

THE HONORABLE THOMAS S. ZILLY

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27

U.S. DISTRICT COURT
WESTERN DISTRICT OF WASHINGTON

STRIKE 3 HOLDINGS, LLC, a Delaware corporation,

Plaintiff,

vs.

JOHN DOE, subscriber assigned IP address 73.225.38.130,

Defendant.

NO. 2:17-cv-01731-TSZ

**DECLARATION OF DR. KAL TOTH
IN SUPPORT OF DEFENDANT'S
OPPOSITION TO PLAINTIFF'S
MOTION FOR SUMMARY
JUDGMENT**

JOHN DOE subscriber assigned IP address 73.225.38.130,

Counterclaimant,

vs.

STRIKE 3 HOLDINGS, LLC,

Counterdefendant.

1 I, Dr. Kal Toth, hereby declare under the penalty of perjury under the laws of the
2 United States of America, the following:

3 1. I have been asked to testify as an expert witness in this case for the defendant.
4 My fee as an expert witness is \$ 350.00 per hour.

5 2. My expertise is software verification and validation. I have over 25 years of
6 experience in this field.

7 3. I worked on a number of bittorrent cases and have written expert reports that
8 involve the German forensic investigative company, "IPP". "IPP" goes by other names, such
9 as Guadaley, MaverickEye, and Excipio. This company uses software to purportedly monitor
10 bittorrent data on the internet.

11 4. I reviewed data provided by Strike 3 Holdings, LLC involved in this case. I
12 have also reviewed a declaration of Tobias Fieser at Docket 4-3. I am familiar with Mr. Fieser
13 from other cases, in particular, Malibu Media v John Doe in the Northern District of California.
14 (3:15-cv-04441)

15 5. Exhibit 1 contains my initial reliability assessment of the IPP software.


16 6. I reviewed the first amended complaint at Docket Entry 43 and the list of 87
17 works at Exhibit 43-1. My understanding that the movies referenced at Docket Entry 43-1 are
18 graphic pornographic works.

19 7. I reviewed the data provided in Plaintiff's first set of discovery responses.
20 There were over 80 graphic pornographic works.

21 8. The movies are encoded in an "MP4" format.

22 I declare under penalty of perjury under the laws of the United States that the foregoing
23 is true and correct.

24 EXECUTED this 25 day of February, 2019, at Portland, Oregon.

25 
26 Kal Toth, PhD.
27 Portland, Oregon

CERTIFICATE OF SERVICE

I, Adrienne D. McEntee, hereby certify that on February 25, 2019, I electronically filed the foregoing with the Clerk of the Court using the CM/ECF system which will send notification of such filing to the following:

Bryan J. Case, WSBA #41781
Email: bcase@foxrothschild.com
FOX ROTHSCHILD LLP
1001 Fourth Avenue, Suite 4500
Seattle, Washington 98154
Telephone: (206) 624-3600
Facsimile: (206) 389-1708

Lincoln D. Bandlow, *Admitted Pro Hac Vice*
Email: lbandlow@foxrothschild.com
FOX ROTHSCHILD LLP
10250 Constellation Blvd., Suite 900
Los Angeles California 90067
Telephone: (310) 598-4150
Facsimile: (310) 556-9828

Attorneys for Plaintiff

DATED this 25th day of February, 2019.

TERRELL MARSHALL LAW GROUP PLLC

By: /s/ Adrienne D. McEntee, WSBA #34061
Adrienne D. McEntee, WSBA 34061
Email: amcentee@terrellmarshall.com
936 North 34th Street, Suite 300
Seattle, Washington 98103-8869
Telephone: (206) 816-6603
Facsimile: (206) 319-5450

Attorneys for Defendant

— EXHIBIT 1 —

Exhibit 1

Initial Reliability Assessment of the IPP Software

Feb. 25, 2019

Prepared by Dr. Kal Toth, P.Eng., Portland, OR 97205

For Mr. J. Curtis Edmondson, Law Offices of J. Curtis Edmondson, Hillsboro, OR 97124

I refer to the IPP software as NARS throughout this document and exhibits because the original expert report I authored referred to this system as NARS (see [3a], [3b], [3c] and [3d] below).

The purpose of this document is to report my preliminary reliability assessment of NARS as it relates to Strike 3 Holdings Inc. vs John Doe, Case No. 2:17-cv-01731-TSZ and the motion by Strike 3 Holdings Inc. for Partial Summary Judgment filed 2/17/2019 in U.S. District Court, Western District of Washington for Seattle.

My assessment is based on the evidence I have been provided to date. I have independently arrived at the opinions expressed in this report which depend on the accuracy of this evidence. My opinions are informed by my systems and software engineering qualifications, knowledge and experience.

Attached to this declaration are: Exhibit A, *NARS System Context* and Exhibit B, *NARS Reliability Assessment*.

1. Software Engineering Standards and Guidance on which I Rely

I rely on the following standards and guidance in support of my expressed opinions:

- a. Exhibit C, *BitStalker: Accurately and Efficiently Monitoring BitTorrent Traffic* by Buer et. al.: Describes their investigation and experiments evaluating the reliability of their proposed active monitoring method relative to traditional methods for identifying users sharing content across a BitTorrent network.
- b. Exhibit D, *Validation of Forensic Tools and Software, A Quick Guide for the Digital Forensic Examiner* by Josh Brunty: Relies on the *Daubert Standard* and NIST's Computer Forensic Tool Testing Project (CFTT) providing guidance for validating software-based systems;
- c. *Software Engineering Institute's (SEI) Capability Maturity Model (CMM)*: Refined and widely applied for over two decades assisting organizations choose and tailor the most appropriate software behaviors, practices, and processes in order to achieve software reliably and sustainably goals.
- d. Exhibit E, *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).
- e. *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 the requirements stage to system retirement.

2. Most Relevant Qualifications

The opinions expressed in this report are drawn from my professional experience which is detailed in the Annex to this report, where I highlight my most relevant qualifications for conducting this reliability assessment:

- a. Independent validation and verification (IV&V) for External Affairs Canada
- b. Quality, reliability, maintainability, safety, security, and software engineering for Hughes Aircraft
- c. Software engineering practice leader for CGI Group and Hughes Aircraft
- d. Software engineering courses for 10 universities including Oregon State, Portland State and TechBC.

3. Evidence Reviewed and Referenced

- a. Skype Deposition of Michael Patzer, October 13, 2016.
- b. Declaration of Michael Patzer, September 30, 2016.
- c. Expert Report, Patrick Paige, October 26, 2016.
- d. Supplemental Expert Report, Patrick Paige, December 16, 2016.
- e. Supplemental Report and Opposition to Kal Toth and Bradley Wittman's Expert Report, Michael Patzer, Dec. 30, 2016.
- f. Expert Report of Benjamin Perino, November 23, 2017.
- g. Functional Description, IPP International IPTRACKER v1.2.1 appearing as Exhibit 1 of Declaration of Tobias Fieser in Support of Plaintiffs Motion for Leave to Take Discovery ... filed 08/16/11.
- h. IPTRACKER software provided under Stipulated Protective Order Case No. 3:15-cvv-00907-AC.
- i. Expert Witness of Dr. Simone Richter, April 2, 2014.
- j. Expert Report of Robert D. Young, February 11, 2015.
- k. Deposition of Robert D. Young, January 2, 2018.
- l. Declaration of Stephen M. Bunting, Case 2:17-cv-00988-TSZ, Document 34, filed 2/05/2018.
- m. Declaration of Tobias Fieser in Support of Plaintiff's Motion for Leave to Serve a Third Party Subpoena Prior to a Rule 26(f) Conference, US District Court, Western District of Washington at Seattle, 11/29/2017 [Docket 4-3].

4. My Referenced Expert Reports

I have documented my reviews of some of the above documents in the following reports:

- a. *Expert Report Re. Malibu LLC vs. John Doe*, Kal Toth, Dec 14th, 2016. I pointed out the lack of evidence supporting Patzer's claim in [3a] that NARS is free of defects (is flawless) using the well-known Therac-25 case to illustrate. I also addressed the inadequacy of testing by Paige [3c].
- b. *Expert Report Re. Malibu LLC vs. John Doe, Rebuttal of Patzer Declaration and Paige Expert Report*, Dec. 28, 2017. I rebutted Patzer's declaration [3b] and Paige's supplemental expert report [3d] pointing out the absence of technical specifications, lack of software process, inadequate testing, etc.
- c. *Expert Report Re. Malibu LLC vs. John Doe, response to Patzer Supplemental Expert Report*, Kal Toth, Jan 6th, 2017. I rebutted several claims by Patzer [3e] including that an agile process was used.
- d. *Expert Report of Kal Toth Concerning Technical Report to Maverickeye*, May 10th, 2017. I compared the Maverickeye and Malibu technical reports demonstrating the equivalency of the systems generating them.
- e. *Second Expert Report of Kal Toth Regarding the Maverickeye Case*, Dec 24, 2017. I critiqued the "Functional Description" provided in the Declaration of Fieser [3g], and provided a preliminary analysis of the IPTRACKER source code [3h] observing that NARS is adapted open source software.
- f. *Third Expert Report: Assessment of MaverickMonitor Software Reliability*, February 27, 2018. I assessed the reliability of NARS, a software-based forensics tool used to detect the IP address of alleged copyright infringers of videos shared across BitTorrent networks.

5. Assessment of the Reliability of NARS

I have reviewed NARS and MaverickMonitor which are closely related, if not identical, software-based forensics tools used to detect IP addresses of users alleged to be infringing video copyrights using BitTorrent technology. NARS and MaverickMonitor are operated in Germany. Users (a.k.a. "peers") install BitTorrent client software that support BitTorrent protocols to efficiently share files including software distributions and videos. Patzer [3b], Perino [3f], and Richter [3i] have asserted that these tools are 100% accurate and free of defects. I do not agree with their assessments.

5.1 System Description (Context)

Exhibit A, *NARS System Context* depicts the context of my analysis (follow ❶, ❷, ❸, ❹ and ❺ on the figure).

5.1.1 BitTorrent Network ❶

Partially described in Exhibit D (BitStalker article), users cooperate with each other to share files using BitTorrent software modules (“clients”) installed on their personal computers by leveraging BitTorrent Trackers and Torrent files that are hosted by well-known various service providers across the web.

5.1.2 NARS System ❷

Operated by Excipio, NARS is a software-based system tool used to detect the sharing of selected video files of their customers among BitTorrent peers (users). NARS is connected to the Internet by way of an ISP. Operators (IPP) specify the names of the video files to be tracked while NARS repeatedly searches BitTorrent “swarms” using an unspecified probing technique, and generates reports tracking the IP addresses of peers from which they have received “pieces” of the files. Shared video files detected by NARS are typically around 40 minutes in length, consist of a few hundred pieces, and consume in the vicinity of 800 Mbytes of storage. NARS reports users as infringing after a few pieces (ranging from 16 Kbytes to 2 Mbytes) of a given file have been detected (received) from them. All the pieces of a given tracked file are rarely captured from a given IP address by NARS. But all pieces of a tracked file are usually captured from the IP addresses of peers participating in the swarm.

5.1.3 No Evidence Supporting Claims that NARS is Reliable ❸

In my expert reports [4a, 4b and 4c], I have documented the lack of technical documentation provided about NARS. There is no evidence that NARS developers produced documents or specifications of NARS’ such as, a theory of operation documenting the methods used by NARS to probe BitTorrent for infringing content; requirements and architecture documentation showing how NARS was designed to implement these methods and report infringement; technical reviews, test case procedures and expected results to demonstrate how comprehensively NARS was tested to show requirements were met and the software does not contain critical defects (a.k.a. faults); and quality assurance, bug tracking, configuration management, and release processes and procedures to demonstrate that the NARS software can be reliably maintained throughout its operational life.

5.1.4 Accuracy of IP Addresses and How they Map to Infringing Content Unknown ❹

Given the virtual absence of technical artifacts provided to me, I conclude that the NARS software code base is not reliable. It is therefore not surprising, to me, that NARS is not offered as a commercially available product. This in turn implies that the risks are very high that the software contains many latent defects that would cause NARS to incorrectly match IP addresses to pieces of copyrighted videos being captured by NARS. These sort of deep-rooted defects cause silent and subtle errors, particularly in heavily loaded concurrent processing / real-time systems. Consider the classic Therac-25 case which I have document in Toth [4a].

Experts connected with this case and the related cases have provided little or no visibility into the technical implementation of the NARS system. Investigations by plaintiffs have relied on brief descriptions of rudimentary test cases conducted by Paige [3b], and Richter [3i] and Bunting [3l] and other experts reporting on behalf of plaintiffs. I will refer to these as “demonstration tests”.

