symmetry

MDPI

Article

A Methodology and Tool for Investigation of Artifacts Left by the BitTorrent Client

Algimantas Venčkauskas [1], Vacius Jusas [2,*], Kęstutis Paulikas [3] and Jevgenijus Toldinas [1]

[1] Computer Department, Kaunas University of Technology, Studentu St. 50, Kaunas LT-51368, Lithuania; algimantas.venckauskas@ktu.lt (A.V.); eugenijus.toldinas@ktu.lt (J.T.)
[2] Software Engineering Department, Kaunas University of Technology, Studentu St. 50, Kaunas LT-51368, Lithuania
[3] Applied Informatics Department, Kaunas University of Technology, Studentu St. 50, Kaunas LT-51368, Lithuania; kestutis.paulikas@ktu.lt
* Correspondence: vacius.jusas@ktu.lt; Tel.: +37-065-676-159

Academic Editor: Young-Sik Jeong
Received: 1 April 2016; Accepted: 19 May 2016; Published: 26 May 2016

Abstract: The BitTorrent client application is a popular utility for sharing large files over the Internet. Sometimes, this powerful utility is used to commit cybercrimes, like sharing of illegal material or illegal sharing of legal material. In order to help forensics investigators to fight against these cybercrimes, we carried out an investigation of the artifacts left by the BitTorrent client. We proposed a methodology to locate the artifacts that indicate the BitTorrent client activity performed. Additionally, we designed and implemented a tool that searches for the evidence left by the BitTorrent client application in a local computer running Windows. The tool looks for the four files holding the evidence. The files are as follows: *.torrent, dht.dat, resume.dat, and settings.dat. The tool decodes the files, extracts important information for the forensic investigator and converts it into XML format. The results are combined into a single result file.

Keywords: BitTorrent protocol; forensics investigation; XML format; cybercrime

1. Introduction

Internet technologies are improving every day, occupying new areas of our daily life, presenting new services, making things easier, and changing the way we live. Millions of individuals benefit from the online virtual world. This phenomenon has exponentially increased in recent years. One of these Internet technologies is peer-to-peer (P2P) applications [1]. P2P applications enable file sharing over the Internet. There are many kinds of P2P network protocols. One of the best known is the BitTorrent protocol [2]. This protocol introduced a new technology and a new terminology to file sharing and became the *de facto* standard for file distribution over the Internet. The BitTorrent protocol was designed to ease the sharing of large files, but this technology was so innovative and simple to use that people adopted it to carry out illegal activities, like sharing of copyrighted material [2]. Illegal copies of copyrighted content can be found in more than two- thirds of torrents registered at one of the most popular BitTorrent trackers [3]. Such a use of the protocol reduces the copyright holder's possible revenue due to lost sales of the original copy. Consequently, the use of the BitTorrent protocol for illegal purposes creates a new type of cybercrime and new challenges for forensics investigators in order to fight against it. Moreover, users involved in the use of the BitTorrent protocol have to allow their resources to be used for file sharing since BitTorrent applications may punish non-uploaders by limiting their download bandwidth [4]. In such a way, users who desire the service have to choose either to be involved in committing cybercrime or refuse the service. Only aware users can make this

choice. However, many users are unaware and take part in cybercrime without knowing it. Therefore, the fight against illegal uses of the BitTorrent protocol must be intensified.

The universe of the BitTorrent protocol can be regarded as a structure consisting of several levels of hierarchy. At the highest level, the universe of the BitTorrent network can be represented as being divided into many BitTorrent swarms [5]. Each shared piece of content forms a BitTorrent swarm that consists of trackers and peers [6]. A peer is an agent that runs an implementation of the protocol. To initiate a download of the file, there are two possibilities: (1) the user must download a metadata .torrent file from a BitTorrent indexing website, or (2) the user must download a magnet universal resource identifier from a BitTorrent indexing website. Then, the user's BitTorrent client application interprets the metadata and uses it to detect other peers using one of the following methods: a tracker, a distributed hash table (DHT), or peer exchange (PEX). A tracker is a server that maintains a list of peers. Peers are required to distribute the file, exchange control messages in order to update its status and keep the list of active peers up to date. DHT is a distributed tracker that allows peers to locate the other peers requesting information from BitTorrent clients without the requirement for a central server. PEX enables a direct interchange of peer lists with other peers.

The family of products using the BitTorrent protocol constantly improves and expands. The most recent product in this family is BitTorrent Sync. It is a file replication utility released in April 2013 [7]. The utility is very convenient for those who are involved in illegal activities, because it guarantees the data transfer to be secure from inspection while in transit [8]. Therefore, the utility can be exploited for several potential cybercrimes as follows: distribution of child pornography, distribution of copyrighted material, delivery of malicious software, industrial espionage, *etc.* [9].

Not only are new applications and software released, but new versions of operating systems are released, as well. Several years ago, Microsoft released a new version of the operating system Windows 8, which was then improved and upgraded to Windows 8.1. This operating system is the newest one in the Windows family, except Windows 10. Windows 8.1 is quite successful, and the popularity of Windows 8.1 is growing [10]. The purpose of this paper is to investigate and locate the digital fingerprints [11], called artifacts at the higher abstraction level, left by the BitTorrent client on a local computer and to present the tool we created that provides a convenient way to study those artifacts.

The artifacts of the BitTorrent client can be separated into two parts: (1) the artifacts left on the local client computer; and (2) the artifacts observed on the computer network. In the next section, we review the related work that considers the artifacts left by the BitTorrent client on a local computer.

2. Review of Related Work

We present this review of related work in chronological order of their emergence, because it might make it easier to understand the direction of the development of BitTorrent client forensic investigation on local computers.

Adelstein and Joyce [12] introduced the File Marshal tool for automatic detection and analysis of P2P application usage. The tool is oriented toward the analysis of a static image of the computer disk. File Marshal is a universal tool; it is not dedicated to the specific P2P protocol. The tool's management is based on the configuration file. The file indicates the location of log files and names of registry keys. If a special code is required to analyze a file (e.g., to decode a peer list or time format), the configuration file enumerates the Java modules, which are used for parsing; new parsers have to be created as needed. Subsequently, experience is needed to successfully use the File Marshal tool. Nevertheless, the tool had success and maintenance was ensured for it. Later, the File Marshal project was renamed P2P Marshal [13]. One might note that the File Marshal tool initially did not include the analysis of BitTorrent protocol artifacts.

Anyone consciously participating in illegal activity tries to hide the evidence. Special erasure programs are created for such a purpose. Woodward and Valli [14] examined whether the available erasure programs remove the evidence of BitTorrent protocol artifacts. The erasure programs MaxErase, P2PDoctor, Privacy Suite, Window Washer, Windows R-Clean and Wipe were investigated on a

machine that had used the BitTorrent client Azureus. Woodward and Valli concluded that the available erasure tools are not effective at removing evidence of the BitTorrent protocol and the forensic investigator can successfully recover the data.

