

available at www.sciencedirect.comjournal homepage: www.elsevier.com/locate/diinDigital
Investigation

Limewire examinations

Joseph Lewthwaite^{a,*}, Victoria Smith^b

^aDefense Cyber Crime Institute, Washington, DC, USA

^bDepartment of Defense, Computer Forensic Laboratory, Washington, DC, USA

A B S T R A C T

Keywords:

Limewire
P2P
Gnutella
Digital forensics
Java
Windows
AScan

In the world of information sharing Limewire is one of the more popular means for exchanging illicit material and therefore often features in child pornography (CP) cases. In this paper we look at evidence that examiners have available to them, the artifacts left behind by installation and use of the Limewire client that will tell them what the user did and their intent behind that use. We will also look at tips and techniques for finding and extracting evidence from unallocated space, slack space and other corners of the digital evidence. Lastly we introduce a tool AScan that will allow the investigator to extract all the evidence and expand the investigation into the child pornography networks the suspect was a member of.

© 2008 Digital Forensic Research Workshop. Published by Elsevier Ltd. All rights reserved.

Limewire is a peer-to-peer (P2P) application that is based around the Gnutella protocol. The Gnutella protocol is a communications protocol that allows a user to connect to a network with no centralized server, every node in a P2P network can talk to every other node. This allows the community to determine content with no supervision making it ideal for the trading of illicit material.

As a Java application, Limewire writes out the application settings as either text or XML and it writes out the database, log and cache files in accordance with the Java Object Serialization (JOS) specification. The JOS specification is a specification that allows objects within a Java application to write themselves out to disk in a standardized manner. In this paper we introduce the JOS to the investigator and a tool, AScan, which uses the JOS specification to retrieve the binary evidence in addition to the text evidence so the investigator will get a more complete picture of the evidence.

Limewire files of interest include: 'library.dat', 'createtimes.cache', 'version.xml', 'fileurns.cache', and 'Limewire.props'. Taken together the content of these files will give the investigator a picture of the users' library, what they downloaded, dates and times, SHA1 values, what the user shared and what they didn't.

The 'downloads.dat' is another Limewire file of interest that can provide search terms, SHA1 values and the paths of currently or recently downloaded files. Found in the 'incomplete' folder where Limewire tracks what the user was in the middle of downloading the 'downloads.dat' file is a snapshot of Limewires' outbound connections and is used by Limewire to reestablish a connection should it go down, either the user shuts down or Limewire crashes. As a backup in case of crashes it is written and deleted many times. Using the tools EnCase, Wireshark and Process Monitor we show that a search of slack/unallocated space can result in the retrieval of the suspects' search terms from previous 'downloads.dat' files with no contamination from external sources.

The last file of interest is the 'spam.dat' file. The 'spam.dat' is the database behind the users' spam filter. A spam filter rates results in an attempt to create a better search result set. The spam filter caches keyword terms and IP addresses and rates them according to the users' preferences. Terms and IP addresses that result in many downloads get high ratings. Pulling out these ratings will show the investigator the trends behind the users' searches and more importantly give the investigator IP addresses that were used to download files from. By matching the contents of the library to the individual

* Corresponding author.

IP addresses the investigator can see who is hosting the illicit material and penetrate the distribution network.

Limewire is one of the more popular methods for exchanging illicit imagery. In this paper we have shown that there is a lot of evidence in the Limewire artifacts that is not readily visible to the investigator. With the AScan tool and the proper knowledge the investigator can get a clearer picture of the evidence and the means to expand the investigation.

1. Overview

Limewire is a peer-to-peer file sharing client for the Java Platform (current version 4.16, 4.17 is under Beta), which uses the Gnutella network to locate and transfer files. Released under the GNU General Public License, Limewire is free software. While Limewire is open source software and it is free there is a company, Limewire LLC, that offers a pro version for a fee and is responsible for integrating outside programmers' changes.

Gnutella is a file sharing network based on peer-to-peer (P2P) technology. What marks a P2P network is that there are no central servers, in theory every node plays an equal part, in practice and in the Gnutella network you end up with a two-tiered system. Stronger nodes take the part of 'ultra-peers', caching results, executing searches and serving as connection points for leaf nodes. Leaf nodes are those nodes that haven't been connected long or don't have the bandwidth to handle multiple clients or searches.

The Gnutella protocol is concerned with the discovery of computers, connecting them and then searching shared libraries. Once a connection is established and a search has returned results the final transfer of the file is handled directly between the nodes using the HTTP protocol. Therefore a client wanting to establish itself with the network has only to support five messages:

- ping/pong – host discovery;
- query/query hit – search and responses;
- push – The node is across a firewall.

While Limewire was built around the Gnutella protocol, the latest versions are Bittorrent enabled. They also can connect and integrate to iTunes using Digital Audio Access Protocol (DAAP).

