Global Information Assurance Certification Paper

Dangers And Containment Of P2P Utilities On A Corporate Network

By:

Armin Froemmel

GSEC Version 1.4b

22. April 2003

Table of Contents

1 Abstract

This paper will discuss the issue of P2P utilities. It will cover different security concerns associated with this kind of utilities. Furthermore it will show general measurements to contain respectively stop the usage of P2P data sharing tools on a corporate network. The basic functionality of several commonly used P2P tools will be explained. Based on the different techniques of the P2P tools possibilities how to stop the P2P traffic with a firewall will be shown.

2 Introduction

The Internet is a common place to share files and information – in short: data - with a defined group of people. Actually this is the main function of the Internet. For quite a long time this Internet data sharing was mainly based on client-server architecture. Server stored data that can be accessed by computers via the Internet.
This changed mainly because of two reasons: first, nowadays DSL, Digital Subscriber Line or cable modem offers an affordable high connection speed to nearly everybody who wants it. So any machine has the network bandwidth to be a server. Second, the invention of music file compression like mp3 made downloading music from the Internet realistic. That brought a new community to the Internet keen on swapping music files.
In 1999 the 18-year-old Shawn Fanning, nicknamed “Napster” started to work on a program for Peer-to-Peer (P2P) music sharing. Since that time a variety of programs were made to make it easy for everybody to share music, films and other files on the Internet.
Discussions are going on if P2P sharing is “good” or “bad”. Companies like Palisade claim that P2P data sharing tools has no justifiable value in a workplace. According to them P2P networks are primarily used for illegal sharing of copyrighted data[1]. On the other hand many people find P2P networks to be “A New Generation of Networked Filesystems”[2] with great possibilities.

Imho, there is no reason to blame P2P data sharing in general since data sharing is – as already stated – one of the main functions of the internet. But unfortunately some people always misuse good inventions. This creates security problems – consciously or not. And ‘misuse’ potentially applies to developers and users of P2P data sharing tools. Hence there are people who want to protect their networks against those P2P utilities, especially if it is a corporate network.

3 Preface

During studying this topic I’ve tested several P2P data sharing tools. This was done the following way

- Set up a Symantec Enterprise Firewall
- Set up a fully patched Windows2000 SP3 machine with Internet Explorer 6 SP1; installed “Spybot – Search & Destroy 1.2[3]” ; run Spybot and cleaned all problems found

[1] See http://www.palisadesys.com/news&events/p2pstudy.pdf
[2] See http://webservices.xml.com/pub/a/ws/2002/02/12/oram.html
[3] Tool to find and remove ad-/spyware; http://www.webattack.com/php/download.php?id=105384

- Created an image of this system
- Restored the image before installing another P2P client in order to have a clean system
- Run Spybot after installation of each P2P client
- Run 'netstat –an' on the client machine to get ports used
- Monitored the firewall logfile
- Run 'tcpdump -x' on the firewall machine to analyse the traffic

I found that a lot of these tools are doing strange things users often not aware of, e.g. transmitting data to suspicious sites although the tool was said to be free of adware and spyware. On the other hand there are also tools that I found to be completely clean and do exactly what they claim. Still I will not mention the quality or strange behaviour of a single tool explicitly. This document is not intended to be a P2P client evaluation. It's a document for a firewall/IDS/network administrator's point of view that wants to know how to counteract to this tools and only basically how the tools work.

4 Problems of P2P Utilities

4.1 Adware, Spyware

Many P2P utilities contain adware and spyware. What is spyware? Steve Gibson, a well-known IT security expert defines: "Spyware is any software which employs a user's Internet connection in the background (the so-called "backchannel") without their knowledge or explicit permission."[4]
This implicitly installed spyware programs are used to collect information about a user like Internet surfing behaviour and send it to somebody making use out of this information. Unfortunately some spyware programs are said to collect more information than need. It is not 100% clear, which spyware program collects which data.
Adware in contrast is merely found rather bothering than dangerous. It just displays some ads downloaded from the Internet. Users should be aware, that some adware programs also collect data and send it out (often tracking cookies are used to do this). But since ads are running in the foreground not hidden from the user, adware companies refuse the name "spyware".
Although neither adware nor spyware are illegal, these programs can be a security problem, since information may leave a company that should not.
Just to prove that spy/adware gets installed without blaming a single P2P tool, I've installed 4 randomly selected P2P tools on a clean machine and ran Spybot afterwards. Following spy/adware programs got installed:

Advertising.com	Sends IP address and a list of visited
Commision Junction Inc.	Tracking users' activities
eAcceleration	User tracking
EBates Money Maker	Adware
FastClick	User tracking
Gator	User tracking, can automatically fill in passwords
HitBox	User tracking
New.net	Suspicious; gives access to non-official top-level-domains

[4] See http://grc.com/optout.htm

4.2 Vulnerabilities

There's no software without vulnerabilities, but some facts make vulnerabilities inside Internet data sharing utilities especially dangerous:

- Internet data sharing Utilities are turning any PC it is installed on into an Internet file server. This makes the PC easy to access, the vulnerability is potentially easy to exploit.
- They are often installed and run by persons without much IT and security knowledge. In addition they are widely spread, hence an interesting aim for attackers.
- Illicitly used from inside of company networks the utilities have not gone through a certification process by the company. The responsible IT personnel are possibly not aware of the existence of these utilities on the corporate network and haven't applied proper security measures.
- In general these utilities are aimed to quick and easy use and not to security.
- The QA process some of these utilities run trough is not as good as the QA process for business server products, so they tend to have more vulnerabilities.
- There are a huge variety of these utilities out on the Internet connecting to the same network. Additional, version updates are very frequently and the change are often not transparent. This makes it hard to stay aware of potential security risks inside these products.

