Forensic Investigation of Peer-to-Peer File Sharing Network

By

Robert Erdely, Thomas Kerle, Brian Levine, Marc Liberatore and Clay Shields

From the proceedings of

The Digital Forensic Research Conference

DFRWS 2010 USA

Portland, OR (Aug 2nd - 4th)

DIGITAL INVESTIGATION 7 (2010) S95–S103

available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/diin

Digital Investigation

Forensic investigation of peer-to-peer file sharing networks

Marc Liberatore [a,*], *Robert Erdely* [c], *Thomas Kerle* [b], *Brian Neil Levine* [a], *Clay Shields* [d]

[a] *Dept. of Computer Science, Univ. of Massachusetts Amherst, Computer Science Building, 140 Governors Drive, Amherst, MA 01003-9264, USA*

[b] *Massachusetts State Police, USA*

[c] *Pennsylvania State Police, USA*

[d] *Dept. of Computer Science, Georgetown Univ., USA*

ABSTRACT

The investigation of peer-to-peer (p2p) file sharing networks is now of critical interest to law enforcement. P2P networks are extensively used for sharing and distribution of contraband. We detail the functionality of two p2p protocols, Gnutella and BitTorrent, and describe the legal issues pertaining to investigating such networks. We present an analysis of the protocols focused on the items of particular interest to investigators, such as the value of evidence given its provenance on the network. We also report our development of RoundUp, a tool for Gnutella investigations that follows the principles and techniques we detail for networking investigations. RoundUp has experienced rapid acceptance and deployment: it is currently used by 52 Internet Crimes Against Children (ICAC) Task Forces, who each share data from investigations in a central database. Using RoundUp, since October 2009, over 300,000 unique installations of Gnutella have been observed by law enforcement sharing known contraband in the the U.S. Using leads and evidence from RoundUp, a total of 558 search warrants have been issued and executed during that time.

1. Introduction

Where goes the data, so go the investigators. The strong impact of computing on everyday life — and criminal life — has increased the need for tools that can investigate computers and their data. This fact is particularly relevant to the Internet, where the ease and prevalence of data transfer notably facilitates certain types of illegal activity by its users. In this paper, we focus on criminal investigations of the trafficking of digital contraband on peer-to-peer (p2p) file sharing networks. P2P systems have become the standard instrumentality for the sharing and distribution of images of child sexual exploitation.

First, we examine the technical and legal issues inherent in forensic investigations of p2p systems. Shoddy investigative techniques lead to bad evidence, as ably demonstrated in a recent paper by Piatek et al. (2008). Such mistakes can be costly in terms of resources and erosion of the public trust, particularly in the context of criminal rather than civil law. These mistakes are a product of insufficient understanding of the information being provided by the underlying p2p system.

In order to prevent such mistakes, investigators need to understand p2p systems at a level sufficient to relate the technical and legal issues of investigating the system correctly. Our goal is to enable accurate online investigations of such systems, where investigators: (i) can confidently state from where and how various forms of evidence were acquired; (ii) can understand the relative strength of that evidence; and (iii) can validate that evidence from the fruits of a search warrant. To accomplish this goal, we describe and analyze the functionality of two p2p file sharing systems, Gnutella and BitTorrent, as they pertain to digital

* *Corresponding author.*

E-mail addresses: liberato@cs.umass.edu (M. Liberatore), rerdely@state.pa.us (R. Erdely), thomas.kerle@pol.state.ma.us (T. Kerle), brian@cs.umass.edu (B.N. Levine), clay@cs.georgetown.edu (C. Shields).

1742-2876/$ – see front matter
doi:10.1016/j.diin.2010.05.012

investigations. We provide a forensic analysis of the network protocols of these systems.

Second, we present *RoundUp*, a tool we developed to facilitate investigation of the Gnutella p2p system and in use by law enforcement RoundUp enables users to perform forensically sound investigations of Gnutella following the principles and techniques we detail here. RoundUp users can perform investigations in both a localized and a loosely coordinated fashion, using a centralized database in the latter case. We show that RoundUp has been quickly and widely adopted by law enforcement and that it is effective in generating leads and evidence. Specifically, over 40 agencies share data from RoundUp investigations in a central database. Since October 2009, over 300,000 unique installations of Gnutella have been observed sharing known child pornography in the US; this represents an upper bound on the number of users seen. Using leads and evidence from RoundUp, at least 558 search warrants have been issued and executed.

2. Background

When investigating a p2p system,[1] an investigator must be cognizant of the related legal and technical issues. In this section, we provide a technical overview of two p2p systems, Gnutella and BitTorrent. In a later section, we detail and analyze the specific functionality and mechanisms of these protocols as they relate to digital investigations.

2.1. *Overview of P2P file sharing systems*

P2P file sharing systems allow users to download and upload files from other users, referred to as *peers*, on the Internet, typically from within an *application* running on their local computer that follows a particular *protocol*. By p2p *network* we mean a set of Internet peers communicating and sharing files via a specific protocol. Particular p2p applications may support multiple protocols and thus multiple p2p networks. Table 1 summarizes common p2p protocols and applications.

The primary goal of every p2p file sharing system is to support efficient distribution of content shared among peers. Many p2p systems also directly support content searches by peers, and some allow a *direct browsing* of the files that a remote peer makes available.

