A Study of Malware in Peer-to-Peer Networks

Andrew Kalafut
akalafut@cs.indiana.edu

Abhinav Acharya
aacharya@cs.indiana.edu

Minaxi Gupta
minaxi@cs.indiana.edu

Computer Science Department
Indiana University, Bloomington
Bloomington, Indiana, USA

ABSTRACT

Peer-to-peer (P2P) networks continue to be popular means of trading content. However, very little protection is in place to make sure that the files exchanged in these networks are not malicious, making them an ideal medium for spreading malware. We instrument two different open source P2P networks, Limewire and OpenFT, to examine the prevalence of malware in P2P networks. Our results from over a month of data show that 68% of all downloadable responses in Limewire containing archives and executables contain malware. The corresponding number for OpenFT is 3%. Also, most infections are from a very small number of distinct malware. In particular, in Limewire, the top three most prevalent malware account for 99% of all the malicious responses. The corresponding number for OpenFT is 75%. We also investigate the sources of malicious responses. To our surprise, 28% of all malicious responses in Limewire come from private address ranges. In OpenFT, the top virus, which accounts of 67% of all the malicious responses, is served by a single host. Further, our study provides a useful insight into filtering malware: filtering downloads based on the most commonly seen sizes of the most popular malware could block a large portion of malicious files with a very low rate of false positives. While current Limewire mechanisms detect only about 6% of malware containing responses, our size based filtering would detect over 99% of them.

Categories and Subject Descriptors

C.2.4 [**Computer-Communications Networks**]: Distributed Systems

General Terms

Measurement, Security

Keywords

Malware, peer-to-peer, Limewire, OpenFT, filtering

IMC'06, October 25–27, 2006, Rio de Janeiro, Brazil.

1. INTRODUCTION

Peer-to-peer (P2P) networks like Limewire [8] continue to be a popular medium for sharing content and many organizations regularly rely on them for large-scale software distribution. Several previous studies have evaluated the nature of traffic on P2P networks [4, 5, 7] and the presence of malware and other types of attacks in various Internet environments [6, 11, 9, 10, 12, 18]. In particular, work in [18] uses firewall logs to determine the Internet-wide distribution of malware and a crawler-based study of malware in the Internet is conducted by [11]. Work in [20, 19] discusses the threat of worms in P2P networks based on network characteristics. Jung et al [13] crawled the Kazaa P2P network to monitor the presence of malware in popular downloaded content over two periods of 3 days each. Our study looks at two different P2P networks, Limewire [8] and OpenFT [2]. Further, we observe all search queries for a much longer period of time to estimate the extent of malware. In particular, we ask the following questions:

- What types of malware are being spread through P2P?
- What is the extent of malware in P2P networks?
- Do certain queries fetch more malware than others?
- What are the characteristics of hosts serving malware?
- How does malware differ across different P2P networks?
- Can effective filtering techniques be designed to protect P2P systems from malware?

We instrument Limewire and OpenFT to passively monitor the results of content search queries and then actively download files that match our criteria of being potentially harmful. These files are then examined by ClamAV malware scanner software [16] in order to determine if they contain malware. In order to keep the bandwidth used to download files low, we focus primarily on executable, archival, and Microsoft Office file formats because of the immense damage malware embedded in such files can do. Another reason to eliminate media files is because malware contained in media files must rely on buffer overflow type of attacks in the playback code of specific applications to do damage.

The data presented here is collected over a period of 45 days for Limewire and 37 days for OpenFT. The key findings from this preliminary version of our study include:

- *Malware types:* Both P2P systems experience a large number of distinct types of malware. We find 95 distinct types of malware in Limewire data and 38 in OpenFT.

- *Commonality:* The most popular malware is the same across Limewire and OpenFT. Of the top 10 distinct malware in each system, 5 are common. Further, a majority of infections in both systems are from just a few distinct types of malware. In particular, the top three most prevalent malware account for 98.5% of all the malicious responses in Limewire. The corresponding number for OpenFT is 75%.

- *Extent:* 68% of all downloadable responses containing executable, archival, and Microsoft Office file extensions contain malware in Limewire. The corresponding percentage for OpenFT is 3%. Most of the infections appear in *zip* and *exe* files.

