

# Forensics of BitTorrent

Jamie Acorn

Technical Report

RHUL-MA-2008-04

15 January 2008



Department of Mathematics

Royal Holloway, University of London

Egham, Surrey TW20 0EX, England

<http://www.rhul.ac.uk/mathematics/techreports>

# Forensics of BitTorrent

**Jamie Acorn**

**Supervisor: John Austin**

Submitted as part of the requirements for the award of  
the MSc in Information Security at Royal Holloway,  
University of London.

I declare that this assignment is all my own work and that I have acknowledged all quotations from the published or unpublished works of other people. I declare that I have also read the statements on plagiarism in Section 1 of the Regulations Governing Examination and Assessment Offences and in accordance with it I submit this project report as my own work.

Signature:

Date:

# Table of Contents

Table of Contents .....	2
EXECUTIVE SUMMARY .....	4
INTRODUCTION .....	5
1.1 What is Bit Torrent and how does it work?.....	5
1.2 The BitTorrent Client.....	7
1.3 Legal Issues.....	9
1.4 Security Issues .....	10
1.5 BitTorrent Erasure .....	11
Aims of this study.....	12
METHOD.....	13
2.1 Testing .....	13
2.2 Imaging .....	15
ANALYSIS.....	17
3.1 Analysis Methodology .....	17
3.2 Torrent file analysis.....	18
BitComet (0.89) .....	19
3.3.1 Test 1_1 analysis .....	19
Registry Analysis.....	19
Encase Analysis.....	21
3.3.2 Test 1_2 analysis .....	23
3.3.3 Test 1_3 analysis .....	24
uTorrent (1.6.1) .....	26
3.4.1 Test 2_1 analysis .....	26
Registry analysis .....	26
EnCase Analysis .....	27
3.4.2 Test 2_2 analysis .....	30
3.4.3 Test 2_3 analysis .....	30
3.4.4 Test 2_4 analysis .....	32
Azureus (3.0.1.2).....	34
3.5.1 Test 3_1 analysis .....	34
Registry analysis .....	34
Encase Analysis.....	35
3.5.2 Test 3_2 analysis .....	39
3.5.3 Test 3_3 analysis .....	40
Another BitTorrent Client (3.1).....	42
3.6.1 Test4_1 analysis .....	42
Registry Analysis.....	42
Encase Analysis.....	43
3.6.2 Test4_2 analysis .....	46
3.6.3 Test 4_3 analysis .....	47
BitTornado (0.3.18) .....	49
3.7.1 Test 5_1 analysis .....	49
Registry analysis .....	49
EnCase Analysis .....	51
3.7.2 Test 5_2 analysis .....	53

DISCUSSION.....	54
4.1 Erasure of BitTorrent activity.....	59
4.2 Limitations of the study and implications for future research .....	60
BIBLIOGRAPHY.....	62
APPENDICES .....	64

# EXECUTIVE SUMMARY

The aim of this study was to identify forensic artefacts produced by BitTorrent file sharing, and specifically, to establish if the artefacts could lead to identification of the files downloaded or the files shared. A further objective was to identify any artefacts that could determine IP addresses of remote computers from which data was downloaded, or shared, during the test phase. The final aim was to test whether automated erasing software would delete the BitTorrent artefacts identified. The BitTorrent clients BitComet, uTorrent, Azureus, ABC, and BitTornado were chosen to test as these were determined to be the most 'popular' at the time of this study. Each client was analysed with forensic software on generated image files and also *in situ*.

The analysis demonstrated that it was possible to identify files that were currently being downloaded and files currently being shared. It was also possible to identify the amount of data that had been exchanged i.e. uploaded or downloaded for specific files. Some clients produced artefacts that revealed a complete record of the torrent files that had been downloaded and shared. Analysis also revealed that some clients stored the Internet Protocol (IP) addresses of remote computers, with which they had connected when downloading or sharing specific files. The detail and forensic quality of information identified, varied between the clients tested.

Finally the Cyberscrub Privicy Suite software (version 4.5) was found to successfully delete (beyond recovery) most of the BitTorrent artefacts identified. The program is designed to specifically delete 'sensitive' information produced by the clients: BitComet, uTorrent and Azureus.

# INTRODUCTION

## 1.1 What is Bit Torrent and how does it work?

'BitTorrent' is a peer to peer application developed by Bram Cohen in 2001. It uses metadata<sup>1</sup> files known as torrents to implement the downloading of files over the internet from remote computers.

The diagrams shown below are taken from [1] and depict how the BitTorrent protocol works. Firstly, an individual creates a torrent using either a BitTorrent client or torrent making application, and publishes it on a website or forum. This individual is known as the 'initial seeder'. Figure 1 shows the 'initial seeder' distributing fragments of a file to different machines connected using a BitTorrent client. It is usual for the shared file to be virtually split into many smaller chunks of data of equal size to aid file transfer (it is not all ways possible to divide the file equally and therefore the last chunk may be truncated). The connected individuals are collectively known as a 'swarm'.

Fig 1

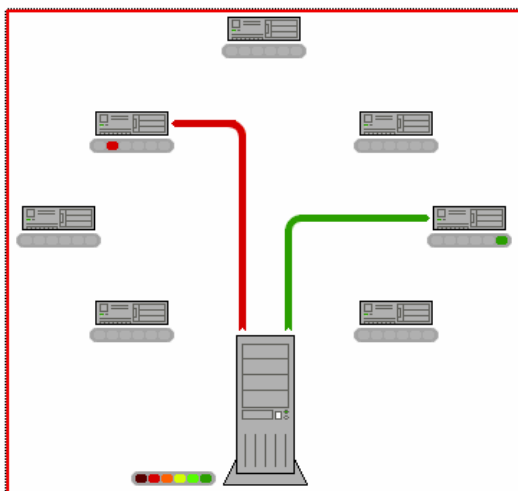
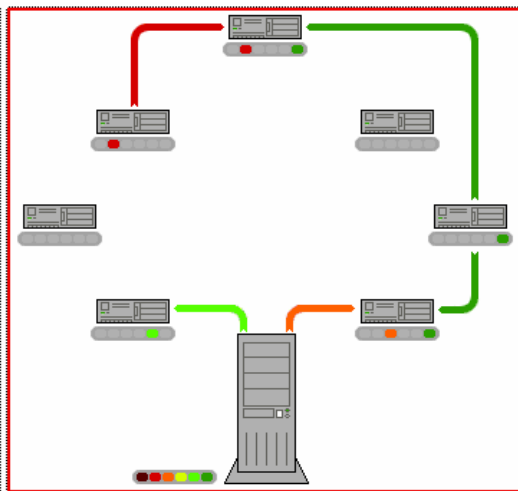


Fig 2



<sup>1</sup> Metadata is data that describes data

Fig 3

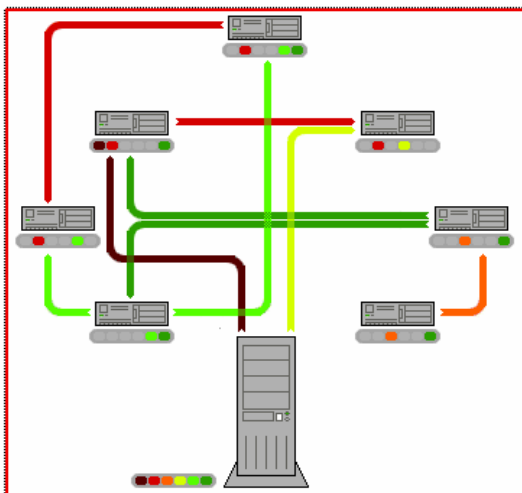
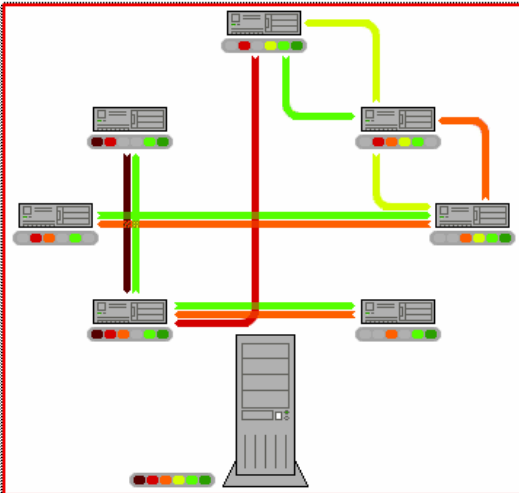


Fig 4



**Figures 1 – 4: The BitTorrent file sharing process**

Figures 2 and 3 show the swarm sharing data chunks between each other as well as the 'initial seeder'. Figure 4 shows the point where the 'initial seeder' has shared all the small data chunks and now no longer needs to seed these files. The individuals that form the swarm now possess the sum of all parts of the file being shared. The swarm will continue to share data with each other and any newly connected individuals. The beauty of this protocol is that an 'initial seeder' only needs to share each data chunk once in order for the file to be shared with many individuals and this means the initial seeder's bandwidth is not being constantly depleted by the people they are sharing files with. A detailed description of the protocol can be found in the paper entitled 'Incentives Build Robustness in BitTorrent' [2].

The torrent file consists of metadata written in bencode (see section 3.2 for a full description), which gives instructions to the BitTorrent client, and facilitates the connection to remote computers and the downloading of files (of any size and type). The instruction information that the torrent files possess are essentially: the name of the file to be shared, the size of each piece and the number of pieces that make up the file, and the Uniform Resource Locator<sup>2</sup> (URL) of a tracker. A 'tracker'

<sup>2</sup> URL is an identifying address for of a web site

is a dedicated server that links all the peers (remotely connected computers) associated with a particular torrent file. Some clients are able to connect to swarms via Distributed Hash Tables. These are databases that store IP<sup>3</sup> addresses and User Datagram Protocol (UDP)<sup>4</sup> port numbers. Each peer (i.e. each client connected) essentially becomes a tracker as they store the contact information of other clients downloading the same torrent file [3]. Remote individuals download the torrent from the website. When the .torrent file is opened within a BitTorrent client, it points the client to the 'initial seeder' using the tracker URL.

The effectiveness of the protocol relies on every individual sharing pieces of the file they are downloading; hence, while an individual is downloading pieces of the file, they are also uploading or seeding the pieces of the file that they already have. It is possible to prevent seeding by changing preferences within the BitTorrent client but trackers and individuals will ban these users or limit their download speed. Thus, it is the general rule that, by using BitTorrent to download files, the user is also sharing files. BitTorrent users have adopted the terms 'seeders' and 'leechers' to refer to computers that are connected to the 'swarm' that have all parts of the file and are constantly uploading to those that do not, respectively.

## **1.2 The BitTorrent Client**

Since Bram Cohen's creation of the Original BitTorrent Client (as it is now known), there have been many variations of BitTorrent clients. As well as having a different Graphical User Interface (GUI)<sup>5</sup>, these clients offer the user varying degrees of download management and the protocol has been tweaked to optimise download and upload speeds. Clients are now available for all common computer operating systems. As of August 2007 there were at least 56 clients. A survey of their capabilities can be found on the Wikipedia [4]. Over the years, programmers have

---

<sup>3</sup> Internet Protocol addresses provide computers unique identifying addresses so computers can communicate on a network

<sup>4</sup> User Datagram Protocol allows programs to send data over a network

<sup>5</sup> Graphical user interfaces are visual displays that allow users to perform functions on a computer by use of a mouse



made BitTorrent clients more user friendly and this has made their popularity as a downloading tool soar. Of particular interest, two studies by CacheLogic [5] followed the trends in peer to peer (P2P)<sup>6</sup> internet traffic and found that, in June 2004, 53% of all P2P traffic was a result of BitTorrent downloading, and roughly 62% of all internet traffic was P2P protocol. In 2005, an updated a more detailed study of global internet traffic showed huge variations in P2P protocol type usage throughout the world. This report showed that BitTorrent internet traffic was slightly behind eDonkey (a detailed graph showing the percentage of world peer to peer traffic is shown in the appendix 1(This graph was taken from the CacheLogic report). In February 2006, the 'Opera' internet browser incorporated the BitTorrent protocol allowing users to download torrent files without the need to install a separate client application. The 'Mozilla Firefox' internet browser also created an extension called 'AllPeers' in August 2006. 'Allpeers' also allows users to download BitTorrent files, and it has the function of allowing users to share files strictly with their friends. The incorporation of BitTorrent into these internet browsers can only increase the popularity of BitTorrent as a tool for file sharing.

Although there are plenty of clients in the wild in actuality only a few are predominantly used. The article 'LimeWire Most Installed P2P Application, BitTorrent Clients Runner up' [6] presents the likelihood of finding Peer to Peer software installed on computers. The authors discuss the findings of a company called PC Pitstop who performed online diagnostic virus scanning for one and a half million computers. In the process, they obtained registry data about the software installed on the scanned computers. The following data refers to the percentage of all computers scanned containing installations of BitTorrent clients: Azureus = 3.2%, uTorrent = 2.7%, BitTorrent = 2.6%, BitComet = 2.0%. While having a piece of software installed does not mean that it is being used, it does give some indication of interest in use at some stage in time. The article mentions that although the Mainline BitTorrent client has a high percentage of installations it is no longer a popular client. Another study entitled 'BitTorrent Client Comparison' [7] compared over 20 clients for nine specific properties that they considered

---

<sup>6</sup> Peer to Peer is the term used to describe a network of computers where data can be shared between computers on the network

essential and came to the conclusion that the clients with the best properties, in descending order, are: uTorrent, Azureus, BitComet and XBT, ABC and BitTornado. The 'net for beginners' website [8], provided a list of the most commonly used BitTorrent clients for sharing 'pirated music'. The list was assembled in May 2007 from 'hundreds of user comments and reader suggestions'. They found that the most commonly used BitTorrent clients (in descending order) were as follows: uTorrent, Azureus, ABC, TurboBT, BitComet, The Original BitTorrent Client.

### **1.3 Legal Issues**

It is not illegal to use any BitTorrent application to share data. However, because the application is extremely efficient at sharing large amounts of data to as many people that can connect to the tracker, and because the internet is unregulated, it is mainly used to share illegal content. Copyrighted material (software, music and films), obscene material (paedophilic, bestiality, bondage), and confidential material can find their way on to torrent sites and be shared with thousands. It only takes a few minutes to create a torrent file and then upload it onto a hosting website. Every country has different laws regarding copyright protection and digital replication of data. BitTorrent file sharing works differently to most other file sharing systems because the torrent files hosted on websites and news groups do not themselves contain any copyrighted material. The following Disclaimer statement was taken from the 'Demonoid' torrent website "Disclaimer: None of the files shown here are actually hosted or transmitted by this server. The links are provided solely by this site's users. The site moderation is also a service provided by the site's users. The administrator of this site (Demonoid.com) cannot be held responsible for what its users post, or any other actions of its users. You may not use this site to distribute or download any material when you do not have the legal rights to do so. It is your own responsibility to adhere to these terms" [9]. The disclaimer makes it harder for authorities to prosecute the owners of BitTorrent websites, but not impossible. One of the most famous cases of law enforcement, leading to BitTorrent site closure, was that of the website [www.elitetorrents.org](http://www.elitetorrents.org). A

torrent file which downloaded the Starwars movie 'Episode III revenge of the Sith' was hosted on the Elite Torrents website weeks before the film was released in cinemas [10]. In 2005, the estimated cost of Internet piracy to the Worldwide Motion Picture Industry was 7.1 Billion Dollars [11]. The huge sums of revenue lost incited the Recording Industry of America (RIAA) and Motion Picture Association of America (MPAA) to mount anti-piracy campaigns and to help law enforcement agencies to find people involved in illegal file sharing in order to shut down the torrent sites. Canada has become a safe haven for file sharers and P2P networks servers since 31<sup>st</sup> March 2004, when the Canadian judge, Konrad von Finkenstein, of the Federal Court of Canada ruled P2P file sharing legal [12]. A consequence of this ruling has been for BitTorrent tracking servers and websites to be hosted by Canadian internet service providers so the owners can evade prosecution.

#### **1.4 Security Issues**

BitTorrent clients must open ports to process the BitTorrent protocol and transfer data between remotely linked computers. This creates a security problem because BitTorrent does not award anonymity to those that use it. The IP address and often port numbers of the users downloading torrents are disclosed to every computer connected in a swarm via their BitTorrent client. This gives hackers effortless intelligence and, indeed, a gateway into insecure systems. The security website Sophos (<http://www.sophos.com>) defined the Azureus, BitComet, BitTornado and uTorrent clients as 'Controlled applications', which they further define as 'a legitimate program but one which Sophos recognizes that some IT administrators might wish to block or authorize, depending on the application's usefulness within a business environment, and its potential impact on business productivity and resources' [13]. BitTorrent downloading can be resource intensive depending on the client used, and it is also bandwidth intensive as speeds of 4Mb of data transfer per second can easily be achieved. Both these factors make BitTorrent file sharing undesirable within a business environment, in addition to the possible legal implications for the company. As stated earlier, P2P protocols make up the majority of internet traffic. File sharers have thus become a liability to the internet service providers' (ISPs) resources. In the past ISPs attempted to throttle

BitTorrent traffic by identifying and restricting the BitTorrent protocol. To combat traffic throttling, the makers of BitTorrent clients incorporated encryption into the latest versions to evade detection. The encryption of BitTorrent protocol traffic prevents the ISP's detecting and filtering BitTorrent traffic.

Potentially any downloaded file could contain malware<sup>7</sup> because downloading is 'blind' and is based on trust; thus, using BitTorrent to download files increases the risk of infecting computers and is a useful system for attackers to deploy malware to thousands of people. In June 2007, the worm 'W32/Impard-A' was the first reported case of any malware actually using a BitTorrent client as a channel for spreading itself. Once a computer is infected the worm tries to spread to other networked computers using the BitTorrent mainline client (if installed) to create and seed a torrent of itself. [14] One of the functions of the worm is to establish a backdoor for an attacker to remotely access and gain control using IRC channels [15].

## 1.5 BitTorrent Erasure

Woodward (2005) produced a paper entitled 'The effectiveness of commercial erasure programs on BitTorrent activity' [16]. In his study, he used the Azureus BitTorrent client to download two files onto a computer with the Windows XP operating system installed. He then ran the commercial erasure programs R-Clean and Wipe, Privacy Suite, and Window Washer separately on imaged copies of the computer. The aim of his study was to evaluate if these software could erase traces of the downloading when the erasing programs were run in 'default mode'. It was concluded that none of these erasure programs removed the torrent files that were used for downloading, nor did they remove the downloaded files. Registry artefacts that were found remained the same for all but the computer wiped using Privacy Suite. His study identified Privacy Suite version 4.0 as being the most effective erasing application tested. A thorough forensic examination of

---

<sup>7</sup> Malware is the term used to describe a computer program that's intent is to penetrate a computer without the owners consent. The function of malware is usually undesired the user

the hard drives was not conducted in this study (i.e. investigation was only carried out on key files and no data analysis of program files or metadata was carried out).

