Software and Methods Supporting Legal Proceedings: Home-Brewed and Speculative or Verifiably Reliable and Definitive?

Kalman C. Toth Ph.D. P.Eng.

kalmanctoth@gmail.com

Abstract

Software-based tools are often used to collect evidence for court proceedings. Some are well-tested and commercially available. Others are not. Over the last fifteen years, tens of thousands of lawsuits claiming copyright infringement using BitTorrent networks have been brought by media companies. The software-based tools used to report copyright infringement are not commercially available, not qualified by third parties, yet promoted as “forensic tools”. Explained is how wrongfully accused users are harmed; how Daubert [6] addresses admissibility of evidence; how BitTorrent seeders share infringing movies in pieces with peers; how such BitTorrent monitoring software collects pieces of movies from IP addresses; and why IP addresses do not distinguish infringers from non-infringers. The tools collect only a few pieces of movies from targeted IP addresses, while no pieces are collected from seeders holding infringing copies. The methods speculate about infringement rather than definitively collecting complete playable copies for validation. The software does not overcome critical failure modes including abandonment, piece-unavailability, space-depletion, and choking. Explained is how repeatability and reproducibility testing, completeness, playability, transaction validation, and audio-visual matching yield verifiably reliable software and infringement reporting for acceptance in software engineering and forensics communities. Using unreliable software for forensics purposes is malpractice.

Biography

***Kal Toth** provides technical expertise to law offices defending individuals claimed to have used BitTorrent to infringe copyrighted movies. He has published several papers and US patents in the field of digital identity. Kal has worked for Hughes Aircraft, Datalink Systems Corp., the CGI Group Inc., the Software Productivity Centre (BC), Intellitech Canada (Ottawa), and the Canadian federal departments of Defence, Communications, Transportation, Revenue, External Affairs, and the Communications Security Establishment. He is former Executive Director of the Oregon Master of Software Engineering (OMSE) program, obtained his Ph.D. in computer systems and electrical engineering from Carleton University, and is a registered professional engineer with a software engineering designation in British Columbia.*

1. Introduction

BitTorrent [1] is used to legally share software, movies, games, and other content over the Internet - BitTorrent is also used to illegally share copyrighted movies. Defective methods and software monitoring BitTorrent for copyright infringement inflicts much harm on wrongfully accused users exposing them to significant litigation cost, stress, lost privacy, and damaged relationships. Prenda Law[1] used supposedly reliable BitTorrent monitoring software to claim thousands of users had infringed. Likely many users were harmed. Evidence of self-seeding collapsed the case and bankrupted the firm.

NARS [14], MaverickMonitor [15][16] and VXN Scan [2] [17] are tools routinely used to monitor BitTorrent usage and assert IP addresses were used to infringe copyrighted movies. Owners refer to their tools and methods as *infringement detection systems* and/or *forensics tools* and claim the implementing software is free of defects (reliable) and reports infringement accurately. Owners provide no evidence supporting such claims and normally decline to disclose the source code. In one case, however, the source code was provided and author comments suggested numerous bugs remain latent in the software.

This paper discusses challenges BitTorrent monitoring tools do not overcome, and explains how software engineering and forensics standards can be exploited to overcome most (but not all) of them.

2. Copyright Infringement in BitTorrent Context

According to the US Copyright Office, copyright infringement occurs when a copyrighted work is reproduced (i.e. copied), distributed, performed, publicly displayed, or made into a derivative work without the permission of the copyright owner [2]. With respect to movies (e.g. MP4s, MOV, AVI, WebM), this implies that possession and playability by alleged infringers should also be demonstrated. After all, holding a file of unplayable zeros and ones does not infringe copyright. In the BitTorrent context, digital media is infringed when a person:

- holds a complete or substantial playable portion of a copyrighted movie obtained from BitTorrent, or
- distributes a complete or substantial playable portion of a copyrighted movie into BitTorrent.

3. Admissibility of Technical Evidence - Daubert

In the United States, the admissibility of computer evidence is typically introduced at a very late stage of the proceedings where both sides have incurred tremendous economic and emotional capital. Daubert [6, 10,12] addresses the admissibility of technical evidence in court proceedings. Daubert principles rely on the sufficiency of facts and data, reviews by peers, acceptance by the relevant scientific community, and qualifications of relied-upon experts. Daubert also implies software-based tools used to collect forensics evidence for court proceedings should undergo empirical testing and be presented and/or published at/in conferences, workshops, and journals/periodicals for review and acceptance by the relevant scientific community, in this case, the software engineering and forensics communities.

4. BitTorrent Clients Installed on PCs Share Movies in Pieces

To share content over BitTorrent (BT), users install software applications called BitTorrent clients on their personal devices. These clients implement the BitTorrent protocol specification [1] enabling peer-to-peer (P2P) sharing by way of IP addresses. BitComet, BitTorrent, Deluge, qTorrent, rTorrent, uTorrent, and Vuze are popular BitTorrent clients. Currently, uTorrent[3] has the largest market share.

[1] https://en.wikipedia.org/wiki//Prenda_Law

[2] Excipio operates NARS (Network Activity & Recording Supervision); MaverickEye owns MaverickMonitor; Strike 3 owns VXN Scan

[3] uTorrent has about 69% of the market for BitTorrent clients

BitTorrent clients enable their owners to share movies and other media in small “pieces” using their IP addresses. A BitTorrent swarm is composed of IP addresses sharing pieces of the same movie. A BT client holding all pieces of a movie (a complete playable copy) is called a “seeder” or “seed” while a BT client interested in obtaining pieces of the movie is called a “peer”. Swarms sharing popular movies can attract 100s, 1000s or more peers. “Have”, “interested”, and other such protocol messages are used by seeders and peers to share pieces. Peers can obtain pieces from seeders (seeds) or from other peers. Peers refusing to contribute pieces to a swarm, called “leechers”, can be “choked” from obtaining pieces thereby throttling the rate at which pieces are obtained. The hash of each piece is verified when received. Once a peer has obtained and verified all pieces of a movie, the BitTorrent client assembles the pieces into an exact playable copy. Peers obtaining complete playable copies can choose to become seeders actively uploading pieces to peers in the swarm, or they can leave the swarm (become inactive).

Figure 1 depicts ❶ a 47-minute 391 MB playable movie (#m) fragmented into 746 x 524,488B (1/2 GB) pieces by the initial seeder’s BT client. Also depicted is a user with a BT client ❺ sharing a router and IP address wishing to obtain a copy of movie #m. The seeder’s BT client creates a .torrent file (.torrent m#) specifying certain characterizing attributes (metadata) of movie #m; a pointer to the file holding the movie; and a SHA1 hash (160-bit/20-byte) of all 746 pieces (piece-hashes) of movie #m. The .torrent file itself is uniquely identified by a SHA1 hash, infohash #m, calculated from the .torrent metadata and the 746 piece-hashes. Infohash #m identifies the BitTorrent swarm sharing the pieces comprising movie #m.

