IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 3 Issue 9, September 2016
ISSN (Online) 2348 – 7968 | Impact Factor (2015) - 4.332
www.ijiset.com

A Survey of Covert Channels in BitTorrent Network

Bin Gao, Jiangtao Zhai*

School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu, 212003, China

Abstract

Covert channels are used for the secret transfer of information, which aim to hide the very existence of the communication. Initially, covert channels were concentrated in computer network protocols for huge flux data transportation of Internet. Recently, with the extensive application of the Peer-to-Peer (P2P) network, focus has shifted towards covert channels in P2P protocols. In these protocols, BitTorrent (BT) protocol is the most popular one due to its high-speed and huge-flux file-sharing service, which help BT network seems ideal as a high-bandwidth vehicle for covert communication. This article is a survey of the existing techniques for creating covert channels in BT protocols. We also give an overview of common methods for their concealment, embedded capacity, required to improve security in future BT network.

Keywords: *Covert Channel, BT Network, Concealment, Embedded Capacity.*

1. Introduction

Often it is thought that the use of encryption is sufficient to secure communication. However, encryption only prevents unauthorized parties from decoding the communication. In many cases the simple existence of communication or changes in communication patterns, such as an increased message frequency, are enough to raise suspicion and reveal the onset of events. Therefore, covert channel techniques which aim to hide the very existence of the communication appeared [1-6]. Typically, covert channels use means of communication not normally intended to be used for communication, making them quite elusive.

Nowadays, P2P networks [7], such as BitTorrent [8-10], Gnutella, and eDonkey, have become irreplaceable media for information dissemination and sharing over the Internet due to its extensive application. The huge amount of data exchanged through P2P networks and the global-scale of P2P infrastructure creates an environment where communication has a potential to be hidden among the mass of regular users and corresponding host message exchanges. BitTorrent network which inherited the characteristics of high speed and huge flux of P2P networks is extensively used throughout the world.

In BitTorrent network, researchers find that BT protocols offer several covert channels, through which hidden information can be transmitted. Understanding existing covert channel techniques is crucial in developing countermeasures. The detection and elimination of covert channels are challenging but need to be addressed to secure future BitTorrent network. Therefore, this article is a survey of the existing techniques for creating covert channels in BT protocols, and the rest of the article is organized as follows. Firstly, we define the terminology and explain the basic communication principles of BitTorrent network. Then, currently known covert channel techniques and their covert performance in BT protocols will be discussed. Finally, we conclude and identify future research.

2. The Structure of BitTorrent Network

BitTorrent network is a file distribution system based on BT protocols. In the system, there are five main entities, namely, a torrent file, a tracker server, a web browser, and two peers. Furthermore, these entities are divided into three categories as shown below.

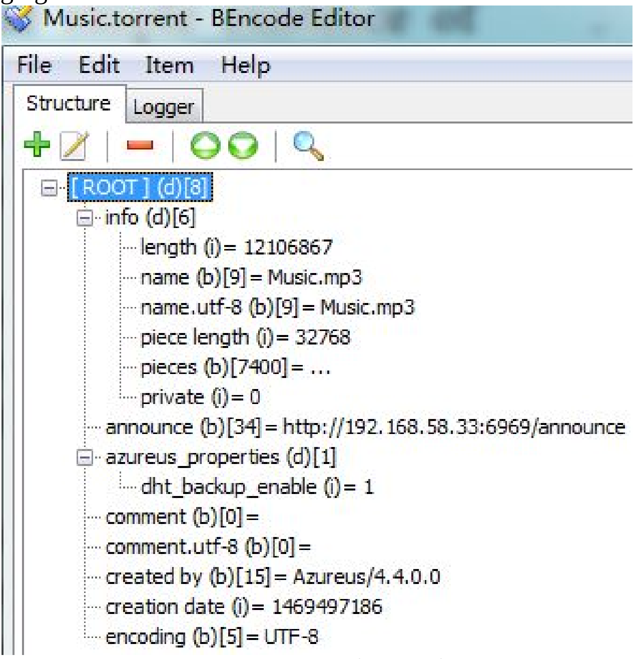
2.1 Torrent File

The torrent file, a vital part of BitTorrent network, is also called metafile, which is the beginning of file-sharing and information dissemination. The torrent file is a static file with the extension '.torrent' and is put on an ordinary web browser by the original peer. It is used to store the information of shared file, its length, name, hashing value, the URL of a tracker server, etc. which is indicated in Figure 1.

