ISSN:0975-9646

Analysis and Prevention of Malware in P2P

Prof.Puram.Pradeep Kumar, Naini Shekar Reddy, Saleha Saudagar, T. Puneeth Chandra, Ch. Kishor Kumar

VITS Engineering College, KarimNagar.

Abstract-Peer-to-Peer (P2P) Networks continue to be popular means of trading content. However the files exchanged in these networks are not malicious, making them an ideal medium for spreading Malware. Some existing studies have shown that Malware proliferation can pose significant threats to P2P Networks, defending against such an attack are largely an open problem. This paper aims to develop the countermeasure that can effectively mitigate the Malware proliferation while preserving P2P networks performance and provide software implementations for P2P.Malware is highly pervasive in P2P file-sharing systems and is difficult to detect. To alleviate this problem, we analyze and provide preventive measures for Malware. Analysis include two types for detecting Malware and provide two basic approaches and Advanced techniques for preventing Malware. We instrument two different open source P2P networks; KaZaA and IMesh to examine the prevalence of Malware in P2P networks and provide an algorithm for detecting Malware.

Keywords: P2P Networks, Malware, KaZaA, IMesh.

1 INTRODUCTION

A P2P computer Network[1] is a Network that relies primarily on the computing power and bandwidth of the participants in the Network rather than concentrating it in a relatively low number of servers. Although this statement is mostly correct there are a few different types of P2P architectures which should be outlined:

1.1 Centralized architecture - Requires a centralized server which hosts connect with in order to access a list of shared items. Each host provides a list of items they are willing to share. The server maintains this list of shareable items from all hosts. The actual download itself is performed between the hosts when an item is requested, not by the server.

1.2 Decentralized Architecture - This model does not require a centralized server. All hosts which connect to a decentralized P2P Network send a request to all hosts which are currently logged on. The requesting host then receives a response from one or more hosts currently connected the network. Different sections of a file can be downloaded from multiple hosts.

1.3 Hybrid Architecture - This architecture offers a combination of the centralized and decentralized architecture.

The legal liabilities which organizations face due to their users downloading intellectual property such as music, software, literature, etc, for free, there is a ton of malicious code which traverses these networks. Malicious code such as Trojans and spyware can be wrapped in legitimate looking packages using all sorts of programs and downloaded via a P2P network. Unsuspecting users will launch these programs believing that they are legitimate, but not realizing that a Trojans was installed. An attacker may now have remote access to an organization's internal Network or potentially gathering confidential user information via a spyware program. Most organizations are under the impression that P2P networks can simply be stopped by blocking the default port that is required for these networks to communicate. Think again. Most P2P networks can be configured to listen on TCP port 80 (HTTP). Almost every organization in the world permits the use of HTTP through their firewall.

Along with viruses, one of the biggest threats to computer users on the Internet today is Malware. It can hijack the browser, redirect the search attempts, serve up nasty pop-up ads, track what web sites visited, and generally screw things up. Malware programs are usually poorly-programmed and can cause the computer to become unbearably slow and unstable in addition to all the other havoc they wreak. Many of them will reinstall themselves even after thinking that they are removed, or hide themselves deep within Windows, making them very difficult to clean. Although also considered to be Malware, programs such as viruses, worms, Trojans, and everything else generally detected by anti-virus software will not be discussed here, and the use of the word Malware will only explicitly refer to software that fits in the categories listed below. Malware often comes bundled with other programs (KaZaA, iMesh, and other file sharing programs seem to be the biggest bundlers). Malware is highly pervasive in P2P file-sharing systems and is difficult to detect. In order to lessen this problem, we analyze and provide preventive measures for Malware. Here the analysis includes two types for detecting Malware and provide two basic approaches reactive and proactive and Advanced techniques for preventing Malware. We instrument two different open source P2P Networks; KaZaA and iMesh to examine the prevalence of Malware in P2P networks and provide an algorithm for detecting Malware.

