Basic Mac Forensics

Nick Peelman

Purdue University

Abstract

Computer Forensics is an area that is very Windows-centric. Many tools pay lip service to Apple's Macintosh (Mac) platform, and others do not even recognize it at all. The few Mac tools available are either expensive or inadequate. Regardless, it is necessary for an investigator to know what to look for and where to look. This paper is intended to give investigators a brief outline of what the file system and structure of a Mac looks like and to give a basic criteria on what to look for, as well as some generalized locations for where to look. It is far from a comprehensive forensic manual for Macintosh computers, but it does attempt to give an examiner relatively comfortable with Windows environments a place to start learning about Mac forensics.

Introduction

Whether or not the PC crowd is ready, Apple is clawing its way back up the ladder in market share. The success of the iPod has made Apple a household name once again, and with the frustrations of Windows Vista now more than ever people are flocking towards the inviting and seemingly more stable environment Apple's Mac OS X offers. This presents a significant hurdle for forensic investigators because many are not familiar with Mac OS X or UNIX systems in general. The law of numbers indicates that as Macintosh systems re-saturate the personal computer market space, their use in illicit or illegal activities will increase as well, necessitating the need for investigators to become cross-platform in their abilities. This platform agnosticism will not come easy, and it will be difficult for investigators to be "experts" at both systems, simply because the rapid evolution of both platforms and the amount of time it takes to become fully comfortable with both sides.

With this in mind, this paper was written to give investigators already familiar with at least basic Windows forensics a "Quick Start" guide to the Mac platform and its internals. It begins with a logical look at the file structure, and then moves into a summary of the more common places investigators need to examine. As is resonated throughout this document, it is not meant as a comprehensive guide, since the scope of such a guide would be outside the space constraints of this paper.

The Mac OS X File Structure

Many forensic investigators are familiar with the hierarchical file structure in Windows which typically begins with C:\ or some other drive letter. The Mac follows a similar hierarchical structure, but with a different approach. This section addresses the "trees" of the Mac OS X file structure, the purpose of each, and how it relates to a Windows structure.

The Unix Root

The fact that OS X is a UNIX system is important to investigators because many standard UNIX conventions are followed at a level most Mac users never venture to, and can be leveraged to an investigator's advantage. Access at the UNIX level is often more rewarding forensically than access from most graphical tools. UNIX compliance also means that many of the open source tools UNIX systems enjoy are available for forensic investigators working with Mac hardware and software. At the same time, such a basis presents many more challenges, especially given the lack of experience many investigators have with Mac hardware and software. As an example, many system files are marked Invisible in the Finder by a very neat metadata trick Apple built into their system so that normal Finder users never see them. A moderately technical user can set other files to be invisible very easily, using the SetFile utility Apple provides to developers. This process hides files or directories from the prying eyes of a casual user, but not from a typical UNIX user. Anybody with reason enough to hide something would be able to find instructions on how to do this easily through the internet. Figure 1 shows what a typical user would see in Finder compared with what is truly there at a the UNIX level.

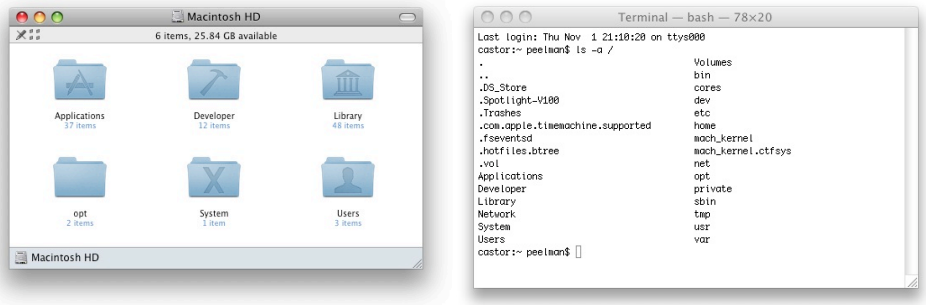

Figure 1: *A comparison of what Finder users see and what is really there at a UNIX level.*

Notice in Figure 1, we only see 6 folders when using the Finder, with no files visible at the root level. However, when we look at the system at the UNIX level using Terminal, we see 28 files and folders. Something that is typically irrelevant but may bear weight in certain cases is that /etc/, /tmp/, and /var/ all actually reside in /private/, but are symbolically linked to their standard UNIX location at the root of the file system.