2.1.1. *Gnutella*

Gnutella is a completely decentralized protocol for p2p file sharing. Peers bootstrap the process of joining the network by first contacting a known Web server that provides a partial list of current peers (called a *GWebCache*), or by using a list of known peers distributed with the Gnutella application. The joining host creates TCP connections to some of the peers on the list, becoming their *neighbor* on the network. Additional peers can be learned from these first neighbors. Hence, the peer topology is unstructured and fairly random. Peers are uniquely identified by a self-assigned, randomly chosen 16-byte ID, called a *globally unique ID* (GUID). The GUID is consistent across changes to the computer's IP address, but it can be changed at will by the user.

Table 1 – A sample of common p2p file sharing protocols and applications. Note that a protocol can be supported by several applications, and an application can support several protocols.

Protocol	Applications	Search Support	Browse Support
Gnutella (Klingberg and Manfredi, 2002)	LimeWire Shareaza BearShare Phex	Yes	Yes
BitTorrent (Cohen, 2009)	BitTorrent Vuze LimeWire Shareaza	No	No
eDonkey (Kulbak and Bickson, 2005)	eMule Shareaza	Yes	No

Users search for shared files by issuing queries to neighbors. Queries broadcast on the Gnutella network are text strings, and remote peers match the text in these strings to file names. Any peer that has content that matches the query's text replies with a response that is usually routed back along the path the query traveled along. Remote peers respond with their IP address and port, GUID, and information about matching files, including names, sizes, and hash values.

According to the Gnutella specification, queries can also be for a specific hash value, however, this feature is deliberately not fully supported in many clients. For example, versions of the Phex client support relaying and answering queries of specific hash values, but recent versions of Limewire will drop such queries.

The querying peer downloads content by selecting a file from received query responses. Files are identified by their hash, and they are downloaded through a direct TCP connection with remote peers known to possess that file. Separate portions of a file may be downloaded from distinct peers in parallel. If a remote peer is behind a firewall, a *push* message is used to request a connection from that remote peer to the originator. Push messages are relayed through intermediaries in the Gnutella network to initiate the connection. If both peers are behind a firewall, the push connection is not possible. In either case, the IP address and GUID of the remote peer is easy to record. Additionally, during the file transfer, the remote peer may relay to the requester the IP addresses and ports of other peers known to have the file. Peers may also directly connect to a remote peer to *browse* it; the remote peer replies with a list of query responses describing all files it shares, including SHA-1 values of each.

In the above description, we have elided some details; in particular, Gnutella limits the number of messages on the network by a division of labor. A subset of peers that form the network as described above, are known as *ultrapeers*. Ultrapeers are responsible for query and query response message routing, and typically connect to many other ultrapeers. Other peers, known as *leaves*, connect to five or fewer ultrapeers, and rely on these ultrapeers to relay these messages for them. Whether to an ultrapeer or leaf, a download or browse is always a direct TCP connection.

For a more complete description of the Gnutella protocol, the RFC (Klingberg and Manfredi, 2002) provides a fixed

[1] Throughout this paper, our use of the term "p2p systems" refers to only p2p file sharing systems.

starting point. A more up-to-date but changing reference is maintained by the Gnutella Developers Forum.[2]

2.1.2. *BitTorrent*

BitTorrent is a protocol for p2p file sharing, but unlike Gnutella, it requires ancillary support to search for files and to find peers with those files. Users start by locating a *torrent* file describing content they wish to download. Any user may create a torrent; each torrent describes a set of files that can be obtained through the BitTorrent protocol, and provides enough information to enable this process. At a minimum, this information includes file names, sizes, and SHA-1 hash values for power-of-two-sized *pieces* of the concatenated file set (see Fig. 1), as well as the URLs of one or more *trackers*. Some torrents contain additional optional information such as per-file hashes. If the per-file hashes are omitted, it is less straightforward to determine if content is known contraband as the pieces will not align with complete files. To overcome this problem, investigators can determine the hash values of the corresponding piece-wise subset of each file, but this must be performed after the torrent is observed. Torrents usually also contain an extensive comment field. Along with file names described by the torrent, this comment field is typically used by web-based torrent aggregation and search sites such as isohunt.com and thepiratebay.org to allow users to quickly find torrents of interest by using a simple text search.

To find peers sharing files described by a specific torrent, a peer next queries one of the trackers listed in the torrent file. The tracker identifies whether it manages a matching torrent by its *infohash*, which is the SHA-1 hash of fixed fields within the torrent that identify the files being distributed — the file names, sizes, and piece sizes and hashes. The peer's request to the tracker includes this infohash as well as the peer's ID (a 20-byte GUID that includes encoded application and version information), IP and port, and information about how much of the file the peer has already downloaded. The tracker responds with a list of peers claiming recent interest in this torrent, created by keeping track of previous queries and peers. These peers are described by at least their IP address and port, and optionally by their peer ID as well. Peers contact trackers periodically to update each other's list of peers and to keep trackers informed of their download progress and sustained interest.