2 RELATIONSHIP TO PRIOR WORK

Peer-to-peer file sharing[2] is a form of file sharing using peer-to-peer networking. P2P allows users to download files such as music, movies, and games using a file sharing software client that searches for other connected computers (called ‘peers’).File sharing began in 1999 with the introduction of Napster, a file sharing program and central server that linked people who had files with those who requested files. The central index server was

meant to index all of the current users and to search their computers. When you searched for a file, the server would find all of the available copies of that file and present it to you. The files would be transferred between the two private computers. One limitation was that only music files could be shared. In June 1999, Napster was released as a centralized unstructured peer-to-peer system, requiring a central server for indexing and peer discovery. It is generally credited as being the first peer-to-peer file sharing system. Napster provided a service where they indexed and stored file information that users of Napster made available on their computers for others to download, and the files were transferred directly between the host and client users after authorization by Napster. Gnutella, eDonkey2000, and Free net were released in 2000, as MP3.com and Napster were facing litigation. Gnutella, released in March, was the first decentralized file sharing network. In the gnutella network, all connecting software was considered equal, and therefore the Network had no central point of failure. In July, Free net was released and became the first anonymity network. In September the eDonkey2000 client and server software was released. In 2001, KaZaA and Poisoned for the Mac was released. The Network was proprietary and encrypted, and the KaZaA team made substantial efforts to keep other clients such as Morpheus off of the Fast Track network. In July 2001, Napster was sued by several recording companies. This drove users to other P2P applications and file sharing continued its exponential growth. The Audio galaxy Satellite client grew in popularity, and the Lime Wire client and Bit Torrent protocol were released. In 2002, a Tokyo district court ruling shut down File Rogue and an RIAA lawsuit effectively shut down Audio galaxy. From 2002 through 2003, a number of Bit Torrent services were established, including Suprnova.org, isoHunt, Torrent Spy, and The Pirate Bay. With the shutdown of eDonkey in 2005, eMule became the dominant client of the eDonkey network. In 2006, police raids took down the Razorback2 eDonkey server and temporarily took down The Pirate Bay. In 2009, the Pirate Bay trial ended in a guilty verdict for the primary founders of the tracker. The decision was appealed, leading to a second guilty verdict in November 2010.

3 MALWARE PROPAGATION IN P2P

3.1 Peer-to-peer files sharing

Peer-to-peer file sharing[2] is a form of file sharing using peer-to-peer networking. P2P allows users to download files such as music, movies, and games using a file sharing software client that searches for other connected computers (called ‘peers’). Similarly, other computers on the Network are able to search for files on your computer. This differs from traditional file downloading that searchers server for the requested file. The widespread adoption and facilitation of peer-to-peer file sharing was helped by several factors. These include increasing Internet bandwidth, the widespread digitization of physical media files, and the capabilities of home PC's increasing to better handle playing and storing digitized audio and video files. Users were able to transfer either one or more files from one computer to another across the Internet through various file transfers and file-sharing networks.

3.2 Software Implementations For P2P

The following are the software implementations[3] for P2P

JXTA

JXTA™[4] technology, created by Sun™-is a set of open protocols that allow any connected device on the Network ranging from cell phones and wireless PDAs to PCs and servers to communicate and collaborate in a P2P manner. JXTA peers create a virtual network where any peer may interact with other and their resources directly even when some of the peers and resources are behind firewalls and NATs or are on different network transports. The project goals are interoperability across different peer-to-peer systems and communities, platform independence, multiple/diverse languages, systems, and networks, and ubiquity: every device with a digital heartbeat. The technology is licensed using the Apache Software License.

iFolder

iFolder[5] is an still in early development open source application, developed by Novell, Inc., intended to allow cross-platform file sharing across computer networks by using the Mono/.Net framework. iFolder operates on the concept of shared folders, where a folder is marked as shared and the contents of the folder are then synchronized to other computers over a network, either directly between computers in a peer-to-peer fashion or through a server. This is intended to allow a single user to synchronize their files between different computers (for example between a work computer and a home computer) or share files with other users (for example a group of people who are collaborating on a project).The core of the iFolder is actually a project called Simias. It is Simias which actually monitors files for changes, synchronizes these changes and controls the access permissions on folders. The actual iFolder clients (including a graphical desktop client and a web client) are developed as separate programs that communicate with the Simias back-end. The iFolder client runs in two operating modes, enterprise sharing (with a server) and workgroup sharing (peer-to-peer, or without a server).