2.2 Tracker Server and Web Browser

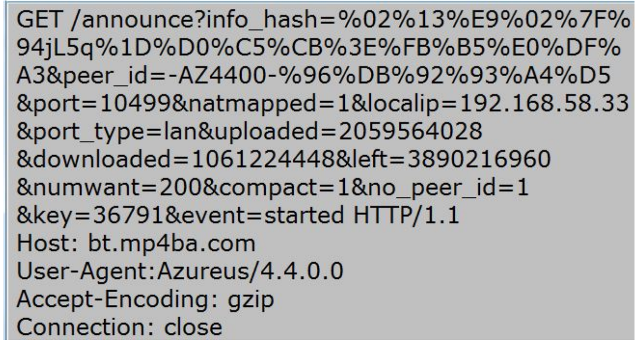
The BitTorrent tracker system, composed of a tracker server and a web browser, is based on the BitTorrent tracker protocol which functions as a transmitter to exchange the information between a tracker server and peers. On the one side, as shown in the torrent file above, the web browser is storage of metafiles. Peers not only can upload metafiles to the site, but also can download them from it. On the other side, the tracker server commonly hosts lists of peers for multiple content files, and it allows users to not only add content to a swarm but also to retrieve the complete list of content "monitored" by a tracker. Thus one peer is able to connect to some appropriate peers in the list, which can greatly improve the efficiency of file-sharing and information dissemination in BitTorrent network.

The BitTorrent tracker protocol operates over HTTP. When a peer downloads and opens a metafile, its BT client will perform a GET request to the tracker in order to get the information of peers currently seeding or leeching of the shared file. Figure 2 shows an example announce request to bt.mp4ba.com. This GET request contains the info hash from the metafile, the peer id, IP and port number of the client, etc. Afterwards, the tracker will respond to the client's request in a standard HTTP format, which is indicated in Figure 3. In the response, the tracker sends a list of IP, port, and peer id of clients currently related to the shared file. Additionally, some trackers provide this information under the form of an aggregated file like the text file.



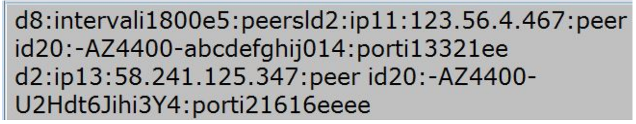
```
[ROOT] (d)[8]
  info (d)[6]
    length (i)= 12106867
    name (b)[9]= Music.mp3
    name.utf-8 (b)[9]= Music.mp3
    piece length (i)= 32768
    pieces (b)[7400]= ...
    private (i)= 0
  announce (b)[34]= http://192.168.58.33:6969/announce
  azureus_properties (d)[1]
    dht_backup_enable (i)= 1
  comment (b)[0]=
  comment.utf-8 (b)[0]=
  created by (b)[15]= Azureus/4.4.0.0
  creation date (i)= 1469497186
  encoding (b)[5]= UTF-8
```

Fig. 1 The structure of Torrent file



```
GET /announce?info_hash=%02%13%E9%02%7F%
94jL5q%1D%D0%C5%CB%3E%FB%B5%E0%DF%
A3&peer_id=-AZ4400-%96%DB%92%93%A4%D5
&port=10499&natmapped=1&localip=192.168.58.33
&port_type=lan&uploaded=2059564028
&downloaded=1061224448&left=3890216960
&numwant=200&compact=1&no_peer_id=1
&key=36791&event=started HTTP/1.1
Host: bt.mp4ba.com
User-Agent:Azureus/4.4.0.0
Accept-Encoding: gzip
Connection: close
```

Fig. 2 Tracker GET request



```
d8:intervali1800e5:peersld2:ip11:123.56.4.467:peer
id20:-AZ4400-abcdefghij014:porti13321ee
d2:ip13:58.241.125.347:peer id20:-AZ4400-
U2Hdt6Jihi3Y4:porti21616eeee
```

Fig. 3 Tracker dictionary response

2.3 Peers

Peers are the most active part in BitTorrent network, and the communication protocol between peers is called peer wire (PW) protocol which is an application layer protocol based on TCP. When a BT client is used by a user to join in BT network, the BT client is just a peer. The peer is not only a server but also a client, thus in order to identify them, peers that are actively downloading files are known as leechers, while peers that have completed downloading a file but remain uploading to other peers are known as seeders.

