Malware Prevalence in the KaZaA File-Sharing Network

Seungwon Shin
ETRI
swshin@etri.re.kr

Jaeyeon Jung
Mazu Networks
jyjung@mazunetworks.com

Hari Balakrishnan
MIT CSAIL
hari@csail.mit.edu

ABSTRACT

In recent years, more than 200 viruses have been reported to use a peer-to-peer (P2P) file-sharing network as a propagation vector. Disguised as files that are frequently exchanged over P2P networks, these malicious programs infect the user's host if downloaded and opened, leaving their copies in the user's sharing folder for further propagation. Using a light-weight crawler built for the KaZaA file-sharing network, we study the prevalence of malware in this popular P2P network, the malware's propagation behavior in the P2P network environment and the characteristics of infected hosts. We gathered information about more than 500,000 files returned by the KaZaA network in response to 24 common query strings. With 364 signatures of known malicious programs, we found that over 15% of the crawled files were infected by 52 different viruses. Many of the malicious programs that we find active in the KaZaA P2P network open a backdoor through which an attacker can remotely control the compromised machine, send spam, or steal a user's confidential information. The assertion that these hosts were used to send spam was supported by the fact that over 70% of infected hosts were listed on DNS-based spam black-lists. Our measurement method is efficient: it enables us to investigate more than 30,000 files in an hour, identifying infected hosts without directly accessing their file system.

Categories and Subject Descriptors

C.2.0 [**COMPUTER-COMMUNICATION NETWORKS**]: General

Keywords

Peer-to-peer, KaZaA, Virus Prevalence

General Terms

Measurement, Security

1. INTRODUCTION

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

With few defense mechanisms in place, peer-to-peer (P2P) file-sharing networks have been known to be vulnerable to many security attacks. Recent papers discuss the threat of malware spread in P2P networks [30, 26]. One experimental study reports that 44% of the 4,778 executable files downloaded through a KaZaA client application contain malicious code [28]. In this study, we perform a large scale measurement study on the prevalence of malicious programs in the KaZaA file-sharing network.

In recent years, KaZaA has been one of most popular P2P networks and the number of active users far outnumbers that of Overnet and Gnutella. Despite a number of lawsuits [6], KaZaA still attracted more than 2.5 millions users in May 2006 [16]. However, little has been known about details the operation of the KaZaA network and the characteristics of the shared files in the network. Liang et al. present the dynamics of KaZaA overlay structure by analyzing KaZaA network traffic and the behavior of a KaZaA client application running on their machine [10]. They also examined the impact of corrupted files intentionally injected by a few hosts ("pollution servers") to hinder file sharing practices [11, 7].

Because of the distributed nature of peer-to-peer clients, a crawler-based measurement approach is commonly used to study the characteristics of client hosts and the files that are shared among them. Using a crawler, Liang et al. [11, 7] collected data from KaZaA, Stutzbach et al. [17, 18, 29] from Gnutella, and Fessant et al. [3] from eDonkey. However, since there is no publicly available crawling software for KaZaA, we develop our own crawling software that quickly visits distributed "indexing" hosts in the KaZaA network and gathers information about active client hosts and the files they share. The architecture of our KaZaA crawler, Krawler, is similar to that of the KaZaA Crawling Platform by Liang et al. [11]. However, Krawler is optimized for processing a large number of query strings, whereas the KaZaA Crawling Platform is used for visiting a large number of indexing hosts with a small set of query strings.

Using 71 different malicious programs (e.g., viruses, Trojan horses), we construct 364 different signatures, with which we can identify infected files only using the information that KaZaA application provides. In analyzing the crawled data in combination with the signature-based detection method, we make the following contributions:

- We find that over 12% of KaZaA client hosts are infected by over 40 different viruses both in February and in May, 2006. The number of viral files that these infected client hosts bring into the KaZaA P2P network is significant. Viral files account for more than 22% and 15% of the toal crawled data in February and May 2006, respectively.
- We find that prevailing viruses in a P2P network exhibit two characteristics: (1) a virus multiplies itself into many copies

with different names corresponding to popular executable files. As a result, certain query strings such as "ICQ" and "Trillian" elicit more infected files than legitimate ones in the KaZaA P2P file-sharing network. (2) a virus generates many binary variants to avoid detection. We find that both factors appear necessary for wide spreading.