To download files, a peer either directly connects to a remote peer at an IP address and port provided by the tracker, or it is correspondingly contacted by a remote peer. The BitTorrent protocol makes no distinction between inbound and outbound connections — it is assumed the goal of all peers interested in a torrent is to upload and download as much of the file as possible to and from any interested peers. The peers exchange a list of the pieces that they possess, and then request pieces from one another. Periodically, the peers may update one another when they come into the possession of new pieces from other peers.

Because bandwidth is a limited resource, the BitTorrent protocol has a mechanism, known as *tit-for-tat*, to encourage peers to upload as well as download. A peer keeps track of how much data a remote peer has provided to it. If this amount is below some threshold, the peer *chokes* the remote peer. Choked peers do not get uploaded to, unless there is available bandwidth after serving all *unchoked* peers. Occasionally, a peer will *optimistically unchoke* choked peers — this serves to bootstrap new peers into the network and to prevent the degenerate case of all peers choking each other.

Various extensions for BitTorrent exist. Of particular interest is a mechanism that eliminates the need for a tracker. Known as the *Distributed Hash Table* (DHT), this mechanism spreads the responsibility for handing tracking among all running BitTorrent peers that support the DHT. The responsibility for tracking each torrent is allocated to a subset of these peers. Additional mechanisms for *peer exchange* allow peers to share their lists of other peers among themselves without contacting a tracker.

More complete descriptions of the torrent file format, the tracker protocol, and the peer protocol can be found at wiki.theory.org. The official specification is located at bittorrent.org, which also distributes proposals describing the DHT protocol and other draft but widely implemented extensions to the BitTorrent protocols. The Vuze wiki (wiki.vuze.com) also contains references to Vuze (formerly Azureus) extensions to the BitTorrent protocol, notable due to the wide use of the Vuze client.

3. Legal issues in P2P investigations

There are many motivations to perform investigations of p2p file sharing networks. Our work is motivated by the presence and trafficking of images of child sexual exploitation — colloquially referred to as "child pornography" (CP) — on these networks (Mitchell et al., 2009; Wolak et al., 2009). Knowing possession or distribution of contraband is a felony offense in most U.S. states, and our focus is on such criminal investigation. Past studies have found that 21% of CP possessors had images depicting sexual violence to children such as bondage, rape, and torture; 28% had images of children younger than 3 years old; and most notably that 16% of investigations of CP possession ended with discovery of persons who directly victimized children (Wolak et al., 2005). In fact, a primary goal of these investigations is to catch child molesters and help children that are being victimized (often by family members), rather than to simply confiscate these images.

In this section, we survey only the legal issues relating to criminal possession of contraband. In particular, we do not address civil infractions due to p2p file sharing, including copyright infringement, which has a different set of relevant case law, evidentiary standards, and investigative goals.

3.1. *Investigative process*

An investigator's end goal is to obtain evidence through observation of data from the Internet. Whenever an investigator collects such evidence, it is of one of two varieties: *direct* or *hearsay*. When an investigator has a direct connection, that is, a TCP connection to a process on a remote computer, and receives information about that specific computer, that information is direct. For example, when using HTTP to transfer files, the file that is sent from the remote machine's web server is

[2] http://wiki.limewire.org/index.php?title=GDF

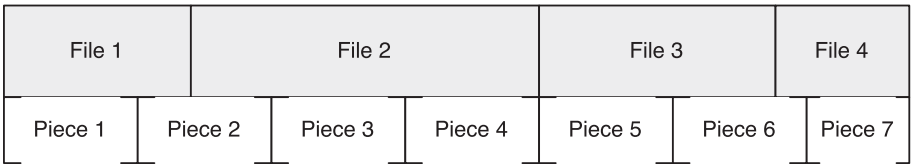

Fig. 1 – File and piece boundaries may not align on a torrent; only the first file is guaranteed to start at a piece boundary. This potential misalignment complicates the use of centralized registries of hashes of known contraband.

direct tied to that remote machine. Hearsay is when a process on one remote machine relays information for or about another, different machine. For example, a peer in a p2p system may claim another peer possesses a specific file. Depending upon the purpose, hearsay may be less useful than direct evidence.

In a typical investigation, the investigator performs the following steps:

1. One or more *files of interest* (FOIs) are identified. FOIs may be actual contraband (that is, CP), or may consist of material that is indicative of a sexual interest in children (e.g., textual stories). These files are acquired through Internet searches, p2p downloads, or from seized media. FOIs are uniquely identified by hash values; investigators need only have access to these hash values to identify FOIs.[3]
2. The p2p system is used to locate a set of *candidates*: IP addresses corresponding to potential possessors and distributors of FOIs. The early stages of most p2p investigations typically need not be covered under a search warrant, and are analogous to a police officer "walking their beat" watching for signs of criminal activity. Thus, only information that is accessible publicly (in "plain view"), such as through keyword searches conforming to p2p protocols, is collected. Since the main goal of p2p file sharing systems is broad dissemination of files, investigators usually need only connect to the system as a user to obtain information on candidates. The controlling case law in this area suggests that law enforcement officers are legally present (as are millions of other users) and that evidence collected is in "plain view". See, for example, *United States v. Borowy*, 595 F.3d 1045 (9th Cir. 2010). At this point, the investigator must understand the types of information being collected. Both hearsay and direct evidence is being collected. At this stage, the investigator is collecting leads, so both are valid.
3. Of these candidates, some subset is chosen for further investigation. This decision may be influenced by factors such as investigator's jurisdiction, the type and quantity of files of interest possessed by the candidate, and the observed history of the candidate.
4. The investigator then will attempt to verify a candidate's possession or distribution of contraband. From here on, when practical, the investigator will rely on direct communications, such as browsing and downloads, to build the case and gather evidence for legal processes including charging and search warrants. Hearsay evidence should be used as a last resort, and if used, should include evidence over a period of time and from different sources. Ideally, the investigator connects directly to the candidate, and notes the files that the candidate freely claims to have possession of. In some cases, the investigator may download the entire file from only the candidate, and not other peers (called a *single-source download*), as stronger evidence of possession and perhaps evidence of distribution.
5. As part of the previous two steps, each candidate's IP address and other p2p-level identifying information is logged. Candidates located behind a NAT device have both an internal and external IP address; the latter may be transmitted through the p2p protocol. Most p2p client assign a unique identifier to each installation — while these Globally Unique IDentifiers (GUIDs) exist to aid routing in the p2p network, they are also strong evidence. Any other potential corroborating evidence, such as application version information, is also collected.
6. On the basis of this information, a subpoena to the ISP associated with the candidate's IP address(es) is obtained, to determine a person a responsible and a location associated with the observed behavior. The exact information the subpoena yields will vary based upon the ISPs record-keeping policies.
7. On the basis of the evidence of contraband and the subpoenaed information, a search warrant is issued in search of the computer and contraband associated with the investigation. IP addresses corresponding to mobile devices may introduce additional difficulties in ascertaining the physical location of the device and materials to be searched for.
8. At this point, an investigator has a warrant for a location, but the computers and individuals involved in the crime are typically unknown at this point. A search is performed, and if relevant evidence is obtained, it may be used as the basis for an arrest and further legal action provided it can be linked to an individual. Investigators will locate the computers used by verifying the link between observed p2p behavior and the discovered evidence. Usually this process includes examining media for known contraband and correlating GUIDs of p2p clients installed on local machines with GUIDs observed during a p2p investigation. Once the computer and account is identified the link to the responsible person can be made.

[3] In the U.S., the National Center for Missing and Exploited Children maintains a registry of known, verified CP and attempts to correlate victims with images, as well as to identify new victims. In general, law enforcement personnel are strongly recommended to verify the contents of any suspected FOI.

3.2. *Legal constraints and issues*

At each step in an investigation, the investigator's behavior is bound by law. First and foremost, the investigator will be

liable for lawbreaking of their own. Additionally, gathering evidence illegally will likely result in this evidence being inadmissible in court under the fruit of the poisonous tree doctrine, although some states do have a good faith exception. As a result, the investigator must be aware of the specifics of the protocol used by the p2p system under investigation, and must understand how their tools interact with the system. Below, we highlight several of the constraints and potential pitfalls inherent in the investigation of p2p systems. Both investigator and the designer of any tools that the investigator uses should be aware of all of these issues. Ferraro and Casey (2005) provide a more in-depth analysis of many of these issues.

3.2.1. *Searches*

The fourth amendment to the U.S. Constitution reads:

> The right of the people to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures, shall not be violated, and no Warrants shall issue, but upon probable cause, supported by Oath or affirmation, and particularly describing the place to be searched, and the persons or things to be seized.

Law enforcement personnel are thus bound by this principle. P2P network investigations are enabled by the promiscuous nature of the protocols themselves: By freely advertising content, responding to search queries, and handling download requests, these p2p systems are in essence acting in public, rather than under the protection of the fourth amendment. Thus, no warrant is required for issuing keyword queries or download requests to a peer.

3.2.2. *Encryption*

P2P systems may support end-to-end encryption between peers. This feature was not developed to deter investigations, but instead to stop ISPs from throttling p2p traffic. Notably, an investigator running their own p2p client will not be impacted by this encryption — it presents an obstacle only to a third-party packet sniffer (e.g., one operated by the ISP). If encryption is used within the protocol, the key is negotiated between the investigator and the peer under investigation, again precluding any requirement for a warrant. In current p2p implementations, an *anonymous* Diffie-Hellman key exchange takes place, meaning that the keys are generated as needed and used only once; no public-key infrastructure is leveraged.

3.2.3. *Technology*

Kyllo v. United States 533 U.S. 27 (2001) is a U.S. Supreme Court ruling regarding the use of technology in performing surveillance or searches. Roughly, the outcome of *Kyllo* is that the Government is not permitted to conduct searches "using devices not in general public use to explore details of the home that would previously have been unknowable without physical intrusion". *Kyllo* is generally interpreted by investigators and tool builders to mean staying within the bounds of a protocol's specification and using only information provided by the protocol when performing investigations.

3.2.4. *Uploads and downloads*

Distributing contraband is illegal — but most p2p applications default to allowing uploads. Some protocols go further: BitTorrent applications may punish non-uploaders by limiting their download bandwidth. Attempts to circumvent these punishments by uploading junk data are detected by the use of hash trees. Regardless, investigators must not allow their tools to perform uploads of contraband.

P2P systems attempt to perform downloads from many peers simultaneously. When an investigator is attempting a single-source download, multi-peer downloads must be disabled.

