# iTunes Forensic Analysis: Part 2

Wed, 09/21/2011 - 10:44am

Paul B. Ciaccio

Read Part 1 of this article.

## Evidence/Artifacts (Locating fruits of a crime)

Capturing screen shots of iTunes Preferences is much like obtaining settings from LimeWire, they provide the necessary evidence to show which settings were modified from the default configuration. Such screen shots make great exhibits to present in a lab report. However, it may be difficult at times to collect screen shots if an examiner cannot boot the evidential system in a virtual environment, e.g. VMware or Virtual Box. In unforeseen circumstances, the forensic examiner must depend on looking in other areas of a computer system in hopes of obtaining any type of evidence or artifacts.

The registry will provide very little data from the iTunes application: the installed version and the date of installation. However, the date of installation is not necessarily when iTunes was first installed on the system, but the date of when the latest available version/upgrade was installed.

The iTunes version and installation date can be obtained from the logical registry paths noted in Table 1.

Table 1: iTunes version and installation date.

Most systems contain a large number of 16-byte GUIDs listed under the Installer or Uninstall registry keys. Therefore, it would be advantageous for the examiner to search for the word “iTunes” rather than perform the tedious task of selecting each 16-byte GUID one at a time.

Since iTunes is an Apple product, Property List (plist) files exist on a file system, and can be found under the Program Files directory and under the user's profile. However, the content in each plist file is generic and renders no data with regard to iTunes Preference settings nor provide information of evidentiary value.

Another alternative is to perform a search for key terms, particularly in the pagefile.sys and hiberfil.sys files which could provide evidence that the suspect had searched for child pornography on the peer network using a P2P application. But recall in the child porn case, if evidence media are not seized in a timely manner, any valuable data will more than likely be overwritten in the two system files.

Apple's Tech Support Web site references iTunes Library files, which is a database iTunes uses to organize files added to the iTunes Library. Two iTunes Library files are created and maintained by iTunes for different purposes: iTunes Library.itl and iTunes Library.xml.4

iTunes Library.itl

The iTunes Library.itl file is a database of files in the iTunes Library and in the playlists that are created by the user. Some file-specific data is saved in this file. If the file is deleted, iTunes creates a new empty copy when it is opened in the application, but any playlists, song ratings, comments, or other information that is created is lost. The iTunes Library file is only used by iTunes.

The Library filename for versions of iTunes prior to 4.9 was iTunes 4 Music Library for Mac OS X and iTunes 4 Music Library.itl for Windows. After upgrading to iTunes 4.9 or later, the older Library files are moved to a “Previous iTunes Libraries” folder.4

iTunes Library.xml

This file contains most of the information stored in the iTunes Library.itl file. The purpose of the iTunes Library.xml file is to make the iTunes Library and playlists available to other Apple type applications on the computer. Mac OS X and other iLife applications (iPhoto, iDVD, and iMovie) use this file to make it easier to add music from the iTunes Library to any projects.

The iTunes Library files can be found under the user's profile in the sub-directories noted in Table 2.

Table 2: iTunes Library files are under user profile subdirectory.Currently, there isn’t a forensic tool that will parse the iTunes Library.itl file. However, using a hex editor, the version of iTunes can be found (readable) at file offset 17.

The iTunes Library.xml file on the other hand is text-based and can be examined with any XML or text editor. The iTunes Library.xml file contains valuable information to any media file that is added to the iTunes Library (music, video, audio books, etc.).

## iTunes Library.xml File Dissected

The iTunes Library.xml file will contain data pertaining to multimedia files in the user’s iTunes Library. Recall after the initial startup, items added to the library are the result of user action, i.e. adding files individually or at the folder level (drag-n-drop), purchasing videos from the iTunes store, dragging files from a Home Shared library, etc. When data pertaining to files are found within the iTunes Library.xml file, an examiner can report with certainty that the data is a result from the user physically adding files to his/her iTunes library.

Much like an HTML file, XML files contain tags that typically represent a known value. In the iTunes Library.xml file, key tags represent column headings on the iTunes program interface; the column heading is found in between the beginning tag and the ending key tag . For instance, the column heading for ‘Artist’ will look like Artist.

Figure 7: iTunes XML files after 2 videos are added.

Using the example with the 2 added videos displayed in Figure 4, Figure 7 provides an internal, yet partial, view of the iTunes Library.xml file for one of the videos. The content in Figure 7 was formatted to obtain an improved visual depiction of the XML file; blue-text refer to the XML tags that represent information pertaining to the iTunes application and the column headings to the Library, the red text are the data referring in the iTunes Library.itl file; multimedia files that are displayed in the iTunes Library and playlists.

