Skype Log File Analysis

Introduction

Skype software uses a number of files to store data. These files relate mainly to historical information, call histories, file transfers, messaging sessions, etc. They also cache user profiles. The interpretation of these log files can yield a significant amount of information about communications that have taken place through the software. This note considers the range and format of user data stored on the platform.

Note that the machine used to conduct this analysis was a Windows platform. There may be some platform specific elements, the extent of which should be outlined below.

Under Skype for Windows, user data files are stored in the user's Application Data folder under the Skype subfolder; this is then subdivided based on Skype user, allowing multiple Skype users to operate under the same Windows account.

Information available in log files

This section details the information available for extraction from Skype logs. Note that the sequence number allows the order of events to be determined, without relying on the resolution of the timestamp. The timestamps give date and time to a resolution of one second.

Calls (e.g. ca11256.dbb)

- Sequence Number
- Time stamp
- Username (remote end)
- Screen name (remote end)
- Duration of call (seconds)
- PSTN number (when using SkypeIn or SkypeOut)
- PSTN status (when using SkypeIn or SkypeOut)

File transfers (e.g. transfer512.dbb)

- Sequence Number
- User name (remote end)
- Display name (remote end)
- Full saved file path
- Filename
- File size
- Time stamp

Messages (e.g. msg256.dbb or chatmsg256.dbb)

- Sequence Number
- Message content
- Chat ID (groups messages within a chat session)
- Timestamp
- User name (sender)
- Display name (sender)

User profiles (e.g. user1024.dbb or profile1024.dbb)

- Username
- Display name
- Language (2-character ISO code)
- Province / city
- Country code (2 character ISO code)
- Phone number
- Office number
- Mobile number
- Thumbnail image

Voicemails (e.g. voicemail256.dbb)

- Sequence Number
- Username
- Display name
- Time stamp
- VM Filename

Interpretation of Skype log files

File-naming convention

Files are stored with a .dbb extension with the filename consisting of a string describing the contents followed by a number which indicates the record length (e.g. call256.dbb, chatmsg512.dbb etc). The minimum record length observed is 256 bytes, with files seen up to 16384 bytes. Items are stored in the smallest length format possible with blank padding to fill any space remaining in the record. Therefore it is quite common to have multiple files with the same prefix and different record lengths.

Skype Log File Analysis	**Skype Log File Analysis**
call*.dbb	Call history
chatmsg*.dbb	Chat history
profile*.dbb	Details of user profiles
transfer*.dbb	Details of file transfers
chat*.dbb	Chat history
contactgroup*.dbb	Unknown
user*.dbb	Local user's profile
voicemail*.dbb	Details of voicemail messages (no contents)

File format

The Skype log files are stored in a flat-file database format.

- Although what's stored in each record differs between files, the underlying structure is the same (e.g. records, record headers etc).
- Only items of the type indicated by the filename are stored in a file.
- Numeric items are stored in little-endian format. For example 1453 is 0x05AD and would appear as two bytes, 0xAD, 0x05.
- A record begins with a 4-byte record header, which is always 0x6C, 0x33, 0x33, 0x6C (or "1331" in ASCII).
- The record header is followed by a 4-byte, little-endian, unsigned integer which indicates the length of the following data. The filename indicates the maximum number of data bytes to be stored in a record in the file. The record length in the file will be less than or equal to this number. The total space allocated to each record is the maximum data length (given by the filename) plus the header size (8 bytes).
- Blank entries have been observed in files. These include the 4 byte header, but have a data length indicator of 0 with all following data bytes set to 0x00.
- There is no indicator for the end of a record - any space remaining in the record after the data is set to 0x00. The last record in a file is often not padded out.
- Any communications item (call, message, file transfer etc) has a sequence number associated with it. This is used by the software to reassemble the order in which events occurred. This is particularly useful as items of different sizes can often be spread across multiple files.
- It is unknown how multi-byte character sets are handled as standard ASCII is all that has been observed for storing textual information.
- Within a record, and between different pieces of information, delimiters indicate the next data item. The meaning of these delimiters varies between files and they are summarised in the individual file type sections below.

Start	Length	Value	Description
0	4	0x6C 0x33 0x33 0x6C	Record header
4	4	Length	Length of record following this entry
8	4	Sequence number	Only appears in communication records
Multiple locations throughout the file	Varies	Data item indicator	Item delimiter within the field. Value indicates what information is stored
	Varies	Data item	Item following indicator. Can be any length – strings are null-terminated.