3.2.5. *Record keeping*

Investigators must keep careful track of all relevant information recovered during their work. The provenance of evidence is critical when obtaining subpoenas and warrants, and when entering evidence into a criminal proceeding. Times and dates, methods, search terms, hash values, IP addresses, and GUIDs are among the data that are typically required. Good tools will record all of these items, and make clear the distinction between direct and hearsay evidence.

3.2.6. *Validation*

When a search warrant is executed, the investigator should link their observations through the p2p network to evidence obtained under the warrant. In p2p investigations, this means performing an onsite triage-style investigation of seized machines and media, or a more thorough forensic investigation in a lab. The goal is to find the presence of previously observed p2p identifiers, such as GUIDs and contraband, on the media.

The range of evidence that can be legally searched for is dependent upon the language in the search warrant. Warrants are usually written to search a premises for any collections of child pornography (thus searching all digital media) or evidence of intention to possess or distribute contraband. The latter can include stored keyword searches, carefully organized and sorted collections, backups of contraband to fixed media, and so on (Howard, 2004). In other words, there does not have to be a strict link: the evidence gathered from the online investigation can serve as probable cause justifying a search warrant only. If a different GUID and different contraband is found when the warrant is executed, the owner of the content would still be charged with a crime.

4. Protocol analysis

In this section, we present an analysis of the protocols of two popular p2p file sharing protocols, Gnutella and BitTorrent. We pay particular attention to forensically relevant data that allow investigators to meet legal standards for subpoenas, search warrants, and prosecution. We discuss techniques for validation of evidence obtained through these protocols, and also describe the ways in which investigation may fail.

The most critical aspects of any network investigation is knowing the *provenance* of evidence gathered over the network. The investigator must be aware of the source of the

evidence, whether hearsay or directly observed. Because *intent* is a requirement of CP possession, the context of the evidence is also critical. A single contraband file among hundreds of non-contraband may not be sufficient to show intentional possession or distribution, while a carefully organized collection tells a different story (Howard, 2004). Similarly, repeated observations of a growing collection over long periods of time inform an investigator's view of a candidate and speak to *mens rea*.

The end goal of an investigation of a p2p system varies, as does the level of evidence required. The investigator may only be collecting *leads*, which could consist of hearsay and will require further corroboration to have evidentiary value. In some cases, the goal is a search warrant, with the aim of using the fruits of the search warrant as evidence in a criminal trial for possession of contraband. For such a search warrant, *probable cause* is sufficient; evidence consisting of one or more instances of a remote peer claiming to have possession of a known piece of contraband is typically accepted by magistrates.

The main distinction between a lead and evidence is that evidence is directly observed, rather than found through hearsay. For stronger evidence, or to show distribution, an investigator may further attempt a *single source download* (SSD), where the entire file is retrieved from only the peer under investigation. A SSD is sufficient to show both possession and distribution, particularly in concert with the fruits of a search warrant. Obtaining an SSD successfully can be challenging, as discussed below.

4.1. Gnutella

There are four primary avenues for gathering evidence using the Gnutella protocol: queries, swarming information, browse hosts, and file downloads.

4.1.1. Queries

Queries based on terms associated with CP can quickly discover leads for investigator. The *hits* that are returned to the investigator contain the IP address, GUID, and names and SHA-1 values of remote files. One of Gnutella's main design goals is ensuring that content is successfully found. Gnutella's random topology construction and progressively wider-ranging flooding of queries attempts to provide a large number of results while minimizing network traffic. From a forensics perspective, such evidence is not at the level of probable cause for several reasons. First, ultrapeers answer queries on behalf of their leaf peers. Second, query results may be relayed back along the network of peers in the reverse path that the query took from the originator. In both cases, intermediate peers could falsify the results to indicate a victim IP address is sharing contraband; however, these *query hits* are an excellent source of leads, as in practice they are not typically falsified.

4.1.2. Swarming information

When one peer downloads a file from another, the source peer will notify the downloading peer of others on the network that are sharing the file (as identified by SHA-1 hash). The remote peers are identified by both IP and GUID. This list allows the downloading peer the chance to request portions of the file from many peers in parallel. This information can be falsified, but is again a good source of leads for investigators.

4.1.3. Browse host

Gnutella allows a peer to create a TCP connection directly to another peer in order to query the full set of files that is being shared. Specifically, the remote peer will report the names and SHA-1 hash values of its shared files. This information is considered strong evidence by courts for probable cause since it is coming directly from the remote machine. It is unlikely that it will purport to share contraband if it does not have it, and furthermore, the probability of a non-contraband file forming a hash collision with the set of known contraband files is vanishingly small.

Specifically, for a 160-bit SHA-1 hash, the probability that a non-contraband file's hash value collides with a known contraband file's hash value is $p = 1/2^{160}$. We are interested in a more general scenario where there are a set of non-contraband files shared by users, and investigators have a distinct set of files that are known contraband. We want to know the probability of a false positive, where any one of all non-contraband files shared on Gnutella (by all users) forms a hash collision with one or more of the investigators' distinct set of contraband files. If we assume both sets are each of size n, then Girault et al. (Girault et al., 1988; Trappe and Washington, 2006) have shown this probability is approximated by

$$\Pr\{\text{False Positive}\} = 1 - e^{-n^2/2^{160}}$$

For example, when $n = 10^{16}$, the probability of a false positive is about 1.11×10^{-16} and falls precipitously with smaller n.