- We observe that 4.8% of infected client hosts seen in February returned still infected in May 2006. Moreover, 70% of infected client hosts are listed in one or more of six DNS-based blacklists that we tested. We believe that this is a strong indicator that these infected client hosts are used for relaying spam email. More importantly, some client hosts are infected with viruses that steal a user's confidential information (Darby) and turn the victim machine into a *bot* (IRCBot).

The paper is organized as follows: §2 presents an overview of KaZaA, our crawling software and the malware propagation in the KaZaA file-sharing network. §3 presents our data collection methods. §4 analyzes two datasets collected in February and May, 2006. §5 summarizes the paper with a discussion of future work.

2. BACKGROUND

The KaZaA network uses the proprietary FastTrack protocol [25] whose technical details are not publicly available. As such, this overview comes from the examination of related documents as well as our experience in reverse engineering its operation[1] and analyzing the giFT source code [5], giFT is an open source project that attempts to support multiple peer-to-peer networks including KaZaA. We then describe our crawling software that quickly visits "indexing" hosts in the KaZaA network and collects information about files that KaZaA hosts offer to share.

2.1 A Basic KaZaA Operation

When a user runs a KaZaA client application, the client establishes a connection with an "indexing" hosts, called supernodes [9]. The client has a hard-coded list of possible supernodes. These supernodes form an overlay network with other supernodes and propagate queries received from their client hosts. Any client host may server as a supernode if it is accessible from the Internet (i.e., not behind a firewall or a NAT box) and is connected with a fast enough link [10, 9].

When a client connects to a supernode, it sends two types of information to the supernode [10]. The first is the list of files that the peering client host has in the sharing folder [2] [4]. A supernode creates and updates a search index using the information received from client hosts. Second, the client informs the peering supernode of the host's information, such as a client nickname, a (download) port number, and an IP address at which other clients can request a file download.

A user can locate a file by issuing a query that contain a substring of the name of the file of interest. Each query is sent to a directly connected supernode, which is called a *parent supernode* [10]. Figure 1 illustrates an example. A user who is looking for an ICQ chatting client sends a query "ICQ" to the parent supernode. If the parent supernode is unable to find a match for the query, it forwards the query to other peering supernodes and the search continues until the TTL reaches zero [12]. If the parent supernode receives an answer from a peer, it forwards the answer back to the original client. Typically, an answer is a list of the IP addresses and source ports of the hosts offering a matching file, as well as the information about the files themselves. Finally, the client directly connects to one of the hosts listed in the answer and downloads the "ICQ.exe" file. This signaling traffic between a client and a parent supernode is encrypted with a key that is exchanged in the beginning of a session.

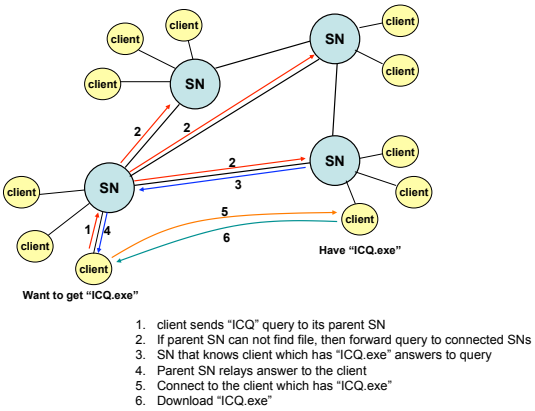

Figure 1: Example of a KaZaA search sequence

2.2 Krawler: A KaZaA Crawler

This section describes the operation of our KaZaA crawling software, Krawler, that queries tens of thousands of active supernodes in the KaZaA network. The Krawler has two main components; The *dispatcher* maintains a list of supernodes. The *fetcher* communicates with the dispatcher, updates a set of supernodes to crawl, sends query strings to individual supernodes and processes responses as follows:

- Krawler starts with a set of the IP addresses of 200 known supernodes in the KaZaA network and a set of query strings associated with the files that we seek.
- Krawler attempts to make a TCP connection to each supernode in the list. If it fails to establish a TCP connection, then it waits until the next round to get a new IP address. If it succeeds, it exchanges handshake messages with the supernode.
- Krawler receives from the supernode a *supernode refresh list*, consisting of up to 200 super node IP addresses. This list is stored in the *dispatcher*.
- A *fetcher* sends out a set of query strings to each supernode and waits for responses. A response includes the metadata and content hash for the files that match the query string and the peer hosts' IP address and download port number. If there is no response from a particular supernode for 60 seconds, the fetcher drops the corresponding query session.[3] All these responses are logged for further analysis.

2.3 Virus Propagation in P2P Networks

Over the past few years, more than 200 viruses and worms have been reported to employ a peer-to-peer network as one of their

[1]We ran various KaZaA clients, such as KaZaA [8], Poisoned [14], and KaZaA Lite [27], and analyzed packets generated from these clients using Ethereal [1].

[2]A KaZaA client also scans the sharing folders every 6 minutes to check for added or deleted files [21].

[3]We polled a set of 100 randomly selected supernodes at every 5 seconds for their availability. 100% of supernodes that did not respond for the first 30 seconds never responded for the entire polling duration.

spreading platforms [2]. Unlike self-propagating network worms such as Code Red [19] and Slammer [20], most malicious programs in a P2P file-sharing network do not send their copies in the network by themselves. Instead, these viruses[4] propagate to other client hosts as these clients engage in file exchanges.

One characteristic of P2P viruses is that they tend to generate a large number of viral files in the user's sharing folder upon infection. Each viral file has a different filename that is likely to be popular and thus have a high chance of getting downloaded by other clients. Examples of the filenames often chosen by P2P viruses include "Adobe Photoshop 10 full.exe", "WinZip 8.1.exe", and "ICQ Lite (new).exe", all of which may appear legitimate to an unwary user.

3. DATA COLLECTION

We used three machines to run Krawler: one machine (2.1 GHz dual-core CPU and 1 GByte RAM) was dedicated for a fetcher. The other two machines (2.1 GHz CPU and 1.5 GBbyte RAM and 1.42 GHz CPU and 1 GByte RAM) ran 20 and 10 fetchers, respectively.[5] Although using more fetchers could speed up crawling, the CPU and bandwidth constraints kept us from running more than 30 fetchers at a time. With the above configuration, Krawler was capable of investigating more than 60,000 files in an hour on average. In this section, we describe two measurement methods that enable us to (1) collect a large number of popular executable files distributed in the KaZaA network and (2) identify malicious programs in the set of collected files.

3.1 Query Strings

Ideally, an extensive analysis would be possible if we could look up the complete information about files stored in each client's KaZaA sharing folder. However, to our knowledge, there is no way to extract such information from a supernode[6]. When queried, a supernode returns only the information about files whose name (or metadata information) contains the query string. We observe that this query process often takes long (several seconds and longer) when more than 10 query strings are sent at once. As a result, many studies rely on a small set of files that match a particular set of query strings [11, 7].

As discussed in §2.3, many viruses propagate in a P2P network by tricking users into downloading an infected file and executing it. To increase the chances of being queried, some viruses generate multiple copies of a viral program, giving each copy a different name (e.g., ICQ Lite .exe, ICQ Pro 2003b.exe, MSN Messenger 5.2.exe), and placing them in the user's sharing folder. Therefore, in this study, we investigate only `.exe` files and not `.mp3` and other media files that are rarely exploited by known viruses.

