Comment Number:	OL-100056
Received:	12/29/2004 7:15:37 PM
Submitted As:	CW Web Form
Organization:	RazorPop
Commenter:	Marc Freedman
Agency:	Federal Trade Commission
Rule:	Notice Announcing Public Workshop and Requesting Public Comment and Participation
Docket ID:	Not yet available
Attachment:	filtering-ftc.htm
Attachment:	filtering-ftc.htm

Comments:

A Dirty Word: Filtering

INTRO

Filtering has become a dirty word in the entertainment vs. P2P war. The entertainment industry trots it out regularly as the solution for stopping copyright infringement. The RIAA says that it's just like filtering adult content or IP addresses. The P2P players reject it outright as impractical and a return to the illegal past. This paper dives past the rhetoric with an analysis of whether and how it could work.

DETAIL

PREFACE

This paper is not a comprehensive analysis of filtering. It does not address:

- Social, moral, consumer, and legal aspects of filtering
- The economics and costs of such filtering
- Whether such filtering should or must be mandatory or voluntary.
- How a song is filtered (fingerprinting, file name match, etc.) and acceptable levels of false positives.

The objective herein is only to look at the practicality of implementing filtering similar to keyword or IP blocklists in current P2P software. It does not look at other types of filtering architectures, such as third party software that would be called like anti-virus or anti-spyware software. Such software would have different performance, usage, and distribution considerations.

Today P2P is a ubiquitous technology. There are hundreds of P2P software developers categorized below. My company, RazorPop, is the developer of TrustyFiles, which will be used to guide this paper.

- Commercial consumer developers that support US copyright law like RazorPop (TrustyFiles), Sharman (Kazaa), Streamcast (Morpheus), and Metamachine (Edonkey).
- International developers not bound to US law.
- Rogue developers like Earthstation 5 that do not abide by any copyright law.
- Open source programmers such as Bram Cohen (Bit Torrent)
- Group P2P developers that provide networks for group rather than public use.
- Secure network developers, such as darknets, that strongly protect file names and content.
- Amateur developers such as students
- Dr. Edward Felten, who wrote TinyP2P in just 15 lines of code.
- Microsoft (1, 2), Intel, and other software operating system and tool providers that provide P2P developer kits.
- Business and enterprise P2P network developers
- Grid computing developers

THE PAST

First let's review the only real world attempt at P2P filtering - Napster 1.0 circa 2001. The architecture was centralized. Initially 1.6 million files were filtered by song title and artist, and various word and letter combinations. This was not quite close to the 8 million files RIAA requested filtered. Users got around the filters by altering the artist and title names and through software like NapCameback title scrambling. Napster spent over $2 million to add fingerprinting for 2 million songs via Relatable and Loudeye/Gracenote. The only acceptable standard was total 100% compliance, which was never reached. And so Napster was shut down and never re-opened.

TODAY

TECHNOLOGY. The Napster architecture was integrated with search performed centrally at the Napster server and downloading decentralized directly between two users. Today's architecture is fully decentralized and disaggregated. Today's software does not access developer-maintained resources for file registration, sharing, search, or downloading. There is no central database of files being shared like Napster. Developers have no knowledge of user activity. The exception is server-based network nodes, used by developers and third parties to offer legal content, for which filtering is not an issue.

Current consumer networks are typically open and accessible by over a hundred proprietary and open source software clients. The networks are standards-based public utilities. The P2P technology provider is not the combined network operator/software developer as in the case of Napster, but the software developer only whose client accesses one or more networks.

This paper assumes content fingerprinting is used as the most effective means of identifying files that infringe copyrights.

HASH CODES. P2P networks exchange a small amount of information when users run searches, attempt downloads, and share files. This data includes file name and hash code. Files names can be the same for different files. So the hash code is used as a unique file identification. It ensures that the file requested is the same one received, whether from one or hundreds of users. Most P2P networks use different hashing algorithms.

