files from /var/rwho:
Removing scratch fax files:
Disk status:
Filesystem      Size  Used Avail Capacity  Mounted on
/dev/disk1      698Gi 431Gi 267Gi   62%    /
localhost:/NW-3H1UZ/FYg0rbEggLJ 698Gi 698Gi 0B  100%  /tmp
Network interface status:
Name  MTU  Network Address          Ipkts Ierrs
lo0  16384 <Link#1>                38376  0
lo0  16384 localhost  fe80:1:1:1  38375  - 
lo0  16384 127  localhost          38375  - 
lo0  16384 ip6-localhost ::1       38376  - 
gif0# 1288 <Link#2>                0     0
stf0# 1288 <Link#3>                0     0
en0  1500  <Link#4>  c4:3c:03:09:ca:fd  0     0
en1  1500  <Link#5>  00:22:c4:f8:e6:5f  37611865  0  185742931  0  0
en1  1500  bpf_local  f980:5:9227:ce4f  37611865  - 185742931  - 
eni  1500  192.168.1.1  p1c              37611865  - 185742031  - 
fu0  6078 <Link#6>  00:06:BB:ff:fe:dd  5d:00  0
p2p0  2304 <Link#7>  02:27:c4:f8:e6:5f  0     0
vmnet 1500 <Link#8>  00:59:56:c0:00:01  0     0
vmnet 1500 172.16.73.24  172.16.73.1  0     0
vmnet 1500 <Link#9>  00:50:56:c0:00:08  0     0
vmnet 1500 192.160.158  192.160.158.1  0     0
Local system status:
3:15 up 3 days, 8:06, 4 users, load averages: 0.55 0.57 0.56
-- End of daily output --

```

**Configuration Plist:**

Key	Type	Value
Label	String	com.apple.periodic-daily
ProgramArguments	Array	{2 items}
Item 0	String	/usr/sbin/periodic
Item 1	String	daily
LowPriorityIO	Boolean	YES
Nice	Number	1
StartCalendarInterval	Dictionary	{2 items}
Hour	Number	3
Minute	Number	15
AbandonProcessGroup	Boolean	YES

**Watermark:** © SANS. All Rights Reserved.

Mac Forensic Analysis

Good examples of launch daemons are the Unix periodic maintenance scripts; daily, weekly, and monthly.

The screenshot shows the output from the daily script in the `daily.out` file. The output in the log has a timestamp of April 13<sup>th</sup> at 03:15:00 EDT 2012. The coordinating Launch Daemon, `com.apple.periodic-daily.plist` contains many of the same keys as the LaunchAgents. The `ProgramArguments` key contains the path to the daily script with the argument “`daily`”.

The property list also contains the key `StartCalendarInterval` which starts the daemon at a specified time. This key shows the hour and minute of the day that this script is intended to run, 03:15.

```

Fri Apr 13 03:15:00 EDT 2012
Removing old temporary files:
Cleaning out old system announcements:
Removing stale files from /var/rwho:
Removing scratch fax files

Disk status:
Filesystem          Size  Used Avail Capacity Mounted on
/dev/disk1          698Gi 431Gi 267Gi   62%   /
localhost:/YNU-3NIW2FYxg8rbEqggLJ 698Gi 698Gi 0Bi    100%   /Volumes/MobileBackups

Network interface status:
Name  Mtu Network      Address           Ipkts  Ierrs     Opkts  Oerrs     Coll
lo0   16384 <Link#1>          38376   0       38376   0       0
lo0   16384 localhost    fe80:1::1        38376   -       38376   -       -
lo0   16384 127          localhost          38376   -       38376   -       -
lo0   16384 ip6-localho  ::1        38376   -       38376   -       -
gif0* 1280  <Link#2>          0       0       0       0       0
stf0* 1280  <Link#3>          0       0       0       0       0
en0   1500  <Link#4>  c4:2c:03:09:ca:fd  0       0       0       0       0
en1   1500  <Link#5>  90:27:e4:f8:e6:5f  37611065 0       105742931 0       0
en1   1500  bit.local    fe80:5::9227:e4ff 37611065 -       105742931 -       -
en1   1500  192.168.1    bit               37611065 -       105742931 -       -
fw0   4078  <Link#6>  e8:06:88:ff:fe:d5:08  0       0       0       0       0
p2p0  2304  <Link#7>  02:27:e4:f8:e6:5f  0       0       0       0       0
vmnet 1500  <Link#8>  00:50:56:c0:00:01  0       0       0       0       0
vmnet 1500  172.16.73/24 172.16.73.1     0       -       0       -       -
vmnet 1500  <Link#9>  00:50:56:c0:00:08  0       0       0       0       0
vmnet 1500  192.168.158 192.168.158.1    0       -       0       -       -

Local system status:
3:15 up 3 days, 8:06, 4 users, load averages: 0.55 0.57 0.56

-- End of daily output --

```

Key	Type	Value
Label	String	com.apple.periodic-daily
▼ ProgramArguments	Array	(2 items)
Item 0	String	/usr/sbin/periodic
Item 1	String	daily
LowPriorityIO	Boolean	YES
Nice	Number	1
▼ StartCalendarInterval	Dictionary	(2 items)
Hour	Number	3
Minute	Number	15
AbandonProcessGroup	Boolean	YES

# Autoruns

## Login Items

Launched when user logs into system via GUI

### Locations

- ~/Library/Preferences/com.apple.loginitems.plist
- <application>.app/Contents/Library/LoginItems/

### Global Login Item

- /Library/Preferences/com.apple.loginwindow.plist
- AutoLaunchedApplicationDictionary Key

© SANS,  
All Rights Reserved

Mac Forensic Analysis

Previously mentioned in Section 2, login items are launched when a user logs into the system using the GUI. Login items are similar to the Microsoft registry key HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Run. These are usually small programs or helper applications.

Each user has their own login items listed in the com.apple.loginitems.plist file located in their /Library/Preferences/ directory. The login item application is usually found within an application bundle's Library directory.

While rare, global login items may be listed in the com.apple.loginwindow.plist file in the /Library/Preferences/ directory under the AutoLaunchedApplicationDictionary key.

## Autoruns XPC Services

Privilege Separation & Stability

Sandboxed Environment

Runs in user context

Services a single application

Location:

- Application Bundle: /Contents/XPCServices/
- /System/Library/XPCServices/

© SANS.  
All Rights Reserved

Mac Forensic Analysis

XPC (Interprocess Communication) services are used to control the stability of a system. These services will only service one application. If the service crashes, only the application will crash rather than the whole system. These services run in the user context and are located in various XPCServices directories.

### References:

Creating XPC Services - Apple Developer Documentation

<http://developer.apple.com/library/mac/#documentation/MacOSX/Conceptual/BPSystemStartup/Chapters/CreatingXPCServices.html>

# Autoruns

## XPC Service Example

Key	Type	Value
BuildMachineOSBuild	String	11D17a
Localization native development region	String	English
Executable file	String	com.apple.qtkitserver
Bundle identifier	String	com.apple.qtkitserver
InfoDictionary version	String	6.0
Bundle name	String	com.apple.qtkitserver
Bundle OS Type code	String	XPC!
Bundle versions string, short	String	1.0
Bundle creator OS Type code	String	????
Bundle version	String	1
DTCompiler	String	
DTPlatformBuild	String	11D17a
DTPlatformVersion	String	GM
DTSDKBuild	String	11D17a
DTSDKName	String	
DTXcode	String	0410
DTXcodeBuild	String	11D17a
Application is agent (element)	Boolean	YES
▼ XPCService	Dictionary	(2 items)
▼ EnvironmentVariables	Dictionary	(1 item)
MallocCorruptionAbort	String	1
ServiceType	String	Application

© SANS,  
All Rights Reserved.

Mac Forensic Analysis

The screenshot shows an example of an XPC service property list for the qtkitserver located in /System/Library/XPCServices/com.apple.qtkitserver.xpc/Contents/Info.plist. The property list contains the bundle type code of “XPC!” and a key labeled XPC service that will contain many of the same keys used in launch agents and daemons.

Key	Type	Value
BuildMachineOSBuild	String	11D17a
Localization native development region	String	English
Executable file	String	com.apple.qtkitserver
Bundle identifier	String	com.apple.qtkitserver
InfoDictionary version	String	6.0
Bundle name	String	com.apple.qtkitserver
Bundle OS Type code	String	XPC!
Bundle versions string, short	String	1.0
Bundle creator OS Type code	String	???
Bundle version	String	1
DTCompiler	String	
DTPlatformBuild	String	11D17a
DTPlatformVersion	String	GM
DTSDKBuild	String	11D17a
DTSDKName	String	
DTXcode	String	0410
DTXcodeBuild	String	11D17a
Application is agent (UIElement)	Boolean	YES
▼ XPCService	Diction...	(2 items)
▼ EnvironmentVariables	Diction...	(1 item)
MallocCorruptionAbort	String	1
ServiceType	String	Application

# Autoruns

## Deprecated Methods

<b>/etc/crontab or /var/at/tabs</b>	<ul style="list-style-type: none"><li>• Still supported, not recommended</li></ul>
<b>Login/Logout Hooks</b> Deprecated as of 10.6	<ul style="list-style-type: none"><li>• Run as root</li><li>• com.apple.loginwindow.plist – LoginHook/LogoutHook Keys</li></ul>
<b>Startup Item</b> Deprecated as of 10.4	<ul style="list-style-type: none"><li>• /Library/StartupItems</li><li>• /System/Library/StartupItems</li></ul>
<b>mach_init Daemon</b> Deprecated as of 10.5	<ul style="list-style-type: none"><li>• Property List File in /etc/mach_init.d</li></ul>
<b>mach_init Agent</b> Deprecated as of 10.5	<ul style="list-style-type: none"><li>• Property List file in /etc/mach_init_per_user.d/</li></ul>
<b>inetd/xinetd Daemon</b> Deprecated as of 10.4	<ul style="list-style-type: none"><li>• Line in /etc/inetd.conf</li><li>• Config file in /etc/xinetd.d/</li></ul>
<b>System Login Item</b> Deprecated as of 10.5	<ul style="list-style-type: none"><li>• Replaced with pre-login launchd agent.</li></ul>

© SANS.  
All Rights Reserved

Mac Forensic Analysis

There are quite a few deprecated methods for starting applications on Mac OS X systems.

The Unix standby `/etc/crontab` can still be implemented, however the previously mentioned methods are preferred. The

Login and logout hooks are located in `com.apple.loginwindow.plist` property list as keys of the same name.

Startup Items has been reinvented as Login Items; however, they may still be found in various `StartupItems` directories.

`mach_init` agents and daemons were once located in the `/etc` directory as property lists.

`inetd/xinetd` daemons may be found in the `/etc` directory in the `inetd.conf` file or in the `xinetd.d/` directory.

The System Login Items were replaced with pre-login launch agents, however recall that it is still possible to configure a global login item in the `com.apple.loginwindow.plist` file in the `/Library/Preferences` directory.

## Firewall Software

Application Level Firewall  
(ALF)

10.6 - IP Firewall (ipfw)  
10.7+ - Packet Filter Firewall (pfctl)

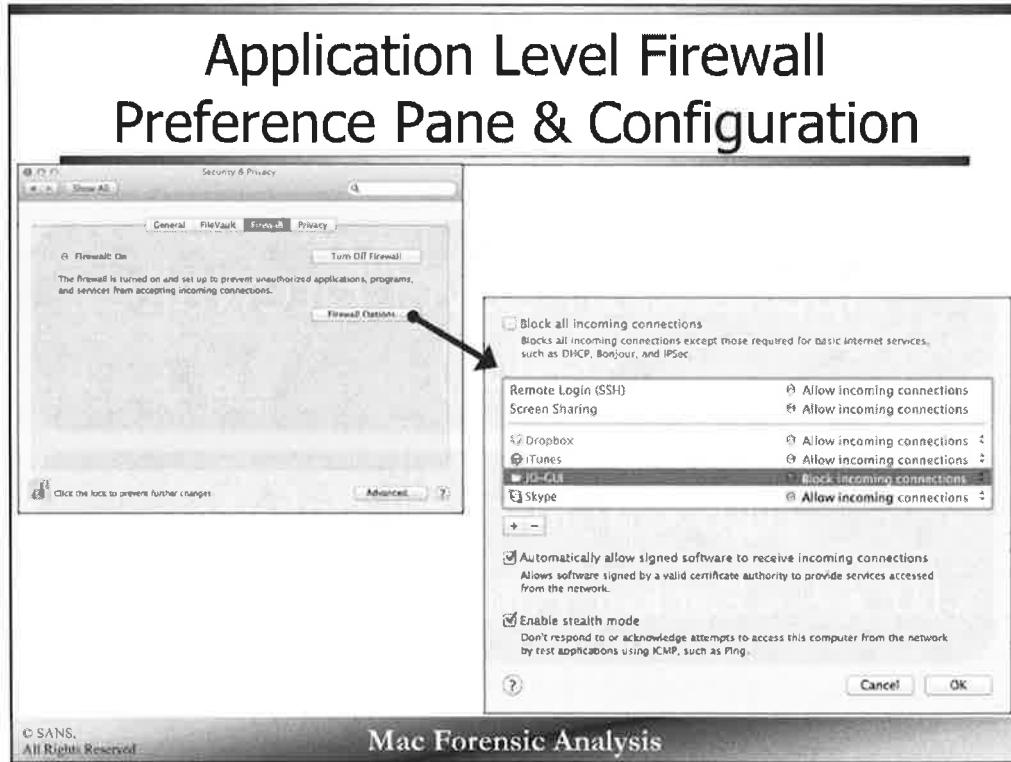
© SANS.  
All Rights Reserved

Mac Forensic Analysis

Mac OS X systems come with two firewalls. An application level firewall (host-based application firewall) and an IP/Packet Filtering firewall.

A non-savvy user may use the application level firewall via the GUI, while a more technical user may use the IP firewall via the `ipfw` or `pfctl` command.

If used, the Packet Filtering Firewall configuration can be found in `/etc/pf.conf`.



The default firewall program on Mac OS X is the application level firewall (ALF) which can be configured in the Security & Privacy preferences panel under System Preferences. The firewall is not turned on by default on a newly installed system.

The screenshot on the left shows the Firewall tab on the Security & Privacy preferences pane. The screenshot on right shows the Firewall Options which show the remote access options and specific application configurations.

# Application Level Firewall – Configuration /Library/Preferences/com.apple.alf.plist

Path	Type	Value
Root	Dictionary	{10 items}
allowsignedenabled	Number	1
exceptions	Array	{5 items}
globalstate	Number	1
stealthenabled	Number	0
firewall	Dictionary	{9 items}
version	String	1.0x24
loggingenabled	Number	1
firewallunload	Number	0
applications	Array	{4 items}
Item 0	Dictionary	{4 items}
bundleid	String	com.getdropbox.dropbox
reqdata	Data	<fafe0c00 000000c4 00000000>
alias	Data	<3c3f786d 6c207665 7273>
state	Number	0
Item 1	Dictionary	{4 items}
bundleid	String	com.apple.iTunes
reqdata	Data	<fafe0c00 0000002c 00000000>
alias	Data	<3c3f786d 6c207665 7273>
state	Number	0
Item 2	Dictionary	{4 items}
bundleid	String	jd.jd-gui
reqdata	Data	<fafe0c00 00000028 00000000>
alias	Data	<3c3f786d 6c207665 7273>
state	Number	2
Item 3	Dictionary	{4 items}
bundleid	String	com.skype.skype
reqdata	Data	<fafe0c00 000000bc 00000000>
alias	Data	<3c3f786d 6c207665 7273>
state	Number	0
> explicitauths	Array	{2 items}

© SANS,  
All Rights Reserved

Mac Forensic Analysis

The com.apple.alf.plist property file in the /Library/Preferences directory contains the configuration for the application level firewall.

The globalstate key determines if the firewall is enabled.

- 1 = Firewall Enabled
- 0 = Firewall Disabled

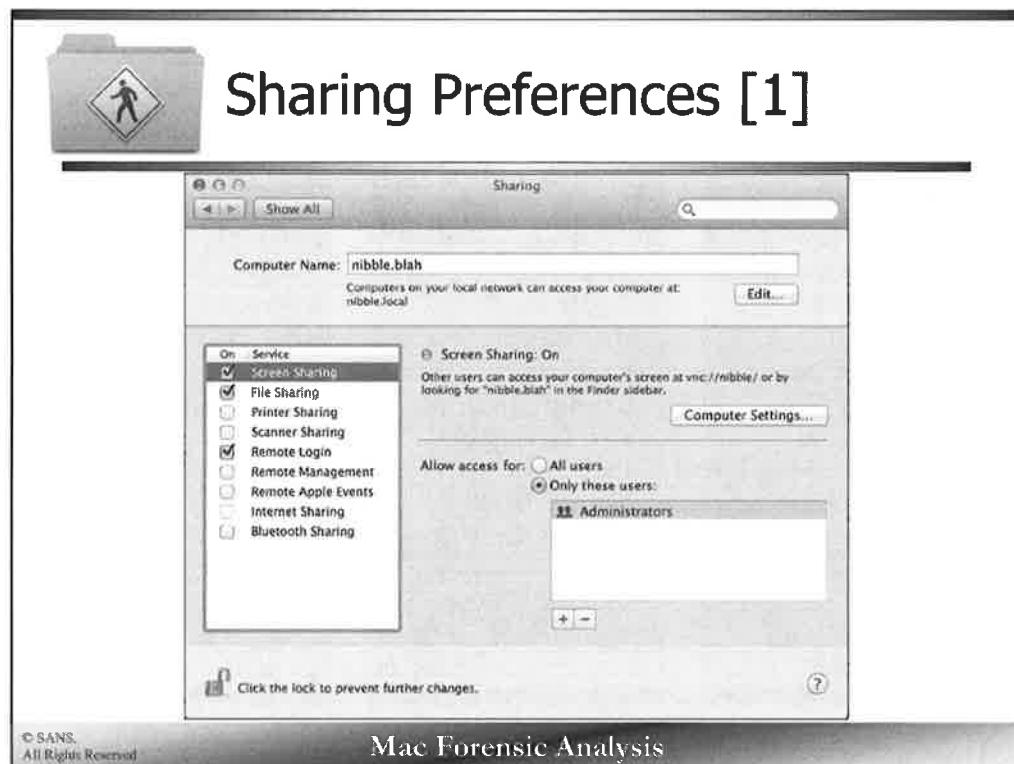
In the previous screenshot of the application firewall, there are two checkboxes the users can configure. The following keys hold this configuration:

- The allowsignedenabled key determines if the checkbox to allow signed software to receive incoming connections is checked (1 = checked).
- The stealthenabled key determines if the stealth enabled checkbox was checked (1 = checked). Enabling stealth mode ignores acknowledgement from network utilities like ping.

The applications key lists the applications configured in the firewall as shown in the screenshot on the previous slide. Each application item contains the applications bundle ID, alias data and the state. A state of “0” allows incoming connections, while a state of “2” blocks incoming connections to the application.

The remote access options can be found in the Sharing preferences pane.

▼ Root	Dictionary	(10 items)
allowsignedenabled	Number	1
► exceptions	Array	(5 items)
globalstate	Number	1
stealthenabled	Number	0
► firewall	Dictionary	(9 items)
version	String	1.0a24
loggingenabled	Number	1
firewallunload	Number	0
▼ applications	Array	(4 items)
▼ Item 0	Dictionary	(4 items)
bundleid	String	com.getdropbox.dropbox
reqdata	Data	<fade0c00 000000c4 00000000
alias	Data	<3c3f786d 6c207665 72730000
state	Number	0
▼ Item 1	Dictionary	(4 items)
bundleid	String	com.apple.iTunes
reqdata	Data	<fade0c00 0000002c 00000000
alias	Data	<3c3f786d 6c207665 72730000
state	Number	0
▼ Item 2	Dictionary	(4 items)
bundleid	String	jd.jd-gui
reqdata	Data	<fade0c00 00000028 00000000
alias	Data	<3c3f786d 6c207665 72730000
state	Number	2
▼ Item 3	Dictionary	(4 items)
bundleid	String	com.skype.skype
reqdata	Data	<fade0c00 000000bc 00000000
alias	Data	<3c3f786d 6c207665 72730000
state	Number	0
► explicitauths	Array	(7 items)



The Sharing preferences pane shown in the screenshot above contains the items that can be shared from the system. By default none of these are enabled.

## Sharing Preferences [2]

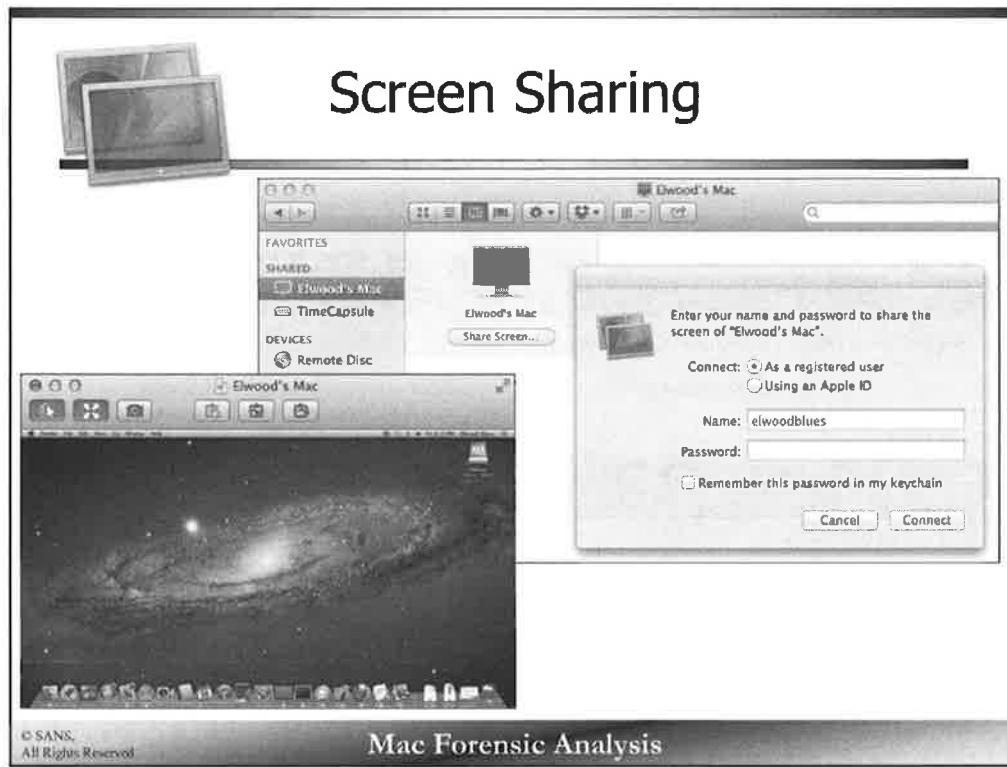
10.6 & 10.7	10.8	10.9+
<ul style="list-style-type: none"><li>• DVD or CD Sharing</li><li>• Screen Sharing</li><li>• File Sharing</li><li>• Printer Sharing</li><li>• Scanner Sharing</li><li>• <b>Web Sharing</b></li><li>• Remote Login</li><li>• Remote Management</li><li>• Remote Apple Events</li><li>• <b>Xgrid Sharing</b></li><li>• Internet Sharing</li></ul>	<ul style="list-style-type: none"><li>• DVD or CD Sharing</li><li>• Screen Sharing</li><li>• File Sharing</li><li>• Printer Sharing</li><li>• <b>Scanner Sharing</b></li><li>• Remote Login</li><li>• Remote Management</li><li>• Remote Apple Events</li><li>• Internet Sharing</li><li>• Bluetooth Sharing</li></ul>	<ul style="list-style-type: none"><li>• DVD or CD Sharing</li><li>• Screen Sharing</li><li>• File Sharing</li><li>• Printer Sharing</li><li>• <b>Scanner Sharing</b></li><li>• Remote Login</li><li>• Remote Management</li><li>• Remote Apple Events</li><li>• Internet Sharing</li><li>• Bluetooth Sharing</li></ul>

© SANS.  
All Rights Reserved

Mac Forensic Analysis

The shared preferences options have kept changing from each OS X version. The changes bolded are no longer available on the next OS X version due to discontinued hardware and software (DVD/CD drive, Xgrid) or just limiting user issues (Web Sharing can still be enabled via the command line.)

It is worth noting the DVD or CD Sharing option depends on the hardware of the system. If the system does not have a CD/DVD drive to share, the option will not be listed.



The Screen Sharing application, located in /System/Library/CoreServices/ rather than /Applications, is used to connect to other systems using the VNC protocol.

# Sharing Preferences - Screen Sharing Property Lists

/private/var/db/launchd.db/com.apple.launchd/overrides.plist

```
<key>com.apple.screensharing</key>
<dict>
    <key>Disabled</key>
    <false/>
</dict>
```

```
<key>com.apple.screensharing</key>
<dict>
    <key>Disabled</key>
    <true/>
</dict>
```

/Library/Preferences/com.apple.RemoteManagement.plist

▼ Root	Dictionary	(2 items)
VNCLegacyConnectionsEnabled	Boolean	YES
ScreenSharingReqPermEnabled	Boolean	NO

© SANS.  
All Rights Reserved

Mac Forensic Analysis

The overrides.plist file contains many configuration keys for various sharing settings. The com.apple.screensharing key contains the Disabled subkey. The value for this subkey can either be true or false. If screen sharing is enabled, the disabled key is “false”, while if screen sharing is disabled, the key will be “true”.

The com.apple.RemoteManagement.plist is created in the /Library/Preferences/ directory when the Screen Sharing or Remote Management options are selected in the Sharing preferences pane.

10.10 does not appear to use the overrides.plist, more research is need to determine where this information is stored.

# Sharing Preferences - Screen Sharing

## /Library/Preferences/com.apple.VNCSettings.txt

```
nibble:Preferences sledwards$ sudo cat com.apple.VNCSettings.txt
6755221DBA9AF6E2FF1C39567390ADCA
nibble:Preferences sledwards$ sudo cat com.apple.VNCSettings.txt | perl -wne 'BEGIN { @k = unpack "C*", pack "H*", "1734516E8BA8C5E2FF1C39567390ADCA"; chomp; @p = unpack "C*", pack "H*", $_; foreach (@k) { printf "%c", $_ ^ (shift @p || 0) }; print "\n"'
```

© SANS,  
All Rights Reserved

Mac Forensic Analysis

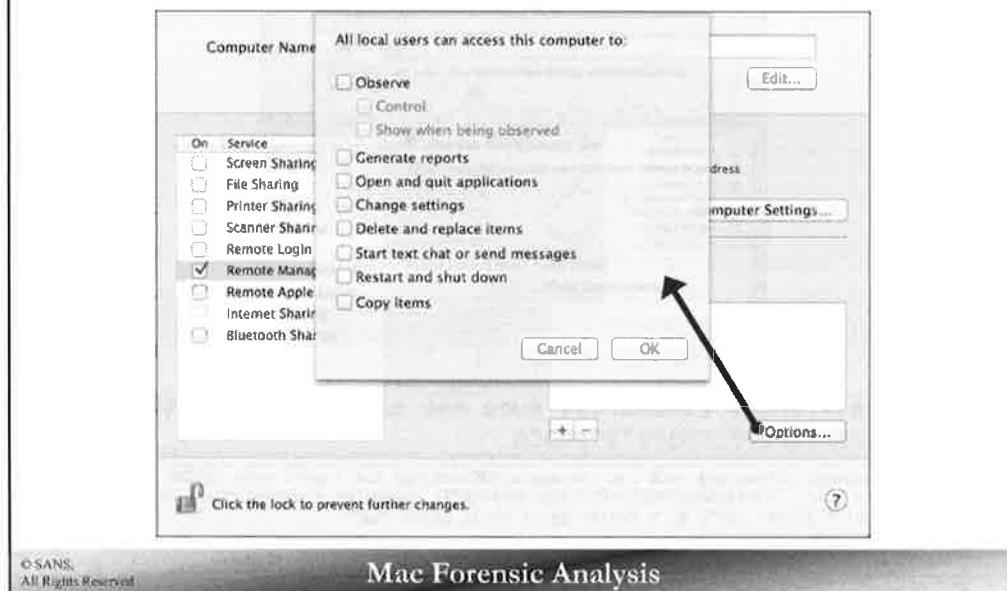
The com.apple.VNCSettings.txt text file contains the XOR'ed “encrypted” password to access the system via a VNC viewer client. We can use the Perl script created by Ben Low (<http://lists.apple.com/archives/remote-desktop/2005/Oct/msg00026.html>) to “decrypt” the VNC password using the XOR key “1734516E8BA8C5E2FF1C39567390ADCA”.