Limewire is an open source Java client that has featured in many child pornography cases. As an open source client it has spawned a number of knock-offs (some of them include: Acquisition, FrostWire and MP3Rocket) that use Limewire technology under the hood and therefore are open to examination using the techniques in this paper, though they haven't been tested.

2. Downloading/distribution

2.1. Download model

As a P2P piece of software the sharing of files is important to the building of the P2P community. Limewire shares

files by default and allows the user to add to the library, sharing whole directories or individual files. In older versions by default Limewire downloaded to the shared directory "<users>\documents and settings\shared".

In 4.16 Limewire separates the default download destination and shares. Instead of moving downloaded files into the 'shared' directory, 4.16 moves them to the "My Documents\Limewire\Saved" directory and then automatically shares the files in the 'library.dat' file.

2.2. Previews

When Limewire executes a download it downloads the file to a temporary file in the 'Incomplete' folder, once completely downloaded the file is copied to its destination. While it is downloading the temporary file name will be the same as the original prefixed with 'T-<size in bytes>'. Should the investigator find a file with the 'Preview-T-<size in bytes>' prefix it is an indicator that the user previewed the file. To play it while Limewire is downloading Limewire creates a copy of the first complete segment and puts the 'Preview-T-<size in bytes>' in front to avoid locking the temporary file while downloading.

2.3. Sharing

The 'library.dat' file is an important component of the users' library as it is the place where exceptions to the general directory sharing structure are made, either excluding or including files/directories.

In essence the new download model means the examiner has to be careful in interpreting entries in the library.dat file as explicit user shares since Limewire now makes entries in there for downloaded files if the file is going to a non-shared directory, which by default the 'My Documents\Limewire\Saved' directory is. On the other hand anything placed in the 'Shared' directory in a default install would have been explicitly placed there by the user. There are several variables that the investigator needs to look for in the 'Limewire.props' file to determine the users' configuration. For sharing downloads the default destination is specified by the variable:

• DIRECTORY_FOR_SAVING_FILES

The user can specify by file type where downloads go and override 'DIRECTORY_FOR_SAVING_FILES'. In this case the investigator might see the variables:

- DIRECTORY_FOR_SAVING_video_FILES
- DIRECTORY_FOR_SAVING_audio_FILES
- DIRECTORY_FOR_SAVING_image_FILES

If the user has turned off the option to automatically share downloaded files the variable 'SHARE_DOWNLOADED_FILES_IN_NON_SHARED_DIRECTORIES' will be set to false. If this variable is missing or set to true, Limewire will specifically share any downloads.

The users' shared library is defined in a couple of ways, the user can add whole directories, these will be found under the "DIRECTORIES_TO_SEARCH_FOR_FILES" variable in the 'Limewire.props' file. Limewire also allows the user to share individual files, these shares will be found in the 'library.dat' file under the "SPECIAL_FILES_TO_SHARE" category.

By default the users' client is setup to share, Limewire has upload connections available. For the user to turn off upload connections they have to explicitly set the variable 'HARD_MAX_UPLOADS' in the 'Limewire.props' file to 0. If the variable is missing or set to anything other than 0 then sharing is enabled.

3. Limewire structure

3.1. Java Object Serialization specification

The Java Object Serialization (JOS) specification is a fundamental part of Java and allows objects within a Java application to store and retrieve themselves. As Limewire is a Java application it is important to the investigator to have a basic understanding of the JOS to enable them to search for and interpret what they are looking for. With this knowledge they will be able to search for any Java files not just Limewire files.

JOS files have a header but no footer. The first two bytes of a JOS file are '0xAC 0xED'. The next two bytes will be the version, currently they will be '0x00 0x05'. Following this will be class and object definitions followed by data.

Java is an object orientated language. This means that the Java programmer breaks up an application into objects, referred to as classes, an object is a collection of information, variables, and actions, or functions that can act on the variables. A game about car races might have an object for a car, lets call the object 'o_Cars', that has variables that include a name, v_name, how many people can sit in it, variable 'v_seats', and what colour the car is, 'v_colour'. When Java stores the object it writes out the object definition followed by the variable values. If there was more than one car in the game the JOS would just reference the first object definition and just write out the variables associated with the second car and so on. So a JOS file about the game with three cars:

1. Minivan, seats 6, silver;
2. Sedan, seats 4, tan;
3. Roadster, seats 2, red;

might look like:

```
'com.mycargame.o_Cars (reference r1) ... v_name Minivan
v_seats 6 v_colour silver (r1) Sedan, 4, tan (R1) Roadster, 2, red.
```

Notice how the first record is intermingled among the definition. This all means that an examiner by searching for known class/object, names can find the first record of a particular class/object in storage. Common terms for finding Limewire files that have been deleted might include 'limegroup' and 'gnutella'. As Limewire stores and references files by the files SHA1 value looking for 'urn:sha1:' could also be productive.