In the next study, Woodward [15] studied whether the BitTorrent clients running on Windows 7 leave behind the meaningful data. The BitTorrent client programs BitComet, BitTornado, µTorrent, and Vuze (formerly Azureus) were investigated using default settings. The research was dedicated to studying the registry and local data area within the Windows operating system. Woodward concluded that all BitTorrent client programs created the same data during their operation. This data could be used as a basis for the location of the initial source of a downloaded file.

Lallie and Briggs [16] explored three BitTorrent client applications, namely BitComet, Vuze and µTorrent. The purpose of the study was to outline the registry artifacts created during the installation of the BitTorrent client on a Windows 7 machine. Several authors [15,17] were already doubtful about the evidential value of registry keys. The study showed that many artifacts are created in the registry keys. However, these artifacts can mainly determine that a BitTorrent client has been used on the computer. The most significant discovery in the registry was an identification of the BitComet subkey that contains a record of the website URL from which the last torrent was opened and downloaded. Lallie and Briggs confirmed the already known result that the artifacts in the registry keys can only show who installed and who used the application.

The latest product in the BitTorrent family is BitTorrent Sync, which is mainly devoted to file replication. Farina *et al.* [7] investigated in detail the evidence left behind by BitTorrent Sync in the client computer. The authors conducted an experiment during which they installed the BitTorrent Sync client on a Windows XP machine. A complete list of the files created during the installation procedure is provided. The registry keys created during the installation procedure are represented, as well. It is worth noting that the default installation process creates a BTSync folder with three hidden files: .SyncID, .SyncIgnore, and .SyncArhive (folder). Finally, the BitTorrent Sync application was uninstalled. The remaining Windows registry keys after uninstallation are provided, as well.

Venčkauskas *et al.* [18] investigated the latest version of the BitTorrent client for the Windows 8.1 operating system. The contents of the changed registry keys and the configuration files are provided. During the experiment, the authors considered the possibility of hiding the evidence of the use of the BitTorrent client. The investigation found that the evidence is left in the Windows registry or in the BitTorrent configuration files, or both, depending on the method used to remove the BitTorrent client application.

The BitTorrent client program always generates several files that contain the information related to the BitTorrent program. All these files are stored in BEncode format [19]. Dedicated tools are needed to open and read such files. The BEncode Editor [20] allows BEncoded files to be opened. However, the editor provides only the basic functionality of extracting the keys and values. Many values are presented in integer or binary format that cannot by comprehended by a human reader. Other utilities are needed to convert them into a more readable form.

The Plaso framework [21] allows automatic information extraction from a large number of sources including the Windows registry, browsers histories, file systems and others. The Plaso framework provides the basic interface for BEncode plugins [22]. However, the framework lacks many functions useful for forensic investigators when analyzing BitTorrent files. Therefore, more elaborate tools are needed to aid forensic investigators in the analysis of BitTorrent files.

The review revealed that the BitTorrent protocol and the supporting environment are constantly evolving. New BitTorrent clients, which generate their own application data, enter the market. The application data generated by the different BitTorrent clients have separate structure and distinct contents. Therefore, new research is needed in order to keep pace with the state of the art for the BitTorrent protocol and its supporting environment. Next, the reviewed tools lack many functions useful for forensic investigators to analyze BitTorrent files. Consequently, new, more elaborate tools,

which would enable analysis of BitTorrent client artifacts without knowing the particular structure of the application data, are needed to facilitate analysis.

3. Methodology to Locate BitTorrent Client Artifacts and Results of the Experiment

To locate the artifacts left by the BitTorrent client on a local computer, we suggest the following methodology:

1) Create an experimental environment consisting of the server and virtual machines.
2) Install the BitTorrent client onto the virtual machines and analyze the changes in the file system.
3) Download a file using the BitTorrent client without an Internet connection and analyze the changes in the BitTorrent configuration files.
4) Download the file using the BitTorrent client when the virtual machines are connected to the Internet and analyze the changes in the BitTorrent configuration files.

The goal of the second step of the methodology is to locate the files which are added by the BitTorrent client application during installation and investigate their contents. The goal of the third step is to locate the evidence of BitTorrent activity in the BitTorrent configuration files. In order to keep a clean experimental environment and to avoid possible pollution by Internet artifacts, we suggested the virtual machines not connect to the Internet during the third step. Firstly, it needs to investigate the BitTorrent configuration files in a limited environment. That would ensure the minimal contents of the files.

The goal of the fourth step of the methodology is to locate the evidence of BitTorrent activity in the BitTorrent configuration files when the virtual machines are connected to the Internet. This would allow us to observe the artifacts added by the Internet connection and to investigate the real environment. That is the difference from the third stage.

To start an experiment according to the methodology, we created the experimental environment that consisted of a host machine and a virtual environment running on the host machine (Figure 1). The virtual environment included three virtual machines two client machines, VM1 and VM2, and a virtual server to connect these machines to the virtual local network and provide an Internet connection. This a model of an ISP (Internet service provider).

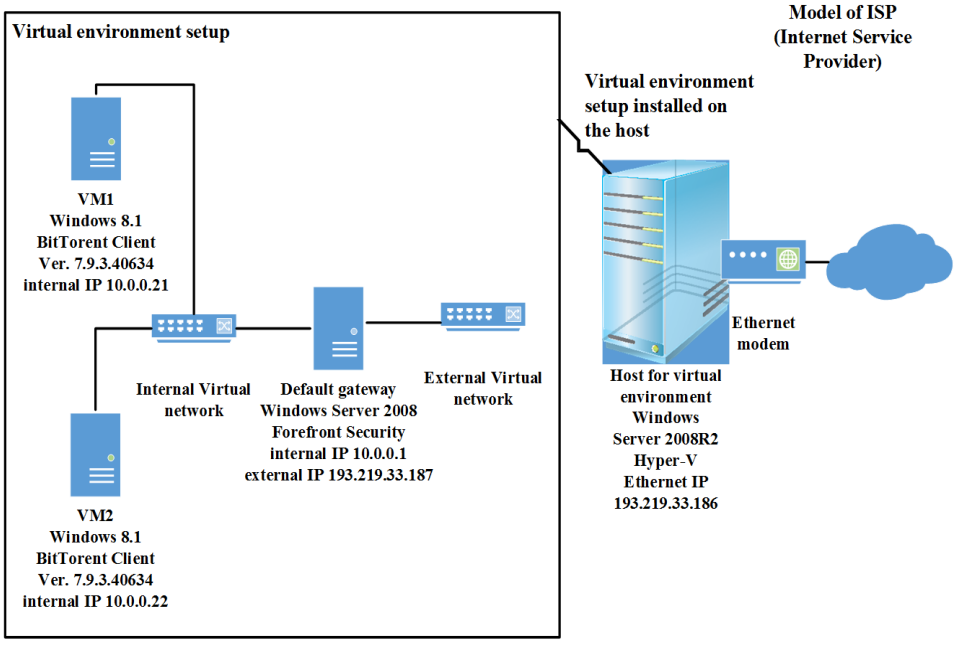


Figure 1. Diagram of the experimental setup.