```
cat com.apple.VNCSettings.txt | perl -wne 'BEGIN { @k = unpack "C*", pack "H*", "1734516E8BA8C5E2FF1C39567390ADCA"; chomp; @p = unpack "C*", pack "H*", $_; foreach (@k) { printf "%c", $_ ^ (shift @p || 0) }; print "\n"'
```

### References:

- VNC Password - command line configuration of VNC  
<http://lists.apple.com/archives/remote-desktop/2005/Oct/msg00026.html>

# Sharing Preferences Remote Management



Remote Management is inclusive of Screen Sharing. When the Remote Management box is checked, the Screen Sharing box will be grayed out. Shown in the screenshot above, when Remote Management is checked, a box will pop up allowing the user to choose the functions available to remotely manage.

# Sharing Preferences – Remote Management

/Library/Preferences/com.apple.RemoteManagement.plist

The screenshot shows the 'Sharing Preferences – Remote Management' pane. At the top, it displays the path '/Library/Preferences/com.apple.RemoteManagement.plist'. Below this, a table titled 'Root' lists several keys:

	Dictionary	(5 items)
ARD_AllLocalUsersPrvs	Number	239
VNCLegacyConnectionsEnabled	Boolean	YES
LoadRemoteManagementMenuExtra	Boolean	NO
ARD_AllLocalUsers	Boolean	YES
ScreenSharingReqPermEnabled	Boolean	NO

A callout arrow points from the 'ARD\_AllLocalUserPrvs' key in the dictionary to a 'Control' checkbox in a secondary window titled 'All local users can access this computer to:'.

**All local users can access this computer to:**

- Observe
- Control
- Show when being observed
- Generate reports
- Open and quit applications
- Change settings
- Delete and replace items
- Start text chat or send messages
- Restart and shut down
- Copy items

Cancel OK

O SANS,  
All Rights Reserved

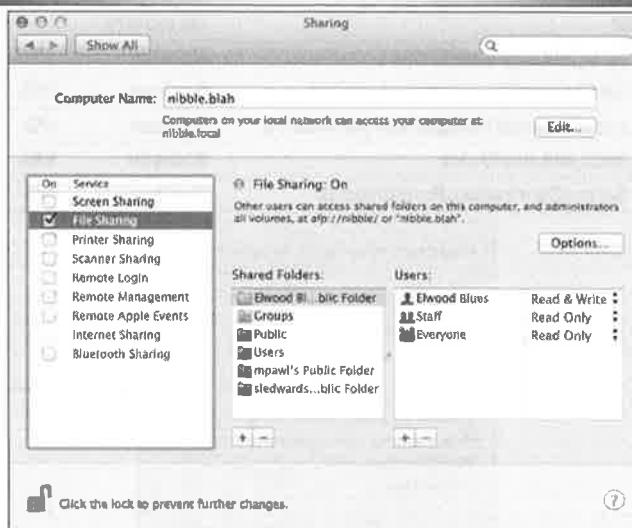
Mac Forensic Analysis

The `com.apple.RemoteManagement.plist` is created in the `/Library/Preferences/` directory when the Screen Sharing or Remote Management options are selected in the Sharing preferences pane.

Selecting Remote Management sharing creates new Apple Remote Desktop (ARD) keys. The `ARD_AllLocalUserPrvs` key contains a number that correlates to the specific options selected in the Remote Management Options pop-up window.

# Sharing Preferences

## File Sharing



© SANS.  
All Rights Reserved

Mac Forensic Analysis

File Sharing allows users to share files on the network or with other users on the system.

# Sharing Preferences - File Sharing

/Library/Preferences/com.apple.filessharingui.plist

The screenshot shows the Mac OS X Sharing preferences window. Under 'Computer Name', 'File Sharing' is selected. In the main pane, 'Share files and folders using AFP' and 'Share files and folders using SMB (Windows)' are checked. The SMB sharing section includes a note about entering a Windows user password. Below this, an account list shows 'wood blues' and 'oompa'. To the right, sharing permissions are set to 'Read & Write' for 'Administrators' and 'Read Only' for 'Everyone'. An arrow points from the 'Edit...' button in the preferences window to the 'Options...' button in the Mac Forensic Analysis tool. The tool's interface shows a table with two items:

Root	Dictionary	(2 items)
AFPEnabled	Boolean	YES
SMBEnabled	Boolean	YES

© SANS, All Rights Reserved

Mac Forensic Analysis

The file sharing preferences file, com.apple.filessharingui.plist in the /Library/Preferences/ directory contains two keys each showing a YES or NO value depending if the sharing service is enabled.

- AFPEnabled – (Apple Filing Protocol) If value is YES, AFP sharing is enabled
- SMBEnabled – (Server Message Block) If value is YES, SMB sharing is enabled

On 10.9 systems, this property list does not appear to be used – instead it only uses the overrides.plist described on the next slide.

## Sharing Preferences - File Sharing

/private/var/db/launchd.db/com.apple.launchd/overrides.plist

Root	Dictionary	(2 items)
AFPEnabled	Boolean	YES
SMBEnabled	Boolean	YES

```
<key>com.apple.AppleFileServer</key>
<dict>
    <key>Disabled</key>
    <false/>
</dict>

<key>com.apple.smbd</key>
<dict>
    <key>Disabled</key>
    <false/>
</dict>
```

© SANS,  
All Rights Reserved

Mac Forensic Analysis

The overrides.plist property list located in the /private/var/db/launchd.db/com.apple.launchd/ directory also contains keys that show if AFP or SMB sharing are enabled. In the screenshot above the two “Disabled” keys show false, meaning they are both enabled.

# Sharing Preferences - File Sharing

/Library/Preferences/SystemConfiguration/com.apple.smb.server.plist

Root	Dictionary	(5 items)
AllowGuestAccess	Boolean	NO
DOSCodePage	String	437
LocalKerberosRealm	String	LKDC:SHA1.15035119714DE1
NetBIOSName	String	nibble
ServerDescription	String	nibble.blah

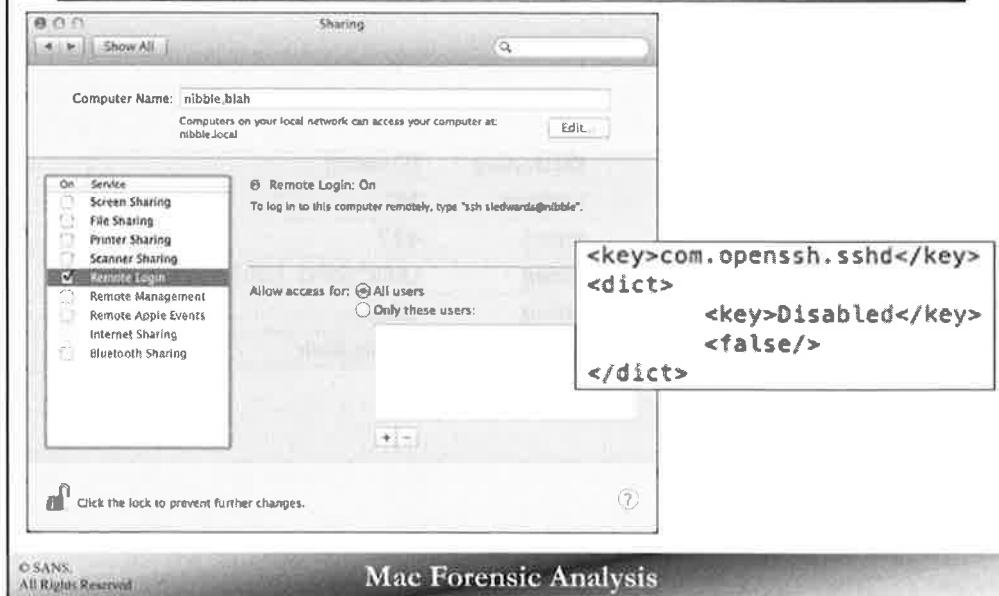
© SANS,  
All Rights Reserved

Mac Forensic Analysis

The com.apple.smb.server.plist file located in  
/Library/Preferences/SystemConfiguration/ contains information specific to SMB sharing.  
The property list contains the NetBIOS name for the local system.

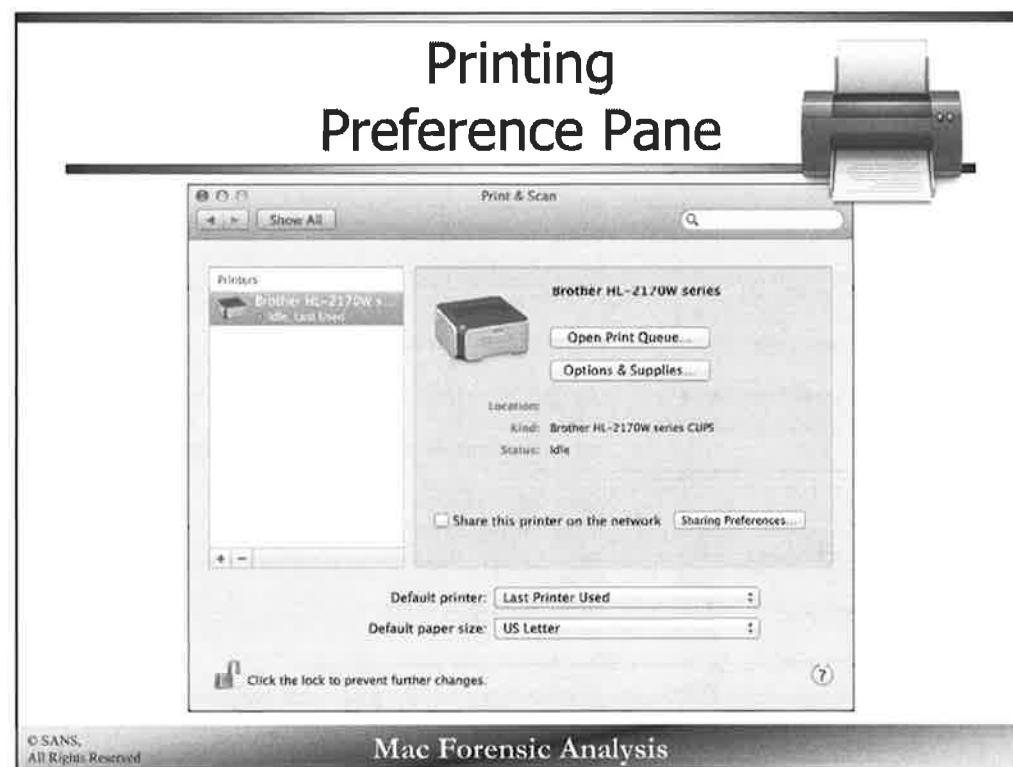
## Sharing Preferences - Remote Login (SSH)

/private/var/db/launchd.db/com.apple.launchd/overrides.plist



Remote Login allows a user to remotely login to the system via the SSH (Secure Shell) protocol. The overrides.plist file located in the /private/var/db/launchd.db/com.apple.launchd/ directory contains the key com.openssh.sshd. When this key is false, SSH is enabled.

# Printing Preference Pane



The Printing preference pane contains printer settings for the system. The screenshot shows one printer, a Brother HL-2170W, setup on the system.

Printing			
/Library/Preferences/org.cups.printers.plist			
▼ Root	Array	(2 items)	
▼ Item 0	Dictionary	(9 items)	
printer-name	String	Brother_HL_2170W_series	
printer-info	String	Brother HL-2170W series	
printer-is-accepting-jobs	Boolean	YES	
printer-location	String		
printer-make-and-model	String	Brother HL-2170W series CUPS	
printer-state	Number	3	
► printer-state-reasons	Array	(0 items)	
printer-type	Number	8,433,732	
device-uri	String	dnsdd://Brother%20HL-2170W%20series._pdl-datastream._tcp.local./?bidl	
▼ Item 1	Dictionary	(9 items)	
printer-name	String	Brother_HL_5140_series	
printer-info	String	Brother HL-5140 series	
printer-is-accepting-jobs	Boolean	YES	
printer-location	String	word	
printer-make-and-model	String	Brother HL-5140 series CUPS	
printer-state	Number	3	
► printer-state-reasons	Array	(1 item)	
printer-type	Number	45,124	
device-uri	String	usb://Brother/HL-5140%20series?serial=J4J541146	
© SANS.	Mac Forensic Analysis		
All Rights Reserved			

The `org.cups.printers.plist` file located in `/Library/Preferences/` contains details about installed printers. Each printer will be under its own `Item` key.

The screenshot above shows two printers. The Brother HL-2170W was accessed via the network, while the Brother HL-5140 was accessed via a USB cable.

<b>Root</b>	Array	(1 item)
▼ Item 0	Dictionary	(9 items)
<b>printer-name</b>	String	Brother_HL_2170W_series
<b>printer-info</b>	String	Brother HL-2170W series
<b>printer-is-accepting-</b>	Boolean	YES
<b>printer-location</b>	String	
<b>printer-make-and-</b>	String	Brother HL-2170W series CUPS
<b>printer-state</b>	Number	3
▼ <b>printer-state-reasons</b>	Array	(0 items)
<b>printer-type</b>	Number	8,433,732
<b>device-uri</b>	String	dnssd://Brother%20HL-2170W%20series._pdl-datastream._tcp.local./7bidi

# Printing /etc/cups/printers.conf

```
# Printer configuration file for CUPS v1.6.2
# Written by cupsd on 2013-05-24 08:22
# DO NOT EDIT THIS FILE WHEN CUPSD IS RUNNING
<Printer Brother_HL_2170W_series>
UUID urn:uuid:d0de313e-f1d1-3dc9-4e35-3453705570a1
Info Brother HL-2170W series
MakeModel Brother HL-2170W series CUPS
DeviceURI dnssd://Brother%20HL-2170W%20series._pdl-datastream._tcp.local./?bidi
State Idle
StateTime 1369355085
Type 8433732
Accepting Yes
Shared Yes
JobSheets none none
QuotaPeriod 0
PageLimit 0
KLimit 0
OpPolicy default
ErrorPolicy stop-printer
Attribute marker-colors \#000000
Attribute marker-levels -3
Attribute marker-names Black
Attribute marker-types toner
Attribute marker-change-time 1350222390
</Printer>
```

© SANS,  
All Rights Reserved

Mac Forensic Analysis

The printers.conf file located in /etc/cups/ contains specific printer information for each printer installed on the system. The screenshot shows the Brother printer is shared amongst other system users (“Shared Yes”).

More printer configurations can be found in a printer specific configuration file similar to Brother\_HL\_2170W\_series.ppd located in the /etc/cups/ppd/ directory. A PostScript Printer Description (PPD) file contains the printer’s specific capabilities such as page size, resolution, color, and fonts.

The StateTime and Attribute marker-change-time timestamps are Unix epoch timestamps that show when the printer was configured on the system.

```

# Printer configuration file for CUPS v1.6.2
# Written by cupsd on 2013-05-24 08:22
# DO NOT EDIT THIS FILE WHEN CUPSD IS RUNNING
<Printer Brother_HL_2170W_series>
UUID urn:uuid:d0de313e-f1d1-3dc9-4e35-3453705570a1
Info Brother HL-2170W series
NameModel Brother HL-2170W series CUPS
DeviceURI dnssd://Brother%20HL-2170W%20series._pd़l-datastream._tcp.local.%7bidi
State Idle
StateTime 1369355085
Type 8433732
Accepting Yes
Shared Yes
JobSheets none none
QuotaPeriod 0
PageLimit 0
KLimit 0
OpPolicy default
ErrorPolicy stop-printer
Attribute marker-colors \x00000000
Attribute marker-levels -3
Attribute marker-names Black
Attribute marker-types toner
Attribute marker-change-time 1350222390
</Printer>

```

# Printing - Page Log

## /var/log/cups/page\_log

- Printer Name/IP
- User
- Job ID
- Date/Time
- Page Number
- Copies
- Job Billing
- Originating Hostname
- Job Name
  - “Print - .....
- Media
  - “Letter”
- Sides
  - “one-sided” or “-”

```
Brother_HL_2170W_series oompa 1 [31/May/2012:20:26:31 -0400] 1 1 - localhost Print - Amazon.com - Returns Center Letter =
Brother_HL_2170W_series oompa 1 [31/May/2012:20:26:31 -0400] 2 1 - localhost Print - Amazon.com - Returns Center Letter -
Brother_HL_2170W_series oompa 2 [31/May/2012:20:27:39 -0400] 1 1 - localhost Print - VIP.Zappos.com UPS Return Label Letter one-sided
Brother_HL_2170W_series oompa 3 [01/Jun/2012:17:15:52 -0400] 1 1 - localhost Print - VIP.Zappos.com UPS Return Label Letter one-sided
Brother_HL_2170W_series oompa 4 [01/Jun/2012:17:26:25 -0400] 1 2 - localhost Print - VIP.Zappos.com UPS Return Label Letter one-sided
Brother_HL_2170W_series oompa 7 [01/Jun/2012:17:32:06 -0400] 1 1 - localhost Print - VIP.Zappos.com UPS Return Label Letter one-sided
Brother_HL_2170W_series oompa 8 [01/Jun/2012:17:39:37 -0400] 1 1 - localhost Print - VIP.Zappos.com UPS Return Label Letter one-sided
Brother_HL_2170W_series oompa 9 [14/Jun/2012:10:36:03 -0400] 1 1 - localhost Print - Platypus - Wikipedia, the free encyclopedia Letter -
Brother_HL_2170W_series oompa 9 [14/Jun/2012:10:36:03 -0400] 2 1 - localhost Print - Platypus - Wikipedia, the free encyclopedia Letter -
Brother_HL_2170W_series oompa 9 [14/Jun/2012:10:36:04 -0400] 3 1 - localhost Print - Platypus - Wikipedia, the free encyclopedia Letter -
```

© SANS,  
All Rights Reserved

Mac Forensic Analysis

OS X systems use the Common Unix Printing System (CUPS) to perform printing functions. Each print job is recorded in a few different logs.

The page\_log located in the /var/log/cups directory contains the print job log. Each print job contains metadata listed above, along with the job name which is usually the name of the document or webpage that was printed.

**Printer Name/IP** – Name or IP of the printer used

**User** – User Account

**Job ID** – Incremental print job ID number (starts at 1)

**Date/Time** – When the job was printed

**Page Number** – Current page number of print job, each page of a document will have a separate record

**Copies** – Number of copies of page are printed

**Job Billing** – Default is dash (-), job-billing attribute if given

**Originating Hostname** – Hostname of system, although in my experience it is always “localhost”

**Job Name** – The name of the document, webpage, etc. that is being printed. “Print - “ is prepended.

**Media** – Print media, in the example “Letter” paper was used.

**Sides** – Printing can occur on one-sided or double-sided. Dash (-) is default.

References:

CUPS Documentation – page\_log

[http://www.cups.org/documentation.php/ref-page\\_log.html](http://www.cups.org/documentation.php/ref-page_log.html)

Brother\_HL\_2170W\_series\_ompa\_1 [31/May/2012:20:26:31 -0400] 1 1 - localhost Print - Amazon.com - Returns Center Letter -  
Brother\_HL\_2170W\_series\_ompa\_1 [31/May/2012:20:26:31 -0400] 2 1 - localhost Print - Amazon.com - Returns Center Letter -  
Brother\_HL\_2170W\_series\_ompa\_2 [31/May/2012:20:27:39 -0400] 1 1 - localhost Print - VIP.Zappos.com UPS Return Label Letter one-sided  
Brother\_HL\_2170W\_series\_ompa\_3 [01/Jun/2012:17:17:53 -0400] 1 1 - localhost Print - VIP.Zappos.com UPS Return Label Letter one-sided  
Brother\_HL\_2170W\_series\_ompa\_4 [01/Jun/2012:17:26:25 -0400] 1 1 - localhost Print - VIP.Zappos.com UPS Return Label Letter one-sided  
Brother\_HL\_2170W\_series\_ompa\_7 [01/Jun/2012:17:32:06 -0400] 1 1 - localhost Print - VIP.Zappos.com UPS Return Label Letter one-sided  
Brother\_HL\_2170W\_series\_ompa\_8 [01/Jun/2012:17:38:37 -0400] 1 1 - localhost Print - VIP.Zappos.com UPS Return Label Letter one-sided  
Brother\_HL\_2170W\_series\_ompa\_9 [14/Jun/2012:10:36:03 -0400] 1 1 - localhost Print - Platypus - Wikipedia, the free encyclopedia Letter -  
Brother\_HL\_2170W\_series\_ompa\_9 [14/Jun/2012:10:36:03 -0400] 2 1 - localhost Print - Platypus - Wikipedia, the free encyclopedia Letter -  
Brother\_HL\_2170W\_series\_ompa\_9 [14/Jun/2012:10:36:04 -0400] 3 1 - localhost Print - Platypus - Wikipedia, the free encyclopedia Letter -

# Printing - Access Log

## /var/log/cups/access\_log

- Hostname
- Group (-)
- User (-)
- Date/Time
- Method/Resource/Version
- Status Code
  - 200 = Successful
- Bytes in Request
- IPP Operation
  - “Create-Job”
  - “Send Document”
- IPP Status
  - “successful-ok”

```
localhost - [01/Jun/2012:17:32:08 -0400] "POST /printers/Brother_ML_2170W_series HTTP/1.1" 200 1243 Create-Job successful-ok
localhost - [01/Jun/2012:17:32:09 -0400] "POST /printers/Brother_ML_2170W_series HTTP/1.1" 200 166037 Send-Document successful-ok
localhost - [01/Jun/2012:17:32:09 -0400] "POST / HTTP/1.1" 200 345 Set-Job-Attributes successful-ok
localhost - [01/Jun/2012:17:32:31 -0400] "POST /printers/Brother_ML_2170W_series HTTP/1.1" 200 1243 Create-Job successful-ok
localhost - [01/Jun/2012:17:32:31 -0400] "POST /printers/Brother_ML_2170W_series HTTP/1.1" 200 166037 Send-Document successful-ok
localhost - [01/Jun/2012:17:32:32 -0400] "POST / HTTP/1.1" 200 345 Set-Job-Attributes successful-ok
localhost - [14/Jun/2012:10:35:57 -0400] "POST /printers/Brother_ML_2170W_series HTTP/1.1" 200 1267 Create-Job successful-ok
localhost - [14/Jun/2012:10:35:57 -0400] "POST /printers/Brother_ML_2170W_series HTTP/1.1" 200 576775 Send-Document successful-ok
localhost - [14/Jun/2012:10:35:57 -0400] "POST / HTTP/1.1" 200 311 Set-Job-Attributes successful-ok
```

© SANS.  
All Rights Reserved

Mac Forensic Analysis

The access\_log in the /var/log/cups/ directory contains more print job metadata including the size of the job and if the print job was successful or not. These entries can be correlated with the jobs found in the page\_log.

IPP = Internet Printing Protocol

**Hostname** – System hostname, localhost is used in most cases

**Group** – Always a dash (-)

**User** – Always a dash (-), username may be filled in if cupsd.conf was changed

**Date/Time** – Date and time of print job

**Method** – HTTP method

**Resource** – Requested resource

**Version** – HTTP version

**Status Code** – HTTP status code, 200 is successful

**Bytes** – Number of bytes in non-IPP request

**IPP Operation** – IPP Operation Name

**IPP Status** – IPP stats response

References:

CUPS Documentation – access\_log

[http://www.cups.org/documentation.php/ref-access\\_log.html](http://www.cups.org/documentation.php/ref-access_log.html)

CUPS Documentation – cupsd

<http://www.cups.org/documentation.php/ref-cupsd-conf.html>

```
localhost -- [01/Jun/2012:17:32:00 -0400] "POST /printers/Brother_HL_2170W_series HTTP/1.1" 200 1243 Create-Job successful-ok
localhost -- [01/Jun/2012:17:32:00 -0400] "POST /printers/Brother_HL_2170W_series HTTP/1.1" 200 166037 Send-Document successful-ok
localhost -- [01/Jun/2012:17:32:01 -0400] "POST / HTTP/1.1" 200 345 Set-Job-Attributes successful-ok
localhost -- [01/Jun/2012:17:38:31 -0400] "POST /printers/Brother_HL_2170W_series HTTP/1.1" 200 1243 Create-Job successful-ok
localhost -- [01/Jun/2012:17:38:31 -0400] "POST /printers/Brother_HL_2170W_series HTTP/1.1" 200 166037 Send-Document successful-ok
localhost -- [01/Jun/2012:17:38:32 -0400] "POST / HTTP/1.1" 200 345 Set-Job-Attributes successful-ok
localhost -- [14/Jun/2012:10:35:57 -0400] "POST /printers/Brother_HL_2170W_series HTTP/1.1" 200 1267 Create-Job successful-ok
localhost -- [14/Jun/2012:10:35:57 -0400] "POST /printers/Brother_HL_2170W_series HTTP/1.1" 200 570775 Send-Document successful-ok
localhost -- [14/Jun/2012:10:35:57 -0400] "POST / HTTP/1.1" 200 311 Set-Job-Attributes successful-ok
```

# Printer Control Files (1)

## /private/var/spool/cups

- Nine Printer Control Jobs (c#####)

```
sh-3.2# pwd
/private/var/spool/cups
sh-3.2# ls -la
total 72
drwx--x--- 13 root _lp      442 Jun 14 10:36 .
drwxr-xr-x  7 root  wheel   238 May  9 19:22 ..
-rw-------  1 root _lp     1841 May 31 20:26 c00001
-rw-------  1 root _lp     1883 May 31 20:27 c00002
-rw-------  1 root _lp     1883 Jun  1 17:26 c00003
-rw-------  1 root _lp     1883 Jun  1 17:31 c00004
-rw-------  1 root _lp     1815 Jun  1 17:28 c00005
-rw-------  1 root _lp      723 Jun  1 17:31 c00006
-rw-------  1 root _lp     1883 Jun  1 17:33 c00007
-rw-------  1 root _lp     1883 Jun  1 17:38 c00008
-rw-----  1 root _lp     1873 Jun 14 10:36 c00009
drwxrwxr-x  8 root _lp      272 Jun 14 10:46 cache
drwxrwx-T  2 root _lp       68 Jun 19 2011 tmp
```

© SANS,  
All Rights Reserved

Mac Forensic Analysis

Each print job has a printer control file located in the /private/var/spool/cups/ directory. Each file labeled c#####, correlates with the job ID found in the page\_log. In the example shown above, the system has printed nine documents (c00001 – c00009). Each control file contains print job metadata.

## Printer Control Files (2) /private/var/spool/cups

```
bash-3.2# strings c00004
attributes-charset
utf-8H
attributes-natural-language
en-us
printer-uri
4ipp://localhost:631/printers/Brother_HL_2170W_seriesB
job-originating-user-name
oompaB
job-name
27-Day Forecast for Latit...d Longitude 77.36&deg;WB
AP_ColorMatchingMode
AP_ApplicationColorMatchingB
AP_D_InputSlot
collate
ColorModel
GrayB
,com.apple.print.DocumentTicket.PMSpoolFormat
application/pdfB
)com.apple.print.JobInfo.PMApplicationName
ChromeB
!com.apple.print.JobInfo.PMJobName
27-Day Forecast for Latit...d Longitude 77.36&deg;WB
"com.apple.print.JobInfo.PMJobOwner
oompaB
```

© SANS.  
All Rights Reserved.

Mac Forensic Analysis

Each printer control file contains metadata about each print job including which printer was used, originating user account, job name, and which application was used to print from. This information is stored in a proprietary format that is easily viewed using the strings command (although some binary characters could be mistaken for ASCII, watch out for this).

In the example above:

- The printer is located on the local network and is a Brother HL-2170W
- The originating user account is “oompa”
- The name of the job is “27-Day Forecast for Latit...d Longitude 77.36&deg;WB”
- The application used to print is the Chrome web browser

# Printer Data Files /private/var/spool/cups

- Data Files (d#####)
- Removed immediately after successful print
- PDF Files

The screenshot shows a Mac Forensic Analysis interface. On the left, a terminal window displays a file listing of printer data files:

```
drwx--x--- 15 root _lp      510 Jun 14 12:36 .
drwxr-xr-x  7 root wheel    238 May  9 19:22 ..
-rw-----  1 root _lp     1841 May 31 20:26 c00001
-rw-----  1 root _lp     1883 May 31 20:27 c00002
-rw-----  1 root _lp     1883 Jun  1 17:26 c00003
-rw-----  1 root _lp     1883 Jun  1 17:31 c00004
-rw-----  1 root _lp     1815 Jun  1 17:28 c00005
-rw-----  1 root _lp      723 Jun  1 17:31 c00006
-rw-----  1 root _lp     1883 Jun  1 17:33 c00007
-rw-----  1 root _lp     1883 Jun  1 17:38 c00008
-rw-----  1 root _lp     1873 Jun 14 10:36 c00009
-rw-----  1 root _lp     1878 Jun 14 12:36 c00010
drwxrwxr-x  8 root _lp     272 Jun 14 12:36 cache
-rw-r-----  1 root _lp   608373 Jun 14 12:35 d00010-001
drwxrwx--T  2 root _lp      68 Jun 19 2011 tmp
```

On the right, a PDF viewer window titled "CUPS Design Description" is open, showing the contents of the file "d00010-001". The PDF document contains several pages of text, likely describing the printer configuration or design.

Each printer control file has a matching printer data file; while the printer control files are persistent, the printer data files are not. These are created at the time of printing and are PDF files. Each are named in line with the printer control jobs (i.e.: Printer control file c00010 would have the data file d00010-001).

These files should be removed immediately after a print job has successfully printed. The files may be recoverable by carving out PDF files from the disk. If the print job has been canceled or otherwise produced an error, the files may persist for a while longer.

**/Library/Preferences/com.apple.Bluetooth.plist (<=10.7)**

Key	Type	Value
► AudioHeadphones	Array	{1 item}
► DaemonNoRoleSwitchDeviceList	Array	{1 item}
BluetoothVersionNumber	Number	3
ControllerPowerState	Number	0
► PANInterfaces	Array	{1 item}
► HIDDevices	Array	{2 items}
► SCOAudioDevices	Dictionary	{1 item}
► PersistentPorts	Dictionary	{2 items}
▼ DeviceCache	Dictionary	{4 items}
► 00-24-ef-be-dc-07	Dictionary	{14 items}
▼ e8-06-88-33-d9-e0	Dictionary	{1 item}
ClassOfDevice	Number	9536
► cc-6d-a0-2c-96-2e	Dictionary	{9 items}
▼ 70-cd-60-f6-eb-de	Dictionary	{1 item}
ClassOfDevice	Number	9620
► PersistentPortsServices	Dictionary	{1 item}
▼ PairedDevices	Array	{3 items}
Item 0	String	70-cd-60-f6-eb-de
Item 1	String	00-24-ef-be-dc-07
Item 2	String	e8-06-88-33-d9-e0

© SANS,  
All Rights Reserved

Mac Forensic Analysis

The Bluetooth preference pane. Configured Bluetooth devices may be shown in the window presented. Devices may be configured using this preferences pane.

The `com.apple.bluetooth.plist` property list located in the `/Library/Preferences/` directory contains the Bluetooth device cache in the `DeviceCache` key and the paired devices in the `PairedDevices` key.

This `DeviceCache` key contains a history of the Bluetooth devices connected to the system and the `ClassOfDevice` associated with it. The device class (`ClassOfDevice` key) is the type of device that was paired, such as headphones or a mouse.

The device class can be reversed using the Bluetooth header files located in `/System/Library/Frameworks/IOBluetooth.framework/Headers/` directory.

- `Bluetooth.h`
- `BluetoothAssignedNumbers.h`.

Key	Type	Value
► AudioHeadphones	Array	(1 item)
► DaemonNoRoleSwitchDeviceList	Array	(1 item)
BluetoothVersionNumber	Number	3
ControllerPowerState	Number	0
► PANInterfaces	Array	(1 item)
► HIDDevices	Array	(2 items)
► SCOAudioDevices	Diction...	(1 item)
► PersistentPorts	Diction...	(2 items)
▼ DeviceCache	Diction...	(4 items)
► 00-24-ef-be-dc-07	Diction...	(14 items)
▼ e8-06-88-33-d9-e0	Diction...	(1 item)
ClassOfDevice	Number	9536
► cc-6d-a0-2c-96-2e	Diction...	(9 items)
▼ 70-cd-60-f6-eb-de	Diction...	(1 item)
ClassOfDevice	Number	9620
► PersistentPortsServices	Diction...	(1 item)
▼ PairedDevices	Array	(3 items)
Item 0	String	70-cd-60-f6-eb-de
Item 1	String	00-24-ef-be-dc-07
Item 2	String	e8-06-88-33-d9-e0

## /Library/Preferences/com.apple.Bluetooth.plist (10.8+)

DeviceCache	Dictionary	(5 items)
60-6b-bd-0e-57-c8	Dictionary	(4 items)
EIRData	Data	<0d094454 56426c75 65746
ClassOfDevice	Number	525,372
Name	String	DTVBluetooth
LastNameUpdate	Date	Oct 26, 2014, 1:55:55 PM
70-3e-ac-16-05-bc	Dictionary	(3 items)
displayName	String	miPhone6
Name	String	iPhone
LastNameUpdate	Date	Nov 5, 2014, 11:24:42 AM
7c-d1-c3-df-64-68	Dictionary	(1 item)
displayName	String	Sarah's MacBook Air
cc-6d-a0-2c-96-2e	Dictionary	(4 items)
EIRData	Data	<0c09526f 6b752050 6c6179
ClassOfDevice	Number	1,060
Name	String	Roku Player
LastNameUpdate	Date	Oct 26, 2014, 1:49:30 PM

© SANS,  
All Rights Reserved

Mac Forensic Analysis

10.8 introduces more information in the com.apple.Bluetooth.plist. Each Bluetooth MAC address under the DeviceCache key may contain detailed information about the device such as the device time, and last communication timestamps.

This is a good place to look for other devices a user may have had near them.

## Bluetooth Devices system.log – Search “blued”

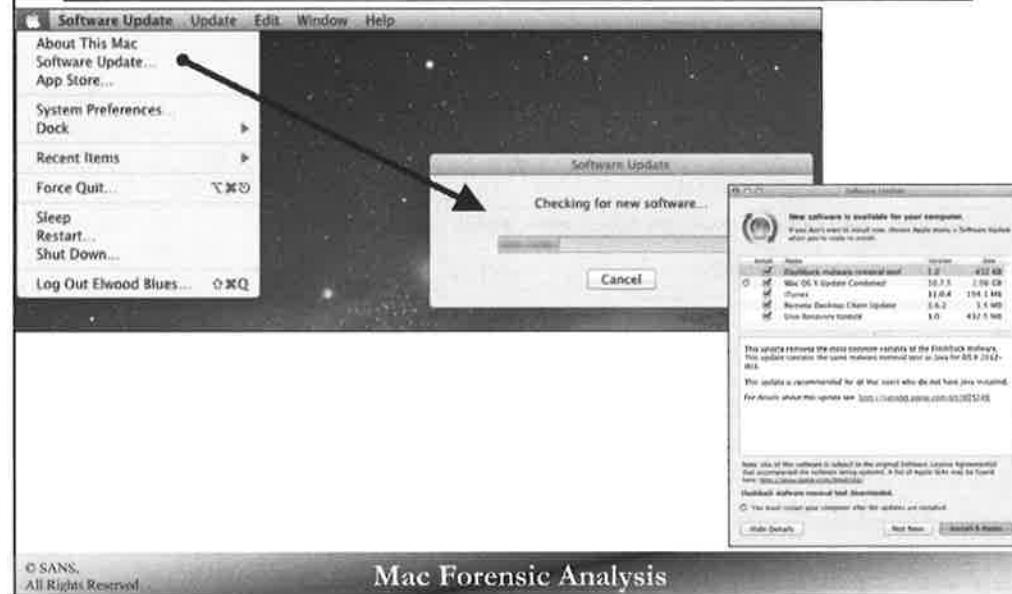
```
Jun 17 09:36:26 bit com.apple.blued[3545]: link key found for
device: 70-cd-60-f6-e8-de
Jun 17 09:36:26 bit com.apple.blued[3545]: link key found for
device: e8-06-88-33-d9-e0
Jun 17 12:57:00 bit blued[3853]: Removing Bluetooth configured
device: 00-24-ef-be-dc-07
Jun 17 13:10:20 bit com.apple.blued[3853]: link key found for
device: 70-cd-60-f6-e8-de
Jun 17 13:10:20 bit com.apple.blued[3853]: link key found for
device: e8-06-88-33-d9-e0
```

© SANS.  
All Rights Reserved.

Mac Forensic Analysis

Bluetooth devices have become increasingly popular and most modern Mac devices have the technology built in. The system.log records the Bluetooth connected devices using the “blued” daemon. Searching for the “blued” term can show us when each device was connected (or disconnected) and its MAC address.

# Software Update & Installation (<=10.7)



The Apple menu contains the item “Software Update...”, when selected a software update window will appear with the message, “checking for new software...”. If updates are found, these updates will be shown in another window where the user is able to pick and choose what to install.

# Software Update (<=10.7)

/Library/Preferences/com.apple.SoftwareUpdate.plist

Key	Type	Value
LastAttemptSystemVersion	String	10.7.4 (11E53)
LastRecommendedUpdatesAvailable	Number	0
► RecommendedUpdates	Array	(1 item)
LastResultCode	Number	100
LastUpdatesAvailable	Number	0
LastAttemptDate	Date	Jun 19, 2012 9:58:37 AM
LastSuccessfulDate	Date	Jun 14, 2012 3:34:31 PM

© SANS,  
All Rights Reserved

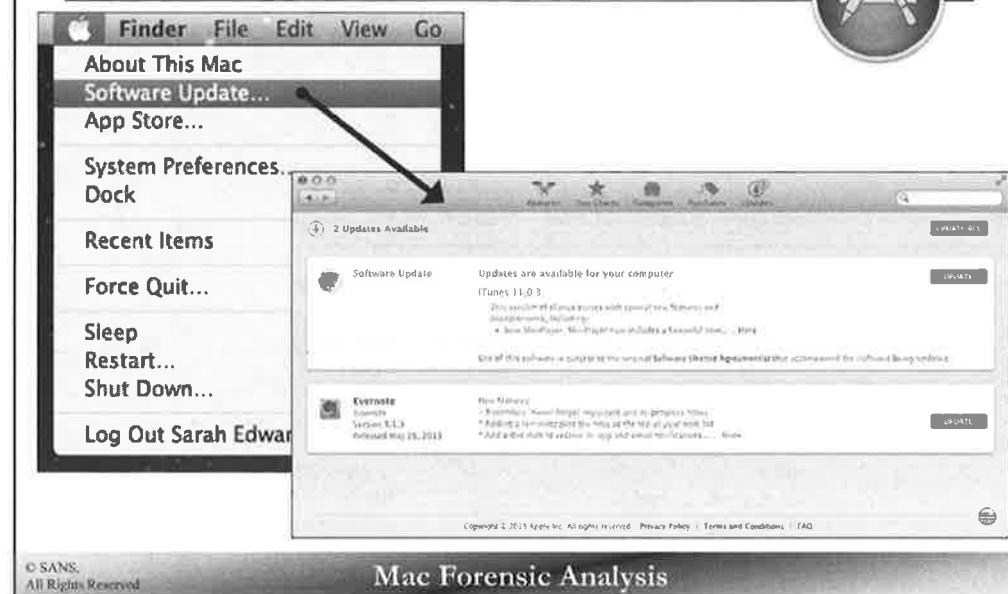
Mac Forensic Analysis

The com.apple.SoftwareUpdate.plist property list located in the /Library/Preferences/ directory contains information about the software update function.

The key “LastAttemptDate” contains the date when the Software Update application last **attempted** to update software, while the “LastSuccessfulDate” contains the date when the software was **successfully** checked for an update.

The property list also shows the last system version that was used when it last attempted an update. It also shows the number of updates available for installation.

# Software Update & Installation (10.8+)



© SANS.  
All Rights Reserved

Mac Forensic Analysis

The Apple menu contains the item “Software Update...”. When selected it opens the Mac App Store on the “Updates” tab as shown in the screenshot above. This window shows the updates that are available, both the Apple system updates and any application that had been downloaded and installed via the Mac App Store.

# Software Update (10.8+)

## /Library/Preferences/com.apple.SoftwareUpdate.plist

Root	Dictionary	(15 items)
LastResultCode	Number	0
IgnoringUnseenRamped	Boolean	NO
LastAttemptSystemVersion	String	10.8.3 (12D78)
DidRegisterLocalUpdates	Boolean	YES
LastUpdatesAvailable	Number	1
LastAttemptDate	Date	May 26, 2013 11:02:58 AM
LastSuccessfulCatalogTag	String	"16780-4dd681942fb40"
LastRecommendedUpdatesAvailable	Number	1
SkipLocalCDN	Boolean	NO
RecommendedUpdates	Array	(1 item)
Item 0	Dictionary	(4 items)
Identifier	String	iTunesXPatch
Product Key	String	zzz041-9781
Display Name	String	iTunes
Display Version	String	11.0.3
LastCriticalSuccessfulDate	Date	May 26, 2013 11:02:58 AM
LastSessionSuccessful	Boolean	YES
LastCriticalSuccessfulCatalogTag	String	"16780-4dd681942fb40"
LastBackgroundSuccessfulDate	Date	May 18, 2013 4:59:39 AM
LastSuccessfulDate	Date	May 26, 2013 11:02:58 AM

© SANS.  
All Rights Reserved

Mac Forensic Analysis

The com.appleSoftwareUpdate.plist property list for 10.8+ systems contains additional data.

This property list contains data such as when critical updates and updates downloaded in the background were last installed.

The key LastBackgroundSuccessfulDate contains the timestamp of the last time updates were installed in the background.

<b>▼ Root</b>	<b>Dictionary</b>	<b>(15 items)</b>
LastResultCode	Number	0
IgnoringUnseenRamped	Boolean	NO
LastAttemptSystemVersion	String	10.8.3 (12D78)
DidRegisterLocalUpdates	Boolean	YES
LastUpdatesAvailable	Number	1
LastAttemptDate	Date	May 26, 2013 11:02:58 AM
LastSuccessfulCatalogTag	String	"16780-4dd681942fb40"
LastRecommendedUpdatesAvailable	Number	1
SkipLocalCDN	Boolean	NO
<b>▼ RecommendedUpdates</b>	<b>Array</b>	<b>(1 item)</b>
<b>▼ Item 0</b>	<b>Dictionary</b>	<b>(4 items)</b>
Identifier	String	iTunesXPatch
Product Key	String	zzzz041-9781
Display Name	String	iTunes
Display Version	String	11.0.3
LastCriticalSuccessfulDate	Date	May 26, 2013 11:02:58 AM
LastSessionSuccessful	Boolean	YES
LastCriticalSuccessfulCatalogTag	String	"16780-4dd681942fb40"
LastBackgroundSuccessfulDate	Date	May 18, 2013 4:59:39 AM
LastSuccessfulDate	Date	May 26, 2013 11:02:58 AM

Install History /Library/Receipts/InstallHistory.plist			
Key	Type	Value	
Item 27	Dictionary	(5 items)	
date	Date	May 27, 2012 3:46:56 PM	
displayName	String	Wireshark 1.6.8 Intel 64	
displayVersion	String		
packageIdentifiers	Array	(3 items)	
processName	String	Installer	
Item 28	Dictionary	(5 items)	
Item 29	Dictionary	(5 items)	
Item 30	Dictionary	(5 items)	
Item 31	Dictionary	(5 items)	
Item 32	Dictionary	(5 items)	
Item 33	Dictionary	(5 items)	
Item 34	Dictionary	(5 items)	
Item 35	Dictionary	(5 items)	
Item 36	Dictionary	(5 items)	
date	Date	Jun 14, 2012 3:34:29 PM	
displayName	String	iTunes	
displayVersion	String	10.6.3	
packageIdentifiers	Array	(6 items)	
processName	String	Software Update	

© SANS.  
All Rights Reserved

Mac Forensic Analysis

The InstallHistory.plist property list located in /Library/Receipts/ contains a software install history that includes the timestamp, software package name, and what process was used to install the software.

The process name, “Installer” is normally seen when a user manually installs a piece of software such as Wireshark while the process “Software Update” is seen when the system installs its updates.

# Receipt Files

## /var/db/receipts/