In the PW protocol, a lot of application layer messages are defined, *Handshake*, *Bitfield*, *Request*, *Piece*, *Have*, etc. Figure 4 is a communication timing diagram of two peers in BT network and Table 1 gives a brief introduction of these messages. After the connection between the two peers is established, *Handshake* messages will be firstly exchanged. Then the two sides can be aware of the occupation rate of the same shared file of each other by interchanging *Bitfield* messages. Afterwards, the two peers will send *Request* messages to each other for their interested data blocks. Then they will respond to the *Request* message though transmitting *Piece* messages which represent data blocks of the shared file. After *Piece* messages are received, *Have* messages is intended to be transferred to claim that the peer have a certain *Piece* message. Finally, the two sides will conduct the resource exchange in the message loop of *Request-Piece-Have* until the shared file has been downloaded.

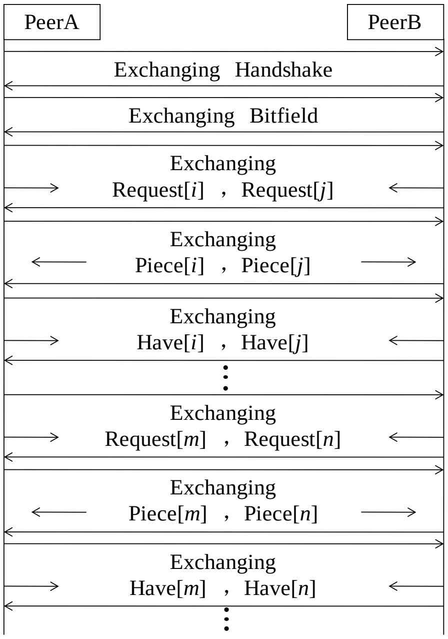


Fig. 4 Communication timing diagram between two peers in BT network

Table 1: The introduction of BT messages

BT Messages	Information/Function
Handshake	The brief Certification Info of peers.
Bitfield	The occupation rate of the shared file.
Keep-alive	Maintaining the connection.
Choke	Prohibiting peers downloading from here.
Unchoke	Permitting peers downloading from here.
Interested	Interested in other peers.
Uninterested	Uninterested in other peers.
Request	Requesting to peers for some data block.

Cancel	Ignoring the request for data blocks.
Piece	A data block of the share file.
Have	Declaring having some data blocks.

In summary, a seeder, a tracker server, a web browser, and a leecher, they cooperate with each other to complete the transfer the shared resource like the file (Music.mp3) which is shown in Figure 5.

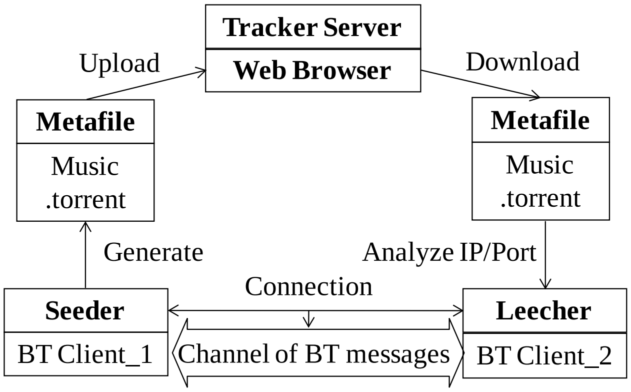


Fig. 5 The structure of BT shared file distribution system

3. Covert Channels in BitTorrent Protocols

In this section, existing covert channel techniques in BT protocols are grouped according to their information hiding mechanism. We believe that this approach is advantageous because it provides a more fine-grained classification and also accommodates the fact that some methods can be adjusted and applied to other situations.

3.1 Optional Field Reusage

Covert channels can be encoded in optional field which is unused or reserved. This is particularly problematic if BT protocol standards do not mandate specific values or receivers do not check for the standard values.

Li et al. proposed a BT covert channel called Field Reusage (FR) using the unused field of the torrent file [11]. This scheme is based on the knowledge that torrent fields are redundant in structure. There are some optional fields with an inconspicuous occurrence in normal situation and these fields can be reused to embed stego-message. These corrupted torrent files have no impact on the overall functionality of the file-sharing and the communication parts can extract stego-message from them successfully. From the structure of the metafile presented in Figure 1, it is obvious that only three fields are suitable for embedding in the approach FR, they are comment, created-by, and creation-date. According to its research, the embedded capacity of the method is big for that it can reach to the average value 35 KB. However, although this method can avoid detection of the metafile format, it will raise the suspicious of the detector in the visibility of human eyes, so its concealment is weak.