The preparation steps of the virtual environment were as follows:

1) We created a virtual machine image using Windows 8.1, installed all updates available on the day of the experiment, downloaded [23] and saved the latest version of the BitTorrent download client application (version 7.9.3 (build 40634)) installation file BitTorrent.exe to the user download directory.
2) On the host computer using the Hyper-V manager, we created a default gateway virtual machine using Windows Server 2008 and Forefront Security, which is used to model an ISP in the real world. The default gateway was set up to have two virtual networks: an external virtual network to provide the Internet connection to the virtual machines, and an internal virtual network to connect the virtual machines to the default gateway.
3) On the host computer using the Hyper-V manager, we created two identical virtual machines, VM1 and VM2, using an image created at step 1.
4) VM1 and VM2 were connected to the internal network using static internal IP addresses as shown in Figure 1.

Next, we proceeded to the second step of the proposed methodology. In order to keep a clean experimental environment, we disconnected the default gateway from the external virtual network to ensure that the virtual environment is only connected to the internal virtual network and does not have an Internet connection via default gateway external virtual network connection. We then started BitTorrent.exe from the current user's downloads directory and the installation process began on virtual machine VM1. During installation, we disabled the option "Start BitTorrent when Windows starts." When the installation process was over, the BitTorrent client application started automatically. We left VM1 running for 20 min. We closed the BitTorrent client application, shut down VM1 and attached the VM1 virtual hard disk to the host computer.

The BitTorrent client installation created a BitTorrent directory in the hidden Roaming catalog (path: VM1vhd: \Users\Eugenijus\AppData\Roaming). The created directory contained various files. We examined the time of the creation of the files. Firstly, the following files were created: BitTorrent.exe, maindoc.ico, settings.dat, settings.dat.old, toolbar.benc, and update.dat. Then, after 10 min the new files were added: dht.feed.dat, dht.feed.dat.old, resume.dat. Finally, after 15 min, more files were added: dht.dat, resume.dat, resume.dat.old, rss.dat, settings.dat, settings.dat.old.

An investigation of the BitTorrent client carried out by [18] revealed that the forensic evidence accumulated in the four following files: dht.dat, resume.dat, *.torrent, and settings.dat. At this step, we decided to examine the contents of the files dht.dat and resume.dat. To view BitTorrent files, the BEncode Editor [20] is very useful tool. However, just BEncode Editor is not enough. It shows the keys and their values in a readable format (Figure 2), but the values are coded. In the initial dht.dat file, we found a non-empty value for key age (i) only, which is formatted as an integer representing a time stamp with the number of seconds since the UNIX epoch. In order to convert the integer, we used a UNIX timestamp converter (Figure 2). The key age (i) shows the last access time of the file dht.dat. Initially, it is the file creation time. As we can see, it is not convenient to examine the contents of the BitTorrent files; we need several tools.

In the files resume.dat and resume.dat.old, we did not find any useful information. They looked identical and they were empty.

During the third step, we installed the BitTorrent client on the virtual machine VM2, as well. The file "Eugenijus DSC04967.jpg" was placed for sharing on the virtual machine VM2. The torrent file "Eugenijus DSC04967.jpg.torrent" was created for this file. This torrent file was copied to the virtual machine VM1. The BitTorrent client application was started on the virtual machine VM1 and the download process of the file was completed in just seconds on the local network without connection to the Internet.

We chose three configuration files for the investigation: dht.dat, Eugenijus DSC04967.jpg.torrent, and resume.dat. The file dht.dat did not have information useful for a forensic investigator. The file Eugenijus DSC04967.jpg.torrent had the keys and values, which are presented in Table 1.

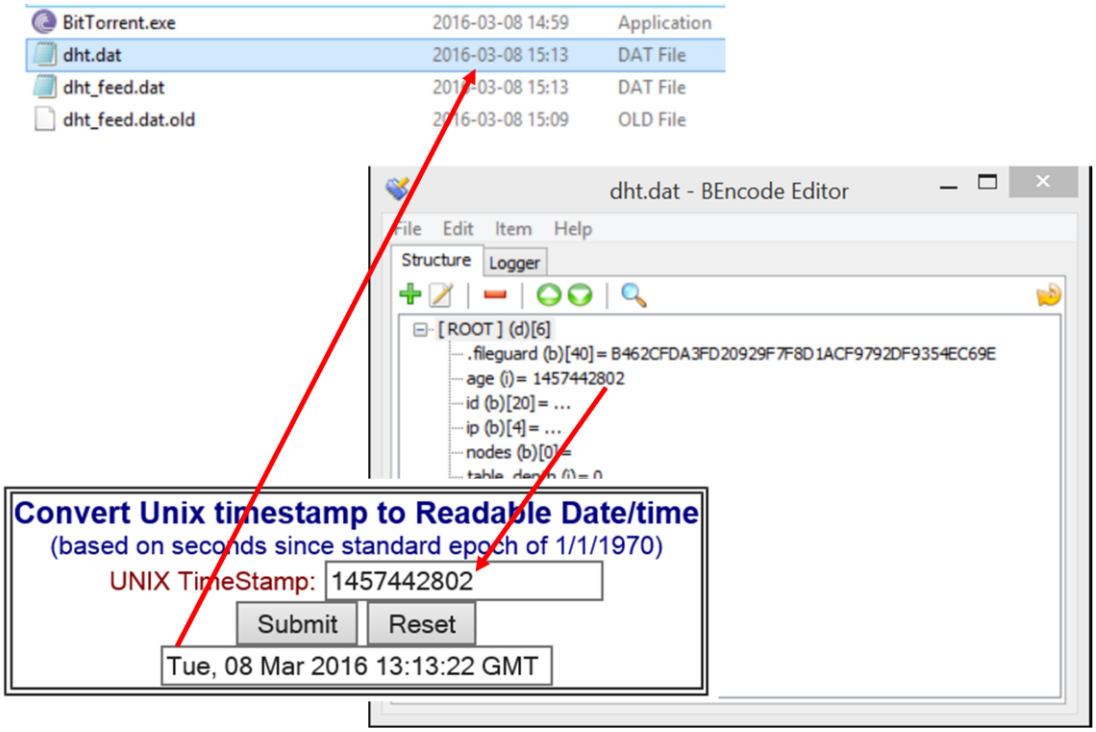


Figure 2. Decoding of key age (i) in the dht.dat file.

Table 1. Keys and values of .torrent file.

Key	Value
announce (b) (44)	udp://tracker.openbittorrent.com:80/announce
announce-list (i) (3)	udp://tracker.openbittorrent.com:80/announce, udp://tracker.publicbt.com:80/announce, udp://tracker.ccc.de:80/announce
created by (b) (16)	BitTorrent/7.9.3
creation date (i)	1,457,115,966 (Friday, 04 March 2016 18:26:06 GMT)
encoding (b) (5)	UTF-8
info (d) (4)	
length (i)	1,374,542 (the length of the file Eugenijus DSC04967.jpg in bytes)
name (b) (22)	Eugenijus DSC04967.jpg
piece length (i)	16,384
pieces (b) (1680)	20 bytes length hash for every file block, total 84 hashes

The file resume.dat had many keys; however, many values were empty. We have chosen the keys and values which are of interest for a forensic investigator (Table 2).