```
-rw-r--r-- 1 root wheel 35290 May 27 15:46 org.wireshark.ChmodBPF.pkg.bom  
-rw-r--r-- 1 root wheel 260 May 27 15:46 org.wireshark.ChmodBPF.pkg.plist  
-rw-r--r-- 1 root wheel 62594 May 27 15:46 org.wireshark.Wireshark.pkg.bom  
-rw-r--r-- 1 root wheel 256 May 27 15:46 org.wireshark.Wireshark.pkg.plist  
-rw-r--r-- 1 root wheel 35138 May 27 15:46 org.wireshark.cli.pkg.bom  
-rw-r--r-- 1 root wheel 255 May 27 15:46 org.wireshark.cli.pkg.plist
```

Key	Type	Value
PackageVersion	String	0.0.0.0
PackageIdentifier	String	org.wireshark.Wireshark.pkg
InstallPrefixPath	String	Applications
InstallDate	Date	May 27, 2012 3:46:56 PM
PackageFileName	String	wireshark.pkg
InstallProcessName	String	Installer

© SANS.  
All Rights Reserved.

Mac Forensic Analysis

The same information can be found in a slightly different format in the /var/db/receipts directory. Each software package install has a .bom and a .plist file. The property list file also contains the software install timestamp, package name and the installer process.

The .bom file is the Mac OS X Bill of Materials (BOM) file that contains a list of files and metadata for the installed application. These can be viewed with the command lsbom, as shown on the next slide.

# Receipt Files - BOM Files (lsbom)

## /var/db/receipts/\*.bom

nibble@receipts: stewards\$ lsbom com.adobe.pkg.FlashPlayer.bom						
.	40755	502/28				
./Library	41775	0/80				
./Library/Application Support	40775	0/80				
./Library/Application Support/Adobe	40775	0/80				
./Library/Application Support/Adobe/Flash Player Install Manager	40755	0/80				
./Library/Application Support/Adobe/Flash Player Install Manager/fpsaud	100744	0/80	59248	3932184723		
./Library/Internet Plug-Ins	40775	0/80				
./Library/Internet Plug-Ins/flashplayer.xpt	100664	0/80	856	1969355171		
./Library/LaunchDaemons	40755	0/0				
./Library/LaunchDaemons/com.adobe.fpsaud.plist	100644	0/0	462	1274181950		
./Library/PreferencePanes	40755	0/0				
./Library/PreferencePanes/Flash Player.prefPane	40775	0/0				
./Library/PreferencePanes/Flash Player.prefPane/Contents	40775	0/0				
./Library/PreferencePanes/Flash Player.prefPane/Contents/Info.plist	100664	0/0	827	B94812453		
./Library/PreferencePanes/Flash Player.prefPane/Contents/MacOS	40775	0/0				
./Library/PreferencePanes/Flash Player.prefPane/Contents/MacOS/Flash Player	100775	0/0	1212576	1112384951		
./Library/PreferencePanes/Flash Player.prefPane/Contents/Resources	40775	0/0				
./Library/PreferencePanes/Flash Player.prefPane/Contents/Resources/FlashPlayerPreferences.nib	100664	0/0	95850	2B46781901		
./Library/PreferencePanes/Flash Player.prefPane/Contents/Resources/FlashPlayerPreferences.png	100664	0/0	1144	1473368541		
./Library/PreferencePanes/Flash Player.prefPane/Contents/Resources/FlashPlayerPreferences.searchTerms	100664	0/0	1466	2094979473		
./Library/PreferencePanes/Flash Player.prefPane/Contents/Resources/info.plist.strings	100664	0/0	244	160668188		
./Library/PreferencePanes/Flash Player.prefPane/Contents/Resources/inusSign.png	100664	0/0	160	2102779873		
./Library/PreferencePanes/Flash Player.prefPane/Contents/Resources/pencilIcon.png	100664	0/0	380	2776592378		
./Library/PreferencePanes/Flash Player.prefPane/Contents/Resources/plusSign.png	100664	0/0	278	162556027		
./Library/PreferencePanes/Flash Player.prefPane/Contents/Resources/version.plist	100664	0/0	185	4240636033		

© SANS,  
All Rights Reserved.

**Mac Forensic Analysis**

The screenshot above shows an example of the output from the command `lsbom`. This shows the various files associated with the BOM file `com.adobe.pkg.FlashPlayer.bom`.

Default output contains:

- File path
- Mode (octal)
- UID/GID

Each type of file will contain different information in the BoM file.

- If the file is a plain file, the UID/GID is followed by the file size and CRC checksum.
- If the file is a symbolic link, the UID/GID is followed by the size and CRC checksum of the link path, and the link path.
- If the file is a device file, the UID/GID is followed by device number.

References:

Man Page for `lsbom`

```

nibble:receipts sledwards$ lsboom com.adobe.pkg.FlashPlayer.bom
.
40755 502/20
./Library 41775 0/0
./Library/Application Support 40775 0/0
./Library/Application Support/Adobe 40775 0/0
./Library/Application Support/Adobe/Flash Player Install Manager 40755 0/0
./Library/Application Support/Adobe/Flash Player Install Manager/fpsaud 100744 0/0
59246 3932184723
./Library/Internet Plug-Ins 40775 0/0
./Library/Internet Plug-Ins/Flash Player.plugin.lzma 100664 0/0
17110174 752483422
./Library/Internet Plug-Ins/flashplayer.xpt 100664 0/0
856 1969355171
./Library/LaunchDaemons 40755 0/0
./Library/LaunchDaemons/com.adobe.fpsaud.plist 100644 0/0
462 1274181950
./Library/PreferencePanes 40755 0/0
./Library/PreferencePanes/Flash Player.prefPane 40775 0/0
./Library/PreferencePanes/Flash Player.prefPane/Contents 40775 0/0
100664 0/0
827 894812453
./Library/PreferencePanes/Flash Player.prefPane/Contents/MacOS 40775 0/0
./Library/PreferencePanes/Flash Player.prefPane/Contents/MacOS/Flash Player 100775 0/0
1212576 1112384951
./Library/PreferencePanes/Flash Player.prefPane/Contents/Resources 40775 0/0
./Library/PreferencePanes/Flash Player.prefPane/Resources/FlashPlayerPreferences.nib 100664 0/0
100664 0/0
95850 2846781901
./Library/PreferencePanes/Flash Player.prefPane/Resources/FlashPlayerPreferences.png 100664 0/0
100664 0/0
1144 1473368544
./Library/PreferencePanes/Flash Player.prefPane/Resources/FlashPlayerPreferences.searchTerms 100664 0/0
100664 0/0
1466 2994979473
./Library/PreferencePanes/Flash Player.prefPane/Resources/InfoPlist.strings 100664 0/0
244 169668188
./Library/PreferencePanes/Flash Player.prefPane/Resources/Resources/minusSign.png 100664 0/0
160 2182779873
./Library/PreferencePanes/Flash Player.prefPane/Resources/pencilIcon.png 100664 0/0
380 2776592378
./Library/PreferencePanes/Flash Player.prefPane/Resources/plussign.png 100664 0/0
278 162556927
./Library/PreferencePanes/Flash Player.prefPane/Resources/version.plist 100664 0/0
185 4240636033

```

# Software Installations

## /var/log/install.log

```
Jun 9 12:43:51 nibble.blah installd[64413]: PackageKit: ---- Begin install ----
Jun 9 12:43:51 nibble.blah installd[64413]: PackageKit: request=PKInstallRequest <1 packages, destination=>
Jun 9 12:43:51 nibble.blah installd[64413]: PackageKit: packages={
    "PKLeopardPackage <file://localhost/var/folders/f1/_wpdftvx3k3_c96vhxd0fkqr0000gn/C/com.apple.appstore/404458553/mzps150188289789740536.pkg@com.omnigroup.OmniGraffiti.MacAppStore.pkg>"}
)
Jun 9 12:43:52 nibble.blah installd[64413]: PackageKit: Extracting file://localhost/var/folders/f1/_wpdftvx3k3_c96vhxd0fkqr0000gn/C/com.apple.appstore/404458553/mzps150188289789740536.pkg@com.omnigroup.OmniGraffiti.MacAppStore.pkg>
Jun 9 12:43:53 nibble.blah installd[64413]: PackageKit: Applying atomic-update from bundle at Applications/OmniGraffiti 5.app
Jun 9 12:43:56 nibble.blah Software Update[71745]: PackageKit: Missing bundle path, skipping: <bundle id="com.apple.SystemProfiler"></bundle>
Jun 9 12:43:56 nibble.blah Software Update[71745]: PackageKit: Missing bundle path, skipping: <bundle id="com.apple.java.JavaPreferences"></bundle>
Jun 9 12:43:56 nibble.blah Software Update[71745]: PackageKit: Missing bundle path, skipping: <bundle id="com.apple.iCal"></bundle>
Jun 9 12:43:58 nibble.blah installd[64413]: PackageKit: Verifying code signature on /var/folders/zz/zyxpvxq6csfxvn_n000000000000/Cleanup At Startup/PKInstallSandboxManager/3.sandbox/Root/Applications/OmniGraffiti 5.app
Jun 9 12:44:01 nibble.blah installd[64413]: PackageKit: Wrote MAS receipt into Applications/OmniGraffiti 5.app
Jun 9 12:44:01 nibble.blah installd[64413]: PackageKit: prevent user idle system sleep
Jun 9 12:44:01 nibble.blah installd[64413]: PackageKit: suspending backup
Jun 9 12:44:01 nibble.blah installd[64413]: PackageKit: temporarily excluding: /Applications, /Library, /System, /bin, /private, /sbin, /usr
Jun 9 12:44:03 nibble.blah installd[64413]: PackageKit: Shoving /var/folders/zz/zyxpvxq6csfxvn_n000000000000/Cleanup At Startup/PKInstallSandboxManager/3.sandbox/Root (1 items) to /
Jun 9 12:44:03 nibble.blah installd[64413]: PackageKit: Writing receipt for com.omnigroup.OmniGraffiti.MacAppStore to /private/var/db/receipts
Jun 9 12:44:03 nibble.blah installd[64413]: PackageKit: Touched bundle Applications/OmniGraffiti 5.app
Jun 9 12:44:03 nibble.blah installd[64413]: PackageKit: Touched bundle Applications/OmniGraffiti 5.app/Contents/Resources/OmniGroupCrashCatcher.app
Jun 9 12:44:03 nibble.blah installd[64413]: Installed "OmniGraffiti" (5.4.3)
```

© SANS.  
All Rights Reserved.

Mac Forensic Analysis

The `install.log` located in `/var/log/` contains the packages and applications installed on the system, whether administrator credentials were needed, and the path to where the application was located when it was installed.

```

Jun 9 12:43:51 nibble.blah install[64413]: PackageKit: ---- Begin install -----
Jun 9 12:43:51 nibble.blah install[64413]: PackageKit: request=>PKInstallRequest <1 packages, destination=>
Jun 9 12:43:51 nibble.blah install[64413]: PackageKit: packages={
    "PKLeopardPackage <file:/localhost/var/folders/f1/_wpdfvtv3k3_c96vhxd0fkqr0000gn/C/com.apple.appstore/404458553/
mzps1501882897900740536.pkg#com.omnigroup.OmniGraffle.MacAppStore.pkg">
}
Jun 9 12:43:52 nibble.blah install[64413]: PackageKit: Extracting file://localhost/var/folders/f1/_wpdfvtv3k3_c96vhxd0fkqr0000gn/C/
com.apple.appstore/404458553/mzps1501882897890740536.pkg#com.omnigroup.OmniGraffle.MacAppStore.pkg (destination=/var/folders/zz/
zyxvpq6csfxvn_n0000000000000000/Cleanup At Startup/PKInstallSandboxManager/3.sandbox/Root/Applications, uid=0)
Jun 9 12:43:53 nibble.blah install[64413]: PackageKit: Applying atomic-update from bundle at Applications/OmniGraffle 5.app
Jun 9 12:43:56 nibble.blah Software Update[71745]: PackageKit: Missing bundle path, skipping: <bundle id="com.apple.SystemProfiler"></
bundle>
Jun 9 12:43:56 nibble.blah Software Update[71745]: PackageKit: Missing bundle path, skipping: <bundle
id="com.apple.java.JavaPreferences"></bundle>
Jun 9 12:43:56 nibble.blah Software Update[71745]: PackageKit: Missing bundle path, skipping: <bundle id="com.apple.iCal"></bundle>
Jun 9 12:43:58 nibble.blah install[64413]: PackageKit: Verifying code signature on /var/folders/zz/zyxvpq6csfxvn_n0000000000000000/Cleanup
At Startup/PKInstallSandboxManager/3.sandbox/Root/Applications/OmniGraffle 5.app
Jun 9 12:44:01 nibble.blah install[64413]: PackageKit: Wrote MAS receipt into Applications/OmniGraffle 5.app
Jun 9 12:44:01 nibble.blah install[64413]: PackageKit: prevent user idle system sleep
Jun 9 12:44:01 nibble.blah install[64413]: PackageKit: suspending backupd
Jun 9 12:44:01 nibble.blah install[64413]: PackageKit: temporarily excluding: /Applications, /Library, /System, /bin, /private, /sbin, /usr
Jun 9 12:44:03 nibble.blah install[64413]: PackageKit: Shoving /var/folders/zz/zyxvpq6csfxvn_n0000000000000000/Cleanup At Startup/
PKInstallSandboxManager/3.sandbox/Root (1 items) to /
Jun 9 12:44:03 nibble.blah install[64413]: PackageKit: Writing receipt for com.omnigroup.OmniGraffle.MacAppStore to /private/var/db/
receipts
Jun 9 12:44:03 nibble.blah install[64413]: PackageKit: Touched bundle Applications/OmniGraffle 5.app
Jun 9 12:44:03 nibble.blah install[64413]: PackageKit: Touched bundle Applications/OmniGraffle 5.app/Contents/Resources/
OmniGroupCrashCatcher.app
Jun 9 12:44:03 nibble.blah install[64413]: Installed "OmniGraffle" (5.4.3)

```

# Kernel Extensions

## /System/Library/Extensions/\*.kext

- Dynamically loaded executable code in kernel space
  - Low Level Device Drivers
  - Network Filters
  - File Systems
  - ...keyloggers?

```
76 0 0xffffffff7f81340000 0xa000 0xa000 com.apple.driver.AppleMCCSControl {1.0.24} <55 9 7 5 4 3 1>
77 0 0xffffffff7f81214000 0x5000 0x5000 com.apple.driver.AppleUpstreamUserClient {3.5.9} <55 9 8 7 5 4 3 1>
78 1 0xffffffff7f813e5000 0xa4000 0xa4000 com.apple.driver.DspFuncLib {2.1.1f12} <67 66 5 4 3 1>
79 0 0xffffffff7f81489000 0xaf000 0xaf000 com.apple.driver.AppleHDA {2.1.1f12} <78 67 65 64 57 55 6 5 4 3 1>
81 1 0xffffffff7f80f57000 0x5000 0x5000 com.apple.kext.triggers {1.0} <7 6 5 4 3 1>
82 0 0xffffffff7f80f6c000 0x9000 0x9000 com.apple.filesystems.autofs {3.0} <81 7 6 5 4 3 1>
83 0 0xffffffff7f81631000 0x5000 0x5000 com.vmware.kext.vmmemctl {0068.29.96} <7 5 4 3 1>
85 0 0xffffffff7f81637000 0xa000 0xa000 com.vmware.kext.VMHGfs {0868.29.96} <5 4 3 1>
88 0 0xffffffff7f80802000 0x4000 0x4000 com.fsb.kext.logKext {2.3} <25 4 3>
```

© SANS.  
All Rights Reserved

Mac Forensic Analysis

Kernel extensions are similar to loadable kernel modules on Linux systems. They are often used as device drivers, network filters, or support for various file systems. They can also be used maliciously as keyloggers as shown in the kextstat output in the screenshot above. The com.fsb.kext.logKext is the Kernel Extension for the open source keylogger LogKext.

You can use the kextstat command on a live system to view the status of the kernel extensions.

76	0	0xffffffff7f81340000	0xa000	0xa000	com.apple.driver.AppleHCCSControl (1.0.24) <55 9 7 5 4 3 1>
77	0	0xffffffff7f81214000	0x5000	0x5000	com.apple.driver.AppleUpstreamUserClient (3.5.9) <55 9 8 7 5 4 3 1>
78	1	0xffffffff7f813e5000	0xa4000	0xa4000	com.apple.driver.DspFuncLib (2.1.1f12) <67 66 5 4 3 1>
79	0	0xffffffff7f81489000	0xaf000	0xaf000	com.apple.driver.AppleHDA (2.1.1f12) <78 67 65 64 57 55 6 5 4 3 1>
81	1	0xffffffff7f80f67000	0x5000	0x5000	com.apple.kext.triggers (1.0) <7 6 5 4 3 1>
82	0	0xffffffff7f80f6c000	0x9000	0x9000	com.apple.filesystems.autofs (3.0) <81 7 6 5 4 3 1>
83	0	0xffffffff7f81631000	0x5000	0x5000	com.vmware.kext.vmmemctl (0068.29.96) <7 5 4 3 1>
85	0	0xffffffff7f81637000	0xa000	0xa000	com.vmware.kext.vmhgfs (0068.29.96) <5 4 3 1>
88	0	0xffffffff7f80802000	0x4000	0x4000	com.fsb.kext.logKext (2.3) <25 4 3>

# Kernel Extensions - Bundle /System/Library/Extensions/\*.kext

## Bundle File

	Dictionary	(22 items)
BuildMachineOSBuild	String	12A251
Localization native development reg	String	English
Bundle display name	String	
Executable file	String	exfat
Get Info string	String	1.3, Copyright Apple Inc. 2009-2012
Bundle identifier	String	com.apple.filesystems.exfat
InfoDictionary version	String	6.0
Bundle name	String	exfat
Bundle OS Type code	String	KEXT
Bundle versions string, short	String	1.3
Bundle creator OS Type code	String	????
Bundle version	String	1.3
DTCompiler	String	com.apple.compilers.llvm clang 1_0
DTPlatformBuild	String	4F212
DTPlatformVersion	String	GM
DTSDKBuild	String	12A251
DTSDKName	String	
DTXcode	String	0440
DTXcodeBuild	String	4F212
► IOKitPersonalities	Dictionary	(0 items)
OSBundleAllowUserLoad	Boolean	YES
► OSBundleLibraries	Dictionary	(4 items)

© SANS,  
All Rights Reserved

Mac Forensic Analysis

Each kernel extension is a bundle file, meaning they appear as a single file in the Finder application.

The screenshot on the left shows the file structure of the `exfat` file system kernel extension. Each will contain a `Contents` directory with an `Info.plist` file as shown on the right. The `Info.plist` file contains the identifying information of the kernel extension including the version number, executable file name, and build information.

## References:

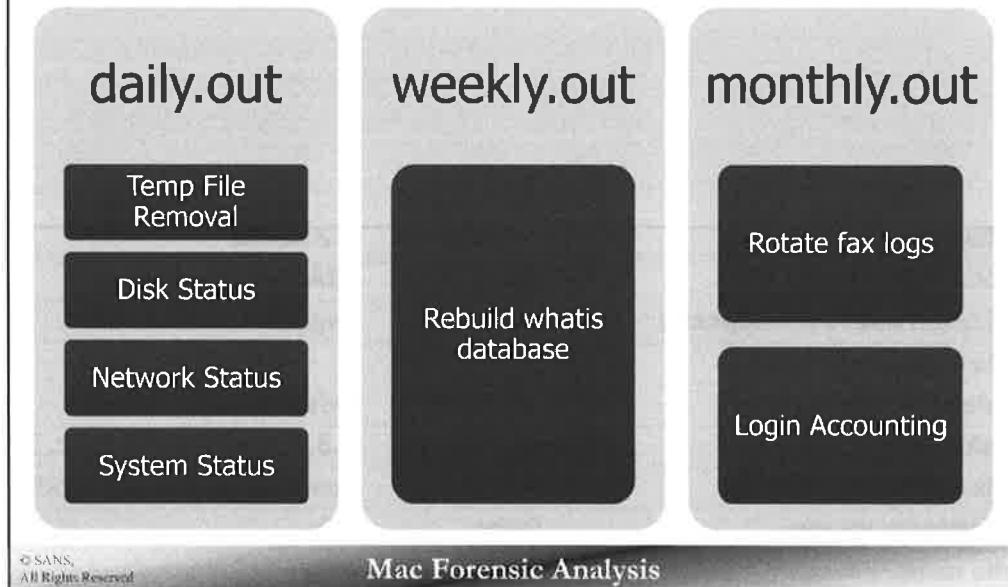
Apple Developer Documentation – Kernel Programming

<https://developer.apple.com/library/mac/documentation/Darwin/Conceptual/KernelProgramming/KernelProgramming.pdf>

▼Information Property List	Dictionary	(22 items)
BuildMachineOSBuild	String	12A251
Localization native development reg	String	English
Bundle display name	String	
Executable file	String	exfat
Get Info string	String	1.3, Copyright Apple Inc. 2009-2012
Bundle identifier	String	com.apple.filesystems.exfat
InfoDictionary version	String	6.0
Bundle name	String	exfat
Bundle OS Type code	String	KEXT
Bundle versions string, short	String	1.3
Bundle creator OS Type code	String	????
Bundle version	String	1.3
DTCompiler	String	com.apple.compilers.llvm clang.1_0
DTPlatformBuild	String	4F212
DTPlatformVersion	String	GM
DTSDKBuild	String	12A251
DTSDKName	String	
DTXcode	String	0440
DTXcodeBuild	String	4F212
► IOKitPersonalities	Dictionary	(0 items)
OSBundleAllowUserLoad	Boolean	YES
► OSBundleLibraries	Dictionary	(4 items)

# Periodic Maintenance Script Logs

## /var/log/\*.out



There are three maintenance scripts that are run periodically. These scripts are located in `/etc/periodic` and are named `daily`, `weekly`, and `monthly` after how often they are run. These scripts are started as launch daemons found in the `/System/Library/LaunchDaemons` directory:

- `com.apple.periodic-daily.plist`
- `com.apple.periodic-monthly.plist`
- `com.apple.periodic-weekly.plist`

Each script produces a log file located in `/var/log/` named similar to the script that created it:

- **daily.out** – Contains output from temporary file removal, disk, network, and system status.
- **weekly.out** – Contains output from rebuilding the `whatis` database. This database contains a description of system commands.
- **monthly.out** – Contains output from rotating the fax logs and login accounting.

In older versions of OS X, these files may be named `daily.log`, `weekly.log`, and `monthly.log`.

Reference:

[periodic Man Page](#)

[periodic.conf Man Page](#)



**COMPUTER FORENSICS**  
and INCIDENT RESPONSE



## Exercise 3.1 – System Data & Preferences

© SANS.  
All Rights Reserved

Mac Forensic Analysis

This page intentionally left blank.

# Agenda

---

Part 1 – System Information

Part 2 – System Preferences & Applications

Part 3 – Log Analysis

Part 4 – Timeline Analysis & Data Correlation

© SANS,  
All Rights Reserved

Mac Forensic Analysis

This page intentionally left blank.



## Section 3 – Part 3

## Log Parsing & Analysis

© SANS.  
All Rights Reserved

Mac Forensic Analysis

This page intentionally left blank.

# Log Parsing & Analysis

Log Basics

Log Formats

Log Recovery

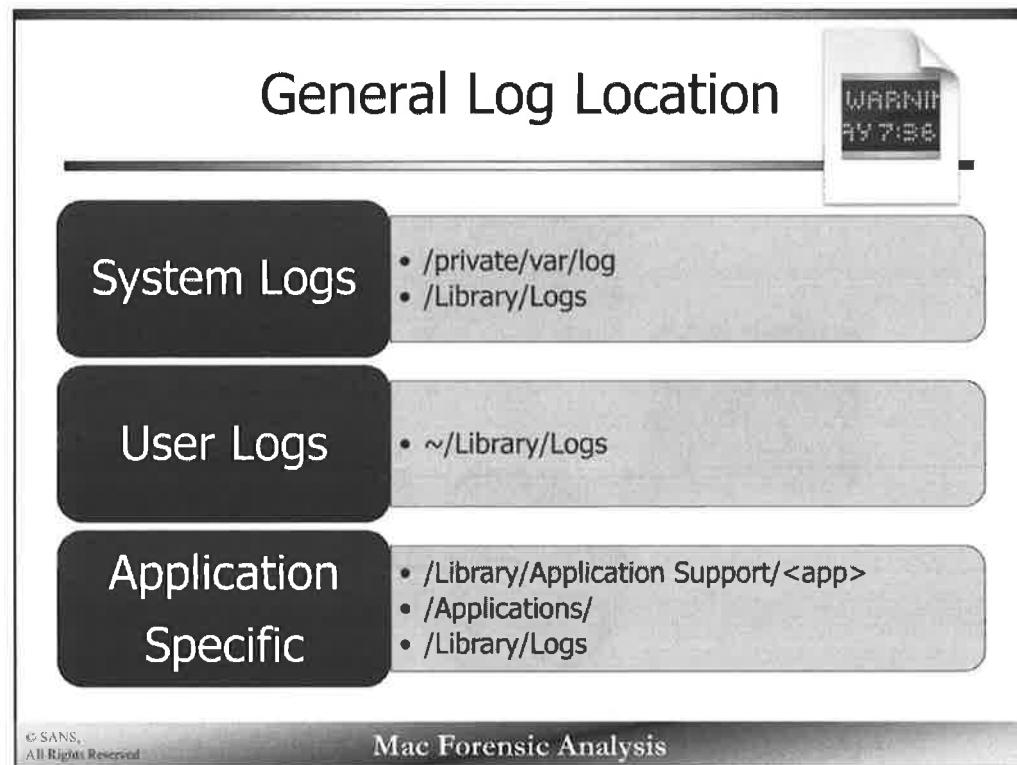
Apple System Logs (ASL)

BSM Audit Logs

© SANS®  
All Rights Reserved

Mac Forensic Analysis

This page intentionally left blank.



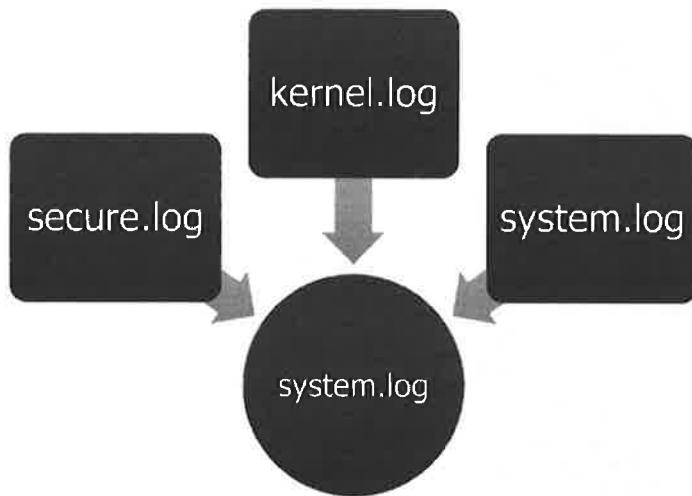
There are three primary locations in OS X where logs are found.

System logs, those that have to do with the operating system, can be found in `/private/var/log` and `/Library/Logs`.

User specific logs are found in each user account in their Library directory, `~/Library/Logs`

Application logs may be found in `/Library/Application Support/` or `/Applications/` directory under the particular application.

## Major Log Changes in 10.8



© SANS  
All Rights Reserved

Mac Forensic Analysis

In 10.8 there no longer exists separate log files for `secure.log`, `kernel.log`, and `system.log`. These files have been combined into one, `system.log` file.

# OS X Log Basics

- Tends to use Standard Unix Log Format
  - MMM DD HH:MM:SS Host Service: Message
- Most are in plaintext
- bzip2 or gzip compression used for archival after log turnover