This UNIX tree is very important and can reveal many key aspects of the system, however examining OS X purely from a UNIX Standpoint would be the topic of a separate, probably larger, paper, since any user who can truly manipulate an OS X system as this level will probably present more challenges to investigators than a normal user. With that in mind, there are a few key points to know about the Mac's file system. Beneath the fancy hardware and flash interface, Macs use a traditional UNIX file system, with / being the root directory containing all other directories and files. Any external file systems (USB/ Firewire hard drives, Flash memory devices, digital cameras, anything that can be seen as a storage device) are mounted under /Volumes/ with a psuedo-unique name for each device (two devices can have the same name, for example, you can have two hard drives named "Macintosh HD". When the system encounters this situation, it would simply append a "1" to one of the drives when mounting). You can think of /Volumes as the My Computer of the OS X file system. Where on Windows you would see a C:\, D:\, and E:\ drives in My Computer, on a Mac if you navigate to /Volumes/ in the Finder you may see MyUSBKey, WDBook, etc. Logically from an access standpoint, the UNIX root is divided into several trees, each tree representing a domain of user access. Each of the four domains is addressed and detailed below.

The System Tree

At the UNIX root of the OS X file tree is the System Folder. This folder is owned by the system and is commonly not writable by the user without providing administrative credentials, meaning that modifications to this part of the system are typically deliberate and often worth investigating. The files

and folders in this tree are restricted to the core files of the operating system. Anything that is necessary for the basic functionality of booting and presenting a basic GUI should be in this folder, including but not limited to: Applications such as Finder and Dock, the default user profile for creating new users, Preference Panes, Frameworks, mostly things that should **never** be touched by a user manually. Typically everything in this folder only changes during software updates such as security packs or OS updates.

The Local Tree

The Local Tree represents files and folders that are not entirely necessary for the OS to function, but are available to all users of the machine and are generally important in day to day use of the computer. This tree includes the /Applications/ and /Library/ folders. If it is installed on the system, the /Developer/ folder would be included in this tree as well. Files and Folders in this tree are modifiable only by a user in an administrative group and those users must authenticate when doing anything that changes these files. Items contained here include not just applications, but support files, and preferences that are available for use by every user on the system. Unlike the System tree, this tree may in fact be modified quite often by a user, either when installing software, setting a global preference, or creating folders or files in a public space for Documents or other media that needs to be shared with the other users of the system. In the cases of installing applications or setting preferences, the user may not even realize the changes are being made at this level, even if they are asked for their user name and password when doing it.

The User Tree

The User Tree is generally rooted in a user's profile and is referred to synonymously with the "home" folder. It is sometimes referenced within the file system as a tilde ("~"), so ~/Documents/ refers to the documents folder of the user in question. For example, if joe is the short name of the current user, then

~/joe/ would actually point to /Users/joe/. It is merely a shortcut, and is used here within as short hand to describe locations based on a user's home directory.

Since OS X was built from the ground up as a multi-user UNIX system, each account a user has on a system can include its own applications, settings, documents, media, and temporary files; basically every user lives in their own little world and rarely needs to write to disk space outside of their profile, but each user still has read and execute access to the applications and capabilities of almost everything in the Local and System trees. On this same note, a user's home directory is very portable and easily moved between systems. It is commonplace in the Mac universe for a user to move their entire profile when upgrading computers, and Apple actually provides a Migration Assistant that does just that (among other tasks such as migrating the local user database, etc). This may become relevant if references are found in a user's home directory to items (applications, documents, etc) that do not exist on the machine being analyzed. A suspect may have moved their home directory to an otherwise clean system and somehow disposed of the dirty machine. In some contexts it may be more important for a suspect to have a computer that appears clean rather than have no computer at all, or a hard drive that has recently been wiped and now contains a fresh OS install.

By default, each home directory on an OS X system does not include an Applications directory, but creation of one is recognized by the OS and it given the same icon the more prominent /Applications/ directory receives. Even on a tightly managed OS X system, users with only standard user accounts and no administrative privileges can download, install, and execute many types of software.