It was evident that these experts ran confidence tests primarily aimed at convincing non-technical observers that NARS “works”. Given the lack of technical documentation, these tests did not convince me that NARS works correctly most of the time or even part of the time. To date I have not received any evidence that NARS was subjected to tests designed to show it operates reliably and correctly under representative operating conditions, typical failure modes, and busy periods.

As documented in my expert reports [4d, 4e and 4f], I have examined MaverickMonitor which is also advertised as a BitTorrent infringement detection system. The reports produced by these two forensics tools are virtually identical. NARS and MaverickMonitor are also identical with respect to the lack of technical specifications provided.

In my expert report [4e], I had the opportunity to review MaverickMonitor's code base which consists entirely of open source software amounting to 140,000 source lines of code (140 KSLOC). I am of the opinion that NARS and MaverickMonitor are composed of similar, and perhaps identical, software code bases.

5.1.5 Only Partial Evidence of Infringement Reported by NARS ⑤

I have observed the following declarations attesting to the amount of data reported by IPP's software (NARS):

- (a) Fieser [3m] page 3, item 9, declared "IPP's software additionally analyzed each BitTorrent "piece" distributed by Defendant's IP Address. It verified that reassembling the pieces using a specialized Bitorrent client results in a fully capable movie."
- (b) The demonstration tests described by Paige [3b], and Richter [3i] and Bunting [3i] attest that NARS is capable of downloading all pieces (100%) of the test video files they used to show that NARS is capable of downloading all the pieces of their test files.
- (c) I analyzed the Pcap data associated with this case, and estimated that IPP's software captured, on the average, about 0.007% of a movie from the IP address purported to be infringing.

Of the 87 movies tracked by IPP: 0 (zero) pieces were captured from 10 of the movies; and either 1 or 2 pieces were captured from the remaining 77 movies. On the average, 1.38 pieces were captured per movie which at 16 Kbytes/piece means that 20 Kbytes of each movie were captured. Since the movies 306 Mbytes in size, only 0.007 of a movie was typically captured.

I conclude the following from this aspect of my analysis:

- a. Mr. Fieser's observation is that neither Mr. Fieser's declaration nor the demonstration tests of Paige, Richter or Bunty, provide credible evidence that the IPP software captures actually all pieces of a movie when in normal operating mode; and
- b. It may well be the case the IPP's software (NARS) is not capable of capturing all pieces of a given video file from any given BitTorrent user except under special conditions. I discuss this below.

5.1.6 Abandoned Sharing Reported as Infringement ⑥

Previously, I have also commented about the problem of abandoned BitTorrent sharing. An innocent BitTorrent user who normally uses BitTorrent to share content legally, say open source software programs, could accidentally click on a link and start unintentionally capturing a copyrighted video file. She may not notice this problem until returning with her mug of coffee. She then realizes her BitTorrent client software is capturing unwanted content, cancels her download, and deletes the partially downloaded video file. However, NARS, has been monitoring the sharing of this copyrighted content and jumps in before she manages cancel and delete. Unknowingly, NARS captures these pieces from her BitTorrent client and reports her as an infringing user.

5.1.7 Identity of Purported Infringer Ambiguous ⑦

NARS normally identifies purported infringers by searching BitTorrent Trackers using the names of copyrighted video files they wish to track. Typically, NARS tracks many IP addresses simultaneously. Exhibit A illustrates a representative household with a single Internet router connected to an ISP (e.g. Comcast). The router may be available for use by the subscriber, family members, tenants, and guests. Neighbors and (drive-by/walk-by) "lurkers" may also be able to connect to the router either because the router is password-less, or because the owner never bothered to change the default password and the lurker knows the common routers and defaults used when they come out of the box. It may also happen that the computer of a member of such a household

becomes infected by way of an email phishing exploit which enabled remote malicious party to download copyrighted video content in BitTorrent thereby implicating the owner of copyright infringement.

5.1.8 Plaintiff's Investigation Relies on Ambiguous, Incomplete and Unproven Factors ⑧

Exhibit A also depicts the investigation process conducted on behalf of the plaintiffs. I believe this process relies excessively on the asserted accuracy of NARS. NARS' lack of specifications and processes suggests that NARS is an unreliable software-based tool that could accuse innocent parties of infringement. This means that under heavy workload conditions NARS may inconsistently and inaccurately map monitored pieces of video content to detected IP addresses. Meanwhile, NARS only captures a small number of pieces from the purported infringer's IP address. And it turns out that this an innocently downloaded the wrong file, aborted the session, and deleted the file. Finally, the IP address happens to have about 10 users sharing the computer.

In other words, the copyright investigation process relies on a combination of ambiguous, incorrect, and incomplete information while using unproven, and hence unreliable software.

5.2 NARS Reliability Assessment

My reliability assessment of NARS explores the following questions:

- Is NARS reliable enough to conduct forensics investigations?
- Is partial evidence of downloaded videos sufficient evidence to conclude copyright infringement?
- Is a monitored IP address enough to suspect an ISP subscriber, family and friends of infringement?

Now refer to Exhibit B which depicts my reliability assessment of NARS in the above context. My assessment examines the likelihood that infringement is correctly detected. The figure in Exhibit B depicts four branches of the reliability analysis: ①, ②, ③, and ④.

5.2.1 Demonstration Test ①

The demonstration tests described by Paige, Richter and Bunting involved setting up three or four test computers with installed BitTorrent clients connected to Internet ISPs configured to share a few predetermined video files by way of BitTorrent. All of these test descriptions confirmed that all the pieces (100%) were detected by NARS. It has been surprising to me is that NARS generates an infringement report after only few pieces have been captured.

After some reflection, it was not surprising to me that these test cases demonstrated that all of the pieces of the test video files were detected from a given IP address. Simply stated, these simplistic tests demonstrate nothing about the reliability of NARS when operated under real-world operating conditions. For example, they do not attempt to address the problems associated with routers using dynamic IP addressing (i.e. IP address resets), or conduct tests that try to determine if more than one user is attached to the router, or that the user has aborted an unintended download. At the very least, these tests should have simulated router resets by powering them down and rebooting them during file sharing, and by running scenarios where BitTorrent users abort the downloading of shared video files before completion. Such operationally representative tests would have confirmed whether NARS could cope with unusual circumstances and events.

Another shortcoming was that these demonstration tests did not document the operating workloads during the test runs, or the number peers participating in the BitTorrent swarm during the period of the tests.

5.2.2 Number of Detected Pieces of Copyrighted Content ②

- a. The IPP software (NARS) reports detected videos as infringing after only a few pieces (less than 1%) are downloaded from a monitored IP address, sometimes fewer. None of the NARS reports I have inspected have provided evidence that NARS waits to detect all pieces of a shared video file to be received from a monitored IP address before reporting infringement.

- b. The fact that NARS assumes infringement after only a few pieces have been detected from a monitored IP address means that someone aborting a session because they made an honest mistake, would be wrongfully accused of copyright infringement..
- c. I understand that partially downloaded Bit Torrent files may require technical skills to view them by means of standard video players. My search of online blogs confirm that VLC and AVI Preview are two such players that can be used to view video files that have been partially downloaded using BitTorrent client software. However, the experience is “choppy” in proportion to the number of missing pieces and how contiguous they are). Furthermore, these players will not render a partially downloaded video if the first part of the video is missing. Since BitTorrent shares pieces randomly, a user is unlikely to be able to play the content until a large percentage of a given video is captured. This means that users attempting to do this need to have technical knowledge and skills, as well as patience, to view such partially downloaded videos. Whether being in possession of such partially downloaded, and potentially unplayable videos constitutes infringement is a legal question outside my scope.

5.2.3 NARS Software Reliability ③

- a. **Unproven Software-based Forensics Tool:** Brunty’s article ([1b] Exhibit D) explains that NIST standards require that forensics software and tools be repeatable and reproducible. Given that virtually no technical specifications or processes have been provided to support the claim that NARS was developed using best software engineering practices, one can only conclude that NARS fails to meet the NIST standard. Using Brunty’s arguments, it can be argued that NARS is not reliable enough to detect IP addresses consistently and correctly, and hence the reports output by NARS may not be reliable enough to be admissible as electronic evidence for forensics purposes (infringement detection).
- b. **Large Number of Latent Software Faults:** I have studied, conducted software process assessments, and taught the widely respected principles of the Software Engineering Institute’s Capability Maturity Model (CMM) [1c] throughout my career. I have also relied on the *software reliability tutorial* by Gullo and Peterson ([1d] Exhibit E) which tabulates empirically derived software fault densities for each CMM level from CMM Level 1 to CMM Level 5. Given the dearth of specifications and processes used to develop NARS, I conclude that NARS must have been developed at the lowest level, CMM Level 1 “Initial” (a.k.a. “Ad Hoc”). Using Gullo and Peterson’s CMM fault density table for CMM Level 1, and the expected software size for NARS of 140,000 software lines of code (140 KSLOC), I have estimated that NARS has between 700 and 4,200 latent faults (a.k.a. defects). Of course, only a fraction of these defects would have critical impacts on NARS operations. However, the high fault density levels associated with CMM Level 1 do lend credibility to NIST’s repeatability and reproducibility standard for forensics software which aligns with CMM Level 2 “Repeatable”. Observe that using fault densities at CMM Level 2 would yield a 10 fold decrease in my fault estimates.
- c. **False Positive Rate is about 11%:** The BitStalker ([1a] Exhibit C) article by Bauer et.al. states that traditional ping probing techniques used to identify file sharing in BitTorrent networks detect IP addresses falsely about 11% of the time. The article demonstrates that BitStalker’s proposed active probing technique can achieve accuracy close to 2%. However, validating information, such as a theory of operation document, has not been provided to confirm which probing method is used by NARS. This means that with information available, the best NARS could hope to achieve would be an 11% false positive rate. Of course this rate could only be achievable if the NARS software was free of all defects (i.e. “bugs”).

5.2.4 Accuracy with which an IP Address can Identify Infringer ④

It has been asserted that the IP address reported by NARS is a strong enough indicator of infringement to warrant the issuance of a subpoena to the ISP. However, there are several scenarios where an innocent party, including the subscriber, could be wrongfully accused. Exhibit A depicts some of these cases.

- a. More than likely NARS has no simple way of uniquely identifying the router using a monitored IP address to verify over a period of monitoring that all the traffic is passing through the same router (i.e. no resets; or to know whether the router is password-protected. Furthermore, NARS cannot be sure whether there is only a single person using the router (i.e. the subscriber), or a large number of persons routinely using it to access the Web.

- b. For example, a household comprised of the subscriber, several family members and/or tenants, visiting guests, and neighbors/lurkers within range of their Internet routers. All of these persons could be routinely using the router. Although cautious subscribers password-protect their routers, default passwords are often left unchanged and therefore guessable, and some people prefer leaving their passwords open. Sometimes passwords cracked by a walker-by or drive-by. The default passwords of routers are infrequently changed and guessable.
- c. Exhibit A depicts 10 potential users. Any one of them could be the infringing user detected by NARS. This scenario illustrates that one of the other 9 persons would be innocent bystanders. This means that there is a 90% change that someone could be wrongfully accused.
- d. Although the population within and surrounding a household varies, in the absence of knowing where the ISP address is located at any given time means that it is unreasonable to assume that a detected IP address is attributable to a single person in most cases (i.e. the subscriber).
- e. The IP addresses allocated to routers are sometimes reset by ISPs, users reset them whenever the router hangs, and some reset them as a routine practice to guard against cybersecurity attacks. Power outages will also reset routers when power returns. Let us assume that the average router is reset, say, four times a month. Now let's assume that an infringer using our ISP is being actively monitored by NARS on a given IP address. That same day the infringer's IP address is reset and allocated to your router. This means that there is a risk that NARS will assume you are the infringer and arrange to send a subpoena accusing you of infringement.
- f. Another possibility is that by way of a phishing attack. Your computer is infected and a botnet takes control of your computer in the background, using it as a proxy to execute various unauthorized activities, including illegal video file sharing by way of BitTorrent. NARS tracking could identify the IP address of your router and hence accuse you as the ISP subscriber of copyright infringement.

6. Summary of Observations and Findings

The Excipio experts have asserted that NARS is 100% accurate and free of defects. In my experience, claiming that operational software is defect-free is not credible. Neither is asserting that a complex software-based system detects infringement flawlessly. If a system is expected to be highly accurate, and advertised as such, the software engineering and development processes must be sufficiently capable and mature, should be guided by credible standards, and be supported by experienced personnel and proven tools. There is no evidence that capable software processes, technical specifications, comprehensive testing, or quality assurances were conducted in the development of NARS. My reviews of Patzer [3b], Perino [3f], Fieser [3g] and Richter [3i] confirmed that a well-articulated expression of the intended purpose of NARS and a suitable theory of operation were not provided.