In addition, Li et al. also proposed two other BT covert channels, the first one is known as Bitfield Field Reusage (BFR) covert channel by reusing the payload field of the Bitfield message, and the second one is called Piece Field Reusage (PFR) covert channel by reusing the payload field of the Piece message [12]. On the one hand, as described in Table 1, the load of the Bitfield message describes the data that the peer already has, the top bit of the first byte represents the first block data, followed by the next bit of the next block. If a certain bit is ‘0’, it is indicated that the peer does not have the corresponding data block, otherwise, the peer has a corresponding data block. On the other hand, the load of the Piece message is just the block data of the shared file because a shared file is divided into lots of data blocks. BFR and PFR all make use of the redundancy of their payload, but the embedded capacity of PFR is bigger than BFR on account of its direct substitution method, while the concealment of BFR is stronger than PFR owing to its matrix encoding method. Thus their covert performance all needs to be improved.

Furthermore, Desimone et al. proposed a BT covert channel by reusing the Peer_id field of the BitTorrent tracker protocol presented in Figure 2 [13]. Similarly, Cunche et al. also proposed a BT covert channel by reusing the Info_hash field of the BitTorrent tracker protocol [14]. The field in tracker GET request is used to embed stego-message, and the tracker dictionary response in the form of TXT shown in Figure 3 is used to extract stego-message. The database of the tracker server functions as the interchange station of the hiding information (Fig. 6). In contrast, they have the same embedded capacity basically, while the concealment of the Peer_id method is stronger than the Info_hash for the randomness of the 12 bytes in the back of the Peer_id field. Thus the Peer_id method is superior to the Info_hash method.

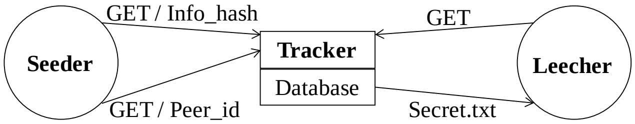


Fig. 6 Covert channels in the BT tracker system

3.2 Letter Case Change

The secret information can be encoded into the channel by the method of Letter Case Change (LCC). Li et al. proposed another BT covert channel called LCC using the Uniform Resource Locator (URL) field of the torrent file [11]. LCC is based on the knowledge that the URL in torrent files is insensitive to the written states of letters. When the written states of some letters are modified arbitrarily, it does not affect the work of BT downloading and peers can connect to the tracker server accurately with the modified URL. Then the receiver could successfully extract stego-message from the stego-torrent. From the structure of the metafile presented in Section 2, the announce field is the only one field suitable for embedding in the approach LCC. In order to enhance the imperceptibility of this approach, LCC just uses the first letter in word in uppercase to denote ‘1’ and the first letter in word in lowercase to denote ‘0’, Therefore, LCC does not increase the file size and has no impact on the overall functionality of the network, its concealment is stronger than FR. However, its embedded capacity is small owing to depending on the number of URL, and the maximum value of the embedded capacity of the method is just 69 bits according to its research.

3.3 Sequence of BT Messages

A set of n packets can be arranged in any n! ways. This approach requires per BT packet sequence numbers to determine the original packet order. Instead of actually sorting packets the method only modifies sequence numbers, thus keeping payload sent across multiple packets intact.

Eidenbenz et al. proposed a Request Order (RO) covert channel based on sorting sequence numbers of Request messages [15]. As shown in Figure 4, over each link, a peer will send one Request for a Piece block at a time to a certain peer, wait until the block has been transmitted completely, and then send the next Request to the same peer. Thus each Request message has a sequence number which is called index, and this approach makes use of the Request order channel to exchange messages in secrecy. RO can permute the order of this Request sequence to transmit $[\log(|B|!)]$ bits, where input B is the block sequence number of the shared file. As an example, the decimal number 17, which represents the binary message 100012, has a factorial number representation of 2210! because $0\times 0!+1\times 1!+2\times 2!+2\times 3!=17$. Compared with the approach LCC, the embedded capacity of this method is bigger. However, the RO cover channel does not take into account the statistical characteristics of sequence numbers of Request messages in BT network, and the order of each sequence is forced to adjusted, thus its concealment is weak.

4. Conclusions and Future Work

We have introduced the idea of BT shared file distribution system and have given an overview of current covert channel techniques in BitTorrent network protocols. By comparison, it is found that many existing BT covert channels are faced with the contradiction between concealment and embedded capacity, thus seeking balance between them is an eternal topic.

There are a number of directions for further research. Firstly, there is a lack of work on countermeasures of detection and elimination that is specifically aimed at BT covert channels. Secondly, real world experience shows that industry products still lack methods to deal with BT covert channels for that their applicability in real high-speed networks is questionable, thus it is essential to take into consideration the robustness of BT covert channels. Finally, it seems likely that single BT message is easy to be detected and its embedded capacity is limited, therefore, how to implement the scheme of multi-link BT covert channels will become a hotspot to study in the future.