The User's Tree is owned by that current user, and that user has full control over everything in it. Other normal users typically cannot even see the contents of a user's home folder. Administrative users can, but typically only from the Terminal, and only after assuming a root shell, or issuing commands using 'sudo', and both of these methods require authentication.

The Network Tree

There is a fourth, lesser known and less often used tree is associated with a networked, highly organized, highly structured environment such as an office or a school. This Network Tree is more scattered than any of the other branches, but most of the applicable files will reside in /Library/Managed Preferences/. The folder /Network has nothing to do with this tree, it is simply a folder used to logically and dynamically manage network file shares and connectivity. The purpose of the Network tree is similar to an Windows Active Directory environment, where certain machine and user preferences are managed from a centralized location for various reasons. When seizing computers that are part of a larger network, it is a good idea to at least be aware of these settings and verify them with the systems or network administrator.

What is with all these Trees?

Each tree represents a logical domain of access to the system. In this way a system can be managed, yet still offer the user(s) quite a bit of freedom. For example, applications that follow Apple's development guidelines may search /Library/Preferences/ for a relevant set of preference files before looking inside the User's own Preferences folder (~/Library/Preferences/) for a similar set. This allows a set of "local" or "machine" preferences that apply to every user of the machine, but still allows the user to keep preferences for his or her self. It should be noted that not all Applications follow those guidelines. Preference Panes are a perfect illustration of this hierarchy and how it is beneficial. To manage system preferences, Apple wrote an application called "System Preferences" (They are so creative!) You can think of System Preferences as the Windows Control Panel, but done right. System Preferences uses small bundles (specially structured folders) called Preference Panes to divide up its tasks. Many key preferences panes that are used to modify and manage important system settings (display resolutions, network settings, etc) are necessary for the system, so they go in the System Tree (/System/Library/Preference Panes/). Later on, a user wants to add some functionality to their system, such as the Growl notification system. If they want the functionality to be available to every user on the system, they

would add it to the Local Tree (specifically in /Library/Preference Panes/, notice the pattern yet?). However, if they only want to add the functionality for themselves, and not share with everybody else, they can simply add it to the User Tree (~/Library/Preference Panes/, note the tilde). System Preferences, being intelligent and properly coded, looks in all three domains for preference panes to load when it starts up.

Comparing Windows and Mac File systems

The System tree (/System, /bin, /usr, /private) is loosely analogous to the C:\Windows\ or C:\WINNT\ directory in a Windows environment. The Local Tree (/Applications, /Library, /Developer) is similarly related to the C:\Program Files\ directory. Typically in a Windows environment, C:\Program Files\ contains both Applications themselves and any associated support files (templates, example files, plugins, etc). In the Mac world, Applications are contained in the /Applications/ directory, and any support files necessary would be contained in the /Library folder, generally under /Library/Application Support/. Similar to the preferences example in the previous paragraph, many applications can look to both /Library/ and the current User's Library Folder (~/Library/) for support files. The User tree (/Users/) is similar to the C:\Documents and Settings\ folder, with a separate home folder for each user on the system.

Where to look

Forensic analysis of a Macintosh system is conceptually no different than on a Windows system. The same principles of isolation, acquisition, imaging, analysis, and reporting apply, but the procedures and context differ for the imaging and analysis stages. In some cases these differences are severe, in others they are subtle. This section is meant to give a brief primer into where to look during an analysis to find the common files that may be relevant to an investigation. It is not a comprehensive guide, and therefore should be used as a learning tool, but not an investigative check list.

Browser Information (history, cache, settings)

Web browsing is a daily past time now, and like every other task, people always have a favorite tool. On the Mac side, there are many, many web browsers. From Apple's own Safari, to the ubiquitous Firefox, to the less-known-but-still popular Camino, OmniWeb, and Opera. Sticking with a "general user" theory, Safari and Firefox are the only two outlined below. Both programs download files to the Desktop (~/Desktop/) by default but can be configured to place them anywhere. The preference files for both programs are located in ~/Library/.