## **Aims of this study**

Given the ever growing use of BitTorrent as a means of file sharing, and the associated costs to media based industries and legal issues, the need for a forensic understanding of this system of file sharing is becoming increasingly important. To date, there are no (known) published studies investigating the forensic aspects of BitTorrent. This study is a preliminary investigation into the forensic artefacts created by BitTorrent use.

The specific aims of the study were as follows:

1. To identify forensic artefacts produced by BitTorrent file sharing, and to establish whether the artefacts lead to identification of the downloaded or shared files.
2. To identify any useful settings that are made by the client configuration file.
3. To identify any artefacts that determine IP addresses of remote computers from which data was downloaded, or shared, during the test phase.
4. To identify whether any of the torrents had been created and seeded by the user.
5. To test whether automated erasing software would delete the BitTorrent artefacts identified.

Five BitTorrent clients were selected for testing as these were determined to be the most 'popular' at the time of this study. Each client was analysed with forensic software on generated image files and also *in situ*. Woodward's study inspired the BitTorrent erasure element to the testing phase of this project. His study defined Privacy Suite version 4.0 to be the most effective erasing application tested and for this reason it was decided to test the most up to date version of CyberScrub Privacy Suite Professional version 4.5 in this project.

# METHOD

## 2.1 Testing

Every client test involved the following procedures:

### The Basic Hard Disk Set Up

- The hard drive was erased using erasure version 5.82 downloaded from <http://www.heidi.ie/eraser>. The erasing setting US DoD 5220.22-M (8-306. /E) with 3 Passes was used to erase data on unused space, files and folders. The EnCase Wipe Drive tool was then used to delete the System Volume information.
- Windows XP Professional with SP1 was installed on the hard drive.
- A BitTorrent client was installed on to the hard drive from an USB storage device using the .exe file downloaded from the following sites:

File	Website Downloaded From
BitComet_0.89_setup.exe	<a href="http://www.download.com/Bitcomet">http://www.download.com/Bitcomet</a>
utorrent 1.6.1.exe	<a href="http://www.utorrent.com/download">http://www.utorrent.com/download</a>
Azureus_2.5.0.4a_Win32.setup.exe	<a href="http://azureus.sourceforge.net/download">http://azureus.sourceforge.net/download</a>
ABC-win32-v3.1.exe	<a href="http://pingpong-abc.sourceforge.net/download">http://pingpong-abc.sourceforge.net/download</a>
BitTornado-0.3.18-w32install.exe	<a href="http://www.Bittornado.com/download">http://www.Bittornado.com/download</a>

Five test disks were constructed:

Test1 = Bitcomet

Test2 = uTorrent

Test3 = Azureus

Test4 = ABC

Test5 = BitTornado

### Testing Part 1 (Downloading Torrents)

- Microsoft Internet Explorer was used to access the website 'http://linuxtracker.org' to download torrents from. This is a legal torrent website containing only open source Linux software.
- The download process comprises of double clicking the mouse cursor on a selected torrent file on the website. The Windows download manager then

gives the option to save this file or open it. Selecting open will automatically start the program associated with the .torrent file extension and the torrent will then be opened within the application (BitTorrent client). Selecting save will save the torrent file to a predefined location (the default is the desktop).

- A number of torrents were selected for download and then different scenarios created to emulate normal usage, such as stopping a torrent during the download, removing a torrent from the client during download, completing a full download and letting the torrent seed.
- An EnCase image of each test disk was taken after this part of the test phase had completed.

### **Testing Part 2 (Creating and Uploading Torrents)**

- Each client (except BitTornado) contains a button or option called make/create a .torrent file. Once selected a GUI appears requiring the input of at least these fields: Source file (the file or folder path of the file(s) that going to be shared), Tracker (the URL address of the tracker that will be used) output file (the destination that the .torrent file is going to be saved which is usually the same destination as the source). A screen shot of the Bitcomet Torrent maker is shown in appendix 2.
- The source file used was a small movie clip I created and thus it contained no copyright and was legal to share.
- Once the torrent was created it was uploaded on to a web site and opened within the client.
- The torrent file was left to seed until it reached 100% upload. During the process screenshots were taken to record peers connected with and any other
- An EnCase image of each test disk was taken after a torrent was uploaded for each client.

### **Testing Part 3 (Erasure of BitTorrent Activity)**

A trial version of CyberScrub Privacy Suite Professional version 4.5 was downloaded from <http://download.com/cyberscrub-privacy-suite->

professional.4.5/3001-2144-4-10654405.htm. The file psuite0992.exe was installed with default settings. This trial version lasts for 15 days and has the same capability as the fully registered version. CyberScrub claim this latest version supports the erasure of sensitive data from more applications, one of which is uTorrent. The full list of features for this product can be found on their website [17].

- The privacy guard option was selected. This has two further options basic and advanced. Selecting the basic options runs preconfigured defaults automatically where selecting advanced options let you select peer2peer application activity to erase as well as the preconfigured defaults. The advanced option was selected each time along with the BitTorrent application relating to the test disk, however this application only has the option to delete Azureus, BitComet, and uTorrent v1.5.
- This software also has the capability of 'scrambling' folder and file name properties, and within preferences you can create a log file of the erasing process.
- An EnCase image of each test disk was taken after the Privacy Guard erasing process had completed.

Throughout the testing procedure a contemporaneous notes were taken providing a step by step guide of the testing performed for each client. This was used for reference during analysis. A simplified version of note book containing the steps taken during testing i.e. the order of downloading and uploading torrent files along with dates and times can be found within the appendix 3. It is advised to refer to the relevant section of the simplified notes before reading each client analysis.

## **2.2 Imaging**

An image was made after each stage of the testing i.e. Testing part 1, part 2 and part 3. Write blocking was performed by insertion of 'FireFly™ IDE' write block hardware device between the IDE cable, and test hard disk. EnCase 4.2 software was used to image the hard disk. The image files were labelled according to the



test client and part i.e. 'Test1\_1' identifies the BitComet - Downloading Torrents Test.

# ANALYSIS

## 3.1 Analysis Methodology

Encase v4.2 was used to analyse the image files made. Registry files (Software and NTUSER.DAT) were copied from the images using the EnCase 'Copy/UnErase' function and analysed using the program MiTec Windows Registry Recovery v 1.3.1.0 (Downloaded from [www.mitec.cz](http://www.mitec.cz)). Searches within the registry hives were conducted for key words such as: Torrent, Client name, BT. The results were analysed and the meanings of the registry keys were looked up from the document entitled 'Registry Quick Find Chart' by Access Data [18].

Encase analysis: The test images were opened, verified, hashed and NSRL Hash sets RDS\_216 (B – D) were imported and hash library Rebuilt. Compressed files were manually searched in areas where client install files were found. If compressed files were found then they were mounted using the EnCase 'View File Structure' function.

Searches were conducted for all the IP addresses that were identified during downloading and uploading torrents. These searches were conducted over all files, files slack and the unused portions of hard drive.

Each client like most applications has preferences. Some of these settings can be of forensic interest e.g. downloads and .torrent files can be saved to directories of the users choosing. To obtain the functions of settings found within each client a copy of the configuration file was made and then for each setting altered or viewed, the copy file was then used to reference any changes. Some of the settings needed a torrent to be downloaded in order to check there functionality and so the site [linuxtracker.org](http://linuxtracker.org) was used. The methodology for the testing required changing, and checking, each preference one by one.

## 3.2 Torrent file analysis

Torrent files are written in 'Bencode' and therefore follow the standard format of Strings, Integers, Lists and Dictionaries which are explained below:

- Strings are length-prefixed base ten followed by a colon and the string. For example 4:spam corresponds to 'spam'.
- Integers are represented by an 'i' followed by the number in base 10 followed by an 'e'. For example i3e corresponds to 3 and i-3e corresponds to -3. Integers have no size limitation. i-0e is invalid. All encodings with a leading zero, such as i03e, are invalid, other than i0e, which of course corresponds to 0.
- Lists are encoded as an 'l' followed by their elements (also bencoded) followed by an 'e'. For example l4:spam4:eggse corresponds to ['spam', 'eggs'].
- Dictionaries are encoded as a 'd' followed by a list of alternating keys and their corresponding values followed by an 'e'. For example, d3:cow3:moo4:spam4:eggse corresponds to {'cow': 'moo', 'spam': 'eggs'} and d4:spaml1:a1:bee corresponds to {'spam': ['a', 'b']}. Keys must be strings and appear in sorted order (sorted as raw strings, not alphanumerics).

The above explanation of bencode was taken verbatim from the article entitled 'Protocol Specification' [19].

A typical torrent file consists of the following Metainfo: all torrents contain a tracking address, name(s) of the file(s) that the torrent saves, the total size of the shared file (measured in bytes), the size for each piece the file is split in to (measured in bytes), and then the number of pieces that make up the file. Lastly the torrent contains pieces data, which consists of a string of SHA1 hashes of length 20, corresponding to each piece of data that the file is split in to.

Torrents can also contain extra data such as the date the torrent was created, the program the torrent was created by and also a comment from the creator of the torrent. All of which can be useful forensic information.

An example of the bencode data contained in a torrent file is shown in appendix 4, (minus the complete string of sha1 hashes). A table dissecting and explaining each bit of code is also shown.

# BitComet (0.89)

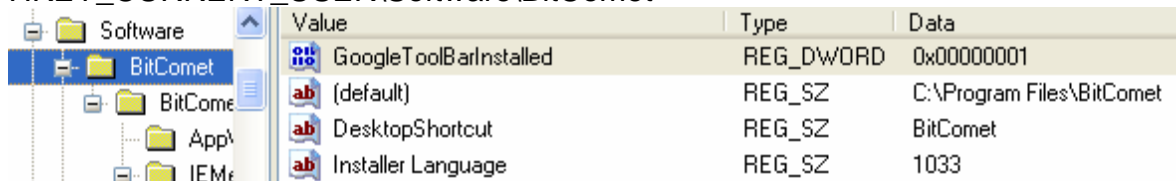
The BitComet client has a preferences tab that allows the user to configure the client to suit their downloading requirements. These settings are stored within a file called 'BitComet.xml' located in the path C:\Program Files\BitComet\BitComet.xml. When the client is freshly installed, the 'BitComet.xml' file does not contain all the feasible settings. Only when the user views the settings within preference tab do the settings then become saved in the BitComet.xml file. Configurations of interest within the 'BitComet.xml' file are shown in appendix 5.

## 3.3.1 Test 1\_1 analysis

### Registry Analysis

The NTUSER.DAT file was extracted from the path C:\Documents and Settings\test\_1\NTUSER.DAT using EnCase. This file was imported in to the MiTec Windows Registry tool and the key word Torrent was searched. The following registry information was identified which could be of interest during a forensic investigation.

HKEY\_CURRENT\_USER\Software\BitComet



Value	Type	Data
GoogleToolBarInstalled	REG_DWORD	0x00000001
(default)	REG_SZ	C:\Program Files\BitComet
DesktopShortcut	REG_SZ	BitComet
Installer Language	REG_SZ	1033

This registry key shows, the 'Bitcomet' software is installed on the computer and is located in the path C:\Program Files\ BitComet. This key also reveals that a BitComet shortcut has been installed to the desk top.

HKEY\_CURRENT\_USER\Software\BitComet\BitComet

Value	Type	Data
CaptureIEDownload	REG_DWORD	0x00000001
IEMonitorFileExt	REG_SZ	.zip;.rar;.iso;.exe;.asf;.avi;.mp3
DefaultDownloadManager	REG_DWORD	0x00000001
IELinkTitle	REG_BINARY	00 00 00 00
IERefUrl	REG_BINARY	68 74 74 70 3A 2F 2F 6C 69 66
IEHtmlText	REG_BINARY	00 00 00 00 00 00 00 00 00 00
IEWebpageTitle	REG_BINARY	4C 00 69 00 6E 00 75 00 78 00

Some of the data shown in the BitComet sub key is in binary form. When converted in to ASCII the following data is revealed:

IERefUrl = <http://linuxtracker.org/torrents-details.php?id=4149>

IEWebpageTitle = LinuxTracker : : Details for torrent "BeleniX 0.6 Live CD"

The Bitcomet sub key contains a record of the website URL, where the last torrent was downloaded and opened from, and the 'title' of the web page (as seen in the source code of the website).

HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.torrent

Value	Type	Data
a	REG_SZ	BitComet.exe
MRUList	REG_SZ	ba
b	REG_SZ	iexplore.exe

This shows that BitComet.exe is the default program used by windows to open files with the extension .torrent.

HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.torrent

Value	Type	Data
0	REG_BINARY	66 00 6F 00 72 00 65
MRUListEx	REG_BINARY	00 00 00 00 FF FF FF

This registry entry shows the files in the recent documents list. This is the location that is accessed from the windows start tab/my recent documents. The binary data translates into the following ASCII information:

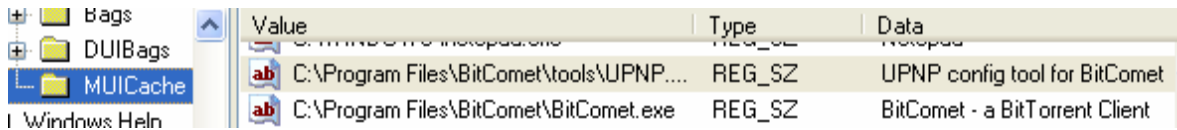
foresight-10710-foresight-1[1].3-x86-dvd1.iso.torrent

foresight-10710-foresight-1[1].3-x86-dvd1.iso.lnk

As explained in the Azureus analysis section 3.5.2, when a user downloads and saves torrent from files the internet, link files are created and thus an entry within

the 'RecentDocs' registry sub key is created by Windows. In this case the 'foresight-10710-foresight-1[1].3-x86-dvd1.iso.torrent', was saved to the desk top.

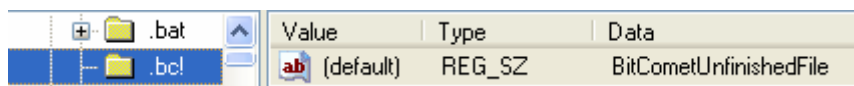
HKEY\_CURRENT\_USER\Software\Microsoft\Windows\ShellNoRoam\MUICache



Entries in this key refer to those programs that have been opened. This entry shows that the BitComet application has been run and the BitComet 'UPNP configuration tool'.

The SOFTWARE registry hive was extracted and the contents of the registry examined. The full path of the SOFTWARE registry hive is C:\WINDOWS\system32\config\software.

HKEY\_LOCAL\_MACHINE\SOFTWARE\Classes\bc!

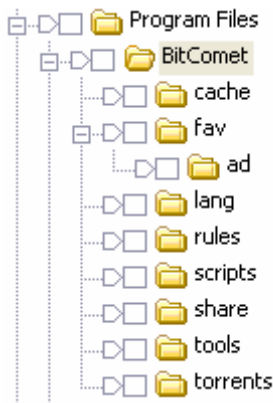


This shows that the '.bc!' file extension is recognized as an incompletely downloaded file associated with the BitComet client.

Sixty one hits were retrieved when the SOFTWARE registry hive was searched with the term 'comet'. These hits referenced similar data to that retrieved within the NTUSER.dat registry hive, thus they did not provide any additional forensic evidence.

## Encase Analysis

The Folders shown below are created upon installing and using the BitComet client.



The 'Bitcomet' folder located in the path 'C:\Program Files\BitComet\' contains a file called 'Downloads.xml'. This file contains information about the states of downloading files. Explanations of the xml code found within the 'Downloads.xml' file are given in appendix 6. Analysis revealed that the 'Download.xml' file contains no information about downloads where the torrent files had been removed from the client (unless the torrent was sent to recycle bin); thus, this file stores information only about downloads the client is currently processing. The file lists information about torrents in the sequence they were opened within the client.

The BitComet GUI has a tab called 'Peers', which displays information about the peers that are currently connected and exchanging data for each torrent currently loaded. Screen shots of the peer information were taken while downloading different torrent files. IP addresses were identified from the screenshots and searched for within all files, file slack, and unallocated clusters of the hard drive. The search identified a file called 'belenix0.6.iso.xml' found in the location C:\Program Files\BitComet\torrents\belenix0.6.iso.xml. The folder 'torrents' contains both '.torrent' files and '.xml' files with the same name. The '.xml' files contain data about the download of the '.torrent' file. These '.xml' files contain IP addresses of connecting or connected peers at the time the torrent was stopped or the BitComet client was stopped. This conclusion was made from further testing (Torrent files were downloaded and opened in BitComet. Screenshots of connected peers were taken as the torrents were stopped. The IP addresses from the screenshots were checked against the IP addressed found in the corresponding '.xml' files).

Appendix 7, provides explanations of the code found within the '.xml' files. Note that the 'torrents' directory only contains files that BitComet is currently processing i.e. The .torrent files in this directory are those currently loaded in BitComet and the .xml files record the 'state' of the downloads for the torrent files they pertain to.

The 'share' directory of location C:\Program Files\BitComet\share\, contains a file called 'my\_shares.xml'. This file contains data of the files that have been selected to share. When a torrent file is opened in BitComet the following GUI named 'Task Properties' appears (refer to appendix 8). Selection of the tick box located on the bottom left, is required for the files to be added to the 'my shares' category in the BitComet application, and hence the 'my\_shares.xml' file. The file lists data for every torrent file shared. The file information includes: an 'infohash' value of the torrent file, the size of the file that the torrent downloads and the name of the file that the torrent downloads. Data entries within the 'my\_shares.xml' file are stored for every torrent file opened and shared unless the corresponding torrent files are deleted from BitComet using 'Delete Task and Downloaded Files' option.

### **3.3.2 Test 1\_2 analysis**

The 'flip1.mpg.xml' (located in the directory C:\Program Files\BitComet\torrents\), contains data that demonstrates the torrent 'flip1.mpg.torrent' has been opened within BitComet and is an 'initial seeder'. The distinguishing statements that identify this torrent as the 'initial seeder' and not one used to download files are: <FileList BaseName="flip1.mpg", DataDownload="0", DataUpload="10430464", ElapsedTimeDI="0". These statements declare that no data has been downloaded for the file flip1.mpg and no time was spent downloading the file however 9.9 Mb of this file has been uploaded. This can only occur if the torrent is seeding a file that is already on the computer. Note the following data <TorrentInfo CreateDate="2007-07-01 13:02:29.583897" and FinishDate="2007-07-01 13:02:29.583897" show the torrent was started and finished at the same time. This is because the client records the time when a downloaded file is complete, and it is complete because the file is already located on the computer.



No peer information was found within the xml file 'flip1.mpg.xml' indicating no peers were connected when BitComet was closed. A search for the term '<Peer downloaded=' was conducted returning one 'hit' within the unallocated cluster area of the hard drive. This hit showed a cached copy of the 'flip1.mpg.xml' file. When analysed, the data showed an IP address of a connected peer downloading data from the flip1.mpg file.