The initial seeder uses her client to post ❷ .torrent #m to Torrent Site(s). When ready to share pieces of movie #m, the initial seeder sends ❸ infohash #m and her IP address to a BitTorrent Tracker which uses infohash #m to track IP addresses participating in swarm #m. Peers interested in movie #m search ❹ Torrent Site(s) to locate and obtain a copy of .torrent #m. To obtain pieces from swarm #m, peers use .torrent #m to calculate infohash #m, send infohash #m and their IP addresses to the BitTorrent Tracker. The BitTorrent tracker returns IP addresses of seeders and peers sharing pieces in the swarm.

User ❺ progressively receives requested pieces ❻ from seeders and peers over the movie timeline. The user’s BT client hashes received pieces verifying they match the .torrent #m file’s piece-hashes. Once all pieces are received and verified, they are assembled into a playable copy of the shared movie. Peers with complete playable movies can opt to seed ❼ pieces they hold back into the swarm (p^1⇨s^1 p^2⇨s^2).

5. Monitoring Software Collects Pieces from IP Addresses

A software-based BitTorrent monitoring tools exploit Torrent web sites, .torrent files, BitTorrent trackers and the BitTorrent Protocol [1] to locate swarms sharing pieces of copyrighted movies of interest. Infringement reports produced by such tools are similarly formatted. BitTorrent monitoring tools mimic peers downloading pieces from IP addresses but do not upload pieces to requesting peers (called leeching). Monitoring tools locate the .torrent file of a targeted movie and use the infohash of the .torrent file to obtain IP addresses participating in the swarm from a tracker. Monitoring tools record BitTorrent transactions in protocol capture files (PCAPs) including date/time, monitor’s IP address, client’s IP address, client ID, .torrent file infohash identifying the swarm, and the pieces transferred (in blocks).

Figure 1 depicts a BitTorrent monitoring tool comprised of a proprietary BitTorrent Client and a repository for storing PCAP files. Figure 1 illustrates the BitTorrent monitor ❽ obtaining the .torrent #m for movie #m from a Torrent Site; ❾ calculating infohash #m from .torrent #m; and using infohash #m and the monitor’s IP address to retrieve IP addresses of users participating in swarm #m from a Tracker; and ❿ collecting pieces of movie #m from selected IP addresses in swarm #m.

There is no evidence owners have demonstrated or otherwise established their BitTorrent monitoring tools conform with, meet, or satisfy guidance, recommendations and/or standards of the software engineering community (e.g. SEI/IEEE [3] [4] [5]) or the forensics community (e.g. NIST [6] [7] [8]).

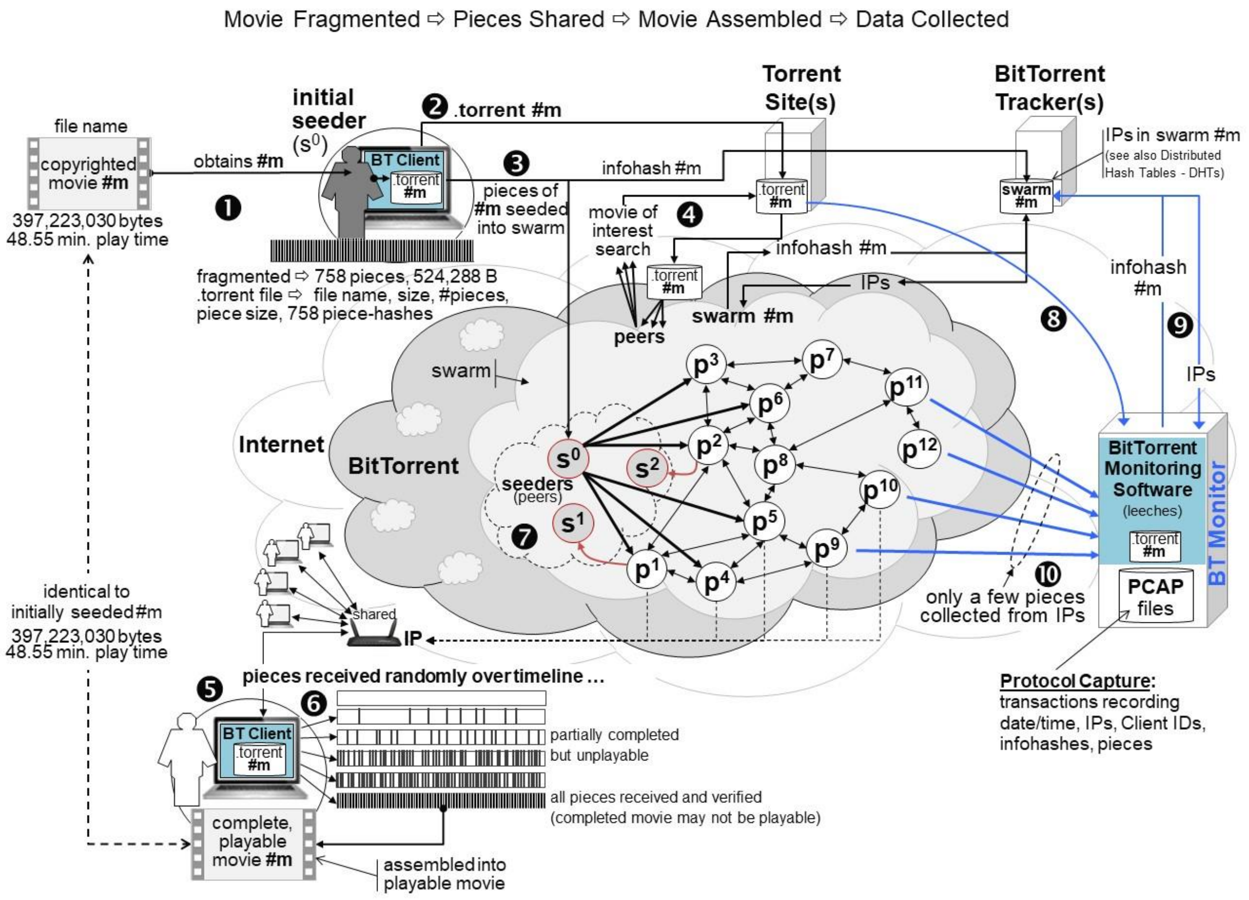

Movie Fragmented ⇨ Pieces Shared ⇨ Movie Assembled ⇨ Data Collected
file name
copyrighted movie #m
397,223,030 bytes
48.55 min. play time
initial seeder (s⁰)
obtains #m
❶
BT Client
.torrent #m
❷ .torrent #m
❸
pieces of #m seeded into swarm
fragmented ⇨ 758 pieces, 524,288 B
.torrent file ⇨ file name, size, #pieces, piece size, 758 piece-hashes
Torrent Site(s)
BitTorrent Tracker(s)
infohash #m
IPs in swarm #m
(see also Distributed Hash Tables - DHTs)
swarm #m
movie of interest search
❹
infohash #m
IPs
peers
swarm #m
swarm
Internet
BitTorrent
seeders (peers)
s0
s1
s2
p1
p2
p3
p4
p5
p6
p7
p8
p9
p10
p11
p12
❼
❽
❾
infohash #m
IPs
BitTorrent Monitoring Software (leeches)
BT Monitor
PCAP files
❿
only a few pieces collected from IPs
Protocol Capture: transactions recording date/time, IPs, Client IDs, infohashes, pieces
identical to initially seeded #m
397,223,030 bytes
48.55 min. play time
shared
IP
pieces received randomly over timeline ...
❺
❻
partially completed but unplayable
all pieces received and verified (completed movie may not be playable)
complete, playable movie #m
assembled into playable movie

Figure 1. BitTorrent Sharing, Monitoring, and Protocol Capture

6. Swarm and IP Address Targeting Not Validated

Figure 1 ❽ does not explain how BitTorrent monitoring tools target .torrent files and swarms suspected of being used to actively share copyrighted movies.