Freenet

The Freenet[6] Project designed to allow the free exchange of information over the Internet without fear of censorship, or reprisal. To achieve this Freenet makes it very difficult for adversaries to reveal the identity, either of the person publishing, or downloading content. The Freenet project started in 1999, released Freenet 0.1 in March 2000, and has been under active development ever since. Freenet is unique in that it handles the storage of content, meaning that if necessary users can upload content to Freenet and then disconnect. We've discovered that this is a key requirement for many Freenet users.

Once uploaded, content is mirrored and moved around the Freenet network, making it very difficult to trace, or to destroy. Content will remain in Freenet for as long as people are retrieving it, although Freenet makes no guarantee that content will be stored indefinitely. The journey towards Freenet 0.7 began in 2005 with the realization that some of Freenet's most vulnerable users needed to hide the fact that they were using Freenet, not just what they were doing with it. The result of this realization was a ground-up redesign and rewrite of Freenet, adding a "darknet" capability, allowing users to limit who their Freenet software would communicate with to trusted friends. This would make it far more difficult for a third-party to determine who is using Freenet. Freenet 0.7 also embodies significant improvements to almost every other aspect of Freenet, including efficiency, security, and usability. Freenet is available for Windows, Linux, and OSX. It can be downloaded from, all software is available on The Freenet Project page.

Frost an application for Freenet that provides usenet-like message boards and file uploading/downloading/sharing functionalities. It should get installed with Freenet 0.7 automatically if you used the standard Freenet installers.

jSite is a graphical application that you can use to create, insert and manage your own Freenet sites. It was written in Java by Bombe.

Thaw is a file sharing utility and upload/download manager. It is used as a graphical interface for Freenet file sharing.

3.3 Malware

Along with viruses, one of the biggest threats to computer users on the Internet today is Malware[7]. It can hijack the browser, redirect the search attempts, serve up nasty pop-up ads, track what web sites visited, and generally screw things up. Malware programs are usually poorly-programmed and can cause the computer to become unbearably slow and unstable in addition to all the other havoc they wreak. Computers can get infected by Malware in several ways. Malware often comes bundled with other programs (KaZaA, iMesh, and other file sharing programs seem to be the biggest bundlers).These Malware programs usually pop-up ads, sending revenue from the ads to the program's authors. Others are installed from websites, pretending to be software needed to view the website. Still others, most notably some of the CoolWebSearch variants, install themselves through holes in Internet Explorer like a virus would, requiring to do nothing but visit the wrong web page to get infected.

Types of Malware

Although there is no official breakdown, we can divide Malware into several broad categories of Malware: adware, spyware, hijackers, toolbars, and dialers. Many, if not most Malware programs will fit into more than one category. It is very common for people to use the words adware, spyware, and Malware interchangeably. Most products that call themselves spyware or adware removers will actually remove all types of Malware.

Adware

Adware is the class of programs that place advertisements on the screen. These may be in the form of pop-ups, pop-unders, advertisements embedded in programs, advertisements placed on top of ads in web sites, or any other way the authors can think of showing an ad. The pop-ups generally will not be stopped by pop-up stoppers, and often are not dependent on having Internet Explorer open. They may show up when playing a game, writing a document, listening to music, or anything else.

Spyware

Programs classified as spyware send information about user and the computer to somebody else. Some spyware simply relays the addresses of sites visited or terms searched for to a server somewhere. Others may send back information type into forms in Internet Explorer or the names of files downloaded. Still others search the hard drive and report back what programs user have installed, contents of the e-mail client's address book (usually to be sold to spammers), or any other information about or on user computer – things such as user name, browser history, login names and passwords, credit card numbers, and phone number and address. Spyware often works in conjunction with toolbars. It may also use a program that is always running in the background to collect data, or it may integrate itself into Internet Explorer, allowing it to run undetected whenever Internet Explorer is open.

Hijackers

Hijackers take control of various parts of the web browser, including the home page, search pages, and search bar. They may also redirect to certain sites should mistype an address or prevent from going to a website they would rather not, such as sites that combat Malware. Some will even redirect to their own search engine when attempt for a search is made. NB: hijackers almost exclusively target Internet Explorer.