```
Apr 18 22:44:02 byte Firewall[89]: Stealth Mode connection attempt
Apr 18 22:44:02 byte Firewall[89]: Stealth Mode connection attempt
Apr 18 22:44:04 byte Firewall[89]: Stealth Mode connection attempt
Apr 18 22:44:04 byte Firewall[89]: Stealth Mode connection attempt
Apr 18 22:44:10 byte Firewall[89]: Stealth Mode connection attempt
Apr 18 22:44:10 byte Firewall[89]: Stealth Mode connection attempt
Apr 18 22:44:22 byte Firewall[89]: Stealth Mode connection attempt
Apr 18 22:44:22 byte Firewall[89]: Stealth Mode connection attempt
Apr 18 22:44:46 byte Firewall[89]: Stealth Mode connection attempt
Apr 18 22:44:46 byte Firewall[89]: Stealth Mode connection attempt
Apr 18 23:01:12 byte Firewall[89]: Stealth Mode connection attempt
```

© SANS.  
All Rights Reserved.

Mac Forensic Analysis

Most of the logs found in OS X tend to use the standard Unix Log Format. This format uses a date format that does not include a year or a time zone to put context to the log data.

Most of the logs are available in plaintext format, meaning they can be read without further processing or parsing. Normal Unix operating systems will archive (rotate) log files after they grow to a certain size or are old enough (look in their associated .conf files in /etc).

Generally speaking, on 10.8 and prior systems, bzip2 compression is used to archive its log data while on 10.9 systems, gzip is used.

## bzip2 or gzip Decompression

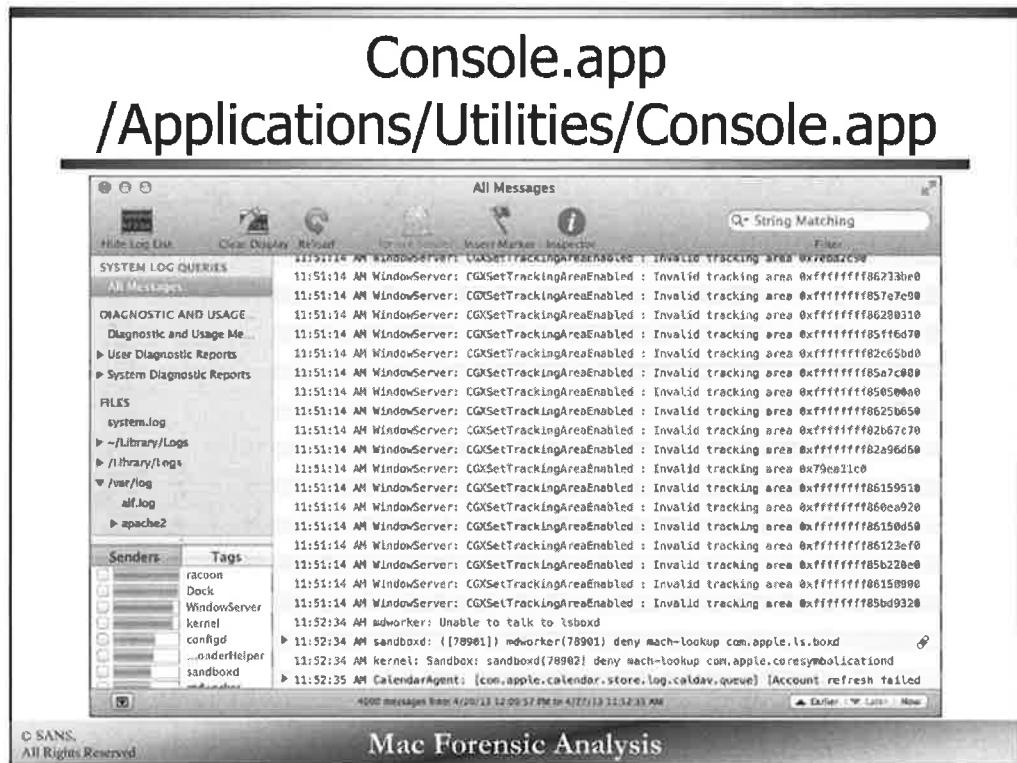
- Use bzcat or gzcat on OS X
  - (oldest -> newest)
  - **Bzip2** - system.log.7.bz2 -> system.log.0.bz2
  - **Gzip** - system.log.7.gz -> system.log.0.gz

```
1.bzcat system.log.7.bz2
    system.log.6.bz2 system.log.5.bz2
    system.log.4.bz2 system.log.3.bz2
    system.log.2.bz2 system.log.1.bz2
    system.log.0.bz2 >> system_all.log
2.cat system.log >> system_all.log
```

If required, the archived logs can be decompressed using the command bzcat. To create a comprehensive log file with the entries in the correct temporal context (oldest to newest) the bzcat command can be used to “concatenate” the contents of the archive files, the larger the number (ie: system.log.7.bz2) the older the log archive.

- 0 = newest
- 7 = oldest

The first command concatenates the contents of the archive files from oldest to newest into a file named system\_all.log, the second command concatenates the current log file system.log to the system\_all.log file to create a complete system.log file for the system.



The Console.app is a native log viewer on OS X. It can be an incredibly powerful and useful tool for log analysis.

In the left pane is an auto-populated log list that includes system, user, and application logs. Below that is a window that shows the most popular message Sender and Tags when used with the “All Messages” display. The button in the lower-left can access this pane.

The main window shows the logs, one line per entry. The “All Messages” display is more dynamic than other log displays. This “All Messages” includes all the syslog entries that the user has access to. These messages have additional metadata that can be viewed using the Message Inspector – the blue circle icon marked with the letter “i”.

All Messages	
<input type="text" value="Q - String Matching"/>	
<input type="checkbox"/> Hide Log List <input type="checkbox"/> Clear Display <input type="checkbox"/> Reload <input type="checkbox"/> Ignore Sender <input type="checkbox"/> Insert Marker <input type="checkbox"/> Inspector <input type="checkbox"/> Filter	
<b>SYSTEM LOG QUERIES</b>	
All Messages	
Diagnostic and Usage Me...	
► User Diagnostic Reports	
► System Diagnostic Reports	
FILES	
system.log	
▼ ~/Library/Logs	
▼ /var/log	
alf.log	
► apache2	
Senders	Tags
racoon	
Dock	
WindowServer	
kernel	
configd	
...orderHelper	
sandbox	
mdworker	
com.apple.calendar.store.log-caldav.queue	
► 11:52:34 AM sandbox: [(78901) mdworker(78901) deny mach-lookup com.apple.coreSymbolicationd	
11:52:34 AM kernel: Sandbox: sandbox(78902) deny mach-lookup com.apple.coreSymbolicationd	
► 11:52:35 AM CalendarAgent: [com.apple.calendar.store.log-caldav.queue] [Account refresh failed	

# Console.app Message Inspector

The screenshot shows the Mac OS X Console application interface. On the left, a list of syslog messages is displayed. On the right, a detailed "Message Inspector" window is open, showing metadata for a selected message. The message in the inspector is:

Key	Value
ASLExpireTime	1368747864
ASLMessageID	3548564
Facility	com.apple.system.lastlog
GID	0
Host	byte
Level	5
PID	39488
ReadGID	80
Sender	sshd
Time	1337125464
TimeNanoSec	436116000
UID	0
ut_host	bit
ut_id	s001
ut_line	ttyS001
ut_pid	39491
ut_tv.tv_sec	1337125464
ut_tv.tv_usec	420174
ut_type	7
ut_user	oompa
Message	USER_PROCESS: 39491 ttys001

At the bottom left, there is a copyright notice: © SANS, All Rights Reserved.

This view shows the “All Messages”, syslog entries alongside the Message Inspector window. This window shows the additional metadata included with each syslog message. This metadata can be extracted from the raw data as shown in a later slide.

Message Inspector	
Key	Value
ASLExpireTime	1368747864
ASLMessageID	3546564
com.apple.system.lastlog	
GID	0
Host	byte
Level	5
PID	39488
ReadGID	80
Sender	sshd
Time	1337125464
TimeNanoSec	436116000
UID	0
ut_host	bit
ut_id	s001
ut_line	ttyS001
ut_pid	39491
ut_tv.tv_sec	1337125464
ut_tv.tv_usec	420174
ut_type	7
ut_user	oompa
Message	USER_PROCESS: 39491 ttys001
sshd: DEAD_PROCESS: 49332 ttys001	
ssh: DEAD_PROCESS: 39491 ttys001	
sshd: DEAD_PROCESS: 39491 ttys001	
sshd: DEAD_PROCESS: 49332 ttys001	
sshd: DEAD PROCESS: 49332 ttys001	

# Log Normalization

Correlate data in a single system or across multiple systems

Must know “originating” time zone for system

## Timestamp Storage

- Apple System Log = UTC
- Most other logs (/var/log, ~/Library/Logs/) = Local System Time

## Timestamp Output

- ASL Logs – praudit may output to local system time
- Use `export TZ="EST5EDT"` command
- Temporarily change time zone of terminal window

Analysts often have to correlate information across multiple systems across different time zones. To combine and understand what each system was doing at the same time, the analyst will need to normalize the log.

Each log type may store its timestamp in various formats while some output tools will export this timestamp in a different time zone or using local system time.

One example of this is the `praudit` tool to read BSM audit logs. This tool outputs XML output with a timestamp in local system time. The analyst must know the original time zone of the system to correctly normalize the `praudit` output.

An investigator can temporarily change the time zone of the Terminal window by using the command `export TZ="EST5EDT"` command with the correct time zone (found in `/usr/share/zoneinfo/`). This time zone change will be removed once you exit out of that Terminal (login) session.

# Apple System Log

- Location: /private/var/log/asl/ (>10.5.6)
- syslog “replacement” (Still uses syslog backend)
- View using Console.app or `syslog` command
- Binary Format – “ASL DB” Signature
- Log Turn Over - 7 Days, ~1 Year (utmp)

```
4153 4c20 4442 0000 0000 0000 0000 0002 ASL DB.....
0000 0000 0000 00f6 0000 0000 51a2 054b .....0..K
0000 0100 0000 0000 0003 6c3a 0000 0000 .....l:....
0000 0000 0000 0000 0000 0000 0000 0000 .....
0000 0000 0000 0000 0000 0000 0000 0000 .....
0001 0000 007b 6861 6e64 6c65 5f77 696c ....{handle_wil
6c5f 736c 6565 705f 6175 7468 5f61 6e64 l_sleep_auth_and
5f73 6869 656c 645f 7769 6e64 6f77 733a _shield_windows:
2072 656c 6561 7369 6e67 2061 7574 6877 releasing authw
2030 7837 6662 3562 6663 3034 3932 3028 0x7fb5bfc04920(
3230 3030 292c 2073 6869 656c 6420 3070 2000), shield 0x
3766 6235 6262 6365 6362 3130 2832 3030 7fb5bbcecb10(200
3129 2c20 6c6f 636b 2073 7461 7465 2033 1), lock state 3
```

© SANS,  
All Rights Reserved

Mac Forensic Analysis

The Apple System Log is Apple’s version of the Unix SYSLOG. After 10.5.6, the ASL data is located in the `/var/log/asl` directory in a proprietary binary format (rather than plaintext). These messages can be viewed using the `Console.app` or the `syslog` command line utility.

The screenshot shows the signature for an ASL log, “ASL DB” which will be found in the first six bytes of each log file.

The default time-to-live (TTL) for SYSLOG messages is seven days, while the default TTL for utmp, wtmp, and lastlog messages (i.e., logon/logoff/boot/shutdown/restart) is one year (366 days or 31622400 seconds via man `asl.conf`).

References:

`asl.conf` Man Page

<https://developer.apple.com/library/mac/documentation/Darwin/Reference/ManPages/man5/asl.conf.5.html>

# Apple System Log File Names

- **Filename Format:**  
`YYYY.MM.DD.[UID].[GID].asl`
- **BB – Best Before**
- **AUX - Auxiliary**

```
nibble:AUX.2013.05.28 sledwards$ pwd
/var/log/asl/AUX.2013.05.28
nibble:AUX.2013.05.28 sledwards$ ls
281501 281597 281592 281790 281884
281503 281599 281698 281792 281886
281505 281604 281700 281794 281892
281511 281606 281702 281801 281894
281513 281608 281708 281803 281896
281515 281614 281710 281805 281902
281521 281616 281712 281810 281904
281523 281618 281718 281812 281906
281525 281624 281721 281814 281912
281531 281626 281723 281820 281914
```

```
May 28 23:57 2013.05.28.000.asl
May 28 23:59 2013.05.28.U0.G00.asl
May 28 23:49 2013.05.28.U0.asl
May 28 22:15 2013.05.28.U501.asl
May 29 23:58 2013.05.29.00.G00.asl
May 29 23:59 2013.05.29.00.G00.asl
May 29 22:45 2013.05.29.U0.asl
May 29 23:21 2013.05.29.U501.asl
May 30 23:57 2013.05.30.G00.asl
May 30 23:57 2013.05.30.U0.G00.asl
May 30 23:49 2013.05.30.U0.asl
May 30 22:41 2013.05.30.U501.asl
May 31 23:59 2013.05.31.G00.asl
May 31 23:59 2013.05.31.U0.G00.asl
May 31 22:52 2013.05.31.U0.asl
May 31 23:00 2013.05.31.U501.asl
Jun 1 23:59 2013.06.01.G00.asl
Jun 1 23:59 2013.06.01.U0.G00.asl
Jun 1 23:17 2013.06.01.U0.asl
Jun 1 21:45 2013.06.01.U501.asl
Jun 2 23:58 2013.06.02.U0.G00.asl
Jun 2 23:58 2013.06.02.U0.asl
Jun 2 23:06 2013.06.02.U0.asl
Jun 2 21:22 2013.06.02.U501.asl
Jun 3 20:08 2013.06.03.G00.asl
Jun 3 20:08 2013.06.03.U0.G00.asl
Jun 3 19:21 2013.06.03.U0.asl
Jun 3 10:57 2013.06.03.U200.asl
Jun 3 19:55 2013.06.03.U501.asl
May 28 23:56 AUX.2013.05.28
May 29 23:57 AUX.2013.05.29
May 30 23:56 AUX.2013.05.30
May 31 23:58 AUX.2013.05.31
Jun 1 23:58 AUX.2013.06.01
Jun 2 23:58 AUX.2013.06.02
Jun 3 20:08 AUX.2013.06.03
Mar 30 09:59 BB.2014.03.31.G00.asl
Apr 25 17:35 BB.2014.04.30.G00.asl
May 29 20:52 BB.2014.05.31.G00.asl
```

© SANS.  
All Rights Reserved

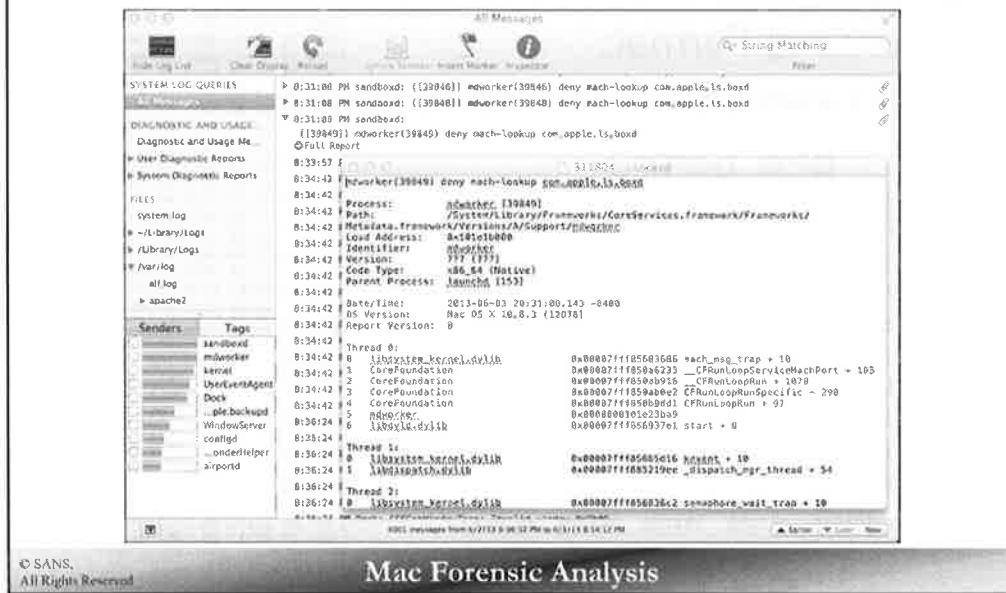
Mac Forensic Analysis

The ASL filename format is standardized to match the month, day, and year the data was recorded. The files are also separated by user and groups IDs. These logs will usually be seen for only the past seven days, after the utmp, wtmp, and lastlog messages get rolled into the ASL log files beginning with “BB”, or “Best Before”.

The “Best Before” log files contains the login records for the month listed in the filename. These records will be kept around for one year.

The directories starting with AUX or “Auxiliary” are used on 10.8+ systems. These directories contain text files comprising of additional data that is referenced by the syslog files for the past week. In my experience these files show information pertaining to the sandboxd process, or Apple Sandboxing.

# Apple System Logs Auxiliary Files



The screenshot above shows how the auxiliary files interface with the `syslog` files. In the Console application the messages with the paperclip “attachment” icon contain additional information. The “Full Report” links to the additional data in the auxiliary directory.

All Messages	
Hide Log List  Clear Display  Reload  Ignore Sender  Insert Marker  Inspector	
All Messages	All Messages
DIAGNOSTIC AND USAGE...	► 8:31:08 PM sandboxd: ([39846]) mdworker(39846) deny mach-lookup com.apple.ls.boxd
Diagnostic and Usage Me...	► 8:31:08 PM sandboxd: ([39848]) mdworker(39848) deny mach-lookup com.apple.ls.boxd
User Diagnostic Reports	▼ 8:31:08 PM sandboxd: ([39849]) mdworker(39849) deny mach-lookup com.apple.ls.boxd
System Diagnostic Reports	● Full Report
FILES	
system.log	8:33:57 F [311804 - Locked]
~ /Library/Logs	8:34:42 F mdworker(39849) deny mach-lookup com.apple.ls.boxd
/Library/Logs	8:34:42 F Process: mdworker [39849] /System/Library/Frameworks/CoreServices.framework/Frameworks/
/var/log	8:34:42 F Path: /System/Library/Versions/A/Support/mdworker. Load Address: 0x101e1b000
als.log	8:34:42 F Identifier: mdworker Version: ??? (???)
apache2	8:34:42 F Code Type: x86_64 (Native) Parent Process: launchd [153]
	8:34:42 F Date/Time: 2013-06-03 20:31:08.143 -0400
	OS Version: Mac OS X 10.8.3 (12D78)
	8:34:42 F Report Version: 8
Senders	Tags
sandboxd	8:34:42 Thread 0:
mdworker	8:34:42 0 libsystem_kernel.dylib
kernel	8:34:42 1 CoreFoundation
UserEventAgent	8:34:42 2 CoreFoundation
Dock	8:34:42 3 CoreFoundation
...plebackupd	8:34:42 4 CoreFoundation
WindowServer	8:36:24 5 mdworker
configd	8:36:24 6 libdyld.dylib
...orderHelper	8:36:24 Thread 1: 0 libsystem_kernel.dylib
airportd	8:36:24 1 libdispatch.dylib
	8:36:24 Thread 2: 0 libsystem_kernel.dylib
	0x000007fff856836c2 semaphore_wait_trap + 10
	0x000007fff856836d16 keyent + 10
	0x000007fff885219ee _dispatch_mgr_thread + 54
	0x000007fff856836c0

# Apple System Log Record Format

The screenshot shows a list of log messages from the Apple System Log and a detailed view of one message using the 'Message Inspector' tool.

**Log Messages:**

```

5/7/13 9:27:03 PM login: DEAD_PROCESS: 97280 ttys002
5/7/13 9:27:03 PM login: DEAD_PROCESS: 97313 ttys004
5/7/13 10:03:36 PM login: USER_PROCESS: 98679 ttys002
5/9/13 7:04:01 PM login: USER_PROCESS: 4500 ttys004
5/9/13 7:04:01 PM login: DEAD_PROCESS: 4500 ttys004
5/9/13 9:13:38 PM login: USER_PROCESS: 4969 ttys004
5/10/13 6:18:23 PM login: DEAD_PROCESS: 4969 ttys004
5/10/13 9:00:19 PM login: USER_PROCESS: 7960 ttys004
5/10/13 9:10:51 PM login: DEAD_PROCESS: 7960 ttys004
5/16/13 10:29:47 PM login: USER_PROCESS: 25177 ttys004
5/16/13 10:29:59 PM login: DEAD_PROCESS: 76504 ttys003
5/16/13 10:36:26 PM login: USER_PROCESS: 37534 ttys003
5/18/13 10:23:22 PM login: DEAD_PROCESS: 76647 ttys000
5/18/13 10:23:22 PM login: DEAD_PROCESS: 91613 ttys001
5/18/13 10:23:22 PM login: DEAD_PROCESS: 25177 ttys004
5/18/13 10:23:22 PM login: DEAD_PROCESS: 98679 ttys002
5/18/13 10:23:22 PM login: DEAD_PROCESS: 37534 ttys003
5/10/13 10:23:26 PM loginwindow: DEAD_PROCESS: 55 console
5/18/13 10:25:07 PM loginwindow: USER_PROCESS: 59 console
5/18/13 10:25:08 PM login: USER_PROCESS: 236 ttys000
5/10/13 10:25:09 PM login: USER_PROCESS: 246 ttys001
5/10/13 10:25:09 PM login: USER_PROCESS: 254 ttys002
5/10/13 10:25:09 PM login: USER_PROCESS: 259 ttys003

```

**Message Inspector:**

Key	Value
ASLExpireTime	1399856419
ASLMessageID	220267
Facility	com.apple.system.lastlog
GID	20
Host	nibble.blah
Level	5
PID	7960
ReadGID	80
Sender	login
Time	1368234019
TimeNanoSec	920375000
UID	0
:ut_id	5004
:ut_line	ttys004
:ut_pid	7960
:ut_tv.tv_sec	1368234019
:ut_tv.tv_usec	918722
:ut_type	7
:ut_user	sledwards
Message	USER_PROCESS: 7960 ttys004

© SANS.  
All Rights Reserved

Mac Forensic Analysis

Each log message contains certain Apple System Log keys. Not all of these keys will be in each message.

- **ASLExpireTime** – Message Expire Timestamp, this is when the message is explicitly set to expire
- **ASLMessageID** – Message ID Number
- **Time** – Timestamp of the record
- **TimeNanoSec** – Nanoseconds recorded
- **Host** – Hostname of the system the message was recorded on
- **Sender** – Default process name or identification string
- **Facility** – Default is “user”, otherwise noted in reverse DNS format
- **PID** – Process ID
- **UID** – User ID
- **GID** – Group ID
- **Level** – Message priority level
- **Message** – Log Message
- **ReadUID** – Read access for user
- **ReadGID** – Read access for group
- **Session** – Session by launchd
- **RefPID** – Reference Process ID (launchd)
- **RefProc** – Reference Process Name (launchd)
- **ASLAuxTitle** – Title (usually “Full Report”)
- **ASLAuxUTI** – ASL auxiliary uniform type identifier
- **ASLAuxURL** – URL to auxiliary file

## References:

syslog Man Page

ASL Man Page

Asl.h - <http://www.opensource.apple.com/source/Libc/Libc-583/include/asl.h>

## syslog Command

<b>Output Format (-F)</b> <ul style="list-style-type: none"> <li><code>bsd</code></li> <li><code>std</code></li> <li><code>raw</code></li> <li><code>xml</code></li> </ul>	<b>Time Format (-T)</b> <ul style="list-style-type: none"> <li><code>sec</code></li> <li><code>local</code></li> <li><code>utc</code></li> </ul>	<b>File or Directory</b> <ul style="list-style-type: none"> <li><code>-f</code></li> <li><code>-d</code></li> </ul>
--	--	---

```

sh-3.2# syslog -d asl/ | more
Mar 12 17:15:01 byte login[63585] <Notice>: USER_PROCESS: 63585 ttys003
Mar 15 01:41:32 byte login[48848] <Notice>: USER_PROCESS: 48848 ttys004
Mar 15 01:44:22 byte login[48905] <Notice>: USER_PROCESS: 48905 ttys005
Mar 15 01:52:19 byte login[48848] <Notice>: DEAD_PROCESS: 48848 ttys004
Mar 15 01:52:19 byte login[48905] <Notice>: DEAD_PROCESS: 48905 ttys005
Mar 15 01:52:21 byte login[48960] <Notice>: USER_PROCESS: 48960 ttys004
Mar 15 01:53:16 byte login[48960] <Notice>: DEAD_PROCESS: 48960 ttys004
Mar 15 01:53:18 byte login[50861] <Notice>: USER_PROCESS: 50861 ttys004
Mar 15 01:53:52 byte login[50861] <Notice>: DEAD_PROCESS: 50061 ttys004
Mar 15 01:53:53 byte login[52753] <Notice>: USER_PROCESS: 52753 ttys004
Mar 15 01:54:19 byte login[53625] <Notice>: USER_PROCESS: 53625 ttys005

```

© SANS,  
All Rights Reserved

Mac Forensic Analysis

A single ASL file or a directory containing multiple ASL files can be used as input into the `syslog` command.

The `syslog` command can be used to output the data in a variety of formats:

- `bsd` – Similar to other system logs such as `system.log`
- `std` – Same as `bsd`, and includes message priority level (default)
- `raw` – Message fields are in square brackets, in key/value format
- `xml` – XML property list

The time output can also be formatted:

- `sec` – Number of seconds since epoch
- `local` – Local time zone (default)
- `utc` – UTC format

In the example shown above, the `syslog` command is taking in the default directory of ASL files and outputting the data in the default standard format.

```
syslog -T utc -F raw -d /asl
```

```
[ASLMessageID 3555356] [Time 2012.05.28 19:39:32 UTC] [TimeNanoSec 887175000] [Level 5] [PID 908] [UID 0] [GID 20] [Host byte] [Sender login] [Facility com.apple.system.utmpx] [Message DEAD_PROCESS: 908 ttys002] [ut_pid 908] [ut_type 8] [ut_tv.tv_sec 1338233972] [ut_tv.tv_usec 886961] [ASLExpireTime 1369856372] [ut_line ttys002] [ut_user oompa] [ut_id s002]  
[ASLMessageID 23869] [Time 2013-03-17 20:12:49Z] [TimeNanoSec 649773000] [Level 5] [PID 21931] [UID 0] [GID 20] [ReadGID 80] [Host nibble.blah] [Sender login] [Facility com.apple.system.utmpx] [Message DEAD_PROCESS: 21931 ttys003] [ut_user sledwards] [ut_id s003] [ut_line ttys003] [ut_pid 21931] [ut_type 8] [ut_tv.tv_sec 1363551169] [ut_tv.tv_usec 647288] [ASLExpireTime 1395173569] [ASLMessageID 28599] [Time 2013-03-23 00:10:53Z] [TimeNanoSec 859756000] [Level 5] [PID 28599] [UID 0] [GID 20] [ReadGID 80] [Host nibble.blah] [Sender login] [Facility com.apple.system.lastlogin] [Message USER_PROCESS: 28599 ttys003] [ut_user sledwards] [ut_id s003] [ut_line ttys003] [ut_pid 28599] [ut_type 7] [ut_tv.tv_sec 1363997453] [ut_tv.tv_usec 859054] [ASLExpireTime 139561985]
```

© SANS.  
All Rights Reserved

Mac Forensic Analysis

The raw output for `syslog` labels each key/value pair in square brackets as shown above. All dates are stored in Unix epoch time. The fields starting with “`ut_`” are a throwback to the `utmp` login data that has been deprecated in OS X since 10.5.

```
[ASLMessageID 23869] [Time 2013-03-17 20:12:49Z] [TimeNanoSec 649773000] [Level 5] [PID 21931] [  
uid 0] [GID 20] [ReadGID 80] [Host nibble.blah] [Sender login] [Facility com.apple.system.utmpx]  
[Message DEAD_PROCESS: 21931 ttys003] [ut_user sledwards] [ut_id s003] [ut_line ttys003] [ut_pi  
d 21931] [ut_type 8] [ut_tv.tv_sec 1363551169] [ut_tv.tv_usec 647288] [ASLExpireTime 1395173569]  
[ASLMessageID 28599] [Time 2013-03-23 00:10:53Z] [TimeNanoSec 859756000] [Level 5] [PID 28503] [  
uid 0] [GID 20] [ReadGID 80] [Host nibble.blah] [Sender login] [Facility com.apple.system.lastlo  
g] [Message USER_PROCESS: 28503 ttys003] [ut_user sledwards] [ut_id s003] [ut_line ttys003] [ut_  
pid 28503] [ut_type 7] [ut_tv.tv_sec 1363997453] [ut_tv.tv_usec 859054] [ASLExpireTime 139561985  
3]
```

# Audit Logs

## /private/var/audit/\*

- Basic Security Module (BSM) Audit Logs
- Binary Format

```
5h-3.2# xxd 20130307232230.20130308000749
00000000: 1400 0000 7dbb af67 0000 5139 2136 0000 ....}..g..09!6..
00000100: 02ed 7101 0000 0000 0000 0007 7355 ..q.....sf
00000200: 6c61 6773 002d 0200 0000 0000 0b61 6d5f logs.=.....am_
00000300: 7375 6363 6573 7300 2403 0000 0000 000b success.=.....
00000400: 616d 5f66 6169 6c75 7265 0024 ffff ffff am_failure.$...
00000500: 0000 0000 0000 0000 0000 0000 0000 0000 .....'.
00000600: 0000 0000 0001 8703 0000 0000 0000 0000 .....'.
00000700: 2700 0000 0000 13b1 0500 0000 7d14 0000 .....}.
00000800: 087d 0ba1 6800 0051 392b d408 0003 e771 .).h.Q9+....q
00000900: 8100 0000 0000 0000 0000 0773 666c 6167 .....$flag
00000a00: 7300 2d02 0000 0000 0000 616d 5f73 7563 s.=.....am_suc
00000b00: 6365 7373 002d 0300 0000 0000 0b61 6d5f cess.=.....am_
00000c00: 6561 696c 7572 6500 24ff ffff f100 0000 failure.$...
00000d00: 0000 0000 0000 0000 0000 0000 0000 0000 .....'.
00000e00: 0000 0187 0300 0000 0000 0027 0000 .....'.
00000f00: 0000 0013 b105 0000 007d 1400 0000 7d0b .....}.
00001000: af65 0000 5139 2bd5 0000 0000 7101 0000 .e..09+....q...
00001100: 0000 0000 0000 0007 7366 6c61 6773 002d .....$flags-
00001200: 0200 0000 0000 0b61 605f 7375 6363 6573 .....am_sucess
00001300: 7300 2d03 0000 0000 000b 616d 5f66 6169 s.=.....am_fai
00001400: 6c75 7265 0024 ffff ffff 0000 0000 0000 ture.$...
00001500: 0000 0000 0000 0000 0000 0000 0000 0001 .....'.
00001600: 8705 0000 0000 0000 0000 2700 0000 0000 .....'.
00001700: 13b1 0500 0000 7d .....}'
```

© SANS,  
All Rights Reserved

Mac Forensic Analysis

The audit logs are one of the few logs not located in the main log directories. These logs are located in the /var/audit/ directory.

The audit logs use the Basic Security Module from OpenBSM (McAfee Research) based upon the definitions created by Sun Microsystems. It has now been taken over by the TrustedBSD project.

The log data is stored in a binary format that can be viewed using tools native to OS X.

More information about BSM Audit logs can be found in Hal Pomeranz's article "Solaris Basic Security Mode (BSM) Auditing" - [www.deer-run.com/~hal/sysadmin/SolarisBSMAuditing.html](http://www.deer-run.com/~hal/sysadmin/SolarisBSMAuditing.html)