Apparently, .torrent files thought to be used to share pieces of a *copyrighted movie* are searched and then audio-visually matched. Since movie titles, actor names, and attributes may be misspelled or modified, search terms may need to be tweaked until a prospective swarm is identified. Pieces are collected from multiple IP addresses participating in a swarm until a playable *full copy* is obtained which is audio-visually matched to a *copyrighted movie.* If successfully matched, the swarm is asserted to be sharing infringing pieces of the *copyrighted movie*. Subsequently, the swarm and IP addresses participating in the swarm are targeted by the BitTorrent monitoring software.

However, PCAPs validating the swarms from which such full copies are obtained are not recorded. There is no evidence that targeted and monitored swarm share pieces of copyrighted movies as alleged.

7. IP Addresses Do Not Identify Infringers

Shared routers are universally used to access the Internet over public (“routable”) IP addresses allocated by Internet Service Providers (ISPs) to the router. There is no evidence BitTorrent monitoring tools can trace transactions to and from private IP addresses of shared routers to: (a) unambiguously identify persons using private IP addresses of a router; or (b) determine which private IP addresses are being used by users with a BitTorrent client; or (c) determine whether only a single private IP address of the router is being used; or (d) if the subscriber is the only person using the IP address at a given date/time.

Consider a household of 2.5 family members[4] plus a guest and a neighbor sharing a router and IP address. Assume that of these 4.5 persons, one of them used the IP to infringe a copyrighted movie. It is only 22% likely a given user sharing the router infringed, while it is 78% likely an arbitrarily selected user, including the subscriber, did not infringe. Yet courts have routinely issued subpoenas to ISPs to disclose the identity of the subscriber of an IP address reportedly used to infringe a copyrighted movie.

Figure 2 ❸ illustrates a critical limitation: *Shared IP addresses do not identity the person or persons (including the subscriber) who may have used a shared IP address to infringe copyrighted movies over BitTorrent. Cobbler [5] [11] concluded that IP addresses alone do not identify infringers [12].*

8. Speculative Methodology – Failure Modes Ignored

Tool owners routinely speculate that collecting a few pieces from an IP address in a swarm is enough to assert that an IP address was used to infringe a copyrighted movie. The BitTorrent monitoring methods and tools referenced in this paper typically collect a tiny fraction (2 or 3 pieces) of a movie from a targeted IP address which represents 0.3% or 9 seconds of play time[6] of an average-sized movie. As illustrated in Figure 2 ❺ and ❻, such miniscule evidence of infringement is not predictive of user actions and critical failure modes preventing the downloading and assembly of complete playable movies. Figure 2 ❶ ❷ ❽ illustrates that BitTorrent monitoring tools discussed do not report the IP addresses of seeders who, by definition, hold all pieces of movies shared into a targeted BitTorrent swarm.

Critical Failure Modes: Critical failure modes include: BitTorrent users abandoning downloads before completion; unavailability of pieces in a swarm preventing completion; memory or disk space depletion causing the software of a BitTorrent client to fail; and BitTorrent protocol “choking” preventing BitTorrent monitoring software from collecting pieces held by a targeted IP address.

Abandoned Downloads: Users may abandon downloading pieces from a swarm before completion because of mistaken intent, lost interest, or frustration due to piece-unavailability. Since monitoring tools collect only a few pieces of movies from targeted IP addresses they do not detect abandonment.

Piece-Unavailability: Download completion and playability relies on seeders and peers making enough pieces available in a swarm and monitoring the swarm long enough to collect all the pieces held by a targeted peer and IP address. Piece-unavailability prevents completion and hence playability.

Space Depletion: Certain BitTorrent clients pre-allocate space for pieces to prevent failure due to space depletion; to eliminate the time required to render playable movies; and to increase download speed. Most BitTorrent clients do not support space pre-allocation thereby increasing the likelihood of software failure before completion especially when downloading high-resolution/feature-length movies.

Choking: Since BitTorrent monitoring tools emulate leeching peers they may be “choked” by peers for refusing to share pieces when requested. Choking can prevent BitTorrent monitoring software from collecting all pieces of movies from targeted IP addresses likely rendering them unplayable.

To prove such failure modes have not corrupted infringement reporting, BitTorrent monitoring tools must:

- Collect all pieces from IP addresses to prove downloads were not abandoned before completion.
- Increase the duration of swarm monitoring to achieve sufficient piece-availability, obtain all pieces of the movie from the swarm, and thereby avoid false infringement reporting.

[4] 2020 US census estimated that the average household has about 2.5 members

[5] *Cobbler Nevada, LLC v. Gonzales*, 901 F.3d 1142 (9th Cir. 2018)

[6] Based on movie size of 397 MB, piece size of 524,288 B and playtime of 48.55 seconds

- Collect pieces only from BitTorrent clients pre-allocating space (e.g. rTorrent and uTorrent) to reduce BitTorrent client space depletion risk and increase likelihood complete playable copies are obtained.
- Collect all pieces of movie from swarm to prove choked monitoring has not prevented completion.

BitTorrent monitoring software must obtain a complete copy (all pieces) of a copyrighted movie to objectively prove that critical failures (abandonment, piece-unavailability, space depletion, and choking) did not occur when monitoring a given IP address in a targeted swarm.