4.3 Access and data security

Groove Networks is an example for a P2P business solution providing built in authentication and encryption. Groove is a business P2P solution. Most private users' P2P sharing tools are built for easy use only, so they don't care much about security. Some, like Napster, authenticate against a central server. But this is more a method for the people running the server to get paid for the service. There's no authentication between clients, where the actual shared data is stored and downloaded from. Actually the main concern of many P2P data sharing tools seems to be the perpetuation of the client's anonymity[5], and not security especially if they are using encryption.

Based on that it's not surprising that a number of worms and viruses use P2P sharing tools to spread, e.g. Worm.P2P.Bare, Worm.P2P.Benjamin, Worm.P2P.Duload, Worm.P2P.Kazmor, Worm.P2P.Relmony, Worm.P2P.Spear or Worm.P2P.Togod. By having a worm on a system not only the security of the shared area but the complete system is compromised.

File and directory shares regardless of worms are not only a security problem of P2P data sharing tools. However it's easier to accidentally share files, because most P2P tools create shares during install without any security warning and, allow access to everybody on the Internet.

4.4 Network usage

'The Register' stated last year, that "P2P activity accounts for up to 60 per cent of the total traffic on any service provider network"[6]. Many network administrators, especially at colleges, universities and schools claim that 50 to 90 percent of the

[5] See e.g. http://freenetproject.org/cgi-bin/twiki/view/Main/WhatIs
[6] See http://www.theregister.co.uk/content/22/27092.html

network traffic is due to P2P data sharing tools. P2P seems to be a very 'hungry' invention.
Hypothetically spoken there could be a hacker creating a P2P tool with the purpose to make a Denial-of-Service attack to networks based on the high bandwidth usage of the P2P tool. So the high network usage would be a directly security related problem. This thesis is quite vague and hypothetically. Hackers find easier and more effective ways to do DoS attacks and it's most likely not the intention of P2P users and developers to create bandwidth bottlenecks. Still illegal bandwidth usage is a risk for a company's security since business application can suffer from low bandwidth. So bandwidth problems are a main reason for administrators to block P2P data sharing tools.

4.5 Legal issues

Legal issues with P2P data sharing are well known. P2P is blamed very often because pirate copies, copyrighted files or illegal files are shared. In many countries companies can be sued for what is stored on their machines, even if an employee did it illegally. That is what already happened.
But there are more dangers regarding legality, which are less obvious. For example somebody could share illegal files anonymously but tracking the IP addresses downloading it. If the IP address resolves to a company, ideally a competitor, he may sue the company. There may not enough evidence to sue them, but very often it's enough to make it public that somebody has illegal files on the network to do damage to a companies image.
Projects like Freenet may cause similar problems. Freenet uses encrypted traffic and encrypted shared data stores. It's built up in that way that nobody knows on which machine is stored which data – not even the owner of the machine knows what is in his shared data store. The reason for that is that nobody can be made responsible for the data on his machine because the content is not known (that is true for several countries). While this is a topic for controversial discussions where nobody knows where freedom ends and anarchy starts, it's easy to harm companies: prove there are illegal files on this net, prove the companies machine is a member of this net and make it public.

5 How can I counteract?

Different items help to increase IT security and could also clog or block the use of P2P data sharing tools:

- A working and well defined security policy
- Vulnerability management to be aware of the latest vulnerabilities and have proper counter-measurements in place
- Security awareness trainings for employees at a regular basis
- Security aware employees
- A protocol aware application proxy firewall to block tunnelled traffic
- Intrusion detection systems with signatures also for P2P data sharing tools
- Antivirus to protect against downloaded malicious code
- Operating systems which does not allow non-administrators to install software
- Access to the network limited to MAC addresses to block wild computers
- Network Scanner to find wild machines and services
- Enough IT security people to manage and monitor the above

Having this all in place would provide a good overall network security making it hard for any problematic program including P2P tools to exist unwanted on this network. In the following a few “point counteractions”, which are often used against P2P tools will be discussed.

5.1 QoS filters

Applying Quality-of-Service filters is often used to balk P2P data sharing. Packet shapers throttle the amount of bandwidth certain protocols can use and guarantee other protocols a minimum of bandwidth. This possibility is mainly use in colleges, universities and schools that want to avoid bandwidth problems but still want to have open networks. This may solve the problem here since the average user does not have enough knowledge to get around this. More sophisticated users would use P2P data sharing tools where they can use a variety of ports or which can tunnel through ports known to have a big piece of the bandwidth (like http). Also Packet shaping inherently will not increase the security regarding P2P data sharing tools.

5.2 Intrusion Detection Systems

Networked IDS has the best possibilities to stop P2P data sharing. By analysing the traffic the IDS can identify P2P traffic and stop it. Hence there are problems using IDS for blocking P2P traffic:

- P2P traffic is not an attack
  if IDS is deployed on a larger network centralized and correlated data about attacks is needed. Since a IDS is not built for stopping normal traffic the P2P events may screw up the correlation.
- Some P2P use encrypted traffic
  possibly hard to find out if the encrypted traffic originates from P2P clients. This may result in false positive.
- Signatures for P2P are hard to get
  most companies offering IDS do not supply signatures for P2P tools. It can become a huge effort to create own ones.