Principle Reliability Assessment Findings: The lack of technical specifications and process documentation confirm that NARS does not meet NIST's standard of repeatability and reproducibility for forensics software tools. NARS must have been developed using ad software engineering processes consistent with CMM Level 1 which suggests that NARS contains a large number of software faults (bugs). NARS is therefore a relatively unreliable system that should not be trusted to detect IP addresses accurately or consistently.

- a. Under normal operating conditions, NARS seems to be unable to download all the pieces of a video file from an IP address suspected of infringement. This means that NARS is cannot distinguish infringing peers from those who have intentionally aborted downloads because they made a mistake. from a swarm. Partially downloaded video content may not be viewed in many cases, and many users don't have the necessary knowledge, skills or patient to figure out which video player to use, and what procedures to use to view a partially downloaded video. This raises the question of whether detecting partial downloads is enough evidence of infringement.
- b. Relying only on the IP address of purported infringement detection by NARS is not enough to assert infringement in many cases. Shared use of a router by the subscriber, family, and friends in an area where neighbors could also camp on the router represents a common situation where a number of users sharing a router could be wrongfully accused of infringement. Router resets and infected computers overtaken by bots also represent scenarios where users could be wrongfully accused.

Possible Undisclosed Problem with NARS: Given the random nature of BitTorrent sharing of pieces, it may not be possible, in all circumstances, especially when swarms are busily sharing pieces, for NARS to capture all pieces of a targeted file from a monitored IP address. More specifically, this may be because BitTorrent is designed for peers to share pieces with many other peers. Once a peer has collected all the pieces wanted, he/she would most likely close the BitTorrent client, even if not finished sharing. NARS lose the opportunity to capture all the available pieces from that peer.

This also seems to help explain why NARS is able to detect all the pieces of test video files configured by Paige, Richter and Bunting. Because their test files have uninteresting titles, are watermarked, and are relatively short, other BitTorrent peers will not be interested in sharing pieces with the test computers. And the test computers will not stop sharing pieces with NARS until the test is declared done.

Fieser's Declaration: I have also observed the following with respect to Fieser's declaration [3m] page 3, item 9, where he declared "IPP's software additionally analyzed each BitTorrent "piece" distributed by Defendant's IP Address. It verified that reassembling the pieces using a specialized Bitorrent client results in a fully capable movie." My analysis suggests that Fieser's assertion does not accurately represent the facts of the matter. I can report that IPP's software captures, on the average, only about 0.007% of a movie from a monitored IP address. This would seem to explicitly contradict Fieser's apparent implication that IPP's software is capable of verifying that a defendant would be to receive and reassemble all of the pieces of a movie, and successfully play said movie.

My rate is \$350.00 per hour.

Handwritten signature of Kalman C. Toth in black ink.

Signed under the Penalty of Perjury,
Kal Toth (Kalman C. Toth), Ph.D., P.Eng.

Annex: My Most Relevant Experience and CV (Kal Toth)

My Most Relevant Qualifications

My name is Kal Toth. I have a Ph.D. in computer engineering from Carleton University and am a professional engineer (P.Eng.) with a software engineering designation registered in BC.

I have practiced in the fields of software and quality engineering, information security, e-commerce, mobile systems, and distributed database systems. My detailed CV below covers my work history and key projects in industry and at universities, also listing my conference and journal publications, industry reports, university courses, and delivered seminars.

Independent Validation and Verification (IV&V)

As Vice President of Systems Engineering for the CGI Group, I led a 3rd Party Validation and Verification team hired by the Canadian Federal Government to oversee their prime contractor's \$50M development of a security-critical global messaging system for Canada's embassies abroad. The primary purpose of this project was to ensure that the prime contractor's development teams developed adequate plans, requirements specifications, designs and test procedures, and executed their plans, reviews and procedures according to their obligations and standards called up under their contract with the Canadian government.

Quality, Reliability, Maintainability, Safety, Security, and Software Engineering

As Director of Quality at Hughes Aircraft of Canada, Systems Division, I lead my team's quality assurance, reliability, maintainability, availability and safety engineering tasks supporting the development of five (5) large software-intensive Air Traffic Control (ATC) systems, including Canada's new ATC system, a \$500M project. I also supported the security working group for the project. My responsibilities included leading the development of our division's new software development methodology, including software requirements, architecture, development, testing and metrics processes, promulgating the division's transition from a traditional plan-based software process to a more flexible iterative software development process.

Software Engineering Practice Leader

At CGI Group (VP Total Quality) and Hughes Aircraft (Director of Quality) I also had the role of software and systems engineering practice leader. I organized working groups and gave seminars aimed at developing skills in the areas of software project management, software processes, software quality assurance, professional issues, process improvement, and metrics.

Software Engineering Programs and Courses

I later joined academia as an Associate Professor teaching software engineering, architectural design, quality, and project management courses to working professionals at the Technical University of BC, the University of British Columbia, Simon Fraser University, Oregon State University and Portland State University. I was the Director of the Oregon Master of Software Engineering program and the Executive Director of the WestMost consortium teaching software technology courses to working professionals across nine (9) universities in western Canada.

VP Engineering for a Real-Time Web-Centric Real-Time Alert System

As Vice President of Engineering for Datalink Systems Corp I managed an agile team of ten software engineers, programmers and testers developing and maintaining a real-time alert system sending stock quotations to mobile devices of customers managing their portfolios online. The system ran on a server farm of a dozen physical servers, supported by an SQL database system. I established and shaped an iterative software development process for the team including functional and design specification, peer-reviews, independent module testing and system integration testing.

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

304-1132 SW 19th Ave Portland OR 97205
kalmanctoth@gmail.com 503.984.3531

Security, Software, Quality, and Systems Engineering Professional

Background / Experience:

- In leadership positions with technology companies in the fields of security, software and IT
- Software, systems and security-related engineering innovator, consultant, and change agent
- Technology solutions and consulting in government, financial and selected industry sectors
- Cybersecurity, identity management, e-commerce, mobile computing, distributed systems, networking, communications, and databases.
- Air traffic control; real-time stock quotation for mobile devices; search and rescue system, security devices and gateways; global secure messaging network; on-line learning systems
- Systems engineering evangelist: traditional and agile software development, project management
- Software engineering, IT and project management courses and training for working professionals.

Competencies:

Systems, security, software and quality engineering, Strategic and business planning, Project management, Digital Identity technology and security engineering, e-learning/distance education

Citizenship and Residency: U.S. Citizen, U.S. Resident, also a Canadian Citizen

Languages: English (mother tongue), Hungarian (father tongue), and French fluency

World: Early IT career with World Health Organization, Geneva, Switz; well-travelled in Europe

Education:

B. Eng. Electrical Engineering

M. Eng. Systems Engineering and Computer Science

Ph.D. Computer Systems Engineering

Professional Engineer (P. Eng.): BC Association of Professional Engineers and Geoscientists

Training/Education Courses: E-commerce, SW engineering, project management, prof. issues (i.e. IP)

Pacific Northwest Software Quality Conference: Board member and 2013 Conference Chair

Portland State University: Faculty Senate Budget Committee; Intellectual Property/DistEd Taskforce

Goose Hollow, Portland Oregon: neighbourhood association Board of Directors

Patent: "Electronic Identity and Credentialing System", US Patent No. 9646150, Apr 20/17.

Patent: "Methods for Using Digital Seals for Non-Repudiation of Attestations", Aug 20/17

Patent-Pending: "Registering and Acquiring E-credentials using Proof-of-Existence & Digital Seals", Feb 18, 2018, No. 15/898,217.

Patent-Pending: "Portable Caching System" submitted in 2007, abandoned in 2015

Expert Reports: copyright infringement cases, multiple expert reports, depositions

Key Positions / Appointments: listed

Expert Reports: listed

Publications, Industry Reports, and Courses: listed

See also <http://www.linkedin.com/pub/kal-toth/2/60b/b19>

Key Positions / Appointments

NexGenID (2013 - 2014), CEO and CTO

- Created innovative identity and credentialing technology: "Electronic Identity and Credentialing Technology" per above-referenced patent and patent-pending identity technology
- Developed detailed functional specification and proof-of-concept for digital identity prototype_(Android-based)

aTrust Inc. (2012 - 2013), Chief Technology Officer (CTO)

- Progressed startup's vision for digital identity, technology roadmap, and product-line development strategy
- Built and maintained partner/vendor relationships in technology and banking sectors
- Managed and evaluated the distributed development team's progress and performance

Portland State University (2003-12), Executive Director and Associate Professor

- Directed, enhanced and evolved the Oregon Master of Software Engineering (OMSE) into a fully online learning program for working software professionals in Oregon's hi-tech sector
- Delivered software engineering, project management, quality engineering, distributed team, estimating, and architectural design courses and seminars – both face-to-face and online
- Investigated identity management technologies targeted at the healthcare and banking sectors creating the "Persona Concept", a framework for managing electronic credentials and private data of users across PCs, smart cards, smart phones, and other personal devices

Oregon State University (2001-2003), Associate Professor Computer Science

Technical University of British Columbia (1999-2001), Assoc. Professor Information Technology

Datalink Systems Corp. (1997-99), Vice President Engineering

- Following a light-weight agile software development process, directed development and operations
- Led the development of a web-based service and payment processor for delivering real-time stock quotes, news, sports, and other services to wireless devices - pagers and cell phones
- Worked with marketing/support to develop requirements and rapid response to user problems
- Removed security weaknesses of the previously deployed service center
- Developed replacement architecture with scalability, backup and recovery features

Hughes Aircraft Systems Division (1992-95), Director of Quality

- Led quality, reliability, maintainability, availability and system safety teams for five (5) large air traffic control projects (Canada, Canadian military, Switzerland, Indonesia and China)
- Leading member of the core team transitioning division from a waterfall to an iterative software process which guided the development of Canada's \$400M air traffic control system ("CAATS")
- Created a new process infrastructure for the division's policies, practices and procedures

CGI Group Inc. (1988-1992), Vice President Systems Engineering, Vice President Total Quality

- Practice leader across CGI's 10 regional offices for project management, software engineering, quality engineering, configuration management, and software estimating
- Led process improvement initiatives across CGI's US and Canadian offices
- Developed and initiated a strategic plan to implement a company-wide total quality process
- Conducted independent verification and validation of a \$50M project to develop a globally secure network across Canada's embassies abroad for External Affairs Canada
- Developed an innovative information security analysis model for Defence Canada

Intellitech Canada Ltd. (1983-88), Founder and President

- Founded Intellitech, growing it into a 25-person systems engineering and consulting firm
- Conducted numerous design and development projects for distributed information systems, networks and security gateways for military, government and industry clients
- Led the development of Intellitech's secure packet-network product and the delivery of prototypes to Communications Canada – funded by the Canadian National Research Council and the Bank of Montreal, and sponsored by the Communications Security Establishment

Carleton University (1980-83), Assistant Professor, Systems Engineering and Computer Science

Expert Reports: Intellectual Property (copyright infringement) Cases

- Expert reports (3) for JC Edmondson law office for defendant in a copyright infringement case, 2016-17
- Expert reports (3) for JC Edmondson another defendant in a copyright infringement case, 2017-2018