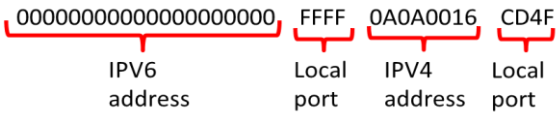
Table 2. Non-empty keys and values of resume.dat file.

Key	Value
added on (i)	1,457,117,887 (Friday, 04 March 2016 18:58:07 GMT)
blocksize (i)	16,384 (length of file block in bytes)
caption (b) (22)	Eugenijus DSC04967.jpg
completed_on (i)	1,457,117,893 (Friday, 04 March 2016 18:58:13 GMT)
downloaded (i)	1,374,542 (length of the downloaded file in bytes)
path (b) (51) =	C:\Users\Eugenijus\Downloads\Eugenijus DSC04967.jpg
peers6 (b) (54)	Peers represented by pair of IP address and port number in binary format

The most valuable information for a forensic investigator is IP addresses of peers that took part in the file downloading process. We examined the key peers6 (54) and decoded it as 18 bytes for every IP address for the peer:

```
00000000000000000000FFFF0A0A001566A4
00000000000000000000FFFF7F00000166A4
00000000000000000000FFFF0A0A0016CD4F
```

Then every 18 bytes can be decoded as follows:



Finally, we converted the hexadecimal notation into usual IP dot notation and obtained the following IP addresses: 10.10.0.21: 26276 (VM1 IP address and local port 26276), 127.0.0.1: 26276 (localhost IP address and local port 26276), 10.10.0.22: 52559 (VM2 IP address and local port 52559). As we can see again, several additional operations are needed to obtain the important information in a readable format. Therefore, it would be very useful to have a tool that would perform all the required transformation for us and present the information in the readable format.

To summarize the third step of the experiment, we concluded that the resume.dat file contains the list of peers which were participating in the download process. The peers are identified by IP address and port number. This information is very important for the forensic investigator, because he can find out who participated in the illegal activity if sharing was illegal.

Finally, we proceeded to the last step of the experiment and connected the virtual machines to the Internet. The virtual machine VM2 was used for file sharing; the virtual machine VM1 was used for the file downloading as in the third step of the experiment. After file downloading, we chose three configuration files for the investigation: dht.dat, Eugenijus DSC04967.jpg.torrent, and resume.dat. The file dht.dat contained additional information in comparison with the previous steps of the experiment (Table 3).

Table 3. Keys from the dht.dat file.

Key	Comment
age (i)	1457118779 (Friday, 04 March 2016 19:12:59 GMT)
Ip (b) (4)	ÁÛ!S (C1DB21BB -> 193.219.33.187–our virtual environment default gateway external IP address)
Nodes (b) (6240)	Binary array having length of 3120 bytes (6240 hexadecimal digits)

The key nodes (b) (6240) contained a lot of coded information (Figure 3). Therefore, we further examined the coded information.

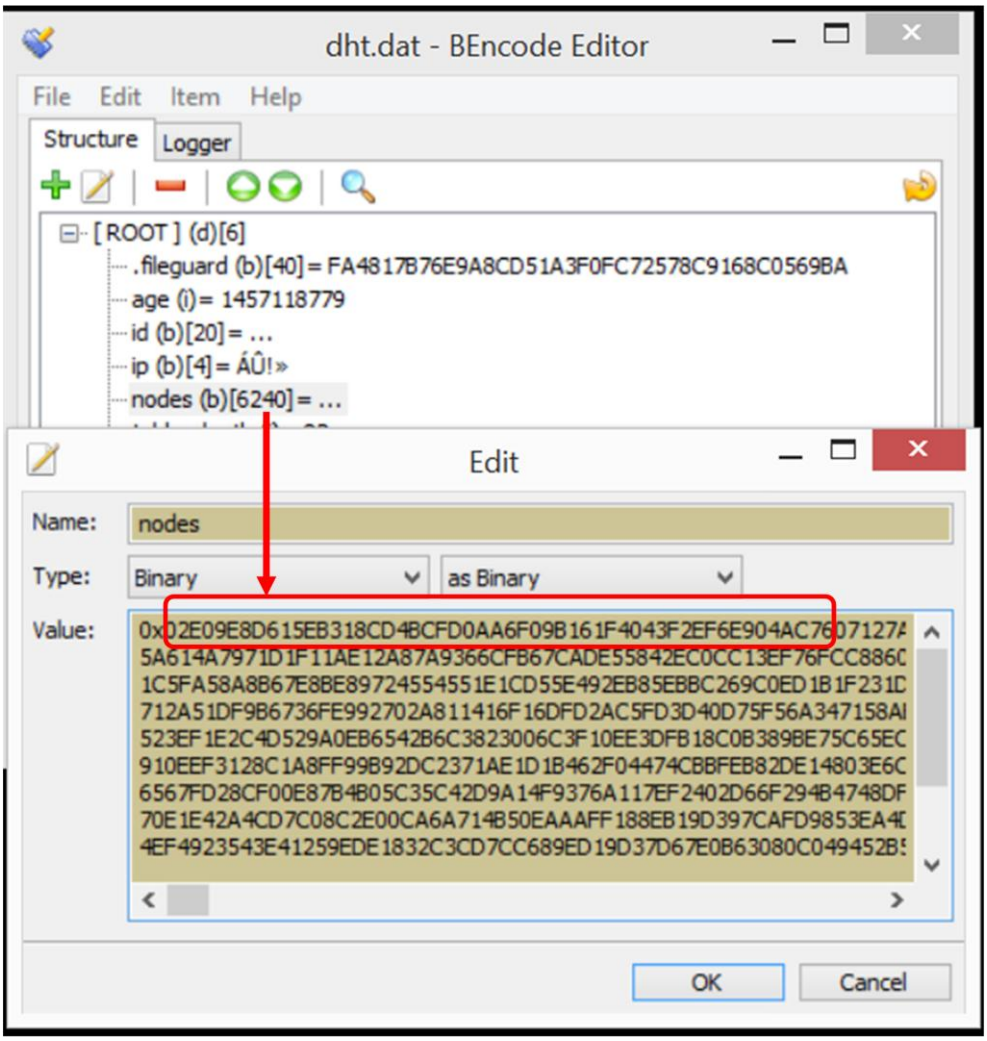
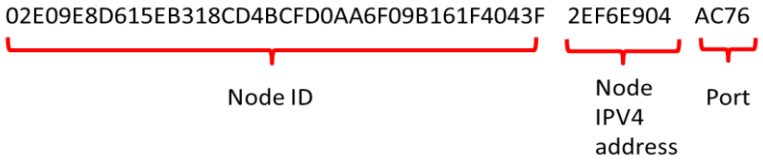


Figure 3. Key nodes (b) in the dht.dat file.

The key nodes (b) are decoded as 26-byte (52 hexadecimal digits) IP addresses for every node:



We found 240 IP addresses (6240/26 = 240). The first four of them were decoded and they are presented in Table 4.