Safari

Apple's native Safari browser is rapidly growing in market share as more and more users turn to Macs. It uses WebKit, an open source framework Apple develops and uses throughout its operating system. Safari spreads its forensically-useful files out over a few directories. Bookmarks, stored form values, browser history, information about the last session, and a icons file, are all contained in ~/Library/Safari/. Cached internet files are stored in a series of folders in ~/Library/Caches/Safari/ and/or ~/Library/Caches/com.apple.safari/ depending on the version of OS X and Safari being used.

With Safari 3 (the latest version that ships with OS X Leopard (10.5) and is available as an update for OS X Tiger (10.4)) Apple added a Private Browsing feature that reduces the amount of cached files to almost zero. There is also a Reset Safari feature that can blow away cache files, history files, etc. Both of these

features create an issue for forensic recovery, but there is no way around it known at this time, so not much more information can be detailed.

Firefox

Many Windows investigators may already be familiar with Firefox profile system, and for all intents and purposes it is the same on Macs. The core files of a users profile, including bookmarks, history, and any installed themes or extensions can be found in ~/Library/Application Support/Firefox/. Cache files for a user can be found in ~/Library/Caches/Firefox/. Each of these will contain a Profiles directory, each Profiles directory will contain one or more folders (profiles) with a unique folder name. Each folder name in the Caches directory will match up to one in the Application Support directory, so if you have multiple profiles you can use this to match the cache files with the proper profile.

Email

Email in general is forensic nightmare, with its lip service to security, lack of a trustable delivery trail, its wide open format that practically screams “edit me!”, and the large variety of clients available. While Macs don’t make email investigations much easier, they do not make it harder either. Macs users will often use one of three email clients: web-based, Apple Mail, Microsoft Entourage. Please note that like web browsers, many other email clients exist for Macs, but that for the vast majority, one of these three will apply.

Web-based Email

The first and most irrelevant to this section is web-based email. You may be able to pull passwords from the Keychain (discussed later), get the addresses of what services were used from the browser history or the Keychain, and maybe recover any attachments that were download, but there is little to no way to get the actual messages read or sent, from the local machine. The service provider will need to be contacted and worked with to acquire the relevant data.

Apple Mail

Apple's native email client is true to to the mantra of OS X, full-featured, flexible, powerful, yet remaining painfully simple. This is even true from a forensic perspective. All of a user's messages are stored inside ~/Library/Mail/. There will be a sub directory for every account a user has setup inside the client, as well as a Mailboxes folder for any local mailboxes a user has created that are not associated with any one particular account. An example using an IMAP account would be: ~/Library/Mail/IMAP-username@mydomain.com/. On newer systems (10.4 and greater), each message is stored as plain text with full headers in its own file with an extension of "emlx" inside of a folder with a .mbox extension. Older systems stored everything in single text file using a standard UNIX convention called "mbox", which is a consolidated mailbox format. This file was generally a plain text document with all the messages (with full headers) in a mailbox appended together, and was actually forensically easier to analyze, especially when compared to a similar situation such as Microsoft Outlook's PST format. The newer method has its own set of "pros" but makes forensic analysis of a user's email cache more difficult.

Also stored in ~/Library/Mail/ are several important plist (Property List) files containing information about Smart Mailboxes, message rules, and of particular forensic importance: opened attachments. On the topic of attachments, IMAP and Exchange accounts store attachments within the account folder in ~/Library/Mail/. These types of accounts loosely tie the messages to the attachments by placing each attachment in a subfolder named after the message ID that it goes with. POP based email accounts generally store attachments in ~/Library/Mail Downloads/. The Mail.app client handles keeping track of which attachments go with which message for a POP account.