Publications and Seminars in the Field of Security, Identity and Authentication

- Kalman C Toth, Brewing Next Generation Identity, Pacific Northwest Software Quality Conference, Oct 2015
- Kalman C. Toth, A Practical Identity Management Reference Implementation, International Conference on Computers and Their Application (CATA), Honolulu, Hawaii, March 28-30, 2007
- Kalman Toth, Persona Concept for Web-Based Identity Management, 2006 International Conference on Privacy, Security and Trust, UOIT, Newmarket, Ontario, Oct 30-November 1 2006
- "Identity Management Systems", tutorial for IEEE International Computer Software and Applications Conference (COMPSAC), Chicago, September 2006
- Information security seminars for the Assoc. of Prof. Engineers and Geoscientists of B.C., 2002 and 2006
- K.C. Toth, M.Subramaniam, Requirements for the Persona Concept, Requirements for High Assurance Systems (RHAS'03) workshop, Monterey, CA, September 9, 2003
- K.C. Toth, M. Subramaniam, The Persona Concept: A Consumer-Centered Identity Model, MobEA (Emerging Applications for Wireless and Mobile Access), Budapest, Hungary, May 2003
- K.C. Toth, M. Subramaniam, Persona Concept for Privacy and Authentication, International Business & Economics Research Journal, June 2003
- K.C. Toth, M. Subramaniam, I. Chen, Persona Concept for Privacy and Authentication, International Applied Business Research Conference, Acapulco, Mexico, March 2003; recipient of best paper award
- K.C. Toth, M.Donat and J. Joyce, Generating Test Cases from Formal Specifications, 1996 International Council of Systems Engineering (INCOSE) Symposium, July 1996
- M.W.L. Dennison, K.C. Toth & J.F. Clayton, Using a Practical Approach to Threat/Risk Analysis, Third Annual Canadian Computer Security Conference, Ottawa, May 14-16, 1991
- K. Toth, Information Security Architectures, AFCEA '90 (Armed Forces Communications & Electronics Association Technical Conference), Hawaii, November, 1990
- K.C. Toth, Security Architectures for Information Networks, AFCEA Canada '90, April 1990
- H. Adra, J. Allen, K. Toth, Trusted Integrated Project Support Environments, Second Annual Canadian Computer Security Conference, Ottawa, March 1990
- K. Toth, Towards an Improved Information Security Model, 1st Canadian Comp. Security Conf, January 1989
- K. Toth, Security Management in Data Networks, 1st Annual Canadian Computer Security Conf, Jan 1989
- AC Capel, C Laferriere & K.C Toth, Protecting the Security of X.25 Comm's, Data Com Mag, November 1988
- M.W.L. Dennison, K.C. Toth & J.F. Clayton, Using a Practical Approach to Threat/Risk Analysis, Third Annual Canadian Computer Security Conference, Ottawa, May 14-16, 1991
- K. Toth, Information Security Architectures, AFCEA '90 (Armed Forces Communications & Electronics Association Technical Conference), Hawaii, November, 1990
- K.C. Toth, Security Architectures for Information Networks, AFCEA Canada '90, April 1990
- K. Toth, Towards an Improved Information Security Model, 1st Canadian Comp. Security Conf, January 1989
- K. Toth, Security Management in Data Networks, 1st Annual Canadian Computer Security Conf, Jan 1989
- AC Capel, C Laferriere & K.C Toth, Protecting the Security of X.25 Comm's, Data Com Mag, November 1988
- System Security and Recovery Procedures, Datalink Systems Corp, January 1999
- "EDI and Security", CGI Group report, Dec. 1990
- "COSICS Security Verification Plan", Intellitech report to External Affairs, December, 1988
- "Information Security Model", report to National Defence, November 15, 1988
- "Data Encryption Equipment Specification", Internal report specifying the components of CryptoNet, Intellitech's X.25/DES product, 1986
- "A New Implementation Strategy for Secure Operating Systems", Intellitech Report, March 1986
- "Design and Security Considerations for a Gateway to Interconnect SAMSON and DATAPAC", Report to the Department of National Defence, 1980

Conferences and Journal Publications

- Kalman C Toth, Brewing Next Generation Identity, Pacific Northwest Software Quality Conference, Oct'15
- Kalman C Toth, Herm Migliore, Critical Factors Characterizing Projects & Lifecycle Models, PNSQC, Oct'13
- Kalman Toth, Learning Software Engineering Online, Pacific Northwest Software Quality Conference, Oct'11
- Kal Toth, Organizational Approach for Sustaining E-Learning in Large Urban University, Future of Ed, Jun'11
- Kal Toth, Software Engineering Online and Hybrid Learning Models at PSU, CATA, March, 2011
- Kal Toth, Raleigh Ledet, Lessons Learned about Distributed Software Team Collaboration, PNSQC, Oct'10
- Kal Toth, Software Estimating: Navigating to Landing Zone, Computers & their App's, Honolulu, HI, Mar'10
- Kal Toth et. al., Distributed Software Engineering Team Collaboration, poster session, PNSQC, October 2009
- Kal Toth, Software Estimating, Flexibility and Principled Negotiation, Computers and their Applications in Industry and Engineering (CAINE), San Francisco, November, 2009
- Kal Toth, Selecting Software Estimating Techniques that Fit the Software Process, Pacific Northwest Software Quality Conference (PNSQC), Portland, Oregon, October, 2008
- Dan Brook, Kal Toth, Levels of Process Ceremony for Software Configuration Management, Pacific Northwest Software Quality Conference (PNSQC), Portland, Oregon, October, 2007
- Kalman C. Toth, A Practical Identity Management Reference Implementation, International Conference on Computers and Their Application (CATA), Honolulu, Hawaii, March 28-30, 2007
- Kal Toth, Experiences with Open Source Software Engineering Tools, IEEE Software, Nov/Dec 2006
- Kalman Toth, Persona Concept for Web-Based Identity Management, 2006 International Conference on Privacy, Security and Trust, UOIT, Newmarket, Ontario, Oct 30-November 1 2006
- L. Grove, R. Hickman, W. Matthews, K. Toth, Open Source Software Engineering Tools, Pacific Northwest Software Quality Conference (PNSQC), Portland, Oregon, October 12-13, 2004
- K.C. Toth, M.Subramanium, Requirements for the Persona Concept, Requirements for High Assurance Systems (RHAS'03) workshop, Monterey, CA, September 9, 2003
- K.C. Toth, M. Subramanium, The Persona Concept: A Consumer-Centered Identity Model, MobEA (Emerging Applications for Wireless and Mobile Access), Budapest, Hungary, May 2003
- K.C. Toth, M. Subramanium, Persona Concept for Privacy and Authentication, International Business & Economics Research Journal, June 2003
- K.C. Toth, M. Subramanium, I. Chen, Persona Concept for Privacy and Authentication, International Applied Business Research Conference, Acapulco, Mexico, March 2003; recipient of best paper award
- K.C. Toth and S. Nagboth, A Constraint-Based Personalization Model for E-Business Applications, International Applied Business Research Conference, Acapulco, Mexico, March 2003
- K.C. Toth, S. Nagboth, Intelligent Agents for Business Applications Using Constraint-Based Personalization, International Business & Economics Research (IBER) Journal, May 2002
- K.C. Toth, Software Product Evolution in the Classroom, American Society for Engineering Education / PSW, Fresno, California, April 8, 2002
- K.C. Toth, Simulating (Software) Product Evolution in the Classroom, The Western Canadian Conference on Computing Education (WCCCE), Nelson, British Columbia, May 3, 2001
- K.C. Toth and H. Todino, Instant Internet Intelligence for Wireless Business Applications, International Applied Business Research Conference, Cancun, Mexico, March 2001
- D Cyr, H Trevor-Smith, T Schiphorst & K.C Toth, A Web-Enabled Case Study in Project Management, International Business Education and Technology Conference, Cancun Mexico, March 2001
- K.C. Toth, M.Donat and J. Joyce, Generating Test Cases from Formal Specifications, 1996 International Council of Systems Engineering (INCOSE) Symposium, July 1996
- R. John, J. Madhur, R. Stewart, K. Toth, Software Quality Metrics Process For Large Scale Systems Development, 1996 INCOSE Symposium, July 1996
- K.C. Toth, J.J. Joyce, J. Masters, G. Pelletier, Precise, Unambiguous, Machine-Readable ATC Standards: Use of "Formal Methods" in the ATC Industry, ATCA Conference Proceedings, September 1995
- K Toth & J. Joyce, Industrialization of Formal Methods Through Process Definition, feature paper at the 1995 National Council on Systems Engineering Symposium, July 1995
- T. Paine, P. Kruchten & K. Toth, Modernizing ATC Through Modern Software Methods, Proceedings of the 38th Annual Air Traffic Control Association, Nashville, Tennessee, October 1993
- M.W.L. Dennison, K.C. Toth & J.F. Clayton, Using a Practical Approach to Threat/Risk Analysis, Third Annual

- Canadian Computer Security Conference, Ottawa, May 14-16, 1991
- K. Toth, Information Security Architectures, AFCEA '90 (Armed Forces Communications & Electronics Association Technical Conference), Hawaii, November, 1990
- K.C. Toth, Security Architectures for Information Networks, AFCEA Canada '90, April 1990
- H. Adra, J. Allen, K. Toth, Trusted Integrated Project Support Environments, Second Annual Canadian Computer Security Conference, Ottawa, March 1990
- K. Toth, Towards an Improved Information Security Model, 1st Canadian Comp. Security Conf, Jan 1989
- K. Toth, Security Management in Data Networks, 1st Annual Canadian Computer Security Conf, Jan 1989
- AC Capel, C Laferriere & K.C Toth, Protecting the Security of X.25 Comm's, Data Com Mag, November 1988
- K.C. Toth, S.A. Mahmoud, J.S. Riordon, Query Processing Strategies in a Distributed Database Architecture, Distributed Data Systems, North-Holland Publishing Co., 1982
- K.C. Toth, S.A. Mahmoud & J.S. Riordon, An Approach to Query Processing in Distributed Databases, Proceedings of the Sixth International Conference on Very Large Data Bases, Montreal, 1980
- Kalman C. Toth, Distributed Database Architecture & Query Processing Strategies, Ph.D. Carleton U 1980
- S.A. Mahmoud, J.S. Riordon & K.C. Toth, Distributed Database Partitioning & Query Processing, G. Bracchi and G.M. Nijessen (ed), Data Base Architecture, IFIP, North Holland, 1979
- S.A. Mahmoud, J.S. Riordon and K.C. Toth, Distributed Database Partitioning and Query Processing Strategies, IFIP Conference on Database Architecture, Venice, June, 1979
- J.S. Riordon, S.A. Mahmoud, K.C. Toth & O. Sherif, Distributed Database Architecture and Query Processing, CIPS/DPMA, Quebec City, June 1979
- K.C. Toth, S.A. Mahmoud, J.S. Riordon, O. Sherif, The ADD System - An Architecture for Distributed Databases, Proc. of the 4th International Conference of Very Large Data Bases, Berlin, September 1978
- S.A. Mahmoud & K.C. Toth, Design Considerations for a Mini-Computer Database, MIMI International Conference, Zurich, June 7-9, 1977
- S.A. Mahmoud, J.S. Riordon & K.C. Toth, Design of a Distributed Database File Manager for a Mini-Computer Network, COMPSAC77, Chicago, November 8-11, 1977
- Kalman C. Toth, Contributions to the Synthesis of Computer-Communication Networks, M.Eng. Thesis, Carleton University, Ottawa, April 1972

Trade Articles

- "What's the hard part of software development anyway?", Software Assoc. of Oregon, Nov. 2007
- "Better Mileage with Hybrid Learning", with Kathy Milhauser, Software Assoc. of Oregon, June 2007
- "Can Software Engineers Develop Communications Skills Online?", Software Assoc. of Oregon, March 2007
- "Is Online Software Engineering Education for You?", Software Association of Oregon (SAO), Feb 2007
- "OMSE Exchange: A Software Engineering Clearing House", Software Assoc. of Oregon (SAO), Nov 2006
- "So Many Engineering Practices: Which to Follow?" (Part III), Software Assoc. of Oregon (SAO), July 2005
- "So Many Engineering Practices: Which to Follow?" (Part II), Software Assoc. of Oregon (SAO), June 2005
- "So Many Engineering Practices: Which to Follow?" (Part I), Software Assoc. of Oregon (SAO), May 2005
- "Which is the Right Software Process for your Problem?", Software Assoc. of Oregon (SAO), April 2005
- "Outsourcing Software Development: A Case for Effective Scope Management", SAO, March 2005
- "Why Invest in Software Engineering Education?", SAO, February 2005
- "EDI and Security", CGI Group report, Dec. 1990
- "COSICS Security Verification Plan", Intellitech report to External Affairs, December, 1988
- "Information Security Model", report to National Defence, November 15, 1988
- "Data Encryption Equipment Specification", Internal report specifying the components of CryptoNet, Intellitech's X.25/DES product, 1986
- "A Survey of Integrated Project Support Environments", Report to the Department of National Defence, 1986
- "A New Implementation Strategy for Secure Operating Systems", Intellitech Report, March 1986

