A comparative study of the structure and behaviour of the operating system thumbnail caches used in Kubuntu and Ubuntu (9.10 and 10.04)

Sarah Morris[1], Howard Chivers[2]

Centre for Forensic Computing,
Cranfield University, Shrivenham, SN6 8LA, UK

[1]S.L.Morris@cranfield.ac.uk

[2]H.Chivers@cranfield.ac.uk

Abstract

Browsing directories in thumbnail mode can assist the user in locating relevant documents quickly by providing a graphical representation of each file. Whilst thumbnail images can assist the user, they can be resource intensive to generate; therefore operating systems generally cache these images, along with associated metadata, to prevent unnecessary rendering. Thumbnail caches can provide a variety of information, including file paths and images of documents; however it is necessary to understand the user activity which resulted in the artefacts being created to understand their forensic significance.

This research used baseline versions of Ubuntu and Kubuntu (both versions 9.10 and 10.04) in virtual machines to determine the effects of a variety of user actions on the information stored in the thumbnail caches. A series of experiments were conducted to identify the structures used to store artefacts both in the thumbnail cache and any related file throughout the system, as well as determining the meaning of each artefact. Each experiment was performed on both operating systems and mimicked a variety of typical user behaviours, such as moving a file or accessing a USB stick.

Whilst both thumbnail caches implement the same structure for storing data, the user behaviour which leads to artefacts being stored in the thumbnail caches differs considerably between the two operating systems. Other information about user activity can be deduced from the thumbnail cache itself; for example, Kubuntu uses an RGB format for items cached without the directory being viewed and uses an RGBA format for standard record creation. This paper also identifies the user activity which led to artefacts being recovered and discusses the strengths and weaknesses of the thumbnail caches.

This research shows that similar artefacts from two closely related operating systems may nevertheless suggest different types of user activity, and hence have a different forensic significance.

Introduction

Thumbnail images are used in a variety of applications and operating systems to enable the identification of various source files; for example when browsing directories. Whilst the thumbnail images enhance the usability of applications, they can be resource intensive to generate, particularly if a directory contains multiple files which require the generation of thumbnails. A thumbnail cache allows images to be stored to prevent unnecessary rendering, therefore requiring less resources. Such caches are of particular significance to forensic examiners as they can contain metadata, a thumbnail image of the original file and information about the user's activity.

There is no standard implementation of a thumbnail cache, therefore software applications and operating systems each use the basic concept in their own way. This can be a source of confusion for forensic examiners as thumbnail caches which look structurally similar may not represent the same activity, and may therefore have a different evidential significance.

This paper shows the structural similarity of the thumbnail caches implemented in Ubuntu and Kubuntu and the differences in the user activity which lead to their generation. It also highlights the requirement to educate forensic examiners to identify the context of any evidence identified. The experiments were performed using standard ISO images of Ubuntu and Kubuntu downloaded from their WebPages.

This paper discusses the operating system thumbnail cache used by Ubuntu and Kubuntu (9.10 and 10.04) and is structured as follows. Firstly *Section 2* provides relevant background information on operating system thumbnail caches and their forensic significance. In *Section 3*, the structures of the thumbnail caches are described. *Section 4* identifies the user actions which cause information to be present within the thumbnail cache. *Section 5* compares the thumbnail caches used in Ubuntu and Kubuntu. Finally *Section 6* concludes the paper and discusses the possibilities for future research.

2.0 Related Work

This section provides relevant background information and explains why thumbnail caches are important in digital investigations.

Paivio [1] has shown that the use of images when searching can improve a user's ability to identify information. This concept has been integrated into computing by the addition of graphical user interfaces and thumbnail images. Ubuntu uses Nautilus, and Kubuntu uses Dolphin, for thumbnail image creation and storage [2]. Thumbnail images of a variety of file types are stored in a centralised directory, which can be accessed by any application. Open Office.org [3] created a script which allows thumbnails for open office documents to be created and stored, increasing the variety of images found in the thumbnail cache. Whilst this script is currently not installed by default on Ubuntu or Kubuntu, it is available to download on the Open Office website.