### 3.3.3 Test 1\_3 analysis

The use of the Cyber scrub has eliminated the following keys from the 'NTUSER.dat' registry hive which contain BitTorrent data:

1. HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedMRU
2. HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU
3. HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.torrent

The deletion of these registry keys removes the torrent link file information, as well as the names and paths of files most recently saved or copied, and most recently opened. The erasure program did not delete the registry key HKEY\_CURRENT\_USER\Software\BitComet\BitComet, which as explained earlier contains a record of the website URL, where the last torrent was downloaded and opened from, and the 'title' of the web page (as seen in the source code of the website). Registry data still, also shows that BitComet is installed and is the default program for opening '.torrent' files.

The following files from the location C:\Program files\BitComet\, were deleted and over written: BitComet.xml, Downloads.xml. The .torrent and <torrent name>.xml files stored in the location C:\Program files\BitComet\torrent\ were also deleted and overwritten. All the .torrent files that were saved within the temporary internet folder were deleted during the privacy guard advanced erasing procedure. The file

names for all deleted files were unreadable as the 'scrambled' setting was selected in the erasing program.

The only file that wasn't erased which contains some history of downloading is the 'my\_shares.xml' file. As already discussed this file contains a list of files that have at one stage been shared.

The following 'Key words' were searched within EnCase to try and find artefacts of BitTorrent downloading activity:

<ListenPort>	found within the 'BitComet.xml' file
DataDownload=	found within the 'Downloads.xml' file
<FileList BaseName=	found within the '<torrent name>.xml' file

I chose these 'key words' as they are unusual strings that I would expect few hits to occur. The search term DataDownload=" found a version of the downloads.xml file within unallocated clusters. The information within piece of text was not the most recent copy of the Downloads.xml file as it contained different download and upload data but it did give an accurate history of download events. The other searches didn't amount to any significant hits.

# uTorrent (1.6.1)

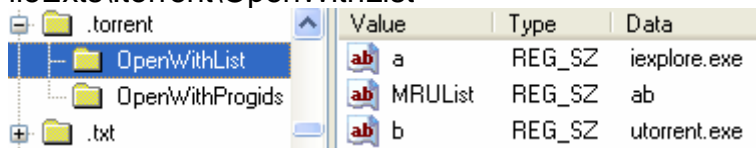
The uTorrent client is a stand alone client meaning it does not pre-install program files before it runs; however it still caches data when running; within the path C:\Documents and Settings\\Application Data\uTorrent\. The client is small in size so can easily be run from a removable storage device, which means it can bypass basic security settings as the program doesn't actually install itself on the computer. The uTorrent client has a range of optional settings that can be configured by the user. These settings are stored within the file called 'settings.dat'. The 'settings.dat' file by default does not contain all the possible settings. Only when the user views the settings within the client do the settings then become saved within the 'settings.dat' file. The 'settings.dat' file is written in bencode. The configurations of forensic interest within the file are shown in appendix 5.

## 3.4.1 Test 2\_1 analysis

### Registry analysis

The NTUSER.DAT registry hive was analysed to find references to torrent activity. The full registry path, screen shots of the registry data, and explanations of these registry keys, are shown below:

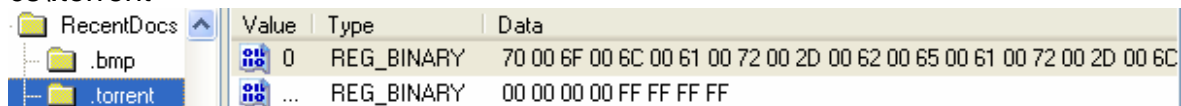
HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.torrent\OpenWithList



Value	Type	Data
a	REG_SZ	iexplore.exe
MRUList	REG_SZ	ab
b	REG_SZ	utorrent.exe

This shows that utorrent.exe is the default program used by windows to open files with the extension .torrent

HKEY\_LOCAL\_MACHINE\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.torrent



Value	Type	Data
0	REG_BINARY	70 00 6F 00 6C 00 61 00 72 00 2D 00 62 00 65 00 61 00 72 00 2D 00 6C
...	REG_BINARY	00 00 00 00 FF FF FF FF

This registry entry shows the files in the recent documents list. This is the location that is accessed from the windows start tab/my recent documents. Translating the binary data into ASCII information shows the following:

polar-bear-linux-ALPHA2[1].iso.torrent  
polar-bear-linux-ALPHA2[1].iso.lnk

Recent documents are usually linked with files that have been recently opened within windows; however in the case of torrent files, linked entries can occur by downloading and saving torrent files from files the internet. The registry data shows that the 'polar-bear-linux-ALPHA2[1].iso.torrent' was saved to, or opened from the desk top. Further evidence would need to be acquired to definitively say either way.

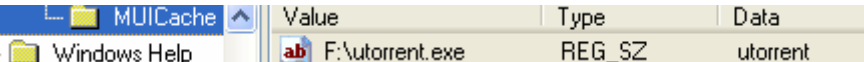
HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU\torrent



Value	Ty...	Data
ab a	R...	C:\Documents and Settings\test_2\Desktop\polar-bear-linux-ALPHA2[1].iso.torrent
ab	R...	a

Entries within this key refer to the names and paths of files most recently saved or copied. The data therefore shows that the torrent file 'polar-bear-linux-ALPHA2[1].iso.torrent' has been saved to the desktop.

HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\ShellNoRoam\MUICache



Value	Type	Data
ab F:\utorrent.exe	REG_SZ	utorrent

Entries within this key refer to those programs that have been opened. Hence this shows that that the uTorrent application has been run.

## EnCase Analysis

The file 'resume.dat' (located in the directory C:\Documents and Settings\\Application Data\utorrent\), contains information that uTorrent uses to resume the application from the point of closure. This file contains

bencode of which definitions are shown in the appendix 6. Note the data pertaining to each torrent recorded in the 'resume.dat' file is defined by the file path of a torrent and the end bencode of '5:wastei0ee'. The data in this file only pertains to those torrent files currently loaded in the uTorrent client.

The uTorrent GUI has a tab called 'Peers', which displays information about the peers that are currently connected and exchanging data for each torrent currently loaded. Screen shots of the peer information were taken while downloading different torrent files. IP addresses were identified from the screenshots and searched for within all files, file slack, and unallocated clusters of the hard drive. The search did not identify any of the known IP addresses of remote computers that were sharing files. Further analysis provided the clues to finding stored peer information. The 'resume.dat' file contains the bencode string '5:peers#:<string of characters>' ( where # = base ten number defining the length of the next string of characters). The string '5:peers42:ø·UZ-·`CRi·ÚsxY¥/^D±U\*uR.ÉB¾·+S·QL4·Î Xá·' was identified from the code relating to the torrent 'Super\_Grub\_Disk\_0.9598.iso.torrent'. When highlighted and bookmarked in EnCase the string can be converted in to 8-Bit Integer. The decoded string is shown in appendix 7. The known IP addresses, that were connected with uTorrent during the downloading of the file 'Super\_Grub\_Disk\_0.9598.iso' were compared to the UInt8 column. It was noticed that the known IP addresses matched the numbers highlighted in yellow. The IP addresses are read in reverse order from that listed and are spaced by two lots of numbers. Therefore the results of the decoded string are as follows:

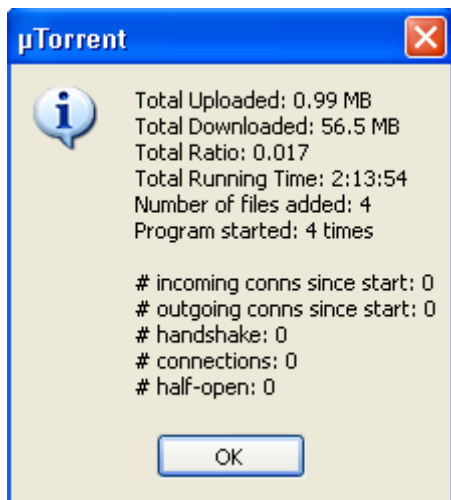
IP address of connected peers	Numbers separating IP addresses
85.22.25.162	2790
82.67.96.45	26239
89.215.115.218	47165
85.177.68.94	11742
66.201.46.82	24190
81.1. 83. 43	5276
88. 160. 206.4	26225

The top three IP addresses in the table above are the known IP's that were identified during testing. The other four could be those connected before or after the screen shot was taken. The numbers separating the IP addresses could be the port numbers set for BitTorrent file transfer. This would need to be tested by scanning network activity while downloading.

uTorrent stores seven files within the windows operating system file 'Application Data', which are used to resume all previous settings and downloads. These files are now listed: dht.dat, settings.dat, rss.dat, resume.dat dht.dat.old, settings.dat.old and resume.dat.old. These files along with all the torrent files found in the location C:\Documents and Settings\\Application Data\uTorrent\. Were copied from the image Test2\_1 using EnCase and were imported into the application data folder within a testing machine. Opening the uTorrent application (offline) with the imported files, emulated the state of uTorrent when it was last closed. The screen shot in appendix 8, shows the emulated state of test 2\_1 when uTorrent was closed.

This method of emulating uTorrent in it's last closure state, provides a quick way of extracting the following data: The time and date the torrent file was opened, the time and date the torrent completed the download process, The volume of data that has been uploaded and downloaded and the total size of the file, the trackers being used by the torrent, the name and location of the file being downloaded, the time elapsed in hours, minutes and seconds, since the torrent was first opened. Comparing the data recorded in this screen shot with that taken in the test notes shows that all the data is the same.

Selecting the 'help' tab in uTorrent and the sub category 'statistics' will produce a GUI that displays useful statistical data regarding the use of the uTorrent application. An example of the GUI is shown below:



Note that the program started = 4 this includes the current opening i.e. in the program was opened 3 times and this is the fourth. The 'Total Running Time' identifies the aggregate time the application has been opened. The time stated includes the time since it was reopened (in the emulation), thus to get a realistic time this tab should be checked from when opening uTorrent with the inserted files. The aggregate

amount of data transferral through downloading and uploading is also shown in this statistics GUI.

### 3.4.2 Test 2\_2 analysis

The use of the 'Create New Torrent' function has resulted in a change to the 'setting.dat' file to include the bencode list: '7:ct\_histl57:C:\Documents and Settings\test\_2\Desktop\box to floor.mpg'. This bencode identifies the source file(s) of newly created torrents. I.e. the 'Create New Torrent' function has been used to create torrents that are able to share the source file(s) listed. Note the actual torrent name and saved location is not recorded. Also this artefact does not require the torrent to have been opened and uploaded. Additional testing and further creation of torrents resulted in each one listed one after the other within the 'setting.dat' file e.g. '7:ct\_histl#:<1<sup>st</sup> created torrent>#:<2<sup>nd</sup> created torrent>#:etc (# represents a base ten number).

### 3.4.3 Test 2\_3 analysis

I analysed the image for indications of erasure of BitTorrent activity. I located the cybscrub.log in the path C:\Documents and Settings\test\_2\Application Data\CyberScrub\Privacy Suite\cybscrub.log. This log file was created by the CyberScrub Privacy Suite program as a result of changing the preference settings (i.e. log files do not appear by default). The file path above is the default location

where log files are saved but it should be noted that this file can be saved to any location. There were two files of this name, of which one was deleted.

Appendix 9, shows a samples of the data from the deleted log file. This log file shows the total amount of erased data, the time and date the data was erased. In relation to uTorrent the log file describes the deletion of all the .torrent files found within the directory: C:\Documents and Settings\\Application Data\uTorrent. Within the same directory these files are deleted: settings.dat, settings.dat.old resume.dat, resume.dat.old. These .dat files and the .torrent files are the crucial files containing data relating to the events of BitTorrent file sharing with uTorrent.

The use of the CyberScrub Privacy Suite has eliminated the following keys from the registry:

1. HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU
2. HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.torrent

The deletion of these registry keys removes the torrent link file information, as well as the names and paths of files most recently saved or copied. The registry data still, however, shows uTorrent has been opened and it is the default program for opening '.torrent' files.

The CyberScrub Privacy Suite application has deleted and overwritten all the '.torrent' files from within the 'temporary internet folder' during the privacy guard advanced erasing procedure.

The 'utorrent.def' file found in the path C:\Documents and Settings\test\_2\Application Data\CyberScrub\Privacy Suite\Deffiles\utorrent.def, contains the definitions for the application to use when deleting uTorrent data. Of particular interest is the following data from this script file:

```
DEFVAR CHK1="set"  
DEFVAR CHK2="set"
```



When these variables are equal to "set", this means the settings within the privacy guard advanced erasing procedure for uTorrent have been activated.

The settings that have been activated are 'wipe information about file transfers' and 'Wipe the uTorrent settings'. Activating these settings doesn't mean this script has been run; but when the privacy guard advanced erasing procedure is activated the script will delete the files previously discussed. Therefore, the presence of 'DEFVAR CHK1="set" DEFVAR CHK2="set"' implies that the user has or intends to use this feature of the CyberScrub Privacy Suite.

### **3.4.4 Test 2\_4 analysis**

The torrent file uploaded in this test was called 'boxs floorss.mpg.torrent'. It took a few attempts at creating and loading torrents within uTorrent before the file initially started seeding and other peers could connect. The evidence from the 'resume.dat' file shows, that the torrent was added on 30/06/07 22:34:23 and completed on 30/06/07 22:34:24 (d8:added\_oni1183239263e, 12:completed\_oni1183239264e). When a torrent is added to the client it interprets the torrent code and inspects the location defined for saving the file pieces. The client does this to identify which pieces of data are already present so it can then coordinate the download remaining files. The torrent is labelled complete when all file pieces listed in the torrent file are identified within the saved directory. Hence the one second gap between the added and completed status, is the time uTorrent took to compute that the file parts were stored on the computer. The torrent had not downloaded any bytes but had uploaded 2.84 Mb (10:downloadedi0e8:uploadedi2975744e). This is evidence that the file must already be present on the computer and is being shared. It is not possible to share data that hasn't already been downloaded unless you already have the data. Further evidence reinforcing this torrent file is an 'initial seeder' is shown by the time the torrent has been running equalling the time it has been seeding (7:runtimei2538e, 8:seedtimei2538e). This file provides evidence of the exact state of downloading and uploading of files but only of those torrents loaded in to uTorrent and not any previous file sharing activity.

During testing the 'Clear Private Data' tab (found within the preferences and sub category other), was pressed. This resulted in the list of created torrents being removed from the settings.dat file. A total of three torrents had been created and none were found listed in the settings.dat file. Artefacts identifying the use of the 'Clear Private Data' tab were not found, which means if it has been used evidence of torrent creation will be unobtainable. The key word '7:ct\_histl' was used to search for cached versions of this file. One search hit identified the keyword within the unallocated clusters area of the hard drive however it did not reference any of the created torrents within the test. This search result means it may be possible to find evidence of torrent creation.

## Azureus (3.0.1.2)

The Azureus client has a range of optional settings that can be configured by the user. These settings are stored within the file called 'azureus CONFIG', full path C:\Documents and Settings\

### 3.5.1 Test 3\_1 analysis

#### Registry analysis

The NTUSER.DAT registry hive was analysed to find references to torrent activity. The full registry path, screen shots of the registry data, and explanations of these registry keys, are shown below:

HKEY\_CURRENT\_USER\Software\Azureus



Value	Type	Data
ab (default)	REG_SZ	C:\Program Files\Azureus

The Azureus program installs a registry key in the software sub-hive. It contains data regarding the path where the software was installed. In this case, it is the default pathway 'C:\Program Files\Azureus'.

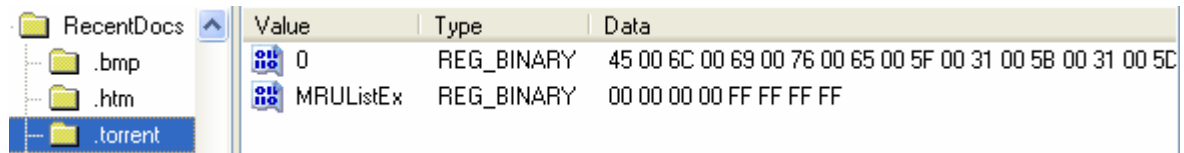
HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\FIleExts\\*.torrent\OpenWithList



Value	Type	Data
a	REG_SZ	iexplore.exe
MRUList	REG_SZ	ab
b	REG_SZ	Azureus.exe

This shows that Azureus.exe is the default program used by windows to open files with the extension .torrent.

HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.torrent

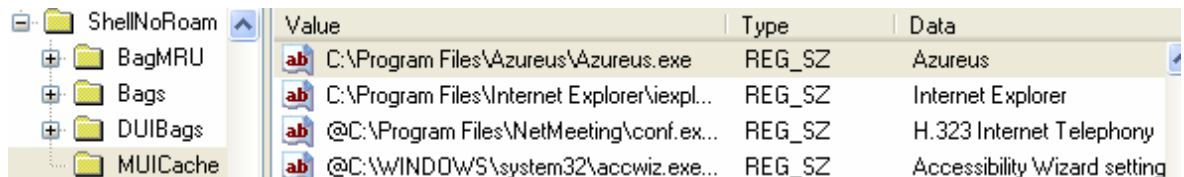


This registry entry shows the files in the recent documents list. This is the location that is accessed from the windows start tab/my recent documents. The data stored in this key is in binary form. The ASCII translation provided the following information:

Elive\_1[1].0\_Gem\_(stable).iso.torrent  
Elive\_1[1].0\_Gem\_(stable).iso.Ink

This result will be discussed in the section 3.5.2.

HKEY\_CURRENT\_USER\Microsoft\Windows\ShellNoRoam\MUICache



Entries in the MUICache key refer to those programs that have been opened. The entry is produced by internet explorer, and shows that the Azureus application was opened.

## Encase Analysis

The file 'downloads.config', located in C:\Documents and Settings\test\_3\Application Data\Azureus\downloads.config, contains bencode information about the torrent files that have been opened within the Azureus client. Appendix 15, provides an example of the content of the 'downloads.config' file for one torrent only. An explanation of this code is also provided. Each torrent in the 'downloads.config' file is defined by the bencode of 'd9:allocatedi1e' at the

beginning and '7:uploads4ee' at the end. Note if the 'secondsDownloading' integer value is equal to zero and the 'secondsOnlySeeding' integer value is equal to one or more, then the .torrent has been uploaded by the user. The downloaded integer value will be the size in bytes of the file and the uploaded integer value will be the amount of bytes shared with externally connected peers.

The 'downloads.config' file only contains information regarding torrents that are currently loaded within the client. The torrent can be of any status (i.e. paused, stopped, downloading or uploading); however, when the torrent is removed from Azureus by the user, the data regarding the torrent is also removed from this file. This was evident because the 'downloads.config' file in 'test 1' and 'test 2' only contained torrents that were not removed.