```

sh-3.2# xxd 20130307232230.20130308000749
0000000: 1400 0000 7d0b af67 0000 5139 2136 0000 ....}..g..Q9!6..
0000010: 02ed 7101 0000 0000 0000 0000 0007 7366 ..q.....sf
0000020: 6c61 6773 002d 0200 0000 0000 0b61 6d5f lags.-.....am_
0000030: 7375 6363 6573 7300 2d03 0000 0000 000b success.-.....
0000040: 616d 5f66 6169 6c75 7265 0024 ffff ffff am_failure.$....
0000050: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000060: 0000 0000 0001 8703 0000 0000 0000 0000 .....
0000070: 2700 0000 0000 13b1 0500 0000 7d14 0000 '.....}...
0000080: 007d 0baf 6800 0051 392b d400 0003 e771 .}..h..Q9+....q
0000090: 0100 0000 0000 0000 0000 0773 666c 6167 .....sflag
00000a0: 7300 2d02 0000 0000 000b 616d 5f73 7563 s.-.....am_suc
00000b0: 6365 7373 002d 0300 0000 0000 0b61 6d5f cess.-.....am_
00000c0: 6661 696c 7572 6500 24ff ffff ff00 0000 failure.$....
00000d0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000e0: 0000 0187 0300 0000 0000 0000 0027 0000 .....'...
00000f0: 0000 0013 b105 0000 007d 1400 0000 7d0b .....}....}.
0000100: af65 0000 5139 2bd5 0000 0000 7101 0000 .e..Q9+....q...
0000110: 0000 0000 0000 0007 7366 6c61 6773 002d .....sflags.-.
0000120: 0200 0000 0000 0b61 6d5f 7375 6363 6573 .....am_succes
0000130: 7300 2d03 0000 0000 000b 616d 5f66 6169 s.-.....am_fai
0000140: 6c75 7265 0024 ffff ffff 0000 0000 0000 lure.$.....
0000150: 0000 0000 0000 0000 0000 0000 0000 0001 .....
0000160: 8705 0000 0000 0000 0000 2700 0000 0000 .....'.....
0000170: 13b1 0500 0000 7d .....}

```

## Audit Logs – Audit Trail Files

- StartTime.EndTime
- YYYYMMDDHHMMSS.YYYYMMDDHHMMSS
- Other Filenames:
  - “current”
  - \*.not\_terminated
  - \*.crash\_recovery

```
drwx----- 8 root wheel 272 May 28 15:22 .
drwxr-xr-X 29 root wheel 986 May 9 21:39 ..
-r--f---- 1 root wheel 48987 May 10 00:46 20120509232853.20120510044637
-r--f---- 1 root wheel 57158 May 12 11:31 20120510204054.20120512153135
-r--f---- 1 root wheel 92166 May 27 20:02 20120512153220.20120528000216
-r--f---- 1 root wheel 20805 May 28 15:20 20120528000250.20120528192006
-r--f---- 1 root wheel 4619 May 28 21:07 20120528192235.not_terminated
lrwxr-xr-x 1 root wheel 40 May 28 15:22 current -> /var/audit/20120528192235.not_terminated
```

© SANS,  
All Rights Reserved.

Mac Forensic Analysis

The log directory should contain many log files, each with a standard naming scheme (StartTime.EndTime) in the format YYYYMMDDHHMMSS. In the screenshot, the first log file (20120509232853.20120510044637) would contain data for the time period in between 05/09/2012 23:28:53 to 05/10/2012 04:46:37. Other file names may also be used to show incompleteness or system error.

The “current” audit trail file will be a symbolic link to the active audit trail file, as shown in the screenshot above.

Those files ending with “not\_terminated” are trail files that were not terminated properly due to a system error, or the file was otherwise inaccessible. The current audit trail that is in use will always have the “not\_terminated” file extensions.

Audit trail files ending in “crash\_recovery” are those files that were not terminated properly due to system or audit crash. The next audit trail file will have an “audit crash recovery” as the first record.”

# Audit Logs – Configuration Files /etc/security/\*

## audit\_class

- Auditable Events (Login/logout, authorization)

## audit\_control

- Auditing parameters (file size, expiration, directory)

## audit\_user

- User specific auditing configuration

## audit\_event

- Audit event descriptions

© SANS,  
All Rights Reserved

Mac Forensic Analysis

### **audit\_class:**

The `audit_class` file contains the auditable event class designations. Each of these will be used to determine which events are audited for the system or for a specific user.

### **audit\_control:**

The `audit_control` file contains the configuration data for the audit process.

This file contains different parameters. The screenshot above shows the default configuration for a 10.8 system:

- **dir** – Directory where audit trail files are stored.
- **flags** – Specifies audit event classes for a user (see `audit_class` file). `lo` = login/logout events, `aa` = audit administrative events
- **minfree** – Minimum space (percentage) required on volume where audit logs are contained.
- **naflags** – Specifies audit event classes that are not attributable to a user (see `audit_class` file). “`naflags`” = non-attributable flags
- **policy** – Global audit policy flags
- **filesz** – Maximum audit trail file size (2 Megabytes)
- **expire-after** – Expire audit trail files after a certain amount of time or size (Expire after 10 Megabytes of audit trail files)

**audit\_user:**

Each user can have specific audit functions recorded. The default configuration only includes one user – root.

The format for each line is `username:alwaysaudit:neveraudit`, where the first parameter is the username, the second are those classes that should be audited, and the third are those classes that should not be audited.

**audit\_event:**

The `audit_event` file contains auditable event types that are classified into various event classes. The entries in this file follow the format `eventnum: eventName:description:eventClass`.

- **eventnum** – Unique number for the event
- **eventName** – Event Name
- **description** – Event Description
- **eventclass** – Event class (see `audit_class`)

**References:**

`audit_class` Man Page  
`audit_control` Man Page  
`audit_user` Man Page  
`audit_event` Man Page

```
praudit -xn /var/audit/*
```

## su Example:

```
<record version="11" event="user authentication" modifier="0" time="Mon May 28 21:12:51 2012" msec=" + 41 msec" >
<subject audit-uid="501" uid="0" gid="20" ruid="501" rgid="20" pid="552" sid="100004" tid="552 0.0.0.0" />
<text>Verify password for record type Users &apos;root&apos; node &apos;/Local/Default&apos;</text>
<return errval="success" retval="0" />
</record>

<record version="11" event="user authentication" modifier="0" time="Mon May 28 21:12:55 2012" msec=" + 449 msec" >
<subject audit-uid="501" uid="0" gid="20" ruid="501" rgid="20" pid="554" sid="100004" tid="554 0.0.0.0" />
<text>Verify password for record type Users &apos;root&apos; node &apos;/Local/Default&apos;</text>
<return errval="failure: Unknown error: 255" retval="5000" />
</record>
```

© SANS,  
All Rights Reserved

Mac Forensic Analysis

The praudit (i.e., print audit) command can be used to view the audit log files. This command is native to OS X.

The praudit command shown above uses the `-xn` options to print the audit records in XML format (`-x`) and does not convert the user and group IDs (`-n`).

Shown above, are two separate records. The first record contains the data consistent with a successful `su` logon, while the second contains a failed `su` logon.

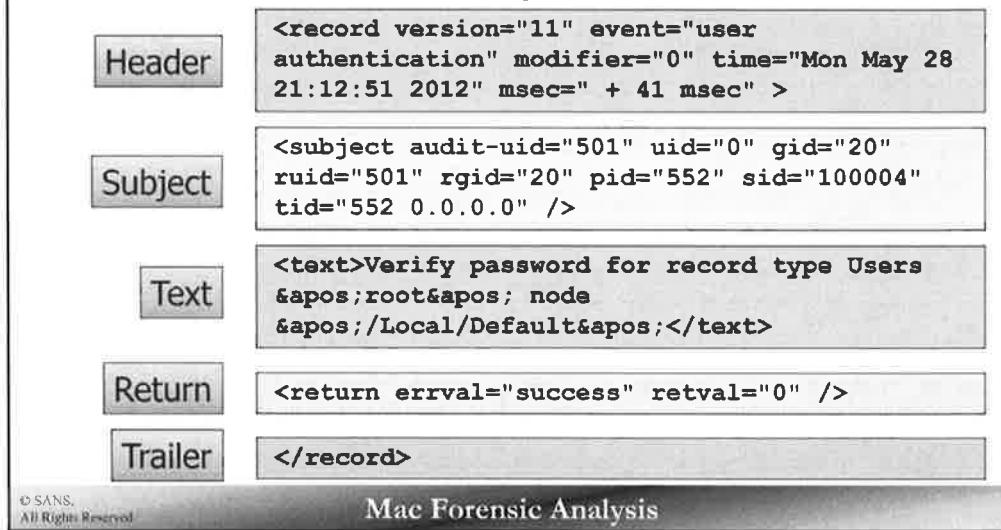
The event identifier “user authentication” can be used to determine logon activities of users.

In the first record, the highlighted “ruid” key contains the UID for user 501 (usually the first user account created on the system). This user is attempting to use the ‘`su`’ command to get root privileges as shown in the “text” key. The “return” key shows that it was successful.

In the second record the same event occurred, but returned with a “failure:Unknown error: 255” error. This error type is returned at a failed ‘`su`’ login.

# Audit Log Records

- Each record is made up of “tokens”



As shown previously, each audit record is made up of various components or ‘tokens’.

The example above contains five of the basic tokens needed for each record. Each record may contain different tokens depending on the contents of the data.

- **Header** – Required Token. Is used to mark the start of a record. It contains data such as the record version, event type/modifier, timestamp and record length. The length is not shown above due to how the record was printed.
- **Subject** – This token contains data associated with the “subject” making the operation. This record will have the audit user ID, effective user ID, effective group ID, real user ID, real group ID, process ID, session ID, and terminal ID. The real user IDs are generally the user doing executing process (UID 501), while the effective user ID is the user the process is running under (i.e., root - UID 0)
- **Text** – The Text token contains a string with a description of the event.
- **Return** – The Return token contains a return value that may be used by the system.
- **Trailer** – Required Token. This contains the record termination and may also contain a magic number or byte count depending on how the record is printed.

# Audit Log Record - Tokens

## Variable number of tokens

### Subject Token

The ``subject'' token contains information on the subject performing the operation described by an audit record, and includes similar information to that found in the ``process'' and ``expanded process'' tokens. However, those tokens are used where the process being described is the target of the operation, not the authorizing party. A ``subject'' token can be created using `au_to_subject32(3)` and `au_to_subject64(3)`.

Field	Bytes	Description
Token ID	1 byte	Token ID
Audit ID	4 bytes	Audit user ID
Effective User ID	4 bytes	Effective user ID
Effective Group ID	4 bytes	Effective group ID
Real User ID	4 bytes	Real user ID
Real Group ID	4 bytes	Real group ID
Process ID	4 bytes	Process ID
Session ID	4 bytes	Audit session ID
Terminal Port ID	4/8 bytes	Terminal port ID (32/64-bits)
Terminal Machine Address	4 bytes	IP address of machine

© SANS.  
All Rights Reserved.

Mac Forensic Analysis

More information about each token can be found on the `audit.log` man page.

Example from the man page for the subject token.

References:

[audit.log Man Page](#)

## auditreduce Command

Filter audit records given:

- Before or after a date/time
- A specific user
- A specific subject token
- An audit event (shown below)

```
bash-3.2# auditreduce -m AUE_lw_login 20140716023538.20140725213130 | praudit -xn
<?xml version='1.0' encoding='UTF-8'?>
<audit>
<record version="11" event="loginwindow login" modifier="0" time="Tue Jul 15 22:47:21 2014" msec=" + 473 msec" >
<subject audit-uid="501" uid="0" gid="0" ruid="501" rgid="20" pid="73" sid="100004" tid="503316500,0.0.0" />
<return errval="success" retval="0" />
</record>
```

© SANS,  
All Rights Reserved

Mac Forensic Analysis

BSM audit records can be filtered using the auditreduce command.

The screenshot shows a filter for a specific audit event to find login window logins (AUE\_lw\_login). The auditreduce command can be piped to the praudit command for easier viewing.

The audit events can be found in /etc/security/audit\_event.

References:

[auditreduce Man Page](#)

## Log Recovery

- Logs get “removed” or “turned over”
- GREP or keyword search for specific date/log formats.
  - “May 18 23:17:15”
  - “Thu May 31 19:35:35 EDT 2012”
  - “ASL DB”
  - “launchctl::Audit startup”
  - “BZh91AY&SY”
  - “1F8B08”

In the eventuality that log files have been archived or removed, they may be able to be recovered.

- A grep search for date formats of the specific log you are looking for may be helpful.
- Log signatures for binary logs such as “ASL DB” for the Apple System Log can be searched for.
- For archived files, the BZip2 signature should be searched, “BZh91AY&SY”
- For gzipped archived files, use “1F8B08”.



## Exercise 3.2 – Log Parsing & Analysis

© SANS,  
All Rights Reserved

Mac Forensic Analysis

This page intentionally left blank.

# Agenda

---

Part 1 – System Information

Part 2 – System Preferences

Part 3 – Log Analysis

Part 4 – Timeline Analysis & Data Correlation

This page intentionally left blank.



COMPUTER FORENSICS  
and INCIDENT RESPONSE



## Section 3 – Part 4

### Timeline Analysis & Data Correlation

© SANS  
All Rights Reserved

Mac Forensic Analysis

This page intentionally left blank.

# Timeline Analysis & Data Correlation

Temporal Context & Timestamps

Volume Analysis

Temporal Changes

System Information & State

Network Analysis

User Access

Privilege Escalation

Account Creation/Deletion

Software Installation

Backup Activity

Locational Data

© SANS,  
All Rights Reserved

Mac Forensic Analysis

This page intentionally left blank.

# Temporal Context

- Carved & Extracted Files
- May not contain context
  - Year
  - Time Zone

```
Jun 19 07:13:14 bit kernel[0]: PPTP domain init
Jun 19 07:13:16 bit kernel[0]: nd6_setmtu: new link MTU on ppp0 (1276) is too small for IPv6
Jun 19 07:13:42 bit kernel[0]: IOSurface: buffer allocation size is zero
Jun 19 07:10:55 bit kernel[0]: hibernate image path: /var/vm/sleepimage
Jun 19 07:19:55 bit kernel[0]: sizeof(IOHibernateImageHeader) == 512
Jun 19 07:19:55 bit kernel[0]: Opened file /var/vm/sleepimage, size 8589934592, partition base 0x0, maxio 400000 ssd 0
Jun 19 07:19:55 bit kernel[0]: hibernate image major 14, minor 0, blocksize 512, pollers 4
Jun 19 07:19:55 bit kernel[0]: hibernate_alloc_pages flags 00000000, gobbling 0 pages
Jun 19 07:19:55 bit kernel[0]: hibernate_setup(0) took 0 ms
Jun 19 07:10:55 bit kernel[0]: en1: BSSID changed to 00:19:07:96:03:10
Jun 19 07:19:55 bit kernel[0]: wlEvent: en1 en1 Link DOWN virtIf = 0
Jun 19 07:19:55 bit kernel[0]: AirPort: Link Down on en1. Reason 8 (Disassociated because station leaving).
```

© SANS.  
All Rights Reserved

Mac Forensic Analysis

Most of the log files on OS X are based upon standard Unix log formats, which do not always contain the year the event happened. While the MAC times of a log file may provide that information to current log files, files extracted from unallocated space do not have these timestamps.

The example shows a log that contains some entries which might be of interest. These entries happened on June 19<sup>th</sup>, but we do not know what year. We can look at the content of a particular entry to gain perspective as to what year the event likely occurred. Viewing the log entries just before and just after the entries in question could be your best bet.

```
[Jun 19 07:13:14 bit kernel[0]:] PPTP domain init link MTU on ppp0 (1276) is too small for IPv6
[Jun 19 07:13:16 bit kernel[0]:] nd6_setmtu: new link MTU on ppp0 (1276) is too small for IPv6
[Jun 19 07:13:42 bit kernel[0]:] IOSurface: buffer allocation size is zero
[Jun 19 07:19:55 bit kernel[0]:] hibernate image path: /var/vm/sleepimage
[Jun 19 07:19:55 bit kernel[0]:] sizeof(IOHibernateImageHeader) == 512
[Jun 19 07:19:55 bit kernel[0]:] Opened file /var/vm/sleepimage, size 8589934592, partition base 0x0, maxio 400000 ssd 0
[Jun 19 07:19:55 bit kernel[0]:] hibernate image major 14, minor 0, blocksize 512, pollers 4
[Jun 19 07:19:55 bit kernel[0]:] hibernate_alloc_pages flags 00000000, gobbling 0 pages
[Jun 19 07:19:55 bit kernel[0]:] hibernate_setup(0) took 0 ns
[Jun 19 07:19:55 bit kernel[0]:] en1: BSSID changed to 00:19:07:96:03:10
[Jun 19 07:19:55 bit kernel[0]:] wlEvent: en1 en1 Link DOWN virtIf = 0
[Jun 19 07:19:55 bit kernel[0]:] AirPort: Link Down on en1. Reason 8 (Disassociated because station leaving).
```

# Date & Time Search

## Epoch & Timestamp Formats

### kernel.log

```
*Jun 19 09:20:16 bit kernel[0]: nspace-handler-set-snapshot-time:  
1340112018  
*Jun 12 10:08:15 bit kernel[0]: RTC: maintenance alarm 2012/6/12  
14:08:14, sleep 2012/6/12 12:08:46
```

### system.log

```
*Jun 13 09:55:31 bit mtmd[64]: Set snapshot time:1339595733  
(current time:1339595731)  
*Jun 12 10:16:35 localhost bootlog[0]: BOOT_TIME 1339510595 0  
*Jun  9 10:21:53 bit shutdown[309]: SHUTDOWN_TIME: 1339251713  
535787  
*Jun 12 17:23:44 bit com.apple.backupd[4046]: Deleted  
/Volumes/Time Machine Backups/Backups.backupdb/bit/2012-06-10-  
012553 (50.5 MB)  
*Jun 12 10:17:42 bit [0x0-0x8008].com.google.Chrome[141]: 2012-06-  
12 14:17:42.785 Google Chrome Helper[196:207] Error received in  
message reply handler: Connection invalid
```

© SANS,  
All Rights Reserved.

Mac Forensic Analysis

Many dates and times are stored in the actual message of the log entry, in many different timestamp formats. An analyst needs to be familiar with these timestamps so they can put context around an unknown event. These date formats can be searched within the unallocated space to find entries that happened on a date of interest for a specific case. Each timestamp type should be searched to find the log data.

The Unix epoch timestamps can be decoded into human readable format by using the `date -ur` command.

## Volume Analysis

USB Volumes

Network Shares

Mounted Volumes

Volume History

© SANS,  
All Rights Reserved

Mac Forensic Analysis

Volume analysis can provide us with information about other items connected to the system or were accessed via the system.

# system.log & daily.out

## Search "/Volumes/"

```
May 19 08:58:23 bit fsevents[20]: log dir: /Volumes/Time Machine Backups/.fsevents getting new uid: 5420A642-0ERC-4B00-B2B4-B94B2B8F5E3F
May 19 16:52:30 bit fsevents[20]: log dir: /Volumes/ND NAME/.fsevents getting new uid: D0649860-F5BC-4870-9110-58027104F862
May 23 20:10:35 bit fsevents[20]: log dir: /Volumes/ND NAME/.fsevents getting new uid: 8D8C8838-0691-43B1-ACEF-8F7F4210120F
May 26 14:01:03 bit fsevents[20]: log dir: /Volumes/WDPassport/.fsevents getting new uid: CDCE4339-A254-4925-A909-97B4553BDAC1
May 26 15:40:38 bit fsevents[20]: log dir: /Volumes/WDPassport/.fsevents getting new uid: D4FFFBA2-16A8-4CB3-8BDE-327C0E1551EC

Fri May 11 17:12:29 EDT 2012
Removing old temporary files:
Cleaning out old system announcements:
Removing stale files from /var/rwho:
Removing scratch fax files
Disk status:
Filesystem          Size   Used  Avail Capacity  Mounted on
/dev/disk0s2        698Gi  22Gi  675Gi    4%       /
localhost:/35wJAmjuh-MSBDh6mJulon 698Gi  698Gi  0Bi     100%    /Volumes/MobileBackups
/dev/disk6s2        107Mi  107Mi  0Bi     100%    /Volumes/Google Chrome

© SANS.
All Rights Reserved
```

Mac Forensic Analysis

Volume analysis may be important to an investigation to see what USB drives were mounted on the system, what software might have been installed, or what the historical usage of a certain volume is.

The term “/Volumes/” can be searched for in the `system.log` and `daily.out` files. The `/Volumes/` directory is the default mount point for any volume that is mounted on the system.

The `system.log` file shows when the volume was automatically mounted and the UUID attached to that volume. It is important to point out that the same USB drive will have a different UUID every time it is mounted on the system.

The `daily.out` file shows what volumes were mounted on the system when the daily maintenance script is run. This log shows the device file the volume is using, the size of the volume, storage utilization, and the volume name and mount point.

**Fri May 11 17:12:29 EDT 2012**

**Removing old temporary files:**

**Cleaning out old system announcements:**

**Removing stale files from /var/rwho:**

**Removing scratch fax files**

**Disk status:**

Filesystem	Size	Used	Avail Capacity	Mounted on
/dev/disk0s2	698Gi	22Gi	675Gi	4% /Volumes/MobileBackups
localhost:/3SwJAmjuh-MSBDDh6mJulon	698Gi	698Gi	0Bi	100% /Volumes/Google Chrome
/dev/disk6s2	107Mi	107Mi	0Bi	100%

```
May 19 08:58:23 bit fsevents[28]: log dir: /Volumes/Tim Machine Backups/.fsevents getting new uid: 5420A642-DE8C-4B90-B284-B948288F5E3F
May 19 16:52:30 bit fsevents[28]: log dir: /Volumes/NO NAME/.fsevents getting new uid: D064986D-F58C-407B-9010-5BD27104F062
May 23 20:10:35 bit fsevents[28]: log dir: /Volumes/NO NAME/.fsevents getting new uid: 0D8CB93B-0691-4381-ACEF-8F7F421D12DF
May 26 14:01:03 bit fsevents[28]: log dir: /Volumes/MoPassport/.fsevents getting new uid: CDCE4339-A254-4925-9099-97B4553BDAC1
May 26 15:40:38 bit fsevents[28]: log dir: /Volumes/MoPassport/.fsevents getting new uid: D4FFFB82-1648-4C83-88DE-327C0E1551EC
```

## Mounted Volumes (10.9+) system.log - Search "hfs:"

```
Aug 2 09:00:28 nibble kernel[0]: hfs: mounted Recovery HD on device disk0s3
Aug 2 09:00:29 nibble kernel[0]: hfs: unmount initiated on Recovery HD on device disk0s3
Aug 2 09:00:51 nibble kernel[0]: hfs: mounted Recovery HD on device disk0s3
Aug 2 09:00:52 nibble kernel[0]: hfs: unmount initiated on Recovery HD on device disk0s3
Aug 2 09:30:13 nibble kernel[0]: hfs: mounted ExifTool-9.69 on device disk2
Aug 2 13:10:11 nibble kernel[0]: hfs: mounted Thunderbolt on device disk3s3
Aug 2 13:11:09 nibble kernel[0]: hfs: unmount initiated on Thunderbolt on device disk3s3
Aug 2 14:55:30 nibble kernel[0]: hfs: mounted Recovery HD on device disk0s3
Aug 2 14:55:30 nibble kernel[0]: hfs: unmount initiated on Recovery HD on device disk0s3
Aug 2 15:29:53 nibble kernel[0]: hfs: mounted Thunderbolt on device disk3s3
Aug 2 15:31:29 nibble kernel[0]: hfs: unmount initiated on Thunderbolt on device disk3s3
Aug 2 15:33:09 nibble kernel[0]: hfs: mounted Thunderbolt on device disk3s3
Aug 2 15:33:13 nibble kernel[0]: hfs: unmount initiated on Thunderbolt on device disk3s3
Aug 2 15:35:21 nibble kernel[0]: hfs: mounted Thunderbolt on device disk3s3
Aug 2 15:35:49 nibble kernel[0]: hfs: unmount initiated on Thunderbolt on device disk3s3
Aug 2 15:35:59 nibble kernel[0]: hfs: mounted Thunderbolt on device disk3s3
Aug 2 15:36:18 nibble kernel[0]: hfs: unmount initiated on Thunderbolt_External_Drive on device disk3s3
Aug 2 15:36:31 nibble kernel[0]: hfs: mounted Thunderbolt_External_Drive on device disk3s3
Aug 2 15:37:43 nibble kernel[0]: hfs: unmount initiated on Thunderbolt_External_Drive on device disk3s3
Aug 2 15:37:58 nibble kernel[0]: hfs: mounted Thunderbolt_External_Drive on device disk3s3
Aug 2 15:38:31 nibble kernel[0]: hfs: unmount initiated on Thunderbolt_External_Drive on device disk3s3
```

Note: "Thunderbolt" and  
"Thunderbolt\_External\_Drive"  
are volume names.

© SANS.  
All Rights Reserved

Mac Forensic Analysis

10.9 systems show when volumes were mounted and unmounted.

These records can be found in the system.log by searching for "hfs:", "mounted", or "unmounted".

These records show the /dev mount point as well as the volume name (i.e., "ExifTool-9.96" or "Thunderbolt").

# USB Drives - kernel.log & system.log

## Search “USBMSC”

- Serial Number, Vendor ID, Product ID, Version
  - <=10.7 – kernel.log
  - 10.8+ – system.log

```
Apr 25 12:27:11 Pro kernel[0]: USBMSC Identifier (non-unique): 58A8120830AC8C5C 0x1e1d 0x1101 0x100
Apr 25 12:32:31 Pro kernel[0]: USBMSC Identifier (non-unique): 58A8120830AC8C5C 0x1e1d 0x1101 0x100
Apr 25 12:47:29 Pro kernel[0]: USBMSC Identifier (non-unique): 58A8120830AC8C5C 0x1e1d 0x1101 0x100
Apr 25 12:49:43 Pro kernel[0]: USBMSC Identifier (non-unique): 58A8120830AC8C5C 0x1e1d 0x1101 0x100
Apr 25 12:52:46 Pro kernel[0]: USBMSC Identifier (non-unique): FBF1011220504638 0x90c 0x1000 0x1100
Apr 25 12:53:37 Pro kernel[0]: USBMSC Identifier (non-unique): ABCDEF0123456789 0xe90 0x5 0x0
Apr 25 13:04:21 Pro kernel[0]: USBMSC Identifier (non-unique): 58A8120830AC8C5C 0x1e1d 0x1101 0x100
Apr 25 13:04:29 Pro kernel[0]: USBMSC Identifier (non-unique): FBF1011220504638 0x90c 0x1000 0x1100
Apr 26 12:36:05 Pro kernel[0]: USBMSC Identifier (non-unique): 58A8120830AC8C5C 0x1e1d 0x1101 0x100
Apr 27 09:02:59 Pro kernel[0]: USBMSC Identifier (non-unique): FBF1011220504638 0x90c 0x1000 0x1100
Apr 30 09:07:14 Pro kernel[0]: USBMSC Identifier (non-unique): FBF1011220504638 0x90c 0x1000 0x1100
May 3 05:43:05 Pro kernel[0]: UDMDG Identifier (non-unique): 5DA0120830AC8C5C 0x1e1d 0x1101 0x100
May 3 06:24:05 Pro kernel[0]: USBMSC Identifier (non-unique): SWOC22905731 0x1199 0x1ff 0x323
May 24 11:22:43 Pro kernel[0]: USBMSC Identifier (non-unique): 000000009833 0x5ac 0x5403 0x9833
May 24 11:53:25 Pro kernel[0]: USBMSC Identifier (non-unique): 0911201415f7f3 0x1e1d 0x165 0x100
May 25 12:48:38 Pro kernel[0]: USBMSC Identifier (non-unique): 0911201415f7f3 0x1e1d 0x165 0x100
May 30 06:50:01 Pro kernel[0]: USBMSC Identifier (non-unique): 0911201415f7f3 0x1e1d 0x165 0x100
May 31 13:10:09 Pro kernel[0]: USBMSC Identifier (non-unique): 0911201415f7f3 0x1e1d 0x165 0x100
Jun 1 07:16:03 Pro kernel[0]: USBMSC Identifier (non-unique): 0911201415f7f3 0x1e1d 0x165 0x100
```

© SANS.  
All Rights Reserved.

## Mac Forensic Analysis

The USB Mass Storage Device Class (USBMSC) Identifier, usually the serial number of the device, can be found by doing a search for “USBMSC” in the kernel.log on systems using 10.7 and earlier. With 10.8+, the kernel log data has been incorporated into the system.log.

While the serial number may in fact be the one found on the suspect USB drive, it may not necessarily be unique as the USBMSC message reminded us. An example of one such device is one that uses the letters A-F and 0-9 as its identifier. The additional numbers are the vendor ID, Product ID, and version.

```
Jun 3 11:11:53 bit kernel[0]: USBMSC Identifier (non-unique):
FBF1011220504638 0x90c 0x1000 0x1100
```

The example USBMSC record shown above contains a USB device that is identified by the serial number: FBF1011220504638. The additional data shown in the screenshot using the System Information application shows vendor/product data.

- Vendor ID – 0x090c (Silicon Motion, Inc)
- Product ID - 0x1000
- Version 11.00

The website: <http://usb-ids.gowdy.us/read/UD/> lists vendors by ID and products by ID. It may also link to a forum discussing the physical properties of the device.