Whilst thumbnail caches are implemented by a variety of software and operating systems, there is no standard structure. This can lead to a variety of forensic artefacts being available, each created by a different user activity, making it difficult for a forensic examiner to establish the value of any artefacts they recover. A Free Desktop specification for thumbnail caches was created in 2004 [4], it aims for each system to implement a single cache which can be utilised by both the operating system and applications. File managers such as Konqueror and Nautilus base their thumbnail caches on this specification. The specification does not specify the user activity which leads to the thumbnail cache being manipulated. Therefore whilst both file managers may attempt to implement the specification the user activity involved and therefore the forensic significance of the artefacts may be significantly different in each case.

Whilst images form a significant part of the cache, the metadata may also be useful to investigators. In the case of R v Ross Warwick Porter [5], the defendant was accused of possessing indecent images of children. A forensic examination of his computer showed a piece of software with its own thumbnail cache had been used to store graphical images by the defendant; however the images had been deleted, leaving only metadata in the cache [6]. At appeal the defendant's conviction was overturned on the grounds he was not in possession of the indecent images as he did not have the means or knowledge to retrieve them from slack space. Despite the overturning of the conviction this case still highlights the role metadata from the cache can have in ascertaining whether the files are known to the defendant, had the files not been deleted but moved to another storage media it would have shown a relationship between the devices and may have assisted a conviction.

Modification, access and creation (MAC) times of a file can give the investigator an insight into how the files arrived in their current location. Chow proposed seven rules based on these times, which assist in building event timelines [7]. The rules elaborate on the relationships between MAC times and the behaviour of the file. Chow reviews the value of the thumbnail cache, in relation to these rules, for establishing information relating to the viewing of such files (Scenario 9 in [7]). This suggests that thumbnail caches can be used to assist in building a timeline of activity on a machine.

This section has identified the use of thumbnail images in Ubuntu and Kubuntu, and identified their potential significance in digital investigations. Thumbnail images are used to improve a user's identification of material, in order to minimise the use of system resources the images are usually cached. The thumbnail caches also store metadata, which can be useful in establishing event timelines. Entries in a thumbnail cache can be useful in establishing user activity and forming a relationship with files.

3.0 Methodology

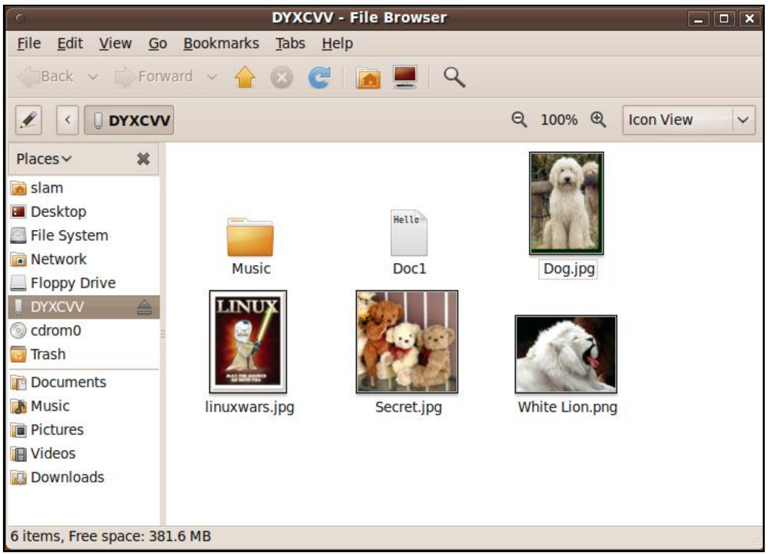
To establish the structure and behaviour of the thumbnail caches, four baseline virtual machines were created. Each virtual machine had a six gigabyte hard drive; the operating systems Ubuntu and Kubuntu were installed using ISO images downloaded from Ubuntu's website. To mimic a standard user's setup, the default options were selected during installation and no additional software was loaded onto the machine. The baseline machines were then cloned, for each experiment; this enabled the results to be compared with a fresh installation.