Table 1 shows a set of 24 keywords that we compiled from the filenames of the 30 most popular files listed at www.download.com. The Web site offers reviews and free downloads of shareware and freeware and attracts a large number of users: a popular file sometimes exceeds a million downloads per week. The files that were listed in February 23, 2006, include spyware and adware detection software (e.g., Spyware Doctor) , Windows utilities (e.g., WinZip), and P2P clients (e.g., Morpheus and BitComet). Since there are no statistics available about file popularity in the KaZaA network, we use these files as proxies for popular files.

[4]We use the term, "virus", as opposed to "worm", to refer to a malicious program spreading in a P2P network since it requires human intervention.

[5]In crawling in February, 2006, we used only one machine for 20 fetchers, but we add another machine later.

[6]Recall that each client periodically uploads the information about the files in the sharing folder to the parent supernode.

Ad, Spyware, LimeWire, ICQ, Registry, SpyBot, WinZip, Morpheus, All, iMesh, IrfanView, WinRar, DivX, BitComet, RealPlayer, PC, Adobe, Trillian, Camfrog, SmartFTP, Nero, MSN, Quick, Knight

Table 1: Query strings used for data collection

3.2 Virus Signatures

We compiled a list of malicious programs that use P2P a propagation vector from many security vendor Web sites (F-secure, McAfee, Sophos). Since 2002, more than 200 such programs have been that were identified by those vendors [2, 23]. Among these, we have the content hashes of 71 distinct malicious programs.[7]

We used the Sig2Dat [24] tool to get the content hash of each malicious program. The KaZaA content hash is 20 bytes in size: the first 16 bytes are the MD5 [15] of the first 300 Kbyte of the file. The last 4 bytes are the value of the custom made hash function of the length of the file. In our study, only the first 16 bytes are used for identifying a known virus because many viruses change their size by appending an arbitrary number of bytes. Table 2 shows a break-down of these malicious programs by the propagation vector.

Propagation	*Virus List*
P2P only	Apsiv, Darker, Doep, Duload, HLLP.Hantaner, Logpole, PMX, SdDrop and variants (2), Sndc, Steph, Tanked and variants (4), Theug, Kwbot and variant, Archar.a, Bare.a, Benjamin.a, Wif, Gotorm, Harex.a, Harex.b, Harex.c, Kazmor.a, Lolol.a, Spear.a, Parite, Togod
P2P + email	Bagle variants (9), Darby, Kindal, Mapson-A, Ronoper
P2P + messenger	Bropia, SdBot, Supova and variants (4)
P2P + backdoor	SpyBot and variant
P2P + email + IRC	Swen
Mail only	Bagle, MyDoom, NetSky, Yosenio, Stator
Etc	IRCBot and variant (IRC), Tenga (RPC), Hidrag, HLLP.19920, Agent.Gen, Cryptexe, Delf and variant, Dropper (Human)

Table 2: Malware breakdown by the propagation vector

4. RESULTS

This section presents the analysis results of two datasets, `feb-06` and `may-06`, each of which contains crawling logs for three days in February and May 2006, respectively.[8] Table 3 summarizes each dataset in terms of the number of responses that match the query strings, the number of unique supernodes these responses came from, and the number of unique KaZaA client hosts that responses are associated with.

There can be duplicate responses included in the dataset because (1) Krawler may have contacted the same supernode multiple times

[7]Actually, we have content hashes of 364 malicious programs, but 71 programs are totally different types of viruses and remaining 293 programs are variants of them.