Toolbars

Toolbars plug into Internet Explorer and provide additional functionality such as search forms or pop-up blockers. The Google and Yahoo! toolbars are probably the most common legitimate examples, and Malware toolbars often attempt to emulate their functionality and look. Malware toolbars almost always include characteristics of the other Malware categories, which is usually what gets it classified as Malware. Any toolbar that is installed through underhanded means falls into the category of Malware.

Dialers

Dialers are programs that set up the modem connection to connect to a 1-900 number. This provides the number's owner with revenue while leaving with a large phone bill. There are some legitimate uses for dialers, such as for people who do not have access to credit cards. Most dialers, however, are installed quietly and attempt to do their dirty work without being detected.

GFI's Top 10 Malware List[8]

Table1 Top 10 detections for December

Detection	Type	Percent
Trojan.Win32.Generic!BT	Trojan	21.38
Trojan.Win32.Generic.pak!cobra	Trojan	3.71
Trojan Spy.Win32.Zbot.gen	Trojan	3.69
INF.Autorun (v)	Trojan	1.68
Trojan.Win32.Generic!SB.0	Trojan	1.59
Worm.Win32.Downad.Gen (v)	Worm.W32	1.47
FraudTool.Win32.FakeAV.hdd(v)	Trojan	1.06
Exploit.AbobeReader.Gen (v)	PDF Exploit	1.06
Exploit.PDF-JS.Gen (v)	PDF Exploit	0.80
Trojan.ASF.Wimad	Trojan	0.73

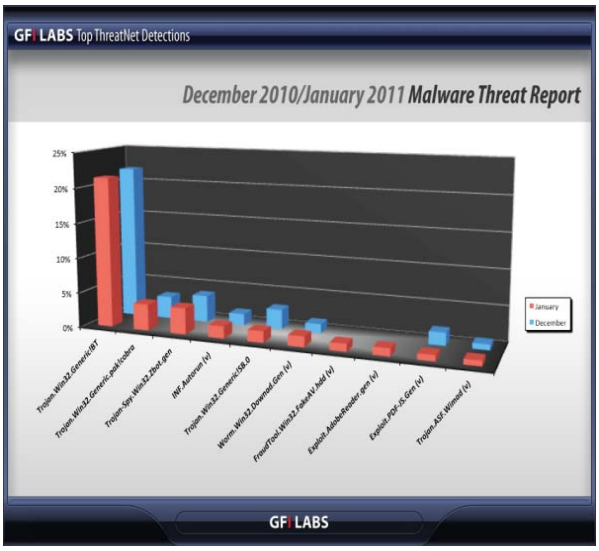

Fig.1 Top ThreatNet Detections

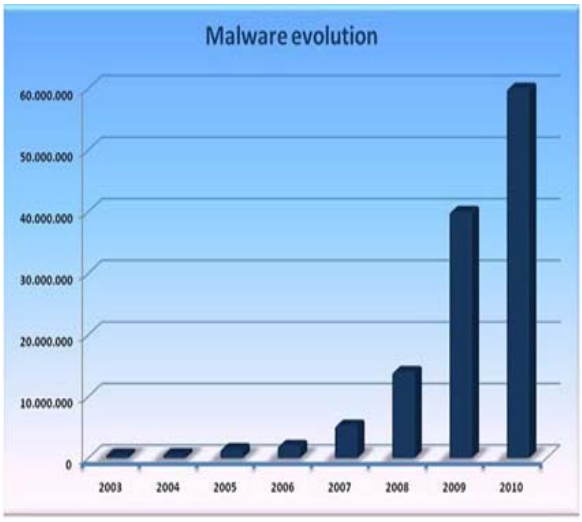

Fig. 2 Malware Evolution

4 ANALYSIS OF MALWARE

4.1 Goals of Malware Analysis

The goal of Malware analysis[9] is to gain an understanding of how a specific piece of Malware functions so that defenses can be built to protect an organization's network. There are two key questions that must be answered. The first: how did this machine become infected with this piece of Malware? The second: what exactly does this Malware do? After determining the specific type of Malware, you will have to determine which question is more critical to the situation. Now that we defined key terms and have determined the goals, it is time to discuss the common types of Malware analysis that are routinely performed.