- *Queries:* Queries containing movie names fetch the most malware in Limewire. In OpenFT, no such trend is observed.

- *Host characteristics:* Malicious responses in Limewire are more likely to come from private address ranges. These addresses account for 1.5% of all responses, but 28% of all malicious responses. In OpenFT, the most seen malware, accounting for 67% of all the malicious responses, is served by a single host.

- *Filtering:* Limewire's current system detects about 6% of malware with a 17% false positive rate. We propose a simple filtering criteria based on file sizes, which detects over 99% of malware with very few false positives.

The rest of this paper is organized as follows. Sections 2 and 3 describe data collection and analysis respectively. Section 4 describes our filtering method for detecting malware in P2P systems. Finally, Section 5 concludes and discusses the limitations of our study.

2. METHODOLOGY

2.1 Systems Studied

We considered certain criteria when selecting which P2P systems to study. First, they had to be *open source* so we could make the necessary modifications. Second, since observing actual downloads is not an option due to privacy concerns, the search mechanism had to be *decentralized* and the query replies had to be returned *in-band*, so we could record the searches. Based on these reasons, and their popularity, we have chosen the Limewire [8] and OpenFT [2] systems.

2.1.1 Limewire

Gnutella [3] is a popular P2P protocol with many software variants. While older Gnutella clients treated all nodes equally, newer implementations have two types of nodes: *leafs* and *ultrapeers*. The *ultrapeers* shield the *leaf* nodes attached to them from seeing most queries in order to minimize traffic. The response to a query is either sent back *in-band* along the query arrival path or *out-of-band* directly to the node originating the query. We use Limewire[8], a popular Gnutella implementation and connect as an *ultrapeer* in order to see more traffic.

2.1.2 OpenFT

OpenFT [2], short for "Open FastTrack," is an open source decentralized peer-to-peer system. Despite the name, it does not use the FastTrack protocol used in Kazaa. Nodes in the OpenFT system can operate in three modes: *user*, *search*, or *index*. A *user* node is the end user who submits queries and receives responses. *Search* nodes are similar to Gnutella's *ultrapeers* and are responsible for storing the list of shared files for users connected to it. An *index* node is responsible for maintaining the list of search nodes and for collecting statistics. When a user submits a query, the *search* node executes the search on its database of shared user files. If the number of matches found is less than a pre-determined number, then it forwards the query to its peer *search* nodes. These then execute the same and pass it on to their peers and so forth. The replies travel back along the same path to the original *search* node which hands the results to the user who requested it. We use the OpenFT plugin of the giFT file transfer program. This is the original implementation of OpenFT. Further, we operate our node in *search* mode in order to see the most queries and responses.

2.2 Instrumentation

To collect data for this study, we made several modifications to Limewire and OpenFT. We have added to both clients the ability to remember information for queries sent through us. This is used to later match the query with the response sent for it. Any files seen in query responses that meet our downloading criteria (described in Section 2.3) are automatically downloaded and scanned with the ClamAV[16] open source malware scanner. ClamAV contains signatures for over 65,000 malware (including variants). The ClamAV signature database is frequently updated, often several times within a day. Since Limewire permits both in-band and out-of-band replies for the queries, we also instrumented our Limewire client to modify all query packets sent through our node to disable requests for out-of-band replies. This allowed us to see all the responses to these queries.

We make both of our clients disconnect and reconnect periodically (every 12 hours for Limewire and every two days for OpenFT) to ensure we see diverse views of the P2P network upon every connection. Further, we have disabled the upload capabilities of both programs in order to avoid serving any illegal content we may inadvertently download during the data collection.

2.3 Data Collection

Since it is not common for files with media files extensions to contain malware, we choose the extensions to download based on what Limewire considers to be program files[1]. To these executable and archival file formats, we add Microsoft Office files, *doc*, *ppt*, and *xls*, because they can potentially contain macro viruses. Files of the types we choose to download account for 7.5% of all responses seen in Limewire and 1.3% of all responses seen in OpenFT. This choice saves the bandwidth required to download files and allows us to focus on the most suspect files in this preliminary study. Notice that executable files disguised as other formats, such as "desired_song.mp3.exe" will be caught by our methodology.