[8]We have more datasets that are collected in April and other dates in May 2006, but do not present their results here as in general they agree with the results presented in this section.

	feb-06	may-06
date	February 23, 2006	May 4, 2006
responses	654,254	532,610
supernodes	10,267	15,522
client hosts	19,919	28,601

Table 3: Summary of the datasets: both datasets used the set of query strings in Table 1 for crawling. Krawler issues 24 queries to each supernode (in series) and gathers responses that may come from any peering supernodes. One query may generate multiple responses from a given supernode.

during the crawling[9] and (2) the way a query is propagated among peering supernodes (see §2.1) makes it possible that Krawler receives the same response multiple times. However, we note that 174,971 (32.9%) responses from `may-06` (15.3% from `feb-06`) correspond to unique content hashes, suggesting that our crawling covered a large number of distinct files in the KaZaA network.

We observe that many responses associate with client hosts behind a network address translation (NAT) box (68% in `feb-06` and 52% in `may-06`). To differentiate between hosts behind two different NATs, we gave a client host a unique ID based on the IP address and the download port number. Because a download port number is often randomly assigned, the chance of two machines having the same ID is low: in fact, we see more than 3,000 different download ports are used in both datasets. We refer to an IP address and a download port pair as a client host unless otherwise stated.

4.1 Malware Distribution

Among the 71 malicious programs for which we have content hashes, we find instances of 45 viruses in `feb-06` and 52 viruses in `may-06`. Only one out 45 viruses (MyDoom) was seen in `feb-06` but not in `may-06`. New viruses that appeared in `may-06` include Darby, SdBot, and Duload. However, given the small number of clients infected by these viruses (less than 4) in the datasets, we believe that these new additions or deletions of viruses are due to host churn (the arrival and departure of client hosts). Figure 2 compares the percentages of infected clients by the set of top 10 viruses seen in each dataset. Interestingly, the percentages remain almost unchanged between the two datasets. The Tanked virus had infected more client hosts by 0.4% (1.7% in `feb-06` vs. 2.1% in `may-06`), but the difference is still small. Overall, the percentage of infected hosts is 12.6% in `feb-06` and 13.0% in `may-06`.

The number viral files that these infected client hosts bring into the KaZaA P2P network is significant. Among the responses, 22.9% in `feb-06` and 15.2% in `may-06` appear infected. More importantly, we find that 22 out of 24 query strings that we used for crawling returned at least one viral file: the responses associated with the "DivX" query string account for 23% of the entire viral files collected in `feb-06` (17.6% in `may-06`). The percentage of viral files is over 20% for the half of the query strings. Depending on the size of the pool of legitimate files that are already placed in a file sharing network, the probability of downloading an infected file varies for each query string (or keyword). As shown in Figure 3, for certain or keywords such as "ICQ" and "Trillian", the chances

[9]We see that the number of new supernodes that Krawler visited sharply increases in the first three hours, but as new hosts arrive, previously unseen supernodes constantly appear (more than 100 an hour) over the entire collection period.

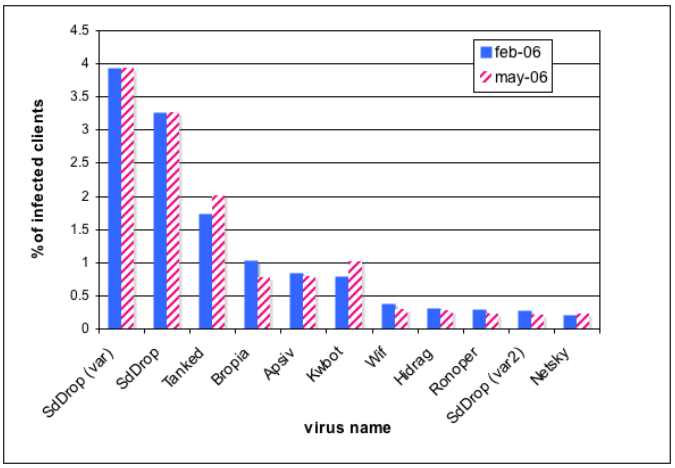

Figure 2: Top 10 viruses: This graph compares the percentages of infected client hosts to the union of top 10 viruses found in `feb-06` and `may-06`. The X-axis is virus names sorted by the percentage of the infected clients in `feb-06`. Netsky, which ranked 11th in `feb-06`, is included as it ranked 8th in `may-06`.

of hitting an infected file is over 70%!