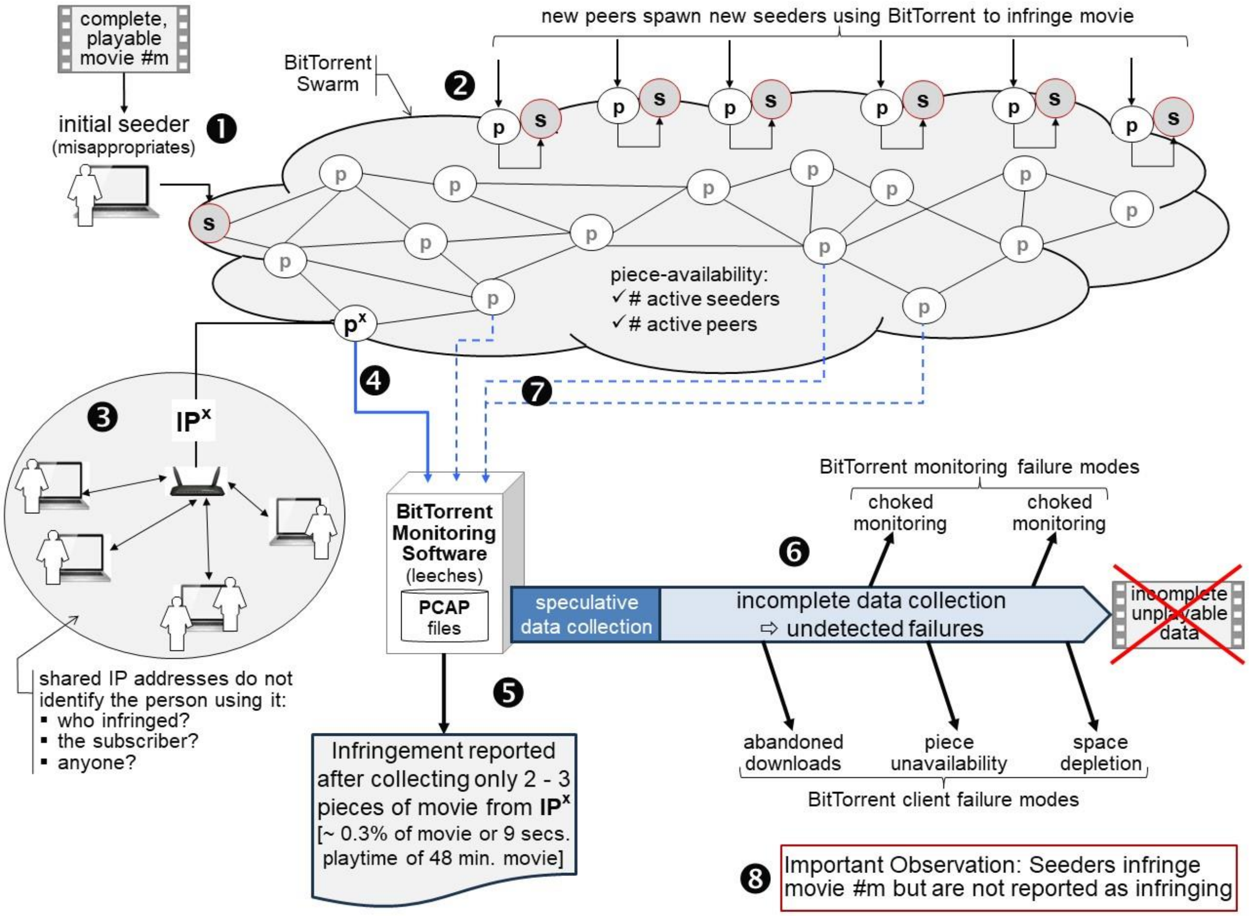

complete, playable movie #m
initial seeder (misappropriates)
❶
BitTorrent Swarm
❷
new peers spawn new seeders using BitTorrent to infringe movie
s
p
piece-availability:
✓# active seeders
✓# active peers
p^x
❸
IP^x
shared IP addresses do not identify the person using it:
▪ who infringed?
▪ the subscriber?
▪ anyone?
❹
❼
BitTorrent Monitoring Software (leeches)
PCAP files
speculative data collection
incomplete data collection
⇨ undetected failures
❻
BitTorrent monitoring failure modes
choked monitoring
choked monitoring
incomplete unplayable data
abandoned downloads
piece unavailability
space depletion
BitTorrent client failure modes
❺
Infringement reported after collecting only 2 - 3 pieces of movie from IP^x [~ 0.3% of movie or 9 secs. playtime of 48 min. movie]
❽
Important Observation: Seeders infringe movie #m but are not reported as infringing

Figure 2. Illustrating Speculative Reporting, Failure Modes, and Unreported Seeders

9. No Evidence the Monitoring Software is Verifiably Reliable

Tool owners have claimed, without evidence, their monitoring software is free of defects. However, no technical documents are provided: no operational theory describing the forensics purpose; no requirements, architecture or design specifications; no use cases or UI specs; no user, operator, and maintenance manuals; and no records of peer reviews, code inspections or design walkthroughs. Tests have typically been conducted after infringement monitoring and reporting. If testing is not conducted prior to use, there is no evidence the monitoring software operates correctly when deployed.

There is no evidence that the monitoring software is verifiably reliable before use.

10. BitTorrent Monitoring Software Trivially Tested

Test procedures, test data and test results were undocumented. A few light-weight (trivial) informally described test cases were conducted using personal computers with installed BitTorrent clients and test movies ranging from a few minutes of play time to 45 minutes which are water-marked. Test computers (PCs) with installed BitTorrent clients were connected to the Internet by a fixed IP address. The BT client of each PC was used to create a .torrent file for each test movie. The tester then provided the infohash derived from each .torrent file to the system administrator enabling the BitTorrent monitoring software to download all pieces of each test movie. The test cases establish rudimentary swarms having (only) two participants - the tester seeding pieces into BitTorrent, and the monitoring tool collecting pieces.

The test cases were not conducted using shared routers and IP addresses allocated by an ISP. No tests were conducted to objectively verify that the monitoring software can identify persons using private IP addresses of routers and/or distinguish infringers from non-infringers using a given router. And the tests were not conducted under loaded conditions with 100s, 1000s, or more peers competing for pieces from seeders and other peers – in other words, there is no evidence BitTorrent monitoring software can download all pieces of a movie under such loaded conditions. Finally, no tests were conducted to verify the monitoring software can collect all pieces of high definition, feature-length movies from IP addresses.

Apparently, comprehensive tests providing strong reliability assurances before use were not conducted. And there is no evidence that robust repeatability and reproducibility testing was conducted.

11. PCAPs and Infringement Reporting Completeness

PCAPs should be exploited to validate the completeness of infringement reporting. The essential purpose of recording transactions (PCAPs) is to introduce redundancy to off-set defects that may be latent in the code, report defects and anomalies, and thereby elevate software reliability. Once BitTorrent monitoring software has collected all available pieces from a targeted IP address in a targeted swarm, the recorded PCAP files should be used to verify all collected pieces were obtained from the same IP address, and hashes of all pieces recorded match the piece-hashes specified in the .torrent file identifying the targeted swarm by way of the derived infohash.

However, the monitoring software does not collect all pieces of a movie from IP addresses in a swarm, apparently the goal being is to identify the swarm's infohash and target all IP addresses in the swarm.