The Azureus GUI has a tab called 'My Torrents', which displays the files that are downloading. Further tabs are produced by double clicking the file downloading. One of the tabs is called 'Peers'. The 'Peers' tab shows various information about the peers that are currently within the swarm for the corresponding torrent file. Screen shots of the information in the Peers tab were taken while downloading different torrent files. IP addresses were identified from the screenshots and searched for within all files, file slack, and unallocated clusters of the hard drive. The IP addresses were searched to find out if Azureus records information regarding, connected peers and any data exchange (uploaded or downloaded) with the peers. The search identified hits within the following files:

C:\Documents and Settings\test\_3\Application Data\Azureus\tmp\AZU32759.tmp

C:\Documents and Settings\test\_3\My Documents\az.log

C:\Documents and Settings\test\_3\Application Data\Azureus\active\B3D58DC8E134104063989DA64DBAE6CD5A4E59F3.dat

C:\Documents and Settings\test\_3\Application Data\Azureus\active\F04F899D5C32918055C73704394B9FF229889F21.dat

C:\Documents and Settings\test\_3\Application Data\Azureus\active\B3D58DC8E134104063989DA64DBAE6CD5A4E59F3.dat.bak

C:\Documents and Settings\test\_3\Application Data\Azureus\active\F04F899D5C32918055C73704394B9FF229889F21.dat.bak

## C:\Unallocated Clusters

Each .dat file contains a combination of data from a .torrent file and from the cache.dat file (the cache.dat file is explained later in this section). The IP addresses that the client was connected to, whilst the download or upload of data occurred, are listed at the end of the torrent data. Data for each IP connection is defined by the bencode 'd2:ip' followed by the IP string and port data e.g. d2:ip11:**192.168.0.14**:port**6881**e. No information on the amount or direction of data transfer was recorded in these files.

The files that contain the IP addresses of connected peers from external computers are all found in the 'active' folder and the 'tmp' folder. The files in the 'tmp' folder are temporary files which only contain information for the period that Azureus was last opened and closed. The active folder contains only those torrents that are still active (downloading or seeding). Any information about previous downloads with peer connectivity is unobtainable unless it can be found within the unallocated cluster area of the hard disk. The search term 'd2:ip' was used to obtain hits of previous downloaded files with IP data. The search produced five hits within unallocated clusters, which relate to the downloading of the file manifestdestiny-respin2-binary-dvd-i386.iso. During the test phase, this file was partially downloaded and then the torrent removed from Azureus. This is why the .dat file for this torrent was not found within the Azureus active folder of the imaged Test 3\_1 disk.

The file 'cache.dat', full path C:\Documents and Settings\test\_3\Application Data\Azureus\active\cache.dat, is a compressed file type. It contains information about all the torrents that are active (open within Azureus). When mounted in EnCase, the 'dat' file information is displayed. The 'cache.dat' file is written in bencode. Explanations of the code from this file are shown in the appendix 16. Note each torrent described in the cache is defined by the start bencode of d10:attributes.

The folder 'tmp' (full path C:\Documents and Settings\test\_3\Application Data\Azureus\tmp\) contains six .tmp files, three of which contain data about the history of downloading. The names of the files begin with 'AZU', which is followed by a string of numbers that change for each session the application is used (as does the data contained within the temp files). One of the files contains the time that the torrents were opened within Azureus. The times are listed in chronological order and they also reference the title of the file being downloaded. Below is the data from this file:

[11:28:36] Tracking geexbox-1.1-rc4-en.i386.iso  
[12:40:21] Tracking UBUNTU  
[12:49:28] Tracking bluewhite64-12.0-rc1  
[13:14:40] Tracking manifestdestiny-respin2-binary-dvd-i386.iso

Note that this file contains a list of files corresponding to opened torrents but only those opened in the last session that the client was launched. This was evident upon examination of the '.tmp' files in the second test. The file in the second test is called 'AZU1533.tmp' and lists only those torrents opened in the second session. Below is the data from this file:

[15:30:09] Tracking UBUNTU  
[15:30:09] Tracking geexbox-1.1-rc4-en.i386.iso  
[15:54:14] Tracking box to floors.mpg

The time of first two torrents opened are the same indicating that these were opened at the same time. This situation has occurred because the 'ubuntu' and 'geebox' torrent files were in an active status when Azureus was previously shut down (in 'test 1'). Upon restarting the application, these torrents automatically become active as though they were just opened.

Another of the '.tmp' files contains the lists of ports that are mapped for Azureus i.e. the TCP and UDP ports that are used to send and retrieve data with connected

peers. The last relevant '.tmp' file contains information about the files being monitored. It provides the names of the files downloaded and the trackers used to coordinate the swarm of seeders and leechers.

### 3.5.2 Test 3\_2 analysis

A file called 'trackers.config' was identified from the location 'C:\Documents and Settings\test\_3\Application Data\Azureus\'. This file was not present in the previous test and it holds the following information:

**d14:multi-trackersde8:trackersl#:http://<tracker url address> ee**

This tracker address was that of the torrent I had created i.e. the 'box to floors.torrent'. All the torrents that were downloaded previously had the tracker http://linuxtracker.org/announce, so it was hypothesised that this file only holds trackers from created torrents. This was tested by installing Azureus fresh onto a computer and creating three torrents with different tracker addresses, using the 'create a torrent' function in Azureus. The result was the creation of the file 'trackers.config' and it held the list of trackers that were entered in reverse chronological order (i.e. **d14:multi-trackersde8:trackersl#:3<sup>rd</sup> tracker url#:2<sup>nd</sup> tracker url#:1<sup>st</sup> tracker url ee**).

Link files relating to .torrent files were found within the Windows operating system's 'Recent' folder. Below is the exported data from EnCase link parser script.

Link File Path: Azureus\test3\_2\C\Documents and Settings\test\_3\Recent\  
Elive\_1[1].0\_Gem\_(stable).iso.lnk  
Created Date: 01/01/70 00:00:00  
Base Path: C:\Documents and Settings\test\_3\Desktop\  
Elive\_1[1].0\_Gem\_(stable).iso.torrent

---

Link File Path: Azureus\test3\_2\C\Documents and Settings\test\_3\Recent\box to  
floors.mpg.lnk  
Created Date: 27/06/07 15:54:14  
Base Path: C:\Documents and Settings\test\_3\Application Data\Azureus\  
torrents\box to floors.mpg.torrent



Link files are shortcuts or pointers to a target file, folder, or physical device (i.e. floppy disk drive or CD ROM) that have either been recently accessed, or where fast and easy access is required. They are created either automatically by the operating system, when installing programs or by a user opening a file. The occurrence of the 'Elive\_1[1].0\_Gem\_(stable).iso.lnk' link file was not expected as this file had not been opened, but was saved to the desktop. The torrent file 'box to floors.mpg.torrent' has a link file. The creation date and time of the link file is the same date and time that the torrent was created within Azureus. All other torrent files were downloaded but opened directly in Azureus from the website and not saved. Directly opening the torrent files leads to the torrent file becoming saved within the file path 'C:\Documents and Settings\\Application Data\Azureus\torrents\' and within the 'Temporary Internet Files' folder (i.e. C:\Documents and Settings\\Local Settings\Temporary Internet Files\Content.IE5\\). These torrent files do not have link files associated with them so the presence of link files for torrents means the torrent files have been saved or made by the user. The created time and date of the saved file was 01/01/70 00:00:00 which was not the time and date it was saved.

### **3.5.3 Test 3\_3 analysis**

The use of the CyberScrub Privacy Suite has eliminated the following keys from the registry:

1. HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedMRU
2. HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU
3. HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.torrent

The deletion of these registry keys removes the torrent link file information, as well as the names and paths of files most recently saved or copied, and most recently

opened. The registry data still, however, shows that Azureus is installed and is the default program for opening '.torrent' files.

The following files were deleted and over written from within the directory 'C:\Documents and Settings\\Application Data\Azureus': 'azureus.config', 'azureus.config.bak', 'tracker.config', 'tracker.config.bak', 'downloads.config',. The '.log' files within the 'logs' folder, the 'cache.dat', '.dat', '.bak' files from the 'active' folder and the '.torrent' files from the 'torrents' folder were all deleted and over written. All the '.torrent' files that were saved within the 'temporary internet folder' were deleted during the privacy guard advanced erasing procedure. The file names were still present as the 'scrambled' setting preference was not selected in the erasing program. The deletion of all these files means that only the '.tmp' files remain to provide clues about the download process.

## Another BitTorrent Client (3.1)

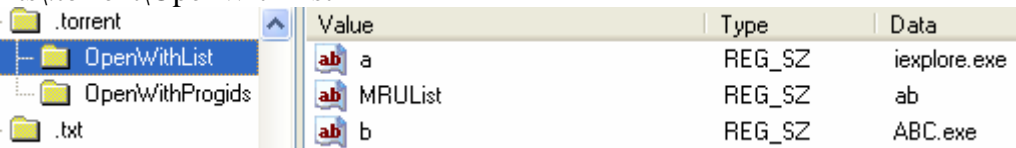
### 3.6.1 Test4\_1 analysis

'Another Bitorent Client' is also known as 'ABC'. It has a range of optional settings that can be configured by the user. These settings are stored within the file called 'abc.conf', full path C:\Documents and Settings\\Application Data\ABC. When freshly installed, the 'abc.conf' file does not display all the possible settings. Only when the user views the settings within the client do the settings then become saved within the 'abc.conf' file. Configurations of interest within the 'abc.conf' file are shown in the appendix 17 (Note each setting value is displayed as the default value found in this file).

#### Registry Analysis

The NTUSER.DAT registry hive was analysed to find references to torrent activity. The full registry path, screen shots of the registry data, and explanations of these registry keys, are shown below:

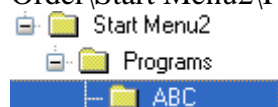
HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\FileE



Value	Type	Data
a	REG_SZ	iexplore.exe
MRUList	REG_SZ	ab
b	REG_SZ	ABC.exe

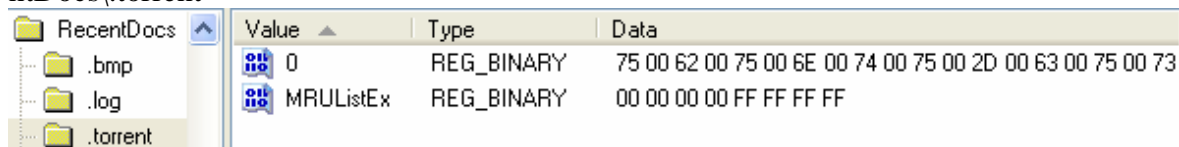
This shows that abc.exe is the default program used by windows to open files with the extension .torrent

HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Menu



No data was found in this registry key but it shows that the program ABC is installed in the start menu of program files.

HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.torrent

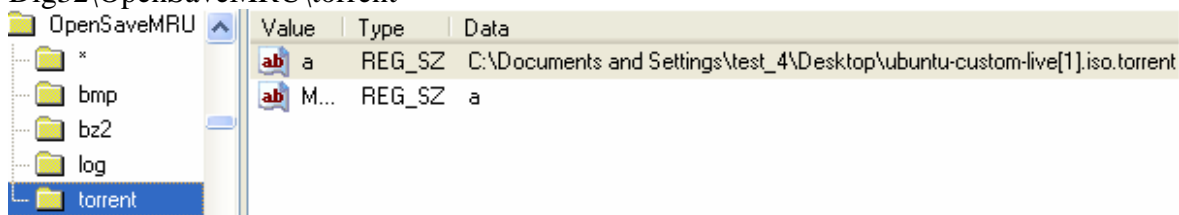


This registry entry shows the files in the recent documents list. This is the location that is accessed from the windows start tab/my recent documents. Translating the binary data into ASCII information shows the following:

ubuntu-custom-live[1].iso.torrent  
ubuntu-custom-live[1].iso.lnk,

The registry data shows that the 'ubuntu-custom-live[1].iso.torrent' was saved to, or opened from the desk top. Further evidence would need to be acquired to definitively say either way.

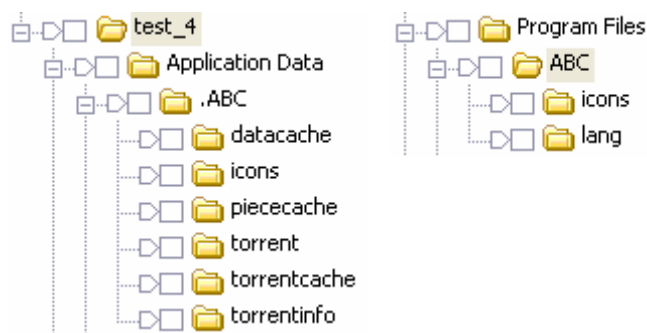
HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU\torrent



Entries within this key refer to the names and paths of files most recently saved or copied. The data therefore shows that the torrent file "ubuntu-custom-live[1].iso.torrent" has been saved to the desktop.

## Encase Analysis

The Folders created upon instillation and use of the ABC BiTorrent client is shown below:



These folders are found in the paths: C:\Documents and Settings\\Application Data\ and C:\Program Files\. The presence of these folders can be used to identify if the application has been installed and therefore further investigation should occur to identify if ABC has been used for illegal file sharing.

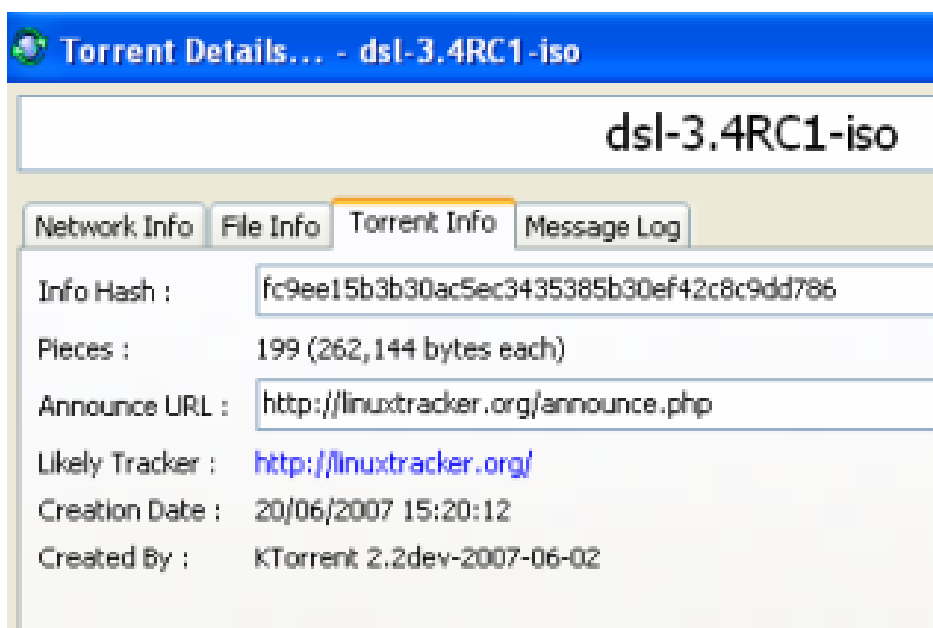
The initial step of the analysis was to examine all the files found within the folders identified above. The file 'torrent.list' is found in the path C:\Documents and Settings\test\_1\Application Data\ABC\. This file contains the following details:

```
1 = "geebox-1[1].1-rc3-en.i386.iso.torrent"  
0 = "dsl-3[1].4RC1.iso.torrent"  
3 = "deli-0[1].7.2.iso.torrent"  
2 = "ubcd411[1].exe.torrent"
```

When ABC was closed it contained the torrent files in the same order as those listed within this file. The 'torrent.list' file, only records a list of those torrents currently loaded in the client i.e. torrent files that have been loaded but were then removed are not found in this list. The torrent files this listed can be of any status i.e. paused, closed, seeding or downloading and will be labelled in the order they were opened. Backups of the 'torrent.list' file are found within the same path and contain the same data.

The ABC application creates '.info' files which contain information relating to the progress of torrent files that have been opened within the ABC client. These info files are found within the folder path: C:\Documents and Settings\test\_1\Application Data\ABC\torrentinfo\. Each file is named the same as the torrent file they represent i.e. name.torrent.info. Details of the information these files provide are shown in appendix 18. The '.info' files represent the history of downloading as they are stored every time a torrent file is opened. These files stay stored within this location even after the original torrent file has been removed from the ABC application. Prolonged usage of the ABC application would be needed to find out if these files are permanently stored.

When torrent files are opened within the client a backup is stored in the path: C:\Documents and Settings\test\_4\Application Data\ABC\torrent\. Torrent files will remain in this path unless the preferences for removing torrents from ABC have been changed to include removing the backup torrents. The ABC client also creates files that record the last state of file transfers this includes information relating to the pieces of data that are still required to complete the download. These files contain bencode and are found within the location C:\Documents and Settings\test\_4\Application Data\ABC\datacache\. These files are named according to the 'Info Hash' value of the torrent file they are associated with. The 'Info Hash' is a SHA-1 hash of the 'info' bencode data found in the torrent. This value uniquely identifies the torrent file [20]. An example is shown in the screen shot below:



E.g. the torrent dsl-3.4RC1-iso.torrent when opened in ABC contains information that the ABC client displays as an Info Hash value:

'fc9ee15b3b30ac5ec3435385b30ef42c8c9dd786'. This hash value is the file name for the file representing the state of file transfer for this torrent, and is stored in the 'datacache' folder. These files contain, the date and time that the torrents were either stopped, removed, or in the case where torrents were still active the last time the program was closed. The information required to assess the status of the torrent can be referenced from the 'torrent.info' file. The 'datacache' files do not actually identify the torrents that they pertain 'state' information about; thus, the file

name must be married up with the 'hash info' value of the torrent file. An alternate method is to identify the bencode '5:filesId6:lengthi <integer>e' found within the torrent file (this integer depicts the total size (in bytes) of the file that is to be downloaded), and match it with the integer value of the bencode 'd5:filesli0ei<integer>ei<ten digit Unix Text Date>ee', found in the datacache file. The date and time information is deduced from the string above and is in the form of a ten digit integer which can be converted using the EnCase bookmark as Unix Text Date function.

The ABC GUI has a tab called 'Network Info', which displays information about the peers that are currently connected and exchanging data for each torrent currently loaded. Screen shots of the peer information were taken while downloading different torrent files. IP addresses were identified from the screenshots and searched for within all files, file slack, and unallocated clusters of the hard drive. The search did not identify any of the known IP addresses of remote computers that were sharing files.

### 3.6.2 Test4\_2 analysis

Below is the data extracted from the file 'boxes to floors.mpg.torrent.info' of location C:\Documents and Settings\test\_1\Application Data\ABC.\torrentinfo\boxes to floors.mpg.torrent.info.

```
[TorrentInfo]
seedtime = 1162
complete = 1
prio = 2
dest = C:\movie\boxs to floorss.mpg
upsized = 10315776
downsize = 0
statusvalue = 2
progress = 100.0
```

The data highlighted in bold distinguishes this torrent as an 'initial seeder'. The information indicates that no data has been downloaded however the file is complete and the progress is one hundred percent. The data that has been

uploaded corresponds to the 'boxes to floors.mpg' file. When the 'statusvalue' equals two this indicates that the file being shared has been completely uploaded.