While JOS files do not have a footer they quite often have a pattern of bytes at the end that can be a clue that the end of file has arrived. In the JOS specification data is stored in data blocks, the end of data block flag is 'x', 0x78. So the JOS when closing out the file will close out all embedded data blocks so the examiner might see a series of 'x' quite often followed by the bytes 'sq', 0x73 0x71.

3.2. Limewire installation

When Limewire installs onto a Windows XP machine it creates several directories. The main program directory goes by default in Windows under

- C:\Program Files\Limewire

For individual users' library and settings Limewire creates a series of directories. In all versions the individual users' directories reside under:

C:\Documents and Settings\<USER>\

The location for the settings and library for versions prior to 4.16 were:

- <USER>\limewire – Users' settings;
- <USER>\share – Files being shared by that user;
- <USER>\incomplete – Files that haven't completed downloading.

In 4.16 the default directory structure for individual users changed and looks like:

- \<USER>\Application Data\Limewire – Users' settings;
- \<USER>\My Documents\Limewire\Shared – Files being shared by that user;
- \<USER>\My Documents\Limewire\Saved – Files downloaded by that user;
- \<USER>\My Documents\Limewire\incomplete – Files that haven't completed downloading.

3.2.1. Registry

Even though Limewire is a Java application and uses settings files a couple of registry entries are made, mainly around file associations.

```
HKEY_Classes_Root\Limewire;
HKEY_Classes_Root\magnet;
HKEY_CurrentUser\Software\Classes\torrent;
HKEY_CurrentUser\Software\Classes\Limewire;
HKEY_CurrentUser\Software\Classes\magnet;
HKEY_CurrentUser\Software\Magnet;
HKEY_LocalMachine\SOFTWARE\Limewire;
HKEY_LocalMachine\SOFTWARE\Microsoft\Windows\Current
Version\Uninstall\Limewire (Version number can be found
here).
```

3.2.2. Files

Limewire installs a number of files under the users' 'Documents and Settings' folder. The files are separated into two directories: under the '<User>\Application Data\Limewire\' directory goes the settings files that determine what is shared, the users' library and the users' personnel settings, under the '<User>\My Documents\Limewire' folder goes the users' default library and the incomplete folder. In Table 1 we list the location of files and in Table 2 possible search. Then we describe each file and what the examiner can get out of them.

Table 2 gives the investigator a series of search terms that can be used to try and locate the Limewire files in unallocated or slack space. In general most of the Limewire JOS files will have the terms 'limegroup.' and 'gnutella.' somewhere inside them.

Table 1

File location	
Downloads.dat	
<=4.15	<user>\incomplete
4.16	My Doc...\Limewire\incomplete
4.17	<user>\Appli... Data\Limewire
Createtimes.cache	
Fileurns.cache	
Library.dat	
Limewire.props	
Spam.dat	
Version.xml	
<=4.15	<user>\limewire
4.16+	<user>\Appli... Data\Limewire

3.2.2.1. *Downloads.dat/.bak*. This file and its backup contain the information needed for Limewire to reestablish any connections for incomplete downloads. It is written periodically as Limewire is downloading, and when it exits with any downloads pending. This enables Limewire to resume downloading when it is restarted. The 'downloads.dat' file is written according to the JOS specification. Given that Limewire could be shutdown at any point in the download each connection may have the following information:

- IP address of server;
- Proxies;
- Host node type (Limewire, Bearshare...);
- File SHA1(base 32);
- Destination file path;
- Temporary Path;
- Search terms.

Table 2

Unallocated space search terms	
File	Term
Createtimes.cache	java.util.HashMap limegroup.gnutella.URN
Download.dat	limegroup.gnutella.downloader IncompleteManager
Fileurns.cache	com.limegroup.gnutella.UrnCache ttrout
Library.dat	java.util.HashMap SENSITIVE_DIRECTORIES SPECIAL_FILES_TO_SHARE
Limewire.props	(Any of the variables)
Spam.dat	gnutella.spam _bad _good _age XMLKeywordToken KeywordToken

3.2.2.2. *Fileurns.cache/.bak*. The 'fileurns.cache' file is saved according to the JOS specification. It is the cache of locally shared files identified by their SHA1:

Available information:

- File SHA1(base 32);
- File last modified time;
- File name.

3.2.2.3. *Createtimes.cache*. 'Createtimes.cache' is a file that contains a listing of files along with their associated system wide creation time. The system wide creation time is the time that the file hits the Limewire network. The 'createtimes.cache' file is saved according to the JOS specification.

Available information:

- File SHA1;
- File Limewire system wide creation time.

A note to interest if the times in this file match the 'fileurns.cache' then this user is introducing the files to the network which could be an indicator of content creation.