```
Apr 25 12:27:11 Pro kernel[0]: USBMSC Identifier (non-unique):  
58A8120830AC8C5C 0x1e1d 0x1101 0x100  
Apr 25 12:32:31 Pro kernel[0]: USBMSC Identifier (non-unique):  
58A8120830AC8C5C 0x1e1d 0x1101 0x100  
Apr 25 12:47:29 Pro kernel[0]: USBMSC Identifier (non-unique):  
58A8120830AC8C5C 0x1e1d 0x1101 0x100  
Apr 25 12:49:43 Pro kernel[0]: USBMSC Identifier (non-unique):  
58A8120830AC8C5C 0x1e1d 0x1101 0x100  
Apr 25 12:52:46 Pro kernel[0]: USBMSC Identifier (non-unique):  
FBF1011220504638 0x90c 0x1000 0x1100  
Apr 25 12:53:37 Pro kernel[0]: USBMSC Identifier (non-unique):  
ABCDEF0123456789 0xe90 0x5 0x0  
Apr 25 13:04:21 Pro kernel[0]: USBMSC Identifier (non-unique):  
58A8120830AC8C5C 0x1e1d 0x1101 0x100  
Apr 25 13:04:29 Pro kernel[0]: USBMSC Identifier (non-unique):  
FBF1011220504638 0x90c 0x1000 0x1100  
Apr 26 12:36:05 Pro kernel[0]: USBMSC Identifier (non-unique):  
58A8120830AC8C5C 0x1e1d 0x1101 0x100  
Apr 27 09:02:59 Pro kernel[0]: USBMSC Identifier (non-unique):  
FBF1011220504638 0x90c 0x1000 0x1100  
Apr 30 09:07:14 Pro kernel[0]: USBMSC Identifier (non-unique):  
FBF1011220504638 0x90c 0x1000 0x1100  
May 3 05:43:05 Pro kernel[0]: USBMSC Identifier (non-unique):  
58A8120830AC8C5C 0x1e1d 0x1101 0x100  
May 3 06:24:05 Pro kernel[0]: USBMSC Identifier (non-unique): SWOC22905731  
0x1199 0xffff 0x323  
May 24 11:22:43 Pro kernel[0]: USBMSC Identifier (non-unique): 000000009833  
0x5ac 0x8403 0x9833  
May 24 11:53:25 Pro kernel[0]: USBMSC Identifier (non-unique):  
0911201415f7f3 0x1e1d 0x165 0x100  
May 25 12:48:38 Pro kernel[0]: USBMSC Identifier (non-unique):  
0911201415f7f3 0x1e1d 0x165 0x100  
May 30 06:50:01 Pro kernel[0]: USBMSC Identifier (non-unique):  
0911201415f7f3 0x1e1d 0x165 0x100  
May 31 13:10:09 Pro kernel[0]: USBMSC Identifier (non-unique):  
0911201415f7f3 0x1e1d 0x165 0x100  
Jun 1 07:16:03 Pro kernel[0]: USBMSC Identifier (non-unique):  
0911201415f7f3 0x1e1d 0x165 0x100
```

# Thunderbolt Drives - system.log

Search “IOThunderboltSwitch” and “hfs:” in Context

```
Aug  2 15:37:54 nibble kernel[0]:  
IOThunderboltSwitch<0xffffffff803c894000>(0x0)::listenerCallback - Thunderbolt  
HPD packet for route = 0x0 port = 1 unplug = 0  
Aug  2 15:37:55 nibble kernel[0]: The USB device Apple Internal Keyboard /  
Trackpad (Port 5 of Hub at 0x14000000) may have caused a wake by issuing a  
remote wakeup (2)  
Aug  2 15:37:56 nibble kernel[0]: [ PCI configuration begin ]  
Aug  2 15:37:56 nibble kernel[0]: [ PCI configuration end, bridges 14, devices  
13 ]  
Aug  2 15:37:58 nibble kernel[0]: hfs: mounted Thunderbolt_External_Drive on  
device disk3s3  
Aug  2 15:38:31 nibble kernel[0]: hfs: unmount initiated on  
Thunderbolt_External_Drive on device disk3s3  
Aug  2 15:38:51 nibble kernel[0]:  
IOThunderboltSwitch<0xffffffff803c894000>(0x0)::listenerCallback - Thunderbolt  
HPD packet for route = 0x0 port = 1 unplug = 1  
Aug  2 15:38:51 nibble kernel[0]: [ PCI configuration begin ]  
Aug  2 15:38:51 nibble kernel[0]: [ PCI configuration end, bridges 12, devices  
13 ]
```

© SANS.  
All Rights Reserved.

Mac Forensic Analysis

The log records show a Thunderbolt external drive plugged into a system, mounted, unmounted, and removed from the system.

The logs with the “hfs:” messages show the name of the mounted volume on the Thunderbolt drive – “Thunderbolt\_External\_Drive”.

## Apple Filing Protocol (AFP) Network Shares – Search “AFP\_VFS”

```
Jun 15 21:00:01 nibble kernel[0]: AFP_VFS afpfs_mount:  
/Volumes/Macintosh HD-1, pid 860  
Jun 15 21:00:01 nibble kernel[0]: AFP_VFS afpfs_mount : succeeded on  
volume 0xffffffff80d5a33008 /Volumes/Macintosh HD-1 (error = 0, retval  
= 0)  
Jun 15 21:00:59 nibble kernel[0]: AFP_VFS afpfs_unmount:  
/Volumes/Macintosh HD-1, flags 0, pid 879  
Jun 15 21:00:59 nibble kernel[0]: AFP_VFS afpfs_unmount : We are the  
last mnt/shmnt using volume /Volumes/Macintosh HD-1  
0xffffffff80d5a33008  
Jun 15 21:00:59 nibble kernel[0]: AFP_VFS afpfs_unmount : We are the  
last volume using socket /Volumes/Macintosh HD-1 0xffffffff80d5a33008  
Jun 15 21:00:59 nibble kernel[0]: AFP_VFS afpfs_unmount :  
afpfs_DoReconnect sent signal for unmount to proceed
```

© SANS.  
All Rights Reserved

Mac Forensic Analysis

When an Apple Filing Protocol (AFP) share is accessed, such as those created via the Sharing preferences, you'll see these shares accessed in the `kernel.log` (<=10.7) or `system.log` (10.8+) files.

Each entry related to these shares can be found by searching for “AFP\_VFS”. The entry that shows the volume has been mounted contains an “`afpfs_mount`” keyword and the name and location of the mount. The entry that shows the share has been un-mounted contains the keyword “`afpfs_umount`” along with the name and location.

## Finder Volumes

~/Library/Preferences/com.apple.finder.plist

- FXDesktopVolumePositions
- FXRecentFolders (10 most recent)

The screenshot shows two panels from the Mac Forensic Analysis tool. The left panel displays the 'FXRecentFolders' key as an array containing 10 items. Each item is a dictionary with 'file-bookmark' and 'name' keys. The 'name' key for the first item is highlighted with a red oval and labeled 'STUFF'. The right panel shows the 'FXDesktopVolumePositions' key as a list of volume names.

**Key**

**FXRecentFolders**

Item	File-Bookmark	Name
Item 0	Data <626f6f6b ac030000	STUFF
Item 1	Data <626f6f6b 3c030000	TechnoSecurity2012
Item 2	Data <626f6f6b 8c020000	oompa
Item 3	Data <626f6f6b c0020000	Dropbox

**Key**

**FXDesktopVolumePositions**

- ▶ STUFF\_-0x1.d27e44p+29
- ▶ VMware Fusion\_0x1.3f5f0e2p+28
- ▶ WDPassport\_-0x1.d27e44p+29
- ▶ DATA\_0x1.3db4fc2p+28
- ▶ OmniOutliner\_0x1.25dc04p+27
- ▶ Sample Docs\_0x1.eefdap+26
- ▶ NO NAME\_-0x1.3c0752p+29
- ▶ OmniOutliner Pro\_0x1.25dcad2p+27
- ▶ Time Machine Backups\_0x1.438f33dp

C SANS.  
All Rights Reserved

Mac Forensic Analysis

Each user will have their own plist files containing their volume history.

The FXRecentFolders key contains the volumes mounted on the system for the user along with a bookmark data blob. This blob data can be extracted and viewed in a hex editor or property list editor to gain more information about the volume.

The FXDesktopVolumePositions contains a list of volumes. Neither of these keys contain timestamps, therefore correlation with other data points will have to be done to determine when these volumes were available on the system.

# Finder Volumes

[~/Library/Preferences/com.apple.sidebarlists.plist](#)

Key	Type	Value
favorites	Dictionary	(7 items)
CustomListProperties	Dictionary	(2 items)
ShowRemovable	Boolean	YES
ShowHardDisks	Boolean	YES
ShowEjectables	Boolean	YES
VolumesList	Array	(57 items)
ShowServers	Boolean	YES
Controller	String	VolumesList
savedsearches	Dictionary	(2 items)
systemitems	Dictionary	(7 items)
CustomListProperties	Dictionary	(1 item)
ShowRemovable	Boolean	YES
ShowHardDisks	Boolean	YES
ShowEjectables	Boolean	YES
VolumesList	Array	(42 items)
ShowServers	Boolean	YES
Controller	String	VolumesList
VolumesList	Array	(42 items)
Item 0	Dictionary	(4 items)
Item 1	Dictionary	(5 items)
Item 2	Dictionary	(3 items)
Item 3	Dictionary	(4 items)
Item 4	Dictionary	(5 items)
Item 5	Dictionary	(4 items)
Item 6	Dictionary	(3 items)
Alias	Data	<00000000 00b40003 00010000 cbc9d31f 0000482b
Name	String	Dropbox Installer
EntryType	Number	1027
Item 7	Dictionary	(3 items)
Alias	Data	<00000000 00a00003 00010000 cbc0e521 0000482b
Name	String	Google Chrome
EntryType	Number	1027
Item 8	Dictionary	(3 items)
Alias	Data	<00000000 00740003 00010000 ca50c8c2 0000482b
Name	String	DATA
EntryType	Number	517

© SANS.  
All Rights Reserved

Mac Forensic Analysis

The general difference between the two VolumeList Keys is the items on the sidebar such as Applications, Documents, and Downloads.

Key	Type	Value
▼ favorites	Diction...	(7 items)
► CustomListProperties	Diction...	(2 items)
ShowRemovable	Boolean	YES
ShowHardDisks	Boolean	YES
ShowEjectables	Boolean	YES
► VolumesList	Array	(57 items)
ShowServers	Boolean	YES
Controller	String	VolumesList
► savedsearches	Diction...	(2 items)
▼ systemitems	Diction...	(7 items)
► CustomListProperties	Diction...	(1 item)
ShowRemovable	Boolean	YES
ShowHardDisks	Boolean	YES
ShowEjectables	Boolean	YES
► VolumesList	Array	(42 items)
ShowServers	Boolean	YES
Controller	String	Volume
► Item 0	Boolean	YES
► Item 1	Boolean	YES
► Item 2	Boolean	YES
► Item 3	Boolean	YES
► Item 4	Boolean	YES
► Item 5	Boolean	YES
► Item 6	Boolean	YES
► Item 7	Boolean	YES
► Item 8	Boolean	YES
Alias	Data	<00000000000000000000000000000000
Name	String	Dropbox Installer
EntryType	Number	1027
► Item 9	Boolean	YES
Alias	Data	<00000000000000000000000000000000
Name	String	Google Chrome
EntryType	Number	1027
► Item 10	Boolean	YES
Alias	Data	<00000000000000000000000000000000
Name	String	DATA
EntryType	Number	517

## Finder Volume Alias Data Volume Format

BDxF - ExFAT	
BDIS – FAT32	
BDcu – UDF (DVD)	
NTcu – Unknown	
H+ - HFS+	
KG - FTP	

Item 38      Diction...      (3 items)

Alias	Data
Name	String
EntryType	Number
	<00000000 008c0003 00010000 cbe15267 0000482b
	Wireshark
	1027

```

000 00 00 00 00 00 8C 00 03 00 01 00 00 CB E1 57 67 00 00 48 2B 00 00 \\\ Wireshark\\
022 00 05 00 00 00 01 00 00 00 02 00 00 CB E1 57 67 00 00 00 00 00 02 \\\ Wireshark\\
044 FF FE 00 00 00 00 00 00 00 00 FF FF FF FF 00 01 00 00 00 0E 00 14 \\\ Wireshark\\
066 00 09 00 57 00 69 00 72 00 65 00 73 00 68 00 61 00 72 00 68 00 0F \\\ Wireshark\\
088 00 14 00 09 00 57 00 69 00 72 00 65 00 73 00 68 00 61 00 72 00 6B \\\ Wireshark\\
110 00 12 00 00 00 13 00 12 2F 56 6F 6C 75 6D 65 73 2F 57 69 72 65 73 \\\ Wireshark\\
132 68 61 72 68 FF FF 00 00 \\\ Wireshark\\

```

© SANS,  
All Rights Reserved

Mac Forensic Analysis

The alias blob data can be extracted and read in a hex editor. Each alias data blob contains a file system format, many of which are listed in the slide itself.

Some of these volume formats are currently unknown. Further testing will be required.

000	00	00	00	00	8C	00	03	00	01	00	00	CB	E1	57	67	00	48	2B	00	00	Wg H+ Wg S+
022	00	05	00	00	00	01	00	00	02	00	00	CB	E1	57	67	00	00	00	00	02	Wg S+
044	FF	FE	00	00	00	00	00	00	00	FF	00	01	00	00	14						
066	00	09	00	57	00	69	00	72	00	65	00	73	00	68	00	61	00	72	00	68	0F
088	00	14	00	09	00	57	00	69	00	72	00	65	00	73	00	68	00	61	00	72	00
110	00	12	00	00	00	13	00	12	2F	56	6F	6C	75	6D	65	73	2F	57	69	72	65
132	68	61	72	68	FF	FF	00	00													

## Finder Volume Alias Data Dates & Mount Point

- Volume Name & Mount Point
- File Creation Time (HFS Date)
- May or may not be present
  - Only seen on H+ formatted disks
  - DMG

000	00	00	00	00	00	8C	00	03	00	01	00	00	CB	E1	57	67	00	00	48	2B	00	00	~~~~~	~~~~~	Wg	~~~~~	H+~~~
022	00	05	00	00	00	01	00	00	00	02	00	00	CB	E1	57	67	00	00	00	00	0D	02	~~~~~	~~~~~	~~~~~	~~~~~	Mg~~~~~
044	FF	FE	00	00	00	00	00	00	00	00	FF	FF	FF	FF	88	81	00	00	00	0E	00	14	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~
066	00	09	00	57	00	69	00	72	00	65	00	73	00	68	00	61	00	72	00	68	00	0F	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~
088	00	14	00	09	00	57	00	69	00	72	00	65	00	73	00	68	00	61	00	72	00	68	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~
110	00	12	00	00	00	13	00	12	2F	56	6F	6C	75	6D	65	73	2F	57	69	72	65	73	~~~~~	~~~~~	~~~~~	~~~~~	Volumes/Wires
132	68	61	72	68	FF	FF	00	00	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~	hork

© SANS  
All Rights Reserved

Mac Forensic Analysis

The alias blob data will also contain the volume name and mount point on the system. Creation dates may also be present on HFS+ volumes.

# Temporal Changes

Intentional or unintentional changes by a user

Time Zone

Time Modifications

Date Modifications

A user may make intentional changes to the system time or date. They might, for example, be traveling and need to change time zones. A user may also make changes to the system time or date to throw a temporal investigation off.

# Time Changes: Going Back in Time

## /var/log/system.log

```
Jun 16 14:50:56 bit System Preferences[1828]: -[GEOWorldTimeZoneView selectedCityLayer] Invalidating _selectedCityLayer
Jun 16 14:50:56 bit System Preferences[1828]: -[GEOWorldTimeZoneView selectedCityLayer] all good cachedValue:1.000000
Jun 16 14:50:56: --- last message repeated 4 times ---
Jun 16 14:50:56 bit System Preferences[1828]: -[GEOWorldTimeZoneView selectedCityLayer] Invalidating _selectedCityLayer
Jun 16 14:50:56 bit System Preferences[1828]: -[GEOWorldTimeZoneView selectedCityLayer] all good cachedValue:1.000000
Jun 16 14:50:56: --- last message repeated 1 time ---
Jun 16 14:50:56 bit System Preferences[1828]: **** ERROR: -[GEOCityPickerView bindPublicToPrivateProperties] UI is already bounded
Jun 16 14:50:59 bit System Preferences[1828]: -[GEOWorldTimeZoneView selectedCityLayer] all good cachedValue:1.000000
Jun 16 11:51:05: --- last message repeated 4 times ---
Jun 16 11:51:05 bit System Preferences[1828]: -[GEOWorldTimeZoneView selectedCityLayer] Invalidating _selectedCityLayer
Jun 16 11:51:05 bit System Preferences[1828]: -[GEOWorldTimeZoneView selectedCityLayer] all good cachedValue:1.000000
Jun 16 11:51:06 bit System Preferences[1828]: -[GEOWorldTimeZoneView selectedCityLayer] Invalidating _selectedCityLayer
Jun 16 11:51:06 bit System Preferences[1828]: -[GEOWorldTimeZoneView selectedCityLayer] all good cachedValue:1.000000
Jun 16 11:51:06 bit ntpd[1848]: proto: precision = 1.000 usec
```

© SANS,  
All Rights Reserved

Mac Forensic Analysis

A user might attempt to hide or obfuscate activity by changing the system time.

In the screenshot above the system time was changed to go back in time three hours. The highlighted timestamps in the system.log file show that the time was changed using the System Preferences (Date & Time) preference panel.

```
Jun 16 14:50:56 bit System Preferences[1828]: -[GEOWorldTimeZoneView selectedCityLayer] Invalidating _selectedCityLayer
Jun 16 14:50:56 bit System Preferences[1828]: -[GEOWorldTimeZoneView selectedCityLayer] all good cachedValue:1.000000
Jun 16 14:50:56: --- last message repeated 4 times ---
Jun 16 14:50:56 bit System Preferences[1828]: -[GEOWorldTimeZoneView selectedCityLayer] Invalidating _selectedCityLayer
Jun 16 14:50:56 bit System Preferences[1828]: -[GEOWorldTimeZoneView selectedCityLayer] all good cachedValue:1.000000
Jun 16 14:50:56: --- last message repeated 1 time ---
Jun 16 14:50:56 bit System Preferences[1828]: **** ERROR: -[GEOCityPickerView bindPublicToPrivateProperties] UI is already bounded
Jun 16 14:50:59 bit System Preferences[1828]: -[GEOWorldTimeZoneView selectedCityLayer] all good cachedValue:1.000000
Jun 16 11:51:05: --- last message repeated 4 times ---
Jun 16 11:51:05 bit System Preferences[1828]: -[GEOWorldTimeZoneView selectedCityLayer] Invalidating _selectedCityLayer
Jun 16 11:51:05 bit System Preferences[1828]: -[GEOWorldTimeZoneView selectedCityLayer] all good cachedValue:1.000000
Jun 16 11:51:06 bit System Preferences[1828]: -[GEOWorldTimeZoneView selectedCityLayer] Invalidating _selectedCityLayer
Jun 16 11:51:06 bit System Preferences[1828]: -[GEOWorldTimeZoneView selectedCityLayer] all good cachedValue:1.000000
Jun 16 11:51:06 bit ntpd[1848]: proto: precision = 1.000 usec
```

## Time Changes

### Time Zone - /etc/localtime

```
bit:etc oompa$ pwd  
/etc  
bit:etc oompa$ ls -l localtime  
lrwxr-xr-x 1 root wheel 39 Jun 16 11:51 localtime ->  
/usr/share/zoneinfo/America/Los_Angeles
```

© SANS,  
All Rights Reserved.

Mac Forensic Analysis

The localtime symlink is updated when the time zone is changed to reflect the new time zone. In the example shown, the time zone was last changed to Los Angeles time on June 16<sup>th</sup>.

## Time Changes: Back to the Future /var/log/system.log

```
_selectedCityLayer
Jun 16 12:08:04 bit System Preferences[1914]: -[GEOWorldTimeZoneView
selectedCityLayer] all good cachedValue:1.000000
Jun 16 12:08:04: --- last message repeated 1 time ---
Jun 16 12:08:04 bit System Preferences[1914]: **** ERROR: -
[GEOCityPickerView _bindPublicToPrivateProperties] UI is already
bounded
Jun 16 12:08:06 bit System Preferences[1914]: -[GEOWorldTimeZoneView
selectedCityLayer] all good cachedValue:1.000000
Jun 16 15:08:09: --- last message repeated 9 times ---
Jun 16 15:08:09 bit System Preferences[1914]: WARNING: -
[GEOTimezoneHitMap fileNameAtLongitude:latitude:] no time zone area
found
Jun 16 15:08:13 bit System Preferences[1914]: -[GEOWorldTimeZoneView
selectedCityLayer] all good cachedValue:1.000000
Jun 16 15:08:15: --- last message repeated 5 times ---
```

© SANS.  
All Rights Reserved

Mac Forensic Analysis

The time can also be changed to the “future” as shown above. The time zone was changed to reflect three hours in advance. The system.log will show entries one right after another with the difference in time., note the same relative minute and second values.

Time Zone Changes daily.out – Search “2012”					
<b>Tue Jun 19 07:12:16 EDT 2012</b>					
Removing old temporary files:					
Cleaning out old system announcements:					
Removing stale files from /var/rwtmp:					
Removing scratch fax files					
Disk status:					
Filesystem					Size
/dev/disk1					698G
localhost:/7YF29FtTiww-stDNE80g6T					698G
/dev/disk5s2					1.8G
Network interface status:					
Name Mtu Network Address					
lo0 16384 <Link#1>					
lo0 16384 localhost fe80:1::1					
lo0 16384 127 localhost					
lo0 16384 localhost ::1					
gif0* 1280 <Link#2>					
stf0* 1280 <Link#3>					
en0 1500 <Link#4>					c4:2c:03:09:c8:
en1 1500 <Link#5>					98:27:e4:f8:e6:
en1 1500 bit.local fe80:5:9227::4					
fw0 4078 <Link#6>					e8:05:88:ff:fe:
p2pb 2304 <Link#7>					02:27:e4:f8:e6:
Local system status:					
7:12 up 4 days, 10:22, 5 users, load					
-- End of daily output --					
© SANS. All Rights Reserved.					
Mac Forensic Analysis					

Time zone changes can also be shown in the daily.out file.

In the example above a search, was completed for the year “2012”. The highlighted entries show a change from Eastern Daylight Time (EDT) to Pacific Daylight Time (PDT) for the days of June 17<sup>th</sup> and June 18<sup>th</sup>.

## System Information & State

System Boot

System Reboot

System Shutdown

System Boot Process

System Boot Device

System Hardware Information

The logs can tell us a lot about how and when a system was used. In a temporal context, we can see when a system was powered on, when it was in a sleep state, and when it was shut down.

The hardware configuration of the system can be determined to see if the system was upgraded or if the boot drive changed systems.

# Boot, Reboot & Shutdown

## /var/log/system.log

```
May  9 16:28:48 localhost bootlog[0]: BOOT_TIME 1336606128 0
May 10 16:40:27 localhost bootlog[0]: BOOT_TIME 1336682427 0
May 12 11:32:16 localhost bootlog[0]: BOOT_TIME 1336836736 0
May 27 20:02:41 localhost bootlog[0]: BOOT_TIME 1338163361 0
May 28 15:22:30 localhost bootlog[0]: BOOT_TIME 1338232050 0
Jun  9 09:27:05 localhost bootlog[0]: BOOT_TIME 1339248425 0
Jun  9 10:15:56 localhost bootlog[0]: BOOT_TIME 1339251356 0
Jun  9 10:33:39 localhost bootlog[0]: BOOT_TIME 1339252419 0
Jun  9 09:27:05 localhost bootlog[0]: BOOT_TIME 1339248425 0
Jun  9 10:15:56 localhost bootlog[0]: BOOT_TIME 1339251356 0
Jun  9 10:33:39 localhost bootlog[0]: BOOT_TIME 1339252419 0
Jun 10 13:33:56 localhost bootlog[0]: BOOT_TIME 1339349636 0
Jun 12 10:16:35 localhost bootlog[0]: BOOT_TIME 1339510595 0

May 27 20:02:14 bit shutdown[42801]: halt by oompa:
May 27 20:02:14 bit shutdown[42801]: SHUTDOWN_TIME: 1338163334 903600
May 28 15:20:06 bit shutdown[2421]: halt by oompa:
May 28 15:20:06 bit shutdown[2421]: SHUTDOWN_TIME: 1338232806 702175
Jun  9 09:25:33 bit shutdown[25868]: halt by oompa:
Jun  9 09:25:33 bit shutdown[25868]: SHUTDOWN_TIME: 1339248333 887656
Jun  9 10:15:24 bit shutdown[546]: reboot by oompa:
Jun  9 10:15:24 bit shutdown[546]: SHUTDOWN_TIME: 1339251324 30856
Jun  9 10:21:53 bit shutdown[309]: halt by oompa:
Jun  9 10:21:53 bit shutdown[309]: SHUTDOWN_TIME: 1339251713 535787
Jun  9 09:25:33 bit shutdown[25868]: halt by oompa:
Jun  9 09:25:33 bit shutdown[25868]: SHUTDOWN_TIME: 1339248333 887656
Jun  9 10:15:24 bit shutdown[546]: reboot by oompa:
Jun  9 10:15:24 bit shutdown[546]: SHUTDOWN_TIME: 1339251324 30856
Jun  9 10:21:53 bit shutdown[309]: halt by oompa:
Jun  9 10:21:53 bit shutdown[309]: SHUTDOWN_TIME: 1339251713 535787
```

© SANS.  
All Rights Reserved

Mac Forensic Analysis

Knowing when the system boots or when it is shutdown can help with the timeline aspect of some investigations. The `system.log` contains the terms “`BOOT_TIME`” and “`SHUTDOWN_TIME`”. The log also records what user account shutdown (“`halt`”) or restarted (“`reboot`”) the system.

```
May  9 16:28:48 localhost bootlog[0]: BOOT_TIME 1336606128 0
May 10 16:40:27 localhost bootlog[0]: BOOT_TIME 1336682427 0
May 12 11:32:16 localhost bootlog[0]: BOOT_TIME 1336836736 0
May 27 20:02:41 localhost bootlog[0]: BOOT_TIME 1338163361 0
May 28 15:22:30 localhost bootlog[0]: BOOT_TIME 1338232950 0
Jun  9 09:27:05 localhost bootlog[0]: BOOT_TIME 1339248425 0
Jun  9 10:15:56 localhost bootlog[0]: BOOT_TIME 1339251356 0
Jun  9 10:33:39 localhost bootlog[0]: BOOT_TIME 1339252419 0
Jun  9 09:27:05 localhost bootlog[0]: BOOT_TIME 1339248425 0
Jun  9 10:15:56 localhost bootlog[0]: BOOT_TIME 1339251356 0
Jun  9 10:33:39 localhost bootlog[0]: BOOT_TIME 1339252419 0
Jun 10 13:33:56 localhost bootlog[0]: BOOT_TIME 1339349636 0
Jun 12 10:16:35 localhost bootlog[0]: BOOT_TIME 1339510595 0
```

```
May 27 20:02:14 bit shutdown[42801]: halt by oompa:
May 27 20:02:14 bit shutdown[42801]: SHUTDOWN_TIME: 1338163334 903688
May 28 15:20:06 bit shutdown[2421]: halt by oompa:
May 28 15:20:06 bit shutdown[2421]: SHUTDOWN_TIME: 1338232806 702175
Jun  9 09:25:33 bit shutdown[25868]: halt by oompa:
Jun  9 09:25:33 bit shutdown[25868]: SHUTDOWN_TIME: 1339248333 887656
Jun  9 10:15:24 bit shutdown[546]: reboot by oompa:
Jun  9 10:15:24 bit shutdown[546]: SHUTDOWN_TIME: 1339251324 30856
Jun  9 10:21:53 bit shutdown[309]: halt by oompa:
Jun  9 10:21:53 bit shutdown[309]: SHUTDOWN_TIME: 1339251713 535787
Jun  9 09:25:33 bit shutdown[25868]: halt by oompa:
Jun  9 09:25:33 bit shutdown[25868]: SHUTDOWN_TIME: 1339248333 887656
Jun  9 10:15:24 bit shutdown[546]: reboot by oompa:
Jun  9 10:15:24 bit shutdown[546]: SHUTDOWN_TIME: 1339251324 30856
Jun  9 10:21:53 bit shutdown[309]: halt by oompa:
Jun  9 10:21:53 bit shutdown[309]: SHUTDOWN_TIME: 1339251713 535787
```

## System Boot kernel.log & system.log

10.8+

- Boot logging starts with "BOOT\_TIME"

10.7

- Boot logging starts with "PMAP: PCID enabled"

10.6

- Boot logging starts with "npvhash=4095"

© SANS.  
All Rights Reserved

Mac Forensic Analysis

In the kernel.log or system.log (depending on the system), you will find the example text shown above when boot logging starts.

Due to the log compilation of all the system.log in 10.8, the boot logging starts with BOOT\_TIME, so it is relatively easy to recognize. In 10.7 the boot logging starts with the line "PMAP: PCID enabled" in the kernel.log. 10.6 starts with the entry "npvhash=4095"

```

Jun 12 10:16:53 localhost kernel[0]: PMAP: PCID enabled
Jun 12 10:16:53 localhost kernel[0]: Darwin Kernel Version 11.4.0: Mon Apr  9 19:32:15 PDT 2012
Jun 12 10:16:53 localhost kernel[0]: vm_page_bootstrap: 2011634 free pages and 69134 wired
Jun 12 10:16:53 localhost kernel[0]: kext submap [0x1fffff7f80732000 - 0xfffffff800000000]
Jun 12 10:16:53 localhost kernel[0]: zone leak detection enabled
Jun 12 10:16:53 localhost kernel[0]: standard timeslicing quantum is 10000 us
Jun 12 10:16:53 localhost kernel[0]: mig_table_max_displ = 73
Jun 12 10:16:53 localhost kernel[0]: AppleACPICPU: ProcessorId=1 LocalApicId=0 Enabled
Jun 12 10:16:53 localhost kernel[0]: AppleACPICPU: ProcessorId=2 LocalApicId=1 Enabled
Jun 12 10:16:53 localhost kernel[0]: AppleACPICPU: ProcessorId=3 LocalApicId=4 Enabled
Jun 12 10:16:53 localhost kernel[0]: AppleACPICPU: ProcessorId=4 LocalApicId=5 Enabled
Jun 12 10:16:53 localhost kernel[0]: AppleACPICPU: ProcessorId=5 LocalApicId=0 Disabled
Jun 12 10:16:53 localhost kernel[0]: AppleACPICPU: ProcessorId=6 LocalApicId=0 Disabled
Jun 12 10:16:53 localhost kernel[0]: AppleACPICPU: ProcessorId=7 LocalApicId=0 Disabled
Jun 12 10:16:53 localhost kernel[0]: AppleACPICPU: ProcessorId=8 LocalApicId=0 Disabled
Jun 12 10:16:53 localhost kernel[0]: calling mpo_policy_init for TMSafetyNet
Jun 12 10:16:53 localhost kernel[0]: Security policy loaded: Safety net for Time Machine
Jun 12 10:16:53 localhost kernel[0]: Jun 5 12:07:52 localhost kernel[0]: npvhash=4095
Jun 12 10:16:53 localhost kernel[0]: PAE enabled
Jun 12 10:16:53 localhost kernel[0]: Jun 5 12:07:52 localhost kernel[0]: 64 bit mode enabled
Jun 12 10:16:53 localhost kernel[0]: Darwin Kernel Version 10.0.0: Fri Jul 31 22:47:34 PDT 2009
Jun 12 10:16:53 localhost kernel[0]: vm_page_bootstrap: 1926010 free pages and 95606 wired
Jun 12 10:16:53 localhost kernel[0]: standard timeslicing quantum is 10000 us
Jun 12 10:16:53 localhost kernel[0]: mig_table_max_displ = 73
Jun 12 10:16:53 localhost kernel[0]: AppleACPICPU: ProcessorId=0 LocalApicId=0 Enabled
Jun 12 10:16:53 localhost kernel[0]: Jun 5 12:07:52 localhost kernel[0]: AppleACPICPU: ProcessorId=1 LocalApicId=1 Enabled
Jun 5 12:07:52 localhost kernel[0]: calling mpo_policy_init for TMSafetyNet
Jun 5 12:07:52 localhost kernel[0]: Security policy loaded: Safety net for Time Machine
Jun 5 12:07:52 localhost kernel[0]: calling mpo_policy_init for Quarantine
Jun 5 12:07:52 localhost kernel[0]: Security policy loaded: Quarantine policy (Quarantine)
Jun 5 12:07:52 localhost kernel[0]: calling mpo_policy_init for Sandbox
Jun 5 12:07:52 localhost kernel[0]: Security policy loaded: Seatbelt sandbox policy (Seatbelt)
Jun 5 12:07:52 localhost kernel[0]: Copyright (c) 1982, 1986, 1989, 1991, 1993
Jun 5 12:07:52 localhost kernel[0]: The Regents of the University of California. All rights reserved.
Jun 5 12:07:52 localhost kernel[0]: MAC Framework successfully initialized
Jun 5 12:07:52 localhost kernel[0]: using 16384 buffer headers and 4096 cluster I/O buffers
Jun 5 12:07:52 localhost kernel[0]: IOAPIC: Version 0x11 Vectors 64:87
Jun 5 12:07:52 localhost kernel[0]: ACPI: System State [S0 S3 S4 S5] (S3)
Jun 5 12:07:52 localhost kernel[0]: mbinit: done (64 MB memory set for mbuf pool)
Jun 5 12:07:52 localhost kernel[0]: rooting via boot-uuid from /chosen: 5D895F73-8D80

```

The screenshot examples shown above are the 10.7 (top) and 10.6 (bottom) versions of the `kernel.log`. 10.8+ systems will show the same as the top screenshot, however the boot logging is clearly shown with the “`BOOT_TIME`” entry.