Industry Reports

- "Requirements for SimbaERP", report to Simba Technologies on the Requirements for a proposed ERP/Data Warehousing product, January, 1999
- System Security and Recovery Procedures, Datalink Systems Corp, January 1999
- "Technology Skills Gap Analysis: B.C. Software Industry", under contract to the Software Development Centre (B.C.) for B.C. Ministry of Education, Skills & Training, and National Research Council, March 1997
- "Process Product Standard", internal Hughes System Division Report, June 1994
- "In-Process Review (IPR) Process", internal Hughes System Division Report, December 1993
- "Change in Development Methodology", internal Hughes Systems Division Report, June 1, 1993
- "Total Quality Implementation Program", internal CGI report to the Management Committee, 1991
- "Total Quality Process: Directions & Priorities", internal CGI report to the Management Committee, 1991
- "TQP: Client Satisfaction Assessment Process", internal CGI guide, 1991
- "Software Quality Assurance Program", internal CGI practice guide, 1990
- "Configuration Management Framework", internal CGI practice guide, 1990
- "EDI and Security", CGI Group report, Dec. 1990
- "COSICS Security Verification Plan", Intellitech report to External Affairs, December, 1988
- "Information Security Model", report to National Defence, November 15, 1988
- "Network Processing Strategy Study", a series of reports to Transport Canada, 1988
- "Data Encryption Equipment Specification", Internal report specifying the components of CryptoNet, Intellitech's X.25/DES product, 1986
- "A Survey of Integrated Project Support Environments", Report to the Department of National Defence, 1986
- "A New Implementation Strategy for Secure Operating Systems", Intellitech Report, March 1986
- "Computer System Study" (Computer Integrated Manufacturing and Manufacturing Requirements Planning), Reports to General Metals Co, El Naser Glass Co. and Delta Steel Mills, 1985/86
- "Search and Rescue Satellite (SARSAT) Aided Tracking System, Ground System Study", five reports regarding Mission Control Centre design to National Defence, 1983 and 1984
- "Design Specification for the NCCS Communications Management System", Atmospheric Env. Serv, Jan 84
- "Design & Analysis of Alternatives for the Integrated Data Network", Report to the Dept. Nat'l Defence, 1982
- "Recovery Mechanisms for the ADD Distributed Database System", Intellitech Report, July 1982 (also presented at a NATO workshop in 1982)
- "Implementation Alternatives and Gateway Considerations for a Data Network to Serve the Defence Research Establishments", Report to the Department of National Defence, 1981
- "Design and Security Considerations for a Gateway to Interconnect SAMSON and DATAPAC", Report to the Department of National Defence, 1980
- "Open System Interconnection: Application Issues Associated with the ISO and CCITT Layered Models", report to the Department of Communications, 1980
- "On Query Decomposition & Processing in Distributed DBs", INRIA Research Report, Spyrtos & Toth, 1980
- "Query Processing Strategy Formulation in ADD", Carleton University report, 1979
- "A Modeling Approach to Systems Analysis of Processing Networks", one of five reports to the Department of Communications, Spectrum Management Systems
- "Design Issues in Distributed Databases", Carleton University report
- "Design & Configuration Analysis of an Aeronautical Satellite Comm. Centre (ASCC)", Transport Canada

Workshops, Seminars, Tutorials, Professional Training Courses

- Professional Development Course in Software Engineering for Regence Group, Portland, Or, June 2007
- "Identity Management Systems", tutorial for IEEE International Computer Software and Applications Conference (COMPSAC), Chicago, September 2006
- Information security seminars for the Assoc. of Prof. Engineers and Geoscientists of B.C., 2002 and 2006
- Extending the Reach of Mobile E-Commerce, Software Productivity Centre, June 2000
- Wireless Handheld Technologies and Telelearning, Telelearning Conference, Toronto, November 2000
- E-Commerce Lifecycle, Transactions and Security, MacDonald Dettwiler & Assoc., November 1999
- Personal Software Process (PSP): Software Productivity Centre / MacDonald Dettwiler & Associates, 1997
- WestMOST Software Engineering Telelearning Workshop, Saskatoon, 1998
- Software Project Management (including software process and metrics) at Carleton University, Dec 1994
- Software Development Methods and Process: Iterative Software Development, for the Canadian Automated Air Traffic System (CAATS) at Hughes Aircraft, Systems Division and Transport Canada, March 1993
- Canadian Automated Air Traffic System, seminars presented at UBC (Computer Science), SFU (Applied Sciences), and Hughes (for staff and graduate students from UVIC, BCIT, SFU and UBC), 1993 and 1994
- Total Quality Management, seminars presented to CGI Group technical staff across Canada, 1991 and 1992
- Total Quality Management, lecture to 4th year computer systems engineers at Carleton University, 1991
- Information Security Technology Overview for AFCEA INFOSEC Course, Canadian Forces Base (CFB) Kingston, October 1991

University Undergraduate and Graduate Courses

For Portland State University:

- Principles of Software Engineering
- Software Project Management
- Software Quality Engineering
- Software Design Techniques
- Software Estimating
- Distributed Software Engineering Team Collaboration
- Software Engineering Practicum
- Computing Fundamentals II (Visual Basic)
- Senior Capstone projects
- Directed studies: IT and software engineering

For Oregon State University:

- E-Commerce Systems
- Software Engineering I: principles, processes, requirements, OO design, architecture, SPM
- Software Engineering II: implementation, SCM, test techniques, reviews and inspections, SQA

For the Technical University of British Columbia and the University of Alberta:

- Software Engineering Best Practices
- E-Commerce Systems

For the University of British Columbia and Simon Fraser University:

- Software Engineering Best Practices
- Software Project Management
- Professional Issues in Software Engineering
- Software Engineering Team Project

For Carleton University:

- Undergrad course on data structures, databases, programming, and computer architecture

Exhibit A: Context

Exhibit A

Context of IPP Software System (NARS)

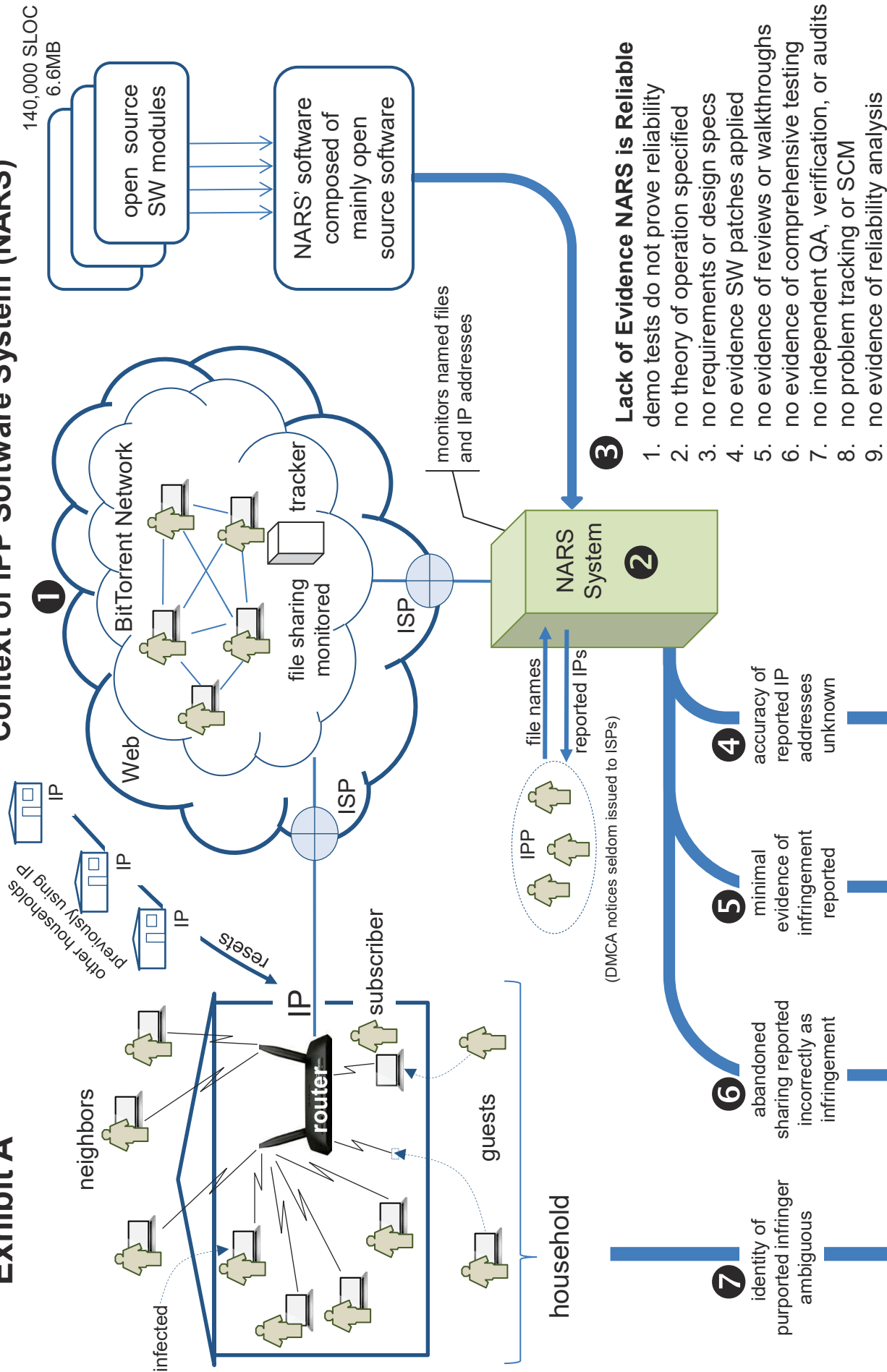


Exhibit B: Reliability Assessment

Exhibit B

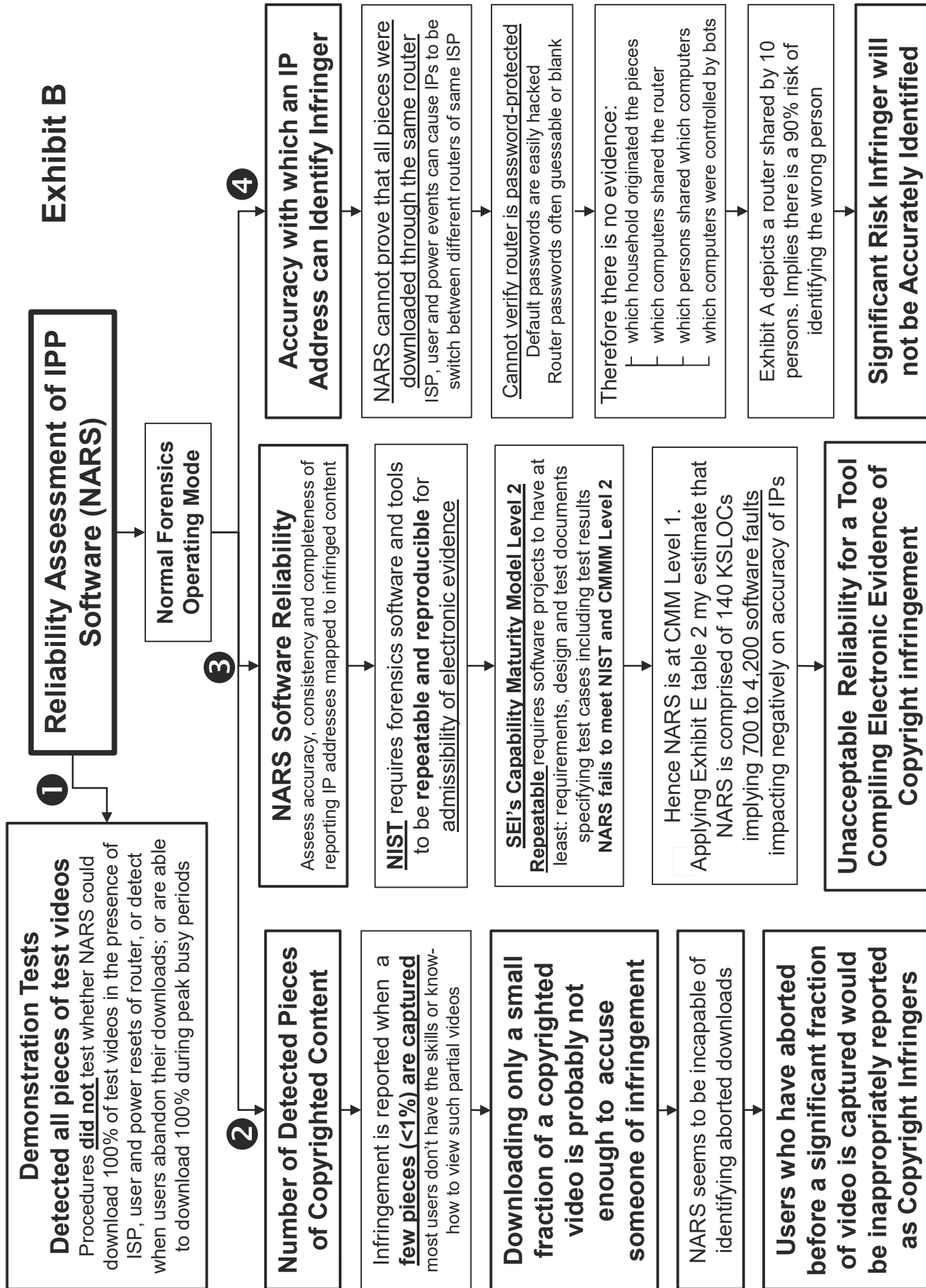


Exhibit C: BitStalker Article

BITSTALKER: ACCURATELY AND EFFICIENTLY MONITORING BITTORRENT TRAFFIC

Kevin Bauer, Damon McCoy, Dirk Grunwald, and Douglas Sicker

University of Colorado, Boulder, CO, USA
 {bauerk, mccoym, grunwald, sicker}@colorado.edu

ABSTRACT

BitTorrent is currently the most popular peer-to-peer network for file sharing. However, experience has shown that BitTorrent is often used to distribute copyright protected movie and music files illegally. Consequently, copyright enforcement agencies currently monitor BitTorrent swarms to identify users participating in the illegal distribution of copyright-protected files. These investigations rely on passive methods that are prone to a variety of errors, particularly false positive identification.