Hash codes have 3 critical characteristics for filtering that make them ideal for disseminating infringing file identifications. They are unique, they are small (100 characters or less regardless of the size of the file, which could be gigabytes), and they are text-based, such as JUKPCGZHIWLK3GFOP4S5OQT7TV5ZWZXF or 28816dc8ea93f91fe45dd61378a25d508a0677e0c44aba5ec11dc3170154b706ceacc70e.

Filtering by hash codes is similar to adult file filtering. For adult files incoming file file names are checked against an adult keyword list. For copyright infringement incoming file hash codes are compared against the hash code list. In this way filtering by hash codes has the potential for easily being accommodated in today's software.

HOW TO GET THE HASH CODE LIST.

1. An agent of the entertainment industry scans P2P networks for suspected files by artist name, song title, copyright, and other metadata
2. Suspected files are downloaded.
3. Content fingerprinting software is used to identify actual copyright infringement against these suspected files.
4. Hash codes are generated for identified violating files and compiled into a hash data file.
5. The file is uploaded by the P2P software, just like the P2P software uploads IP blocklist or adult keyword list updates.

IT'S NOT THE TECHNOLOGY. The problem with filtering is not technology, which is well known and was available at the time of Napster. The issues are file size, agency, and effectiveness.

FILE SIZE. Let us first estimate the number of hash codes required. This is the number of copyrighted songs X number of versions per song X number of hash codes per file.

Let us assume the following

- Number of copyrighted songs = 1 million.
- Number of versions per song = 10. This is conservative. Search for a hit song today on P2P and you don't just see one file but dozens, based on various song versions and how the song was ripped (converted to digital format).

- Number of hash codes per file = 5. Again, this is conservative. TrustyFiles accesses multiple networks and maintains 8 hash codes for each file for different networks.

The total number of hash codes is 1 million X 10 X 5 = 50 million.

A hash filter list is similar in format and size to an IP blocklist. TrustyFiles has a 28,000 rule IP blocklist that is 477KB in size when compressed. A 50 million rule hash filter list translates to 850MB. Even if we consider only 1 hash code per file, the size is still 170MB. In comparison P2P software is 1 to 10 MB in size. The extremely large file size of the hash filter list makes it impractical to bundle or download, and so is incompatible with today's P2P software.

AGENCY. There are numerous software developers, which calls for a standards-based approach with an organization(s) authorized by rights holders to develop, maintain, and distribute a file blocklist and/or software to process it. Such an organization, standards, and software do not exist or are not available to P2P developers.

EFFECTIVENESS. Napster showed that filtering cannot be totally effective. Fingerprinting can be avoided by encryption and other means. The legal standard of 100% effectiveness established by the Napster precedent will never be able to be achieved and so is not acceptable to the P2P industry. There must be some definition of filtering effectiveness and coverage to ensure an acceptable level of filtering for all parties. Such a definition does not currently exist.

CONCLUSION

While on the surface it may appear that filtering files for copyright infringement is just like filtering for adult content or IP addresses, this analysis shows that it is not practical. Any one of the above issues is sufficient for filtering to fail. There is no evidence or guarantee that filtering is compatible with consumer P2P software or that it would be successful to a legal standard.

MOVING FORWARD

The filtering challenge is sufficiently complex that a joint effort of the entertainment and P2P industries would be required to set practical expectations and appropriate standards.

Consumer P2P software is legal in the United States. Thus filtering is at best an option for developers. Incentives are required for developers to develop such filtering and for consumers to take it. Such strategies include the following.

1. IMDEMNIFICATION. Filtering is not an ends, but a means. It is a process that will be continually improved as the technology advances and new content is added. Rights holders should indemnify P2P developers and users who make good faith attempts to comply.

2. LICENSING. The entertainment industry should provide licensed content at competitive prices to replace those files that are being blocked to retain consumer interest.

Marc Freedman
RazorPop, developer of TrustyFiles, the leading multiple network P2P file sharing software
Read more articles at the P2P Insider's Blog.