```
Jun 12 10:16:53 localhost kernel[0]: PMAP: PCID enabled
Jun 12 10:16:53 localhost kernel[0]: Darwin Kernel Version 11.4.0: Mon Apr  9 19:32:15 P
Jun 12 10:16:53 localhost kernel[0]: vm_page_bootstrap: 2011634 free pages and 69134 wi
Jun 12 10:16:53 localhost kernel[0]: kext submap [0xffffffff800732000 - 0xffffffff8000000000
Jun 12 10:16:53 localhost kernel[0]: zone leak detection enabled
Jun 12 10:16:53 localhost kernel[0]: standard timeslicing quantum is 10000 us
Jun 12 10:16:53 localhost kernel[0]: mig_table_max_displ = 73
Jun 12 10:16:53 localhost kernel[0]: AppleACPICPU: ProcessorId=1 LocalApicId=0 Enabled
Jun 12 10:16:53 localhost kernel[0]: AppleACPICPU: ProcessorId=2 LocalApicId=1 Enabled
Jun 12 10:16:53 localhost kernel[0]: AppleACPICPU: ProcessorId=3 LocalApicId=4 Enabled
Jun 12 10:16:53 localhost kernel[0]: AppleACPICPU: ProcessorId=4 LocalApicId=5 Enabled
Jun 12 10:16:53 localhost kernel[0]: AppleACPICPU: ProcessorId=5 LocalApicId=0 Disabled
Jun 12 10:16:53 localhost kernel[0]: AppleACPICPU: ProcessorId=6 LocalApicId=0 Disabled
Jun 12 10:16:53 localhost kernel[0]: AppleACPICPU: ProcessorId=7 LocalApicId=0 Disabled
Jun 12 10:16:53 localhost kernel[0]: AppleACPICPU: ProcessorId=8 LocalApicId=0 Disabled
Jun 12 10:16:53 localhost kernel[0]: calling mpo_policy_init for TMSafetyNet
Jun 12 10:16:53 localhost kernel[0]: Security policy loaded: Safety net for Time Machine
Jun 12 10:16:53 localhost kernel[0]: calling mpo_policy_init for Sandbox
Jun 12 10:16:53 localhost kernel[0]: Security policy loaded: Seatbelt sandbox policy (Sa
Jun 12 10:16:53 localhost kernel[0]: calling mpo_policy_init for Quarantine
Jun 12 10:16:53 localhost kernel[0]: Security policy loaded: Quarantine policy (Quaranti
Jun 12 10:16:53 localhost kernel[0]: Copyright (c) 1982, 1986, 1989, 1991, 1993
Jun 12 10:16:53 localhost kernel[0]: The Regents of the University of California. All ri
Jun 12 10:16:53 localhost kernel[0]: MAC Framework successfully initialized
Jun 12 10:16:53 localhost kernel[0]: using 16384 buffer headers and 10240 cluster IO bu
Jun 12 10:16:53 localhost kernel[0]: IOAPIC: Version 0x20 Vectors 64:87
```

```
Jun 5 12:07:52 localhost kernel[0]: npvhash=4095
Jun 5 12:07:52 localhost kernel[0]: PAE enabled
Jun 5 12:07:52 localhost kernel[0]: 64 bit mode enabled
Jun 5 12:07:52 localhost kernel[0]: Darwin Kernel Version 10.0.0: Fri Jul 31 22:47:34
Jun 5 12:07:52 localhost kernel[0]: vm_page_bootstrap: 1936010 free pages and 95606 w
Jun 5 12:07:52 localhost kernel[0]: standard timeslicing quantum is 10000 us
Jun 5 12:07:52 localhost kernel[0]: mig_table_max_displ = 73
Jun 5 12:07:52 localhost kernel[0]: AppleACPICPU: ProcessorId=0 LocalApicId=0 Enabled
Jun 5 12:07:52 localhost kernel[0]: AppleACPICPU: ProcessorId=1 LocalApicId=1 Enabled
Jun 5 12:07:52 localhost kernel[0]: calling mpo_policy_init for TMSafetyNet
Jun 5 12:07:52 localhost kernel[0]: Security policy loaded: Safety net for Time Machi
Jun 5 12:07:52 localhost kernel[0]: calling mpo_policy_init for Quarantine
Jun 5 12:07:52 localhost kernel[0]: Security policy loaded: Quarantine policy (Quaran
Jun 5 12:07:52 localhost kernel[0]: calling mpo_policy_init for Sandbox
Jun 5 12:07:52 localhost kernel[0]: Security policy loaded: Seatbelt sandbox policy (S
Jun 5 12:07:52 localhost kernel[0]: Copyright (c) 1982, 1986, 1989, 1991, 1993
Jun 5 12:07:52 localhost kernel[0]: The Regents of the University of California. All ri
Jun 5 12:07:52 localhost kernel[0]: MAC Framework successfully initialized
Jun 5 12:07:52 localhost kernel[0]: using 16384 buffer headers and 4096 cluster IO bu
Jun 5 12:07:52 localhost kernel[0]: IOAPIC: Version 0x11 Vectors 64:87
Jun 5 12:07:52 localhost kernel[0]: ACPI: System State [S0 S3 S4 S5] (S3)
Jun 5 12:07:52 localhost kernel[0]: mbinit: done (64 MB memory set for mbuf pool)
Jun 5 12:07:52 localhost kernel[0]: rooting via boot-uuid from /chosen: 5D895F73-8DB0
```

## Sleep Cause /var/log/kernel.log or system.log

May 26 17:27:02 MBP kernel[0]: Previous Sleep Cause: #

5

- Normal Sleep, Closed Laptop Lid

-60

- Unknown

0

- Hibernation

© SANS,  
All Rights Reserved

Mac Forensic Analysis

The system records reasons the computer has been put to sleep, which is known as “Sleep Cause”. Outlined in the screenshot above are some of the reasons have been documented.

The “Sleep Cause” that should be most prevalent on a working laptop system should be cause number “5”.

# Wake Reason

## /var/log/kernel.log or system.log

```
Jun  9 19:45:46 bit kernel[0]: Wake reason: <Message>
```

RTC (Alarm)	<ul style="list-style-type: none"><li>• Wake on Demand, Bonjour Services - Real Time Clock</li></ul>
EC.LID0, EC.LID0.EHC2, EC.LidOpen, EC.LidOpen.XHC1	<ul style="list-style-type: none"><li>• Laptop Lid</li></ul>
EHC1, EHC2	<ul style="list-style-type: none"><li>• Enhanced Host Controller - USB, Bluetooth, Wireless Devices</li></ul>
PWRB (User)	<ul style="list-style-type: none"><li>• Power Button</li></ul>
OHC1	<ul style="list-style-type: none"><li>• Open Host Controller - USB/Firewire, Mouse/Keyboard</li></ul>
? (User)	<ul style="list-style-type: none"><li>• Power Button from hibernation w/ no battery power</li></ul>
USB1	<ul style="list-style-type: none"><li>• Trackpad</li></ul>
EC.ACAttach (Maintenance), EC.ACDetach (Maintenance)	<ul style="list-style-type: none"><li>• Power Adapter</li></ul>

© SANS.  
All Rights Reserved

Mac Forensic Analysis

The system records the reasons the computer has awoken, known as “Wake Reason”. Outlined in the screenshot above are some of the reasons.

# Shutdown Cause /var/log/kernel.log or system.log

```
Jul 23 17:08:52 localhost kernel[0]: Previous Shutdown Cause: #
```

0 • Battery Removal/Power Plug

3 • Hard Shutdown (Hold Power Button)

5 • Normal Shutdown/Reboot

-128 • Unknown

-60 • Unknown

C SANS.  
All Rights Reserved

Mac Forensic Analysis

The system records the reasons the computer was shut down, known as “Shutdown Cause”. Outlined in the screenshot above are some of the reasons.

The “Shutdown Cause” that should be most prevalent on a working system should be cause number “5”.

The documentation on these Sleep/Wake/Shutdown reasons are limited and more experimentation and research need to be performed.

## System Boot Boot Device

```
May  9 16:29:10 localhost kernel[0]: rooting via boot-
uuid from /chosen: 3981E2E6-0CAC-3A3E-BE1D-90D583F89A5D
May  9 16:29:10 localhost kernel[0]: Waiting on <dict
ID="0"><key>IOProviderClass</key><string
ID="1">IOResources</string><key>IOResourceMatch</key><s
tring ID="2">boot-uuid-media</string></dict>
May  9 16:29:10 localhost kernel[0]: Got boot device =
IOService:/AppleACPIPlatformExpert/PCI0@0/AppleACPIPCI/
SATA@1F,2/AppleIntelPchSeriesAHCI/PRT0@0/IOAHCIDevice@0
/AppleAHCIDiskDriver/IOAHCIBlockStorageDevice/IOBlockSt
orageDriver/WDC WD7500BPKT-75PK4T0
Media/IOGUIDPartitionScheme/Untitled@2
```

© SANS.  
All Rights Reserved

Mac Forensic Analysis

The system boot device is always listed in the boot logging which is shown in either the kernel.log or the system.log (depending on the OS X version). The UUID highlighted above is the UUID of the boot volume, which in most cases is labeled as the “Macintosh HD” volume.

The other highlighted string shows some information about the boot device. In the example shown we can tell that the hard drive on the system (and with a quick Google search) is a Western Digital 750GB hard drive.

# System Boot Boot UUID

```

root@SystemConfiguration:~# diskutil list
/dev/disk0
   #: LUN TYPE NAME          SIZE    IDENTIFIER
   0: GUID_partition_scheme -        -
   1: EFI EFI             500.3 MB /dev/disk0s1
   2: Apple_Boot Recovery HD 650.0 MB /dev/disk0s2
   3: Apple_CoresStorage 799.7 MB /dev/disk0s3
/dev/disk1
   #: LUN TYPE NAME          SIZE    IDENTIFIER
   0: Apple_Boot Recovery HD 449.1 MB /dev/disk1s1

```

**diskutil SystemConfiguration listmounts -list**

Device Identifier	disk1
Device Node	/dev/disk1
Port or Media	disk1
Device / Media Name	Macintosh HD
Volume Name	Macintosh HD
Escaped with Unicode	MacintoshHD
Mounted	Yes
Mount Point	/
Escaped with Unicode	/
File System Personality	Journaled HFS+
Type (Bundle)	HFS
Name (User Visible)	Mac OS Extended (Journaled)
Journal	internal size 40960 KB at offset 0x1238000
Owner	EndUser
Content (00Content):	Apple_HFS
Can be Relocated	No
Recovery Disk	inodes3
Media Type	Generic
Protocol	PCI
SMART Status	Not Supported
Volume UUID:	764C1C9A-3E5D-14B2-A5F4-509446F8EE01
Total Size:	999.1 GB (999GB246768 Bytes) (exactly 974720480 512-Byte-Blocks)
Volume Free Space:	126.6 GB (126555356096 Bytes) (exactly 247175440 512-Byte-Blocks)
Device Block Size:	512 Bytes
Read-Only Media:	No
Read-Only Volume:	No
Ejectable:	No
Wipeable:	Yes
Internal:	Yes
Solid State:	Yes
Low Level Wipe:	No
Low Level Format:	Not supported

Feb 13 11:00:23 localhost kernel[0]:  
 rooting via boot-uuid from /chosen:  
**2603DEB0-8EBD-36ED-A5F4-989446F8EE01**

Feb 13 11:00:23 localhost kernel[0]:  
 Waiting on <dict  
 ID="0"><key>IOProviderClass</key><string  
 ID="1">IOResources</string><key>IOResourc  
 eMatch</key><string ID="2">boot-uuid-  
 media</string></dict>

© SANS,  
 All Rights Reserved

**Mac Forensic Analysis**

Some Mac systems such as the older Mac Pro, can have hard drives easily removed and changed. The boot UUID is unique to that volume on that specific hard drive. The UUID will not change if the hard drive is removed and put into a different system. We can search for the term “/chosen” to see if the system was booted from a different volume or if the hard drive was used to boot multiple systems.

The author once had an instance where they needed to find out what systems a particular hard drive was placed into (and consequently booted from). It turns out the same hard drive was booted from three different systems throughout the years, and it was easy to correlate it using the `kernel.log` (or `system.log`). The author was able to determine, using MAC addresses (next slide), that this volume was booted from three different systems. The boot UUID did not change, but the MAC addresses did change – indicating separate machines. The author was also able to physically locate each system by their MAC addresses, for further analysis.

```

nibble:SystemConfiguration sledwards$ diskutil list
/dev/disk0
 #:          TYPE NAME      SIZE IDENTIFIER
 0: GUID_partition_scheme *500.3 GB disk0
   1:      EFI   EFI        209.7 MB disk0s1
   2: Apple_CoreStorage    499.4 GB disk0s2
   3: Apple_Boot Recovery HD 650.0 MB disk0s3
/dev/disk1
 #:          TYPE NAME      SIZE IDENTIFIER
 0: Apple_HFS Macintosh HD *499.1 GB disk1
nibble:SystemConfiguration sledwards$ diskutil info /dev/disk1
Device Identifier:           disk1
Device Node:                 /dev/disk1
Part of Whole:               disk1
Device / Media Name:         Macintosh HD

Volume Name:                 Macintosh HD
Escaped with Unicode:        Macintosh%FF%FE%20%00HD

Mounted:                     Yes
Mount Point:                 /
Escaped with Unicode:        /

File System Personality:     Journalized HFS+
Type (Bundle):               hfs
Name (User Visible):         Mac OS Extended (Journalized)
Journal:                     Journal size 40960 KB at offset 0x1238b000
Owners:                      Enabled

Content (IOContent):         Apple_HFS
OS Can Be Installed:         Yes
Recovery Disk:               disk0s3
Media Type:                  Generic
Protocol:                    PCI
SMART Status:                Not Supported
Volume UUID:                 2603DEB0-8EBD-36BD-A5F4-989446F8EE01

Total Size:                  499.1 GB (499082485760 Bytes) (exactly 974770480 512-Byte-Units)
Volume Free Space:           126.6 GB (126555856896 Bytes) (exactly 247179408 512-Byte-Units)
Device Block Size:            512 Bytes

Read-Only Media:              No
Read-Only Volume:             No
Ejectable:                   No

Whole:                       Yes
Internal:                    Yes
Solid State:                 Yes
OS 9 Drivers:                No
Low Level Format:             Not supported

```

# System Boot MAC Addresses

- Three different systems, boot from same HDD
- Boot-UUID remains the same
- MAC addresses change for each system
- Correlate an HDD moving to/from systems

```
Jun 14 17:41:59 localhost kernel[0]: rooting via boot-uuid from /chosen: 1FDCE218-B7EB-3BAC-9AD6-  
8498D0E2EA9D
```

```
Jun 14 17:42:22 Sarah-Edwardss-MacBook kernel[0]: yukon: Ethernet address 00:19:e3:3c:cb:7e  
Jun 14 17:42:22 Sarah-Edwardss-MacBook kernel[0]: AirPort_Athrfusion21: Ethernet address  
00:1b:63:c3:8d:1a
```

```
Jun 14 19:50:26 localhost kernel[0]: rooting via boot-uuid from /chosen: 1FDCE218-B7EB-3BAC-9AD6-  
8498D0E2EA9D
```

```
Jun 14 19:50:53 Sarah-Edwardss-MacBook kernel[0]: Airport_Brcm4331: Ethernet address 28:cf:da:04:84:77  
Jun 14 19:50:53 Sarah-Edwardss-MacBook kernel[0]: BCM5701Enet: Ethernet address 3c:07:54:03:65:20
```

```
Jun 14 20:33:38 localhost kernel[0]: rooting via boot-uuid from /chosen: 1FDCE218-B7EB-3BAC-9AD6-  
8498D0E2EA9D
```

```
Jun 14 20:34:12 Sarah-Edwardss-MacBook kernel[0]: BCM5701Enet: Ethernet address c4:2c:03:09:ca:fd  
Jun 14 20:34:12 Sarah-Edwardss-MacBook kernel[0]: Airport_Brcm43224: Ethernet address 90:27:e4:f8:e6:5f
```

© SANS.  
All Rights Reserved.

Mac Forensic Analysis

One way to correlate if a hard drive has been moved from system to system is to check for the MAC addresses.

In the example, each system has two network cards (one wireless, one wired). Right off the bat we can probably rule out the MacBook Air as one of the systems as these only have one wireless NIC by default. The message includes a hint as to what types of network cards are installed on the systems.

- System 1
  - Yukon – Yukon network card
  - AirPort\_Athrfusion21 – Atheros network card
- System 2
  - Airport\_Brcm4331 – Broadcom network card
  - BCM5701Enet – Broadcom network card
- System 3
  - BCM5701Enet – Broadcom network card
  - Airport\_Brcm43224 – Broadcom network card

Jun 14 17:41:59 localhost kernel[0]: rooting via boot-uuid from /chosen:  
1FDCF218-B7EB-3BAC-9AD6-8498D0E2EA9D

Jun 14 17:42:22 Sarah-Edwardss-MacBook kernel[0]: yukon: Ethernet address  
00:19:e3:3c:cb:7e

Jun 14 17:42:22 Sarah-Edwardss-MacBook kernel[0]: AirPort\_AthrFusion21:  
Ethernet address 00:1b:63:c3:8d:1a

Jun 14 19:50:26 localhost kernel[0]: rooting via boot-uuid from /chosen:  
1FDCF218-B7EB-3BAC-9AD6-8498D0E2EA9D

Jun 14 19:50:53 Sarah-Edwardss-MacBook kernel[0]: AirPort\_Brcm4331:  
Ethernet address 28:cf:da:04:84:77

Jun 14 19:50:53 Sarah-Edwardss-MacBook kernel[0]: BCM5701Enet: Ethernet  
address 3c:07:54:03:65:20

Jun 14 20:33:38 localhost kernel[0]: rooting via boot-uuid from /chosen:  
1FDCF218-B7EB-3BAC-9AD6-8498D0E2EA9D

Jun 14 20:34:12 Sarah-Edwardss-MacBook kernel[0]: BCM5701Enet: Ethernet  
address c4:2c:03:09:ca:fd

Jun 14 20:34:12 Sarah-Edwardss-MacBook kernel[0]: AirPort\_Brcm43224:  
Ethernet address 90:27:e4:f8:e6:5f

# Disk Usage History

## /var/log/daily.out

```

Sun May 13 04:02:55 EDT 2012          /dev/disk1 698Gi 109Gi 588Gi 16% /
Removing old temporary files:          /dev/disk1 698Gi 123Gi 574Gi 18% /
Cleaning out old system announcements: /dev/disk1 698Gi 172Gi 525Gi 25% /
Removing stale files from /var/rwho:   /dev/disk1 698Gi 181Gi 517Gi 26% /
Removing scratch fax files           /dev/disk1 698Gi 181Gi 517Gi 26% /
Disk status:
Filesystem  Size  Used  Avail Capacity Mounted on
/dev/disk1  698Gi 109Gi 588Gi 16% /

Network interface status:
Name  Mtu  Network      Address      Ipkts  Ierrs    Opkts  Oerrs  Coll
lo0   16384 <Link#1>          6641727  0  6641727  0  0
lo0   16384 localhost  fe80:1::1  6641727  -  6641727  -  -
lo0   16384 127   localhost          6641727  -  6641727  -  -
lo0   16384 localhost  ::1          6641727  -  6641727  -  -
gif0* 1280  <Link#2>          0  0  0  0  0
stf0* 1280  <Link#3>          0  0  0  0  0
en0   1500  <Link#4>  c4:2c:03:09:ca:fd  0  0  0  0  0
en1   1500  <Link#5>  90:27:e4:f8:e6:5f  1823664  0  2065789  0  0
p2p0* 2304  <Link#6>  02:27:e4:f8:e6:5f  0  0  0  0  0
fw0   4078  <Link#7>  e8:06:88:ff:fe:d5:5d:08  0  0  0  0  0

Local system status:
4:03  up 16:31, 2 users, load averages: 10.59 2.96 1.20

```

© SANS,  
All Rights Reserved

## Mac Forensic Analysis

The disk usage for a system might be useful to see how much disk space a system uses over time. The daily.out is one of the log files associated with the Unix maintenance scripts (along with weekly.out and monthly.out), which records “Disk Status”. We can “grep” this log file for the specific disk (/dev/disk1) as shown in the example above. The example shows that disk space out has increased from 16% to 26%.

Sun May 13 04:02:55 EDT 2012

Removing old temporary files:

Cleaning out old system announcements:

Removing stale files from /var/rwho:

Removing scratch fax files

Disk status:

Filesystem	Size	Used	Avail	Capacity	Mounted on
/dev/disk1	698Gi	109Gi	588Gi	16%	/

Network interface status:

Name	Mtu	Network	Address	Ipkts	Ierrs	Opkts	Oerrs	Coll
lo0	16384	<Link#1>		6641727	0	6641727	0	0
lo0	16384	localhost	fe80:1::1	6641727	-	6641727	-	-
lo0	16384	127	localhost	6641727	-	6641727	-	-
lo0	16384	localhost	::1	6641727	-	6641727	-	-
gif0*	1280	<Link#2>		0	0	0	0	0
stf0*	1280	<Link#3>		0	0	0	0	0
en0	1500	<Link#4>	c4:2c:03:09:ca:fd	0	0	0	0	0
en1	1500	<Link#5>	90:27:e4:f8:e6:5f	1823664	0	2065789	0	0
p2p0*	2304	<Link#6>	02:27:e4:f8:e6:5f	0	0	0	0	0
fw0	4078	<Link#7>	e8:06:88:ff:fe:d5:5d:08	0	0	0	0	0

Local system status:

4:03 up 16:31, 2 users, load averages: 10.59 2.96 1.20

/dev/disk1	698Gi	109Gi	588Gi	16%	/
/dev/disk1	698Gi	123Gi	574Gi	18%	/
/dev/disk1	698Gi	172Gi	525Gi	25%	/
/dev/disk1	698Gi	181Gi	517Gi	26%	/
/dev/disk1	698Gi	181Gi	517Gi	26%	/
/dev/disk1	698Gi	180Gi	517Gi	26%	/
/dev/disk1	698Gi	180Gi	517Gi	26%	/

## Network Analysis

Network Changes

Network Configuration

Wireless Access Points

© SANS.  
All Rights Reserved

Mac Forensic Analysis

The network access of a system can help investigators determine what type of access a system had at a particular time. Travel can be shown by viewing access to various wireless access points.

# Network Changes

## /var/log/system.log

```
Jun 12 13:07:11 bit configd[16]: network configuration
changed.
Jun 12 13:07:11 bit configd[16]: setting hostname to
"bit.local"
Jun 12 13:07:11 bit configd[16]: network configuration
changed.
Jun 12 13:07:24 bit ntpd[50]: bind(25) AF_INET6
fe80::9227:e4ff:fef8:e65f%5#123 flags 0x11 failed: Can't
assign requested address
Jun 12 13:07:24 bit airportd[3218]: _doAutoJoin: Already
associated to "PANERA". Bailing on auto-join.
Jun 12 13:07:28 bit configd[16]: network configuration
changed.
```

cSANS.  
All Rights Reserved

Mac Forensic Analysis

Searches for keywords such as “configd”, “airportd” and “ntpd” will show network configuration changes.

In the screenshot above the local system changed the hostname to “bit.local” and shows that the system was already connected to an access point named “PANERA”.

Most desktop systems may not have as many “network configuration changes” as they may have static network settings. Laptops, due to their mobility, tend to change more often to connect to different access points.

These times are stored in local system time.

## Wireless Access (10.9-) system.log - Search “airportd”

```
Jun 12 10:17:24 bit airportd[36]: _doAutoJoin: Already associated to  
"veyron". Bailing on auto-join.  
Jun 12 11:43:17 bit airportd[3105]: _doAutoJoin: Already associated  
to "veyron". Bailing on auto-join.  
Jun 12 13:07:24 bit airportd[3218]: _doAutoJoin: Already associated  
to "PANERA". Bailing on auto-join.  
Jun 12 13:07:29 bit airportd[3218]: _doAutoJoin: Already associated  
to "PANERA". Bailing on auto-join.  
Jun 12 14:51:42 bit airportd[3756]: _processSystemPSKAssoc: No  
password for network <CWNetwork: 0x7f8083c189b0> [ssid=L.A. Boxing  
Customer WIFI, bssid=00:21:29:d5:20:12, security=WPA/WPA2 Personal,  
rssi=-92, channel=<CWChannel: 0x7f8085106d90> [channelNumber=6(2GHz),  
channelWidth={20MHz}], ibss=0] in the system keychain  
Jun 12 16:49:03 bit airportd[3769]: _doAutoJoin: Already associated  
to "veyron". Bailing on auto-join.
```

© SANS.  
All Rights Reserved

Mac Forensic Analysis

The system.log can be searched for “airportd” to determine when a system had been associated with a specific wireless access point (veyron, PANERA). It may also show if an attempt was made to associate to a wireless access point (L.A. Boxing Customer WIFI, shown in the screenshot above).

These times are stored in local system time.

## Wireless Access (10.10) system.log - Search “BSSID changed”

```
Dec 16 04:56:16 word kernel[0]: en0: BSSID changed to c0:3f:0e:8c:59:5b
Dec 16 04:56:16 word kernel[0]: en0: BSSID changed to c0:3f:0e:8c:59:5b
Dec 16 05:56:48 word kernel[0]: en0: BSSID changed to c0:3f:0e:8c:59:5b
Dec 16 05:56:48 word kernel[0]: en0: BSSID changed to c0:3f:0e:8c:59:5b
Dec 16 05:56:48 word kernel[0]: en0: BSSID changed to c0:3f:0e:8c:59:5b
Dec 16 05:56:52 word kernel[0]: en0: BSSID changed to c0:3f:0e:8c:59:5b
Dec 16 13:43:01 word kernel[0]: en0: BSSID changed to 8c:0c:90:53:1b:98
Dec 16 13:43:06 word kernel[0]: en0: BSSID changed to 8c:0c:90:53:1b:98
Dec 16 16:25:56 word kernel[0]: en0: BSSID changed to c0:3f:0e:8c:59:5b
Dec 16 16:26:02 word kernel[0]: en0: BSSID changed to c0:3f:0e:8c:59:5b
Dec 16 16:26:22 word kernel[0]: en0: BSSID changed to c0:3f:0e:8c:59:5b
Dec 16 17:43:47 word kernel[0]: en0: BSSID changed to c0:3f:0e:8c:59:5b
Dec 16 17:43:55 word kernel[0]: en0: BSSID changed to c0:3f:0e:8c:59:5b
Dec 16 21:09:16 word kernel[0]: en0: BSSID changed to 26:f7:e4:87:b6:da
Dec 16 21:09:28 word kernel[0]: en0: BSSID changed to 26:f7:e4:87:b6:da
Dec 16 21:09:35 word kernel[0]: en0: BSSID changed to c0:3f:0e:8c:59:5b
Dec 16 21:09:37 word kernel[0]: en0: BSSID changed to c0:3f:0e:8c:59:5b
```

© SANS,  
All Rights Reserved

Mac Forensic Analysis

The wireless access configuration logs changed in 10.10. They no long provide SSID names, instead they record the MAC address of the access point.

In the example above, the system used three BSSIDs:

- c0:3f:0e:8c:59:5b
- 8c:0c:90:53:1b:98
- 26:f7:e4:87:b6:da

You may be able to correlate these MAC addresses with names using the com.apple.airport.preferences.plist file.

# User Access

User Login

User Logoff

Access Time

Privilege Escalation

Account Creation

Account Deletion

© SANS.  
All Rights Reserved

Mac Forensic Analysis

Logs can tell a lot about how each user makes use of a system. When did they login or logoff, do they have privileged access, and how long were they on the system.

# User Logins / Logouts

The screenshot shows a window titled "User Logins / Logouts" with four sections: Login Window, Local Terminal, SSH, and Screen Sharing. Each section contains a list of log entries.

- Login Window:**
  - May 28 12:42:23 byte loginwindow[66]: DEAD\_PROCESS: 74 console
  - May 28 14:28:04 byte loginwindow[66]: USER\_PROCESS: 60 console
- Local Terminal:**
  - May 28 14:48:04 byte login[693]: USER\_PROCESS: 693 ttys000
  - May 28 14:48:07 byte login[698]: USER\_PROCESS: 698 ttys001
  - May 28 15:07:29 byte login[812]: USER\_PROCESS: 812 ttys002
  - May 28 15:07:51 byte login[812]: DEAD\_PROCESS: 812 ttys002
- SSH:**
  - May 28 15:15:38 byte sshd[831]: USER\_PROCESS: 842 ttys002
  - May 28 15:15:52 byte sshd[831]: DEAD\_PROCESS: 842 ttys002
- Screen Sharing:**
  - 5/28/12 3:31:33.675 PM screensharingd: Authentication: SUCCEEDED  
:: User Name: Sarah Edwards :: Viewer Address: 192.168.1.101 ::  
Type: DH

At the bottom left is a copyright notice: © SANS. All Rights Reserved. At the bottom right is the title: Mac Forensic Analysis.

User logins and logouts can help in determining the usage of the system, who are the main actors on the system, and who might not be allowed onto the system, but is using it anyway!

OS X systems may be logged into by a variety of means, four of which are specified here.

- Login Window – Via the logon GUI
- Local Terminal – Via the Terminal program
- SSH - OpenSSH
- Screen Sharing – A native VNC program.

Performing a search for logins can be accomplished by using the term “\_PROCESS”, except for screen sharing, which can be found by searching for “screensharingd”.

Each login process will be marked with “USER\_PROCESS” and the process ID , while the logoff will be shown as “DEAD\_PROCESS” and the matching process ID.

In the Local Terminal example, we can see the results of three logins (PIDs 693, 698, 812) but only one logoff (PID 812). Three terminal windows have been opened, ttys000, ttys001 and ttys002. The terminal window labeled ttys002 has been exited out of or closed. Note this type uses the ‘login’ process.

The Login Window example shows a login (PID 74) and logoff (PID 60) for two different sessions, labeled as ‘console’. Note this type uses the ‘loginwindow’ process.

SSH uses the ‘sshd’ process to record logins and logoffs. In the example, one login/logoff pair for PID 842 is shown using the ttys002 terminal window.

Screen Sharing logins show an entry like the one shown in the example above. It records the username (may be Full Name or username) and the IP address where the connection is coming from.

These login/logout messages can be found in the system.log or the Apple System Logs. These events in the ASL logs contain more detailed information such as which user logged in.

# Log Analysis monthly.out

- Account Audit
- Monthly
- Uses ac -p command to calculate account time on system
- “Accumulated connected time in decimal hours”

```
-- End of monthly output --
Wed Apr 4 09:15:54 EDT 2012
Rotating fax log files:
Doing login accounting:
      total      3678.85
      sledwards  3678.76
      root       0.09
-- End of monthly output --
Tue May 1 05:30:00 PDT 2012
Rotating fax log files:
Doing login accounting:
      total      4301.95
      sledwards  4301.77
      root       0.18
-- End of monthly output --
Fri Jun 1 06:46:13 PDT 2012
Rotating fax log files:
Doing login accounting:
      total      5047.22
      sledwards  5047.04
      root       0.18
-- End of monthly output --
```

© SANS.  
All Rights Reserved

Mac Forensic Analysis

The monthly maintenance script output, shown in the screenshot, contains data about user accounts on the system. This can be used to determine which accounts get used more often than others.

The time next to each account name is the accumulated connection time, in decimal hours, created by running the ‘ac -p’ command.

The example shows the sledwards account gets used much more often than the other root account. It is worth noting that the root account is not enabled by default on OS X systems. The output shows the root user account being used slightly more in May than it was in April.

## Privilege Escalation /var/log/system.log

### SU

- 5/27/12 8:54:21.646 PM su: BAD SU oompa to root on /dev/ttys001
- 5/28/12 8:57:44.032 PM su: oompa to root on /dev/ttys000

### sudo

- 5/27/12 8:48:15.790 PM sudo: oompa :  
TTY=ttys000 ; PWD=/Users/oompa/Documents ;  
USER=root ; COMMAND=/usr/bin/iosnoop

© SANS,  
All Rights Reserved

Mac Forensic Analysis

Privilege escalation is needed when a Standard user needs to perform root level actions. The command `su` can be used to login as root (or another user) for a session, while the `sudo` command will run a command as root for five minutes (this time can be changed in the `/etc/sudoers` configuration file).

The `su` example shows an entry that failed, while the second entry shows a successful `su` command.

The `sudo` example shows a successful command using `sudo`. The entry includes:

- Terminal window
- Current working directory
- Escalated user account
- Command

# Account Creation

## Audit Logs

```
<record version="11" event="create user" modifier="0"
time="Mon May 28 21:25:49 2012" msec=" + 677 msec">
<subject audit-uid="501" uid="501" gid="20" ruid="501"
rgid="20" pid="585" sid="100004" tid="585 0.0.0.0" />
<text>Create record type Users
&apos;supersecretuser&apos; node
&apos;/Local/Default&apos;</text>
<return errval="success" retval="0" />
</record>
```

## secure.log or system.log (10.8+)

```
• May 28 21:25:22 bit com.apple.SecurityServer[24]: UID
501 authenticated as user compa (UID 501) for right
'system.preferences.accounts'
```

© SANS.  
All Rights Reserved

Mac Forensic Analysis

User account creation can also be of immense interest to an investigator. Audit records for user creation are very verbose. Lots of related entries are created. Audit records also have more detail than records found in the secure.log (or system.log).

One of the records shown in the example above, contains the event “create user”. This entry includes the following data:

- Timestamp
- User who created the new user (uid=501)
- Name of new user (supersecretuser)

In the next example, the secure.log (or the system.log with 10.8+) entry shows who unlocked the account preferences pane. In this log event, it does not show if a new user account was created or what the account name was. In fact, by just looking at this log entry we wouldn’t know a new account was created.

This is a good example of using multiple sources to get a more detailed picture.

# Account Deletion

/Library/Preferences/com.apple.preferences.accounts.plist

Key	Type	Value
▼ deletedUsers	Array	(2 items)
► Item 0	Diction...	(4 items)
▼ Item 1	Diction...	(4 items)
dsAttrTypeStandard:RealName	String	testuser
dsAttrTypeStandard:UniqueID	Number	502
name	String	testuser
date	Date	Jun 13, 2012 8:41:58 PM

```
<record version="11" event="delete user" modifier="0" time="Wed Jun 13 20:41:56  
2012" msec=" + 322 msec" >  
<subject audit-uid="501" uid="501" gid="20" ruid="501" rgid="20" pid="10717"  
sid="100005" tid="10717 0.0.0.0" />  
<text>Delete record type Users &apos;testuser&apos; node  
&apos;/Local/Default&apos;;</text>  
<return errval="success" retval="0" />  
</record>
```

© SANS,  
All Rights Reserved

Mac Forensic Analysis

If account creation is important, then so is account deletion. The same data can be found in the audit logs that can be found for account creation. The event “delete user” records the user (501) who deleted the account, and what account was deleted (testuser) and when.

The deleted users are shown in the com.apple.preferences.accounts.plist file under the deletedUsers key. This key contains the deleted user’s name, UID, username, and the deletion date in local system time.

## Software Installs

Software Updates

Administrator Installs

OS X Installation & Versions

© SANS.  
All Rights Reserved.

Mac Forensic Analysis

Logs can show us when the software was installed and if Administrator privileges were needed. These logs can also show what OS X version is installed and when it was updated.

## Install Details /var/log/install.log

```
May 27 11:59:03 MBP Installer[470]: logKext Installation Log
May 27 11:59:03 MBP Installer[470]: Opened from:
/Users/compa/Downloads/logKext-2.3.pkg
May 27 11:59:03 MBP Installer[470]: Product archive
/Users/compa/Downloads/logKext-2.3.pkg trustLevel=100
May 27 11:59:17 MBP Installer[470]: InstallerStatusNotifications plugin loaded
May 27 11:59:26 MBP runner[477]: Administrator authorization granted.
May 27 11:59:26 MBP Installer[470]:
=====
=
May 27 11:59:26 MBP Installer[470]: User picked Standard Install
May 27 11:59:26 MBP Installer[470]: Choices selected for installation:
...
May 27 12:01:34 MBP installd[481]: Installed "logKext" ()
May 27 12:01:35 MBP installd[481]: PackageKit: ----- End install -----
```

The `install.log` also contains software install details such as:

- Where the software package was opened from on disk
- Was administrator authorization needed

## Installed Software /var/log/install.log – Search “Installed”

```
May  9 16:28:06 localhost OSInstaller[328]: Installed "Mac OS X" ()  
May  9 19:56:21 bit installd[338]: Installed "Evernote" ()  
May 10 00:45:34 bit installd[559]: Installed "Flashback malware removal tool" (1.0)  
May 10 00:45:34 bit installd[559]: Installed "Mac OS X Update Combined" (10.7.4)  
May 10 00:45:34 bit installd[559]: Installed "iTunes" (10.6.1)  
May 10 00:46:33 bit installd[559]: Installed "Lion Recovery Update" (1.0)  
May 10 16:51:51 bit installd[295]: Installed "Xcode" ()  
May 10 16:55:55 bit installd[295]: Installed "iPhoto" ()  
May 11 19:51:09 bit installd[4384]: Installed "Office 2011 14.1.0 Update" ()  
May 14 18:31:44 bit installd[9572]: Installed "Java for OS X 2012-003" (1.0)  
May 19 16:50:20 bit installd[20691]: Installed "TrueCrypt 7.1a" ()  
May 19 17:17:25 bit installd[20847]: Installed "CCleaner" ()  
May 19 17:32:19 bit installd[20847]: Installed "TextWrangler" ()  
May 26 20:15:45 bit installd[39022]: Installed "The Unarchiver" ()  
May 27 15:46:56 bit installd[41936]: Installed "Wireshark 1.6.8 Intel 64" ()  
May 27 20:57:48 bit installd[514]: Installed "Microsoft Error Reporting for Mac" ()  
May 27 20:59:41 bit installd[978]: Installed "Office 2011 14.2.2 Update" ()
```

© SANS,  
All Rights Reserved

Mac Forensic Analysis

Software installations can be a good resource to find out what types of software are used on the system and when they were downloaded.

We can look at the `install.log` on the system and search for the keyword “Installed” (or more specifically “: Installed”) to easily show the software packages installed. This list will not only include user-initiated software like TrueCrypt and TextWrangler but also system software updates like the iTunes 10.6.1 update.

This log does not include installation information of command-line tools such as `macports` or `fink` repositories.

**Caveat:** Software installed via a “Drag and Drop” method such as Firefox and Chrome will not show in this log. Some software allows a user to copy the application directly to the `/Application` directory.

## System Version

### /var/log/install.log – Search “Build:”