To mitigate the potential for false positive peer identification, we investigate the feasibility of using *active* methods to monitor extremely large BitTorrent swarms. We develop an active probing framework called *BitStalker* that identifies active peers and collects concrete forensic evidence that they were involved in sharing a particular file. We evaluate the effectiveness of this approach through a measurement study with real, large torrents consisting of over 186,000 peers. We find that the current investigative methods produce at least 11% false positives, while we show that false positives are rare with our active approach.

Index Terms—Data mining for forensic evidence

1. INTRODUCTION

While BitTorrent provides the ability to transfer files among many users quickly and efficiently, experience has shown that its decentralized architecture also makes it appealing for sharing copyright protected files illegally. With a peer-to-peer network like BitTorrent, content is distributed and replicated among a potentially large set of peers, making the process of finding and contacting each peer hosting the content in question a difficult task. Despite the challenge, entities acting on behalf of copyright holders have begun to monitor BitTorrent file transfers on a massive scale to identify and contact users who violate copyright laws.

In fact, a recent study [1] shows how the entities representing copyright holders use naïve techniques such as querying the BitTorrent tracker servers to identify individual users participating in an illegal file transfer. After being identified, these entities often distribute DMCA take-down notices or even pursue more formal legal sanctions against individuals who appear in the tracker’s peer list. However, this simple approach is prone to a wide variety of errors. For instance, it is trivial to introduce erroneous information into the tracker lists by explicitly registering fake hosts to the tracker. The authors of the recent study demonstrate this type of false positive identification by registering networked devices such as printers and wireless access points to tracker lists and subsequently receiving DMCA take-down notices for their suspected participation in illegal file transfers.

This strategy of polluting tracker lists with fake peers could be used to frustrate anti-piracy investigations. The

Pirate Bay, a popular tracker hosting site, has allegedly begun to inject arbitrary, but valid IP addresses into their tracker lists [2]. This counter-strategy may further increase the potential for false positive identification, which could have serious consequences as this evidence can be used to initiate legal action against suspected file sharers.

Given the inaccurate nature of the current techniques for monitoring BitTorrent file transfers and the clear need for effective anti-piracy tactics, we consider this question: Is it feasible to develop and deploy an efficient technique for identifying and monitoring peers engaged in file sharing that is more accurate than querying the trackers?

To answer this question, we propose a technique that is active, yet efficient. Starting with the tracker’s peer lists, each peer listed by the tracker server is actively probed to confirm their participation in the file sharing and to collect concrete forensic evidence. Our tool, called *BitStalker*, issues a series of lightweight probes that provide increasingly conclusive evidence for the peers’ active participation in the file sharing.

To evaluate the feasibility of this active approach in practice, we conduct a measurement study with real, large torrents. In particular, we quantify the number of peers that can be identified, the potential for falsely identifying peers, the potential for missing peers, and the cost associated with this technique in terms of bandwidth. Our results indicate that active probing can identify a sufficiently large portion of the active peers while requiring only 14.4–50.8 KB/s and about five minutes to monitor over 20,000 peers (using a commodity desktop machine). We also show that the active probing can be parallelized and scale to monitor millions of peers inexpensively using cloud computing resources such as Amazon’s Elastic Compute Cloud (EC2) [3]. Using EC2, we estimate that our method can monitor the entire Pirate Bay (about 20 million peers) for only \$12.40 (USD).

2. BACKGROUND

Before we describe our method for monitoring large BitTorrent swarms, we first provide a description of the BitTorrent protocol and an overview of the techniques currently being applied to identify peers who are sharing a file with BitTorrent.

2.1. The BitTorrent Protocol

To share a file, BitTorrent first breaks the file into several fixed size *pieces* and computes a SHA1 hash of each piece to verify integrity. Pieces are sub-divided into smaller data units called *blocks*, typically 16 KB in size. A metadata file containing the SHA1 hashes for each piece along with other information necessary to download the file including a URI to the *tracker server* is distributed to interested users via an out-of-band mechanism. Once a user has obtained the metadata for a file of interest, they proceed by contacting the tracker server to obtain a randomly chosen subset of peers who are sharing

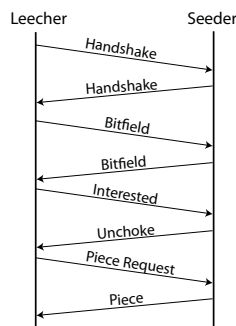


Fig. 1. BitTorrent message exchange to start a piece transfer

the file. This is called the *peer list*. By obtaining a peer list from the tracker (or another distributed hash table-based or gossip-based mechanism), the peer also registers itself with the tracker. The peer then begins requesting blocks of the file. Peers that are downloading pieces of the file are called “leechers,” while peers that possess all pieces and participate as uploaders are referred to as “seeders.”

The precise sequence of messages involved in the request of pieces is shown in Figure 1. A leecher establishes communication with another peer by exchanging handshake messages. The handshake consists of a plain text protocol identifier string, a SHA1 hash that identifies the file(s) being shared, and a peer identification field. After the handshake exchange, the leecher transmits a *bitfield* message. This contains a bit-string data structure that compactly describes the pieces that the peer has already obtained. After exchanging bitfields, the leecher knows which pieces the other peer can offer, and proceeds to request specific blocks of the file. The leecher sends an *interested* message to notify the other peer that it would like to download pieces. The other peer responds with an *unchoke* message only if it is willing to share pieces with the leecher. Upon receiving an unchoke message, the leecher asks for specific blocks of the file.

2.2. BitTorrent Monitoring Practices

While BitTorrent provides an efficient way to distribute data to a large group of users, it is also an appealing technique to distribute copyright protected files illegally. Copyright enforcement is particularly challenging within the context of BitTorrent, since the file(s) in question are distributed among a set of arbitrarily many peers. The copyright holders must first *identify* every user who appears to be sharing the file and ask them to stop sharing.

Despite the significant amount of work required to monitor BitTorrent networks, a recent study has gathered evidence showing that investigative entities acting on behalf of various copyright holders are monitoring and tracking BitTorrent users who are suspected of sharing copyright protected files [1]. These investigators — including BayTSP [4], Media Defender [5], and Safenet [6] who are hired by organizations such as the Motion Picture Association of America (MPAA) and the Recording Industry Association of America (RIAA) — are using *passive* techniques, such as querying the trackers for the peer lists to identify users who are engaged in illegal file sharing. Once a list of peers has been obtained, an ICMP echo (ping) message is sent to each IP address to ensure that it is alive.

However, as the aforementioned study notes, these methods for monitoring large BitTorrent networks can be wildly inaccurate. For instance, it is possible to implicate arbitrary

networked devices by simply registering their IP addresses with the tracker server. In addition, *false positive* identification is also possible as a result of naturally occurring (*i.e.*, non-intentional) activity. For instance, the tracker may provide stale peer information, which may result in a user who recently obtained a DHCP lease on an IP address being implicated in the file sharing. The very real potential for false positives could have serious implications, since the investigators who conduct this monitoring often issue DMCA take-down notices or even initiate legal actions against the suspected file sharers.

3. ACCURATE AND EFFICIENT MONITORING

In order to study the feasibility of collecting forensic evidence to concretely prove a peer’s participation in file sharing, we present *BitStalker*. BitStalker is active, yet efficient, since it consists of small probe messages intended to identify whether a peer is actively engaged in a file transfer. First, to obtain the list of peers who are potentially sharing the file, the tracker is queried. For each IP address and port number returned, we conduct a series of light-weight probes to determine more conclusively whether the peer really exists and is participating in the file transfer.

TCP connection. The first probe consists of an attempt to open a TCP connection to the IP address on the port number advertised by the tracker. A successful TCP connection indicates that the suspected peer is listening for connections on the correct port.

Handshake. If a TCP connection is established, a valid BitTorrent handshake message is sent. If the handshake succeeds, then the investigator has obtained evidence that the suspected peer is responding to the BitTorrent protocol, and may even provide information about the BitTorrent client software being used.

Bitfield. If the handshake probe succeeds, then a BitTorrent bitfield message is sent. This message contains a concise representation of all pieces that have been downloaded by the peer. A random bitfield is generated so that the probe looks like a valid bitfield message. If a peer responds with a valid bitfield message, then the investigator has obtained evidence that the peer has downloaded the part of the file that is described by their bitfield. This also indicates whether the peer is a seeder or a leecher. This provides the strongest form of forensic evidence that the peer is actively sharing the file without exchanging file data.

Block request. If the bitfield probe succeeds, we finally attempt to request a 16 KB block of the file from the peer. First, the peer’s bitfield is examined to find a piece of the file that the peer has obtained. Next, this probe sends an interested message to indicate that we want to exchange pieces with this peer. The peer responds with an unchoke message, which implies that we are allowed to ask for pieces. We finally request a 16 KB block. If the peer responds with the block requested, then this probe succeeds. A single block is the smallest amount of data necessary to confirm that another peer is sharing the file. If the investigator has the remaining blocks of that piece, then they can verify the hash to ensure that the block is valid.

We argue that each probe type provides increasingly conclusive evidence of a peer’s active involvement in file sharing. A successful TCP probe indicates that the peer is listening on the correct port. However, an effective counter-strategy could be to register arbitrary IP addresses with ports that are opened (such as web servers). The subsequent handshake probe is more conclusive, as it indicates that the BitTorrent protocol

Table 1. Summary of data sources

Torrent ID	Total Peers	Media Type
1	20,354	TV Series
2	16,979	TV Series
3	11,346	TV Series
4	14,691	TV Series
5	23,346	Movie
6	20,777	TV Series
7	24,745	TV Series
8	13,560	TV Series
9	19,694	TV Series
10	20,611	Movie
Total:	186,103	

is running on the correct port and also identifies the content being shared by a SHA1 hash. The bitfield probe provides stronger evidence still, since it describes all pieces that the peer has downloaded, which implies active sharing. Finally, requesting and subsequently receiving a block of the file provides the strongest form of concrete evidence for file sharing.

Practical considerations. The active probing framework can monitor peers who are actively participating in the file sharing. However, if a peer has just joined the torrent when they are probed, then they may not have any pieces of the file yet. Consequently, according to the BitTorrent protocol, if a peer has no pieces, then the bitfield probe is optional. Since the peer has not yet obtained any pieces of the file, the probing does not collect any evidence from this peer. If peers are probed repeatedly over time, then the likelihood of this case becomes negligible.

Additionally, “super-seeding” mode is enabled when a torrent is first established and there are few seeders. Super-seeding mode ensures that the original seeder is not overwhelmed by piece requests from other peers before it transfers data to another peer. When super-seeding is activated, the seeder may advertise an empty or modified bitfield, even though they possess every piece. Since we are interested in monitoring mature torrents consisting of at least tens of thousands of peers, we disregard new torrents in super-seeder mode.

Lastly, it is possible that peers may be able to detect the monitors and blacklist them. Siganos *et al.* show that the current passive BitTorrent monitors can be detected by observing that the frequency with which the monitor’s IP addresses occur across a large number of tracker lists is statistically higher than that of normal peers [7]. Our active monitoring may also be identifiable in the same manner. To address this, we recommend that the monitoring be distributed across a large number or dynamic set of IP addresses.

4. EXPERIMENTAL EVALUATION

In this section, we present experiments to quantify both the effectiveness and the cost of monitoring large BitTorrent swarms using the active probing technique. In addition, we compare the accuracy, potential for false positives and false negatives, and the cost with the current strategy employed widely by anti-piracy investigators.

4.1. Data Sources and Methodology

To evaluate our light-weight probing technique, we selected ten large torrents each containing between 11,346 and 24,745 unique peers. In total, our experimental evaluation consists of over 186,000 peers. Peers participating in these torrents were sharing new theatrical releases and episodes of popular television shows (summarized in Table 1). These swarms represent

the type of file sharing that may be monitored by copyright enforcement agencies.

To conduct the active probing, we wrote a tool called BitStalker that can perform the following tasks:

- Establish a TCP connection with another peer
- Exchange handshake messages with the correct SHA1 content hash and receive handshake responses
- Exchange bitfield messages and receive bitfield responses
- Request and receive a 16 KB block of file data

In short, BitStalker efficiently probes for participation in the BitTorrent protocol by sending and receiving a minimal number of small control messages rather than downloading the entire file from other peers.

The experiments were conducted as follows: The tracker server is contacted to obtain a subset of the peers who are currently believed to be sharing the file. Since the trackers only return a randomly selected set of 100 peers, it is necessary to query the tracker several times to obtain a large portion of the hosts registered with the tracker. Once peers are obtained from the tracker, BitStalker attempts to establish a TCP connection with each peer on its advertised TCP port. If a connection is established, a handshake message exchange is attempted. If handshake messages are exchanged, BitStalker attempts to exchange bitfield messages. Finally, if bitfields are exchanged, the tool attempts to retrieve a single block of the file. This procedure is repeated for each torrent to be monitored.