PacketHound from Palisade Systems is a dedicated system to block P2P traffic. It works like a network sniffer and comes with predefined signatures for a major number of P2P utilities. This is probably an easy and promising way e.g. for schools, where P2P utilities are a major problem. Other companies where P2P tools are probably a minor issue will not want to deploy a dedicated system for P2P blocking.

5.3 Firewall

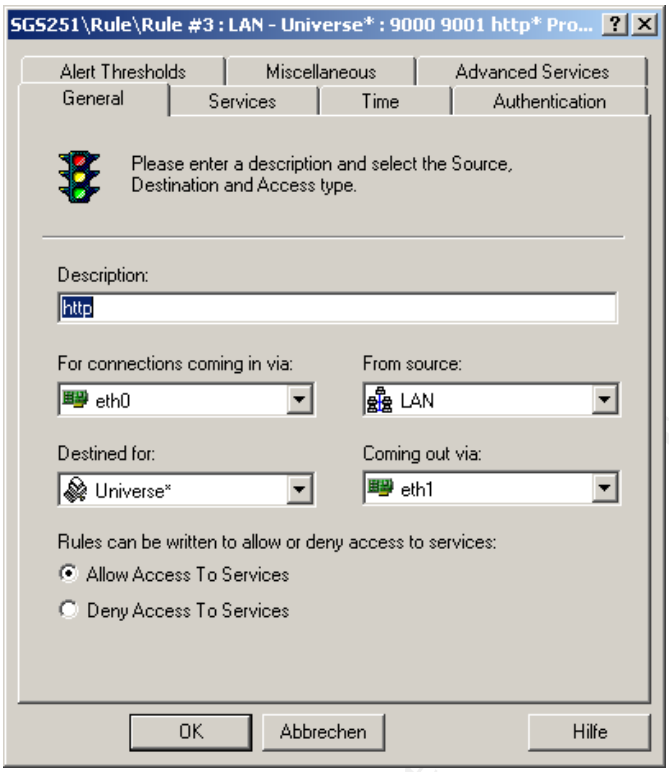
Firewalls have a good potential to stop P2P data sharing even if some P2P state firewall cannot stop them. It really depends on the firewall. In the following I will just take dedicated gateway firewalls into account but not personal firewalls.
A lot of SOHO firewalls are sold pre-configured like “allow all outbound traffic, deny all inbound traffic”. This is where many P2P tools claim to work behind a firewall. Since this is made for convenience, but not security, it must be changed. Basics when trying to stop P2P traffic with a firewall are:

- Open only the ports inbound and outbound that are needed for business.
- Use NAT
- Use a private address space behind the firewall

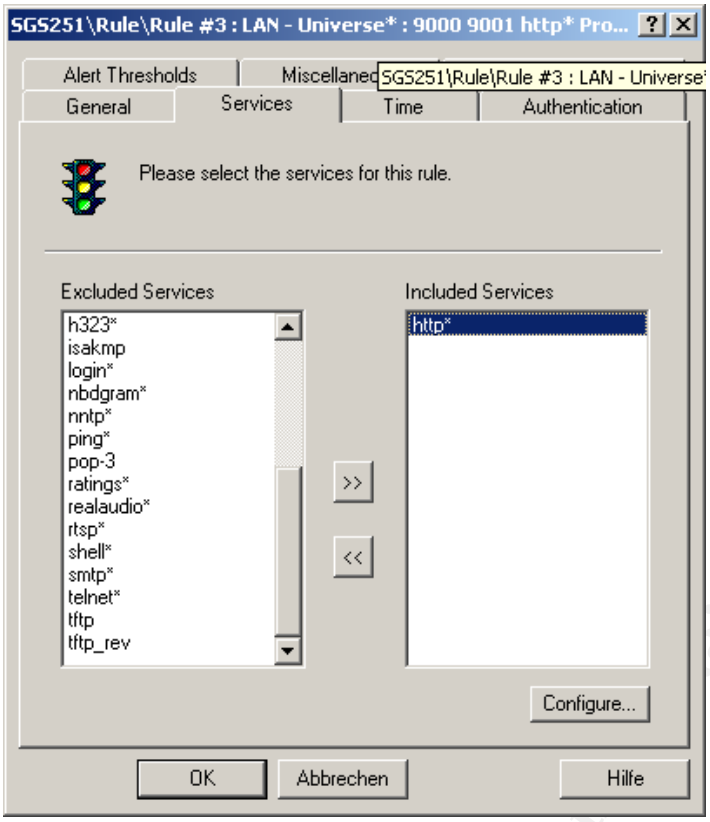
This prevents already a lot of P2P tools from working completely. Most of them would need a service redirect on the firewall to be able to act as server for outside clients.