3.2.2.4. *Library.dat*. A '.dat' file that lists the directories and files that a user has specifically shared or excluded. When a file is downloaded by default it goes into the 'My Documents\Limewire\saved' directory and is shared through the 'library.dat'. A point of interest is that when a file is specifically shared the file entry is made in both the 'library.dat' file and the 'fileurns.cache' file, but when a user adds a shared directory the directory contents go into the 'fileurns.cache' file but no entry is made in the 'library.dat' file, the whole directory is shared by placing the directory entry in the 'Limewire.props' file (DIRECTORIES_TO_SEARCH_FOR_FILES variable). The 'library.dat' file is saved according to the JOS specification.

3.2.2.5. *Limewire.props*. The configuration properties file for the client install. This 'Limewire.props' file is a straight text file. Some of the interesting properties include:

- **HARD_MAX_UPLOADS** – Number of connections allowed for uploads. If present and set to 0 uploads are turned off, otherwise uploads are turned on. If it is missing Limewire by default sets the number of upload connections to 20.
- **CLIENT_ID** – Unique identifier for this client.
- **DIRECTORIES_TO_SEARCH_FOR_FILES** – list of directories that Limewire will search for files to share.
- **DIRECTORY_FOR_SAVING_FILES** – The directory Limewire uses to place files that have completed downloading.

3.2.2.6. *Version.xml*. 'Version.xml' is a plain text file with a portion of an xml document containing the installed version information.

3.2.2.7. *Spam.dat*. A '.dat' file that stores the current status of the Limewire spam filter. The spam filter rates keywords, search terms and IP addresses as to how the user perceives

them in an attempt to filter the users' results so that they only get back what they search for.

Available information:

- Keywords, spam rating;
- Download sources, IP and port.

Limewire does not distinguish between a keyword associated with a download and a search term, but the ratings will give the investigator the trends in the users' activities. Terms that have been searched for or heavily downloaded will receive high ratings. In addition the Spam.dat contains the IP addresses of download sources, this enables the investigator to browse the individual addresses for the users' contents and see who is distributing any particular file.

4. Keywords and intent

In this next section we look at trying to determine what the users' intent was behind their use of Limewire.

4.1. Background on searching in limewire

Computers that connect to the Gnutella network do so in one of two roles, as an ultrapeer or as a leaf. Ultrapeers handle all network traffic and are full participants in the Gnutella network. Leaves connect up to the ultrapeers and use the network through them. Leaves can connect up to three ultrapeers at a time. Ultrapeers can realistically handle approximately 27–30 leaves at a time. Leaves only connect to other leaves doing during a direct connection to share a file. There is no central server that decides whether a connecting computer will be a leaf or an ultrapeer, this decision is made by the individual Limewire clients based on the characteristics: time connected and bandwidth. An ultrapeer functions to shield the leaf nodes from the majority of messaging traffic and to handle most search requests. This is actually a very efficient manner in which to limit the traffic on the network and make it run faster. Instead of each leaf constantly sending search requests across the networks, which could be millions of computers at any one time, the ultrapeers handle the requests significantly dropping the amount of traffic in the network. More bandwidth means faster searches and downloads and happy users.

Limewire used the Query Routing Protocol (QRP) and Dynamic Query Routing to conduct searches. Each node stores a list of its shared files in a Query Routing Table (QRT). The QRT is created by taking all the keywords of the path, name and metadata of each file in the shared folder and hashing the results. The ultrapeer node builds a composite QRT from its own QRT and all its connected leaf nodes enabling the ultrapeer to respond to searches on behalf of all its leaves greatly reducing search traffic.

A user after receiving the search results initiates a download by selecting the file(s) and clicking on the download button. The location of the selected file(s) is present in the data received from the ultrapeer as the results of the search. The leaf wanting to download the file then connects directly to

the other node possessing the wanted file using the Hypertext Transfer Protocol (HTTP). The ultrapeer is not involved in the download process, but merely supplies the information necessary to facilitate the download between the leaves.

5. Testing

Limewire v4.16.3 was installed (with default settings) on a fresh installation of Windows XP Pro service pack 2 with all current updates as of February 2008, see Fig. 1. The test system had 512 MB of RAM installed. The following software was installed to monitor the system: Process monitor v1.26 and Wireshark v0.99.6a (Table 3). DD.exe was used to copy the contents of the memory and to produce images of the test system. EnCase v5.05a and 6.8.1 were used to examine the test system hard drive. The testing of Limewire was conducted at varying times over a six day period between 1 and 15 February 2008.

5.1. Testing

Fig. 1 shows the 'Search' tab of Limewire.

The maximum length of a search term in Limewire is 30 characters, as found in the Limewire Java programming code shown below:

```
MAX_QUERY_LENGTH = FACTORY.createIntSetting ("MAX_QUERY_LENGTH", 30);
```

Search types such as Boolean or Grep do not appear to work as input into the search process as test search for a Grep expression returns results related to the term and not for the expression.