Table 4. Decoded IP addresses of the file dht.dat.

ID	IPV4	Port
0x02E09E8D615EB318CD4BCFD0AA6F09B161F4043F	46.246.233.4	44,150
0x07127AE0BC8CEA311D677C7FFBC5DEBF074D5505	61.98.50.155	34,292
0x05E2BE8EE8331F657E7DF9C7DCBD054F57075D10	102.33.45.249	36,833
0x0F10F4DBF9F041C1A72922202281A1F1F66E7DC7	89.117.201.157	26,085

The torrent file Eugenijus DSC04967.jpg.torrent file was identical to the torrent file of the third step.

Investigation of the file resume.dat revealed that the value of the key peers6 increased to 216, when the value at the third step of the experiment was 54. Consequently, we obtained 12 IP addresses, which were decoded and presented in Table 5.

Table 5. Decoded IP addresses in the file resume.dat.

IPV6	Port	IPV4	Port
0x00000000000000000000	0xFFFF	10.10.0.21	26,276
0x00000000000000000000	0xFFFF	127.0.0.1	26,276
0x00000000000000000000	0xFFFF	10.10.0.22	52,559
0x00000000000000000000	0xFFFF	193.219.33.187	52,559
0x200100005EF579FD3487	0x2187	62.36.222.68	52,559
0x00000000000000000000	0xFFFF	30.81.87.17	32,861
0x00000000000000000000	0xFFFF	96.85.153.149	34,432
0x00000000000000000000	0xFFFF	163.98.220.19	35,522
0x00000000000000000000	0xFFFF	229.96.30.151	36,869
0x00000000000000000000	0xFFFF	230.37.30.229	54,277
0x00000000000000000000	0xFFFF	40.50.97.97	55,879
0x00000000000000000000	0xFFFF	107.54.163.231	56,970

To summarize the fourth step of the experiment we conclude that the file dht.dat contains the IP addresses of many nodes to aid in the process of downloading the file. The number of participating peers during the file download when the virtual machines were connected to the Internet in the file resume.dat was higher than when the connection was disabled.

During the experiment, we experienced many times when it was quite inconvenient to examine the BEncoded BitTorrent configuration files. Therefore, we propose to implement a tool to locate the BitTorrent client artifacts and to present them in a human-readable format. In the next section, we review the implementation of this tool and provide illustrative results of its use.

4. Tool to Locate BitTorrent Client Artifacts and Results of the Experiment

The BEncode Editor is designed to view and edit the encoded BitTorrent files. However, our experiment revealed that many additional utilities are required to feel comfortable with the data opened in BEncode Editor. Therefore, we designed and implemented the tool to present all the valuable data for the forensic investigator in a human-readable format. The choice of the data to be presented is based on our experience during the experiment presented in the previous section. Therefore, we included all the data presented in Tables 1–3. Additionally we decided to include the data from a file settings.dat. The file settings.dat stores the information about configuration parameters of the BitTorrent client. A lot of keys are used, and many of them are not empty. However, only a few of the keys of the file settings.dat are of interest to the forensic investigator. We report them in Table 6.

Table 6. Selected keys of settings.dat file.

Key	Comment
bind_port (i)	Port of the connection
gui.last_bundle_visit (i)	Time of the last connection–UNIX timestamp
isp.peer_policy_expy (i)	Time of the policy expiration–UNIX timestamp
labelDirectoryMap (d) (3) audio (b) (24) documents (b) (28) video (b) (25)	Directories for audio, documents and video files
labelRuleMap (d) (3) audio (b) (18) documents (b) (21)	Rules for audio, documents and video files
video (b) (18)	
settings_saved_systime (i)	Settings saved system time–UNIX timestamp

For all the presented files, we collected the following data concerning the file creation and access from the file system:

- file creation date;
- file modification date;
- file access date;

The most suitable format to represent data with the format of keys and their values is an Extensible Markup Language (XML). XML presents a simple, very flexible text format to meet the challenges of large-scale electronic publishing [24]. XML format looks much like the HTML format used to represent Internet pages. However, XML is much more flexible, since it allows the introduction of one's own tags. XML documents can be viewed in many browsers. Special programs have been created to view and edit XML documents, like XML Notepad [25], which displays friendly colorful views.

We decoded the values of all the keys presented in Tables 1–3 and 6 and we placed them into the XML document. The names of the tags will consist of composite names with the file name and then the name of the key.

When a cybercrime is committed using a BitTorrent utility, it is helpful for the forensic investigator to know all the peers who shared the copyrighted or otherwise illegal material. Therefore, we have decided to aid the forensic investigator and supply additional information on the basis of the decoded values. That is, we verify every IP address taken from considered files in the open Whois database [26]. This database allows the IP provider to be obtained for every IP address. This information enables the forensic investigator to contact the IP provider to find out who is using the particular IP address.

Additionally, we used the GeoLite2 [27] database that provides for the IP address in question the country name, subdivision name, postal code, and two coordinates on the map: latitude and longitude. These coordinates show the location of the IP address considered. These coordinates are not very accurate; however, they enable the forensic investigator to guess the approximate location of the IP address considered.

The tool is implemented as a command line utility with three parameters: the letter of the disk drive to search for, the full path where to place the report file, and the name of the report file. The initial plan was to include the created utility in a larger open program system, for example Autopsy [28].

One of the main challenges was to decode the BEncoded file. Figure 4 shows the hexadecimal view and illustrates the process of the file decoding.

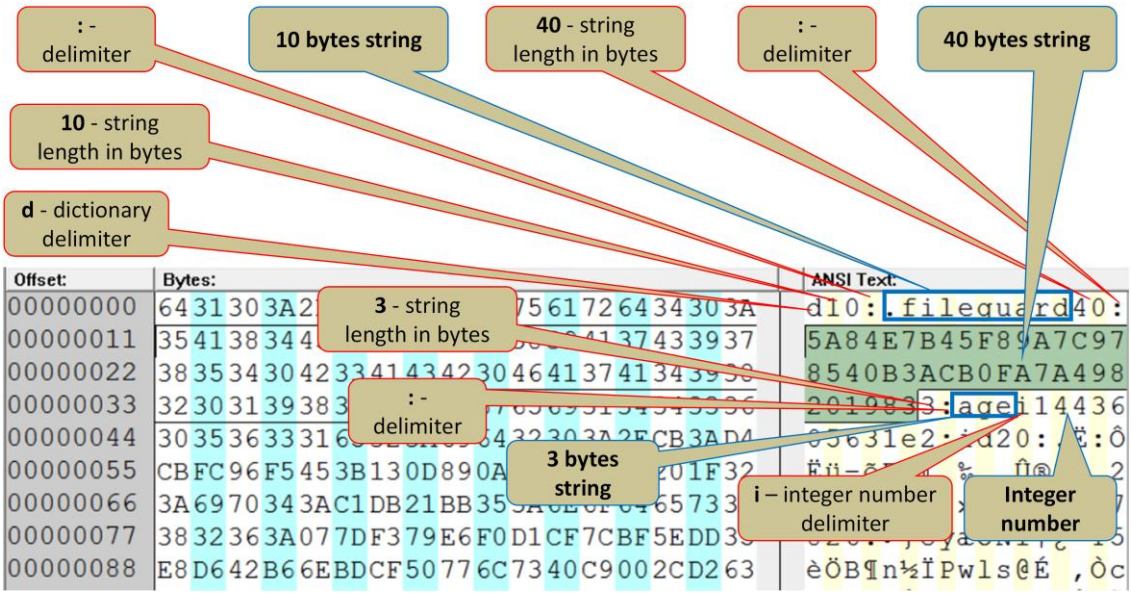


Figure 4. Decoding of the BEncoded file—hexadecimal view.