During the production of the 'boxes to floors.mpg.torrent' file using the ABC 'Create Torrent' tool, the 'save as default config' tab was selected and resulted in a file named 'maker.conf' becoming saved within the location C:\Documents and Settings\test\_1\Application Data\ABC\maker.config. This file contains the saved settings of the 'Create Torrent' tool and always contains the heading '[ABC/TorrentMaker]'. Explanations of the data stored in the maker.config file are shown in appendix 19. Note creating a torrent will not cause the 'maker.config' file to be saved unless the 'save as default config' tab has been selected.

### **3.6.3 Test 4\_3 analysis**

The use of the CyberScrub Privacy Suite has eliminated the following keys from the registry which contain torrent information:

1. HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedMRU
2. HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU
3. HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.torrent

The deletion of these registry keys removes the torrent link file information, as well as the names and paths of files most recently saved or copied, and most recently opened. The registry data still, however, shows ABC is the default program for opening '.torrent' files.

The CyberScrub Privacy Suite application has deleted and overwritten all the '.torrent' files from within the 'temporary internet folder' during the privacy guard advanced erasing procedure. The file names and the creation time and dates of the torrents are still present as the 'scrambled' setting preference was not selected



in the erasing program. These deleted files were found within the 'Lost Files' folder. The 'Lost Files' folder is a virtual directory created by EnCase and contains all the deleted files which have been recovered but due to the lack of information about the files parentage the original storage location cannot be determined. The CyberScrub Privacy Suite has not altered any of the files containing information relating to BitTorrent activity which were identified prior to the running of the privacy guard advanced erasing procedure.

## BitTornado (0.3.18)

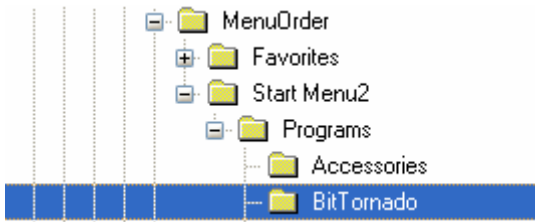
The BitTornado client has a range of optional settings that can be configured by the user. These settings are stored within the file called 'config.gui.ini', full path C:\Documents and Settings\

### 3.7.1 Test 5\_1 analysis

#### Registry analysis

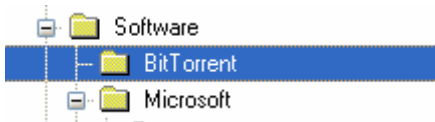
The NTUSER.DAT registry hive was analysed to find references to torrent activity. The full registry path, screen shots of the registry data, and explanations of these registry keys, are shown below:

HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\MenuOrder\Start Menu2\Programs\BitTornado



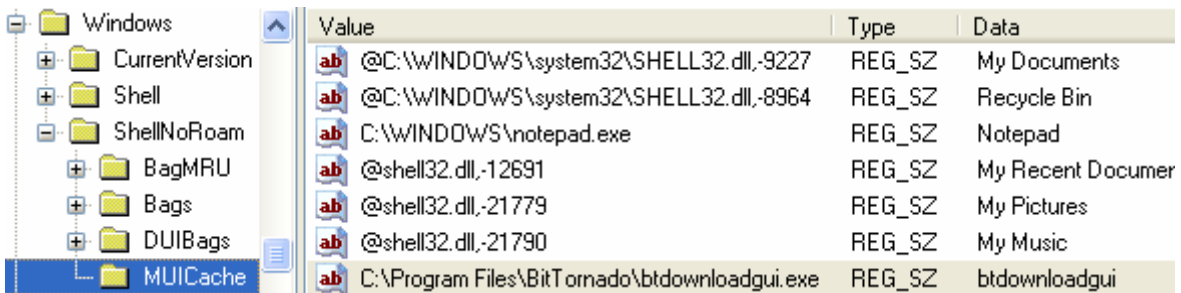
No data was found in this registry key but it shows that the program BitTornado is installed in the start menu of program files.

HKEY\_CURRENT\_USER\Software\BitTorrent



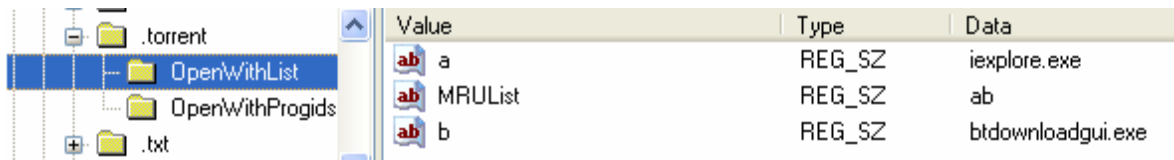
No data was found in this registry key but it shows that BitTorrent software is installed.

HKEY\_CURRENT\_USER\Software\Microsoft\Windows\ShellNoRoam\MUICache



Entries in this key refer to those programs that have been opened. This entry shows that the BitTornado application has been run.

HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Fi  
leExts\.torrent\OpenWithList



This shows that btdownloadgui.exe is the default program used by windows to open files with the extension .torrent.

HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\R  
ecentDocs\.torrent

Value	Type	Data
0	REG_BINARY	53 00 61 00 62 00 61 00 79 00 6F
MRUListEx	REG_BINARY	00 00 00 00 FF FF FF FF

This registry entry shows the files in the recent documents list. This is the location that is accessed from the windows start tab/my recent documents. The binary data translates into the following ASCII information:

```
SabayonLinux-x86-3[1].4.Loop3a.iso.torrent
SabayonLinux-x86-3[1].4.Loop3a.iso.lnk
```

As previously explained in the analysis section 3.5.2, when a user downloads and saves torrent from files the internet, link files are created and thus an entry within the 'RecentDocs' registry sub key is created by Windows. In this case the 'SabayonLinux-x86-3[1].4.Loop3a.iso.torrent', was saved to the desk top.

The Software registry hive was analysed to find references to torrent activity. The full registry path, screen shots of the registry data, and explanations of these registry keys, are shown below:

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\BitTornado

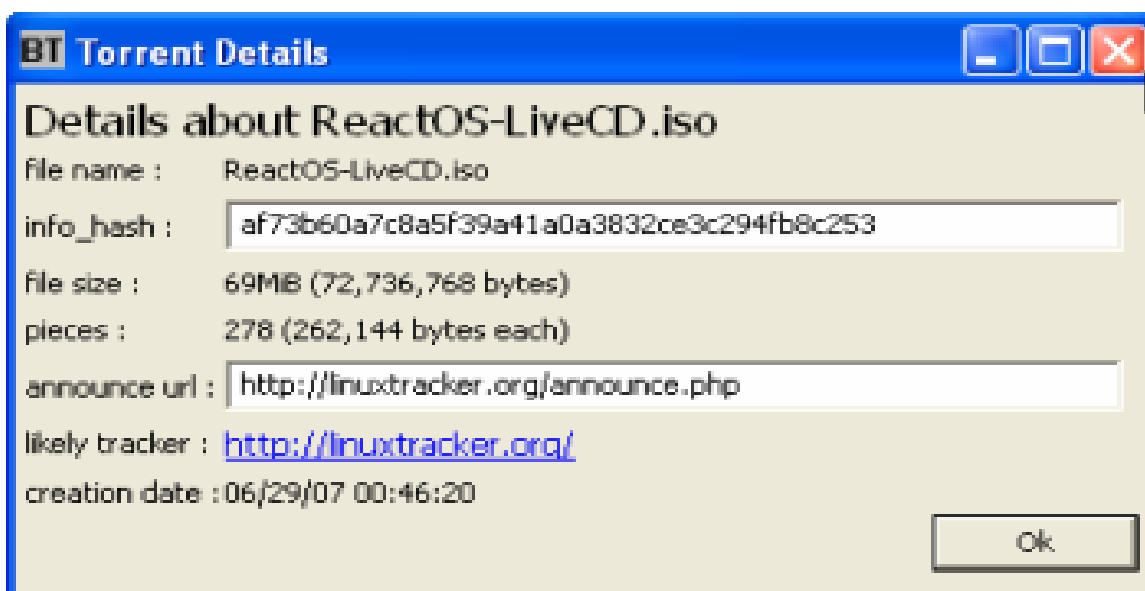
Value	Type	Data
ab DisplayName	REG_SZ	BitTornado 0.3.18
ab UninstallString	REG_SZ	C:\Program Files\BitTornado\uninst.exe
ab DisplayIcon	REG_SZ	C:\Program Files\BitTornado\btdownloadgui.exe
ab DisplayVersion	REG_SZ	0.3.18
ab URLInfoAbout	REG_SZ	http://www.bittornado.com/
ab Publisher	REG_SZ	John Hoffman

The uninstall registry key shows the version of BitTornado and the path of the uninstall install executable file.

## EnCase Analysis

The only torrent files found on this image were those stored within the Temporary Internet Files folder and the one saved to the desktop. This evidence suggests the BitTornado client doesn't save a copy of opened torrent files within its application data storage areas. Evidence of downloading can be found within the folder 'datacache' located in the path C:\Documents and Settings\\Application

Data\BitTornado\datacache. The files in this folder contain bencode data relating to the progress of download files for particular torrents, and are saved in this location upon exit of BitTornado. Appendix 21, provides an explanation of bencode found within 'datacache' files. The filenames are the 'infohash' value of the related torrent file pieces. For example the file 'af73b60a7c8a5f39a41a0a3832ce3c294fb8c253' contains bencode of which the string '8:saved as' identifies the path and name of the file being downloaded in this case it identifies the file path: C:\ReactOS-LiveCD.iso. The corresponding torrent file 'ReactOS-LiveCD[1].torrent' when opened in any client reveals the 'infohash' value 'af73b60a7c8a5f39a41a0a3832ce3c294fb8c253' which is the same as the file name in the 'datacache' folder. The screen shot (shown below) taken during testing shows the 'infohash' value.



The BitTornado GUI has a tab called 'Advanced', which displays information about the peers that are currently connected and exchanging data. Screen shots of the peer information were taken while downloading different torrent files. IP addresses were identified from the screenshots and searched for within all files, file slack, and unallocated clusters of the hard drive. The IP addresses were searched to find out if BitTornado records information regarding, connected peers and any data exchange (uploaded or downloaded) with the peers. The search did not identify any of the known IP addresses of remote computers that were sharing

files. This result indicates that BitTornado doesn't cache IP addresses during data exchange.

### 3.7.2 Test 5\_2 analysis

The use of the CyberScrub Privacy Suite has eliminated the following keys from the registry:

4. HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedMRU
5. HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU
6. HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.torrent

The deletion of these registry keys removes the torrent link file information, as well as the names and paths of files most recently saved or copied, and most recently opened. The registry data still, however, shows the version BitTornado installed, and that it is the default program for opening '.torrent' files.

The CyberScrub Privacy Suite application has deleted and overwritten all the '.torrent' files from within the 'temporary internet folder' during the privacy guard advanced erasing procedure. The file names and the creation time and dates of the torrents are still present as the 'scrambled' setting preference was not selected in the erasing program. The deletion of the torrent files within the 'Temporary Internet File' folder means that the only evidence of downloading can be found from files within the 'datacache' directory. Files within this directory contain the names of downloaded files, the file path and the volume of data downloaded and uploaded. This evidence can be used to show how particular files arrived on the computer. The CyberScrub Privacy Suite does not provide effective BitTornado evidence erasing but then again it didn't claim to.

# DISCUSSION

The aim of this study was to identify forensic artefacts produced by BitTorrent file sharing, and specifically, to establish if the artefacts could lead to identification of the files downloaded or the files shared. A further objective was to identify any artefacts that could determine IP addresses of remote computers from which data was downloaded, or shared, during the test phase. The final aim was to test whether automated erasing software would delete the BitTorrent artefacts identified. Five BitTorrent clients were selected for testing as these were determined to be the most 'popular' at the time of this study. Each client was analysed with forensic software on generated image files and also *in situ*.

Every client that was tested has a 'settings or preferences' function where the user can tweak the configurations of how the client will operate. Analysis of the clients showed that they vary in complexity and operability and thus varied in the amount of useful forensic information stored in the settings. However, investigating the settings of a client is key to understanding how it has been used, as the settings can determine information such as: where downloads will be stored, if default settings have been altered, where torrents will be stored, deletion settings, if logging is enabled, if a password has been set, the version of the client used, the ports used, the last time the client was used, and the seeding settings etc. Thus analysis of the various settings can be used to form a profile of a user and to distinguish a zealous user from a recreational user.

Torrent files are a fundamental component of the BitTorrent file sharing procedure. They are, in effect, pointers to the target files that are to be shared meaning that there is no difference between a torrent file that is used to share or download a file. The only way to determine what a torrent file has been used for (i.e. to download or share a file) is to investigate artefacts produced by the BitTorrent client used.

There are different ways that torrent files may arrive on a computer. A user can create a torrent, and save it anywhere on the system. A user can open a torrent

from a website; this causes the torrent file to be saved in the 'Temporary Internet File' folder (if Internet Explorer is used as a web browser). It is also possible a user might save a torrent from a website, email, IRC, external storage device, to any location on the system. With the exception of BitTornado, all clients analysed create a backup of torrent files (these are stored in application specific directories) when they are opened. The backups are direct copies of the torrent files opened. Azureus, uTorrent and ABC store all backup torrent files within their designated directories where they remain stored even after torrents are removed from the GUI. The backup torrent files for these clients can be deleted by the user using the GUI, but not by using the main removal tab. The BitComet client does not continue storing the torrent backups once they are removed from the GUI; hence only backup torrent files currently loaded in the GUI are stored. Torrent files contain information such as the names and sizes of files that are downloaded or shared. This information can be used as a guide to determine which files may have been downloaded or shared but the presence of a torrent file alone is not evidence of file downloading/sharing; further evidence would have to be gathered showing that the torrent has been opened within a client. As previously discussed, backup torrents are created (within specific application data directories) when torrents are opened in the clients (except BitTornado) and this is evidence of intention to download or share files.

All clients generated files containing data regarding the 'state' of downloads. These files are used as a recovery system so that torrents can resume from the same point if the program is interrupted (either by being stopped or when the client is closed). These 'cache' files can provide informative data such as: the directory where downloads are saved, the amount of data downloaded or uploaded for specific torrent files, the time and date torrents were started or stopped and the status of the torrent (i.e. complete, downloaded/seeding, or stopped). These 'cache' files, thus, provide evidence for the downloading or seeding of specific files. The locations of the 'cache' files for each torrent client are specified in the analysis sections 3.3.1, 3.4.1, 3.5.1, 3.6.1, 3.7.1.



The ABC client stores files containing downloading information on every torrent that has ever been opened within the client. The .info files provide a complete picture of the downloading history of ABC. Similarly, the BitComet client can provide a complete downloading history from the data entries within the 'my\_shares.xml' file. Data is stored in this file for every torrent file opened and 'shared'. However, the user has to select a tab during the torrent opening process in order for the torrent files to be 'shared'. The data for each torrent remains within the 'my\_shares.xml' file unless the corresponding torrent files are deleted from BitComet using 'Delete Task and Downloaded Files' option. No other BitTorrent client produced artefacts that reveal a complete history of torrents downloaded and opened.

The torrents created and seeded in the testing phase created identifiable artefacts in the ABC, BitComet, and uTorrent clients. The analysis of files generated by these clients identified data that defined the torrents that are used as 'initial seeders' to share file data. The files and the data are specified in the analysis sections 3.3.2, 3.4.4, 3.6.2, and can be used as evidence to substantiate file sharing. It should be noted that this evidential data identified in the BitComet and uTorrent artefacts only relates to torrents currently loaded in their respective clients. The analysis of test images demonstrated that only the uTorrent client produced data artefacts that identify torrent files that have been created by the client. This evidence can be used in conjunction with the data identifying torrents as 'initial seeders', to verify that file sharing was initiated on a particular computer.

Link files were found to be indicators of torrent files being created and used as 'initial seeders'. Analysis revealed two scenarios where torrent link files were created; either a torrent file was downloaded and saved from a torrent website or a torrent file was created, opened, and seeded in a client. The link files of all downloaded and saved torrents had a creation time and date of: 01/01/70 00:00:00 when parsed by the Link file EnScript. The parsed link files of the created, opened, and seeded torrents had the time and date corresponding to the time these torrents were created by the client. Link files are created either automatically by the operating system when installing programs, or by a user

opening a file; it was, therefore, not expected that the downloaded and saved torrents would produce link files. The time and date information of the link files makes it easy to identify how the corresponding torrent files arrived on the computer. However, the time and date information of saved link files may change when the torrent is opened in a client. Further testing needs to be carried out in order to determine whether this is the case.

Analysis identified that it was possible to acquire the IP addresses of connected peers for each torrent currently loaded in the client but only for uTorrent, Azureus, and BitComet. The type of peer information that was stored differed between these clients. The IP addresses, the amount of data exchanged, and the direction of exchange can be acquired from Bitcomet, but only for the point in time at which the torrent was last stopped/closed. Azureus and uTorrent, on the other hand, store a list of connected peers that are recorded throughout the duration of download or upload. It was ascertained that IP addresses stored in uTorrent and Azureus were not just stored from the moment of torrent closure, as IP addresses were retrieved from relative files corresponding to torrents that had finished downloading and were no longer connected to peers. Further tests would need to be conducted in order to determine whether every IP address connected becomes stored in the areas identified during analysis. No data artefacts relating to 'peer' information were discovered when analysing the ABC and BitTornado clients. Concretely identifying peers that have downloaded, or shared, illegal or confidential material is valuable intelligence. Internet service providers can be sanctioned to release the details of individuals that pertain to the IP addresses identified. Further investigations can then transpire.

'Emulation' is a useful tool in forensics as it provides a quick approach to understanding how applications work, and it can also be used to visually display information on the computer as seen by the user. The uTorrent application is a self contained application and does not install program files to the computer. This property makes it easy to emulate. Files stored in the directory 'C:\Documents and Settings\\Application Data\utorrent\' can be extracted and loaded into the

same directory on another computer (with no internet connection). Running the uTorrent application will begin the emulation and display the 'state' of file sharing as it was when uTorrent was last closed. Analysis of the emulated uTorrent provided information regarding the number of times uTorrent had been opened, the total amount of data uploaded and downloaded, the number of added torrents and the total time that the application ran for. This information had not previously been observed via the 'normal' method of analysis. Emulating other clients may be possible using the same technique and could result in data not previously found by the usual analysis techniques, as was the case for uTorrent.

