Evidence Collection in Peer-to-Peer Network Investigations

Teja Myneedu, Yong Guan

Chapter 15

EVIDENCE COLLECTION IN PEER-TO-PEER NETWORK INVESTIGATIONS

Teja Myneedu and Yong Guan

Abstract Peer-to-peer (P2P) file sharing networks are often abused to distribute content that is prohibited by law. Strong evidence of suspicion must be provided to obtain a court order to identify the location of an offender. However, initial evidence collection from a P2P network is a challenge due to the lack of a central point of control and the dynamic nature of the network. This paper describes an initial evidence collection tool for P2P network forensics. The tool performs active and passive monitoring by inserting a modified peer node in a P2P network that records relevant information about nodes that distribute contraband files. It logs data sent by suspicious nodes along with timestamps and unique identification information, which provides a strong, verifiable body of initial evidence.

Keywords: Peer-to-peer networks, evidence collection

1. Introduction

The distribution of child pornography is one of several criminal activities that are carried out in peer-to-peer (P2P) networks. A Government Accountability Agency (GAO) report noted that 42% of the search results using child-pornography-related keywords yielded P2P files [5]. A Palisade Systems survey discovered that more than 40% of the 40,000 queries issued in Gnutella over a three-week period were for adult or child pornography [7]. The prevalence of this illegal content and its easy access are troubling.

Law enforcement has to be particularly cautious when investigating illegal P2P file sharing activities. Evidence must be verified at every level to build a strong case and to ensure that innocent individuals are

not falsely implicated. An example is the Interhack investigation where a teenager was accused of possessing child pornography [14]. An outside investigator used login and file access timestamps recorded by the operating system to prove that the contraband files were downloaded by malware instead of by the teenager.

A critical step in a P2P network investigation is the correct identification of the IP address and the physical location of the computer that was used to distribute contraband files. This paper describes an evidence collection tool designed for real-world P2P networks. The tool supports the Gnutella2 protocol [6], which uses a hierarchy of nodes to implement an efficient search mechanism. The tool can be used to uniquely identify P2P nodes based on their IP addresses and globally unique identifiers (GUIDs). Also, the tool can collect critical information such as the contraband files associated with a keyword and the nodes in a P2P network that share a particular file.

2. Background

A P2P protocol defines the network maintenance and search mechanisms for connecting nodes and locating and transferring files. A P2P network investigation involves the monitoring of the network topology and the messages exchanged between nodes to obtain information about contraband files that are being shared. Monitoring can occur at the network layer and/or at the application layer. In general, monitoring can be classified as active monitoring or passive monitoring. Our work builds on the P2P traffic monitoring study of Hughes, *et al.* [11], which did not consider passive application-level monitoring.

2.1 Passive Network-Level Monitoring

Passive network-level monitoring records TCP/IP packets by positioning agents at appropriate locations in the physical network infrastructure. The agents capture data in a manner that is transparent to the underlying network. The positioning of data collection agents is critical. For example, placing an agent at the gateway of a large network enables the recording of all traffic routed to nodes in the network. On the other hand, placing an agent in a network segment of a switched LAN only enables the recording of traffic associated with a single node.

An important requirement in performing passive network-level monitoring is effective traffic classification. In addition to distinguishing between non-P2P traffic and P2P traffic, the different types of P2P traffic should be classified [17]. Saroiu, *et al.* [18] and Gummadi, *et al.* [8] identified traffic by port number and header data, respectively. Sen,

et al. [20] have used "application signatures" to classify real-time network traffic. Iliofotou, *et al.* [13] have proposed a dispersion graph based approach to classify different types of network traffic.

These studies have found that passive network-level monitoring is transparent, but it introduces additional overhead for effective traffic classification. If the payload is encrypted, little can be done after the traffic is classified. Also, access to key segments of the network infrastructure such as Internet service provider gateways is required, but such access may not be available.

2.2 Active Network-Level Monitoring

Active network-level monitoring injects packets into a network or marks packets near a source node and looks for the marked packets near a destination node. This method cannot be used for evidence collection because introducing content into a network raises questions about the validity of the collected evidence.

2.3 Passive Application-Level Monitoring

Passive application-level monitoring involves the introduction of a modified client in a network. The modified client participates in regular message exchanges with its neighbors and logs messages (e.g., network maintenance and search messages) received from its neighbors. In general, passive application-level monitoring is transparent to a P2P network.