4.2 Types of Malware Analysis

There are two types of Malware analysis[9] that security professionals perform:

Code (Static) analysis

Code analysis[9] is the actual viewing of code and walking through it to get a better understanding of the Malware and what it is doing.

Behavioral (Dynamic) analysis

Behavioral analysis[9] is how the Malware behaves when executed, who it talks to, what gets installed, and how it runs. Although both types accomplish the same goal of explaining how Malware works, the tools, time and skills required to perform the analysis are very different. When performing Malware analysis, both static and dynamic analysis should be performed to gain a complete understanding on how that particular Malware functions. Knowing how Malware functions allows for better defenses to protect the organization from this piece of Malware, and possibly Malware that attempt to infect a host using the same vulnerabilities are weaknesses.

Code analysis is performed by looking at the software code of the Malware to gain a better understanding on how the Malware functions. While performing code analysis, antivirus software will run on the Malware, string searches will be performed, and files such as shell scripts will be analyzed. Most likely, reverse engineering will have to be performed using programs such as disassemblers, debuggers and decompilers. After successfully reversing Malware, it will be able to see how the "source" code of the Malware functions. Seeing how the code functions allows the reader to build better defenses to protect their organization as well as serve as a sanity check on the completed behavioral analysis. Once the Malware code has been reversed, an understanding on how the Malware infects the system will become clear. With Malware today becoming more targeted, understanding how Malware infects systems can reduce infections to an organization, thus reducing the overall cost.

Behavioral analysis is the "quick and dirty" way of Malware analysis. When performing a behavioral analysis, look at how the Malware behaves and what

changes the Malware makes on a base lined system. It should be noted, when performing behavioral analysis it is critical the Malware lab in not connected to another network. For the best protection of production networks, the Malware lab should never be connected to any network. If files must be transferred use a read only media such as CD-ROM. When performing behavioral analysis, look for changes to the system as well as any unusual behavior on an infected system. Changes on the system that should raise a red flag include files that have been added and/or modified, new services that have been installed, new processes that are running, any registry modifications noting which modifications took place, and finally, if any systems settings have been modified. This would include DNS server settings of the workstation which have been changed. Beside the behavior of the system itself, Network traffic will also be examined. Now that an understanding of what behavior the Malware does to systems and networks, the reader may have the desire to understand how the Malware actually performs these activities. The answers to that question require the reader to perform an analysis of the Malware.

5 PREVENTIVE MEASURES

5.1 Tools and Techniques

Basic protection approaches[10] to Malware Organization/users can formulate their anti Malware strategy depending upon the type and complexity of Malware attacks that they are exposed to, and the level of risk associated with such attacks. Different organizations use different tools and approaches to counter Malware attacks. Selection of such tools and approaches is often based on their functionality, suitability and cost. The basic anti-Malware approaches that are traditionally used by organizations/users can be broadly classified under two heads based on their nature of their action. They are :

Reactive approach

Reactive approach is an incident response process. In this method once a problem is encountered, the investigation of the problem, analysis and finding remedy, and documenting the resolutions for future remedial are done, mostly in the same order. The existing anti-Malware tools available identify the Malware by scanning the computer executable files and check if any known Malware have sneaked into the system. This is done by detecting programs that are making changes to operating system registry. Here, the anti-Malware tools and products chase the Malware by identifying them after they have entered the system and the system shows some symptoms of being infected, depending on their behavior and instances. When dealing with reactive approach of the system, which is being infected, corporates have three alternatives for dealing with Malware. They are:

1. Running Malware removal tool to detect and repair Malware.
2. If, anti–Malware tools fail, Malware can be removed manually by the administrator or by formatting the system.
3. Use anti-Malware tool to prevent them from entering the system.

As preventive measures companies include disaster recovery plans, reinstalling operating system, system formatting and moving to alternatives as their reactive approaches. All these methods need to be in place, so that they can function as and when they are needed. As with any reactive approach, these techniques are time-consuming, error prone and costly.