Analysis of registry files of BitTorrent activity for each client produced many artefacts, although the evidential value of these artefacts is debatable. Artefacts created within application files show the same information and much more, for example, time and date data references are provided. In addition the presence of client application file data such as a 'cache' data verifies that the applications have been used, which is mainly what of the registry data shows. The most significant data discovered in the registry was identified in the BitComet sub key 'HKEY\_CURRENT\_USER\Software\BitComet\BitComet'. This key contains a record of the website URL, from which the last torrent was downloaded and opened from, and the 'title' of the web page (as seen in the source code of the website). This is valuable information as web page information cannot be found from analysing BitComet specific files, and the last torrent opened within BitComet is not explicitly stated in any file artefact. Two other useful registry keys provide information regarding torrent files, but the creation of these keys are not a result of BitTorrent client use. The registry keys:

'HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU\torrent' and 'HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.torrent', refer to the names and paths of files most recently saved or copied, or opened torrents. This information could be of value in circumstances where backup torrent files have been removed from the client.

## 4.1 Erasure of BitTorrent activity

'CyberScrub Privacy Suite' Version 4.5 has many options to erase different data types from a computer. The 'Privacy Guard' function is the only automated procedure for erasing data. This function does have an advanced setting where specific options can be selected. Options in the advanced setting are straight forward and refer to types of data and specific applications, rather than files or locations which a user might find confusing. This version of privacy suite has the advanced options to delete data from the BitTorrent client's uTorrent, BitComet and Azureus. Each client, when selected, lists further options depicting the type of file sharing data that can be deleted from these clients. For example, the options list relating to data that can be wiped from Azureus includes: the settings, the torrent categories, the logs, information about share folders and information about file transfers. After options are selected, automated deleting will occur.

It was found that the Cyber Scrub application uses definition files that contain scripts which detect and delete specific files generated by the clients. The definition files also contain data variables which if equal 'set' demonstrate that a user has selected the options which would delete the BitTorrent related files, when the privacy guard advanced erasing procedure is activated. The files that are targeted for deletion in each client are the 'configuration settings', backup torrent files (the script targets the default location for where the client stores the backup torrents) and the 'cache' files (described previously). The privacy guard advanced erasing procedure, when run, automatically deletes 'routine data areas' not selected by the user, such as internet browsing data. Thus, it was identified that running the privacy guard advanced erasing procedure deletes registry keys defining the names and paths of torrent files that have been recently saved, copied or opened, and also any torrent files saved in 'Temporary Internet Files' folder.

'CyberScrub Privacy Suite' can be set to create log files of an erasing procedure. The log files provide details of the names and locations of deleted files. To create log files, the user has to select the option within the program preferences. This is unlikely to occur, as the motive for using the application is to remove evidence and

logging creates evidence. However, for testing purposes, the log was useful because it revealed which files were deleted. The 'CyberScrub Privacy Suite' application does successfully delete the data that it targets. The file data is deleted and over written so that it is irretrievable using EnCase. If the 'Scramble files and folder properties' option is selected when the application is run then, not only is the file data deleted, but the name, date, and time of the files are altered. The file name is shown as random characters, and the date and time of file creation is changed to a random date and time. Thus, if this option has been selected there is no way of retrieving the data or knowing what has been deleted, using EnCase. If the 'Scramble files and folder properties' option is not selected when the application is run then deleted files can be identified by their name but their contents are not retrievable. In this case the names of the files could be used as evidence that the files were deleted using this software.

Other than the 'routine data' deletion, all BitTorrent artefacts created by ABC and BitTornado are not affected by the running of the privacy guard advanced erasing procedure. BitTornado does not store backup torrent files and uses internet cached torrent files or specifically saved torrent files to download files. Deletion of torrent files from the 'Temporary Internet Files' folder, thus, essentially removes the 'indicator' files representing the downloading history of BitTornado.

Key words were devised to enhance data retrieval from the uTorrent and BitComet clients. The key words locate file information that is cached, but not saved, and thus produce hits within the 'unallocated cluster' area of the hard disk. The key words essentially identify historic data which is can be useful because these clients mainly save data about the torrent files currently loaded.

## **4.2 Limitations of the study and implications for future research**

There are many BitTorrent clients available to download from the internet. The choice of clients tested in this project was based on two articles published on the internet [8,9] neither of which had large data sets on which to base their findings. Screen shots were produced throughout the testing phase which detailed the

names of clients connected in swarms. The client samples identified in this study corroborate the findings of these articles; however, the number of clients identified is, again, small. Long term analysis would be the only way of truly knowing which clients are the most popular. The development of BitTorrent client applications is still continuing. Newer versions of Utorrent BitComet and Azureus have been made available for download since the time testing began. The sheer number of available clients means their popularity may change over time, and indeed new clients may be developed thus, further forensic tests will need to be carried out.

# BIBLIOGRAPHY

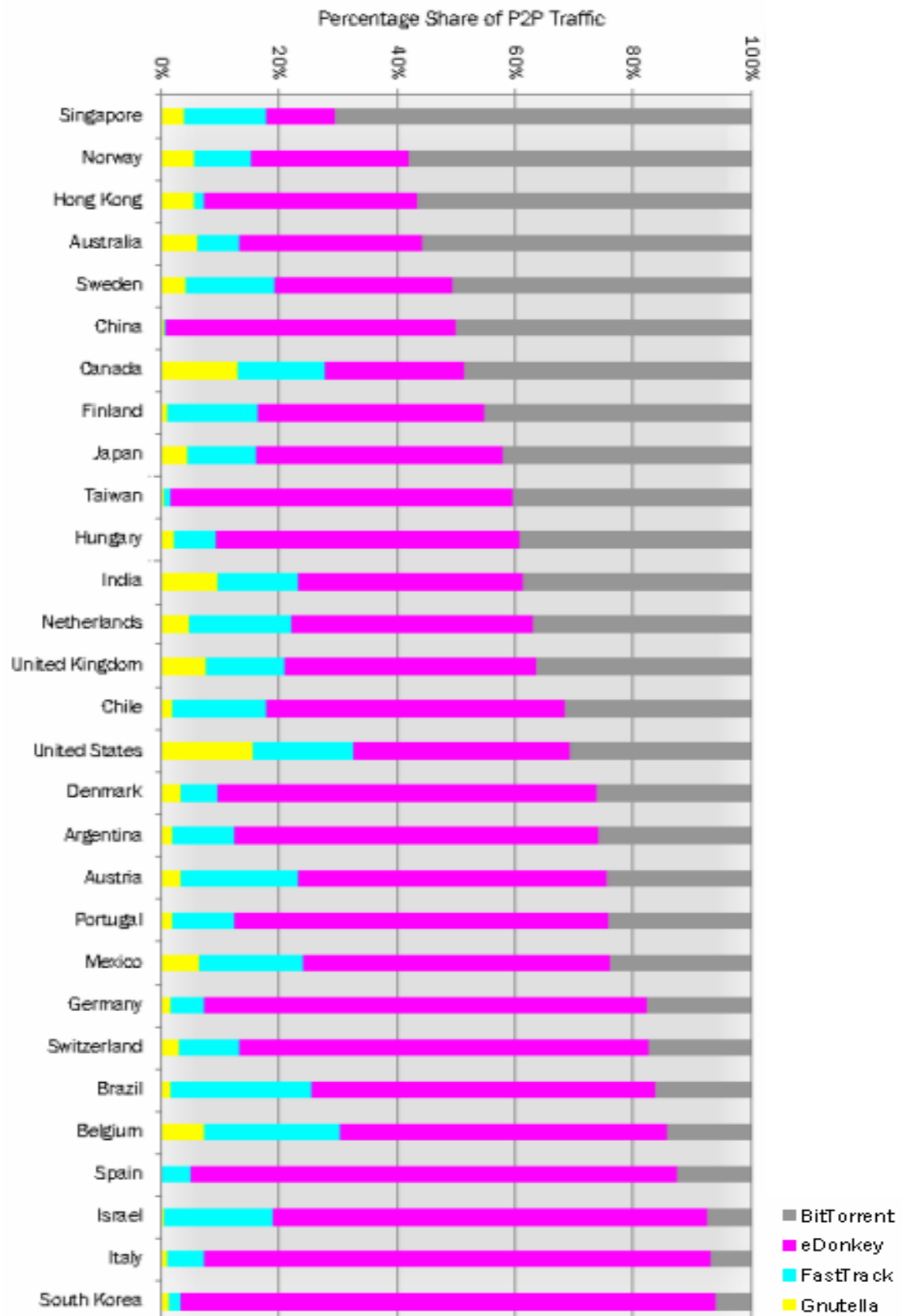
- [1] Wikimedia Foundation, Inc. (September, 2007): BitTorrent, <http://en.wikipedia.org/wiki/BitTorrent>
- [2] Cohen, B. (May 2003): Incentives Build Robustness in BitTorrent, <http://www.bittorrent.org/bittorrentecon.pdf>
- [3] BitTorrent.org (2006): DHT protocol, [http://www.bittorrent.org/Draft\\_DHT\\_protocol.html](http://www.bittorrent.org/Draft_DHT_protocol.html)
- [4] Wikimedia Foundation, Inc. (September, 2007): BitTorrent Client [http://en.wikipedia.org/wiki/BitTorrent\\_client](http://en.wikipedia.org/wiki/BitTorrent_client)
- [5] TorrentFreak (September, 2006): BitTorrent: The “one third of all Internet traffic” Myth, <http://torrentfreak.com/bittorrent-the-one-third-of-all-internet-traffic-myth/>
- [6] TorrentFreak (March, 2007): LimeWire Most Installed P2P Application, BitTorrent Clients Runner up, <http://torrentfreak.com/limewire-most-installed-p2p-application-bittorrent-clients-runner-up/>
- [7] TorrentFreak (April, 2006): BitTorrent Client Comparison, <http://torrentfreak.com/BitTorrent-client-comparison/>
- [8] Gil, P. (June 2007): About.com: Internet for beginners, <http://netforbeginners.about.com/od/peersharing/f/torrentclients.htm>
- [9] Demonoid.com (2007): Disclaimer, <http://www.demonoid.com>
- [10] Reuters Ltd. (May, 2005): Federal agents shut down network that leaked 'Star Wars', <http://www.governmentsecurity.org/archive/t14909.html>
- [11] Music Publishers Association (MPA) & L.E.K. Consulting (2006): The Cost of Movie Piracy, [http://www.mpaa.org/2006\\_05\\_03leksumm.pdf](http://www.mpaa.org/2006_05_03leksumm.pdf)
- [12] Borland, J. (March, 2004): "Judge: File sharing legal in Canada", <http://news.com.com/2100-1027-5182641.html>
- [13] Sophos (November, 2001): Glossary of terms, [http://www.sophos.com/pressoffice/news/articles/2001/11/va\\_glossary.html#controlled\\_application](http://www.sophos.com/pressoffice/news/articles/2001/11/va_glossary.html#controlled_application)

- [14] TorrentFreak (June, 2007): Windows Worm Uses BitTorrent to Propagate, <http://torrentfreak.com/windows-worm-uses-bittorrent-to-propagate/>
- [15] Sophos (September, 2007): W32/Impard-A Worm, <http://www.sophos.com/virusinfo/analyses/w32imparda.html>
- [16] Woodward, A. (2005): The effectiveness of commercial erasure programs on BitTorrent activity, [http://scissec.scis.ecu.edu.au/conference\\_proceedings/2005/forensics/woodward.pdf](http://scissec.scis.ecu.edu.au/conference_proceedings/2005/forensics/woodward.pdf)
- [17] CyberScrub LLC (September, 2007): CyberScrub Privacy Suite 4.5, <http://www.cyberscrub.us/products/privacysuite/features.php>
- [18] AccessData Cooperation (September, 2007): "AccessData: Registry Quick Find Chart", [http://www.accessdata.com/media/en\\_US/print/papers/wp.Registry\\_Quick\\_Find\\_Chart.en\\_us.pdf](http://www.accessdata.com/media/en_US/print/papers/wp.Registry_Quick_Find_Chart.en_us.pdf)
- [19] BitTorrent.org (2006): BitTorrent protocol specification, <http://www.bittorrent.org/protocol.html>

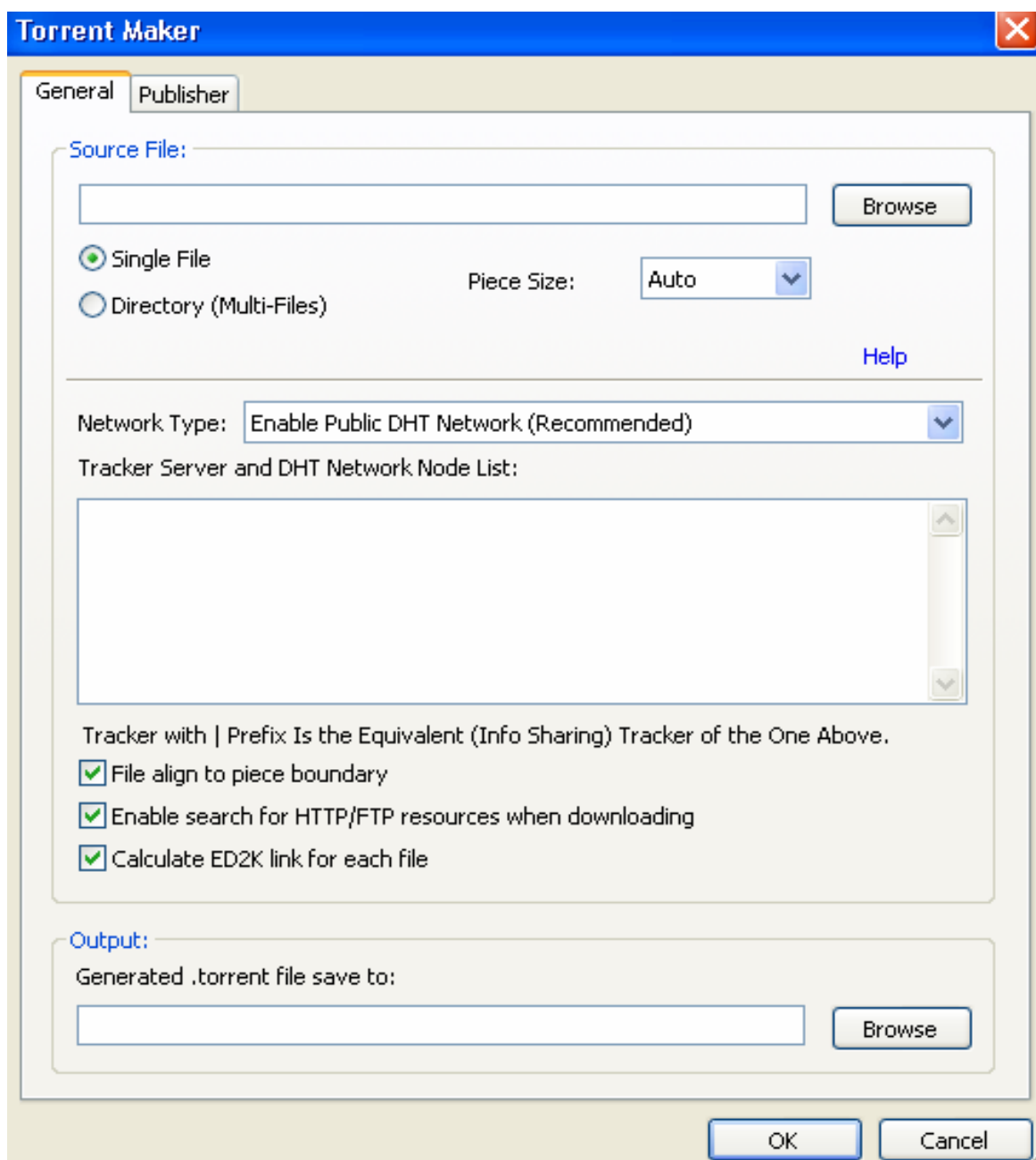


# APPENDICES

**Appendix 1: The graph below shows the percentage of world peer to peer traffic**



## Appendix 2: An example of the 'Torrent Maker' GUI from BitComet



## **Appendix 3: A simplified version of tests carried out for each BitTorrent client**

### **BitComet**

#### **Testing Part 1 (Downloading Torrents) 16/06/07**

- 1) I downloaded and opened the torrent file 'openSUSE-10.2-GM-i386-mini.iso.torrent' at 22:42. I stopped the download at 22:54. No data had downloaded as peers couldn't connect
- 2) I downloaded and opened the torrent file 'austrumi-1[1].5.0.torrent' at 22:59. I deleted the torrent in BitComet using the delete task and downloaded files option.
- 3) I downloaded and saved the torrent file 'foresight-10710-foresight-1[1].3-x86-dvd1.iso.torrent' to the desktop at 23:17
- 4) I downloaded and opened the torrent file 'Super\_Grub\_Disk\_0.9598.iso.torrent' at 23:20. The torrent was left to fully download the file.
- 5) I downloaded and opened the torrent file 'belenix0.6.iso.torrent' at 23:38. I stopped the download at 23:40

#### **Testing Part 2 (Creating and Uploading Torrents) 1/07/07**

- 1) I downloaded and saved the torrent file 'dl\_studio[1].iso.torrent' to the desktop at 11:12
- 2) I downloaded and opened the torrent file 'UBUNTU.torrent' at 11:23. I stopped the download at 98.5% completion. 8.65Mb had been downloaded by this point.
- 3) I downloaded and opened the torrent file 'geebox-1.1-rc4-en.i386.iso.torrent' at 11:30. The torrent was left to fully download the file.
- 4) I used the 'create a torrent' tab to create the torrent file 'flip1.mpg.torrent' at . At 13:02. I seeded the torrent until it had fully uploaded.

#### **Testing Part 3 (Erasure of BitTorrent Activity) 1/07/07**

- 1) I ran the Cyberscrub privacy guard advanced option to delete BitComet p2p data at 14:14. The Scramble file and folder properties option was enabled.

### **uTorrent**

#### **Testing Part 1 (Downloading Torrents) 25/06/07**

- 1) I downloaded and saved the torrent file 'polar-bear-linux-ALPHA2[1].iso.torrent' to the desktop at 12:15

- 2) I downloaded and opened the torrent file 'thunderbird-2.0.0.4.tar.gz.torrent' at 12:24. I enabled the 'Log Peer Traffic' option at 12:30 and disabled it at 12:31. At 14:06 I paused the download. 8.27Mb had been downloaded by this point which equalled 75.9% completion of the total file size.
- 3) I downloaded and opened the torrent file 'openSUSE-10.2-GM-i386-mini.iso.torrent' at 12:44. I closed uTorrent at 14:12. 41.3Mb had been downloaded by this point.
- 4) I downloaded and opened the torrent file 'lg-live-0.9.iso.torrent' at 12:57. I deleted the torrent from uTorrent using the 'remove' tab at 13:03. 2.75Mb had been downloaded by this point.
- 5) I downloaded and opened the torrent file 'Super\_Grub\_Disk\_0.9598.iso.torrent' at 13:58. The download completed at 14:00.
- 6) I closed uTorrent at 14:14

### **Testing Part 2 (Creating and Uploading Torrents) 25/06/07**

- 1) I opened uTorrent at 19:26 and deleted all the torrents from uTorrent using the 'remove' tab.
- 2) I used the 'create a torrent' tab to create the torrent file 'box to floor.mpg.torrent' at . At 19:31. I was unable to seed this torrent as peers could not connect (I think I had router/firewall configuration problems).