12. Matching Infringing Copies to Copyrighted Movies

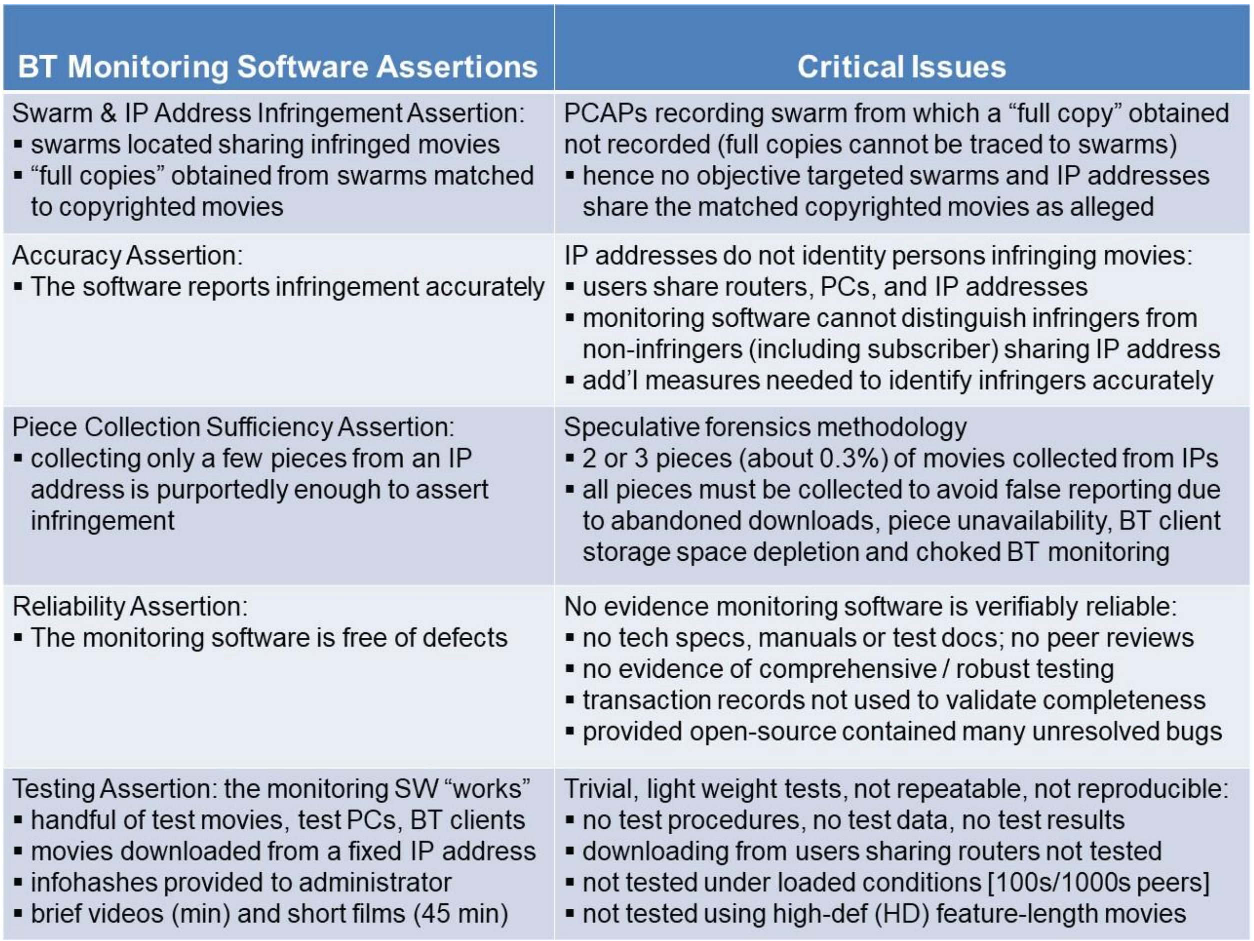
Ideally, upon successfully validating all pieces of a movie (complete copy) have been collected from a targeted IP address, the pieces would be assembled and rendered into a playable movie that would be audio-visually matched to the copyrighted movie apparently shared over the targeted swarm.

Given only a few pieces are collected from an IP address, audio-visual matching is not possible.

13. Inspected Code – Numerous Unresolved Bugs

In one case, source code was provided. Presumably it represented the operational software implementing the BitTorrent monitoring tool. No technical specifications or build/make files were provided. It was therefore not possible to establish if the provided source code compiled into the executable code. Code inspection revealed the software consisted of a collection of open-source software modules. No documents were provided explaining how the modules were compiled into the runtime, or how they were integration-tested. Author comments implied the software contained many unresolved bugs.

No evidence the software was reliable, or the runtime code was compiled from the open-source software.



BT Monitoring Software Assertions	Critical Issues
Swarm & IP Address Infringement Assertion: ▪ swarms located sharing infringed movies ▪ "full copies" obtained from swarms matched to copyrighted movies	PCAPs recording swarm from which a "full copy" obtained not recorded (full copies cannot be traced to swarms) ▪ hence no objective targeted swarms and IP addresses share the matched copyrighted movies as alleged
Accuracy Assertion: ▪ The software reports infringement accurately	IP addresses do not identity persons infringing movies: ▪ users share routers, PCs, and IP addresses ▪ monitoring software cannot distinguish infringers from non-infringers (including subscriber) sharing IP address ▪ add'l measures needed to identify infringers accurately
Piece Collection Sufficiency Assertion: ▪ collecting only a few pieces from an IP address is purportedly enough to assert infringement	Speculative forensics methodology ▪ 2 or 3 pieces (about 0.3%) of movies collected from IPs ▪ all pieces must be collected to avoid false reporting due to abandoned downloads, piece unavailability, BT client storage space depletion and choked BT monitoring
Reliability Assertion: ▪ The monitoring software is free of defects	No evidence monitoring software is verifiably reliable: ▪ no tech specs, manuals or test docs; no peer reviews ▪ no evidence of comprehensive / robust testing ▪ transaction records not used to validate completeness ▪ provided open-source contained many unresolved bugs
Testing Assertion: the monitoring SW "works" ▪ handful of test movies, test PCs, BT clients ▪ movies downloaded from a fixed IP address ▪ infohashes provided to administrator ▪ brief videos (min) and short films (45 min)	Trivial, light weight tests, not repeatable, not reproducible: ▪ no test procedures, no test data, no test results ▪ downloading from users sharing routers not tested ▪ not tested under loaded conditions [100s/1000s peers] ▪ not tested using high-def (HD) feature-length movies

Table 1: Assertions vs. Critical Issues

14. Analysis: Existing BitTorrent Monitoring Software

Existing BitTorrent monitoring software does not detect abandoned downloads, piece-unavailability, client space depletion, or choking. Such failure modes can be overcome by collecting all pieces of allegedly infringing content from targeted swarms and IP addresses. However, BitTorrent monitoring tools report IP addresses as infringing after collecting only a few pieces – such sparse data cannot be played (viewed). The methodology speculates about infringement rather than obtaining complete playable content that can be viewed and compared to allegedly infringed movies. Given the monitoring software is undocumented and was not tested under typical workloads using shared routers or feature-length movies, the software is not verifiably reliable. Finally, IP addresses alone do not identify infringers or distinguish them from non-infringers nor provide evidence that the subscriber had a BitTorrent client and used it to infringe a movie. Figure 3 presents a top-down analysis of BitTorrent monitoring software. The forensics process is split into three branches: the forensics methodology; software development; and implemented software.