Far more likely reasons for false positives are that the user is claiming to share files that he does not actually possess, or that the hash value does not actually correspond to contraband. In the former case, the user could be deliberately reporting incorrect hash values for some reason, such as a malware infection (Brenner et al., 2004). The negligible probability of collision is why the hash values are sufficient to show probable cause, but the possibility of malware reporting a value for a file that is not possessed limits the usefulness of such evidence if gathered over the network in criminal prosecution. Stronger evidence can be acquired by downloading the file in question and evaluating its contents rather than its hash value.

4.1.4. File download

As with browses, file downloads are direct TCP connections to a remote peer, yielding an IP address and port. The remote peer directly transmits the requested portions of the file, and the content (and its hash value) can be validated directly by the investigator. Typically an investigator will attempt a single-source download, retrieving the entire file from a single peer.

Two complications can arise when attempting an SSD. First, the remote peer may be busy, and may place downloaders into a queue. In this case, the investigator must wait and risks the peer going offline in the interim. Second, the peer may be behind a firewall that prevents a direct TCP connection from the investigator to the peer. The Gnutella

protocol allows initiating peers to request that a remote peer connect out from behind their firewall back to the initiator; these *push* requests are routed back through the Gnutella network and thus penetrate the firewall.

4.1.5. *Other sources of evidence*

While not specific to Gnutella, it is possible to use any method of profiling a remote host over the Internet to collect evidence. Nmap (NMap) can profile many aspects of non-firewalled remote hosts, and more esoteric techniques can fingerprint remote devices on the basis of TCP clock skew (Kohno et al., 2005) through firewalls. We are not aware of the use of such techniques by law enforcement; the legality of either is murky, the former may be considered an intrusion attempt in some jurisdictions, and the latter is untested in court.

4.2. *BitTorrent*

There are four primary methods of gathering evidence using the BitTorrent protocol, which closely correspond with the methods for investigating Gnutella. However, specific features of BitTorrent make investigations more challenging.

4.2.1. *Tracker messages*

The primary purpose of the tracker is to track peers' interests in torrents and distribute contact information among peers interested in the same torrent. Its transmission of IP addresses and ports could be used as hearsay evidence to generate leads, however, such evidence is unreliable. As reported by Piatek et al. (2008), some trackers deliberately lace their replies with a small amount of false information. The purpose of such misinformation is to detract from evidence used in DMCA suits, but it provides a clear example of why hearsay evidence can be unreliable. Multi-tracker and distributed trackers provide the same hearsay evidence, with the same caveat regarding reliability.

4.2.2. *Piece information exchange*

When peers connect for a download, they transmit a list of the pieces of the files they possess, as described by the torrent of interest. Additionally, they send updates when a new piece is obtained. Like the browse in Gnutella, this information is relayed directly and is likely strong enough to issue a search warrant. Again like Gnutella, it is possible a peer could falsely state interest in a torrent or possession of a piece, though the motivation for doing so is unclear.

4.2.3. *Peer exchange*

The mainline BitTorrent client and the popular Vuze client both implement *peer exchange* protocols, analogous to download swarms in Gnutella. Specifically, they occasionally relay IP addresses and ports of other peers interested in the same torrent that is being exchanged. As above, this data is hearsay evidence.

4.2.4. *File download*

A file download occurs using a direct TCP connection. A peer requests and downloads *blocks*, which are small (typically 16 kB) fragments of pieces. Blocks are then assembled into pieces, which are checked against the per-piece hashes in the torrent to verify they are correct.

Performing a single-source download in BitTorrent brings additional complications not present in Gnutella. The choking of clients unwilling to upload — which describes law-abiding investigators of contraband — will severely limit an investigator's download rates. Implementations vary, but the mainline BitTorrent client rotates a single optimistic unchoke slot among connected peers, giving each 30 s of upload before choking again. Newer connections are weighted three times as heavily as long-running connections in the unchoke rotation. This behavior is intended to discourage leeching — that is, downloading without uploading — which is the very behavior an investigator attempting a SSD must exhibit.

Attempting to upload bogus data to peers to avoid choking is counterproductive; all recent BitTorrent applications will not only detect this behavior through the piece hash check, but ban the peer responsible for it. In the absence of specialized investigative tools that distribute downloads across multiple, coordinated peers and then reassemble the results, an investigator will be forced to contend with extremely long download times for larger files.

One solution to this problem is to focus on small, known contraband files (or portions of files, such as frames of a video) within the torrent. If these portions can be prioritized for download, this wait can be shortened commensurately.

4.3. *Evidence use and validation*

All of this evidence described in the preceding section is circumstantial, in that we are inferring that a computer (and ultimately, a person) behind an IP address is responsible for possessing or distributing contraband. Here, we discuss the specific legal uses of that evidence.