We compare our active probing method with the current approach to peer identification described in Section 2.2. After obtaining the list of suspected peers from the tracker, our tool sends precisely five ICMP echo (ping) messages to each IP address in the peer list. If a host responds to at least one ping, then it is assumed (perhaps erroneously) to be alive and sharing the file.

4.2. Experimental Results

We evaluate the proposed peer probing technique with regard to the number of peers that can be identified, an estimate of the number of peers that are falsely identified as being a file sharer (false positives), an estimate of the number of peers that this technique fails to identify (false negatives), and the measured cost of performing this active probing. The probing mechanism is compared along each of these metrics to the passive identification process using ping messages to verify the tracker’s peer list.

Fraction of peers that respond. We first consider how many peers can be identified by active probing. As shown in Table 2, the fraction of peers that can be positively identified by each probe type increases with additional repetitions. To determine if additional peers can be identified through multiple probing attempts, the experiments are repeated ten times. Even though the number of peers probed remains constant for each repetition, we find that the fraction of peers that respond to probes increases, since some peers may be busy interacting with other peers when we probe.

The complete results for each torrent are given in Figure 2. Across the ten torrents, we could establish a TCP connection with between 26.7–44.6% of the peers listed by the tracker. While this percentage seems low, it is reasonable since many BitTorrent clients impose artificial limits on the number of open connections allowed, in order to reduce the amount of bandwidth consumed. A similar fraction of peers that establish connections is reported by Dhungel *et al.* [8].

The naïve ping method returns roughly the same fraction of peers as the active TCP connection probe. However, as we

Table 2. The average fraction of peers identified in one, five, and ten iterations of the monitoring across all ten torrents

Repetitions	Connection	Handshake	Bitfield	Block Request
1	30.8%	18.9%	17.7%	0.29%
5	35.9%	26.3%	25.3%	0.80%
10	36.9%	28.4%	27.6%	1.13%

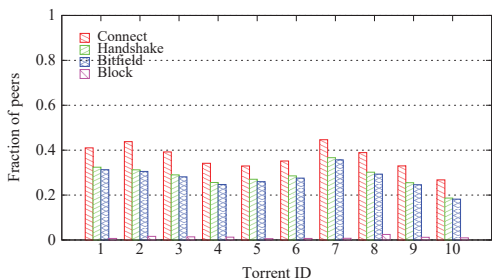


Fig. 2. Over ten runs, the cumulative fraction of peers identified with connections, handshakes, bitfields, and block requests across all ten torrents

will show, the ping probes are susceptible to an intolerably high number of false positives, while active probing significantly reduces the potential for false positives.

Both the handshake and bitfield probes succeed for between 18.6–36.6% of the peers. While this is lower than the TCP connection probe, it provides significantly stronger evidence for file sharing. For this fraction of the peers, an investigator can tell that the peer is obeying the BitTorrent protocol, sharing the correct file identified in the handshake probe by a SHA1 hash, and advertising the pieces of the file that the peer already possesses as identified in the bitfield probe. We argue that this small reduction in the fraction of peers that respond to bitfield probes is a small price for greater confidence in the identification results.

Finally, we observe that block request probes succeed for a very small fraction of the peers, only 0.6–2.4%. This may be partly a result of BitTorrent’s tit-for-tat incentive mechanism [9], which attempts to mitigate selfish leechers by enforcing reciprocity in the piece request process. This is implemented by uploading to other leechers from whom you download. The leecher with the highest upload rate receives download priority. Since BitStalker has a zero upload rate, it does not receive priority for piece requests. However, BitTorrent does offer optimistic unchoking, which enables a leecher to download regardless of their upload rate. BitStalker only receives pieces from other peers who have chosen to optimistically unchoke.¹ Since only about 1% of the peers respond to our block requests on average, we argue that the minimal additional evidence obtained through this probe is not worth the extra time and bandwidth required to collect this evidence.

False positives. The most serious flaw with the past and present investigative tactics based on tracker list queries and ping probes is the real potential for a high number of false positives. Furthermore, active peer list pollution further increases the potential for false positives.

To establish a lower bound on false positives obtained by the naïve investigative strategy, we count the number of peers that respond to pings yet show no indication of running any network service on their advertised port. More technically, if

¹Additional blocks may be received if BitStalker offered blocks before asking for blocks.

a peer responds to a TCP SYN request with a TCP RST (reset) packet, this indicates that the remote machine exists, but it is not running any service on the advertised TCP port. From our experiments, we observe that 11% of peers exhibit this behavior on average and are, therefore, definite false positives using this naïve investigative strategy.

In addition, we count the number of peers that *could* be false positives with the ping method. These are the peers that respond to ping probes, but ignore the TCP probe (*i.e.*, no connection or reset packet). From our experiments, we find that on average an additional 25.7% of the peers could potentially be false positives, but we cannot say this conclusively. It’s possible that some of these peers could have reached a connection limit in their BitTorrent client or could be filtering incoming traffic.

In contrast to the naïve ping method, the active probing strategy offers more reliable peer identification with few avenues for false positives. For instance, a successful TCP probe indicates that the peer is listening for connections on its advertised port. However, one could envision a more intelligent pollution strategy where arbitrary IP addresses with open ports are inserted into trackers (*i.e.*, real HTTP or FTP servers). The subsequent handshake and bitfield probes would then eliminate this form of pollution by checking that the host is running the BitTorrent protocol.

However, the active probing approach is not entirely immune from the possibility of false positive identification. For example, peers using an anonymizing network such as Tor [10] may produce false positives, since the last Tor router on the client’s path of Tor routers (called a Tor exit router) would be implicated in the file sharing. In fact, a recent study has found that BitTorrent is among the most common applications used with Tor [11].

To determine how common this type of false positive is in practice, we compare the list of potential BitTorrent peers obtained through our experiments to the list of all known Tor exit routers provided by Tor’s public directory servers. On average, we find that only approximately 1.8% of the peers are using Tor to hide their identities.² However, these are not false positives using active probing, since a peer using Tor (or another anonymizing network or proxy service) cannot bind to the advertised port on the exit host to accept incoming connections. Consequently, active probing does not provide any evidence for these peers. Furthermore, peers using Tor are easily identifiable and can be filtered out of the results.

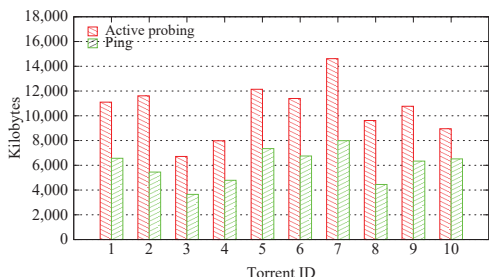
In addition to general-purpose anonymizing networks, solutions have been proposed specifically for anonymizing BitTorrent. For instance, SwarmScreen’s goal is to obscure a peer’s file sharing habits by participating in a set of random file sharing swarms [12]. Also, BitBlender attempts to provide plausible deniability for peers listed by the trackers by introducing relay peers that do not actively share files, but rather act as proxies for other peers actively sharing the file [13]. The active methods we propose would identify peers utilizing SwarmScreen and BitBlender as file sharers. While these peers are not intently sharing content, an investigator may still be interested in pursuing these peers since they contribute pieces of the file to other peers who are actively sharing.

False negatives. False negative identification occurs when a peer who is actively sharing a file cannot be identified as a file sharer. Both the active probing technique and the naïve ping method suffer from the potential for false negatives. The ping method may miss peers who are behind a firewall that blocks incoming ICMP traffic. For example, this is the default configuration for Windows Vista’s firewall settings. The active probing method may also suffer from false negatives when a

²However, several peers could be using each of these Tor exit nodes.

Table 3. Size of each probe type (assuming no TCP options)

Probe Type	Description	Size
TCP connection	Three-way handshake	162 Bytes
Handshake	Handshake request/reply	244 Bytes
Bitfield	Bitfield request/reply	Variable
Block Request	Block request/reply	16.688 KBytes
ICMP Ping	Ping request/reply	86 Bytes

**Fig. 3.** Total amount of traffic necessary to monitor each torrent using active probing and pings

peer’s number of allowed connections is at the maximum. In this case, the initial TCP connection probe will fail to identify that the peer is listening on its advertised port. In general, we found that repeating the monitoring procedure decreases false negatives. Table 2 shows that the number of false negatives decreases as the experiment is repeated. Although there are diminishing returns, as the false negatives do not decrease significantly between 5 and 10 iterations of the monitoring.

We can, however, provide a lower bound on false negatives obtained with the naïve ping method. This is achieved by counting the number of peers that do not respond to pings, but do respond to the TCP connection probe. Our experiments show that the naïve ping method would fail to identify at least 22.3% of the peers on average.

Cost. In order for an active probing strategy to be a feasible technique to monitor large BitTorrent swarms in practice, it is necessary for the probing to be as efficient as possible. Table 3 shows that the size of each probe is small and Figure 3 shows the amount of traffic that was required to monitor each torrent using the active probing technique. For comparison, the cost for the ping method is also plotted. While the ping approach requires less bandwidth, we have shown that it is not sufficiently accurate in identifying active file sharers. Using a modest Linux desktop machine, it took 304.5 seconds on average to monitor an entire torrent, which required only 14.4–50.8 KB/s of bandwidth. The active probing overhead is dependent on the fraction of peers that respond to active probes. This is an intuitive result, implying a direct relationship between the number of peers identified and the amount of bandwidth required by the probing.

The active probing method is also highly scalable, particularly when inexpensive cloud computing resources such as Amazon’s Elastic Compute Cloud (EC2) [3] are utilized. Machines from EC2 are available at a small cost dependent on the execution time and bandwidth usage of the jobs. From our experiments, on average we probed approximately 61 peers/second, uploaded 288.2 bytes/peer and downloaded 296.6 bytes/peer. Using EC2’s pricing model, we estimate that it is possible to monitor peers at an expected cost of roughly 13.6 cents/hour (USD). In fact, it’s possible to scale the active probing to monitor the entire Pirate Bay, which claims to track over 20 million peers [14]. We estimate that this method can monitor the Pirate Bay for \$12.40 (USD).

5. CONCLUSION

This paper presents *BitStalker*, a low-cost approach to monitoring large BitTorrent file sharing swarms. *BitStalker* collects concrete evidence of peers’ participation in file sharing in a way that is robust to tracker pollution, highly accurate, and efficient. In contrast, the past and present investigative monitoring strategy consists of tracker server queries and ICMP ping probes. While this method is simple, it is also prone to a variety of significant errors, especially false positive identification, since this monitoring technique does not verify participation in the file sharing. We present an alternative monitoring strategy based on actively probing the list of suspected peers to obtain *more conclusive* evidence of participation in the file sharing.

There are several aspects of our approach that warrant additional attention. In particular, a specific definition of what constitutes “evidence” in the context of file sharing across various legal systems should be explored. Also, the general legal issues that this type of monitoring exposes should also be investigated further.

Acknowledgments. The authors thank the anonymous reviewers for their valuable comments and suggestions. We also thank Claire Dunne and the University of Colorado’s institutional review board for ensuring that this research was conducted with the highest of ethical standards. This research was funded in part through gifts from PolyCipher.

6. REFERENCES

- [1] Michael Piatek, Tadayoshi Kohno, and Arvind Krishnamurthy, “Challenges and directions for monitoring P2P file sharing networks – or – Why my printer received a DMCA takedown notice,” in *3rd USENIX Workshop on Hot Topics in Security (HotSec)*, July 2008.
- [2] “Pirate bay tricks anti-pirates with fake peers,” <http://torrentfreak.com/the-pirate-bay-tricks-anti-pirates-with-fake-peers-081020>.
- [3] “Amazon elastic compute cloud (amazon ec2),” <http://aws.amazon.com/ec2>.
- [4] “BayTSP,” <http://www.baytsp.com>.
- [5] “Media defender – P2P anti-piracy and P2P marketing solutions,” <http://www.mediadefender.com>.
- [6] “Safenet Inc: The foundation for information security,” <http://www.safenet-inc.com>.
- [7] Georgios Siganos, Josep M. Pujol, and Pablo Rodriguez, “Monitoring the BitTorrent monitors: A bird’s eye view,” in *PAM*, 2009, pp. 175–184.
- [8] Prithula Dhungel, Di Wu, Brad Schonhorst, and Keith W. Ross, “A measurement study of attacks on bittorrent leechers,” in *International Workshop on Peer-to-Peer Systems (IPTPS)*, February 2008.
- [9] “BitTorrent protocol specification,” <http://wiki.theory.org/BitTorrentSpecification>.
- [10] Roger Dingledine, Nick Mathewson, and Paul Syverson, “Tor: The second-generation onion router,” in *Proceedings of the 13th USENIX Security Symposium*, August 2004.
- [11] Damon McCoy, Kevin Bauer, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker, “Shining light in dark places: Understanding the Tor network,” in *Proceedings of the 8th Privacy Enhancing Technologies Symposium*, July 2008.
- [12] David R. Choffnes, Jordi Duch, Dean Malmgren, Roger Guierma, Fabian E. Bustamante, and Luis Amaral, “SwarmScreen: Privacy through plausible deniability for P2P systems,” Northwestern EECS Technical Report, March 2009.
- [13] Kevin Bauer, Damon McCoy, Dirk Grunwald, and Douglas Sicker, “BitBlender: Light-weight anonymity for BitTorrent,” in *Proceedings of the Workshop on Applications of Private and Anonymous Communications (AIPACa 2008)*, Istanbul, Turkey, September 2008, ACM.
- [14] “The pirate bay,” <http://thepiratebay.org>.