Further, in general we do not attempt to download a file again if we have already successfully downloaded and scanned a file with an identical name and size. We decide on this bandwidth saving strategy because our initial re-downloads of both "clean" and infected files failed to change the conclusion about the nature of files. Further, many popular malware these days change names upon each infection (and may change size if they are polymorphic), and we have observed the [file name, size (in bytes)] combination in general to be very unique. An exception to the above download strategy is made in Limewire in order to detect malware which may not have had a signature ready when we first see it. In this ex-

[1]These are *ace, arj, awk, bin, bz2, cab, csh, cue, deb, dmg, exe, gz, gzip, hqx, iso, jar, jnlp, lzh, lha, mdb, msi, msp, nrg, pl, rar, rpm, sh, shar, sit, tar, taz, tgz, z, zip, zoo* and *7z*.

ception, if we determine a file to be "clean" and it has been less than 7 days from the first time we saw the file, we will attempt to download it again after 7 days from when we first saw the file. A summary of the collected data is shown in Table 1.

	Limewire	OpenFT
Data collection days	45	37
Start date	4/1/06	4/9/06
Number of queries	34,268,803	12,347,509
Number of responses	32,788,921	30,538,152
Qualifying responses	2,468,327	381,851
Attempted downloads	228,722	22,231
Successful downloads	78,004	17,758
Unique clients	383,601	14,432

Table 1: Aggregate statistics. Qualifying responses are the responses with file extensions that we consider downloading. Attempted downloads are different from successful downloads because many downloads fail due to host non-availability (mostly because its IP address belongs to the private address range).

3. DATA ANALYSIS

3.1 Malware Prevalence and Commonality

As said in Table 1, we successfully downloaded $78,004$ distinct files for Limewire. These files correspond to $1,357,229$ responses, or 54.9% of the unique qualifying responses that contain executable, archival, or Microsoft Office file extensions. Of the downloaded files that are distinct, $27,717$, or 35.5% contained malware. The infected files correspond to $928,644$ responses, or approximately 37.6% of the total qualifying responses (68.4% of the responses for which we were able to download files).

92 different types of malware were found in Limewire[2]. Table 2 summarizes the top 10 malware found in Limewire, along with the number of responses each corresponded to. The top two malware correspond to 98.5% of all the qualifying downloadable responses.

Name	Files	Responses
Trojan.VB-100	19841	774216
Worm.Alcan.D	5978	140428
Worm.VB-16	334	5329
Worm.P2P.Poom.A	372	5120
Worm.SomeFool.P	83	2196
Trojan.Downloader.Istbar-176	331	818
Worm.VB-26	190	557
Trojan.JS.Startpage.C	212	447
Worm.Wupeer.A	159	182
Worm.P2P.Selmo.A	65	66

Table 2: Summary of top malware found in Limewire.

For OpenFT, $17,758$ distinct files were successfully downloaded. They corresponded to $211,604$ responses, or 55.4% of all qualifying responses. Of the downloaded files, 599, or 3.4% contain malware. The infected files correspond to $6,718$, or about 1.76% of the qualifying responses (approximately 3.2% of the qualifying responses for which we were able to download files). These numbers indicate that there is significantly less malware on OpenFT than on Limewire. The popularity of Limewire could contribute to this.

38 different types of malware were found in OpenFT. Table 3 summarizes the top 10 malware found in OpenFT data, along with the number of responses each corresponded to. The top two malware correspond to 74.85% of all the qualifying downloadable responses.

Name	Files	Responses
Worm.P2P.Poom.A	101	4512
Trojan.VB-100	168	512
Worm.Alcan.D	71	395
RAR	71	361
Trojan.Downloader.Istbar-176	51	206
Worm.SomeFool.P	24	149
DOS.HLLC.Slam.6000	1	114
Worm.SomeFool.Gen-1	12	107
Trojan.Downloader.Istbar-172	11	50
Trojan.Downloader.Delf-286	3	47

Table 3: Summary of top malware found in OpenFT.