Passive application-level monitoring records all incoming and outgoing messages between the client and its neighbors. Passive application-level monitoring has been conducted in the Gnutella [1, 9, 10] and eDonkey [2, 3] networks. Passive application-level monitoring is suitable when a non-invasive approach is desired. Unfortunately, due to the small-world properties of modern P2P networks [1], a single client cannot collect pervasive information about all the nodes in a P2P network. Attempts to address this problem include disconnecting and reconnecting the client randomly to increase its coverage [2], and using multiple modified clients [3].

2.4 Active Application-Level Monitoring

Active application-level monitoring deploys a modified client in a P2P network that actively queries the network to collect as much information as possible. In passive monitoring at the application level, only the messages that the client exchanges with its neighbors or receives from

remote clients are logged. However, in the case of active monitoring, the client sends queries to nodes in the network to collect information about them. Also, the modified client connects to a large number of neighbors and repeatedly reconnects to different nodes in the network to cover as much of the network as possible.

A crawler program was used by Saroiu, *et al.* [19] over a period of one month to obtain information from a Gnutella network. The crawler recorded low-level data such as IP addresses and bandwidth, and high-level data about the files being shared. Chu, *et al.* [4] have observed that file popularity is highly skewed with the top 10% of files accounting for more than 50% of the shared files.

Active application-level monitoring can gather a large amount of information about remote nodes in a network within a short period of time, but at the expense of transparency. Since this type of monitoring is very noisy and intrusive, it can result in nodes being blacklisted and being unable to connect to the network. Active monitoring at the application level can be very effective, especially if intrusiveness is not a factor and global network information is required.

2.5 Evidence Collection in P2P Networks

Latapy, *et al.* [15] conducted a large-scale passive application-level tracing and extraction of child pornography keywords in a P2P network. Liberatore, *et al.* [16] developed a BitTorrent monitoring system for forensic evidence collection that uses GUID information for evidence validation. However, both these efforts did not address the problem of obtaining information about new keywords and files during the initial stages of a P2P network investigation. Also, they did not examine the differences between popular files and less popular files when monitoring P2P networks.

3. P2P Networks

P2P networks can be broadly classified based into three categories based on their topology: (i) centralized networks; (ii) completely decentralized networks; and (iii) semi-centralized networks.

In a centralized network, one central index/directory server maintains a list of all active/live nodes in the network. Nodes that wish to join the network send information about themselves to the central index server. Since the nodes are dependent on the central server for communications, a centralized network is not a pure P2P network. The popular BitTorrent P2P network is an example of a centralized network.

In a completely decentralized network, all the nodes in the network are identical in terms of functionality and information stored. No node contains more information or performs more functions than its peers. Also, there is no central index server. Gnutella v0.4 is an example of a completely decentralized network.

In a semi-centralized network, multiple nodes contain indexing information. An inherent hierarchy in the network is created by elevating a few nodes in the network to be "super nodes" or "hubs." The hubs perform functions in addition to those performed by regular peers. The additional functions include maintaining an index of the information possessed by neighboring nodes, routing queries, and maintaining and sharing routing information about one-hop and two-hop neighbors. Gnutella2 [6] is an example of a semi-centralized network and is the focus in this work.

4. Gnutella2

Although Gnutella2 was inspired by the Gnutella protocol, it is not an official extension of Gnutella. The strength of Gnutella2 lies in its optimization of the search mechanism. A hierarchy is introduced into the network whereby nodes are categorized as leaf nodes or hub nodes.

Leaf nodes are the most common nodes in a Gnutella2 network. These nodes contribute to file sharing, but do not have any special responsibilities. Hence, they are not critical to the network infrastructure and can join or leave the P2P network at any time without affecting the network.

Hub nodes are powerful nodes that contribute substantial resources to the P2P network. Gnutella2 hubs are highly interconnected, forming a "hub network" in which each hub maintains a connection to six to 32 other hubs. Each hub also accepts connections from a large collection of leaf nodes, typically 200 to 300 nodes, depending on the available resources. A group of hubs within a hub network that spans the local hub and its neighbors is called a "hub cluster." Hub clusters are in constant communication with each other, sharing information about network load and statistics, exchanging cache entries and filtering tables. A hub cluster is the smallest searchable unit of a P2P network as far as a search client is concerned.