Microsoft Entourage 2004

Microsoft Entourage is generally referred to by the Mac community as Outlook for Macs. It has the same basic feature set as Outlook, though with a very different interface, and is the Mac's only way to fully interface with Exchange environments. That last fact is relevant because Macs that exist inside many

corporate environments that rely on Exchange will most likely have Entourage on them. Entourage stores all of its data inside the user's Documents folder (~/Documents/Microsoft User Data/). The contents of this folder can vary depending on how much a user utilizes the Office suite, but there will be two folders that should be examined in depth:

~/Documents/Microsoft User Data/Office 2004 Identities/

~/Documents/Microsoft User Data/Saved Attachments/

The relevance of Saved Attachments should be pretty obvious. The Office 2004 Identities folder will contain subfolders for each identity a user has defined, but in most cases there will only be two things in this folder, a folder called "Main Identity" and a file called Newsgroups Cache. Inside a user's Main Identity folder will be a Database file (no really, its called "Database" with no extension, great job Microsoft!) which contains all of the user's messages in a compressed or otherwise scrambled manner. Other files in here include Rules, which is used to store defined message-handling rules, Mailing Lists, which is used to manage Mailing List rules (these differ from normal Mail rules), and Signatures, which keeps a listing of the user's signatures for signing messages. All three of these files are compressed or otherwise scrambled similarly to the Database file. This entire directory is theoretically portable and could potentially be read (through a write blocker) by an examination machine if placed or linked to the correct location on the examiner's Mac.

It should be noted that this covers Entourage 2004. Office/Entourage 2008 is at least a month away from release and with no public betas available, no reliable data is available on what changes have been made to the information storing model for the new revision. This data will most likely not be relevant for a machine with this new software.

Keychains

Apple designed an ingenious method of storing small amounts of data that need to be secured in partly encrypted files called Keychains. These files store user's passwords, certificates, and any secure notes

(plain text only), in a partially encrypted and secure file inside the ~/Library/Keychains/ folder. The keychain is typically named login.keychain and can be moved from system to system, allowing a user to have the same saved credentials available to them on multiple systems. Forensically this little file can often yield a lot of data about a user's habits, since many users, except those with the most security-conscious agendas, make use of the convenience of the Keychain's abilities. if the user chooses convenience over security, keychains can store credentials for several key areas:

- Airport (802.11 wireless networks)
- Instant Message Account Passwords
- VPN Passwords
- Encrypted disk image passwords
- Application passwords for anything that may require authentication
- Website passwords (may include webmail passwords)
- Secure Notes
- Digital Certificates for secure websites
- Digital Certificates for Email SIgning & verification

Obviously this is one file that is very important for investigators to find and analyze. The one caveat to this is that the file is partially encrypted and can only be accessed with the user's login password, or if the user is particularly security conscious, they will use separate passwords for their login and their keychain. This makes discovering the actual passwords difficult without suspect cooperation, however simply viewing the file in plain text will show that not *all* of the data is encrypted, only the passwords themselves. Information such as web addresses, email addresses, services, and other items may be plainly visible. One final caveat is that users are not limited to just one keychain file. They can create as many as they want, each with different data stored in it, and each can have its own password.

System Logs

The system logs on a Macintosh system may provide insight into a user's actions and the timestamps may help present a stable timeline of events. Apple created several locations for log files:

/var/log/

/Library/Logs/

~/Library/Logs/

The relevance of these logs depends on the context of the case, so suffice it to say that some may be more important than others, and some may be completely irrelevant. The most commonly useful ones are outlined below.

System.log

System.log is the catch all log for the system. Many things get logged here, but the content of this log varies system to system, and even minute to minute, depending on the context in which the machine is used. The system will typically compress and achieve these logs on a regular basis, so in addition to /var/log/system.log, there should be several different versions of this log with a number and .bz2 appended to them. This will be true for many of the logs the system keeps.

Secure.log

Secure keeps track of any changes to secure system areas. That means that any time a user has to authenticate to modify a normally off-limits part of the file system, it is logged here. Any time a user modifies a setting in System Preferences that affects more than just their user preferences, it gets logged here. Any time a terminal-savvy user authenticates to a higher level account to do something, it gets logged here. This includes using the 'sudo' command, and in that case the entire command is logged here. If it is not obvious by now, this file may be very relevant if the goal is to prove that somebody modified or deleted data, or otherwise issued commands that required authentication from the system.

Daily.out

Daily.out is the log of the daily event that all Macintosh machines execute by default, once a day, to do housekeeping tasks such as rotate logs and purge temporary files. The reason it is important is that it presents two key pieces of data that are forensically important, it records the results of an "uptime" command, reporting how long the system has been up, and it reports all mounted disks, network, local or otherwise. This provides a very convenient place for investigators to look to see what disks were mounted recently. The only bad part is that this script only runs once every 24 hours (the precise time of

execution daily varies from machine to machine, and can be determined by view the time stamps inside the log), so the disks will only be recorded if they are plugged in at the right time of the day.

ppp.log

This log file provides a time stamped connection history for any VPN connections made from the system using the built in VPN software. Third party VPN software may use a different logging mechanism, see the program's documentation for ideas on where to look.