The first step of resolving an IP address into a person is to determine the location of the machine responsible for traffic on that address. With sufficient direct evidence, an investigator can obtain a subpoena from a magistrate, requesting that an ISP return account information for a given IP address at a given time. ISPs in the U.S. generally assign addresses through DHCP and often keep logs of these assignments. Comcast, for example, keeps these records for six months. Currently, relevant U.S. Federal law, such as the Communications Assistance for Law Enforcement Act (CALEA), does not mandate retention of these records.

If this street address is within the jurisdiction of the investigator, the investigator may obtain a search warrant from a magistrate, again on the basis of directly observed evidence. This search warrant specifies an address and targets; usually the targets are broadly defined as any electronic devices or media capable of storing or transmitting digital contraband, or evidence of intent.

Protocols vary by locale, but investigators will typically perform an onsite investigation of any computers on the scene. One goal is to corroborate evidence observed through network connections with data on the computer. In the case of Gnutella, finding a matching GUID is considered extremely strong evidence tying the computer to network traffic, as the probability of a randomly generated GUID on this computer

Table 2 – A summary of the observations made by law enforcement using RoundUp. All values are cumulative.

Date	Total Records	Recs/Day	U.S. Records	% U.S.	Unique U.S. IPs	U.S. GUIDs
10/31/2009	28,911,286	259,654	12,710,449	44%	1,202,640	149,720
11/30/2009	39,134,353	340,769	16,109,816	41%	1,266,907	175,705
12/31/2009	58,488,760	624,336	22,640,939	39%	1,368,360	221,590
1/31/2010	82,880,576	786,833	30,348,333	37%	1,457,731	261,944
2/28/2010	103,013,042	719,017	36,689,576	36%	1,547,363	306,008

matching a specific GUID is $1/2^{128} \approx 2.94 \times 10^{-39}$. Recovering the GUID is a well understood (Lewthwaite and Smith, 2008) process.[4] Additionally, the investigator may look for a shared folder, and compare its contents against the recorded browse results. For BitTorrent, the torrent file of interest will be sought. In either case, the detection of known contraband that was downloaded from the peer, as well as additional contraband, is a high priority—even if the specific contraband observed on the network is not found, other related contraband may be sufficient to start criminal prosecution. As discussed in Section 3.2, investigators will also seek other indications of knowing possession.

5. Tools and results

5.1. Overview

Our collaboration between law enforcement and academics has led us to develop *RoundUp*, a tool for forensically valid investigations of the Gnutella network. RoundUp is a Java-based tool that allows for both local and collaborative investigations of the Gnutella network, implementing the principles and techniques described in the previous sections. RoundUp is a fork of the Phex Gnutella client,[5] and it retains Phex's graphical user interface. Our changes in creating RoundUp from Phex focused on three key areas: adding specific functionality to augment investigative interactions, exposing information of interest to investigators in the GUI, and automating reporting of this information in standard ways.

Key features are as follows. Investigators can load and work from a list of previously identified files of interest, listed by hash, as well as GUIDs of interest. In addition to a remote peer's self-reported IP address (which may be wrong or non-routable in the case of an intervening NAT device), a peer's publicly visible IP address is displayed when available, such as after a successful push request. If the peer is firewalled, the push proxies it provides are displayed; later, or in another instance of RoundUp, an investigator can use this information to reconnect to that peer. IP geolocation is integrated into the GUI, and search results can be filtered on this basis to aid investigators in staying within jurisdiction. All relevant information can be selectively captured to a local comma-separated-value file during a browse or download, and may optionally be sent to a central server using authenticated HTTPS posts to help coordinate the efforts of law enforcement. Uploading of contraband to other peers is programmatically disabled.

We have also developed a web-based frontend to the centralized database. This frontend authenticates investigators, records the results of their investigations, and allows them to browse the submissions of all other investigators who use the database. It functions as a central point of coordination for investigation, preventing duplication of effort and allowing pooling of resources.

We are developing an analogous tool for BitTorrent investigations with similar functionality.

5.2. Deployment results

RoundUp has been in use since October 2009 by more than 52 ICAC Task Forces. A summary of the number of observations made by members of these Task Forces is in Table 2; each record corresponds to an observation of a file of interest. For example, 306,008 unique GUIDs have been observed sharing files that are known contraband from IP addresses within the U.S. GUIDs are not one-to-one with users over a long period of time; the column represents an upper bound on the number of users sharing at least one file of known contraband. In Table 3, we summarize the reporting thus far of law enforcement actions related to these observations. By the end of February 2010, 193 arrests have been made based on investigations using RoundUp. We note that these data are a lower bound, as not all investigators choose to report their arrest statistics back to us, nor do all investigators use the centralized database service we provide.

Finally, we point out the stark difference between the number of observed GUIDs sharing contraband and the number of search warrants. Identifying candidates sharing contraband on the Internet can take minutes. The remaining process leading to a search is a manual process requiring weeks of effort.

5.3. Distribution information

RoundUp is currently being made available to the law enforcement community on a limited basis. The GPL source code is distributed with the tool. Interested parties should contact the authors for more information. Our BitTorrent tool is currently in beta, and it not yet available.

[4] The GUID is typically saved to disk and stored across runs of a Gnutella client. For example, the GUID is labeled as the CLIENT_ID in LimeWire's limewire.props file, as Network.ServentGuid in Phex's phexCorePrefs.properties, and stored with portions endian-reversed as the <gnutella guid> in Shareaza's profile.xml.