Much of the information in the iTunes Library.xml file is self explanatory, and pertains to any file that is added to the iTunes Library (music, music videos, videos, audio books, etc.). However, a few XML tag items to note:

Size (69308406) the size of the media file is expressed in bytes [66.08 MB]. The size is rounded-up to the nearest tenth as it is displayed in iTunes (Figure 8).

Total Time (416156) the length of the media file expressed in milliseconds, approximately 6-minutes, 5-seconds, and rounded-up on the iTunes interface (Figure 8).

Persistent ID (53067A865DBC8FC7) is an identifier used for tracks and playlists that was first introduced in iTunes 6.0.2. The Persistent ID is used in both the iTunes Library.xml and the iTunes Library.itl files.1

Location (file://localhost/C:/Users/Paul/Desktop/Test/Ali%20Baba%20 %20-%20Bugs %20Bunny%20&%20Daffy%20Duck%20-%20bunny%20%5B English%5D.mpg); the file and path as it was added to the iTunes Library [C:/Users/Paul/Desktop/Test/Ali Baba - Bugs Bunny #38; Daffy Duck –bunny [ English].mpg]

Figure 8: iTunes XML tags.

If the media files were played from the iTunes program, the iTunes Library.xml file will incorporate additional information. When the video Ali Baba – Bus Bunny & Daffy Duck – bunny [English].mpg was played from the iTunes Library, it wasn't until after the video completed and iTunes advanced to the next video when the iTunes Library.xml file was updated with additional data (Figure 9).

Figure 9: iTunes Library.xml file

Three xml tags added…

Play Count (1) the number of times the video was played from the iTunes program via QuickTime.

Play Date (3381853912) UNIX time format; when the video was played in local date/time + 66 years. In other words, 3381853912 will convert to 1 Mar 2077 19:51:51. Subtracting 66 years from the calculated year will provide the correct local date/time. Note: Digital Detective's DCode tool will not properly convert the UNIX time. However, the Online Conversion tool: http://www.onlineconversion.com/unix_time.htm will convert the UNIX time, but the examiner would still have to subtract the 66 years.

Play Date UTC (2011-03-02T00:51:52Z) when the video was played in GMT.

The iTunes application will display the local system date/time (Figure 10).

Figure 10: The iTunes application will display the local system date/time.

It's important to note that any changes to the iTunes Library (deleting, adding, creating playlists, or playing any media files) dynamically updates the iTunes Library.xml file. Therefore, an analysis of the xml content will reflect the last activity in the user's iTunes program.

### Artwork cache files (iTunes' version of thumbnail pictures)

Many multimedia files might incorporate artwork such as an album cover to a music file, book cover to an audio book, or the first frame to a video. Depending on which button (Figure 11) is selected, iTunes will display the artwork on the interface, which also creates thumbnail size pictures that are embedded in a file that contains an itc2 file extension.

Figure 11: Depending on which button is selected, iTunes will display the artwork on the interfaceFigure 11a

The filename to the itc2 files consists of two 8-byte hexadecimal values separated by a hyphen, i.e. C87F7EFB700EB835-53067A865DBC8FC7.itc2. The first 8-bytes in the filename will match the name to its parent folder (two or more levels up) that resides directly below the iTunes Cache folder.

The itc2 files can be found under the user's profile in sub-directories noted in Table 3:

Table 3: The itc2 files can be found under the user's profile in sub-directories. If a user adds video files with child pornography and selects one of the interface option settings, an itc2 file would be created for each video file listed in the Library or playlist. In this circumstance, the itc2 files would be extremely valuable to the forensic examiner. However, unlike Thumbs.db or Thumbcache_xx.db files in Windows 2000/XP and Vista/Windows 7, respectively, the itc2 files do not render any dates or times. Figure 12 shows how the artwork of the two Looney Tunes videos is displayed when the Album List and Grid buttons are selected, respectively.

Figure 12 shows how the artwork of the two Looney Tunes videos is displayed when the Album List and Grid buttons are selected

The file signature to the itc2 file: x00 x01 x1C x69 x74 x63 x68 can be used to search and carve deleted itc2 files in slack or unallocated space.

A few Web sites on the Internet indicate the itc2 files are encrypted. However, an examination of the itc2 files in a hex viewer reveal embedded Portable Network Graphics (PNG) files. These PNG thumbnail picture files are the Artwork that is displayed on the iTunes interface when the Album List, Grid, or Cover Flow buttons are selected. Therefore, the itc2 files may contain one or more PNG files with different logical sizes for the same thumbnail picture (Artwork).

The PNG thumbnail pictures are easily identified by their file signature and footer (Figure 13), which can be exported and viewed with any compatible picture viewer, or within

Bookmarks using EnCase Forensics.

Figure 13: The PNG thumbnail pictures are easily identified by their file signature and footer.