The following steps should be performed if System is infected with Malware Using Reactive Approach[11]:

1. Make sure the firewall in place is working. Get positive control over inbound and outbound traffic on the systems and on the network.
2. Address the most likely suspects first. Clean the most common Malware threats and then check for unknown threats.
3. Isolate the infected system. Get it off the Network and the internet. Stop the infection from spreading to other systems on the Network during the cleaning process.
4. Research outbreak control and cleanup techniques.
5. Download the latest virus definitions from anti-virus software vendors.
6. Ensure that anti-virus systems are configured to scan all files.
7. Run a full system scan.
8. Restore missing or corrupt data.
9. Remove or clean infected files.
10. Confirm that the computer systems are free of Malware.
11. Reconnect the cleaned computer systems to the network.

Proactive approach:

Experiences state that proactive approach has its own advantage over reactive approach. As new technologies emerge, Malware writers are adopting high–level programming languages, new technologies and methods of attacks with varied features and payloads. In reactive approach a Malware can be identified only if they are in existence, i.e. at least executed once. Whereas in a proactive approach a Malware can be identified as new, as they are and they can be quarantined or deleted even before they get executed. Proactive approaches include various techniques that can enable the user to indentify the Malware when they attempt to invade the system. Unfortunately, getting infected with Malware is usually much easier than getting rid of it, and once you get Malware on the computer it tends to multiply.

The following steps should be performed if System is Infected with Malware Using Proactive Approach[11]:

1. Apply the latest firmware to hardware systems and routers as recommended by vendors.
2. Apply the latest security patches to server applications and other applications.
3. Subscribe to security–related e-mail lists from vendors and patches when recommended.

4. Ensure that all Microsoft computer systems are running anti-virus software.
5. Ensure that automated processes are running to regularly update the virus definitions.
6. Maintain a database that keeps track of what patches have been applied.
7. Review security logs.
8. Enable perimeter or host-based firewalls.
9. Use vulnerability scanner, such as the Microsoft Baseline Security Analyzer that helps to detect common security misconfigurations and missing security updates on your computer systems.
10. Use Least- privileged User Accounts (LUA). If low-privileged processes are compromised, they will do less damage than high–privileged processes. Consequently, using a non-administrator account instead of an administrator account while completing daily tasks offers the user, added protection against infection from a host of Malware, external or internal security attacks, accidental or intentional modifications to system setup and configuration, and accidental or intentional access to confidential programs or documents.
11. Enforce strong passwords policies.

5.2 Advanced Anti-Malware Techniques

The gaps or drawbacks of traditional security tools warrant a new, improvised and holistic approach to pre-empt Malware attacks. Similarly the anti-virus tools, the Malware tools need to be up dated frequently, to be effective in defending them. The following section list few advanced anti-Malware techniques.

Integrating Filters 'with signatures'

Having layers of application filters on the network will help in increasing the efficiency of the security tools. Advanced anti-virus tools, firewalls web and mail filters can be integrated together with their latest updates/patches, as multi-layered filters to prevent. The idea behind integrating the multiple layers of filters is to capture Malware that bypass the first layer with the later ones. This approach reduces the probability of Malware intrusion to minimum possible, though not zero.

For instance, Malware that attacks the web-browser normally bypasses the firewalls, but gets identified and deleted by web filters. Similarly a virus that spread through e-mail attachments or as a spam mail goes undetected, until they reach the e-mail filters. Thus, the integration of the tools solves the problem and makes the network and the system more immune to Malware attacks. All the filters in this multi-layered approach identified Malware with their respective definitions or signatures. As such, new Malware whose signature is not there in any of the filters can still sneak into the network unnoticed.

The following are five essentials before one deploys an integrated filter:

1. Have a user acceptable policy and create user awareness.
2. Guard the inlet and outlet of the traffic.
3. Have the updated and advanced tools and filters, which have the definition of the latest Malware.
4. Have good contact with security providers.
5. Have enforceable laws.