Many of the malware observed in OpenFT are also observed in Limewire, including 8 of the top 10 (all except `Trojan.Downloader.Delf-286` and `RAR`). Of the top 10 in Limewire, 8 are also seen in OpenFT (all except `Worm.VB-26` and `Worm.P2P.Selmo.A`). In fact, 5 of the top 10 malware shown in Tables 2 and 3 are common, implying the co-existence of the same pieces of malware across different systems.

Although we downloaded files from a large list of extensions, we actually found only *zip* and *exe* files to be infected in large amounts. We also found a very small number of *rar* and *doc* files that were infected. The rest of the file formats did not contain any malware.

3.2 Malware Functionality

We find a large variety of malware in the files we downloaded for Limewire and OpenFT, with downloaders and worms being the most popular types of malware. Table 4 shows the functionality of different malware we found (the categorizations are done per information available at [1, 14, 15, 17]). Some malware embed more than one functionality, hence the percentages in Table 4 exceed 100%.

Function	Limewire	OpenFT
Downloader	45.16%	34.78%
Worm	40.32%	39.13%
Unknown	30.65%	30.43%
Backdoor	25.81%	17.39%
Adware	4.84%	8.70%
Dialer	4.84%	4.35%
Keylogger	3.23%	0.0%

Table 4: Functionality and percentage of malware in Limewire and OpenFT. The percentages do not add up to 100% due to malware programs with more than one function.

3.3 Malware Growth

Figure 1 shows the percentage of qualifying traffic we see each day on the two networks that is malicious. Since we can only identify malicious files from those we downloaded successfully, this

[2]Three more found by our re-download strategy, which we discuss in Section 3.4.

figure shows a lower bound on the extent of malware. A large percentage of Limewire responses are infested with malware. On the other hand, malware on OpenFT spikes only on one day, the day Poom.A worm hits.

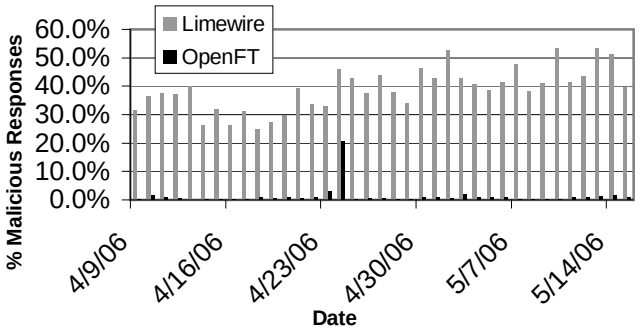

Figure 1: The percentage of qualifying responses per day on Limewire and OpenFT that were classified as malware. The spike in OpenFT corresponds to the Poom.A worm, which is shown in Figure 3.

Next, we investigate if the growth pattern of specific malware change over time. In general, we find that most malware follow similar growth trends during the span of our data. An example of this is depicted in Figure 2, which shows the prevalence of Alcan.D worm in Limewire. An exception to this growth trend is the Poom.A worm, which shows distinct spikes in the number of responses seen on different days in both OpenFT and Limewire. Figure 3 shows the prevalence of Poom.A over time in Limewire and OpenFT.

3.4 Effect of Re-downloads in Limewire

As mentioned in Section 2.3, we re-download "clean" files in Limewire after 7 days from when we first saw the file to ensure that we detect new malware which ClamAV [16] may not have had signatures created for when we first saw the files. We found 3 new malware in the 30 unique files that we re-downloaded. These are `Worm.VB-26`, `Trojan.Clicker.VB-20`, and `W32.Polipos.A`. This brings up the total count of distinct malware found in Limewire to 95.

3.5 Queries

We now analyze if certain queries fetch more malware than others. Tables 5 and 6 show the top 10 queries in Limewire and OpenFT respectively that fetch the most malware. It appears that while names of movies return the most malware in Limewire, such is not the case for OpenFT.