Known Facts #2

System registry:

The iTunes application version and install/upgrade date can be found in the software registry file.

iTunes Music Library.xml:

The default location of the Music Folder

Which files the user added to the iTunes Library

When the files were added to the Library

The size of the file

The length in time of a video

How many files and folders were added to the Library

The location of where file resides on the file system

And other non-pertinent information

Artwork cache files (itc2):

Found under the user’s profile

Contain PNG thumbnail pictures to album covers, music files, audio books, and first frames to videos

PNG thumbnail pictures of child pornography indicate the user added explicit files to the iTunes Library

LimeWire – Interaction With iTunes

As previously mentioned, it would take user interaction to add media files to the iTunes Library. But as we look at the iTunes configuration settings in LimeWire 5.2.13, the program has the capability to share downloaded audio and video files from the peer-to-peer network to iTunes users on a local network. The screen shot in Figure 14 displays configuration settings of how LimeWire interacts with iTunes. The first and second options: “Add audio files I downloaded from LimeWire to iTunes” and “Share audio and video files in Public Shared list on my local network with iTunes” are enabled by default.

The screen shot in Figure 14 displays configuration settings of how LimeWire interacts with iTunes.

Unfortunately at the time of this writing, the second configuration setting in LimeWire was only tested on a local network with iTunes. Files cannot be downloaded from the P2P network as a result of an injunction against LimeWire, LLC, et al. The injunction prevents the distribution (download) of the LimeWire program and the search and download of files from the peer-to-peer network.7 A copy of the court order can be read from the LimeWire website: http://download.limewire.com/injunction/Injunction.pdf.

The LimeWire software program can still be acquired from third-party Web sites and be fully installed onto a computer system. If the computer is disconnected from the Internet, LimeWire will continue to launch successfully and appear to be fully functional, and can still function on a local network.

But when the system is connected to the Internet, LimeWire will display an Attention note (Figure 15), thus preventing users from accessing the peer-to-peer network.

Figure 15: When the system is connected to the Internet, LimeWire will display an Attention note.

When the “Share audio and video files in Public Shared list on my local network with iTune”s remains enabled, any files in LimeWire’s Public Shared list will display on the iTunes interface to other users on a local network. The host computer sharing files from the LimeWire Public Shared list does not need to have iTunes open concurrently. If fact, LimeWire will still share files on a local network even if iTunes never existed on the host system. This is because LimeWire utilizes the Digital Audio Access Protocol (DAAP), which allows multimedia files to be listed and streamed across a local network with other DAAP compatible applications.

LimeWire will positively identify a user on a local network by obtaining the user account name followed by “…’s LimeWire Tunes.” For example, if the user account name is Paul, other iTunes users will see “Paul’s LimeWire Tunes” in the side-bar of their iTunes program. Unlike the accessible iTunes Preference setting, users cannot change the default shared name in LimeWire, thus a user sharing files from their LimeWire Public Shared list can possibly be identified on a local network.

Capturing option settings in LimeWire when an evidence computer is viewed in a virtual environment is ideal. Again if this cannot be accomplished, the examiner can analyze the LimeWire program file limewire.props. The limewire.props file can be analyzed using most forensic tools in text-based view; the file can reveal data that would indicate if any of the sharing options in LimeWire are enabled, but particularly when downloaded files are shared with iTunes users. Figures 16–21 are screen captures from the limewire.props file with LimeWire 5.2.13 installed on a Windows XP operating system. However, data in the limewire.props file appear to remain persistent with LimeWire installed on Windows 7.

## Limewire.props File Dissected

When default sharing options in LimeWire remain unchanged, there are no values added in the limewire.props file. Figure 16 (truncated content) displays the typical values found in the limewire.props based on default settings, less any data that are affected when connected to the Internet, such as GPS location, IP addresses, etc.

Figure 16: displays the typical values found in the limewire.props based on default settings

Default Settings

Note the values that are added in the limewire.props file when sharing options are changed from default settings (Figures 17 through 21). When the sharing options are changed back to default settings, the values are removed from the limewire.props file.

When the “Add files I download from P2P Users to my Public Shared List” option is disabled from its default setting, two values are added (yellow highlight).

If the sharing option was enabled back to its default setting, the values in Figure 17 would not be set to true, but would be removed from the limewire.props file instead.

If the sharing option was enabled back to its default setting, the values in Figure 17 would be removed from the limewire.props file instead.

When the “Allow me to search for and share Programs with anyone” option is enabled from the default setting, one value is added.

Figure 18: limewire.props file

As previously noted, when the sharing option is disabled, back to its default setting, the value would be removed from the limewire.props file.

When the “Allow me to add Documents to my Public Shared list and share them with the world” option is enabled from the default setting, one value is added.

Figure 19

Again, the value would be removed if the sharing option was disabled.

As previously noted, LimeWire utilizes the Digital Audio Access Protocol (DAAP) for sharing files with iTunes users on a local network. The DAAP value can be identified in the limewire.props files when the “Share audio and video files in Public Shared list on my local network with iTunes” option is disabled from its default setting.