We find that both keywords ("ICQ" and "Trillian") are exploited by over 10 different viruses. Clearly, viruses favor popular file names to disguise themselves. For instance, about 20 different viruses are found from the crawled files in response to "DivX", "WinZip", or "Adobe". In the next section, we discuss other tactics that P2P viruses employ in addition to choosing a popular file name.

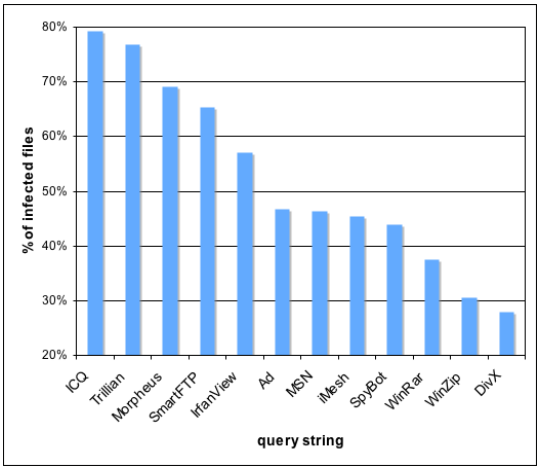

Figure 3: Query strings vs. % of infected files for the `may-06` dataset

4.2 Virus Propagation Behavior in P2P Network

We note that some viruses seen in the datasets are not known to be P2P viruses. For instance, Hidrag [22] is a Win32 virus that infects `.exe` files in the victim's logical drives. Therefore, if the victim had executable fies in the KaZaA sharing folder, the virus could then spread into the KaZaA P2P network. In principle, any

virus can propagate in a P2P network as long as it places an infected binary in a folder used for file sharing. Here, we discuss a few factors that affect the propensity of a virus to spread in a P2P network.

As shown in Figure 2, the top 3 viruses in terms of the percentage of infected clients are the ones that are crafted to efficiently spread in a P2P network. On a closer examination, we find that (1) these viruses produce copies in many different files names associated with over 10 query strings, which we call an aliasing factor and (2) they generate many binary variants (62 Tanked variants, 14 SdDrop variants) to render inefficient a simple signature-based detection.

To determine a relationship of each factor, x, with the spreadability, y, we compute a coefficient coefficient, r, defined as

$$r = \frac{N\sum xy - \sum x \sum y}{\sqrt{(N\sum x^2 - (\sum x)^2)(N\sum y^2 - (\sum y)^2)}}$$

where N is the number of sample points (97, the number of viruses found from both datasets). It appears that both factors have a positive correlation with the degree of propagation. Figure 4 is a scatter plot showing a strong positive correlation, $r = 0.74$, between the percentage of infected clients (spreadability) and the combination of both factors.[10]

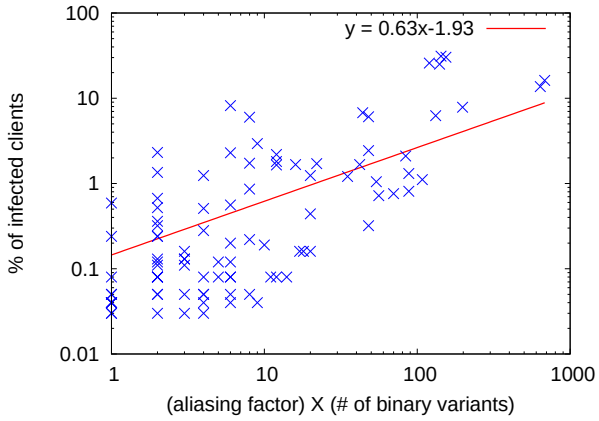

Figure 4: A dot corresponds to each virus found in `feb-06` and in `may-06`. The x-axis is the multiplication of an aliasing factor and the number of binary variants found from each virus. The y-axis is the percentage of infected clients. The solid line represents the result of the linear regression to the data.