5. Evidence Collection Challenges

A forensic investigation of a digital crime involves several stages of evidence collection. The challenges faced during an investigation can be broadly classified as legal and technical challenges.

The legal challenges include jurisdiction, privacy and the dissemination of illegal content. Only a few countries in the world have cyber laws. When key evidence is outside the jurisdiction of a law enforcement agency, it often becomes infeasible to conduct the investigation.

The Fourth Amendment of the United States Constitution protects the privacy of citizens. Therefore, an investigation that is conducted before a formal search warrant is issued can only use public domain information. Since information is exchanged freely by nodes in a P2P network and the information collected via a search is voluntary, the collected information can be considered to be in the public domain. Nevertheless, to abide by privacy laws, it is important not to perform an intrusive search or evaluation.

Forensic tools sometimes download files from a P2P network for purposes of verification. However, in order to maintain the soundness of a forensic investigation, it is imperative that no files are uploaded (even accidentally) during the investigation. Indeed, great care should be taken to ensure that an evidence collection tool for P2P networks will never upload criminal content.

5.1 Blacklisting

IP filtering software can be installed on nodes to prevent devices with certain IP addresses from accessing P2P nodes. Services such as iblocklist [12] provide IP addresses associated with government organizations, corporations and other entities that should be "blacklisted" for anti-P2P activities. Obviously, it is important to ensure that the IP address associated with an initial evidence collection tool for P2P networks is not blacklisted.

5.2 Encryption

Encrypted communications between peers complicates the monitoring of P2P traffic at the network level. Even if network monitors are placed at multiple locations on the Internet, the use of encryption makes it virtually impossible to obtain meaningful information from a network packet dump. To be effective, an initial evidence collection tool should be able to perform its functions even when network data is encrypted.

5.3 Evidence Validation

Evidence collected at every stage of an investigation must be validated. A flaw in the initial stages of evidence collection can ruin an entire investigation. Our goal is to ensure that the information collected is validated sufficiently in the early stages of an investigation. Care

should be taken when collecting information to enable it to be verifiable at a later stage of the investigation. Moreover, all relevant information – incriminating and exculpatory – must be collected and safeguarded.

5.4 Storage

It is important to collect as much relevant information as possible during the initial stages of an investigation. However, nodes in a P2P network share a large amount of metadata for maintenance and operational purposes. Logging every single incoming and outgoing packet would quickly require a massive amount of storage. Therefore, it is necessary to filter packets that are not relevant to an investigation.

5.5 Processing Power

The processing power of an initial evidence collection system has a direct impact on its cost. The large size of a P2P network makes it difficult to conceive of a solution where a single monitor can obtain information about the entire network, unless, of course, it probes every node in the network. It is, therefore, necessary to perform a cost-benefit analysis and optimize the processing power requirements.

5.6 Network Access

The positioning of monitors is an important factor when network-level monitoring is to be performed. If a graph of a P2P network is available, then the placement of monitors can be formulated as a graph covering problem. Obviously, access to the underlying network infrastructure would be required to implement network-level monitoring.

6. Evidence Collection Tool

Based on the P2P network monitoring challenges and the evidence collection goals described above, the following constraints must be satisfied for a P2P monitoring tool to be effective:

- The tool must collect as much information as possible about the location of the criminal content and the time that it existed.
- It must be possible to validate the evidence collected by the tool at some stage of the investigation.
- The tool must monitor the network as non-intrusively as possible.
- The tool must not participate in uploading criminal content.
- The tool must be able to collect encrypted information.