Recently I was asked for settings on the Symantec Enterprise Firewall to block P2P tools. So I used the Symantec Enterprise Firewall to protect my test network and to find out the proper settings.
The Symantec Enterprise Firewall denies all inbound and outbound traffic by default. No rules are predefined. The Firewall is proxy firewall that can analyse traffic over certain protocols like http very granular and block accordingly. This is called 'http pattern matching' and can be enabled a follows:

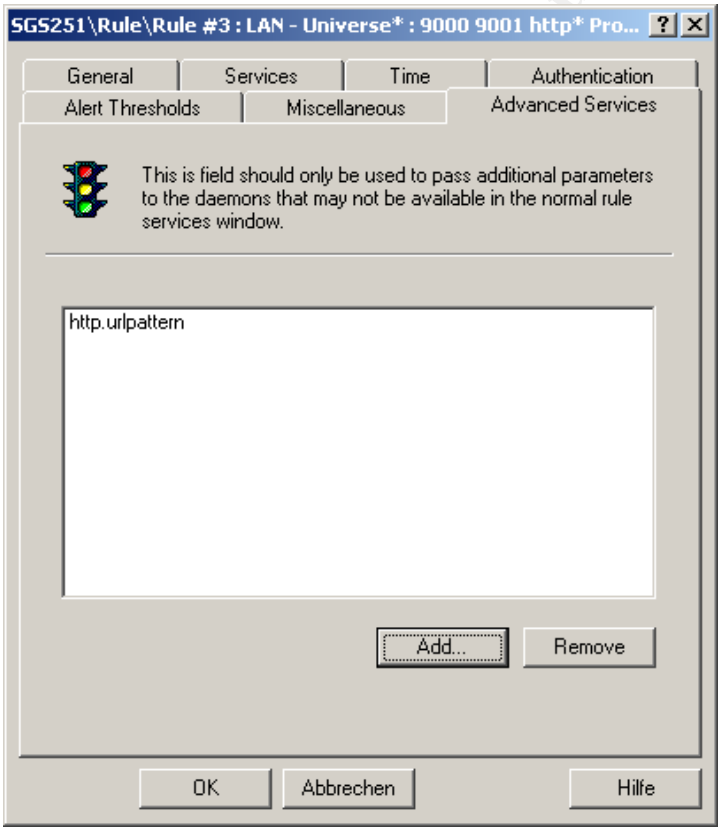
.



Open or create an http rule where pattern matching should be enabled

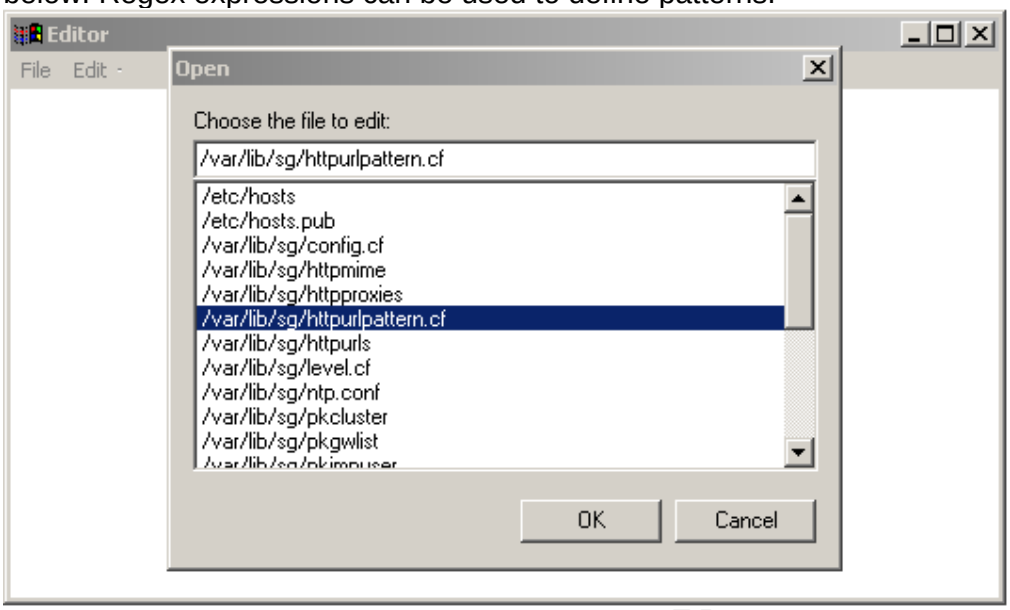


Select http* as service. The asterisk shows, that this is native (protocol aware) firewall proxy. Only if this proxy is used, pattern matching will work.



Select the “Advanced Services” tab and add the string “http.urlpattern” as shown.

After saving the settings edit the configuration file called httpurlpattern.cf as shown below. Regex expressions can be used to define patterns.



Later several P2P tools will be analysed in detail. Where suitable, useful patterns for will be provided.

6 How does P2P work?

Today's available P2P data sharing tools are based on different techniques. To know how the tools work and which ports are used is essential to assess the risk and to find proper counteractive measures.
In addition lists of common used P2P data sharing tools will be provided. These lists may not complete. There are probably some more and there will be more in future, but the techniques used are predominantly based on one of these below. Just to mention that there are also hybrid P2P clients who have different techniques built in (like MyNapster) to connect to different networks.
Below you will find commonly used P2P techniques/tools and following information:

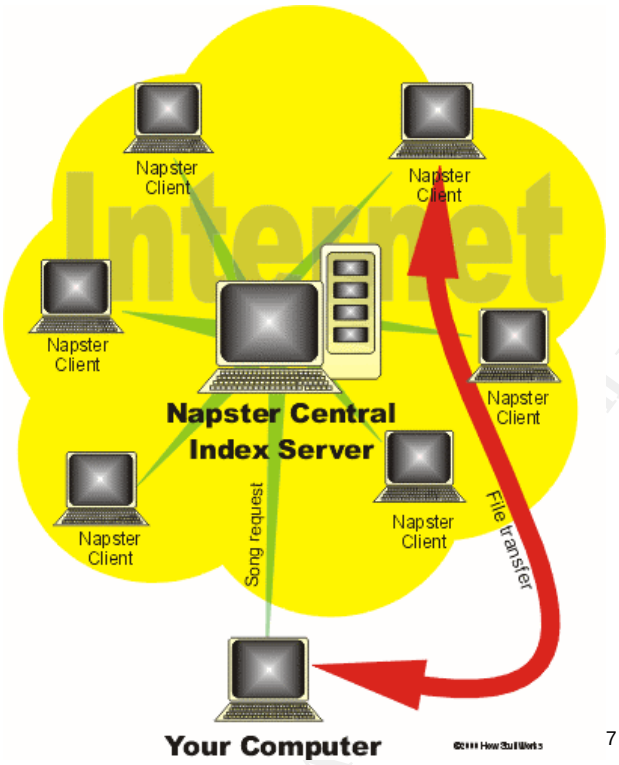
- **How it works**: a brief description of a P2P technique/tool
- **Ports used**: TCP/UDP ports needed by a P2P tool
- **Clones**: similar P2P tools connecting to the same network
- **Counteract with a firewall**: what can be done on a companies site firewall to clog or stop usage. Examples are based on the Symantec Enterprise Firewall.

6.1 Napster network

6.1.1 How it works

Napster is a P2P MP3 sharing tool, which works with a central index server. This server maintains a database with the information about all the files shared by all his clients. So all shared files are on the clients, the Napster Central Index Server just keeps the information where to find this shared files.
Because of maintaining this information on the server, Napster was sued. The result was, that now most Napster based P2P networks are payable services now or restricted to a group of people. Napster users have to authenticate when they access

the Napster Central Index Server. There is no authentication from client to client where files actually are downloaded.



[7]

6.1.2 Ports used

The connection to the server Napster Central Index Server is done via ports 8875/tcp, 4444/tcp, 5555/tcp, 6666/tcp, 7777/tcp or 8888/tcp. The actual download from other clients is done via port 6699, 6700 or 6701. The ports can be tcp or udp ports.

6.1.3 Clones

- Amster
- AutoNap
- BeNapster
- BitchX
- Blazter
- Crapster
- Console Napster CLT
- DeWrapster
- DiaRRIA
- DJnap
- Fanster
- File Navigator
- Gnap
- Gnapster
- NapMan
- Napster
- Napsack
- Napster/2
- Napsterminator
- iNapster
- JNap
- J Napster
- Jnerve
- KNapster
- Koog Epsilon
- Lopster
- Macstar
- Macstar
- OpenNap
- Pakster
- PMNapster
- Rapigator
- Rapster
- Riscster
- Shuban
- Snap
- Socks2HTTP
- Spyster
- Swaptor
- TekNap
- TKNap
- Unwrapper

[7] http://computer.howstuffworks.com/napster2.htm

- Gnome-Napster
- GTK-Napster
- Hackster
- nap
- NapAmp
- Napigator
- Napkin
- Macster
- Music City
- MyNapster
- Napster Unban
- Netstreak iAssimilator
- N-Dream Plug-In for Napster
- Webnap
- WinMX
- Wrapster
- XMNap
- Xnap
- Xnapster

6.1.4 Counteract with a firewall

Inbound:

Protected by default through using NAT. For inbound access (acting as server) a Napster client would need a service redirect on the firewall to work. Additionally block Napster client ports inbound.

Outbound:

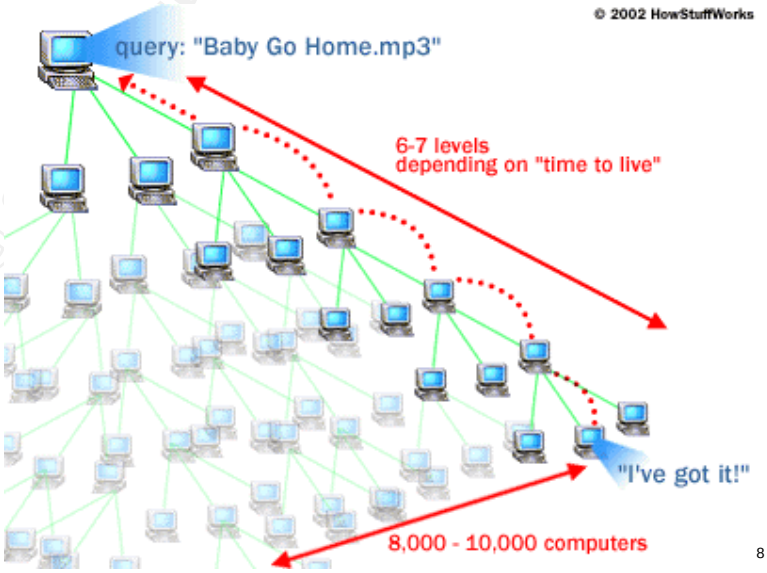
Since each client needs to contact the Napster Central Index Server to be able to take part on the Napster network, Napster clients are quite easy to block by denying outbound access via all ports listed above. If only the ports to the Napster Central Index Server are blocked, the connection to the Napster network cannot be established either.