- *IP addresses do not identify infringers or distinguish them from non-infringers or subscribers.*
- *The monitoring software is not verifiably reliable – no technical specs; no peer reviews; trivial testing.*
- *Daubert principles not satisfied – no peer reviews; no tech pubs; and no tech community acceptance.*
- *Speculative methodology – sparse data collection, no verifiable evidence of copyright infringement.*
- *Failure modes not overcome – abandonment, piece-unavailability, space depletion, and choking*
- *PCAP data and audio-visual matching not used to validate infringing IPs, completeness, playability.*

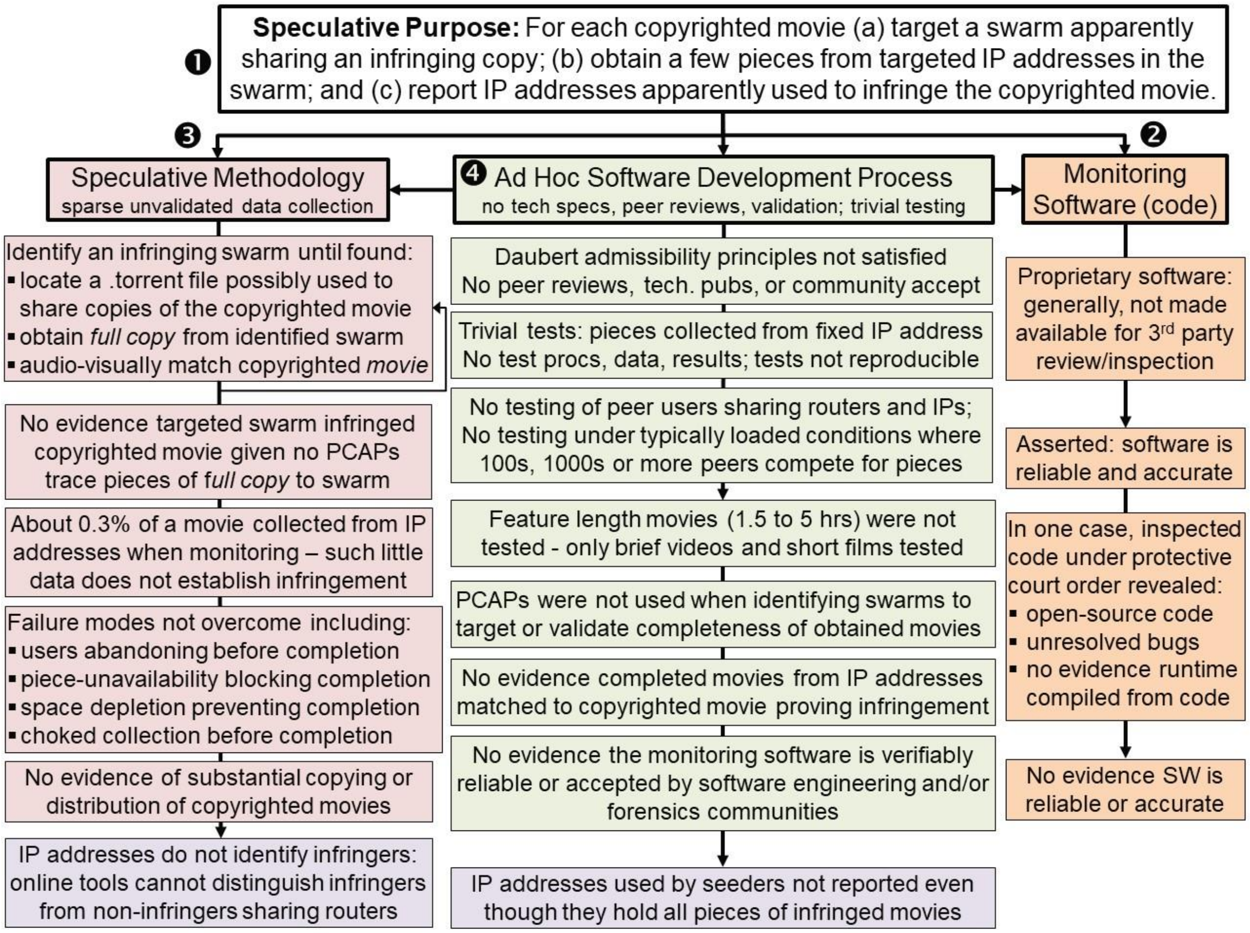

❶
Speculative Purpose: For each copyrighted movie (a) target a swarm apparently sharing an infringing copy; (b) obtain a few pieces from targeted IP addresses in the swarm; and (c) report IP addresses apparently used to infringe the copyrighted movie.
❸
❷
Speculative Methodology
sparse unvalidated data collection
❹ Ad Hoc Software Development Process
no tech specs, peer reviews, validation; trivial testing
Monitoring Software (code)
Identify an infringing swarm until found:
▪ locate a .torrent file possibly used to share copies of the copyrighted movie
▪ obtain full copy from identified swarm
▪ audio-visually match copyrighted movie
No evidence targeted swarm infringed copyrighted movie given no PCAPs trace pieces of full copy to swarm
About 0.3% of a movie collected from IP addresses when monitoring – such little data does not establish infringement
Failure modes not overcome including:
▪ users abandoning before completion
▪ piece-unavailability blocking completion
▪ space depletion preventing completion
▪ choked collection before completion
No evidence of substantial copying or distribution of copyrighted movies
IP addresses do not identify infringers: online tools cannot distinguish infringers from non-infringers sharing routers
Daubert admissibility principles not satisfied
No peer reviews, tech. pubs, or community accept
Trivial tests: pieces collected from fixed IP address
No test procs, data, results; tests not reproducible
No testing of peer users sharing routers and IPs;
No testing under typically loaded conditions where 100s, 1000s or more peers compete for pieces
Feature length movies (1.5 to 5 hrs) were not tested - only brief videos and short films tested
PCAPs were not used when identifying swarms to target or validate completeness of obtained movies
No evidence completed movies from IP addresses matched to copyrighted movie proving infringement
No evidence the monitoring software is verifiably reliable or accepted by software engineering and/or forensics communities
IP addresses used by seeders not reported even though they hold all pieces of infringed movies
Proprietary software: generally, not made available for 3rd party review/inspection
Asserted: software is reliable and accurate
In one case, inspected code under protective court order revealed:
▪ open-source code
▪ unresolved bugs
▪ no evidence runtime compiled from code
No evidence SW is reliable or accurate

Figure 3. Analysis of BitTorrent Monitoring Software

15. Software Engineering Community Acceptance

Software engineering methods, guidelines and standards are prescribed by the Software Engineering Institute (SEI) [3] [4] and Institute of Electrical and Electronic Engineers (IEEE) [5]. There is no evidence referenced BitTorrent monitoring tools have been accepted in the software engineering community.

- Development is not guided by software engineering methods, standards and best practices.
- There is no theory specifying how infringement data is collected, inspected, analyzed, and reported.
- There are no technical specs of the design, critical functions, internal/external, or user interfaces.
- Apparently, technical specifications not reviewed by peers to verify software integrity and reliability.
- Trivial tests simply show monitoring tools obtain all pieces of test movies from a known IP address.
- Testing not conducted under typical workloads where 100s, 1000s, or more users share pieces.
- Tests were not configured with multiple peers using routers and sharing hi-def, feature-length movies.
- Critical failure modes are not tested (abandonment, piece-unavailability, space depletion, choking).
- There is no evidence critical functions were tested (i.e. piece-hashes, infohashes, movie rendering).
- Transaction records (PCAPs) are not used to validate IP addresses, completeness, and playability.
- No evidence infringing pieces collected from allegedly infringing IPs are matched to infringed movies.