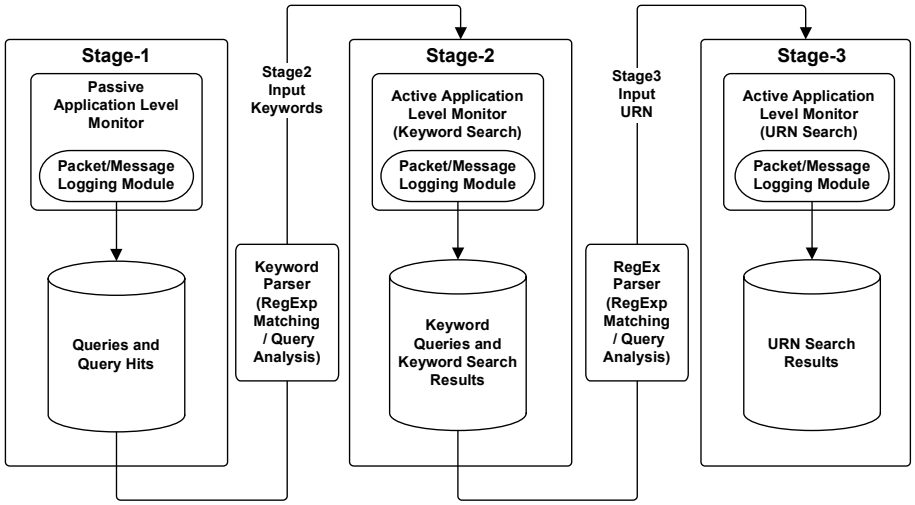


Figure 1. Monitoring framework.

Active application-level monitoring is the most appropriate method that satisfies these constraints. As discussed above, this type of monitoring can collect a large amount of information in a short period of time. However, because of its invasive nature, active monitoring poses a significant risk of the IP address of the evidence collection tool being blacklisted. Passive application-level monitoring addresses this issue, but it collects a limited amount of information. Since both approaches have complementary advantages, our evidence collection tool combines the two approaches so that the disadvantages of passive monitoring are offset by active monitoring and vice versa.

Figure 1 presents the monitoring framework, including the different stages of evidence collection and monitoring approaches.

In Stage 1, a passive monitor is deployed in the network to log all application-level messages. The monitor records all Query and Query Hit messages. The Query messages are parsed to obtain keywords related to illegal content in the network. The Query Hit messages are used to identify the network nodes that contain illegal content.

Stage 2 uses the keywords identified in Stage 1 that potentially list contraband files. Active keywords searches are performed and the universal resource names (URNs) of the files returned for each keyword are logged. Note that URNs are hash values that uniquely identify files.

Stage 3 uses the URNs obtained in Stage 2. This stage is crucial because it accurately identifies the nodes that possess and distribute criminal content. Active searches are also launched during Stage 3, but

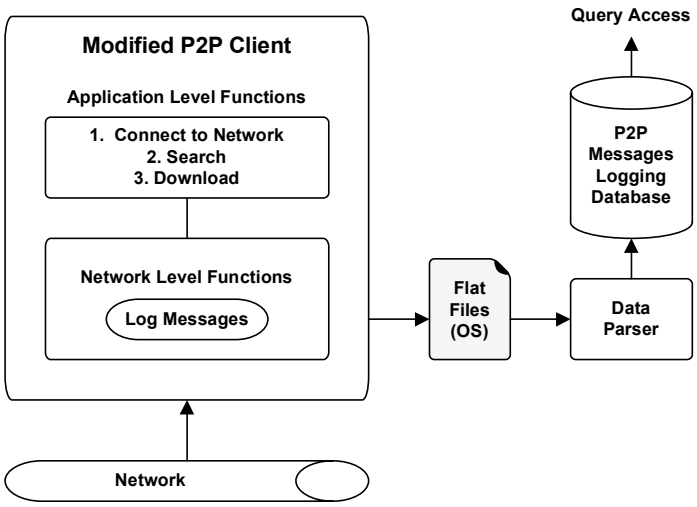


Figure 2. P2P monitor.

URNs are used instead of keywords to identify as many nodes as possible that share contraband files.

Each stage in the evidence collection framework uses a P2P monitor. Figure 2 shows the major components of a monitor: (i) re-engineered P2P client; (ii) message parsing tool; (iii) database for storing the collected information; and (iv) database querying tool.

Since the hubs in a semi-centralized network are responsible for forwarding search queries, they see a larger number of search queries and results compared with leaf nodes. Therefore, passive application-level monitoring is most effective in a semi-centralized network when the monitoring tool is located at a hub.

The modified client is a re-engineered open-source Windows-based Gnutella 2 client called Shareaza [21]. The re-engineered client is introduced into the network to log application-level messages (packets) that are sent and received. The messages are logged in flat files. The re-engineered client is programmed to halt its execution twice a day, at which time a Perl script copies the flat files to a MySQL database. The MySQL database is then queried to analyze the collected information.