Conclusion (One more thing…)

Macs really are a world apart from their Windows-running counterparts, but I hope I have outlined the similarities in a way that takes some of the fear out of having to look at one forensically. As I said in the introduction, many of the same forensic principles, SOPs, logic, and instinct can be carried over from Windows forensics to the Mac side, and even the approach taken can be very similar, its just the format of the data being analyzed will be slightly different.

Unfortunately, many things were cut from this paper for the sake of space, since this was originally intended to cover a broader scope. Some of the things left out includes chat/Instant Message logging, encrypted disk images including FileVault, secure virtual memory and its impact, locations of applications, support files, preferences, etc. The prominence of XML configuration files for almost every facet of the system and which ones may hold the key to a mystery. On the investigative side, there are disk and partition operations in Disk Utility, some tips for live analysis of a system, the new rules that OS X 10.5 Leopard makes with features like Time Machine, a walkthrough of the Mac boot process, and a basic guide to using Terminal.app, which would have evolved into an introduction to using the UNIX side of the operating system from the command line. There is a lot of data available on any of these topics, but not much of it has been digested into a forensically useful format. Also, as was noted earlier, the platform is evolving rapidly, with a new major release from Apple every 18-24 months. This makes getting and staying current on Macs that much more difficult.

Bibliography

API Reference: Mac OS X Manual Pages. (n.d.). Retrieved Dec. 5, 2007, from http://developer.apple.com/documentation/Darwin/Reference/ManPages/index.html.

Craiger, P., Burke, P., Olivier, M., & Shenoi, S. (2006). Advances in Digital Forensics II (IFIP International Federation for Information Processing). New York: Springer.

Donnelly, D. (n.d.). Mac Forensics - BlackBag Technologies. Retrieved Dec. 5, 2007, from http://www.macos.utah.edu/documentation/security/forensics.html.

File System Overview. (2006, June 28). Retrieved Dec. 5, 2007, from http://developer.apple.com/documentation/MacOSX/Conceptual/BPFileSystem/index.html.

Heiser, J., & Ii, W. (2001). Computer Forensics : Incident Response Essentials. New York: Addison-wesley Professional.

Hillegass, A. (2004). Cocoa(R) Programming for Mac(R) OS X (2nd Edition). New York: Addison-wesley Professional.

Keychain Services Programming Guide. (2007, January 8). Retrieved Dec. 5, 2007, from http://developer.apple.com/documentation/Security/Conceptual/keychainServConcepts/index.html.

Knaster, S. (2005). Hacking Mac OS X Tiger : Serious Hacks, Mods and Customizations. New York, NY: Wiley.

Lamb, D., Miquelon-Weismann, M., Moreau, D., & Orton, I. (2006). Cybercrime: The Investigation, Prosecution and Defense of a Computer-Related Crime. Durham: Carolina Academic Press.

Mccormack, D., & Trent, M. (2005). Beginning Mac OS X Programming. Indianapolis: Wrox.

Open Source - Internet & Web - WebKit. (n.d.). Retrieved Dec. 5, 2007, from http://developer.apple.com/opensource/internet/webkit.html.

Stauffer, T. (2001). Mastering Mac OS X. New York: Sybex Inc.

Technical Note TN1150: HFS Plus Volume Format. (2004, March 5). Retrieved Dec. 5, 2007, from http://developer.apple.com/technotes/tn/tn1150.html.

Technical Note TN2166: Secrets of the GPT. (2007, Nov. 6). Retrieved Dec. 5, 2007, from http://developer.apple.com/technotes/tn2006/tn2166.html.

The WebKit Open Source Project. (n.d.). Retrieved Dec. 5, 2007, from http://webkit.org/.

Welcome to Growl!. (2007, Nov. 3). Retrieved Dec. 5, 2007, from http://www.growl.info.