4.3 Characteristics of Infected Hosts

Once compromised, a host can be used for such nefarious activities as DoS attacks, spam relaying, and botnet command and control [13]. Table 4 categorizes the viruses found in the `may-06` dataset by the employed attack method if such information is available.

In the `feb-06` dataset, we observe that among the KaZaA client hosts that are not behind a NAT box, 1,618 hosts contain an infected file. Interestingly, we see 79 of these infected client hosts in the `may-06` dataset as well. Except the one host that no longer has the infected file that we detected in February, 78 of them (4.8%) appear still to be infected in May. This result suggests that these users are either unaware that their machines were infected for months or that their machines have been reinfected.

To check whether these infected hosts may have been used for relaying spam email, we check the IP addresses of the infectees against six DNS-based black lists (DNSBLs), `bl.spamcop.net`, `cbl.abuseat.org`, `dnsbl.sorbs.net`, `list.dsbl.org`, `opm.blitzed.org`, `sbl.spamhaus.org`. As shown in Table 5, over 70% of infectees are listed in one or more of DNSBLs, which is a strong indicator that hosts in question may have engaged in sending out spam email.

[10] As the number of binary variants goes up to several hundreads for certain virus, we computed a correlation coefficient between $\log(x)$ and $\log(y)$.

Attack	*Virus list*
Backdoor	Sndc, Tanked, Kwbot, Bagle, Darby, SdBot, SpyBot, Swen, IRCBot, Agent.Gen, Delf, Dropper
Spam (email)	Bagle, Darby, Mapson-A, Ronoper, Swen, NetSky
Spam (messenger)	Bropia, Supova
DDoS	Darby, Kindal, SdBot
Information stealing	Darby, SdBot

Table 4: Known attack methods by the viruses found in `may-06`

	`feb-06`	`may-06`
infected	1,618	2,576
listed in DNSBLs	1,146 (70.83%)	1,825 (70.85%)

Table 5: KaZaA client host statistics: In generating these statistics, we exclude hosts in private address space, 10/8, 172.16/12, and 192.168/16 since their information is unavailable in DNSBLs.

5. SUMMARY AND FUTURE WORK

The work in this paper was motivated by a dramatic rise in network viruses and other malicious programs that propagate over P2P networks. Using a crawler developed for the KaZaA file-sharing network, we have collected data in February and May 2006. The analysis results show that about 15% of sampled executable files contain a viral code. We have found 52 different viruses that are active in the KaZaA network in May, 2006. SdDrop and its variants and Tanked viruses are more prevalent than other viruses, collectively resulting in 71% of the total infected clients.

Our future work includes better understanding the distribution of the malware in a P2P network, tracking the change of prevalent viruses, and developing a model to explain the propagation behavior of a virus in a P2P network. In parallel with more measurement efforts, we are currently developing a crawling-based malware detection system that automatically identifies infected executables in a P2P network.

Acknowledgments

We gratefully acknowledge funding from Electronics & Telecommunications Research Institute (ETRI), Korea.

6. REFERENCES

[1] Ehtereal.com. Ethereal. http://www.ethereal.com/.

[2] FaceTime Security Labs. IM and P2P Threats. http://www.facetime.com/securitylabs/imp2pthreats.aspx.

[3] F. Le Fessant, S. Handurukande, A.-M. Kermarrec, and L. Massouli. Clustering in Peer-to-Peer File Sharing Workloads. In *Proceedings of Peer-to-Peer Systems, 3th International Workshop (IPTPS)*, February 2004.

[4] FreJon. FastTrack File Format. http://members.home.nl/frejon/55/ft/KazaaFileFormats.html#dbbformat.