Based on the Gnutella2 protocol specification [6], the following application level packets (messages) yield information that can be used in P2P network investigations:

- **Search Query (Q2) Messages:** These messages contain the universal resource name (URN), descriptive name (DN) (generic) criteria, metadata criteria and object size restriction criteria.

- **Query Acknowledgement (QA) Messages:** These messages are used by a target hub to acknowledge the receipt of a query and to indicate that it is processing the query. The target hub also provides information to ensure that hubs are not searched more than once in a given query. Lists of "done nodes" and "search hub" nodes are sent to the querying client. The done nodes are neighbors of the target hub for which search queries do not have to be sent because they have already been searched by the target hub. The search hub nodes are additional hub nodes in the cache of the target hub that have not been searched by the target hub.
- **Query Hit (QH) Messages:** These messages contain one or more search result descriptors, along with information about a node such as GUID, node address (IP address + port number), neighbor hub address, descriptive name, metadata and timestamp when the result was generated. In response to a single query message, a node could respond with multiple Query Hit messages.
- **Local Node Information (LNI) Messages:** These messages are sent to the neighbors of a node at regular intervals. The messages contain the GUID and local library information.

The Gnutella2 application layer protocol messages listed above can be used to collect information about a P2P network and the files distributed by its nodes. The collected information can be analyzed to provide valuable information about nodes that possess and distribute illegal content.

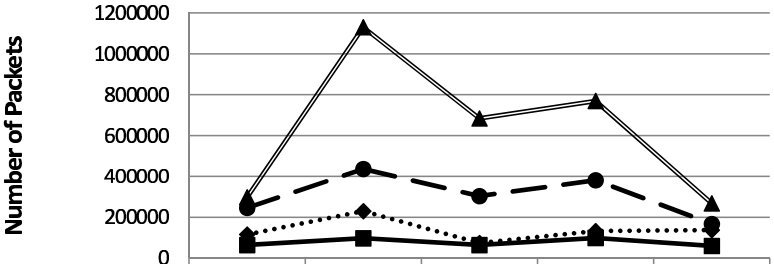
7. Evaluation

The monitoring tool can be evaluated based on its ability to: (i) identify new criminal content in a network by passively logging search hits; and (ii) identify the nodes in a network that possess a particular file along with their GUID values. This section describes three experiments that were conducted to evaluate the tool and its ability to conduct active and passive monitoring of a Gnutella2 network.

7.1 Passive Monitoring

This experiment attempted to assess the operation of the re-engineered client. In the experiment, the re-engineered client was introduced into the Gnutella-2 network and allowed to operate for five days.

The re-engineered client successfully logged the IP address, date, time and message payload for all Gnutella 2 packets. The numbers of relevant incoming messages by type are shown in Figure 3 and Table 1.



	Day 1	Day 2	Day 3	Day 4	Day 5
QH2 Packets	115785	230024	74474	132655	137496
QA Packets	64238	97321	64210	99033	59652
Q2	297854	1130569	683694	769490	268320
LNI Packets	246034	436424	303545	381272	167270

Figure 3. Messages recorded by type.

Table 1. Keywords extracted (five-day data collection).

Day	Keywords Extracted from Q2	Keywords Extracted from QH2
Day 1	2,592	1,317
Day 2	4,427	1,908
Day 3	3,646	1,756
Day 4	3,853	1,689
Day 5	2,014	1,352

7.2 Active vs. Passive Monitoring

In this experiment, two re-engineered clients were introduced into the network, one as a leaf node and the other as a hub. The leaf node operated as an active monitor and performed searches for two keywords, a popular keyword (Keyword A) and a mildly popular keyword (Keyword B). The searches were executed once a day and the results were logged.

The logged messages were analyzed to obtain the number of nodes with unique IP addresses that possessed files containing the keywords of interest. The re-engineered hub client ran in a passive monitoring mode for five consecutive days to collect all incoming and outgoing Query Hit messages containing Keyword A and Keyword B.