Acceptance by the software engineering community is unlikely without objective evidence attesting to the reliability and accuracy of BitTorrent monitoring software (software conferences, workshops and journals).

16. Forensics Community Acceptance

As mentioned, BitTorrent monitoring tool owners routinely assert their tools are reliable and report infringement accurately. There is no evidence these tools have been presented or published at/in forensics conferences, workshops, or journals substantiating such reliability and accuracy assertions.

The National Institute of Standards and Technology (NIST) is a well-respected forensics authority:

- NIST promotes equity in the criminal justice system and strengthening of the scientific basis for forensics, namely, collecting, examining, analyzing, and reporting evidentiary data.
- NIST's Computer Forensics Tool Testing (CFTT) program establishes methods for testing forensics tool reliability including appropriate test procedures, test criteria, test sets, and test hardware [8].
- NIST maintains a searchable "Computer Forensics Tools and Techniques Catalogue". The BitTorrent monitoring tools referenced in this paper do not appear in NIST's catalogue.

NIST states [6] [7] forensics tools producing digital evidence admissible in court must produce repeatable and reproducible results. Repeatability implies the same test results are achieved in the same testing environment, while reproducibility implies the same test results are achieved in different testing environment(s). Test data and test results should be observable by 3rd parties.

In the current context, a .torrent file represents test data for a BitTorrent tool monitoring a swarm. Test results include reviewable transaction records (PCAPs) and playable movies.

- Repeatability tests are successful when all pieces of a movie shared across a swarm have been collected from an IP address, piece-hashes verified, and the pieces assembled into a playable movie.
- Reproducibility tests are successful when all pieces of a movie have been collected from different IPs in the same swarm, hashes verified, and pieces assembled into copies of the same playable movie.
- Repeatability and producibility testing should also be conducted across different swarms using different IPs (e.g. IP^1 & IP^2 obtain movie^A from swarm^A; and IP^3 & IP^4 obtain movie^B from swarm^B).
- The above tests should be conducted at different times under small, medium and large workloads.
- Test movies should range from brief videos to feature-length movies.

There is no evidence existing BitTorrent monitoring software was repeatability and reproducibility tested.

17. Software and Forensics Standards Elevate Reliability

Satisfying published software and forensics standards, guidelines and recommendations substantially increases the reliability of deployed BitTorrent monitoring software while also increasing the likelihood reported IP addresses were used to infringe copyrighted movies.

- Specify the definitive forensic purpose of the software implementing the BitTorrent monitoring tool.
- Collect pieces from targeted IP addresses in a swarm until complete playable copies are obtained.
- Employ a proven software development process to create verifiably reliable software.
- Use shared routers and IP addresses, feature-length movies, and active swarms when testing.
- Conduct repeatability/reproducibility testing across different IP addresses and different swarms.
- Conduct tests over active (live) swarms with 100s to 1000s sharing short to feature-length movies.
- Use transaction data (PCAPs) to validate test results and infringement data when monitoring.
- Have qualified 3rd parties inspect PCAPs to validate IP addresses, completeness and playability.
- Provide PCAPs and infringing copies of movies to defendants to enable them to validate evidence.

Figure 4 identifies critical enhancements to the forensics methodology and BitTorrent monitoring software needed to achieve acceptance in the software engineering and forensics communities.