In some cases, it was necessary to compare the results of several experiments, such as when looking at the modification of data within the thumbnail cache. All comparisons were conducted in WinHex, with the Virtual disk file being 'interpreted as a disk' in order to ascertain the structure of the operating system.

To investigate the structure of the thumbnail caches, various files were loaded onto the computer using a mixture of methods: USB sticks, CDs, downloading from the Internet and creation of a file using pre-installed software. The records created in the cache were then analysed and compared with both the Free Desktop Thumbnail Cache and PNG specifications. The data collected from these experiments also assisted in establishing how information is added to the thumbnail cache. To investigate the modification of data within the thumbnail cache, source files were altered, renamed and given new Uniform Resource Identifiers (URI). To find out when information is deleted, source files were deleted from the virtual disk. Also USB devices were not plugged into the system for a period of thirty days to establish if this would cause the thumbnail cache to remove unused data.

4.0 The structure of thumbnail cache information

In Ubuntu and Kubuntu, the default setting within the file browser is to display thumbnail images of files. However not all file types display thumbnail images by default, for example office documents currently do not. For files such as image formats, the thumbnails shown in the browser are of a large enough size to aid file identification. Whilst generally thumbnail images are thought to be a graphical interpretation of an image, or the first page of a document, thumbnails of text files use the first few characters from the document. The thumbnail images of text documents are not stored in the thumbnail cache and are generated every time the directory is viewed. Figure 1 shows a file browser and the thumbnail images generated when using Ubuntu's default settings.



6 items, Free space: 381.6 MB

Figure 1: The default view in Ubuntu, when opening a directory in a file browser

Whilst the specific methods of thumbnail image generation are beyond the scope of this paper, it is worth noting that most thumbnail images are created through the default thumbnail process. In Ubuntu this is known as the Gnome Thumbnail Factory; other software can use its own thumbnail creation process and use the .thumbnails cache.

Structurally, both Ubuntu and Kubuntu have implemented elements from the Free Desktop specification for their thumbnail caches [4]. The following sections provide an overview of the structures present in both the Ubuntu and Kubuntu thumbnail caches. Firstly the overall structure of the thumbnail cache is described; this is followed by an examination of the thumbnail cache record structure.

4.1 .thumbnails

The main thumbnail cache structure is a hidden directory in the user's home folder, named '.thumbnails'. This directory behaves like any other hidden directory in the operating system. Figure 2 shows the directory structure for the thumbnail cache. Thumbnail records are not stored in the .thumbnails directory itself, but in one of three sub-directories: fail, large and normal. There is generally only one thumbnail cache present on the system which can be accessed by any software as required

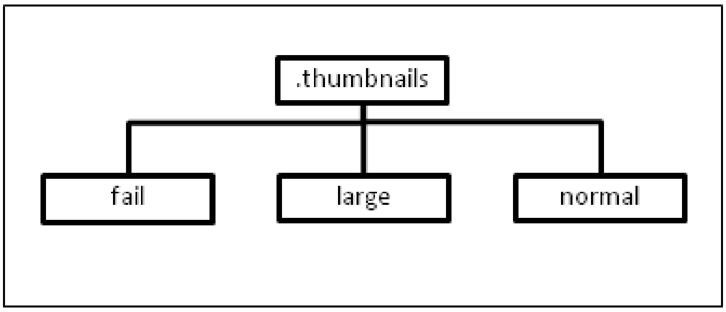


Figure 2: The structure of thumbnail caches in Ubuntu and Kubuntu (9.10 and 10.04)

4.2 thumbnail cache records

A *record* within the thumbnail cache can be defined as a complete set of data relating to an original file. In the case of Ubuntu and Kubuntu, a thumbnail cache record can be defined as the set of files located in .thumbnails or any of its subdirectories which have the same filename and extension as each other. A s*ub-record* can be defined as a file stored in the thumbnail cache, which contains metadata about an original file; sub-records may contain a thumbnail image. In Figure 3, shows the thumbnail sub-records for the file z2nxaujt.jpg, together these sub-records form a complete record of thumbnail cache information about the source file.