Figure 20: limewire.props file

When enabled, the value is removed.

It should be noted when the value exists within the .props file, user interaction with the setting preclude any files in LimeWire’s Public Shared List to be shared on a local network with iTunes user’s, or any other program that supports DAAP.

When the “Share audio and video files in Public Shared list on my local network with iTunes” option remains enabled, and the “Require password” option is enabled and a password is added in the password field, two DAAP values are added to the limewire.props file.

Figure 21: two DAAP values are added to the limewire.props file.

Note the DAAP_Enabled=false value in Figure 20 is removed. This is because the sharing option was switched back to its default setting (enabled).

FrostWire – Interaction With iTtunes

Of the numerous peer programs available on the Internet, FrostWire functionally behaves similarly to LimeWire with regard to downloading and sharing files from the P2P network (Gnutella based); however, FrostWire is also a BitTorrent application. FrostWire also incorporates the ability to utilize the Digital Audio Access Protocol, and therefore can share files to iTunes users on a local network.

As with its predecessor, FrostWire also obtains the user account name to identify the host that is sharing files on a local network, e.g. Paul’s FrostWire Tunes. However, FrostWire provides more flexibility in its naming convention than LimeWire by allowing users to access and change the shared name (Figure 22). The ability to change the shared name encompasses the same issue as the Shared Library feature in iTunes—the host cannot be positively identified on a local network.

Figure 22: FrostWire provides more flexibility in its naming convention than LimeWire by allowing users to access and change the shared name.

The iTunes Shared Name is reflected in the FrostWire.props file (Figure 22a).

Figure 22a: The iTunes Shared Name is reflected in the FrostWire.props file.

The iTunes Sharing feature is enabled by default. But when changed, the DAAP value is reflected in the FrostWire.props file (Figure 23a)

Figure 23: the DAAP value is reflected in the FrostWire.props file.

Figure 23a: the DAAP value is reflected in the FrostWire.props file.

To avoid redundant screen shots, the forensic examiner will see many similarities during the analysis of the .props file as well as other FrostWire program files when compared with LimeWire.

As noted, when a suspect’s system cannot be viewed in a virtual environment, it would be advantageous for a forensic examiner to obtain the original program version and take note of default settings, and then compare those settings with the suspect’s system. Any differences noted, particularly with DAAP value settings will indicate whether a suspect was sharing files across a local network with other users who have iTunes installed.

## Conclusion

The digital forensic community has learned that peer-to-peer programs such as LimeWire, FrostWire, and many other peer applications have the ability to share files across the Internet. But now the community is receiving more criminal cases involving iTunes or any other program that supports the Digital Audio Access Protocol where video files of suspected child pornography are shared (streamed) across a local network. Such cases are stirred up when iTunes users eyeball shared video files on their iTunes interface with filenames indicative of child pornography, and are hopefully clever enough to take screen shots of what they discover.

This article is shared with the forensic community in hopes of helping to tackle the criminal mind when inappropriate video files are shared on a local network as well as through the peer-to-peer network. Any recent discoveries are encouraged to be shared with our unique, yet highly skilled, community.

## References

AppleScript, (2011), Doug's AppleScripts for iTunes, Retrieved from http://www.dougscripts.com/itunes/itinfo/itunes602info.php.

The Apple Museum, (2010), Timeline of iPod and iTunes, Retrieved from http://www.theapplemuseum.com/index.php?id=43.

Developer; Open Source, (2011), Bonjour, Retrieved from http://developer.apple.com/opensource/.

iTunes Support, (Jan. 6, 2011), iTunes: What are the iTunes Library files?, Retrieved from http://support.apple.com/kb/HT1660.

iTunes Support, (Feb. 5, 2011), iTunes: Understanding Home Sharing, Retrieved from http://support.apple.com/kb/HT3819.

SourceForge.net, (2010), Digital Audio Access Protocol, Retrieved from http://daap.sourceforge.net/docs/index.html

United States District Court; Southern District of New York, (Oct. 26, 2010), Court Order, Retrieved from http://download.limewire.com/injunction/Injunction.pdf.

## Software Tested

iTunes versions: 9.0.1.8; 10.1.2.17; 10.2.1.1

LimeWire: 5.2.13

FrostWire: 4.21.3

Operating Systems Used For Testing

Windows XP, SP2, 32-bit

Windows 7, 64-bit

Paul B. Ciaccio, DFCP|CFCE|CDFE|CEECS|EnCE, works in Advanced Information Systems at General Dynamics and is contracted to the Defense Computer Forensics Laboratory in Linthicum, Maryland. He can be reached at paul.ciaccio@gd-ais.com or paul.ciaccio@dc3.mil.

Topics

Computer Forensics

Network Forensics