3.6 Host Characteristics

Table 7 depicts the top 10 IP addresses that serve the most malware in Limewire. Table 8 does the same for OpenFT. We see very different host characteristics across the two systems. In Limewire, a surprisingly high amount of malware is served from private IP addresses[3] while we do not see any private address in OpenFT. In Limewire, most of the top sources are serving more than one distinct malware. This is not the case in OpenFT. Also, all of the top

[3]Notice that this does not mean that the private IP addresses we mention are routable - we typically ended up downloading the file served by a host with a private IP address from another peer with a routable address.

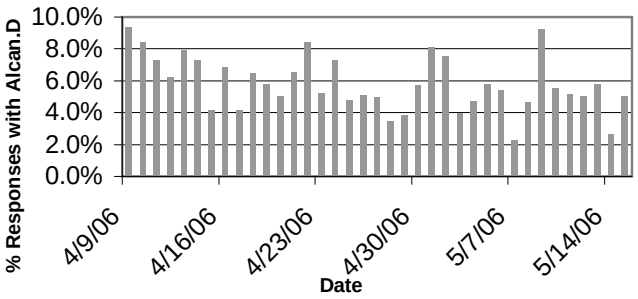

Figure 2: Distribution of Alcan.D as percent of responses observed over a period of 5 weeks on Limewire. Most malware follow a similar distribution.

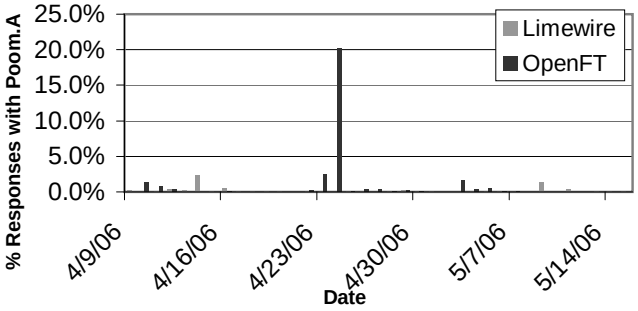

Figure 3: Number of responses infected each day from the Poom.A worm. We see a sudden increase in the number of infections on both Limewire and OpenFT, though on different days.

Query	Files	Responses
scary movie 4	64	19003
ice age 2	145	17020
2006	1706	12008
lost	237	10550
silent hill	65	10117
ice age	145	9388
sex	248	7600
prison break	78	6704
hostel	41	5571
nero	180	5406

Table 5: Top 10 malware returning queries in Limewire.

Query	Files	Responses
crack	94	872
adobe	16	422
sims	14	407
limewire	94	280
games	13	230
windows xp	7	222
macromedia	7	188
dreamweaver	3	165
zip	80	164
the sims	14	155

Table 6: Top 10 malware returning queries in OpenFT.

IP	Infected Files	Infected Responses	Malware ware	Clean Files	Clean Responses
192.168.1.11	3982	72418	4	231	419
65.34.187.196	4782	18257	2	27	63
192.168.1.100	7593	14664	10	3827	7787
192.168.1.2	5624	9393	12	4046	9551
192.168.1.101	4736	9219	8	1789	3543
192.168.1.47	1593	7851	2	1778	3275
192.168.1.3	4710	7418	9	1582	2449
192.168.0.10	3471	6421	2	434	631
85.167.159.53	3248	5823	1	4	7

Table 7: Top 10 locations serving malware in Limewire.

IP	Infected Files	Infected Responses	Malware ware	Clean Files	Clean Responses
24.185.43.12	101	4512	1	55	185
200.193.133.181	12	91	1	0	0
68.199.111.60	24	83	1	3	9
24.108.148.47	1	63	1	8	342
67.70.44.58	1	50	1	11	186
24.73.2.236	23	44	2	99	124
200.63.211.100	3	42	2	2	10
69.157.73.229	1	33	1	23	486
67.8.149.155	2	32	2	11	210
69.253.47.181	1	30	1	0	0

Table 8: Top 10 locations serving malware in OpenFT.

malware sources in Limewire are also serving good files, although in smaller numbers in most cases. In OpenFT, most are serving good files as well, but two are not.

In OpenFT, none of the top 10 sources of malicious files is also one of the top 10 sources of good downloads. In Limewire several of the top malicious sources are also top good sources, although all of these are private IP addresses. Note that the cases of private IP addresses, NAT and DHCP effect make it impossible to determine if the files are really coming from the same host.