The graphs in Figures 4 and 5 show the total number of Query Hit messages received by the active and passive monitoring clients for Keyword A and Keyword B, respectively. In order to get a better idea of the number of network nodes that possessed the file, the number of unique IP address and port number combinations from which a Query Hit mes-

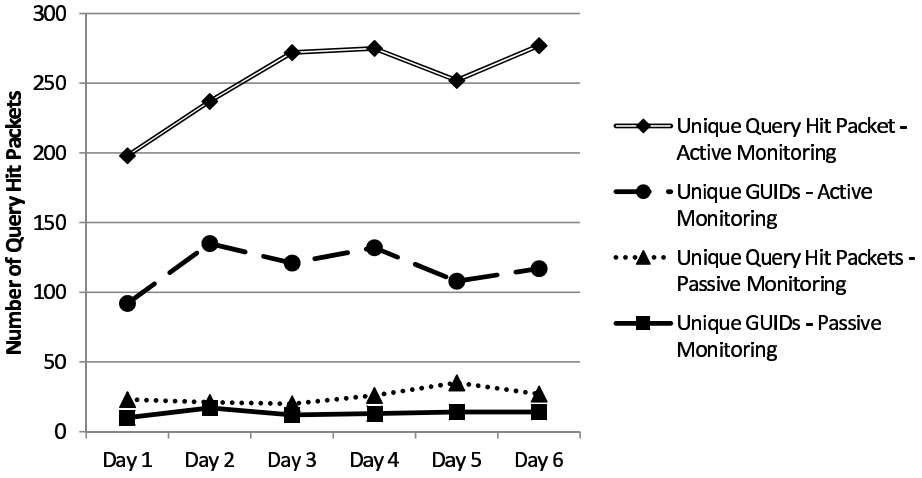


Figure 4. Active vs. passive monitoring (Keyword A).

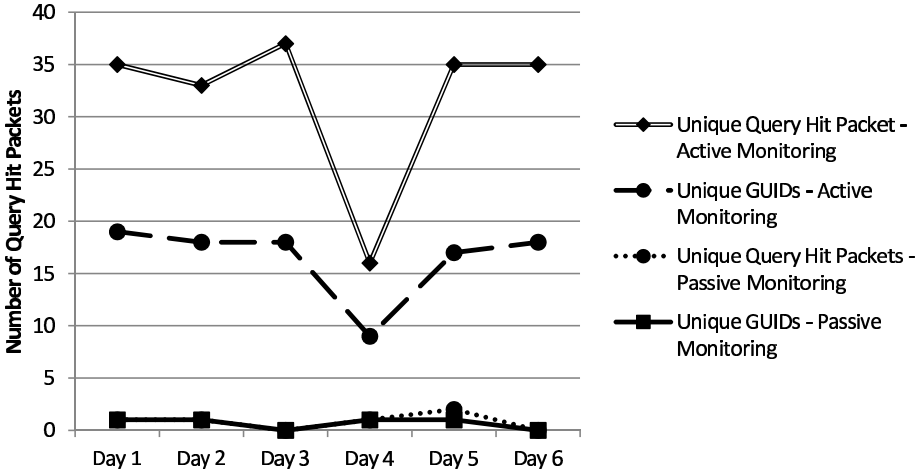


Figure 5. Active vs. passive monitoring (Keyword B).

sage was received was extracted from the collected data. Note that the combination of IP address and port number was used to uniquely identify nodes because most many-to-one NATs map unique ports on a public-facing IP address to computers in an internal network.

Figures 4 and 5 show that, for Keyword A, the number of search hits logged by the passive monitoring node was about 10% of the hits logged by the active monitoring node. Hence, active keyword-based searches can identify more unique files than passive keyword-based searches.

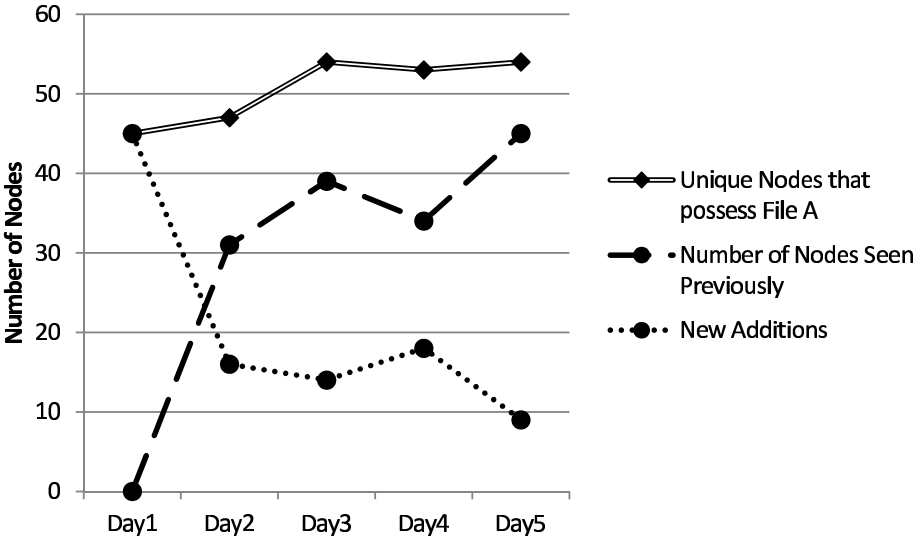


Figure 6. Five-day tracking information (Keyword A file).