'Multi-Layered Defense' without signatures

This technique and the principles followed in this approach are similar to the 'Integrating Filters' approach. The only differentiating factor in these tools is that they can identify any Malware even without their definition or signatures. In the above technique, the integrated tools will have a definition for every Malware are identified and controlled. This leaves vulnerability, as the signature-based filters are amenable to attacks by unidentified or not reported Malware. To prevent 'Malware without signature' security strategy must be little different and hardened from the routine one. They should have strict policy on the authorized users and deploy improved file anti-tampering mechanisms. Incoming and outgoing traffic should be transmitted only after the authentication check. Users should be provided with unique and valid identifications. Machine number and the internet protocol should be continuously monitored and checked for unusual information.

The following are few technologies and approaches that are used in 'Multi-Layered Defense without signature':

Behavior-Based Security is a technology that is followed in advanced tools that help the user to identify the Malware that are new and unknown. Using this technology the anti-Malware tools are designed and developed to identify the Malware by analyzing the content and determining its behavior. Such tools will block the malicious content and allow the appropriate code.

Intelligent Layered Security is a technology that helps in filtering and controlling inbound and outbound traffic by monitoring the protocol. Strict user based authentication, well-structured policy and secured proxies are followed, while enforcing this technology.

Automated Outbreak Detection is an evolving approach, which helps to identify and delete the Malware, before they get into triggering or initiating mode. The blocked Malware codes continue to travel, blocking the traffic with slow response rate until they outburst or become unharmful.

Recurrent Pattern Detection (RPD) is a pattern pending technology that is developed by Commtouch Software Ltd. Its helps in stopping the Malware before they release their code or signatures.

6 PREVALENCE OF MALWARE IN P2P NETWORKS (KAZAA, IMESH)

6.1 Basic KaZaA Operation

When a user runs a KaZaA[12] client application, the client establishes a connection with an "indexing" hosts, called supernodes .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 serve 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 .When a client connects to a supernode, it sends two types of information to the supernode . The first is the list of files that the peering client host has in the sharing folder. 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, port number, and an IP address at which other clients can request a file download.

6.2 iMesh

The iMesh[13] is a good file-sharing program which can make users share media like songs and videos. But iMesh can be utilized by virus or spyware related with malicious sites which it going to redirect to commercial advertisement once it's installed on a system, especially those systems have lots of security loopholes. iMesh virus is a dangerous parasite to the affected computer. iMesh virus can monitor user network activities and violate private information, install other toolbar without permission. It's not easy to get rid this threat, it pops up ads constantly to annoy computer owners. If you're already troubled by iMesh virus, remove it with the manual guide for a complete removal instantly.

It was polled with a set of 100 randomly selected supernodes[12] at every 5 seconds for their availability. 100% of supernodes that did respond for the first 30 seconds never responded for the entire polling duration. 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 spreading platforms. Unlike self-propagating network worms such as Code Red and Slammer, most malicious programs in a P2P file-sharing network do not send their copies in the network by themselves. Instead, these viruses 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.

7 B3 DISCOVERY ALGORITHM

A basic building block[14], which is a model of Malware attacks, is constructed from a set of attack and non-attack programs as follows:

Convert each program (attack or non-attack) into a graph

A graphical representation is used because it is easier to generalize over. Since an attack program's source code is often unavailable, executable binary must first be transformed into a tractable high-level representation for the graph. The IDA Pro disassemble is used to automate this process; it obtains assembly code from binary code. Because IDA Pro is unable to unpack/decrypt binary code, we rest manually unpack and/or decrypt the program. The assembly code is then converted to a graph that is a hybrid of control flow and data dependence graphs.

Partition the graph into sub graphs

For abstraction, the overall graph is divided into sub graphs, each containing a program subgoal or terminal function.