```
May  9 16:14:10 localhost Install Mac OS X Lion[339]: Running OS Build: Mac OS X 10.7 (11A511)
May  9 16:19:25 localhost OSInstaller[328]: Running OS Build: Mac OS X 10.7 (11A511)
May 11 19:23:47 bit Installer[3177]: Running OS Build: Mac OS X 10.7.4 (11E53)
May 11 19:40:47 bit Installer[3755]: Running OS Build: Mac OS X 10.7.4 (11E53)
May 11 19:49:02 bit Installer[4114]: Running OS Build: Mac OS X 10.7.4 (11E53)
```

```
Jul 15 17:54:37 word Install OS X Mavericks[406]: Running OS Build: Mac OS X 10.9 (13A2093)
Jul 15 19:19:54 localhost OSInstaller[375]: Running OS Build: Mac OS X 10.9.4 (13E28)
Jul 16 19:42:59 compas-mbp Installer[1647]: Running OS Build: Mac OS X 10.9.4 (13E28)
```

© SANS.  
All Rights Reserved.

Mac Forensic Analysis

A historical view of the system version can be seen if a search for “Build:” is performed.

In the first example above, the system was first installed on May 9<sup>th</sup> with Mac OS X 10.7. A small clue to tell us this is the first install is that the hostname of the system has not yet been set.

The second example shows the same artifacts for a 10.9 to 10.9.4 system install.

## System Version – Search “Darwin” /var/log/kernel.log or system.log

```
Jul 22 06:49:23 localhost kernel[0]: Darwin Kernel Version 11.0.0: Sat Jun 18  
12:56:35 PDT 2011; root:xnu-1699.22.73~1/RELEASE_X86_64  
Aug  8 21:43:11 localhost kernel[0]: Darwin Kernel Version 11.0.0: Sat Jun 18  
12:56:35 PDT 2011; root:xnu-1699.22.73~1/RELEASE_X86_64  
Aug 20 20:21:18 localhost kernel[0]: Darwin Kernel Version 11.1.0: Tue Jul 26  
16:07:11 PDT 2011; root:xnu-1699.22.81~1/RELEASE_X86_64  
Oct  5 06:59:00 localhost kernel[0]: Darwin Kernel Version 11.1.0: Tue Jul 26  
16:07:11 PDT 2011; root:xnu-1699.22.81~1/RELEASE_X86_64  
Oct 12 19:36:33 localhost kernel[0]: Darwin Kernel Version 11.2.0: Tue Aug  9  
20:54:00 PDT 2011; root:xnu-1699.24.8~1/RELEASE_X86_64  
Dec 30 19:21:03 localhost kernel[0]: Darwin Kernel Version 11.2.0: Tue Aug  9  
20:54:00 PDT 2011; root:xnu-1699.24.8~1/RELEASE_X86_64  
Feb  2 20:05:19 localhost kernel[0]: Darwin Kernel Version 11.3.0: Thu Jan 12  
18:47:41 PST 2012; root:xnu-1699.24.23~1/RELEASE_X86_64  
Apr  8 15:13:53 localhost kernel[0]: Darwin Kernel Version 11.3.0: Thu Jan 12  
18:47:41 PST 2012; root:xnu-1699.24.23~1/RELEASE_X86_64  
May 10 19:35:18 localhost kernel[0]: Darwin Kernel Version 11.4.0: Mon Apr  9  
19:32:15 PDT 2012; root:xnu-1699.26.8~1/RELEASE_X86_64
```

© SANS,  
All Rights Reserved

Mac Forensic Analysis

Using the kernel log (10.7 and below) or the system.log (10.8+), we can view the kernel versions in use at a given time by searching for the term “Darwin”. The kernel versions can be traced back to specific OS X versions.

For example:

Kernel version 11.0.0 is OS X version 10.7  
Kernel version 11.4.0 is OS X version 10.7.4  
Kernel version 12.0.0 is OS X version 10.8  
Kernel version 12.5.0 is OS X version 10.8.5  
Kernel version 13.0.0 is OS X version 10.9  
Kernel version 14.0.0 is OS X version 10.10

Reference:

[http://en.wikipedia.org/wiki/Darwin\\_\(operating\\_system\)](http://en.wikipedia.org/wiki/Darwin_(operating_system))

## Backup Activity

Other Evidential Items

Backups

System Usage

© SANS.  
All Rights Reserved

Mac Forensic Analysis

Evidence of backups can provide new or unknown evidential items for investigators. Backups can provide snapshots in time of a particular system, and can show how often and how much a system was used.

# Backup Log Entry /var/log/system.log

```
Jun 16 15:18:10 bit com.apple.backupd[1957]: Starting standard backup
Jun 16 15:18:10 bit com.apple.backupd[1957]: Attempting to mount network destination URL:
afp://Sarah%20Edwards;AUTH=SRP&Delorean.local/Data
Jun 16 15:18:19 bit com.apple.backupd[1957]: Mounted network destination at mountpoint: /Volumes/Data
using URL: afp://Sarah%20Edwards;AUTH=SRP&Delorean.local/Data
Jun 16 15:19:23 bit com.apple.backupd[1957]: QUICKCHECK ONLY; FILESYSTEM CLEAN
Jun 16 15:19:26 bit com.apple.backupd[1957]: Disk image /Volumes/Data/bit.sparsabundle mounted at:
/Volumes/Time Machine Backups
Jun 16 15:19:26 bit com.apple.backupd[1957]: Backing up to: /Volumes/Time Machine Backups/Backups.backupdb
Jun 16 12:19:00 bit com.apple.backupd[1957]: 100.0 MB required (including padding), 516.13 GB available
Jun 16 12:19:00 bit com.apple.backupd[1957]: Waiting for index to be ready (101)
Jun 16 12:22:08 bit com.apple.backupd[1957]: Copied 1115 files (26.1 MB) from volume LION.
Jun 16 12:22:09 bit com.apple.backupd[1957]: 1.23 GB required (including padding), 516.13 GB available
Jun 16 12:22:51 bit com.apple.backupd[1957]: Copied 971 files (1.1 MB) from volume LION.
Jun 16 12:22:57 bit com.apple.backupd[1957]: Starting post-backup thinning
Jun 16 12:23:43 bit com.apple.backupd[1957]: Deleted /Volumes/Time Machine
Backups/Backups.backupdb/bit/2012-05-19-004000 (21.3 MB)
Jun 16 12:24:22 bit com.apple.backupd[1957]: Deleted /Volumes/Time Machine
Backups/Backups.backupdb/bit/2012-06-08-004822 (87.3 MB)
Jun 16 12:25:11 bit com.apple.backupd[1957]: Deleted /Volumes/Time Machine
Backups/Backups.backupdb/bit/2012-06-10-002525 (168.2 MB)
Jun 16 12:25:11 bit com.apple.backupd[1957]: Post-back up thinning complete: 3 expired backups removed
Jun 16 12:25:11 bit com.apple.backupd[1957]: Backup completed successfully.
Jun 16 12:25:51 bit com.apple.backupd[1957]: Ejected Time Machine disk image.
Jun 16 12:25:51 bit com.apple.backupd[1957]: Ejected Time Machine network volume.
```

© SANS,  
All Rights Reserved

## Mac Forensic Analysis

Access to other systems that contain backups could be a potential source of more evidence. OS X uses the application Time Machine to interact with backup drives. The system.log file contains various backup related events such as:

- When the backup was started
- Network location of backup (or local location if a USB HDD is used)
- Local mount point of network hard drive
- Amount of data backed up
- Volume name to be backed up
- Deletion of old backups
- When the backup completed

Jun 16 15:18:10 bit com.apple.backupd[1957]: **Starting standard backup**

Jun 16 15:18:10 bit com.apple.backupd[1957]: Attempting to mount network destination URL:  
**afp://Sarah%20Edwards;AUTH=SRP@Delorean.local/Data**

Jun 16 15:18:19 bit com.apple.backupd[1957]: Mounted network destination at mountpoint: /Volumes/Data using URL:  
**afp://Sarah%20Edwards;AUTH=SRP@Delorean.local/Data**

Jun 16 15:18:23 bit com.apple.backupd[1957]: QUICKCHECK ONLY;  
FILESYSTEM CLEAN

Jun 16 15:18:26 bit com.apple.backupd[1957]: Disk image  
**/Volumes/Data/bit.sparsebundle** mounted at: **/Volumes/Time Machine Backups**

Jun 16 15:18:26 bit com.apple.backupd[1957]: Backing up to:  
**/Volumes/Time Machine Backups/Backups.backupdb**

Jun 16 12:19:00 bit com.apple.backupd[1957]: 100.0 MB required  
(including padding), 516.13 GB available

Jun 16 12:19:00 bit com.apple.backupd[1957]: Waiting for index to  
be ready (101)

Jun 16 12:22:08 bit com.apple.backupd[1957]: Copied 1115 files  
(26.1 MB) from volume **LION**.

Jun 16 12:22:09 bit com.apple.backupd[1957]: 1.23 GB required  
(including padding), 516.13 GB available

Jun 16 12:22:51 bit com.apple.backupd[1957]: Copied 971 files (1.1  
MB) from volume LION.

Jun 16 12:22:57 bit com.apple.backupd[1957]: Starting post-backup  
thinning

Jun 16 12:23:43 bit com.apple.backupd[1957]: **Deleted /Volumes/Time Machine Backups/Backups.backupdb/bit/2012-05-19-004000 (21.3 MB)**

Jun 16 12:24:22 bit com.apple.backupd[1957]: Deleted /Volumes/Time Machine Backups/Backups.backupdb/bit/2012-06-08-004822 (87.3 MB)

Jun 16 12:25:11 bit com.apple.backupd[1957]: Deleted /Volumes/Time Machine Backups/Backups.backupdb/bit/2012-06-10-002525 (168.2 MB)

Jun 16 12:25:11 bit com.apple.backupd[1957]: Post-back up thinning  
complete: 3 expired backups removed

Jun 16 12:25:11 bit com.apple.backupd[1957]: **Backup completed successfully.**

Jun 16 12:25:51 bit com.apple.backupd[1957]: Ejected Time Machine disk image.

Jun 16 12:25:51 bit com.apple.backupd[1957]: Ejected Time Machine network volume.

## Locational Activity

---

Wireless Access Point Location

Travel Timeline

Local and International Travel

© SANS,  
All Rights Reserved

Mac Forensic Analysis

There are many data points in logs showing locational activity, specifically on laptop systems. Laptops are meant to be mobile and to travel around the city, country, or world. The logs can provide a timeline of travel which can help an investigator correlate where a user may have been at a particular time.

# Wireless Networks

com.apple.airport.preferences.plist

- Determine general location based upon SSID
- Last Connected Time – Local System Time

The screenshot shows the Mac Forensic Analysis interface. On the left, a tree view displays the contents of the 'com.apple.airport.preferences.plist' file, specifically focusing on the 'LastConnected' and 'SSIDString' keys for five networks: 'veyron', 'Washington Dulles WiFi', 'Marriott Guest', 'Marriott Conference', and 'CLTNET'. On the right, a window titled 'Network' shows the 'Preferred Networks' tab, listing the same five networks with their SSID strings: 'veyron', 'Washington Dulles WiFi', 'Marriott Guest', 'Marriott Conference', and 'CLTNET'. The 'Marriott Guest' network is currently selected.

**com.apple.airport.preferences.plist**

Key	Type	Data	Date
LastConnected	Date	<26657972 kICe>	Jun 13, 2012 9:16:56 AM
SSID	String	veyron	
SSIDString	String	WIFID_P_veyron	
... (other entries for Washington Dulles WiFi, Marriott Guest, Marriott Conference, CLTNET)			

**Network**

Network	SSID	Type	Auth Type	Encryption	Channel	Quality	Strength
veyron	veyron	WPA2	WPA2	TKIP	6	100%	-52 dBm
Washington Dulles WiFi	Washington Dulles WiFi	WPA2	WPA2	TKIP	6	100%	-52 dBm
Marriott Guest	Marriott Guest	WPA2	WPA2	TKIP	6	100%	-52 dBm
Marriott Conference	Marriott Conference	WPA2	WPA2	TKIP	6	100%	-52 dBm
CLTNET	CLTNET	WPA2	WPA2	TKIP	6	100%	-52 dBm

C SANS,  
All Rights Reserved

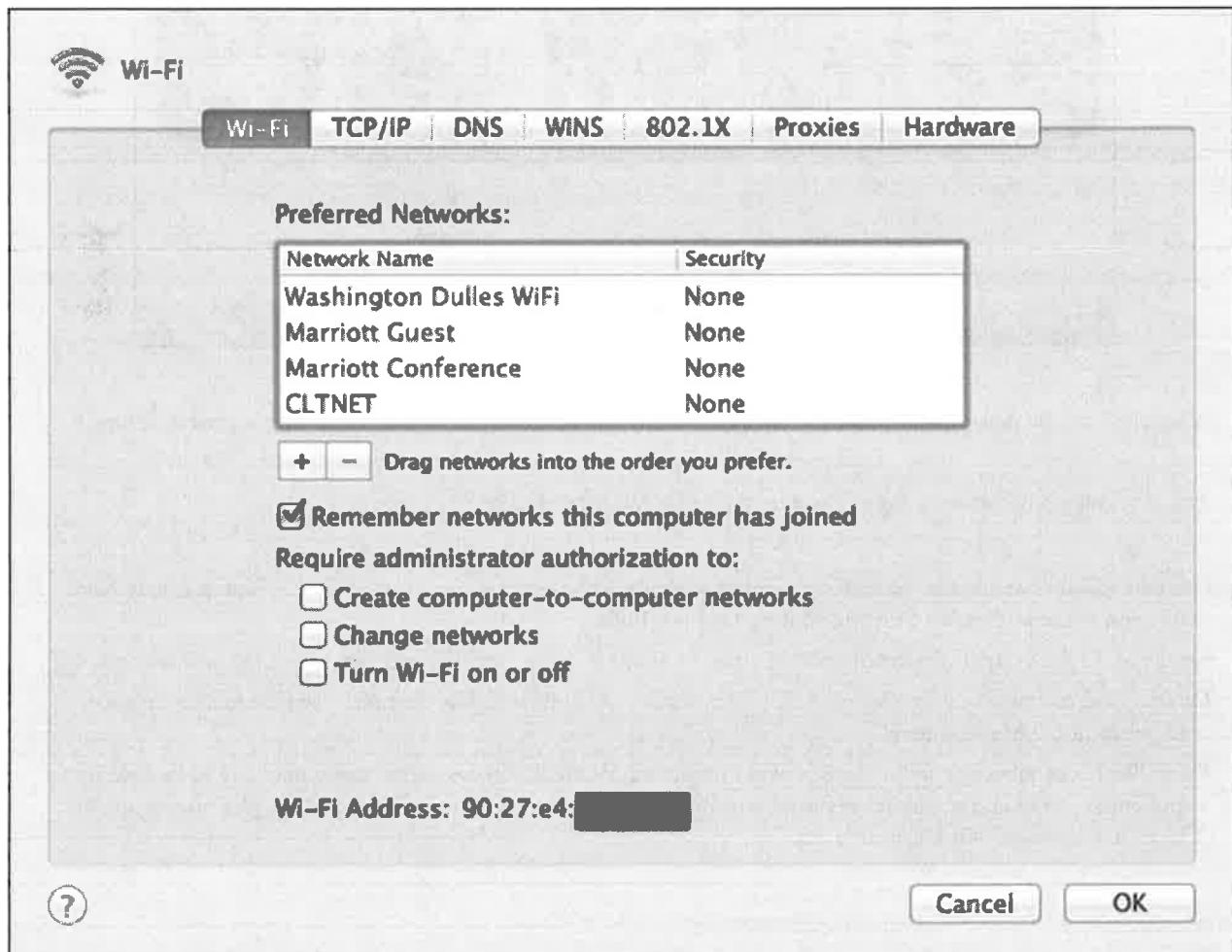
Mac Forensic Analysis

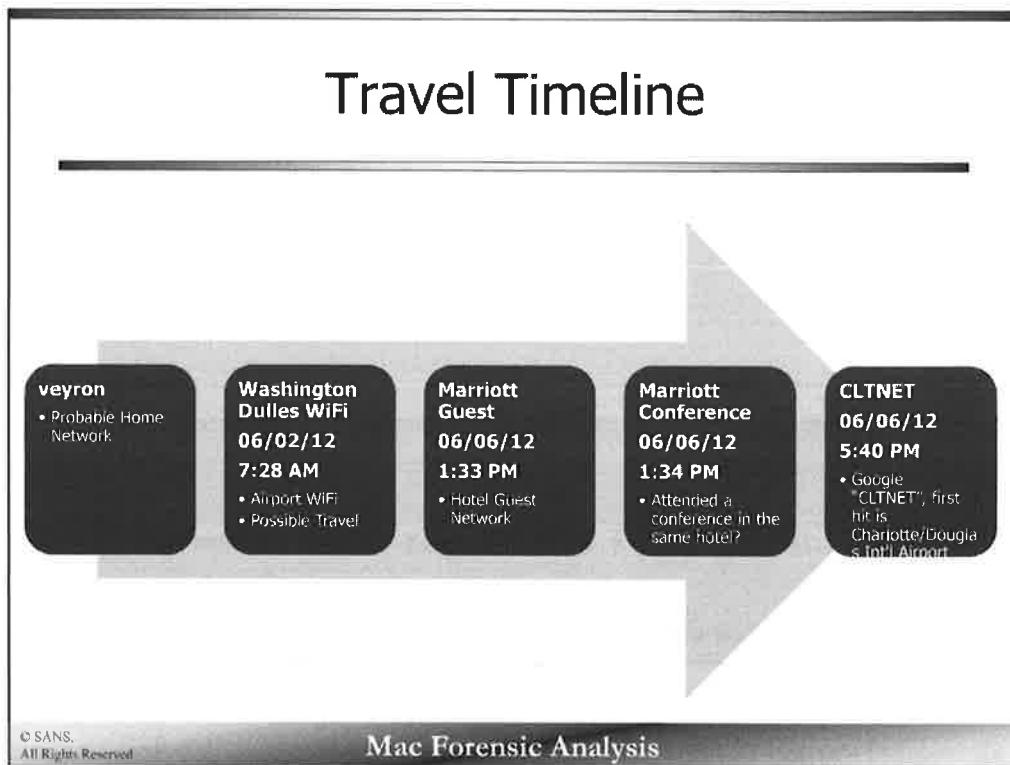
The com.apple.airport.preferences property list contains the “remembered” networks. These networks are saved by default on OS X until the user purges them.

The sidebar shows the raw property list data that the “Preferred Networks” get populated from. The highlighted sections show the last connected time and network SSID string for five networks:

- veyron
- Washington Dulles WiFi
- Marriott Guest
- Marriott Conference
- CLTNET

Key	Type	Value
LastConnected	Date	Jun 13, 2012 9:16:56 AM
SSID	Data	<76657972 6f6e>
SSIDString	String	veyron
SecurityType	String	WPA2 Personal
SystemMode	Boolean	YES
TemporarilyDisabled	Boolean	NO
▼ Item 1	Dictionary	(11 items)
AutoLogin	Boolean	NO
► CachedScanRecord	Dictionary	(14 items)
Captive	Boolean	NO
Closed	Boolean	NO
Disabled	Boolean	NO
LastConnected	Date	Jun 2, 2012 7:28:21 AM
SSID	Data	<57617368 696e5774 6f6
SSIDString	String	Washington Dulles WiFi
SecurityType	String	Open
SystemMode	Boolean	YES
TemporarilyDisabled	Boolean	NO
▼ Item 2	Dictionary	(11 items)
AutoLogin	Boolean	NO
► CachedScanRecord	Dictionary	(15 items)
Captive	Boolean	YES
Closed	Boolean	NO
Disabled	Boolean	NO
LastConnected	Date	Jun 6, 2012 1:33:59 PM
SSID	Data	<4d617272 696f7474 204
SSIDString	String	Marriott Guest
SecurityType	String	Open
SystemMode	Boolean	YES
TemporarilyDisabled	Boolean	NO
▼ Item 3	Dictionary	(11 items)
AutoLogin	Boolean	NO
► CachedScanRecord	Dictionary	(13 items)
Captive	Boolean	YES
Closed	Boolean	NO
Disabled	Boolean	NO
LastConnected	Date	Jun 6, 2012 1:34:40 PM
SSID	Data	<4d617272 696f7474 204
SSIDString	String	Marriott Conference
SecurityType	String	Open
SystemMode	Boolean	YES
TemporarilyDisabled	Boolean	NO
▼ Item 4	Dictionary	(11 items)
AutoLogin	Boolean	NO
► CachedScanRecord	Dictionary	(14 items)
Captive	Boolean	NO
Closed	Boolean	NO
Disabled	Boolean	NO
LastConnected	Date	Jun 6, 2012 5:40:22 PM
SSID	Data	<434c544e 4554>
SSIDString	String	CLTNET





A timeline can be developed from the networks to show where the device has traveled over a period of time.

The probable home network for this system has been determined to be “veyron”.

On 06/02/12 the system can be seen connecting to the “Washington Dulles WiFi”. The user may have connected to the airport Wi-Fi while waiting for their flight.

On 06/06/12 the system connected to “Marriott Guest”. This may be the hotel where the user was staying.

On 06/06/12 a connection to “Marriott Conference” is shown. The user may have been attending a conference at the Marriott hotel.

On 06/06/12 a connection to “CLTNET” was completed. While this access point name may not be as descriptive as the others, a Google search for it shows it as the wireless network used at Charlotte/Douglas International Airport in Charlotte, North Carolina.

## Detailed Timeline /var/log/system.log - search “airportd”

```
Jun  1 19:52:04 bit airportd[3492]: _doAutoJoin: Already associated to "veyron". Bailing on auto-join.
Jun  2 07:24:23 bit airportd[3848]: _doAutoJoin: Already associated to "Washington Dulles WiFi". Bailing on
auto-join.
Jun  2 14:44:32 bit airportd[4944]: _doAutoJoin: Already associated to "Marriott Guest". Bailing on auto-join.
Jun  3 17:12:14 bit airportd[6538]: _doAutoJoin: Already associated to "Marriott Guest". Bailing on auto-join.
Jun  4 01:33:29 bit airportd[7841]: _doAutoJoin: Already associated to "Marriott Guest". Bailing on auto-join.
Jun  5 08:50:16 bit airportd[17054]: _doAutoJoin: Already associated to "Marriott Guest". Bailing on auto-join.
Jun  6 13:34:01 bit airportd[20160]: _doAutoJoin: Already associated to "Marriott Guest". Bailing on auto-join.
Jun  6 13:34:40 bit airportd[20160]: _doAutoJoin: Already associated to "Marriott Conference". Bailing on auto-
join.
Jun  6 17:40:23 bit airportd[20286]: _doAutoJoin: Already associated to "CLTNET". Bailing on auto-join.
Jun  9 09:24:24 bit airportd[25924]: _doAutoJoin: Already associated to "veyron". Bailing on auto-join.
Jun 12 13:07:24 bit airportd[3218]: _doAutoJoin: Already associated to "PANERA". Bailing on auto-join.
Jun 12 18:49:03 bit airportd[3769]: _doAutoJoin: Already associated to "veyron". Bailing on auto-join.
```

© SANS.  
All Rights Reserved

Mac Forensic Analysis

Searching for “airportd” in the system.log can show a more detailed timeline of wireless activity. The same network access points mentioned previously are connected at specific times.

06/01/12 – Home network ‘veyron’  
06/02/12 – Airport Wi-Fi – ‘Washington Dulles Airport WiFi’  
06/02/12 – Hotel Wi-Fi – ‘Marriott Guest’  
06/03/12 – Hotel Wi-Fi – ‘Marriott Guest’  
06/04/12 – Hotel Wi-Fi – ‘Marriott Guest’  
06/05/12 – Hotel Wi-Fi – ‘Marriott Guest’  
06/06/12 – Hotel Wi-Fi – ‘Marriott Guest’  
06/06/12 – Conference Wi-Fi – ‘Marriott Conference’  
06/06/12 – Airport Wi-Fi – ‘CLTNET’  
06/09/12 – Home network – ‘veyron’  
06/12/12 – Restaurant – ‘PANERA’  
06/12/12 – Home network – ‘veyron’

Jun 1 19:52:04 bit airportd[3492]: \_doAutoJoin: Already associated to "veyron". Bailing on auto-join.

Jun 2 07:24:23 bit airportd[3848]: \_doAutoJoin: Already associated to "Washington Dulles WiFi". Bailing on auto-join.

Jun 2 14:44:32 bit airportd[4944]: \_doAutoJoin: Already associated to "Marriott Guest". Bailing on auto-join.

Jun 3 17:12:14 bit airportd[6538]: \_doAutoJoin: Already associated to "Marriott Guest". Bailing on auto-join.

Jun 4 01:33:29 bit airportd[7841]: \_doAutoJoin: Already associated to "Marriott Guest". Bailing on auto-join.

Jun 5 08:50:16 bit airportd[17054]: \_doAutoJoin: Already associated to "Marriott Guest". Bailing on auto-join.

Jun 6 13:34:01 bit airportd[20160]: \_doAutoJoin: Already associated to "Marriott Guest". Bailing on auto-join.

Jun 6 13:34:40 bit airportd[20160]: \_doAutoJoin: Already associated to "Marriott Conference". Bailing on auto-join.

Jun 6 17:40:23 bit airportd[20286]: \_doAutoJoin: Already associated to "CLTNET". Bailing on auto-join.

Jun 9 09:24:24 bit airportd[25724]: \_doAutoJoin: Already associated to "veyron". Bailing on auto-join.

Jun 12 13:07:24 bit airportd[3218]: \_doAutoJoin: Already associated to "PANERA". Bailing on auto-join.

Jun 12 16:49:03 bit airportd[3769]: \_doAutoJoin: Already associated to "veyron". Bailing on auto-join.

# Country Codes - kernel.log & system.log

## Search "country code"

```
Aug  5 09:49:13 MBP kernel[0]: en1: 802.11d country code set to 'US'.
Aug  5 09:49:13 MBP kernel[0]: en1: Supported channels 1 2 3 4 5 6 7 8 9 10 11 36 40 44 48 52
56 60 64 100 104 108 112 116 120 124 128 132 136 140 149 153 157 161 165
Aug  5 09:49:40 MBP kernel[0]: Auth result for: 00:0c:e5:0e:65:bd MAC AUTH succeeded
Aug  5 09:49:40 MBP kernel[0]: AirPort: Link Up on en1

Sep  1 17:42:13 MBP kernel[0]: en1: 802.11d country code set to 'AU'.
Sep  1 17:42:13 MBP kernel[0]: en1: Supported channels 1 2 3 4 5 6 7 8 9 10 11 12 13 36 40 44
48 52 56 60 64 149 153 157 161 165
Sep  1 17:46:13 MBP kernel[0]: Auth result for: 00:26:b0:fe:76:74 MAC AUTH succeeded
Sep  1 17:46:13 MBP kernel[0]: AirPort: Link Up on en1

Jun  5 12:08:49 MBP kernel[0]: en1: 802.11d country code set to 'SE'.
Jun  5 12:08:49 MBP kernel[0]: en1: Supported channels 1 2 3 4 5 6 7 8 9 10 11 12 13 36 40 44
48 52 56 60 64 100 104 108 112 116 120 124 128 132 136 140
Jun  5 12:09:14 MBP kernel[0]: Auth result for: 98:f0:77:2f:75:70 MAC AUTH succeeded
Jun  5 12:09:14 MBP kernel[0]: AirPort: Link Up on en1

Aug  5 09:49:07 MBP kernel[0]: en1: 802.11d country code set to 'X0'.
Aug  5 09:49:07 MBP kernel[0]: en1: Supported channels 1 2 3 4 5 6 7 8 9 10 11 36 40 44 48 52
56 60 64 100 104 108 112 116 120 124 128 132 136 140 149 153 157 161 165
Aug  5 09:49:10 MBP kernel[0]: NVEthernet::setLinkStatus - Valid but not Active
Aug  5 09:49:10 MBP kernel[0]: NVEthernet::mediaChanged - Link is down
Aug  5 09:49:10 MBP kernel[0]: NVEthernet::setLinkStatus - Valid but not Active
```

© SANS.  
All Rights Reserved

Mac Forensic Analysis

International travel may also be very interesting for your investigation. The country codes for wireless access points are recorded in the kernel.log (In 10.7 and before) or the system.log (10.8+),

802.11d is an amendment to 802.11 that allows specification on regulatory domains that includes country information to beacons.

In the example above, the country codes are shown whenever a connection is made to a wireless access point. Each connection is colored in a different color to show related records.

- The first connection shows it was connected to an access point in Australia (AU) on September 1<sup>st</sup>.
- The second connection to the country code (X0) is the default country code when one is not available or is being determined.
- The third connection shows a connection to an access point in the United States (US) on August 5<sup>th</sup>.
- The fourth connection on June 5<sup>th</sup> shows a connection to wireless in Sweden (SE).

```
Sep  1 17:42:13 MBP kernel[0]: en1: 802.11d country code set to 'AU'.
Sep  1 17:42:13 MBP kernel[0]: en1: Supported channels 1 2 3 4 5 6 7 8 9 10 11
12 13 36 40 44 48 52 56 60 64 149 153 157 161 165
Sep  1 17:46:13 MBP kernel[0]: Auth result for: 00:26:b0:fe:76:74 MAC AUTH
succeeded
Sep  1 17:46:13 MBP kernel[0]: AirPort: Link Up on en1
...
Aug  5 09:49:07 MBP kernel[0]: en1: 802.11d country code set to 'X0'.
Aug  5 09:49:07 MBP kernel[0]: en1: Supported channels 1 2 3 4 5 6 7 8 9 10 11
36 40 44 48 52 56 60 64 100 104 108 112 116 120 124 128 132 136 140 149 153
157 161 165
Aug  5 09:49:10 MBP kernel[0]: NVEthernet::setLinkStatus - Valid but not
Active
Aug  5 09:49:10 MBP kernel[0]: NVEthernet::mediaChanged - Link is down
Aug  5 09:49:10 MBP kernel[0]: NVEthernet::setLinkStatus - Valid but not
Active
Aug  5 09:49:13 MBP kernel[0]: en1: 802.11d country code set to 'US'.
Aug  5 09:49:13 MBP kernel[0]: en1: Supported channels 1 2 3 4 5 6 7 8 9 10 11
36 40 44 48 52 56 60 64 100 104 108 112 116 120 124 128 132 136 140 149 153
157 161 165
Aug  5 09:49:40 MBP kernel[0]: Auth result for: 00:0c:e5:0e:65:bd MAC AUTH
succeeded
Aug  5 09:49:40 MBP kernel[0]: AirPort: Link Up on en1
...
Jun  5 12:08:49 MBP kernel[0]: en1: 802.11d country code set to 'SE'.
Jun  5 12:08:49 MBP kernel[0]: en1: Supported channels 1 2 3 4 5 6 7 8 9 10 11
12 13 36 40 44 48 52 56 60 64 100 104 108 112 116 120 124 128 132 136 140
Jun  5 12:09:14 MBP kernel[0]: Auth result for: 88:f0:77:2f:75:70 MAC AUTH
succeeded
Jun  5 12:09:14 MBP kernel[0]: AirPort: Link Up on en1
```

## Correlate With...

---

Photo  
EXIF Data

Calendar

E-mail  
Itineraries

Internet  
History

Travel  
Websites

Search  
History

The travel data found in the logs can be correlated with many other forensic artifacts.



## Exercise 3.3 – Timeline Analysis & Data Correlation

© SANS,  
All Rights Reserved

Mac Forensic Analysis

This page intentionally left blank.

# Agenda

---

Part 1 – System Information

Part 2 – System Applications

Part 3 – System Preferences

Part 4 – Log Analysis

Part 5 – Timeline Analysis & Data Correlation

© SANS,  
All Rights Reserved

Mac Forensic Analysis

This page intentionally left blank.



## FOR518

# Mac Forensic Analysis

The **SANS** Institute



Sarah Edwards  
oompa@csh.rit.edu  
@iamevl twin



@sansforensics

<http://computer-forensics.sans.org>

© SANS.  
All Rights Reserved

Mac Forensic Analysis

Author: Sarah Edwards  
oompa@csh.rit.edu  
<http://twitter.com/iamevl twin>  
<http://twitter.com/sansforensics>