A total of only 1.5% of all responses in Limewire come from private IP addresses when they are responsible for 20% of our failed downloads. They are also responsible for a disproportionately high amount of the malicious responses: 28%. The top source of infected responses in Limewire, 192.168.1.11 is a private address and is responsible for 7.8% of infected replies we see. The top offender in OpenFT, 24.185.43.12 is responsible for 67.2% of infected replies in this system. This address is registered to a New York (USA) based cable Internet provider.

Further, in Limewire, we also see a difference between how many hosts are serving individual infected files vs how many hosts are serving individual clean files. On average, we see 22.9 distinct IP addresses serving each infected file, but only 6.2 for clean files. Interestingly, we see somewhat of the opposite effect in OpenFT, with 3.8 hosts serving each clean file but only 1.5 for infected files.

4. FILTERING MALWARE

4.1 Filtering in Limewire

While OpenFT does not attempt to filter malware, Limewire has a built in capability to flag some query responses as suspicious. Such responses are not shown to the user. It bases this on three criteria: 1) the file name or metadata does not match the query; 2) the extension of the returned file is not considered by Limewire to match the file type asked for; or 3) Limewire believes the response contains the Mandragore worm[4].

In all of query responses we have seen, only 22, all with the filename "control.exe," have been classified by Limewire as the Mandragore worm. No malware is detected in this file by ClamAV. It is likely that this check is outdated and this file is legitimate.

Limewire determined that 3.5% of responses contained malware because they did not match the query sent. This check produces a very high false positive: 15.6%! Similarly, 3.0% of responses containing malware were determined by Limewire not to match the requested file type. The false positive rate for this check was 3.5%. We conclude that filtering checks in Limewire are inadequate, producing nearly 17% false positives.

4.2 File Size-based Filtering

We now perform a preliminary investigation of a simpler and more effective filtering mechanism for P2P networks based on file sizes. We first look into the names and sizes of the files containing malware. As shown in Table 9, much of the malware is seen at only a few distinct file sizes, with a very large proportion in a single size. In fact, much of the top malware occurs in its top size over 90% of the times it is seen. This seems to indicate that the size of a file may be a good indicator of malware. Thus we hypothesize that *adding filters to discard responses whose size match that of known malware can be used as a simple and effective heuristic to block malware on P2P systems*.

	Limewire		OpenFT	
Name	**Sizes**	**% Largest**	**Sizes**	**% Largest**
VB-100	7	99.95%	3	99.22%
Alcan.D	6	99.85%	2	96.29%
VB-16	2	99.98%	1	100%
Poom.A	34	93.13%	6	99.67%
Somefool.P	6	99.09%	1	100%

Table 9: Number of file sizes and the percent of responses containing each malware at the most common file size in both Limewire and OpenFT (for top 5 malware).

We now test this heuristic for false positives. Table 10 shows the percentage of blocked files at each size which would have been false positives, if this filter was in place. We assume that the most frequently occurring size of the malware is used in the filter. We conclude the percentage of false positives is negligible in most cases, with the exception of `VB-16` in OpenFT. While a deeper analysis of the heuristic is needed before any firm conclusion can be drawn, it is important to note that `VB-16` shows up only four times in OpenFT.

5. LIMITATIONS AND FUTURE WORK

This paper presented an initial look at malware in P2P systems and how to filter responses containing malware. The current study has several limitations: 1) a relatively short duration of the study; 2) it was possible only to observe malware returned in the search queries and the distribution of malware may be different in downloaded files and across all the files available for download; 3) mal-

[4]Limewire determines if a response is the Mandragore worm by checking if the file size is 8192 bytes and the file name is the same as the query string with the addition of *.exe*.