Table 2. Five-day tracking information (Keyword A file).

Day	Unique Nodes with File A	Nodes Seen Previously	New Additions
Day 1	25	0	25
Day 2	27	21	6
Day 3	33	23	4
Day 4	32	32	6
Day 5	33	33	0

7.3 Active File Tracking

In this experiment, a leaf node was used to query the P2P network for URNs. File URNs for files associated with Keywords A and B were chosen from the keyword search results and messages logged in Experiment 2. Two URN searches, one each for the Keyword A and Keyword B files, were performed each day for five consecutive days. The search results were logged by the node.

The data was analyzed to determine the unique nodes that contained a file on Day 1 and the subsequent propagation of the file. There was no indication when a file was downloaded by a node. However, if a node was in the search results and did not possess the file, then it is possible that the file was recently downloaded by the node.

Figure 6 and Table 2 show the trend in the unique nodes that possessed a file with Keyword A. Figure 7 and Table 3 provide similar infor-

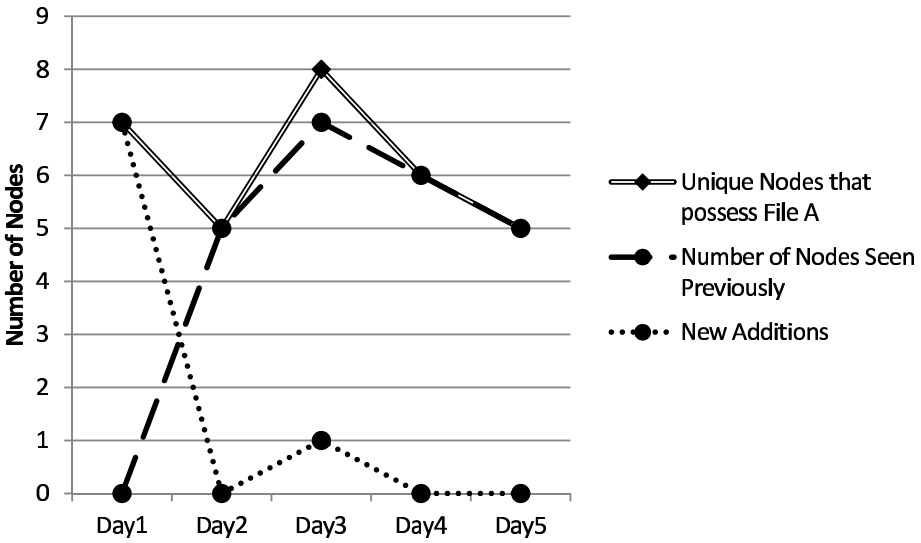


Figure 7. Five-day tracking information (Keyword B file).

Table 3. Five-day tracking information (Keyword B file).

Day	Unique Nodes with File B	Nodes Seen Previously	New Additions
Day1	7	0	7
Day2	5	5	0
Day3	8	7	1
Day4	6	6	0
Day5	5	5	0

mation corresponding to Keyword B. The results show that the number of new nodes that possessed a file corresponding to the popular keyword (Keyword A) reduced drastically after three to four days. However, in the case of the mildly popular keyword (Keyword B), the number of new nodes that possessed a file was almost zero every day.

8. Conclusions

The evidence collection tool described in this paper is designed to perform active and passive monitoring of P2P networks by inserting modified peer nodes that record P2P network activity. The experimental results demonstrate that the combination of passive and active monitoring is effective at identifying Gnutella2 nodes that possess and distribute contraband files. Furthermore, techniques such as timestamping

and public domain information gathering render the tool practical for real-world P2P network investigations.

Our future research will attempt to maximize the information collected in the passive monitoring mode. Also, it will leverage the logged messages to reconstruct hub networks.