[5] giFT Project. gift: Internet File Transfer. http://gift.sourceforge.net/.

[6] Iain Ferguson. Sharman Cuts Off KaZaA Downloads in Australia. CNET News.com http://news.com.com/Sharman+cuts+off+Kazaa+downloads+in+Australia/2100-%1027_3-5983455.html.

[7] N. Naoumov Jian Liang and Keith W. Ross. The Index Poisoning Attack in P2P File-Sharing Systems. In *Proceedings of INFOCOM 2006*, April 2006.

[8] KaZaA.com. KaZaA. http://www.kazaa.com/us/index.htm.

[9] KaZaA.com. KaZaA SuperNode. http://www.kazaa.com/us/help/faq/supernodes.htm.

[10] Jian Liang, Rakesh Kumar, and Keith W Ross. The FastTrack overlay: A measurement Study. In *Computer Networks*, pages 842–858, August 2005.

[11] Jian Liang, Rakesh Kumar, Yongjian Xi, and Keith W Ross. Pollution in P2P File Sharing Systems. In *Proceedings of INFOCOM 2005*, March 2005.

[12] Bryn Loban. Between rhizomes and trees: P2P information systems. In *First Monday Peer-Reviewed Journal on the Internet*, September 2004.

[13] Martin Overton. Bots and Botnets: Risks, Issues and Prevention. In *Proceedings of the 15th Virus Bulletin Conference*, 2005.

[14] Posined Project. Poisoned. http://gottsilla.net/poisoned.php.

[15] Ron Rivest. *The MD5 Message-Digest Algorithm*. Internet Engineering Task Force, April 1992. RFC 1321.

[16] Slyck.com. Slyck's P2P Network Stats Page. http://slyck.com/stats.php.

[17] Daniel Stutzbach and Reza Rejaie. Capturing accurate snapshots of the gnutella network. In *Proceedings of 8th IEEE Global Internet Symposium*, pages 127–132, March 2005.

[18] Daniel Stutzbach, Reza Rejaie, and Subhabrata Sen. Characterizing unstructured overlay topologies in modern P2P file-sharing systems. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference*, October 2005.

[19] Symantec Security Response. CodeRed. http://www.symantec.com/avcenter/venc/data/codered.worm.html.

[20] Symantec Security Response. Slammmer. http://www.symantec.com/avcenter/venc/data/w32.sqlexp.worm.html.

[21] University of Delaware Police Computer Forensic Lab. DBB KaZaA Share File. http://128.175.24.251/forensics/lastsharedate.htm.

[22] Viruslist.com. Hidrag. http://www.viruslist.com/viruses/encyclopedia?virusid=20627.

[23] Viruslist.com. Peer-to-Peer Worms. http://www.viruslist.com/en/virusesdescribed?chapter=153311928.

[24] VLAIBB. Sig2Dat. http://www.geocities.com/vlaibb/.

[25] Wikipedia. FastTrack. http://en.wikipedia.org/wiki/FastTrack.

[26] Wei Yu. Analyze the Worm-Based Attack in Large Scale P2P. In *8th IEEE International Symposium on High-Assurance Systems Engineering (HASE)*, pages 308–309, March 2004.

[27] Zeropaid.com. KaZaA Lite. http://www.zeropaid.com/kazaalite/.

[28] Kim Zetter. KaZaA Delivers More Than Tunes. *The Wired Magazine*, January 2004.

[29] Shanyu Zhao, Daniel Stutzbach, and Reza Rejaie. Characterizing files in the modern gnutella network: A measurement study. In *Proceedings of SPIE/ACM Multimedia Computing and Networking*, January 2006.

[30] Lidong Zhou, Lintao Zhang, Frank McSherry, Nicole Immorlica, Manuel Costa, and Steve Chien. A First Look at Peer-to-Peer Worms: Threats and Defenses. In *Proceedings of the IPTPS*, February 2005.