Name	Limewire			OpenFT		
	Total Responses	**False Positives**	**%**	**Total Responses**	**False Positives**	**%**
VB-100	776666	796	0.10	510	2	0.39
Alcan.D	140832	98	0.07	385	5	1.30
VB-16	5330	2	0.04	29	25	86.2
Poom.A	4768	0	0	4497	0	0
Somefool.P	2176	0	0	149	0	0

Table 10: False positives for our size based filter for top 5 malware. For each malware, the most common occurring file size is used in the filter.

ware could be present in file formats other than executable, archival, and Microsoft Office; 4) ClamAV [16] may fail to detect some malware; and 5) it was not possible to observe malware in other popular P2P networks, which are either not decentralized or are not open source. It is possible to eliminate several of these limitations in favor of drawing more comprehensive conclusions. We plan to undertake a more detailed study in the future.

We believe the simple size based filtering we propose is a significant step in filtering malware in P2P networks. It has the additional advantage that it allows filtering to be done before any download is performed. However, an aspect of this filtering technique remains to be investigated: When should the size-based filtering rules be updated to avoid false positives due to outdated malware? A knowledge of malware life-cycle is necessary to propose a reasonable heuristic for this. Unfortunately, our data duration is not long enough to propose such a study. We plan to look into this issue with on-going data collection.

Acknowledgements

We would like to thank Rob Henderson for support in data collection. Michel Salim and Nikhil Balchandani worked on the class project that led to this paper. Their input is appreciated.

6. REFERENCES

[1] Computer associates virus information center. `http://www3.ca.com/securityadvisor/virusinfo`.

[2] The giFT project homepage. `http://gift.sourceforge.net`.

[3] Gnutella protocol specification. `http://www.the-gdf.org/wiki/index.php?title=Gnutella_Protocol_Development`.

[4] K. Gummadi, R. Dunn, S. Saroiu, S. Gribble, H. Levy, and J. Zahorjan. Measurement, modeling, and analysis of peer-to-peer file-sharing workload. In *ACM SOSP*, 2003.

[5] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang. Measurement, analysis, and modeling of bittorrent-like systems. In *ACM SIGCOMM Internet Measurement Conference (IMC)*, 2005.

[6] T. Holgers, D. Watson, and S. Gribble. Cutting through the confusion: A measurement study of homograph attacks. In *USENIX Annual Technical Conference (USENIX)*, 2006.

[7] J. Liang, R. Kumar, and K. W. Ross. The fasttrack overlay: A measurement study. *Computer Networks*, 50(6):842–858, 2006.

[8] LimeWire homepage. `http://www.limewire.org`.

[9] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the slammer worm. In *IEEE Security and Privacy*, 2003.

[10] D. Moore, C. Shannon, and J. Brown. Code red: A case study on the spread and victims of an internet worm. In *ACM/USENIX IMW*, 2002.

[11] A. Moshchuk, T. Bragin, S. Gribble, and H. Levy. A crawler-based study of spyware on the web. In *Internet Society Network and Distributed System Security Symposium (NDSS)*, Feb. 2006.

[12] S. Saroiu, S. Gribble, and H. Levy. Measurement and analysis of spyware in a university environment. In *USENIX Networked Sytems Design and Implementation (NSDI)*, 30 Mar. 2004.

[13] S. Shin, J. Jung, and H. Balakrishnan. Malware prevalence in the kazaa file-sharing network. In *ACM SIGCOMM Internet Measurement Conference (IMC)*, 2006.

[14] Sophos virus analyses. `http://www.sophos.com/virusinfo/analyses`.

[15] Symantec security response. `http://www.symantec.com/avcenter`.

[16] Tomasz Kojm. ClamAV homepage. `http://www.clamav.net`.

[17] Kapersky lab virus encyclopedia. `http://www.viruslist.com/en/viruses/encycolpedia`.

[18] V. Yegneswaran, P. Barford, and J. Ullrich. Internet intrusions: Global characteristics and prevalence. In *ACM SIGMETRICS*, 2003.

[19] W. Yu. Analyze the worm-based attack in large scale p2p networks. *8th IEEE International Symposium on High-Assurace Systems Engineering*, pages 308–309, Mar. 2004.

[20] L. Zhou, L. Zhang, F. McSherry, N. Immorlica, M. Costa, and S. Chien. A first look at peer-to-peer worms: Threats and devenses. In *Proceedings of the IPTPS*, Feb. 2005.