The C# programming language was chosen as the programming language of the implementation. Firstly, the programming language is supported by a rich .NET platform. Next, this programming language is modern and it has features enabling easy analysis of text files. One such features is a dictionary data type. This data type provides a mapping from the set of keys to the set of values.

The architectural view of the program, which provides more implementation details, is shown in Figure 5.

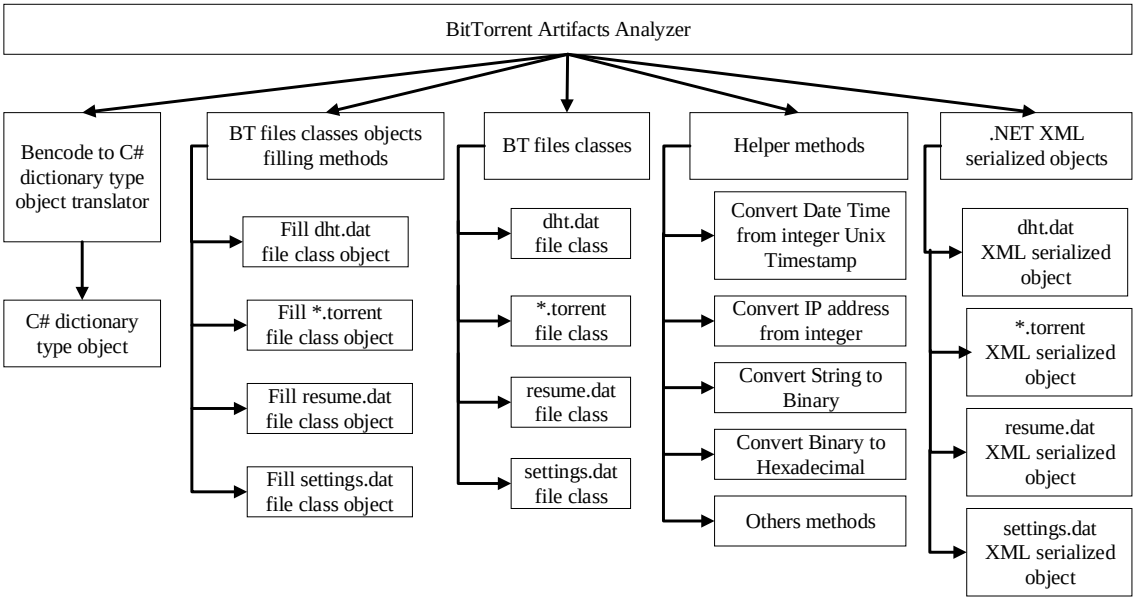


Figure 5. The architecture of the program.

The presented architecture (Figure 5) implements all the ideas presented so far in Section 3. The tool searches for the family of BitTorrent files on the supplied disk drive and presents their contents in the XML format. The family of BitTorrent files consists of the following files: dht.dat, resume.dat, settings.dat, and *.torrent. The first three files are common to BitTorrent client users. The *.torrent file is separate for every downloaded file (Figure 6).


```
- <L3CEBitTorrentAnalyzerStart Started="L3CE BitTorrent Analyzer Start: 2016-03-23 17:46:59+02:00">
  - <Users>
    - <User UserName="Eugenijus">
      + <BitTorrent_dht.dat_file_information dhtFileName="G:\Users\Eugenijus\AppData\Roaming\BitTorrent\dht.dat"></BitTorrent_dht.dat_file_information>
      - <BitTorrent_resume.dat_file_information resumeFileName="G:\Users\Eugenijus\AppData\Roaming\BitTorrent\resume.dat">
          <resumeFileCreated>2016-03-16T16:19:52.879292+02:00</resumeFileCreated>
          <resumeFileModified>2016-03-16T15:02:41.4117864+02:00</resumeFileModified>
          <resumeFileAccessed>2016-03-16T16:19:52.879292+02:00</resumeFileAccessed>
        + <resumeDownloadedFiles></resumeDownloadedFiles>
        </BitTorrent_resume.dat_file_information>
      + <BitTorrent_settings.dat_file_information settingsFileName="G:\Users\Eugenijus\AppData\Roaming\BitTorrent\settings.dat">
        </BitTorrent_settings.dat_file_information>
      + <BitTorrent_.torrent_file_information torrentFileName="G:\Users\Eugenijus\AppData\Roaming\BitTorrent\Eugenijus DSC04967.jpg.torrent">
        </BitTorrent_.torrent_file_information>
      + <BitTorrent_.torrent_file_information torrentFileName="G:\Users\Eugenijus\AppData\Roaming\BitTorrent\Ubuntu 710 Desktop Live CD.torrent">
        </BitTorrent_.torrent_file_information>
      + <BitTorrent_.torrent_file_information torrentFileName="G:\Users\Eugenijus\AppData\Roaming\BitTorrent\Ubuntu Eee 8.04.1.torrent">
        </BitTorrent_.torrent_file_information>
      + <BitTorrent_.torrent_file_information torrentFileName="G:\Users\Eugenijus\AppData\Roaming\BitTorrent\ubuntu-15.04-desktop-amd64.iso.torrent">
        </BitTorrent_.torrent_file_information>
      + <BitTorrent_.torrent_file_information torrentFileName="G:\Users\Eugenijus\AppData\Roaming\BitTorrent\VirtualBox - BlackBox Linux 2 i386 Virtual Appliance -
        [VirtualBoxImages.com].torrent"></BitTorrent_.torrent_file_information>
      + <BitTorrent_.torrent_file_information torrentFileName="G:\Users\Eugenijus\AppData\Roaming\BitTorrent\ZveeDee-EyesVirtualBox_Xubuntu-
        9.10_SCREENSHOTS-[VirtualBoxImages.com].torrent"></BitTorrent_.torrent_file_information>
      </User>
    </Users>
    <L3CEBitTorrentAnalyzerFinish Finish="L3CE BitTorrent Analyzer Finished: 2016-03-23 18:09:38+02:00"/>
</L3CEBitTorrentAnalyzerStart>
```


Figure 6. View of the results in the Mozilla Firefox browser.

Figure 6 presents a contracted view of the results produced by the tool. As we can see, the current user downloaded six files using the BitTorrent client. The information for every item can be expanded. The fully expanded file of the results is quite lengthy. We provide two small excerpts from this file in Figures 7 and 8.

