4. BitTorrent and Attacks against It

- BitTorrent and Attacks against It

    - Motivation
    - Terms
        - Tracker
        - Seeds
        - Leechers
        - Swarm
        - .torrent
    - The BitTorrent protocol
        - Available pieces
        - Downloading
        - Piece strategy
    - Attacking BitTorrent
        - Harming the swarm

1. Motivation

Great, so welcome to Piracy 101!

Kidding a side, BitTorrent is an easy way to share (very) massive files.

2. Terms

BitTorrent is based on a number of different participants and actors, so let's quickly make the different aspects of it clear.

2.1. Tracker

BitTorrent is a semi decentralised system, with a single centralised component for discovery of other peers; this is called the tracker.

This tracker helps peers come in contact with each other. Doing so, is done completely random, among available peers.

- ***Tracker***
- *Help peers come in contact, at random*

- *Never shares torrent data*

The tracker is thereby only to connect with other peers, and gets out of the way after this step. But as always, there is an exception to the rule.

Some networks use privat trackers, where one have to be logged in. These networks often keep tracks of ones upload and download ratio, to reward good behaviour and punish bad. Thereby, this is a job for the tracker, why it keeps track of such.

2.2. Seeds

If a peer has the whole file in the storage, it is named as a Seeder, meaning that it can share every bit of the file.

- ***Seeder***
- *Has the whole file*
- *Can share the whole file*

When other peers has fully downloaded the file, they too will become seeders; where it is common practice to leave the torrent client running for others to become seeders.

2.3. Leechers

Leechers are the peers which do not have the whole file yet, and is still downloading it.

- ***Leecher***
- *Downloads a torrent*
- *Uploads the bits it got*

But even though the leecher is downloading, doesn't mean that it should not upload; why it just upload those bits which it already has.

2.4. Swarm

The swarm is pretty simple; it is the union of the seeders and the leechers for a given torrent.

$$Swarm = leechers \cup seeders$$

2.5. .torrent

A .torrent file describes a given torrent. For a file to become a torrent, it gets broken down, often into thousand of parts, and then it is the job of the .torrent file to allow us where to find each one.

- ***.torrent***
- *describes a torrent*
- *trackers, "file" locations*

Information about trackers, and other kinds of meta data is also part of the file.

The file is not part of the BitTorrent system it self, and is shared traditionally, like simply downloading a file from the web of distributed via email.

More technically, the .torrent file contains a *bencoded* python dictionary, containing at minimum the keys: 'announce' and 'info'.

- ***Keys***
- `announce` *is the url of the tracker*
- `info` *is another dict with the following keys*
- `- name` *of the file*
- `- piece length` *the length of the pieces which the file is broken into*
- `- pieces` *pieces is a string containing the SHA1 hashs of all the pieces*
- `- length` *and at least the total length of the shared file*

3. The BitTorrent protocol

Let's now dive deeper down into the protocol of BitTorrent.

When retrieving a list of roughly 50 peers from the tracker, the new peer will connect to about 30 of them, called its neighbours, over TCP.

3.1. Available pieces

This is where the new peer will send a *bitfield* message to its new neighbours.

Bitfield: 0b1101111111

The bitfield is a space efficient representation of the pieces of the file which the peer got of the file. Though if the new peer does not have any part of the file, it will not be send.

Being that the bitfields are zero-indexed, this means that here, the third piece of the file is missing.

Once the piece has been obtained and its SHA1 is guaranteed, it sends a *have* message to its neighbours, letting them know its available at this peer.

3.2. Downloading

You cant just go ahead and download from the other peers, they will need to grant you permission of each piece first. This is a multi-part process.

Now that the peer is ready to download, it starts out by indicating to a peer that it is interested in download a particular piece.

choke/unchoke = no/yes to download

Now its up to the peer if we are allowed to download or not, referred to as being unchoked or choked. This is a way for the peers to make the game fair. If a peer does not contribute, we can choose to choke those peers, punishing them for not sharing and thereby trying to make them share. Choking is reconsidered every ten second ish.

But what if everybody is choked? That could happen. Every 30 seconds or so, one or more peers will get *optimisticly unchoked*, why it will download and thereby should unchoke those which is download from.

3.3. Piece strategy

Great, so quickly this last thing about the protocol: which pieces to downloading.

You could do a *random first* or a *rarest first*, kind of strategy, and I think those are pretty self explanatory.

Though, when joining and having nothing to share, one relies on *optimistic unchoke* to be able to download and one should just be happy to download something, why *random first* should be used, and then later shift over to *rarest first* when you have something others are interested in.

4. Attacking BitTorrent

Great so having discussed BitTorrent, lets now discuss how to attack it. Yes. This is war.

4.1. Harming the swarm

The kind of attacks we will be looking at, means to harm the swarm; where we wish to make it difficult for other peers to download a specific file.

This could be pretty simple, as we could have a lot of peers claim to have a piece, and when people try to download it, we simply send them a bad piece.

Claim to have piece, share bad.

Well that works! But it takes a lot.

Lets look more generally on some attacks on a swarm, starting with a Sybil attack of *piece lying*.

- ***Piece lying***
- `join with peers` *as it is a Sybil attack, we start by joining with a bunch of peers*
- `rare first strategy` *now .. we wanna use the rare first strategy.. mmm.. I wonder what a collection of malicuis peers could do with this strategy?*
- `claim rare pieces` *now we wanna lie about having the rare pieces, making them appear not rare. In this way, when the peers having the actual rare pieces leave, the pieces will become extinct, which means the swarm has failed.*
- `any interested?` *it could be that some peers use the random strategy, and in that case and they ask for a piece we lie about, we simply choke them*

We could also make an Eclipse attack, where we aim to isolate a peer from other peers.

- ***Eclipse attack***
- `Join with peers` *again we need to join with a lot of peers*
- `connect to real peer` *then we seek to find a real peer to connect with*
- `isolate!` *and if so, we tell the other malicious peers about this peer, and we all try to connect to him! Doing so, we are able to isolate the peer, and then we can decide what happens next.*