Time stamp

The timestamps in records are stored in the standard UNIX format of seconds since 1/1/1970 00:00:00. However the data isn't represented as the standard 4 byte unsigned integer as is usually the case, instead using 5 bytes to represent the timestamp with the format below (where 'd' indicates a bit of data):

0000dddd 1ddddddd 1ddddddd 1ddddddd 1ddddddd
↑ ↑
Most significant byte *Least significant byte*

Stripping the leading 0's from this and removing the '1' at the most significant bit or each byte yields 32 bits of information. If these are concatenated into 4 bytes, the number can be handled using standard time functions. It is unknown why this format has been chosen.

Status flags

Many of the record types listed below also have a number of status flags associated with them. There is data in each of the records that has not been fully interpreted and so there is the potential for more useful information to be gleaned. In particular the user profiles have large amounts of data which has not been decoded yet.

Call history

The call history file contains the following items:

Data item indicator	Data item	Format
0xA1 0x01	Time stamp	As described above
0xA4 0x01	Username (remote end)	Null-terminated string
0xA8 0x01	Screen name (remote end)	Null-terminated string
0x04 0x00	Direction	0xA1 = incoming 0xF1 = outgoing
0x85 0x02	Duration of call (seconds)	4-byte unsigned integer. If call failed, record will not exist.
0x80 0x02	PSTN number	Null-terminated string
0x8C 0x02	PSTN status	Null-terminated string

File transfer

The file transfer log contains the following data items:

Data item indicator	Data item	Format
0xC4 0x02	Username (remote end)	Null-terminated string
0xC8 0x02	Display name (remote end)	Null-terminated string
0xDC 0x02	Full saved file path	Null-terminated string
0xE0 0x02	Filename	Null-terminated string
0xE4 0x02	File size	Null-terminated string
0xD5 0x02	time stamp	As described above

The file location represents either the location of the file that was sent (if log file belongs to sender), or the location the file was saved to (if log file belongs to receiver).

Message history

The message file contains the following data items:

Data item indicator	Data item	Format
0xFC 0x03	Message content	Null-terminated string
0xE0 0x03	Message ID	Null-terminated string
0xE5 0x03	Time stamp	As described above
0xE8 0x03	User name (sender)	Null-terminated string
0xEC 0x03	Display name (sender)	Null-terminated string

The Message ID is a string which uniquely identifies a chat session. Consequently the thread of messages can be assembled from grouping those items with identical message IDs.

Profile

The profile record stores the local user's profile - it may be in the same format as the user data listed below. It contains the following information:

Data item indicator	Data item	Format
0x03 0x10	Username	Null-terminated string
0x03 0x14	Display name	Null-terminated string
0x03 0x24	Language (ISO code)	Null-terminated string
0x03 0x28	Country code (ISO code)	Null-terminated string

User data

User data records contain the following information:

Data item indicator	Data item	Format
0x03 0x10	Username	Null-terminated string
0x03 0x14	Display name	Null-terminated string
0x03 0x24	Language (ISO code)	Null-terminated string
0x03 0x30	Province / city	Null-terminated string
0x03 0x28	Country code (ISO code)	Null-terminated string
0x03 0x34	Phone number	Null-terminated string
0x03 0x38	Office number	Null-terminated string
0x03 0x3C	Mobile number	Null-terminated string

User records can also include embedded JPEG format images. These are used in the GUI and displayed alongside the username etc. A standard data item indicator has not been found; however the JPEG file will always begin with 0xFF 0xD8 and finish with 0xFF 0xD9.

Voicemail

Voicemail records contain the following information:

Data item indicator	Data item	Format
0x94 0x03	Username	Null-terminated string
0x98 0x03	Display name	Null-terminated string
0xA9 0x03	Time stamp	As described above
0x03 0x0F	VM filename	Null-terminated string

Note: Voicemails are initially stored on the event F/E (front-end) server and are downloaded to the local machine prior to playback. It is not known whether the VM 'chunks' are stored in some intermediate form prior to being reassembled for playback. (Need to add the CODEC used for VM storage.)

(Q: Are voicemails physically deleted, i.e., overwritten, when deleted by clicking on the delete button on the UI? And, if deleted on one platform, such as 'home computer', are they also marked for deletion on the user's other platforms, such as 'work computer'?)

Also needed:

Synchronization buffer – structure and timeouts (i.e., when does an entry in the synch buffer cache expire?)

Credentials storage – In Windows we use the CAPI cryptographic creds storage routines. What about on other platforms? (PPC, Symbian, Linux, Mac)