Acknowledgments

This paper is partially supported by the National Natural Science Foundation of China (No.61472188, No.61602247), the Natural Science Foundation of Jiangsu province (BK20150472，BK20160840), and the CCF-Venustech “Hongyan” Science Foundation. In addition, thanks to the anonymous reviewers for their valuable comments which greatly helped improve the article.

References

[1] P. Dong, H. Qian, Z. Lu, et al. "A Network Covert Channel Based on Packet Classification", International Journal of Network Security, Vol. 14, No. 2, 2012, pp. 147-154.

[2] R. Archibald, D. Ghosal. "A Covert Timing Channel Based on Fountain Codes", in the International Conference on Trust, Security and Privacy in Computing and Communications, 2012, Vol. 18, pp. 970-977.

[3] J. Wu, Y. Wang, L. Ding, et al. "Improving Performance of Network Covert Timing Channel through Huffman Coding", Mathematical & Computer Modeling, Vol. 55, No. 55, 2012, pp. 69-79.

[4] X. Luo, E.W.W. Chan, P. Zhou, et al. "Robust Network Covert Communications Based on TCP and Enumerative Combinatorics", IEEE Transactions on Dependable & Secure Computing, Vol. 9, No. 6, 2012, pp. 890-902.

[5] H. Zhao, Y.Q. Shi. "Detecting Covert Channels in Computer Networks Based on Chaos Theory", IEEE Transactions on Information Forensics & Security, Vol. 8, No. 8, 2013, pp. 273-282.

[6] J. Kaur, S. Wendzel, M. Meier. "Countermeasures for Covert Channel-Internal Control Protocols", in the International Conference on Availability, Reliability and Security. 2015, Vol. 23, pp. 422-428.

[7] X.P. Lu, W.D. Wang, X.Y. Gong, et al. "A P2P Resource Sharing Mechanism for Hybrid Network", Journal of Beijing University of Posts and Telecommunications, Vol. 34, No. 4, 2011, pp. 113-117.

[8] H. Tang, L. Hu, H.Y. Zhu. "Survey of BitTorrent Network Behavior Studies", Journal of Chinese Mini-Micro Computer Systems, Vol. 33, No. 9, 2012, pp. 2002-2007.

[9] R. Nie, L. Nie, C.H. Liu, et al. "Measurement of the Characteristics of the BitTorrent Network", Journal of Beijing University of Posts and Telecommunications, Vol. 35, No. 3, 2012, pp. 125-128.

[10] Y. Yan, H. Tang. "Data Collection and Analysis of Traffic Flow between Peers in BitTorrent Network", Computer Science, Vol. 41, No. 6, 2014, pp. 75-78.

[11] Z.S. Li, X.M. Sun, B.W. Wang, et al. "A Steganography Scheme in P2P Network", in the International Conference on Intelligent

Information Hiding and Multimedia Signal Processing, 2008, Vol. 12, pp. 20-24.

[12] Z.S. Li. "Research of Information Hiding Technology Based on the BitTorrent Network", M.S. thesis, School of Computer and Communication, Hunan University, Changsha, China, 2009.

[13] J. Desimone, D. Johnson, B. Yuan, et al. "Covert Channel in the BitTorrent Tracker Protocol", in the International Conference on Security and Management, 2012, Vol. 23, pp. 223-226.

[14] M. Cunche, M.A. Kaafar, R. Boreli. "Asynchronous Covert Communication using BitTorrent Trackers", in the International Conference on High Performance Computing and Communications, 2014, Vol. 36, pp. 213-291.

[15] R. Eidenbenz, T. Locher, R. Wattenhofer. "Hidden Communication in P2P Networks Steganographic Handshake and Broadcast", IEEE INFOCOM, Vol. 2, No. 3, 2011, pp. 954-962.

First Author Bin Gao, born in 1992, received the B.E. degree in 2014 from Jiangsu University of Science and Technology. Nowadays, he is a graduate student in the University above and has contributed 2 core papers. His research interests include covert communication and network information security.

Corresponding Author Jiangtao Zhai, born in 1983, received the B.E., M.E. and Ph.D. degree respectively in 2006, 2008 and 2013 from Nanjing University of Science and Technology. At the moment, he is a lecturer in Jiangsu University of Science and Technology and has published more than 15 core papers. His research interests include multimedia and network information security.