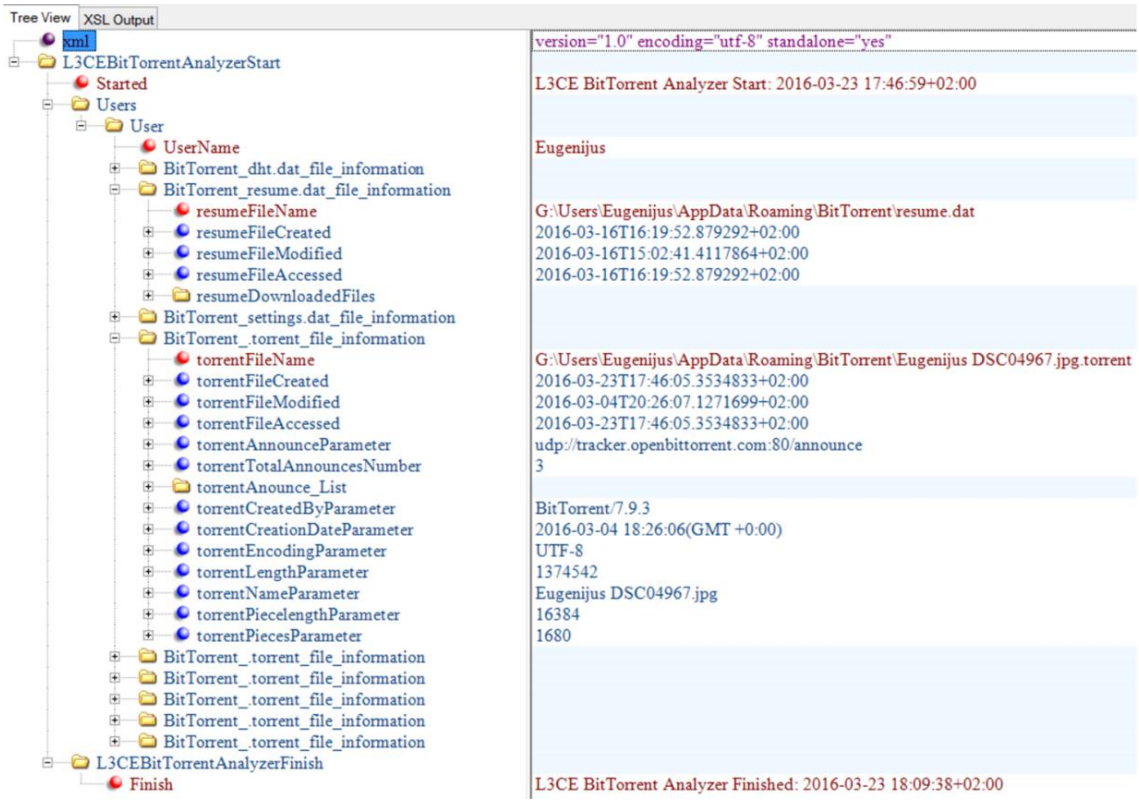


Figure 7. View of the results in the XML Notepad editor.

Figure 8. View of the results in the Internet Explorer browser.

Figure 7 represents a small part of the results file in XML Notepad editor, namely the file Eugenijus DSC04967.jpg.torrent. We see the friendly view of the results, where the provided information is separated into two windows. The left window shows our chosen XML tags; meanwhile the right window provides the information held in our chosen XML tags. The XML Notepad editor hides some details of the XML implementation, for example, it does not show the ending tags.

Figure 8 represents an excerpt from the same file as in Figure 7 in the Internet Explorer browser; however, the part of the file is different. In this figure, we provide the expanded geolocation information that is supplied additionally by our tool according to the IP address found in the peers list. Now, we see all the details of the XML implementation. Moreover, we can use the browser capabilities to search the XML text, to either contract or expand the XML tree. The browser provides a more technical view than the XML Notepad editor that offers more colorful, more abstract and friendlier view.

Finally, we summarize all the features of our implemented tool and compare them with possible competitors. We know two possible competitors: the BEncode Editor [20] and the Plaso framework [22]. The results of the comparison are reported in Table 7.

Table 7. Summary of features and comparison with possible competitors.

Feature	BEncode Editor	Framework Plaso	Our Tool
Deserializes BEncoded file; produces a dictionary containing BEncoded data	+	+	+
Extracts event object from the values of entries within a BEncoded file	+	+- The plugin should implement the logic	+
Parses the file resume.dat to extract the keys and to retrieve their values	Manually	+	+
Parses the file *.torrent to extract the keys and to retrieve their values	Manually	+	+
Parses the file dht.dat to extract the keys and to retrieve their values	Manually	-	+
Parses the file settings.dat to extract the keys and to retrieve their values	Manually	-	+
Applies semantics to convert values of keys into investigator readable form	-	-	+
Produces XML report including extracted values and keys from files: dht.dat, resume.dat, settings.dat, *.torrent	-	-	+
Extracts binary values of IP addresses from dht.dat “nodes” key and converts them into human readable form	-	-	+
Extracts binary values of IP addresses from resume.dat “peers” key and converts them into human readable form	-	-	+
Gets owner information for every extracted IP address holder from RIPE database	-	-	+
Gets geolocation information for every extracted IP address holder from GeoLite2City database	-	-	+
Finds all computer users who were using BitTorrent client and generates joint XML report	-	-	+

We see from Table 7 that our implemented tool provides many useful features for the forensic investigator. Our possible competitors do not have many of these features. Our tool is implemented as a stand-alone tool and it can be integrated into a larger framework like Autopsy or Plaso.

5. Conclusions

The BitTorrent application is a popular tool dedicated for downloading large files from the Internet. It is a very effective instrument to download large video and audio files. However, this powerful instrument can be employed to commit cybercrimes, like sharing of copyrighted movie files, child pornography, and others. In order to help forensic investigators to fight against such cybercrimes, we suggested a methodology to locate the artifacts left by the BitTorrent client on the local computer.

According to the suggested methodology, we carried out an experiment. We can conclude that the BitTorrent client leaves artifacts on the local computer that can be used to reveal the performed activity. One of the main findings is that the BitTorrent configuration file resume.dat contains a list of peers who participated in the sharing of the file. The next important finding is that the file resume.dat accumulates a list of the peers who participated for every downloaded file separately.

During the experiment, we found that the current tools do not provide an easy way to view and analyze the data in the BitTorrent configuration files. Therefore, we designed and implemented a tool that searches the local computer and locates artifacts related to the use of the BitTorrent client. We used the newest version of the BitTorrent client application during the experiment. The tool finds the present BitTorrent files, decodes them into a readable format, extracts the important information for the forensic investigator and writes the results in XML format into a single file. Additionally, the tool delivers the new supplementary information for all the IP addresses found in the peer lists. That is, the tool finds the IP provider and the location coordinates for every IP address. The provided additional information will aid the forensics investigator to locate other users who participated in committing the cybercrime.