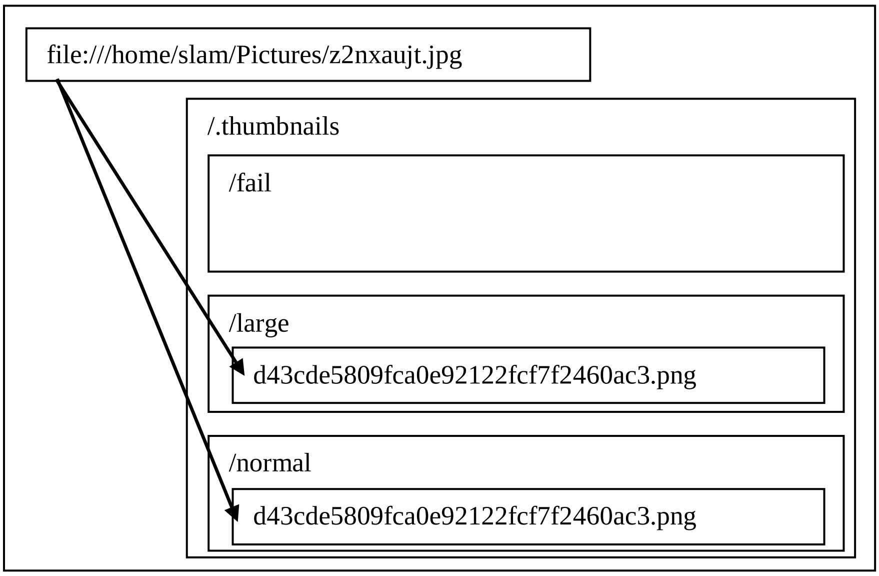

file:///home/slam/Pictures/z2nxaujt.jpg
/.thumbnails
/fail
/large
d43cde5809fca0e92122fcf7f2460ac3.png
/normal
d43cde5809fca0e92122fcf7f2460ac3.png

Figure 3: A diagram showing the thumbnail cache sub-records which make up the complete record for the file z2nxaujt.jpg

In operating systems such as Windows 7, a centralised index file, thumbcache_idx.db is located in the thumbnail cache to provide a single record for each original file. The records within the index contain pointers to the location of the various image sizes and associated metadata for each original file. In Ubuntu and Kubuntu, there is no centralised index for each record, and therefore the only way of identifying a complete record is to search for sub-records which have the same filename and path as each other, and where the filename is an exact MD5 hash of the URI of the source file.

Each sub-record has a PNG file extension; the filename is an MD5 hash of the URI of the original source file. A thumbnail cache sub-record is stored in the PNG format, which is open standard. The structure can be broken down into a series of ‘chunks’, each of which has a defined structure. Each record generally consists of a header and up to five different types of data chunks; IDHR, SBIT, TEXT, IDAT, IEND. Each chunk consists of four bytes denoting the size of the data stored, four bytes denoting the type of the chunk, a data section, and a four byte CRC value.

The header of a sub-record is in the standard PNG format, showing the file consists of a single image. This is followed by the SBIT chunk, which defines the significant bits of the colours; generally this is set to 8bit, non-interlaced, with full alpha transparency.

The TEXT chunks contain the metadata on the original file and the thumbnail image; there are usually several TEXT chunks in each file. Each TEXT chunk holds a single piece of metadata in the form of a key and a value, the key specifies the meaning of the value. The keys may be inbuilt into the PNG format, specific for the thumbnail cache or for the file type. Thumbnail cache specific keys start with ‘Thumb’, all keys are written in Ascii, and are easy for an forensic examiner to interpret; for example ‘Thumb:MTime’, is the key for the modification date of the source file. The modification time and URI of the source file are generally present in all sub-records. Other information, such as the dimensions of the original file and the number of pages it contained may also be present.