Acknowledgements

This research was partially supported by NSF Grants CNS 0644238 and CNS 0831470.

References

[1] E. Adar and B. Huberman, Free riding on Gnutella, *First Monday*, vol. 5(10), October 2, 2000.

[2] F. Aidouni, M. Latapy and C. Magnien, Ten weeks in the life of an eDonkey server, *Proceedings of the Twenty-Third IEEE International Symposium on Parallel and Distributed Processing*, 2009.

[3] O. Allali, M. Latapy and C. Magnien, Measurement of eDonkey activity with distributed honeypots, *Proceedings of the Twenty-Third IEEE International Symposium on Parallel and Distributed Processing*, 2009.

[4] J. Chu, K. Labonte and B. Levine, Availability and locality measurements of peer-to-peer file systems, *Proceedings of the SPIE Information Technologies and Communications Conference*, pp. 310–321, 2002.

[5] General Accounting Office, Peer-to-Peer Networks Provide Ready Access to Child Pornography, Technical Report GAO-03-351, Washington, DC, 2003.

[6] Gnutella2 Developer Network, The Gnutella2 Protocol, Boston, Massachusetts (`g2.trillinux.org`).

[7] S. Grabowski, The Real Cost of "Free" Programs such as Instant Messaging and Peer-to-Peer File Sharing Applications, InfoSec Reading Room, SANS Institute, Bethesda, Maryland, 2003.

[8] K. Gummadi, R. Dunn, S. Saroiu, S. Gribble, H. Levy and J. Zahorjan, Measurement, modeling and analysis of a P2P file-sharing workload, *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, pp. 314–329, 2003.

[9] D. Hughes, G. Coulson and J. Walkerdine, Free riding on Gnutella revisited: The bell tolls? *IEEE Distributed Systems*, vol. 6(6), 2005.

[10] D. Hughes, J. Walkerdine, G. Coulson and S. Gibson, Peer-to-peer: Is deviant behavior the norm on P2P file sharing networks? *IEEE Distributed Systems*, vol. 7(2), 2006.

[11] D. Hughes, J. Walkerdine and K. Lee, Monitoring challenges and approaches for P2P file sharing systems, *Proceedings of the International Conference on Internet Surveillance and Protection*, 2006.

[12] iblocklist, I-Blocklist (`www.iblocklist.com/lists.php`).

[13] M. Iliofotou, P. Pappu, M. Faloutsos, M. Mitzenmacher, S. Singh and G. Varghese, Network Traffic Analysis using Traffic Dispersion Graphs (TDGs): Techniques and Hardware Implementation, Technical Report UCR-CS-2007-05001, Department of Computer Science and Engineering, University of California Riverside, Riverside, California, 2007.

[14] Interhack, There goes the neighborhood: A forensic computing case study, Columbus, Ohio (`web.interhack.com/publications/crimdef-casestudy.pdf`), 2009.

[15] M. Latapy, C. Magnien and R. Fournier, Quantifying pedophile queries in a large P2P system, *Proceedings of the Thirtieth IEEE International Conference on Computer Communications*, pp. 401–405, 2011.

[16] M. Liberatore, R. Erdely, T. Kerle, B. Levine and C. Shields, Forensic investigation of peer-to-peer file sharing networks, *Digital Investigation*, vol. 7(S), pp. S95–S103, 2010.

[17] D. Plonka, UW-Madison Napster Traffic Measurement, University of Wisconsin – Madison, Madison, Wisconsin (`net.doit.wisc.edu/data/Napster`), 2000.

[18] S. Saroiu, K. Gummadi, R. Dunn, S. Gribble and H. Levy, An analysis of Internet content delivery systems, *Proceedings of the Fifth Symposium on Operating System Design and Implementation*, pp. 315–327, 2002.

[19] S. Saroiu, K. Gummadi and S. Gribble, Measuring and analyzing the characteristics of Napster and Gnutella hosts, *Multimedia Systems*, vol. 9(2), pp. 170–184, 2003.

[20] S. Sen, O. Spatscheck and D. Wang, Accurate, scalable in-network identification of P2P traffic using application signatures, *Proceedings of the Thirteenth International Conference on the World Wide Web*, pp. 512–521, 2004.

[21] Shareaza Development Team, Shareaza P2P (`shareaza.sourceforge.net`).