6.2 Gnutella network

6.2.1 How it works

The Gnutella network does not rely on a central index server. It's completely client based. As long a client can find at least one other Gnutella client, the network is working. The search request is sent directly to other clients. This client sends on the search request to other clients. If a client finds a match to the search request he sends back the filename, IP address and port to the requester.

.



[8]

[8] http://computer.howstuffworks.com/file -sharing3.htmb

Gnutella also supports a technique called “push route”: if an outside client wants do download something from a machine behind a firewall, but there’s no inbound rule for this connection, the inside client can initiate the connection from his side.

6.2.2 Clones

- Bearshare
- Bodetella
- Cooltella
- Freewire
- Furi Launcher
- Furi Updater
- Gnewtella
- Gnewtella 2
- GnOtella
- GnuCache
- Gnucleus
- Gnujatella
- Gnumm
- Gnuspace
- Gnutella
- Gnut
- Gnutella.it
- Gobobo
- GTK-Gnutella
- Hagelslag
- Limewire
- Mactella
- Morpheus
- MyGnut
- MyNapster
- MyTella
- N-Tella
- Newtella
- PeaGnut
- Pi
- Phex
- Pygnut
- Reflector
- SeachLord
- Gnute
- Gnutmeg
- Gnutella Crawler
- Shareaza
- Tellaseek
- Toadnode
- XoLoX

6.2.3 Tested clients

Bearshare 4.2.5, Limewire 2.9.8, XoLoX v.1.5 built 740, Gnucleus 1.8.4

6.2.4 Ports used

80/tcp
6346/tcp
6346/udp
6347/tcp
6347/udp

These are the mainly used (IANA registered) Gnutella ports. In addition Gnutella clients sometimes use other ports. Ports found when testing with Limewire, Gnucleus and XoLoX are (udp and tcp):

5302	6641	7216	9193
5476	6813	7275	9492
5705	7069	7650	9533
5770	7623	7878	9645
5985	6268	8777	9654
6348	7883	9071	
6420	7691	9189	

How are clients found attaching on non-standard ports to the Gnutella network? Limewire, Gnucleus and many others have hardcoded links to GWebCache web-based Gnutella host cache scripts. These cache scripts can supply IP addresses and ports of Gnutella clients and also links to other GwebCache scripts.
Example:

Hardcoded link:
http://www.gamerspage.com/lynn.asp?client=LIME&version=2.9.8&urlfile=1

Content of link above

http://www.riverstyx.com/gcache.php http://www.hottdeals.com/postNuke7/gWebCache/index .php
http://dannyj.dynip.com/gwebcache/gcache.php http://www.jetzweb.de/metacyborg/gc/gcache.php
http://members.lycos.nl/killerarnold/gcache.php
http://members.lycos.co.uk/espinhoso/Gcache/gcache.php
http://www.lostdaemon.net/gwebcache/gcache.php
http://theccu.org/webservices/GnuCache/index.php http://www.kayaman.net/gweb http://cgi -
bin.spaceports.com/~gnucache/gcache.php http://intranet.ktg.se/~andbje/gnutella/gcache.php
http://luna.acad.bg/gcache/gcache. php http://www.tonehog.net/gwebcache/gcache.php
http://www.jillyboel.com/gwebcache/gcache.asp http://www.visualcave.com/gcache/gcache.php
http://mitglied.lycos.de/versus167/gnetcache/index.php
http://www.infowebmaster.com/gnutella/gcache.php http://gwebcache3.jonatkins.org.uk/perlgcache.cgi
http://www21.brinkster.com/icarus2de/lynn.asp http://kjellman.com/gcache/gcache.php
http://www.propension.net/ gwebcache/gcache.php http://borednow.net/GWebCache/GCache.asp
http://www.donutz.de/GWebCache/gcache.php http://gwebcache2.jonatkins.com/cgi -
bin/gwebcache.cgi http://www.hottdeals.com/postNuke7/gWebCache/gcache.php
http://www.inthetrunk.com/gcache/gcache.php http://www.polarhome.com/~sirjr/gcache.php
http://lunaris.neomain.com/gwc/gcache.php http://www.theholt.net/gcache.php
http://www.easypublish.net/gwebcache/gcac he.asp

Content of last link above

GWebCache
[main | stats | test]

Stats
Total Requests: 3339

So Far This Hour (in the last 33 minutes):
821 Requests (1492.7 per hour)
71 Updates (129.1 per hour)

Hosts in cache: 20 of 20
URLs in cache: 10 of 10

Hosts
62.131.160.33:6346
24.89.22.222:6349
157.158.56.66:6346
217.229.13.201:6346
62.179.99.89:6347
209.142.131.81:7567
62.3.122.101:6346
24.79.241.35:6346
198.82.94.79:6346
213.65.171.236:6346
195.56.225.239:6346
207.177.68.246:6347
142.163.96.68:7684
66.168.132.225:7828
218.8.217.244:2500
81.103.78.196:6346
195.241.46.80:28125
67.39.178.249:6346
24.26.129.138:6346
66.74.112.34:6346

URLs
http://207.71.250.4/gcache/gcache.php
http://www.asiinfo.net/gwebcache/gcache.php
http://www.jetzweb.de/metacyborg/gc/g cache.php
http://www.commontology.de/andreas/gwebcache/gcache.php

http://s91.tku.edu.tw/~291510500/gwebcache/gcache.php
http://cache.m ynapster.com/index.php
http://gwebcache.amateur -hour.net/gcache.php
http://membres.lycos.fr/waryde/gnuc/gcache.php
http://www.texasrulz.com/gcache.asp
http://www12.brinkster.c om/chris5g/gwebcache/index .asp

XoLoX is using a slightly different approach. It has a built in web browser with a search bar. You can select if you want to do a web search or a P2P file search. Based on the selection it connects to a XoLoX search site and submits the search string to it. The search is done from the search site directly (probably by using GWebCache data), the result of the search is displayed in the browser window of XoLoX. In case of a P2P file search this is the filename matching the search, IP address and port of the client providing the file.