The thumbnail image itself is stored within one or more IDAT chunks, which are compliant with the PNG specification. Finally each sub-record file ends with an IEND chunk, which denotes the end of a file.

4.3 The normal and large directories

The normal and large directories contain sub-records which the thumbnail creation methods have generated successfully. The normal directory is most frequently populated and contains sub-records whose thumbnail image is no larger than 128x128 pixels. In the large directory, the thumbnail image may be up to 256x256 pixels. Whilst the sizes for the directories have been defined, individual thumbnail creation methods may not follow the specification and therefore anomalies may occur.

4.4 The fail directory

The fail directory contains incomplete sub-records, where the generation was unsuccessful. The sub-record may have failed for a variety of reasons, for example, the source file contains an error. The fail directory provides a way of storing information on files which were not successfully added to the cache to prevent the system attempting to create the thumbnail image again, saving resources. These records may contain some useful metadata; however it is necessary to establish the reliability of the artefacts within the fail directory.

4.5 Conclusion

This section has described the structure of the thumbnail caches used in the Ubuntu and Kubuntu operating systems, which are contained within the hidden directory .thumbnails. This directory contains three sub-directories, which allow the storage of multiple sizes of thumbnail images and information on files which have unsuccessfully attempted to generate data for the thumbnail cache. Each original source file may have multiple sub-records within the cache, each stored in the PNG format. Each sub-record contains metadata about both the sub-record and the original source file and it may also contain an image. Whilst it is outside the scope of this paper, there are a number of interesting artefacts throughout the Ubuntu and Kubuntu file systems, which are created as a result of actions on the thumbnail cache which may provide information to a forensic examiner; for example the directory listings.

This section has identified the structures used in the Ubuntu and Kubuntu thumbnail caches as being the same, the next section shows the user activity which leads to artefacts being present in the cache, therefore allowing a context to be placed on the evidence.

5.0 The behaviour of the thumbnail cache

By default, when Ubuntu and Kubuntu are installed, the thumbnail cache directory does not exist. Instead, the directory is created when the first call to create a thumbnail image is made. This section shows the primary methods used to alter the information within the thumbnail cache, and discusses the resulting artefacts.

5.1 Adding a sub-record

When a thumbnail image is requested, the system first checks if the image is already being stored in the thumbnail cache. In order for this check to occur, the system attempts to form a relationship between the original source file and a sub-record within the thumbnail cache. To establish if a relationship occurs, an MD5 hash of the source files URI is calculated, which is used as a filename. An extension of ‘PNG’ is added to the filename which is then compared with sub-records in the thumbnail cache to look for an exact match. If a record is found with a matching file name and extension, further checks are performed to ascertain whether the sub-record requires modification. These checks are discussed further in *5.2 Modifying a sub-record*.

If no matching sub-record is identified, then a new one is created, using the structure identified in *Section 3*, and the thumbnail creation method for that file type. When the sub-record is initially being written, a six character random extension is appended to the end of the filename. This extra extension is removed upon the successful creation of the sub-record and ensures that only complete records are accessed through the thumbnail cache. If a sub-record is not successfully completed, the extra extension is removed, and the file is moved to the fail directory. When performing an analysis it is possible a forensic examiner would find several entries in the directory with the file extension; occasionally these can contain partial sub-records.

5.2 Modifying a sub-record

When a request is made to find a thumbnail image for a source file and a match is found, a further check must be performed. This is to ensure the thumbnail image is an accurate representation of the current state of the source file. The last modification time of the source file is checked with the thumbnail record, to establish if they are the same. Checking the dates are the same ensures the image is updated if the file has been modified since the sub-record was last updated.

For example, if a file with the same filename and extension as an existing file is added to a directory, the existing file may be overwritten. Since the new file has the same URI, it would generate the same MD5 hash (see 4.2), therefore the thumbnail cache entries for that URI would be outdated. Since the new file may have a modification time that is greater than the previous file, checking the time is an exact match helps to ensure the correct record is displayed.