A number of parameters can be set depending on the type of search being conducted. The type of parameters available are saved in XML schema files that are located in the \Documents and Settings\%username%\Application Data\Limewire\XML\ folder. It was noted during testing that using these parameters just added a term to the search string and did not appear to refine the search. For example, a search for the movie 'casablanca' was started by selecting the 'video' tab in the search window and typing the term

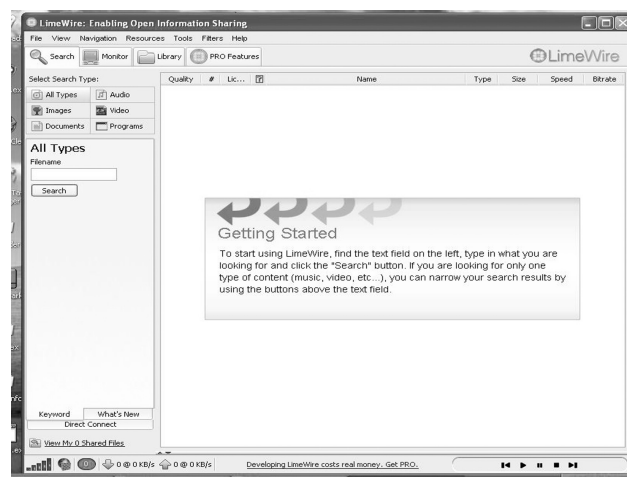


Fig. 1

Table 3

Test environment	Test software	Other software
Windows XP, service pack 2, base install of 32 bit OS 512 MB RAM	LimeWire 4.16.3 the free version with default settings	Wireshark 0.99.6a Process Monitor 1.26 DD.exe EnCase 5.05a and 6.8.1 to conduct keyword searches

'casablanca' in the input slot. The following parameters, 'commercial' and 'NP-17', were selected by using the drop down box and selecting these names. Upon initiation of the search it was noted that the search string now included 'casablanca+commercial+np-17'. The results did not appear to be refined by these parameters and only appeared to include additional search results that included the terms 'commercial' and 'np-17' producing a number of unwanted results.

The testing of Limewire involved a number of unique keyword searches. Wireshark was used to monitor and record the network traffic between the test system and the network. The logs were then analyzed to determine if the test search keywords could be found in outbound traffic and to determine if any data that would appear to be inbound searches of the test system were noted. DD.exe was used to perform a memory dump while Limewire was conducting a search to determine if any evidentiary data might be held in memory such as search terms. Several files were started as downloads were canceled or paused before completion. Upon completion of the downloaded files, the test system was imaged using DD.exe and analyzed using EnCase.

During the test it was noted that all files as they are being downloaded are first placed in the 'incomplete' folder and as the files complete the download they are moved to the 'saved' folder. Both of these folders by default are located in the users' '\My Documents\Limewire\' folder. A file named 'download.dat' and its backup 'download.bak' maintain the records of the files not completely downloaded to allow Limewire to resume the download when Limewire is restarted. This could be the result of either the user shutting down in the middle of downloading or a system or Limewire crash. The test included the pausing and canceling of several downloads. Fig. 2 shows an example of the data contained within the file 'downloads.dat' and of particular note is the data near the end of the file that follows after the term 'SearchInformation-Maps'. Here the search term was found that was used to conduct the search for the movie 'Casablanca'. The other file in the 'incomplete' folder, C:\Documents and Settings\LaCFG\MyDocuments\Limewire\Incomplete\T-346579019-Adobe Photoshop CS2 v9.0 Final + KeyGen&Activator = .zip was the results of a search for the term 'photoshop'. The download of the file was started and then cancelled before completion. The file name of this partial download is still maintained in the 'downloads.dat' but there is no search term saved that is associated with the file. Deleting the partial file from the

```
-i·sr·java.util.ArrayListx·Ö·Ca·I·siz
exp·w·sr·3com.limegroup.gnutella.down
loader.ManagedDownloader&z&uTtY·xr·4com.l
imegroup.gnutella.downloader.AbstractDownloa
derExE·HÄI·xpsr·java.util.HashSet°D·4·xpw
·?@·sr·com.limegroup.gnutella.RemoteFile
Desc[Y·\·%Dú·Z·browseHostEnabledZ·_chat
EnabledZ·_firewalledZ·_http11J·_indexI·_
portI·_qualityZ·_replyToMulticastI·_sizeI
·-
...
"·sq·~·{·\·T·sq·~·{·È·'·sq·~·{·(·
·sq·~·{·^·
H·sq·~·{·8·"·sq·~·{·î9°·°·sq·~·{·"·K·"·
·xxsq·~·?@·w·sq·~·w·)urn:sha1:HEPIOESII
H22LIBFHUI7N3AZH7WPXNRKq·~·xsq·~·wq·~·yw·\
xsq·~·
·w·)urn:sha1:WRTTPE6NKZ7RZRCG4T3BSWZQBWOKSP
K4q·~·xsq·~·wq·~·w·\xxxxsq·~·?@·w·t·
attributessq·~·?@·w·t·searchInformation
Mapsq·~·?@·w·t·mediasr·
com.limegroup.gnutella.MediaTypeE·+ia;]·Z
·isDefaultL·descriptionKeyq·~·L·extsq·~·
L·schemaq·~·xp·t·Videosr·java.util.TreeSe
tY~P·i#
[·xpsr·Corg.Limewire.collection.Comparator
s$CaseInsensitiveStringComparator·!f~s·Bî·
xpw·~·lt·asft·asxt·avit·cdgt·dcr·div
t·divxt·dvt·dvd·dvxt·flct·flit·flvt·
flxt·idxt·jvet·mlvt·m2pt·m2vt·mkvt·m
ngt·movt·mp2t·mp2vt·mp4t·mpet·mpegt·m
pgt·mpgvt·mpvt·mpv2t·nsvt·ogmt·qtt·ra
mt·rmt·rmmt·rmvbt·rvt·smit·smilt·srtt
·subt·swft·vcdt·vobt·vrmlt·wmlt·wmvxt
·videot·queryt·casablanca·xmt·<<?xml
version="1.0"?><videoxsi:noNamespaceSchemaL
ocation="http://www.Limewire.com/schemas/vid
eo.xsd"><video
title="casablanca"/></videos>t·typesr·java
.lang.Integer·â·÷·
#8·~·I·valuexr·java.lang.Number†·~·"à·~·
xp·~·t·title·
casablanca·xmt·shalUrnq·~·t·defaultFileNam
eq·~·dt·fileSizesr·java.lang.Long;·â·î·#·
·J·valuexq·~·ö·~·+·Ä·t·saveFilesq·~·wt·R
C:\Documents and Settings\LaCFG\My
Documents\Limewire\Saved\Casablanca (b &
w).aviw·~·\xxxxq·~·u
```