6.2.5 Counteract with a firewall

Inbound:

Dies not work by default since there are no service redirects on the NAT firewall. Additionally block Gnutella ports inbound listed above.
Data may still be retrieved from the client via push route. To restrict this, block ports used for outbound connections (see below).

Outbound:

It's quite hard to block Gnutella completely. If the default ports 6346/tcp, 6346/udp, 6347/tcp and 6347/udp are blocked searches retrieve still results. But since most Gnutella clients run on the default ports, it cannot be downloaded from most clients. The default ports are IANA registered for Gnutella, so it is probably not used for anything else and can be blocked without problems. It may not be possible to block all ports mentioned above since they may be used for other programs.
In addition to blocking ports the access to the GwebCache file can be blocked on Symantec Enterprise Firewall. Enable pattern matching and add the following lines to the httpurlpattern.cf:

```
.*gcache.php
.*index.php?client=LIME
.*index.php?client=GNUC
.*index.php?client =ATOM
.*index.php?client=MMMM
.*index.php?client=RAZA
.*index.php?client=ACQX
.*index.php?c lient=BEAR
.*index.php?client =GTKG
.*index.php?client=MNAP
.*index.php?client=MRPH
.*index.php?client=PHEX
.*index.php?client=MUTE
.*index.php?client =TOAD
.*index.php?client=GNOT
.*index.php?client=MACT
.*index.php?client=GNUT
.*index.php?client=NAPS
.*index.php?client=HSLG
.*index.php?client=CULT
```

The 4-character vendor code for the client can be obtained by going to several

of the GwebCache sites mentioned above and selecting ‘statistics’.

6.3 Others

There are many more P2P tools like Napster or Gnutella. Some of them are already history; they are not developed any more. This is not a hint for a decreasing use of P2P tools but just a normal selection process. Below is a selection of quite commonly used P2P tools.

6.3.1 Audio Galaxy Satellite

Needs http and ftp open.

How it works

Connects to trickle.Gator.com via http and tells this server the source port for the ftp connection the client will use.

Tested client

Audio Galaxy Satellite 06.09

Ports used

http:80/tcp
ftp: 21/tcp

Counteract with a firewall

Inbound:

Inbound traffic does not reach the client since there are no redirects on the NAT firewall.

Outbound:

AudioGalaxy Satellite is probably hard to block completely with a non-proxy firewall since http and ftp are commonly used ports and therefore open on many firewalls. A proxy firewall will block it by default since each connection to the destination will be established newly by the proxy firewall. Since this will be most likely another source port than the server was told, AudioGalaxy Satellite will not work.

6.3.2 Blubster

How it works

Blubster is a udp based P2P tool. The port is not changeable.
Blubster gets the list of community members via http from Blubster gateways, e.g. www.blubster.net/gateway, www.blubster.com/gateway, www.mp3bytheface.com/gateway

Tested client

Blubster 2.0 B2

Ports used

80/tcp
41170/udp

Counteract with a firewall

Inbound:

For inbound access Blubster clients would need redirects on the firewall to work.

Outbound:

To stop Blubster from working, block port 41170/udp. Since this port is not changeable, Blubster is easy to block.

6.3.3 iMesh

How it works

iMesh clients have to register which includes the specification of an email address. Both, port 80/tcp and 1214/tcp are needed to register. These ports cannot be changed.

Tested client

IMesh 4.0 built 131

Ports used

80/tcp
1214/tcp

Counteract with a firewall

Inbound:

For inbound access iMesh clients would need redirects on the firewall to work.

Outbound:

Block port 1214/tcp. iMesh cannot even be installed if 1214/tcp is blocked. The port cannot be changed, so it's easy to block.

6.3.4 eDonkey

How it works

eDonkey works on dedicated ports. It can also be configured to tunnel the traffic through port 80/tcp

Tested client

eDonkey2000 v.047

Ports used

80/tcp
4661/tcp
4662/tcp
4665/udp

Counteract with a firewall

Inbound:

For inbound access eDonkey clients would need redirects on the firewall to work. Additionally block port 41170/udp inbound.

Outbound:

Block ports 4661/tcp, 4662/tcp and 4665/udp

If tunnelling through port 80 is enabled, the Symantec Enterprise Firewall http proxy blocks the traffic by default, since the traffic is not RFC conform http traffic:

Apr 10 16:03:30.098 SGS251 httpd[31355]: 238 httpd Notice: Specified request method not implemented
Apr 10 16:03:30.098 SGS251 httpd[31355]: 219 Can't parse url (\343\")

6.3.5 Filetopia

How it works

Filetopia encrypts all traffic. It works by default on randomly selected ports. If "behind firewall" is specified during installation, 443/tcp will be selected as server port. There's not much information on the Internet about how Filetopia works. It definitely connects encrypted to servers where it gets IP's and ports of other Filetopia clients. In my case this was always the same server and tcp port: 213.73.224.160/20443.