By default, no checks, other than those in the PNG standard, are performed on the IDAT chunks of a sub-record to ensure they are an accurate representation of the file. Should a file require modification, the entire record is re-written, with the additional six character extension appended to the end, which is removed after successful creation.

5.3 Deleting a record

If a record is no longer required, a call can be made to the system to delete a record. As each sub-record is a unique file, their deletion is handled the same as any other file. Traces of sub-records can still be found, which may enable a relationship with a source file to be formed.

5.4 Conclusion

This section has shown how the thumbnail cache structures outlined in *Section 3* can occur in the system. To find a sub-record for a file, the MD5 hash of the source file’s URI, with the extension PNG, is searched for. If an exact match cannot be found, a new sub-record is created. If a match is found, the last modification time is checked to establish if the time is an exact match with that of the source file. An exact match of time is used as there are circumstances where the last modification date of the file may be greater than the one stored in the sub-record. Since any addition or modification to the cache involves writing an entire sub-record, the system appends a six character extension to the filename to ensure it can only be accessed when it is successfully created.

6.0 A comparative review of Ubuntu and Kubuntu thumbnail caches

In the previous two sections, the structure of the thumbnail caches were identified, together with structural changes to the information within the cache when information is added, modified or deleted. This section shows the user activity, which can cause the structures previously identified to be present in the cache and how they differ between the two operating systems.

6.1 Adding and modifying a sub-record

For thumbnail creation, both operating systems operate a queue, into which potential source files can be added. Once a source file reaches the front of the thumbnail creation queue, the system attempts to locate a matching sub-record in the appropriate directory of the thumbnail cache. If a matching file cannot be found, or the URIs match but the modification times do not, the system creates a new sub-record using the appropriate generation method for that file type. This section identifies when source files are added to the thumbnail generation queue.

In Ubuntu, sub-records are generally added to the *normal* directory, with each potential source file viewed in the file browser added to the thumbnail queue. Sub-records may not be created or updated if a directory contains a substantial number of files, which are potential thumbnail cache source files, and the directory is no longer being viewed in the file browser. This is because the files are taken out of the thumbnail cache queue when the directory is no longer being viewed. By default, potential thumbnail cache source files for Ubuntu are standard image formats and PDFs.

GIMP, a pre-installed graphics program can initiate the creation and update of sub-records to the *normal* directory when the user selects ‘open image, and potential thumbnail cache source files are viewed as directories are being viewed. F-Spot also causes sub-records to be updated and created when importing files; however these sub-records are only added to the *large* directory.

In Kubuntu, both the *large* and *normal* directories are frequently used; therefore a greater amount of information is available to a forensic examiner, than that provided by Ubuntu. When the file browser is open, any potential source files are added to the thumbnail queue for both the large and normal directories. If there is only a single potential source file in the directory, only a large record is added on the first viewing. If within the file browser there are directories that contain potential source files, these may also be added to the thumbnail queue; this is discussed further in *Section 6.4.* The information panel within the file browser is the reason for the creation of sub-records in the large directory, which enables the file to be previewed in a larger size.

Gwenview adds files to the thumbnail queue, when browsing directories in the main window of the application; however it only uses the normal thumbnail cache directory, and therefore the large directory is not modified.

If within Kubuntu, the large thumbnail cache directory is viewed, the files within it are added to the thumbnail queue for the creation and modification of sub-records within the normal directory. These sub-records do not conform to the standard structure; instead they use the same structure as the partial sub-records which are discussed in *Section 6.4*. Potential thumbnail source files in Kubuntu by default are only image formats. Interestingly, thumbnails for text files are not available in the file browser. If a text file is selected within the file browser, a thumbnail image is rendered in the information panel, but this is not stored in the cache.

In both operating systems, potential source files viewed on the desktop are added to the thumbnail queue for sub-records to be added or modified in the normal directory only. However if the desktop is viewed in the file browser in Kubuntu, sub-records may also be created or modified in the large directory.