### **Testing Part 3 (Erasure of BitTorrent Activity) 26/06/07**

- 1) I ran the Cyberscrub privacy guard advanced option to delete uTorrent p2p data at ~ 21:27.

### **Testing Part 4 (Creating and Uploading Torrents) 30/06/07**

- 1) I used the 'create a torrent' tab to create the torrent file 'box floor.mpg.torrent' at 21:01. I initiated seeding at the time of creation. Other peers were unable to connect and download this torrent so I deleted using the remove tab.
- 2) I used the 'create a torrent' tab to create the torrent file 'boxs floors.mpg.torrent' at 21:36. I initiated seeding at the time of creation. Other peers were unable to connect and download this torrent so I deleted using the remove tab.
- 3) I used the 'create a torrent' tab to create the torrent file 'boxs floorss.mpg.torrent' at 21:36. I initiated seeding at the time of creation. Other peers were unable to connect and download this torrent until I changed the setting 'enable UPnP port mapping'. The torrent uploaded 2.4Mb of the file but then failed to connect with any peers.
- 4) I clicked on the 'clear Private Data' tab found within preferences and sub category 'other'.

## **Azureus**

### **Testing Part 1 (Downloading Torrents) 27/06/07**

- 1) I downloaded and saved the torrent file 'Elive\_1[1].0\_Gem\_(stable).iso.torrent' to the desktop at 11:20
- 2) I downloaded and opened the torrent file 'geebox-1[1].1-rc4-en.i386.iso.torrent' at 11:28. The torrent was left to fully download the file.
- 3) I downloaded and opened the torrent file 'ubuntu-7[1].04-professional.torrent' at 12:40. I stopped the download at 12:44. 312Kb had been downloaded by this point.
- 4) I downloaded and opened the torrent file 'bluewhite64-12[1].0-rc1-dvd.torrent' at 12:49. At 12:54 I deleted the torrent in Azureus using the remove tab. 3Mb had been downloaded and 1.75MB uploaded by this point.
- 5) I downloaded and opened the torrent file 'manifestdestiny-respin2-binary-dvd-i386[1].iso.torrent' at 13:14. At 13:19 I deleted the torrent in Azureus using the remove tab.
- 6) I closed Azureus at 13:22

### **Testing Part 2 (Creating and Uploading Torrents)**

- 1) I deleted the torrent 'geebox-1[1].1-rc4-en.i386.iso.torrent' in Azureus, using the remove and delete .torrent and data option.
- 2) I deleted the torrent 'ubuntu-7[1].04-professional.torrent' in Azureus, using the remove and delete .torrent option.
- 3) I used the 'create a torrent' tab to create the torrent file 'box to floors.mpg.torrent' at 15:54. At 16:24 I seeded the torrent until it had fully uploaded.
- 4) I closed Azureus at 16:24

### **Testing Part 3 (Erasure of BitTorrent Activity)**

- 1) I ran the Cyberscrub privacy guard advanced option to delete Azureus p2p data at 17: 45.

## **ABC**

### **Testing Part 1 (Downloading Torrents) 28/06/07**

- 1) I downloaded and saved the torrent file 'ubuntu-custom-live[1].iso.torrent' to the desktop at 12:16
- 2) I downloaded and opened the torrent file 'dsl-3[1].4RC1.iso.torrent' at 12:33. I directed the download file to be saved in the directory 'C:\ New Folder' . I paused the download at 12:54. I
- 3) I downloaded and opened the torrent file 'postal2\_aw\_et\_awp\_inx\_binary[1].torrent' at 13:12. I directed the download file to be saved in the directory 'C:\ New Folder'. I deleted the torrent from ABC using the remove tab at 13:30.
- 4) I changed the preferences to download files by default to the directory 'C:\my downloads' and I closed ABC.

- 5) I started ABC and on opening, the paused dsl-3[1].4RC1.iso.torrent' became active so I stopped it. I downloaded and opened the torrent file 'ubuntu-7[1].04-professional.torrent' at 14:37. I changed the preferences so that the remove tab actually deletes .torrent files from the uTorrent default torrent storage area (C:\Documents and Settings\\Application Data\ABC\torrent\). I then deleted the torrent using the remove tab at 14:50.
- 6) I downloaded and opened the torrent file 'puppy\_215CE\_RC3a[1].iso.torrent' at 14:56. I then deleted the torrent using the remove tab at 14:59.
- 7) I downloaded and opened the torrent file 'ubcd34-full[1].zip.torrent' at 15:01. I saved a message log for this torrent in the directory 'C:/New Folder/ubcd34-full.zip.log' at 15:07. I then deleted the torrent using the remove tab at 15:08
- 8) I downloaded and opened the torrent file 'geebox-1[1].1-rc3-en.i386.iso.torrent' at 15:18. This torrent was queued by ABC and no data was downloaded.
- 9) I downloaded and opened the torrent file 'ubcd411[1].exe.torrent' at 15:42. The torrent was left to fully download the file.
- 10) I downloaded and opened the torrent file 'deli-0[1].7.2.iso.torrent' at 15:49. The torrent was left to fully download the file.

### **Testing Part 2 (Creating and Uploading Torrents) 26/06/07**

- 1) I opened ABC at 17:44 and deleted 'geebox-1[1].1-rc3-en.i386.iso.torrent' and 'dsl-3[1].4RC1.iso.torrent' using the 'remove' tab. The torrent 'ubcd411[1].exe.torrent' was left seeding until it uploaded 6.56Mb and then the torrent was stopped. The torrent file 'deli-0[1].7.2.iso.torrent' was also stopped.
- 2) I used the 'create a torrent' tab to create the torrent file 'boxs to floors.mpg.torrent' at 11:14 28/06/07. The torrent didn't seed and I removed it from ABC.
- 3) I used the 'create a torrent' tab to create the torrent file 'boxs to floorss.mpg.torrent' at 13:05 28/06/07. I seeded the torrent until it had fully uploaded.

### **Testing Part 3 (Erasure of BitTorrent Activity) 28/06/07**

- 1) I ran the Cyberscrub privacy guard advanced option (ABC was not available as an option for P2P data deletion)

## **BitTornado**

### **Testing Part 1 (Downloading Torrents) 29/06/07**

- 1) I downloaded and saved the torrent file 'SabayonLinux-x86-3[1].4.Loop3a.iso.torrent' to the desktop at 18:15.

- 2) I downloaded and opened the torrent file 'ReactOS-LiveCD[1].torrent' at 18:24. I deleted the torrent by using the cancel tab at 19:31 by this point 11.5Mb had been downloaded.
- 3) I downloaded and opened the torrent file 'ubuntu-7[1].04-professional.torrent' at 19:09. I paused the torrent by using the pause tab at 19:17.
- 4) I downloaded and opened the torrent file 'geebox-1[1].1-rc4-en.i386.iso.torrent' at 14:47. The torrent was left to fully download the file; this took 5min 23 sec.

### **Testing Part 2 (Erasure of BitTorrent Activity) 29/06/07**

- 2) I ran the Cyberscrub privacy guard advanced option (BitTornado was not available as an option for P2P data deletion).



## Appendix 4: An example of the bencode data contained in a torrent file

```
d8:announce17: tracker url address7:comment11:testing 12310:created
by13:uTorrent/161013:creation datei1183239136e8:encoding5:UTF-
84:infod6:lengthi10315776e4:name16:boxs floorss.mpg12:piece
lengthi65536e6:pieces3160:5 .°_· ñSÛ=< 1ß ?Pçμ
```

The table below show a dissection of the above torrent bencode.

Torrent bencode	Explanation
d8:announce17: tracker url address	This 'anounce' dictionary defines the tracking URL
4:name16:boxs floorss.mpg	The name of the file that the torrent saves
4:infod6:lengthi10315776e	The length of the file 'boxs floorss.mpg' is 10315776 bytes
12:piece lengthi65536e	The length for each piece the file is split in to is 65536 bytes
6:pieces3160:5 .°_· ñSÛ=< 1ß ?Pçμ	There are 158 file pieces each with a SHA1 hash value of length 20. The first hash value of the first file piece is shown.
13:creation datei1183239136e	The cration date is coded in accordance with Unix Text Date.
10:created by13:uTorrent/1610	The torrent file was created by uTorrent version 1.6.1
7:comment11:testing 123	The creator of the torrent file made the comment: testing 123

**Appendix 5: The following table defines the bencode found in the configuration file 'BitComet.xml'**

<b>Forensically Interesting Bencode</b>	<b>Explanation</b>
</ListenPort> <b>24019</b> </ListenPort>	This is the port used for BT file transfer it is chosen at random and can be randomized at any time the user wishes
</EdListenPort> <b>22942</b> </EdListenPort>	This is the eDonkey port for file transfer
<DefaultBtClient> <b>0</b> </DefaultBtClient>	0 = not set as default BT client. This setting only appears if it is not the default client
<BitcometLoginPassword> <b>c8590d3573cb1a29123e3da2ae3b1f62b5de62a37d1d9fd2ec0f</b> </BitcometLoginPassword>	A login password has been set by the user to access the BitComet application
<DefaultDownloadPath> <b>C:\user defined</b> </DefaultDownloadPath>	Only appears if the default destination has been changed from C:\download
<LastExitTime> <b>1182038313</b> </LastExitTime>	This is the time and date that BitComet was last shut down. The date and time can be decoded using the EnCase bookmark function with Unix Text Date

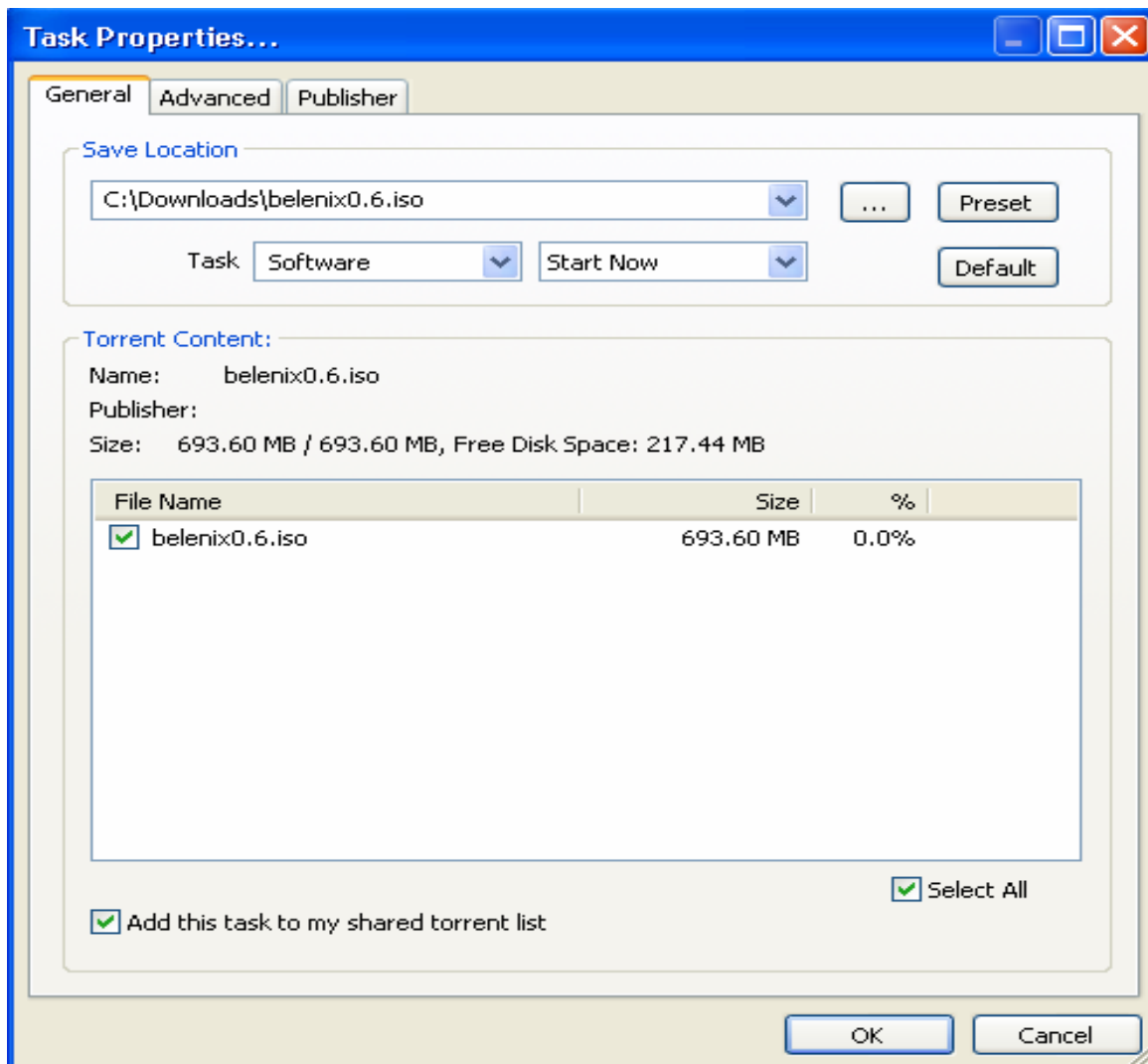
**Appendix 6: The following table defines the code found within the 'Downloads.xml' file**

Xml code	Explanation
Catalog=" "	The BitComet client has a file category section for downloaded torrents. This is the category that the torrent is being associated with and it is a way of cataloging download torrents. There are four default categories (Video, Music, Software, Picture) however other categories can be added by the user.
Size=" "	The total size of the file(s) i.e the size of the file(s) when completely downloaded
DataUpload=" "	This is the amount of data uploaded to other computers on the internet
DataDownload=" "	The amount of data that has been downloaded
Left=" "	If the value is greater than zero the download is incomplete and the amount displayed is the amount of data left to complete the download.
CreateDate=" "	This is the date and time the torrent was first opened in the BitComet client
FinishDate=" "	This is the date and time when files are completely downloaded i.e. when Left="0". If the FinishDate = the CreateDate then this torrent is being uploaded by the user. A generated .torrent file must always point to the source file (the file to be shared); therefore when the generated torrent is opened, the client begins to seed because it detects the source file that the torrent points to.
Torrent=" "	This is the name of the torrent file that has been opened within the client
TorrentFile="Torrents\ "	This is the name of the torrent file that has been opened within the client and the location it is saved i.e. C:\Program Files\BitComet\torrents.
SaveLocation=" "	This is the location that the file downloading will be saved to. It includes the name of the file being saved.
InRecycleBin="true"	Bitcomet has its own Recycle Bin function. When the user deletes torrent files they have the option to place it in the Recycle Bin. The value "true" occurs when the torrent file has been sent to the Bitcomet Recycle bin. Torrents within the Recycle Bin can be restored back into the client.

**Appendix 7: The following table defines the code found within the .xml files**

<b>Xml code</b>	<b>Explanation</b>
<FileList BaseName=" "	This identifies the file name that will be downloaded
SaveLocation=" "	This identifies the location the file will be saved to
<TrackerList>	A list of trackers used by the torrent follows this statement e.g. <Tracker>http://linuxtracker.org/announce.php</Tracker>
<Peer downloaded="1622016" ip="71.198.179.146" port="32459" uploaded="0"></Peer>	This string of statements identifies the remote IP address and port number of the computer that data is being shared with. The Peer downloaded statement identifies the amount of data in bytes that was downloaded from the remote computer and the uploaded statement identifies the amount of data that was downloaded by the remote computer.
<FileEntry LastWriteTime="1183287254" Priority="0" RelativePathName="ubuntu-7.04-professional-i386.iso"	This string of statements identifies the last time the file in the 'RelativePathName' was written to The date and time can be decoded using the EnCase bookmark function with Unix Text Date. The priority value '0' is normal (and is the default value), '-1' is disabled, '1' is high and '2' is highest. These priorities are the preference given to the file being downloaded i.e. BitComet will try to download a file given the highest priority first.
Size=" "	This value is the total size in bytes of the file to be downloaded
<TorrentInfo CreateDate=" "	This identifies the date and time that the torrent was first opened within BitComet
FinishDate=" "	This identifies the date and time that the download was completed
DataDownload=" "	This is the total amount of data that has been downloaded from all the connected peers.
DataUpload=" "	This is the total amount of data that has been downloaded by all the connected peers.
TorrentPathName=" "	This identifies the name of the .Torrent file that has been opened within BitComet

**Appendix 8: The following snapshot shows the ‘Task Properties’ GUI**



**Appendix 9: The following table identifies the bencode from the 'settings.dat' file and explains the function of the code**

<b>Forensically Interesting Bencode</b>	<b>Function</b>
19:dir_active_download52:C:\<user defined>	This is the directory that downloaded files will be saved to.
24:dir_active_download_flagi1e	The presence of this string means the download directory is enabled else it is disabled
22:dir_completed_download12:C:\<user defined>	This defines the directory that completed downloads will be moved to. Incomplete ones will be left in the default download dir.
27:dir_completed_download_flagi1e	The presence of this string means the completed download directory is enabled else it is disabled
14:move_if_defdiri0e	If this string is present then all completed downloads will move to the complete download directory else only default dir downloads will move.
13:dir_add_labeli1e	This string enables the appendage of the torrent label to the directory name (i.e. the name of the torrent will be the name of the new folder where files are stored)
17:append_incompletei1e	This string performs the function of appending '.!ut ' to incomplete files i.e. The file extension ends with .!UT for all incomplete downloaded Files.
15:encryption_modei1e	This string indicates outgoing encryption is enabled. if this string is absent then encryption is disabled
15:encryption_modei2e	outgoing encryption is forced
18:rand_port_on_starti1e	If the integer value for this string equals 1 then the port that is used for data sharing is randomised each time uTorrent starts
9:bind_porti61008e	The integer value represents the port used for incoming connections
22:dir_torrent_files_flagi1e	This string indicates the storage of .torrent files box is ticked thus all .torrent files will be stored in the user defined directory
17:dir_torrent_files13:C:\<user defined>	This string defines the path where .torrent files will be stored. If the user doesn't define a storage directory for .torrent files then the default storage

	path is: C:\Documents and Settings\ <user>\Application Data\uTorrent</user>
27:dir_completed_torrents_flag1e	This string enables the directory for storage of completed .torrent to become active
22:dir_completed_torrents13:C:\<user defined>	This string identifies the path where .torrent files will be stored once the download has completed
12:dir_autoload7:C:\<user defined>	This string defines a directory which if .torrent files are present then they will be automatically loaded in to uTorrent; however the user still has to acknowledge the process and press continue for the torrent to begin downloading
17:dir_autoload_flag1e	This string enables the autoload directory to become active

**Appendix 10: The following table defines bencode found in the 'resume.dat' file**

<b>Forensically Interesting Bencode</b>	<b>Explanation</b>
d8:added_oni1182776330e	This is the time and date that the torrent file was opened within uTorrent. The time and date is revealed by decoding with Unix Text Date
12:completed_oni0e	The integer value represents the time and date that the download was completed. The time and date is revealed by decoding with Unix Text Date. If the integer = 0 then the download is incomplete
10:downloadedi3518464	The integer value is the size in bytes of data that has been downloaded of the target file
4:path:C:\<file path>\<file name>	The string 'path' identifies the path and file name of the downloading file is saved as.
7:runtimei819e	The integer value is the total time in seconds that the torrent has been open (running)
8:seedtimei747e	The integer value is the total time in seconds that the torrent has been seeding
4:timei1182777150e	The integer value is the time and date that the uTorrent application was last shutdown.
8:trackersl36:http://linuxtracker.org/announce.php	This string identifies the tracker used by the torrent
8:uploadedi0	The integer value defines the amount of data in bytes that has been uploaded to remotely connected computers.