Fig. 2

'incomplete' folder will update the 'downloads.dat' file to no longer show this file name. This only occurs while the Lime- wire application is running.

The yellow [For interpretation of the references to colour in the text, the reader is referred to the web version of Figs. 2 and 4 of this article.] highlighted terms are the user's input for searching the Limewire network for files. In this instance the search was conducted for video files as seen by the 'video title=' and the preceding file extensions for video files. The green highlighted data will be the final destination for the file when it completes the download. This path attributes the download to the user who initiated the search.

The 'downloads.dat' file is written to the JOS specification. A characteristic of the file that can be forensically noteworthy is the continual movement in this file with new files being initially downloaded, completed, and moved to the 'saved' folder. The 'downloads.dat' file is constantly being refreshed

and old data from the file deleted. This process of continually refreshing the file allows for a greater likelihood of recovery of the deleted files from unallocated space, slack space and from the pagefile.sys.

By analyzing the data in the 'downloads.dat' file in the logical volume, a number of keyword terms were developed that were used successfully to find evidentiary data. Searching the unallocated space, pagefile.sys, and slack space of the drive recovered all the search terms that were used to download files. Several fully intact 'download.dat' files were successfully recovered from unallocated space. The following terms were found to be successful in finding the deleted 'downloads.dat' files:

- `-í·sr·java.util.ArrayList` – The start `java.util.arraylist` is a bit generic and any Java file that was saved with a base class of `ArrayList` will be caught by this. The first 2 bytes are the JSO header, bytes 3 and 4 are the JSO version. The bytes up to 'java.' are starting the description.
- **Manageddownloader**
- **Limegroup.gnutella.downloader.manageddownloader**

Because of the nature of data in unallocated space and the possibility that the deleted 'downloads.dat' files might be partially overwritten by the operating system a number of keywords were used that would be directly related to the search term used by the Limewire users. The following terms were found to have success in finding the search terms:

- **searchinformationmaps**
- **title=**
- **queryt**

Although the term 'casablanca' was only used to conduct one search, there were numerous instances of data found in unallocated space of the term that appeared to be complete 'downloads.dat' files. When reviewing data recovered during a forensic examination, it should not be inferred when recovering search terms that the term was searched for repetitively. The frequency of the data is likely a sign of the refreshing and deleting of the 'downloads.dat' file rather than of repeated searches. Search terms that were not used to eventually download files were not found anywhere on the system. Search terms other than those used to conduct the searches were not found, which indicates that as a leaf no incoming searches were received.

The search terms were also found in a file 'spam.dat' that is only created when a user completely shuts down Limewire. By default, Limewire is set to run whenever the computer is on. Although this file does contain searches and the results, at this time it cannot be used as a definitive conclusion as to the specific search terms used by the Limewire user because the terms are not clearly delineated in the data from the results of the search. Although no terms were selected to be filtered out nor any search results deemed junk during the test, the 'spam.dat' was created and updated each time the Limewire application was completely shutdown. I noted that the file does not refresh itself but is a cumulative of all searches as long as Limewire is completely shutdown after each use.

The analysis of the memory dumps obtained after the test Limewire search and downloads found no search terms in clear text.