The results file can be viewed and analyzed either in a special XML editor or in any browser. A special XML editor like XML Notepad provides a more colorful and more abstract view than the browser. The browser provides a view that is more technical without hiding details of the implementation of the XML format.

This methodology for the investigation of BitTorrent client artifacts can be applied to other BitTorrent clients as well. New BitTorrent clients can be included in the automated tool. This methodology can be applied to other operating systems, as well.

Supplementary Materials: The source code of the BitTorrent artifacts analyzer is available online at www.mdpi.com/2073-8994/8/6/40/s1.

Acknowledgments: The investigation presented in this paper has been carried out partially under the project "Lithuanian Cybercrime Centre of Excellence for Training, Research and Education", Grant Agreement No. HOME/2013/ISEC/AG/INT/4000005176, co-funded by the Prevention of and Fight against Crime Programme of the European Union.

Author Contributions: Algimantas Venčkauskas provided general guidance for the research; Vacius Jusas wrote the paper; Kęstutis Paulikas contributed to the initial design of the experiment and to the interpretation of the results of the experiment; Jevgenijus Toldinas carried out the experiment and implemented the tool.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Legout, A.; Liogkas, N.; Kohler, E.; Zhang, L. Clustering and sharing incentives in BitTorrent systems. In Proceedings of the 2007 ACM Sigmetrics International Conference on Measurement and Modeling of Computer Systems; Association for Computing Machinery: New York, NY, USA, 2007.
2. Park, S.; Chung, H.; Lee, C.; Lee, S.; Lee, K. Methodology and implementation for tracking the file sharers using BitTorrent. *Multimed. Tools Appl.* **2015**, *74*, 271–286. [CrossRef]
3. Schmidt, A.H.; Antunes, R.S.; Barcellos, M.P.; Gaspary, L.P. Characterizing Dissemination of Illegal Copies of Content through Monitoring of BitTorrent Networks. In Proceedings of Network Operations and Management Symposium (NOMS), Maui, HI, USA, 16–20 April 2012.
4. Liberatore, M.; Erdely, R.; Kerle, T.; Levine, B.N.; Shields, C. Forensic investigation of peer-to-peer file sharing networks. *Digit. Investig.* **2010**, *7*, S95–S103. [CrossRef]
5. Atlidakis, V.; Roussopoulos, M.; Delis, A. EnhancedBit: Unleashing the potential of the unchoking policy of the bittorrent protocol. *J. Parallel Distrib. Comput.* **2014**, *74*, 1959–1970. [CrossRef]
6. Mansilha, R.B.; Bays, L.R.; Lehmann, M.B.; Mezzomo, A.; Gaspary, L.P.; Barcellos, M.P. Observing the BitTorrent Universe through Telescopes. In Proceedings of International Symposium on Integrated Network Management, Dublin, Ireland, 23–27 May 2011.
7. Farina, J.; Scanlon, M.; Kechadi, M.-T. BitTorrent Sync: First impressions and digital forensic implications. *Digit. Investig.* **2014**, *11*, S77–S86. [CrossRef]

8. Scanlon, M.; Farina, J.; Kechadi, M.-T. BitTorrent Sync: Network Investigation Methodology. In Proceedings of Ninth International Conference on Availability, Reliability and Security (ARES 2014), Fribourg, Switzerland, 30 September 2014.
9. Scanlon, M.; Farina, J.; Kechadi, M.-T. Network investigation methodology for BitTorrent Sync: A Peer-to-Peer based file synchronisation service. *Comput. Secur.* **2015**, *54*, 27–43. [CrossRef]
10. The Register. Windows 8.1 Becomes World's Fourth-Most-Popular Desktop OS. Available online: http://www.theregister.co.uk/2014/02/03/windows_81_becomes_worlds_fourthmostpopular_desktop_os/ (accessed on 20 November 2015).
11. Chu, H.-Ch.; Wang, G.-G.; Park, J.H. The digital fingerprinting analysis concerning google calendar under ubiquitous mobile computing era. *Symmetry* **2015**, *7*, 383–394. [CrossRef]
12. Adelstein, F.; Joyce, R.A. File marshal: Automatic extraction of peer-to-peer data. *Digit. Investig.* **2007**, *4*, S43–S48. [CrossRef]
13. P2P Marshal. Available online: http://forensicswiki.org/wiki/P2PMarshal (accessed on 29 November 2015).
14. Woodward, A.J.; Valli, C. Do Current Erasure Programs Remove Evidence of BitTorrent Activity? In Proceedings of Conference on Digital Forensics, Security and Law, Arlington, VA, USA, 18–20 April 2007.
15. Woodward, A. Do Current BitTorrent Clients running on Windows 7 beta leave behind meaningful data? In Proceedings of International Conference on Security and Management, Las Vegas, NV, USA, 13–16 July 2009.
16. Lallie, H.S.; Briggs, P.J. Windows 7 registry forensic evidence created by three popular BitTorrent clients. *Digit. Investig.* **2011**, *7*, 127–134. [CrossRef]
17. Acorn, J.; Austin, J. Forensic Studies in BitTorrent. Produced by the Information Security Group at Royal Holloway, University of London in conjunction with TechTarget. 2008 TechTarget. Available online: http://cdn.ttgtmedia.com/searchSecurityUK/downloads/RH6_Acorn.pdf (accessed on 23 November 2015).
18. Venčkauskas, A.; Jusas, V.; Paulikas, K.; Toldinas, J. Investigation of artifacts left by bittorrent client on the local computer operating under Windows 8.1. *Inf. Technol. Control* **2015**, *44*, 451–461. [CrossRef]
19. BitTorrent org. The BitTorrent Protocol Specification. Available online: http://bittorrent.org/beps/bep_0003.html (accessed on 12 November 2015).
20. Ultima's Projects. Available online: https://sites.google.com/site/ultimasites/bencode-editor (accessed on 23 November 2015).
21. Chabot, Y.; Bertaux, A.; Nicolle, Ch.; Kechadi, T. An ontology-based approach for the reconstruction and analysis of digital incidents timelines. *Digit. Investig.* **2015**, *15*, 83–10. [CrossRef]
22. Plaso. Available online: http://plaso.readthedocs.org/projects/plaso-api/en/latest/plaso.parsers.bencode_plugins.html (accessed on 9 December 2015).
23. BitTorrent.org. Download BitTorrent. Available online: http://www.bittorrent.com/downloads/complete/track/stable/os/win/ (accessed on 12 November 2015).
24. W3C.org. Extensible Markup Language (XML). Available online: http://www.w3.org/XML/ (accessed on 12 November 2015).
25. CodePlex. XML Notepad. Available online: https://xmlnotepad.codeplex.com/ (accessed on 12 November 2015).
26. RIPE NCC. RIPE database query. Available online: https://apps.db.ripe.net/search/query.html (accessed on 12 November 2015).
27. MaxMind. GeoLite2 Free Downloadable Databases. Available online: http://dev.maxmind.com/geoip/geoip2/geolite2/ (accessed on 16 November 2015).
28. The Sleuth Kit. Autopsy. Available online: http://www.sleuthkit.org/autopsy/ (accessed on 12 November 2015).