[5] http://www.phex.org.

Table 3 – A summary of the RoundUp related law enforcement activity. All values are cumulative.

Date	Investigators	ICAC Task Forces	Cases	Search Warrants	Arrests
10/31/2009	102		748	242	15
11/30/2009	429	28	875	316	57
12/31/2009	472	48	1096	367	93
1/31/2010	502	51	1291	471	144
2/28/2010	587	52	1606	558	193

6. Conclusion

We have presented the Gnutella and BitTorrent p2p protocols and explored some of the legal and forensic issues relating to investigating these protocols. In particular, we have shown the importance of law enforcement personnel understanding the underlying systems, so as to know how their actions within an application correspond to their legal authority and limits, and the importance of tool developers understanding these constraints. We also presented RoundUp, an investigative tool built through close collaboration between law enforcement and computer science. We believe the success of RoundUp points the way toward the future of scientifically based investigative tools for crimes on the Internet.

Acknowledgements

We thank Janis Wolak and Dana Babbin for many illuminating discussions centered around this project.

This work was supported in part by National Institute of Justice Award 2008-CE-CX-K005 and in part by the National Science Foundation awards CNS-0905349 and DUE-0830876. The opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect those of their employers, the U.S. Department of Justice, or National Science Foundation.

REFERENCES

Brenner S, Carrier B, Henninger J. The trojan horse defense in cybercrime cases. Santa Clara Computer and High Technology Law Journal 2004;21(1).

Cohen B. The BitTorrent protocol specification, http://bittorrent.org/beps/bep_0003.html; June 2009. Version 11031.

Ferraro M, Casey E. Investigating child exploitation and pornography. Elsevier Academic Press; 2005.

Girault M, Cohen R, Campana M. A generalized birthday attack. In: LNCS on advances in cryptology (EUROCRYPT); 1988, p. 129–56.

Howard T. Don't cache out your case. Berkeley Technology Law Journal 2004;19(Fall).

Klingberg T, Manfredi R. The Gnutella RFC, version 0.6, http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html; June 2002.

Kohno T, Broido A, Claffy K. Remote physical device fingerprinting. IEEE Transactions on Dependable and Secure Computing April–June 2005;2(2).

Kulbak Y, Bickson D. The eMule protocol specification, http://www.cs.huji.ac.il/labs/danss/p2p/resources/emule.pdf; January 2005.

Lewthwaite J, Smith V. Limewire examinations. In: Proc. annual DFRWS conference; August 2008, p. S96–104.

Mitchell KJ, Wolak J, Finkelhor D. The national juvenile online victimization study: methodology report. UNH Crimes Against Children Research Center, http://unh.edu/ccrc/pdf/N-JOV2_methodology_report.pdf; Revised 2009.

NMap. Network mapper, http://nmap.org.

Piatek M, Kohno T, Krishnamurthy A. Challenges and directions for monitoring P2P file sharing networks. In: Proc. USENIX HotSec; July 2008.

Trappe W, Washington L. Introduction to cryptography: with coding theory. 2nd ed. Prentice Hall PTR; 2006.

Wolak J, Finkelhor D, Mitchell K. Trends in arrests of "online predators. Technical report. UNH Crimes Against Children Research Center, http://www.unh.edu/ccrc/pdf/CV194.pdf; March 2009.

Wolak J, Finkelhor D, Mitchell KJ. Child-pornography possessors arrested in internet-related crimes: findings from the National Juvenile Online Victimization Study. Technical report. National Center for Missing & Exploited Children; 2005.

Marc Liberatore joined the Dept. of Computer Science at the University of Massachusetts Amherst as a Research Scientist in January 2009. Previously, he served as a Mellon Postdoctoral Fellow at Wesleyan University in Middletown, Connecticut. He earned his Master's and PhD is Computer Science from UMass Amherst in 2004 and 2008, respectively. His research focuses on anonymity systems, digital forensics, peer-to-peer architectures, and disruption tolerant networking.

Robert Erdely is a member of the Pennsylvania State Police. He is the supervisor of the Computer Crime Unit which is responsible for computer crime investigations and the digital evidence section of the State Police lab. He holds numerous IT and digital forensic certifications. He has been a member of the Pennsylvania State Police for over 18 years and a member of the Computer Crime Unit for over 11 years.

Thomas Kerle is a Detective Captain with the Massachusetts State Police. He supervises the State Police Forensic Services Group which consists of the Digital Evidence and Multimedia Section (DEMS), the Internet Crimes against Children program (ICAC), the Crime Scene Services Section, and the Firearms Identification Section. He also trains police and prosecutors throughout the Nation and on occasion internationally and works closely with the National Internet Crimes Against Children program to provide training to its participants.

Brian Neil Levine is an Associate Professor in the Dept. of Computer Science at University of Massachusetts Amherst. He received a PhD in Computer Engineering from the University of California, Santa Cruz in 1999. His research focuses on mobile networks, forensics and privacy, and the Internet, and he has authored more than 60 papers on these topics.

Clay Shields is an Associate Professor in the Dept. of Computer Science at Georgetown University. He received a PhD in Computer Engineering from the University of California, Santa Cruz in 1999. His research focuses on digital forensics.