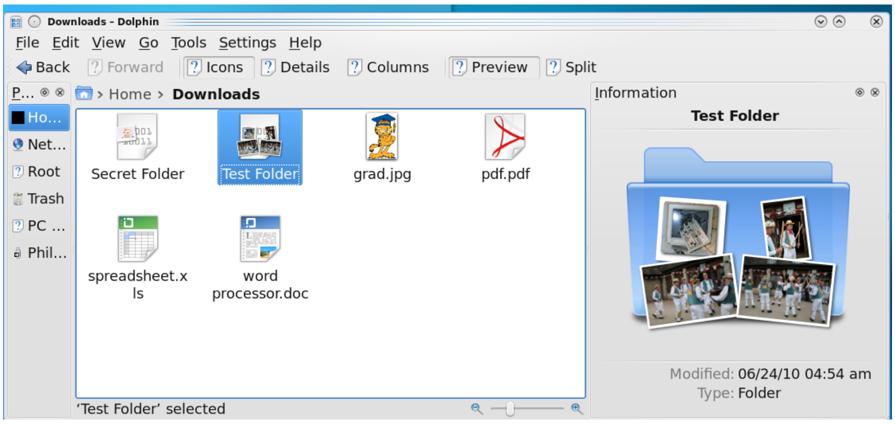
6.3 Deleting a record

From the tests conducted on Ubuntu and Kubuntu, no evidence of the deletion of sub-records was recorded. Tests involved deleting and moving files on the hard drive and external storage media. A test was also carried out where a USB stick was mounted on the virtual machines, to enable the creation of thumbnail cache records and then the device was dismounted. The machine was then left for thirty days to ascertain if unused sub-records would be removed, however no entries in the thumbnail cache were altered.

6.4 Anomalies

Within Kubuntu, it is interesting to note that there are circumstances where thumbnail sub-records are unexpectedly created. Partial sub-records within the cache are automatically generated for a selection of files in the first level sub-directories of a directory being viewed in the file browser. An example is shown in Figure 4, which shows the Downloads directory being viewed in the file browser. When this directory is opened, each sub-directory is represented by a series of small thumbnails, such as those shown for ‘Test Folder’. These small thumbnails represent files present in the sub-directory, and to create them, the system creates partial thumbnails of files within the sub-directory. If the sub-directory is hovered over with the mouse, the mini-thumbnails rotate through any files within the sub-directory that have thumbnail generation methods.

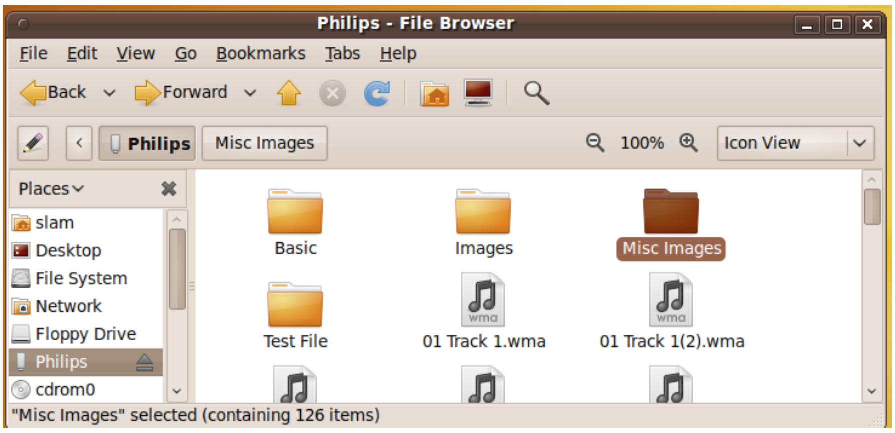
Therefore from each first level sub-directory in a browsed folder, there may be sub-records created for up to the number of files which have thumbnail generation methods, even though the user has not browsed the sub-directory. Interestingly, partial thumbnail sub-records have two properties which allow them to be distinguished from sub-records created using standard methods. Firstly, there is a noticeable lack of metadata within the PNG, including no information on the last modified time and URI for the source file. This means there are two main methods of forming a relationship between a partial sub-record and the source file; the image stored in the sub-record and the MD5 hash of the URI which is present in the sub-records filename. The second difference in partial sub-records is noticeable in the IDAT chunks, where upon comparison with a standard sub-record entry for the same file, it is noticeably shorter as the alpha values are missing.