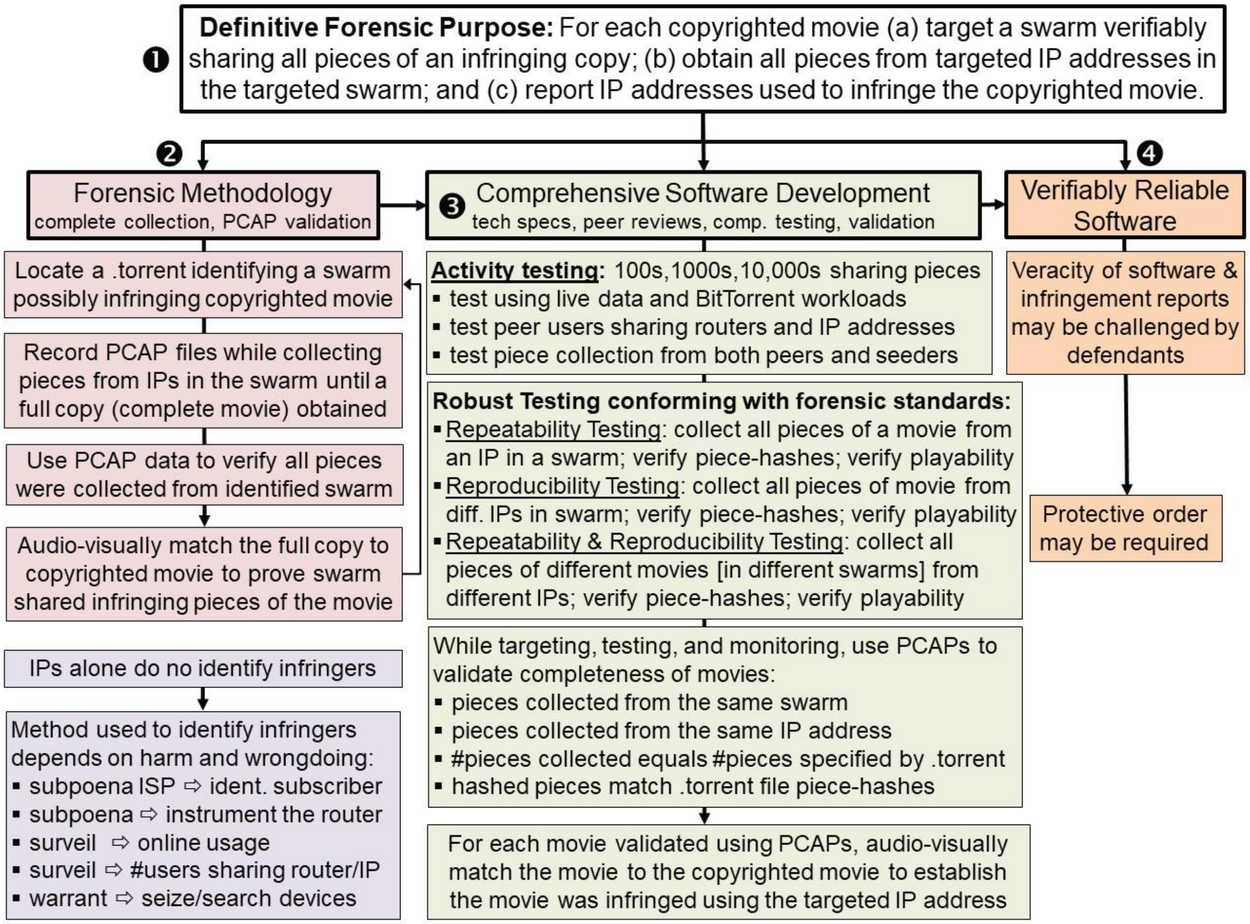

❶
Definitive Forensic Purpose: For each copyrighted movie (a) target a swarm verifiably sharing all pieces of an infringing copy; (b) obtain all pieces from targeted IP addresses in the targeted swarm; and (c) report IP addresses used to infringe the copyrighted movie.
❷
Forensic Methodology
complete collection, PCAP validation
❸
Comprehensive Software Development
tech specs, peer reviews, comp. testing, validation
❹
Verifiably Reliable Software
Locate a .torrent identifying a swarm possibly infringing copyrighted movie
Record PCAP files while collecting pieces from IPs in the swarm until a full copy (complete movie) obtained
Use PCAP data to verify all pieces were collected from identified swarm
Audio-visually match the full copy to copyrighted movie to prove swarm shared infringing pieces of the movie
IPs alone do no identify infringers
Method used to identify infringers depends on harm and wrongdoing:
▪ subpoena ISP ⇨ ident. subscriber
▪ subpoena ⇨ instrument the router
▪ surveil ⇨ online usage
▪ surveil ⇨ #users sharing router/IP
▪ warrant ⇨ seize/search devices
Activity testing: 100s,1000s,10,000s sharing pieces
▪ test using live data and BitTorrent workloads
▪ test peer users sharing routers and IP addresses
▪ test piece collection from both peers and seeders
Robust Testing conforming with forensic standards:
▪ Repeatability Testing: collect all pieces of a movie from an IP in a swarm; verify piece-hashes; verify playability
▪ Reproducibility Testing: collect all pieces of movie from diff. IPs in swarm; verify piece-hashes; verify playability
▪ Repeatability & Reproducibility Testing: collect all pieces of different movies [in different swarms] from different IPs; verify piece-hashes; verify playability
While targeting, testing, and monitoring, use PCAPs to validate completeness of movies:
▪ pieces collected from the same swarm
▪ pieces collected from the same IP address
▪ #pieces collected equals #pieces specified by .torrent
▪ hashed pieces match .torrent file piece-hashes
For each movie validated using PCAPs, audio-visually match the movie to the copyrighted movie to establish the movie was infringed using the targeted IP address
Veracity of software & infringement reports may be challenged by defendants
Protective order may be required

Figure 4. Software Engineering and Forensics Standards Elevate Software Reliability

18. Closing Remarks: Methodology and Software Reliability

Home-Brewed and Speculative: Without technical specs or evidence that proven development and testing processes and methods were used, one can only conclude the monitoring software is ad hoc "home-brewed" and the software is not verifiably reliable. Trivial testing without shared routers under lightly loaded conditions; and capturing transactions (PCAPs) without using them to validate infringement claims, significantly undercut reliability assertions. Collecting only 0.3% of a movie from an IP address participating in a targeted swarm does not represent substantial copying or distribution and speculates, without evidence, that the IP address holds the remaining 99.7%. Collecting such sparse data ignores the implications of abandoned downloads, piece-unavailability, space depletion, and choked piece collection each use case representing non-infringement.

Verifiably Reliable and Definitive: Targeted swarms are those thought to infringe a copyrighted movie. Repeatability and reproducibility testing across active swarms and IP addresses before use ensure the monitoring software is verifiably reliable. To overcome failure modes, monitoring software collects pieces from IP addresses in targeted swarms to completion. PCAP validation verifies that the hashes of all pieces collected from an IP address in a targeted swarm match the piece-hashes of the associated .torrent file. If PCAP validation is successful, audio-visual matching is conducted to verify the pieces collected from the IP address render a playable movie matching the copyrighted movie.

Limitation: *Verifiable evidence that a complete playable copy of a copyrighted movie was obtained from an IP address proves infringement, but it does not identify the person who used the IP address to infringe.*

19. References

[1] *BitTorrent Protocol Specification v1.0*, http://wiki.theory.org/BitTorrentSpecification.

[2] *U.S. Copyright Office - Definitions (FAQ),* www.copyright.gov/help/faq-definitions.html.

[3] *Software Engineering Institute's (SEI) Capability Maturity Model (CMM):* Assists organizations choose and tailor the most appropriate software behaviors, practices, and processes to achieve software reliably and sustainably goals.

[4] *Software Reliability Tutorial*, 2011-2015 by Gullo and Peterson: Pages 25 and 29 tabulate empirically derived software fault rates across the five (5) maturity levels of the CMM determined by leading software reliability experts in the field (Keene, Jones and Krasner).

[5] *IEEE Software Engineering Standards including IEEE Std 12207, Systems and Software Engineering Software Life Cycle Processes*: Documents common frameworks with well-defined terminology for developing software-based systems from requirements stage to system retirement.

[6] *Validation of Forensic Tools and Software, A Quick Guide for the Digital Forensic Examiner* by Josh Brunty: Explains that the *Daubert Standard* and NIST's Computer Forensic Tool Testing Project (CFTT) provide requisite guidance for validating forensics tools and software.

[7] General Testing Methodology for Forensics Tools V1.9, NIST, 11-7-20.

[8] NIST Computer Forensics Tool Testing Project, http://www/cftt.nist.gov; Annex E addresses forensics tools and software.

[9] *BitStalker:* Accurately and Efficiently Monitoring BitTorrent Traffic by Buer et. al.: The authors conduct experiments, investigate and evaluate the reliability of an active monitoring method for reducing false positives.

[10] *Open-Source Digital Forensics Tools, The Legal Argument*: by Brian Carrier: Explains that scientific evidence in US court must be reliable and relevant, tool reliability can be tested applying "Daubert".

[11] *Ninth Circuit Ruling Reminds Copyright Owners that Failing to Allege Enough Facts can cost them*! Jaburg-Wilk, Attorneys at Law, Phoenix, AZ, 10/22/18, discuss the *Cobbler Nevada LLC v. Gonzales, 901 F.3d 1142* case concluded an infringing IP address, standing alone, does not create a reasonable inference that the subscriber was the infringer, more is required than mere allegation.

[12] *Daubert v. Merrell Dow Pharmaceuticals Inc.*, 509 U.S. 579 (1993).

[13] *Challenges and Directions for Monitoring P2P File Sharing Networks*, M. Piatek, T. Kohno and A. Krishnamurthy, University of Washington: authors' experiment using Gnutella and BitTorrent protocols; False positives rates may be elevated by malicious users, malware and buggy software.

[14] Case No.: 3:16-cv-432-BAS-NLS, USDC (CA), *Malibu Media, LLC v. Doe*, Mar. 28, 2016, Order Granting Plaintiff's Ex Parte Motion …, https://casetext.com/case/malibu-media-llc-v-doe-72.

[15] Case No. 3:15-cv-0907-AC, USDC (OR), *Dallas Buyers Club v. Huszar, Motion for Summary Judgement*, Feb. 18, 2018, https://bahnhof.se/filestorage/userfiles/Nyheter%202018/Huszar.pdf.

[16] E. Van der Sar, *Torrent Tracking Evidence is Flawed and Unreliable, Alleged Pirate Argues,* Mar 20, 2017, https://torrentfreak.com/torrent-tracking-evidence-is-flawed-and-unreliable-alleged-pirate-argues-180307/.

[17] E. Van der Sar, *Copyright Troll Now Has its Own Piracy Tracking Tool*, Feb 23, 2020 https://torrentfreak.com/copyright-troll-now-has-its-own-piracy-tracking-tool-200223/.