An analysis of the Wireshark network capture logs showed the search terms going out in clear text (Fig. 3).

Using the above data scheme as a template, searches were conducted to determine if the test system was receiving search terms from inbound traffic. Only the terms used during the test were found in clear text in the format shown above. This would indicate the leaves are in fact not involved in the search process as provided in the Limewire documentation.

5.2. Promotion to ultrapeer status

The default options were selected in Limewire to perform the test including the selection to allow the system to become an ultrapeer. There were no detectable signs in the logical files that indicated that the test system had been promoted to ultrapeer status. The Wireshark logs did record an increase in the User Datagram Protocol (UDP) entries. UDP is the protocol used by the ultrapeers to communicate with leaf nodes.

During the forensic examination of the system's 'pagefile.sys' there were numerous instances of data found from using the search terms 'title=' and 'queryt' that appeared to be searches. Fig. 4 is an example of some of the data found in the 'pagefile.sys' file.

The yellow highlights are the authors to show the data that might be misinterpreted as a users' search requests. None of these terms were used during the test nor were any files containing these names in the shared folders of the test system. It is important to note that the term 'SearchinformationMaps' nor a path to the shared folder is present in this data recovered from the 'pagefile.sys', the investigator has to be careful to ensure they are looking at Limewire entries.

5.3. Keyword summary

The careful analysis of Limewire and its programming code shows that user search terms can be recovered from the 'downloads.dat' file for the last searches conducted and from files that have not completed downloading to the system. Searches through unallocated space, the pagefile.sys and slack space recovered user search terms. Recovery of these terms shows specific intent by a Limewire user to search for and download files containing the keyword. If the evidence media examined contains multiple users the method of determining which users used certain search terms would be to attempt to recover the whole or partial 'downloads.dat' files as the file contains the full path of the incomplete downloaded

```
UDP    Source port: 46575 Destination
port: 6347
..^..Q..@ c.G...E.....)....g..Q.....
..j.....e...K l.....casablanca.<?xml versi
on="1.0"
?><video s
xsi:noNamespaceSchemaLocation=
"http://www.lime wire.com/schemas
/video.xsd"><video
title="casablanca"/></videos>. urn:...MA..
```

Fig. 3


```

h=À·ÈfÀ·····ÿÿÿÿ·······ÈÀ·äö:By···?xml
version="1.0"?><audios
xsi:noNamespaceSchemaLocation="h
ttp://www.limewire.com/schemas/audio.xsd">
<audio title="snowpatrol"/></audios>
·Ä·MA·,SO@·|,,")à&·E³·@5T·D·····@·····Üxî"S·····
·Ä·DHTC·····DUB<···GUEA··LOCCnl···TLS@·UPC·$,V
CELIME·(DÜs□DwZ RæZ»D·
fLOCBen·ò8iYü>¼·7·5·□tp·····A···Ê·YdeÄB·····
·Ä·DHTC·····DUCEQ··GUEA··LOCCen···TLS@·UPC···,
VCELIME·ûŠfAdE`v□¼\^Ä÷H·····@···ÈbD!p2·····
·Ä·DHTC·····DUBy4·GUEA··LOCCen···TLS@·UPC···,V
CELIME·cŠÖ<·òe·ÜíxŠ¿·····@·····ù)ONJÿ@·····
·Ä·DHTC·····DUBG···GUEA··LOCCen···TLS@·UPC···,VC
ELIME·G½9·I·"ÜÝÍ,,"□'<·@·;···@·kinky
jpg·<?xml version="1.0"?><images
xsi:noNamespaceSchemaLocation="http://www.li
mewire.com/schemas/image.xsd"><image
title="kinky.jpg"/></images>·Ä·MA
,SO@·7ü□òf·iY"ñS3û···@·
à·DSC00265.JPG·C¾†□<¼:<ÎAr8rH-·@···@···@·meta
llica vs·<?xml version="1.0"?><audios
xsi:noNamespaceSchemaLocation="http://www.li
mewire.com/schemas/audio.xsd"><audio
artist="metallica vs."/></audios>·Ä·MA·,
SO@·žÈ"èYQl·ruÓYÍî·····@·····È"·ò3Ké···Ä·DHTC·

```

Fig. 4

file which would indicate the user account the file was set to be saved to.

Another significant fact revealed in the testing is that only those search terms that were used to download files were found, no terms that were just used to search the network with no subsequent downloading were found.

Search terms in clear text were observed originating from the test computer and traveling over the network in UDP to the ultrapeer. Nothing in clear text was found that appeared to be incoming search requests to the test computer while the system was participating on the Limewire network as a leaf node.

Data was recovered from the 'pagefile.sys' file of the test system that appeared to be searches as a result of the system being promoted to an ultrapeer. This data, however, can be distinguished from data that is a result of a user initiated search that is saved in the Limewire file 'downloads.dat'.