Modified: 06/24/10 04:54 am
Type: Folder

'Test Folder' selected

Figure 4: The directory 'Downloads' being viewed in the file browser within Kubuntu



"Misc Images" selected (containing 126 items)

Figure 5: The directory 'Misc Images' being viewed in the file browser within Ubuntu

By default, when viewing directories in Ubuntu, sub-directories are represented by folders, as can be seen in Figure 5. Upon selection of a sub-directory, there is no visual change in the thumbnail image, and no partial sub-records have been stored in the thumbnail cache.

In Ubuntu a curiosity was discovered when attempting to alter the thumbnail image stored in the IDAT section of a sub-record. In order to successfully fool the system into displaying an incorrect thumbnail, all other metadata within the sub-record must be accurate. This includes the last modification time, as discussed previously, which must be exactly the same as the source file, or the sub-record is rewritten. It is also important to ensure that each part of the chunks within the sub-record are accurate, for example, if an existing sub-record was being altered it would be necessary to alter the size of each chunk and the CRC data to ensure they matched the new information. If the sub-record uses a valid PNG structure, and the last modification date is equal to the source file then, the new image placed in the IDAT section would be viewed instead of a representation of the actual file.

Table 1: A summary of the user activity which may lead to sub-records being present in the thumbnail cache

		Ubuntu	Kubuntu
Sub-records in /*fail*		- Sub-record creation halted before creation - The file contained an error - An error occurred during sub-record generation	
Sub-records in /*normal*	Standard sub-record	- Viewing directories in the file browser - Viewing a directory in GIMP using ‘open image’	- Viewing directories in the file browser - Viewing directories in Gwenview
	Partial sub-record		- Viewing a folder for a directory in the file browser - Viewing records in the *large* directory
Sub-records in /*large*		- Viewing directories in F-spot	- Viewing directories in the file browser
Sub-records in any directory with an additional six character extension		- Sub-records which are in the process of being generated - The remains of a generated sub-record	

However if this ‘tampered’ record is selected, a new thumbnail image is shown which accurately represents the file. The tampered record is not, however overwritten in the thumbnail cache. Instead, the accurate image is generated when required, by the system and does not appear to be stored anywhere. If the browser window is refreshed or reopened, the tampered thumbnail is still displayed, until the file is selected.

6.5 Conclusion

This section has shown that the user behaviour which leads to information within the thumbnail cache being altered differ among the two operating systems.

Table 1 shows a breakdown of the user behaviour, which may be responsible for artefacts being present in the thumbnail cache, based on the default packages installed during installation. This section has also shown that Kubuntu makes significantly more thumbnail sub-records, thereby providing a greater range of evidence to a forensic examiner.

7.0 Conclusions

This paper has shown that despite the structure of the thumbnail caches implemented in Ubuntu and Kubuntu being the same, the user behaviour which leads to the creation of artefacts can be significantly different. This highlights the need for forensic education to take careful account of the context of an artefact and not just the structure.

8.0 References

1 Paivio A, Begg I (1974). Pictures and Words in Visual Search. *Memory & Cognition 2*, 3, 515-521, 1974.

2 *Ubuntu: Lucid*, www.ubuntuguide.org/ (visited June 2010)

3 *Open Office.org – The Free and Open Productivity Suite*, www.openoffice.org (visited June 2010)

4 *Thumbnail Managing Standard*, http://people.freedesktop.org/~vuntz/thumbnail-spec-cache (visited June 2010)

5 R v Porter (Ross Warwick) [2006] EWCA Crim 560

6 *The Spanner Trust*, http://www.spannertrust.org/documents/R_v_Porter.pdf (visited June 2010)

7 Chow K, Law F, Kwan M, Lai P (2007). The Rules of Time on NTFS FileSystem, Systematic Approaches to Digital Forensic Engineering