Semantic abstraction

Semantic abstraction is the key to making our approach scalable. With abstraction, the graph is boiled down to its skeletal semantic essence. Our abstraction algorithm inputs a graph that has been divided into sub graphs, and outputs a finite-state machine (FSM) that captures global program semantics. An FSM representation has been chosen because it simplifies the induction process.

```
int main(void)
{
FILE*  fp  = NULL;              //file pointer
Char*  data = "abcde";
fp = fopen("test", "w");        //opens a file
if(fp == NULL)  exit(1); //if fopen() fails, then exit
fputs(data, fp); //writes data in the file
 fclose(fp); //closes the file
CreateProcess(program1,…);  //runs a program
int c;
fp = fopen("foo", "r"); //opens a file
c = fgetc(fp); //reads data
fclose(fp);//Closes the file
Return 0; //returns to operating system
}
```

Program A

```
int main(void)
{HANDLE h;  //file handle
Char buffer [1024];
Strncpy(buffer, "abcde" 5);
h= CreateFile("test"…); //opens a file
if(h=INVALID_HANDLE_VALUE)
 ExitProcess(1); //if CreateFile() fails, then exit
WriteFile(h, buffer, 5,…); //writes data in the file
CloseHandle(h);//closes the file
WinExec("program1", SW_SHOW); //runs a program
Return 0; //returns to operating system
}
```

Program B

wo general programs in C. Programs A and B are syntactically different but have semantically identical goals (goal1: write data into a file and goal2: execute a process). Note that **fgetc** in program A does not contribute to emitting an output.

Inductive inference

The final step is to perform inductive inference (Which is a form of machine learning) over strings (i.e., possible

executions) of all the FSMs - {for the purpose of inferring one general model (Signature) of all Malware seen so far that are in a certain class. With Inductive inference, strings from attack FSMs are treated as "positive examples" And strings from non-attack FSMs as "negatives examples" to Train on. After training on these examples, the general model will include Features of attacks, while excluding features of non-attacks. The resulting General model is a basic building block, or b^3, of Malware of a certain Class. This b^3, which is in the form of a generalized string (i.e., a string With disjunction allowed), is used for classifying new, previously unseen Programs as "ATTACK" or "NON-ATTACK.".

7.1 Virus Signatures

Compiled[12] 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. Among these, we have the content hashes of 71 distinct malicious programs. The Sig2Dat 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 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. It is stated that, 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. The following Table shows a break-down of these malicious programs by the propagation vector.

Table 1: Virus List

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 + messsenger	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)

CONCLUSION

Peer-to-Peer (P2P) Networks continue to be popular means of trading content. Some existing studies have shown that Malware proliferation can pose significant threats to P2P Networks, defending against such an attack are largely an open problem. In this paper we have explained the file sharing and problems in file sharing in P2P Networks and provide software implementations for P2P.Malware is highly pervasive in P2P file-sharing systems and is difficult to detect. Here in order to detect the Malware we have provided two basic analysis Code (Static) analysis, Behavioral (Dynamic) analysis and explained the goals for analyzing the Malware. The two basic approaches reactive and proactive and some advanced techniques are provided to prevent the malware. Two different open source P2P networks; KaZaA and IMesh are provided to explain the prevalence of Malware in P2P Networks. We also provide B3 Discovery Algorithm and virus signatures for detecting the Malware in P2P.

REFERENCES

[1] http://en.wikipedia.org/wiki/Peer-to-peer

[2]http://en.wikipedia.org/wiki/Peer-to peer_file_sharing

[3] http://en.wikibooks.org/wiki/The-world-of-peer-to-peer-%28p2p%29/Networks-and-protocols/other-software-Implementations#Mute_file_sharing

[4] http://www.jxta.org

[5] http://www.ifolder.com

[6] http://freenetproject.org

[7] Adam Baratz and Charles McLaughlin. Malware: What it is and How to Prevent It.

[8] http://www.gfi.com/page/67883/january

[9] Dennis Distler. Malware Analysis: An Introduction

[10] Ravi Kumar Jain Bandamutha and Roopa Ananth. Preventing Malware: Tools and Techniques

[11] Strategies for Managing Malware Risks, Published in August 2006 by Microsoft.

[12] Seungwon Shin, Jaeyeon Jung, Hari Balakrishnan. Malware Prevalence in the KaZaA FileSharing Network.

[13]http://www.exterminateit.com/malpedia/file/imesh.lnk

[14] The Basic Building Blocks of Malware.Jinwook Shin and Diana F. Spears

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

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

[17] Bryn Loban. Between Rhizomes and Trees: P2P Information Systems. In *First Monday Peer-Reviewed Journal on the Internet*, September 2004.

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

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

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

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

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