TorrentTrust: Trust-based BitTorrent Scoring

Ian Sibner
Quanze Chen
Evelyn Yeung
David Xu
{isibner,cquanze,eyeung,davix}@seas.upenn.edu

Motivation
The BitTorrent network provides no guarantees regarding file quality.
- Files are only identified by the hash of their contents
- Hackers can upload viruses/malware to scam money from downloaders

Challenge
Determine whether a file is legitimate.
- Current solutions:
  - Private tracker voting (e.g. Pirate Bay rating)
  - Voting weighted by users' vote correlation (e.g. Credence)
- Not decentralized
- Vulnerable to vote manipulation

Goal
Build a system to help detect malicious torrents.
- Desired properties:
  - Fully decentralized
  - Difficult/expensive for hackers to manipulate votes
  - Accurate
  - Able to classify most content

Approach
Trust-based vote weighting
- Weight a vote based on relationship with user who cast it
  - Concretely: add web of trust term to Credence weighting
  - Difficult for scammers to infiltrate social networks
- Store data (signed with public key) in DHT
  - Completely decentralized

Implementation
- Java for maximum portability
- Users can look up torrent scores and manage trust network from the browser

Simulation
Setup
- 10 large scammer cliques upvoting a malicious virus
- Randomly connected to “good” users in a simulated social network graph
- Measured rate of correct virus classification, as well as overall coverage (percent of content with a score)
  - Tried several trust metrics: BFS-depth, Eigentrust

Findings
- BFS outperformed Eigentrust
- Tradeoff between coverage and virus identification
- Attackers did not gain from creating additional accounts

Future Work
- Cache the trust network graph to improve system performance
- Create mobile app (iOS and Android) let users add friends to their trust graph in person
- Experiment with different trust metrics

Questions? Please Ask!