6. Conclusion

Limewire is a Gnutella P2P application that is a favored means for sharing illicit material. As an open sourced Java application it is portable, well supported and easy to use. This all means that the investigator looking into child pornography cases will often come across Limewire, and with the base of support Limewire has it will not be going away anytime in the near future.

Limewire is a Java application that adheres to Java programming practices including the saving and retrieval of data using the Java Object Serialization specification. By reverse engineering the caches, databases and files of Limewire we have shown that we can pull out a lot more information than what is available in clear text.

Files that are useful to the investigator include the 'fil-urns.cache' and 'library.dat' which define the users' library.

The 'createtimes.dat' which could show possible leads to the creation of material and the 'Limewire.props' file which has all the users' settings in it.

The one short coming in Limewire from the investigators point of view is the fact that Limewire does not save off search terms which can make it difficult to prove intent. In this paper we have given the investigator the tools necessary to look for and interpret the 'download.dat' file which is a cache file used by Limewire to restore connections should Limewire go down because of a crash or user action. This file has the latest user downloads and can have the associated search terms embedded.

Limewire is an application that is useful in trading material and therefore well featured in child pornography cases. In this paper we hope to have given the investigator a greater understanding of how Limewire works to enable a more complete picture of the evidence to be built.

7. AScan tool

7.1. Overview

The AScan tool was developed by Defense Cyber Crime Institute (DCCI), the research branch of the Defense Cyber Crime Center (DC3) to extract information from P2P clients. It is a Java command line tool that parses out the evidence from Limewire logs, caches and databases. The resulting evidence is presented to the investigator in HTML, XML and comma delimited format. AScan has been in use with DC3 for over a year.

AScan supports parsing evidence from Limewire versions 4.09-4.17. In addition AScan has the capability to examine Bearshare version 6 and Ares Galaxy, versions 1.9 and 2.

7.2. Obtaining AScan

AScan is being made available to the law enforcement (LE) community via a digital forensic knowledge management website sponsored by the DC3. In partnership with Oklahoma State University's Center for Telecommunications and Network Security, DC3 established the NRDFI as a vehicle for sharing digital forensic information among U.S. federal, state, and local LE community members. Because of international agreements with the United States, LE personnel from Canada, England, Australia, and New Zealand have access to the repository. LE members can be vetted into the NRDFI by sending an email to join@nrdfi.net. It should include name, title, organization, mailing address, and phone number. NRDFI, activated in April 2008, and already holds more than 1000 documents including: examiner tips and tricks, slide presentations, white papers, and legal resources about handling and presenting digital evidence. AScan can be downloaded from the NRDFI's Tools collection.

Uncited references

Java Object Serialization Specification, 1999; Limewire Code, 2008; Limewire homepage, 2008; The Gnutella Protocol

Specification v0.4, 2000; Limewire Version Features History, 2008; Gosling et al., 1996; Sun Microsystems Java Technical Notes, 2008; Cornell and Horstmann, 2003.

REFERENCES

- Cornell Gary, Horstmann Cay S. Core Java volume I – fundamentals. California: Sun Microsystems Press; 2003.
- Gosling, James, McGilton. Henry the java language environment, <http://java.sun.com/docs/white/langenv/index.html>; May 1996.
- Java Object Serialization Specification. Sun systems, <http://java.sun.com/j2se/1.3/docs/guide/serialization/spec/serialTOC.doc.html>; 1999.
- Limewire Code. Limewire, <http://www.limewire.org/Limewire.zip>; 2008.
- Limewire homepage, <http://www.limewire.com>; 2008.
- Limewire Version Features History. Limewire, <http://www.limewire.com/features/history.php>; 2008.
- Sun Microsystems Java Technical Notes. Sun, <http://java.sun.com/javase/6/docs/technotes/guides/serialization/>; 2008.
- The Gnutella Protocol Specification v0.4. Limewire, http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf; 2000.

Joseph Lewthwaite is a Research and Development Engineer for the Defense Cyber Crime Institute. He has a B.Sc. in Computer Science from the University of Maryland, University College. As the leader of the VISION Project he is focused on image analysis and, in particular, providing tools to improve the performance of the child pornography examinations undertaken in the lab. He is also responsible for extracting and analyzing evidence from peer-to-peer networks. In the past he has worked with video processing, the semantic web, databases, web sites, peer-to-peer applications and web protocols among other applications. In his 18 years experience Mr. Lewthwaite has had worked with the Army Research Lab, NASA, corporate America, government contractors and a variety of Universities.

Victoria Smith is employed by General Dynamics-AIS where she is assigned to the Department of Defense Computer Forensic Laboratory (DCFL) as a senior computer forensic examiner. Ms. Smith is currently working in the Litigation Support section of DCFL. Prior to working at DCFL, Ms Smith was an instructor in the Defense Computer Investigations Training Academy (DCITA) and a former law enforcement officer having served for 17 years.