**Appendix 11: The table below shows the bencode '5:peers42:¢--UZ--`CRi·Ús×Y¥/^D±U\*uR.ÉB¾+·S·QL4·Î Xá·' converted in to 8-Bit Integer using the EnCase bookmark function**


Char	Hex	UInt8	Int8	Binary
¢	a2	162	-94	10100010
·	19	25	25	00011001
·	16	22	22	00010110
U	55	85	85	01010101
Z	5a	90	90	01011010
·	1b	27	27	00011011
-	2d	45	45	00101101
`	60	96	96	01100000
C	43	67	67	01000011
R	52	82	82	01010010
ï	ef	239	-17	11101111
·	1a	26	26	00011010
Ú	da	218	-38	11011010
s	73	115	115	01110011
×	d7	215	-41	11010111
Y	59	89	89	01011001
¥	a5	165	-91	10100101
/	2f	47	47	00101111
^	5e	94	94	01011110
D	44	68	68	01000100
±	b1	177	-79	10110001
U	55	85	85	01010101
*	2a	42	42	00101010
u	75	117	117	01110101
R	52	82	82	01010010
·	2e	46	46	00101110
É	c9	201	-55	11001001
B	42	66	66	01000010
¾	be	190	-66	10111110
·	18	24	24	00011000
+	2b	43	43	00101011
S	53	83	83	01010011
·	1	1	1	00000001
Q	51	81	81	01010001
L	4c	76	76	01001100
4	34	52	52	00110100
·	4	4	4	00000100
Î	ce	206	-50	11001110
	a0	160	-96	10100000
X	58	88	88	01011000
á	e1	225	-31	11100001
·	1a	26	26	00011010

**Appendix 12: The screen shot below shows the emulated state of test 2\_1 when uTorrent was closed**

Name	#	Size	Done	Seeds	Peers	Completed On	Uploaded	Dow
openSUSE-10.2-GM-i386-mini.iso	2	44.0 MB	94.0%	0 (0)	0 (8)		0.0 kB	
Super_Grub_Disk_0.9598.iso		3.35 MB	100.0%	0 (0)	0 (7)	25/06/2007 14:00:04	0.0 kB	
thunderbird-2.0.0.4.tar.gz	1	10.8 MB	75.9%	0 (0)	0 (4)		0.0 kB	

Downloaded:  94.0 %

Availability:  0.932

Time Elapsed: 1h 28m      Remaining: ∞      Share Ratio: 0.000

Downloaded: 41.3 MB      Download Speed: 0.0 kB/s (avg. 8.0 kB/s)      Down Limit:

Uploaded: 0.0 kB      Upload Speed: 0.0 kB/s (avg. 0.0 kB/s)      Up Limit:

Seeds: 0 of 0 connected (0 in swarm)      Peers: 0 of 8 connected (0 in swarm)      Wasted: 0.0 kB (0 hashfails)

Web Seeds:

**Tracker**

Tracker URL: <http://linuxtracker.org/announce.php>

Tracker Status:

Update In:

DHT Status: inactive

**General**

Save As: C:\Documents and Settings\test\_2\My Documents\Downloads\openSUSE-10.2-GM-i386-mini.iso

Total Size: 44.0 MB (41.3 MB done)      Pieces: 177 x 256 kB (have 165)

Created On: 11/06/2007 00:06:24 by KTorrent 2.1.4      Hash: 39AFA485 F44FD82B 977EE2C2 2C27AF07 B475664B

## Appendix 13: This data was recovered from the deleted log file produced by the CyberScrub Privicy Suite erasure process

Privacy Suite Version 4.5  
Copyright © 1997-2006 Beyondsec Technologies

Log file

\*\*\*\*\*

Erase started on 26/06/2007 at 21:27:57

Options :

Security level: U.S. Department of Defense (3 passes)

This is the default erasing mode

Disk buffer size: 512KB

Use ISAAC random generating algorithm: Disabled

Wipe file using current wipe method: Enabled

Wipe compressed files: Enabled

Wipe file slack: Enabled

Use Recycle Bin Disabled

Scramble file and folder properties: Disabled

When enabled, this feature renders folder and file names unreadable

Scramble alternate data streams: Enabled

Allow erase interrupt: Enabled

Action :

Erasing 783 file(s) containing 8.4 MB of sensitive data

Erasing 20 registry entry(ies)

-> File C:\Documents and Settings\test\_2\Application Data\uTorrent\box to floor.mpg.torrent

Erased OK

-> File C:\Documents and Settings\test\_2\Application Data\uTorrent\lg-live-0.9.iso.torrent

Erased OK

-> File C:\Documents and Settings\test\_2\Application Data\uTorrent\openSUSE-10.2-GM-i386-

mini.iso.torrent

Erased OK

-> File C:\Documents and Settings\test\_2\Application Data\uTorrent\Super\_Grub\_Disk\_0.9598.iso.torrent

Erased OK

-> File C:\Documents and Settings\test\_2\Application Data\uTorrent\thunderbird-2.0.0.4.tar.gz.torrent

Erased OK

-> File C:\Documents and Settings\test\_2\Application Data\uTorrent\resume.dat

Erased OK

-> File C:\Documents and Settings\test\_2\Application Data\uTorrent\resume.dat.old

Erased OK

-> File C:\Documents and Settings\test\_2\Application Data\uTorrent\settings.dat

Erased OK

-> File C:\Documents and Settings\test\_2\Application Data\uTorrent\settings.dat.old

Erased OK

**Appendix 14: The following table defines the bencode found within the Azureus configuration file**

<b>Forensically Interesting Bencode</b>	<b>Functionality</b>
15:TCP.Listen.Porti14783e	The integer value identifies the TCP port used by Azureus for file transfer.
15:UDP.Listen.Porti14783e	The integer value identifies the UDP port in use for file transfer
9:User Modei0e	There are three user modes that Azureus can operate in. Each mode has various degrees of complexity in terms of user requirements. The integer value 0 identifies the 'basic' user mode. This is the default mode set during installation
9:User Modei1e	The integer value 1 identifies the 'intermediate' user mode
9:User Modei2e	The integer value 2 identifies the 'advanced' user mode
22:First Recorded Version7:2.5.0.4	This is the version of Azureus first installed
15:azureus.version7:2.5.0.4	This is the current version of Azureus installed
33:General_sDefaultTorrent_Directory <b>63:C:\Documents and Settings\user&gt;\Application Data\Azureus\torrents</b>	This is the default directory where .torrent files are stored. Preferences can be configured so that .torrent files are not always stored in this directory. Examples include .torrent files that are selected to be moved with completed downloads, or when they are selected for removal within the client.
20:Use default data diri1e	This string defines that downloads will automatically be placed within the default directory. The default directory = C:\Documents and Settings\ <user&gt;\my documents\azureus="" downloads<="" td=""> </user&gt;\my>
17:Default save path <b>46:C:\&lt;user defined&gt;</b>	If a path for saving downloads is chosen then they are saved in the location defined by the user
32:Move Deleted Data To Recycle Bini0e	If the integer value = 0, then when data is selected for deletion the data is deleted and it is not moved to the Recycle Bin. By default, the integer value is equal to 1 which means deleted data is transferred to the Recycle Bin. This setting does not appear within the configuration file unless it has been changed.
25:Completed Files Directory <b>48:C:\&lt;</b>	If this string is present within the

<b>user defined&gt;</b>	settings then the user has selected to move completed downloads to a user defined path
24:Move Completed When Donei1e	Completed downloads can only move to the user defined directory if a tick box is selected by the user. If the box is selected the integer value =1, if unselected the integer value =0
8:Password20:L+ò+oÜ H¥- +V8 &+	If a password is made by the user to prevent Azureus from being opened, the pw will be encrypted to the value defined by the string
16:Password Confirm20:L+ò+oÜ H¥- +V8 &+	The password has to be confirmed by the user; if it matches then the password is enabled in the program
16:Password enabledi1e	The password is enabled when the integer value =1; disabled when the value =0
14:Logger.Enabledi1e	The Azureus client has logging capability. If the integer value =0 then logging is not enabled; if the value =1 then logging is enabled
14:Logging Enablei0e	If the integer value =0 then the log created will not be saved; If the value = 1 then a log file is saved
11:Logging Dir49:C:<user defined path>	This string defines the path where logs are saved to
16:Logging Max Sizei5e	the default value for the maximum size of log file is set to 5mb but can be changed to a maximum of 500mb
34:File.move.download.removed.enable di1e	There is an option within preferences to move files downloaded and .torrent files to a user defined directory when torrents are selected for removal from GUI. If a tick box is selected by the user.the integer value =1; If unselected the value =0
31:File.move.download.removed.path14 :C:\ <user defined>	when the torrent is removed from Azureus, this moves the downloaded files and the .torrent file to the folder defined by the user
39:File.move.download.removed.move_torrenti0e	when torrents are removed from Azureus if the integer value =0 then the .torrent file remains in stored in the 'torrent' directory ; If the value = 1 then .torrent files are moved to the user defined 'File.move.download.removed.path'

**Appendix 15: The following bencode was extracted from the 'downloads.config' file, and defines information for one torrent only. The table provides an explanation of the code.**

```
d9:downloadsld9:allocatedi1e9:completedi1000e12:creationTimei1182940115939e9:discardi0e10:downloadedi9287319e15:file_prioritiesl1ee10:forceStarti0e13:hashfailbytesi0e5:maxdli0e5:maxuli0e10:persistenti1e8:positioni1e8:save_dir63:C:\Documents and Settings\test_3\My Documents\Azureus Downloads9:save_file27:geebox-1.1-rc4-en.i386.iso18:secondsDownloadingi832e18:secondsOnlySeedingi5985e5:statei75e7:torrent105:C:\Documents and Settings\test_3\Application Data\Azureus\torrents\geebox-1[1].1-rc4-en.i386.iso.torrent12:torrent_hash20:đO%  \2' UÇ7-9Kÿò)^ÿ!8:uploadedi0e7:uploadsi4ee
```

<b>Forensically Interesting Bencode</b>	<b>Explanation</b>
9:completedi1000e	If the integer value i=1000 then the download has completed
12:creationTimei1182940115939e	The first 10 digits of the integer value represent the time and date that the torrent was loaded in to Azureus. Using the bookmark option in EnCase and encoding as a Unix Text Date gives the actual date and time of 27/06/07 11:28:35.
10:downloadedi9287319e	The integer value of 9287319 is the number of bytes that has been downloaded.
8save_dir63:C:\Documents and Settings\<user>\My Documents\Azureus Downloads	This states the directory into which the downloaded file will be saved. In this case, it is the default directory for files downloaded.
9:save_file27:geebox-1.1-rc4-en.i386.iso	This states the name of the file or folder being saved
18:secondsDownloadingi832e	The integer value of 832 is the number of seconds for which the torrent was being downloaded
18:secondsOnlySeedingi5985e	The integer value of 5985 is the number of seconds that the torrent is being seeded for, after the file has finished downloading
7:torrent102:C:\Documents and Settings\user>\Application Data\Azureus\torrents\geebox-1[1].1-rc4-en.i386.iso.torrent	This is the directory where the torrent file is stored and the name of the torrent file. In this case, it is the default directory for torrent files
8:uploadedi0e	The integer value 0 represents the number of bytes of the 'geebox-1.1-rc4-en.i386.iso' file uploaded for this torrent

**Appendix 16: The table below defines the bencode found within the 'cache.dat' file**

<b>Forensically Interesting Bencode</b>	<b>Explanation</b>
25:stats.download.added.timei <b>1182940116</b> 055ee	The first 10 digits of the integer value represent the time and date that the torrent was loaded into Azureus. Using the bookmark option in EnCase and encoding as a Unix Text Date gives the actual date and time of 27/06/07 11:28:35.
11:primaryfile91:C:\Documents and Settings\test_3\My Documents\Azureus Downloads\geebox-1.1-rc4-en.i386.iso	This shows the file name and location of the file being downloaded
11:timesincedli <b>5986</b> e	This is the time in seconds that data was last downloaded
11:timesinceuli <b>-1</b> ee	The value -1 means that no time has elapsed since data was uploaded as it has not yet happened. A positive value represents the time that has elapsed (in seconds) since the data was uploaded.
9:createdby26:KTorrent 2.2dev-2007-06-02	This is the client that the torrent was created by
4:sizei <b>9246720</b>	The integer value represents the size that the file will be when fully downloaded

**Appendix 17: The table below defines the bencode found within the configuration file 'abc.conf'**

<b>Default Settings</b>	<b>Functionality</b>
defaultfolder = c:\	There is no default destination for files that are downloading. Each time a torrent is opened the user is prompted for a destination to save the downloading file(s). However a default destination can be configured within the preferences.
setdefaultfolder = 0	The value zero means no destination is configured for automatic saving of downloading files. The value One means the downloading files will be saved in the directory referenced by the default folder.
defaultmovedir = c:\	Once a download has completed the downloaded file(s) can be moved in to a directory of the user's choice.
movecompleted = 0	The value Zero means this option is not activated. A value of one means the user has selected for completed downloads to be moved in to the directory referenced in 'defaultmovedir'
numsimdownload = 2	The maximum number of active torrents can be altered by the user to any value. It would be made higher if the user wants to download more than two files consecutively
trigwhenfinishseed = 0	Consider torrents active if they are downloading or seeding, if =1 then only downloading torrents are considered active
defaultpriority = 2	This is the default given to newly opened .torrent files. 0 = Highest priority, 1 = High Priority, 2= Normal priority, 3 = low priority, 4 = Lowest priority.
uploadratio = 100	upload until UL/DL ratio =100%
minport = <random value>	The value stated is the port number used for BitTorrent file transfer
removetorrent = 0	The GUI remove button acts to remove selected torrents from the application and by default the 'removetorrent' value is set as zero. A value of one denotes the remove button will cause the removal of the selected torrents from the GUI and the .torrent backup file stored in the torrent folder (full path: C:\Documents and Settings\test_1\Application Data\ABC.\torrent\)



**Appendix 18: The table below defines the code found within the '.info' files**

<b>Forensically Interesting information</b>	<b>Explanation</b>
[TorrentInfo]	Each file starts with this header
seedtime = 322	The integer value is the number of seconds relating to the length of time since the torrent has completed downloading. Therefore it is the time the bandwidth has been used solely seeding the torrent.
complete = 1	'complete' only occurs for fully downloaded torrents i.e. when the file download has completed.
prio = 2	The 'prio' value is the last priority setting of the torrent as described earlier.
dest = C:\my downloads\<>file name>	'dest' shows the storage path destination and file name that the torrent will download.
upsized = 0	This identifies the amount of data in bytes that have been uploaded by the torrent file
downsize = 130703360	This identifies the amount of data that has been downloaded by the torrent file.
progress = 100.0	This is a percentage value of completion that a file has downloaded.
fileprogress = l2:2%0:e	The 'fileprogress' information is presented a download contains more than one file. The progress represents the percentage of a file downloaded
statusvalue = 1	The status value of 1 means the torrent has been stopped or removed from the client by the user. For torrents that are seeding/downloading or paused there is no status value present in the file

**Appendix 19: The The table below defines the code found within the maker.config file**

<b>Forensically Interesting information</b>	<b>Explanation</b>
comment =	The torrent created can have a comment added to it.
created_by = jamie	The creator of the torrent has the option to write some information here.
savetorrent = 0	This setting defines the directory that the created torrent is saved. The value '0' will save the torrent to the defined default directory, the value '1' will save to the folder containing the source file used to create the torrent file, the value '2' means the creator has to define the path where to save the torrent file.
savetordeffolder =	If defined this value is the default path that created torrent files will be saved.
announce-list	This is a list of tracker sites that have been added by the user for quick reference, and therefore may be the sites used to link torrent files they have previously been created.
startnow = 1	The value '1' triggers the newly created torrent to automatically load into ABC and begin seeding. The value '0' will not start the torrent once it is created.

**Appendix 21: The table below defines the bencode found within 'datacache' files**

<b>Forensically Interesting Bencode</b>	<b>Explanation</b>
datad5:filesli0ei393216ei1183141101ei1ei0ei1183140576ee	This data dictionary lists details of files that are being downloaded. The first file is represented by the integer 0. The following integer value is the size of the file that has been downloaded in bytes. The next integer value is the time the file was last accessed by the BitTornado (the value is encoded as a Unix Text Date). The next integer '1' shown in blue is the second file etc.
8:priority	This string lists file priorities. Each file to be downloaded is given an index number separated by commas (0 = highest, 1 = normal, 2 = lowest, -1 = download disabled). The user can select the download priority for particular files. The default for each file is =1 unless changed by the user.
8:partialsl, 6:placesli	If these two strings are present in the file this means the file being downloaded is not complete. These strings are followed by lists and integers and they indicate the pieces of the file that still need to be downloaded. When the pieces of the file have completely downloaded then these strings do not occur in the bencode.
8:saved as#:C:\<full file path>	This string defines the path where the downloaded file was saved
10:downloadedi12058624e	This integer represents the total amount downloaded in bytes.
8:uploadededi425984e	This integer represents the total amount uploaded in bytes.

**Appendix 20: The table below defines the code found within the 'config.gui.ini' file**

<b>Default Settings</b>	<b>Function</b>
expire_cache_data = 10	The number of days after which you wish to expire old cache data (0 = disabled).
gui_default_savedir = " "	This identifies the path of the default save directory for downloaded files. The default path is not set until the user creates one.
gui_saveas_ask = -1	This value identifies whether the gui prompts the user for a download destination (0 = never, 1 = always, -1 = automatic resume).
last_saved = ""	This identifies the path where the last download was saved.
maxport = 60000	This value is the maximum port number that BitTornado can use for data transfer.
minport = 10000	This value is the maximum port number that BitTornado can use for data
super_seeder = 0	If the value = 1 then the client has been set to use special upload-efficiency-maximizing routines