Exhibit D: Validation of Forensics Tools and Software

Validation of Forensic Tools and Software: A Quick Guide for the Digital Forensic Examiner

Wed, 03/02/2011 - 7:45am by Josh Brunty

With the field of digital forensics growing at an almost warp-like speed, there are many issues out there that can disrupt and discredit even the most experienced forensic examiner. One of the issues that continues to be of utmost importance is the validation of the technology and software associated with performing a digital forensic examination. The science of digital forensics is founded on the

DEEPER INSIGHTS



The Importance of Mobile Forensics for Law Enforcement

Forensic

Tools and software for digital forensic analysis should be validated quarterly.

principles of repeatable processes and quality evidence. Knowing how to design and properly maintain a good

validation process is a key requirement for any digital forensic examiner. This article will attempt to outline the issues faced when drafting tool and software validations, the legal standards that should be followed when drafting validations, and a quick overview of what should be included in every validation.

Setting the Standard: Standards and Legal Baselines for Software/Tool Validation

According to the National Institute of Standards and Technology (NIST), test results must be *repeatable* and *reproducible* to be considered admissible as electronic evidence. Digital forensics test results are repeatable when the same results are obtained using the same methods in the same testing environment. Digital forensics test results are reproducible when the same test results are obtained using the same method in a different testing environment (different mobile phone, hard drive, and so on). NIST specifically defines these terms as follows:

Repeatability refers to obtaining the same results when using the same method on identical test items in the same laboratory by the same operator using the same equipment within short intervals of time.

Reproducibility refers to obtaining the same results being obtained when using the same method on identical test items in different laboratories with different operators utilizing different equipment.

In the legal community, the Daubert Standard can be used for guidance when drafting software/tool validations. The Daubert Standard allows novel tests to be admitted in court, as long as certain criteria are met. According to the ruling in *Daubert v. Merrell Dow Pharmaceuticals Inc.* the following criteria were identified to determine the reliability of a particular scientific technique:

1. Has the method in question undergone empirical testing?
2. Has the method been subjected to peer review?
3. Does the method have any known or potential error rate?

4. Do standards exist for the control of the technique's operation?
5. Has the method received general acceptance in the relevant scientific community?

The Daubert Standard requires an independent judicial assessment of the reliability of the scientific test or method. This reliability assessment, however, does not require, nor does it permit, explicit identification of a relevant scientific community and an express determination of a particular degree of acceptance within that community. Additionally, the Daubert Standard was quick to point out that the fact that a theory or technique has not been subjected to peer review or has not been published does not automatically render the tool/software inadmissible. The ruling recognizes that scientific principles must be flexible and must be the product of reliable principles and methods. Although the Daubert Standard was in no way directed toward digital forensics validations, the scientific baselines and methods it suggests are a good starting point for drafting validation reports that will hold up in a court of law and the digital forensics community.

The Scientific Method and Software/Tool Validations: A Perfect Fit

In the *Daubert* ruling, The Court defined scientific methodology as “the process of formulating hypotheses and then conducting experiments to prove or falsify the hypothesis.” The Scientific Method refers to a body of techniques for investigating phenomena, acquiring new knowledge, or correcting and integrating previous knowledge. To be termed scientific, the method must be based on gathering, observing, or investigating, and showing measurable and repeatable results. Most of the time, the scientific process starts with a simple question that leads to a hypothesis, which then leads to experimentation, and an ultimate conclusion. To exemplify, if you are validating a particular hardware write blocking device you may want to start with the simple question “Does this tool successfully allow normal write-block operation to occur to source media?” Since it is assumed that the write-blocking device supports various types of media (SATA, IDE, and so on) you

may be required to list the various requirements of the tool. Because if this, it is good practice for an examiner to use the scientific method as a baseline for formulating digital forensic validations. It is recommended that forensic examiners follow these four basic steps as a starting point for an internal validation program:

1) Develop the Plan

Developing the scope of the plan may involve background and defining what the software or tool should do in a detailed fashion. Developing the scope of the plan also involves creating a protocol for testing by outlining the steps, tools, and requirements of such tools to be used during the test. This may include evaluation of multiple test scenarios for the same software or tool. To illustrate, if validating a particular forensic software imaging tool, that tool could be tested to determine whether or not it successfully creates, hashes, and verifies a particular baseline image that has been previously setup. There are several publically available resources and guides that can be useful in establishing what a tool should do such as those available from NIST's Computer Forensic Tool Testing Project (CFTT) available from <http://www.cftt.nist.gov>. The CFTT also publishes detailed validation reports on various types of forensic hardware and software ranging from mobile phones to disk imaging tools. In addition to CFTT, Marshall University has published various software and tool validation reports that are publically available for download from <http://forensics.marshall.edu/Digital/Digital-Publications.html>. These detailed reports can be used to get a feel for how your own internal protocol should be drafted. The scope of the plan may also include items such as: tool version, testing manufacturer, and how often the tests will be done. These factors should be established based on your organization standards. Typically, technology within a lab setting is re-validated quarterly or biannually at the very least.

2) Develop a Controlled Data Set

This area may be the longest and most difficult part of the validation process as it is the most involved. This is because it involves setting-up specific devices and baseline images and then adding data to the specific areas of the media or device. Acquisitions would then need to be performed and

documented after each addition to validate the primary baseline. This baseline may include a dummy mobile phone, USB thumb drive, or hard drive depending on the software or hardware tool you are testing. In addition to building your own baseline images, Brian Carrier has posted several publically available disk images designed to test specific tool capabilities, such as the ability to recover deleted files, find keywords, and process images. These data sets are documented and are available at <http://dfft.sourceforge.net>. Once baseline images are created, tested, and validated it is a good idea to document what is contained within these images. This will not only assist in future validations, but may also be handy for internal competency and proficiency examinations for digital examiners.

3) Conduct the Tests in a Controlled Environment

Outside all the recommendations and standards set forth by NIST and the legal community, it only makes sense that a digital forensics examiner would perform an internal validation of the software and tools being used in the laboratory. In some cases these validations are arbitrary and can occur either in a controlled or uncontrolled environment. Since examiners are continuously bearing enormous caseloads and work responsibilities, consistent and proper validations sometimes fall through the cracks and are validated in a somewhat uncontrolled "on-the-fly" manner. It's also a common practice in digital forensics for examiners to "borrow" validations from other laboratories and fail to validate their own software and tools. Be very careful with letting this happen. Keep in mind that in order for digital forensics to be practicing true scientific principles, the processes used must be proven to be repeatable and reproducible. In order for this to occur, the validation should occur within a controlled environment within your laboratory with the tools that you will be using. If the examiner uses a process, software, or even a tool that is haphazard or too varied from one examination to the next, the science then becomes more of an arbitrary art. Simply put, validations not only protect the integrity of the evidence, they may also protect your credibility. As stated previously, using a repeatable, consistent, scientific method in drafting these validations is always recommended.

4) Validate the Test Results against Known and Expected

Results

At this point, testing is conducted against the requirements set forth for the software or tool in the previous steps. Keep in mind that results generated through the experimentation and validation stage must be repeatable. Validation should go beyond a simple surface scan when it comes to the use of those technologies in a scientific process. With that said, it is recommended that each requirement be tested at least three times. If there are any variables that may affect the outcome of the validation (e.g. failure to write-block, software bugs) they should be determined after three test runs. There may be cases, however, where more or fewer test runs may be required to generate valid results.

It's also important to realize that you are probably not the first to use and validate a particular software or tool, so chances are that if you are experiencing inconsistent results, the community may be experiencing the same results as well. Utilizing peer review may be a valuable asset when performing these validations. Organizations such as the High Technology Crime Investigation Association (HTCIA) and the International Association of Computer Investigative Specialists (IACIS) maintain active member e-mail lists for members that can be leveraged for peer review. There are also various lists and message boards pertaining to mobile phone forensics that can be quite helpful when validating a new mobile technology. In addition, most forensic software vendors maintain message boards for software, which can be used to research bugs or inconsistencies arising during validation testing.

Conclusion

Real world laboratory use, controlled internal tests utilizing scientific principles, and peer review should all be leveraged in a validation test plan. Sharing unique results with the digital forensics community at-large helps investigators, examiners, and even software and tool vendors ensure that current best practices are followed. As the field of digital forensics continues to grow and evolve as a science the importance of proper scientific validation will be more important than ever.

References

1. Brown, C. "Computer Evidence: Collection & Preservation." Hingham: Thomson/Delmar. 2006.
2. Carrier, B. "Digital Forensics Tool Testing Images." Accessed 06 Feb 2011. <http://dfft.sourceforge.net/>.
3. *Daubert v. Merrell Dow Pharmaceuticals*, 509 U.S. 579 (1993).
4. High Technology Crime Investigation Association, Accessed 06 Feb 2011. www.htcia.org.
5. International Association of Computer Investigative Specialists, Accessed 06 Feb 2011. www.iacis.org.
6. Maras, MH. *Computer Forensics: Cybercriminals, Laws, and Evidence*. Sudbury: Jones & Bartlett. 2011.
7. Marshall University Forensic Science Center-Digital Publications, Accessed 06 Feb 2011. <http://forensics.marshall.edu/Digital/Digital-Publications.html>.
8. NIST Computer Forensic Tool Testing Project. Accessed 06 Feb 2011. www.cftt.nist.gov.
9. Shroader, A. "How to Validate Your Forensic Tools." Orem: Paraben Corp. 2010.

Josh Brunty currently manages the digital forensics graduate program and the digital forensics research and casework laboratories at the Marshall University Forensic Science Center in Huntington, WV. Josh holds numerous certifications within the digital forensics discipline including: AccessData Certified Examiner (ACE), Computer Hacking Forensic Examiner (CHFI), Seized Computer Evidence Recovery Specialist (SCERS), and is certified in Information Assessment Methodology (NSA-IAM). He has developed a variety of digital forensics training curriculum; including past recertification scenarios/exams for the International Association of Computer Investigative Specialists (IACIS). Josh is an active member of the Mid-Atlantic Association of the High Technology Crime Investigation Association (HTCIA) and the Digital-Multimedia Sciences section of the American Academy of Forensic Sciences (AAFS). He can be reached at josh.brunty@marshall.edu.

Exhibit E: Software Reliability Tutorial

(extracted pages only)

Software Reliability

Lou Gullo

Jon Peterson

Raytheon Company



Raytheon

Customer Success Is Our Mission

Software Failures

- Specification errors
 - Typically the largest source of failure
 - Ambiguous requirements
- Errors in design of code
 - Incorrect interpretation of the spec
 - Incomplete interpretation of the spec
 - Incorrect logic in the interpretation of the spec
 - Timing errors and race conditions
 - Shared data variables



Capability Maturity Model SEI Levels

Level	Description of Organization
Level 1: Initial	Organizations lack effective project management; do not maintain a solid, stable environment ...
Level 2: Repeatable	Organizations maintain policies and procedures for managing and developing ... Project planning based upon experience ...
Level 3: Defined	Organizations have developed and documented a standard process for managing and developing software systems....
Level 4: Managed	Organizations set quantitative goals ... use measurement instruments to collect process and product metrics.
Level 5: Optimizing	Organizations focus on continuous process improvement...



Capability Maturity Model Fault Density at Delivery Studies

CMM LEVEL	FAULTS/KSLOC (Keene Data)	FAULTS/KSLOC (Caper Jones)	FAULTS/KSLOC (Herb Krasner)	Defect Plateau Level
V	0.5	0.5	0.5	1.5%
IV	1.0	1.4	2.5	3.0%
III	2.0	2.69	3.5	5.0%
II	3.0	4.36	6.0	7.0%
I	5.0	7.44	30	10.0%

- Fault density is in defects per thousand lines of code (KSLOC).
- Data represents average expected results gathered from several SEI rated companies.