Clones

Tesla

Tested client

Filetopia v3.04d

Ports used

3 Port listening: 1 random TCP for file server, 1 random UDP port for chat, https for chat. Can also be set to static ports.
server acts as "meeting point", files and chats go p2p
20443/tcp

Counteract with firewall

After initial setup a nickname registration inside Filetopia is needed. This didn't work behind the Symantec Enterprise Firewall at all even all port to all destinations in- and outbound were open. Obviously there's a problem in Filetopia with submitting registration data via a proxy firewall. Switching the client for a short period behind a stateful packet filter firewall made the registration process work.

Inbound:

Filetopia would need service redirects to work behind a firewall

Outbound:

If outbound ports are limited, Filetopia is unusable. This is because destination ports to outside clients are still random and Filetopia will try over and over again.

If port 20443/tcp was closed outbound, Filetopia v3.04d did not work at all in the test scenario.

6.3.6 KaZaA

How it works

KaZaA connects on port 1214/tcp to other KaZaA clients.

Clones

KaZaA-Lite, Grokster, Morpheus, XoLoX

Tested client

Kazaa-Lite 2.1.0

Ports used

Default port: 1214/tcp, 80/tcp optional as incoming port
Outgoing port: 80 (tunneled)

Counteract with a firewall

Inbound:

Since KaZaA would need a redirect on the NAT firewall, it will not work
Inbound access via port 80 can be enabled, but it will not work either because of the missing redirect.
KaZaA also tries to tunnel traffic via HTTP. This is blocked by default by the HTTP firewall proxy:

Apr 10 15:40:31.071 SGS251 httpd[31364]: 238 httpd Notice: Specified request method no t implemented
Apr 10 15:40:31.071 SGS251 httpd[31364]: 219 Can't parse url (N \241\226c&\363\223{)
Apr 10 15:41:04.209 SGS251 httpd[31362]: 238 httpd Notice: An illegal character (0x0e) was found at position 2 in the request (see RFC2068, RFC1738, and RFC18 08)

Outbound:

Block port 1214/tcp open to work.

6.3.7 NetMess

How it works

NetMess works with a technology quite similar to the Gnutella network. It consists of two parts: a NetMess client, which is programmed in Java, hence platform independent, and a NetMess node, which is platform dependent.
The NetMess node does the file downloads and the NetMess nodes discovery.
Initially NetMess will download a list of active nodes from a site referenced in the netmess.ini.
The NetMess Client can connect to different NetMess nodes to initiate searches and downloads.

Tested client

NetMess 0.99.4

Ports used

6868/tcp
7070/tcp

Optional:
80/tcp
443/tcp

Counteract with a firewall

Inbound:

NetMess needs a service redirect for inbound traffic on NAT firewalls, so it shouldn't be a problem on corporate networks.

Outbound:

Block 6868/tcp and 7070/tcp inbound and outbound.
NetMess has the ability to tunnel traffic over http. Since Symantec Enterprise Firewall is a protocol-aware proxy Firewall and the NetMess traffic doe obviously not adhere to the RFC standard, it will not let this traffic through.
Logfile message:

httpd Notice: Specified request method not implemented

6.3.8 Freenet

How it works

Freenet downloads install files from Web during installation. It also downloads seednodes.ref. Seednodes.ref contains existing Freenet nodes with their tcp ports. Users are asked during installation to keep the default Freenet client port (see below).
All data and all traffic with Freenet are encrypted.
Freenet runs as proxy on the local machine (default port 8888/tcp). So user can use the Freenet proxy as their proxy to access external Freenet nodes.

Tested client

Freenet 0.5

Ports used

There were 295 Freenet nodes in seednodes.ref at the time I tested it. These nodes used 284 different tcp ports
Default Freenet client port (FCP): 8481/tcp

Getting a current list of tcp ports used by Freenet nodes can easily be done by a grep on the seednodes.ref. An example list can be found at http://wooledge.org/~greg/seednodes.ref. There's no defined port range and no regularity. Examples of tcp ports used:

9000	31445	28025	8404
19790	19790	46104	14166
20000	18490	23904	37789
7960	31445	28025	24546
8347	15525	1888	39343
2562	57495	61126	

Counteract with a firewall

Inbound:

When acting as a server Freenet requires a service and port redirect if used behind a NAT firewall. So it shouldn't be a real problem on most corporate networks.

Outbound:

Block definitely all inbound and outbound ports not used for corporate traffic. Each blocked connection to the Freenet network makes the tool more useless. Most important, block port 8481/tcp

On Symantec Enterprise Firewall pattern matching can be enabled (see Gnutella network). Add the following lines to the httpurlpattern.cf:

.*seednodes.ref
.*freenet.exe

This block the default installer and the retrieval of the seednodes.ref
The huge number of different ports makes it hard to block Freenet completely. Nevertheless Freenet can be throttled to minimum with a firewall. So it will be not useful for mass file sharing. Since it may not be stopped completely on a corporate network, some security concerns still remain.

7 Conclusion

Many P2P utilities are easier to block than I've assumed before the analysis. They are built really only to do data sharing. That means, they contact other clients over a defined port and download respectively serve on a defined port. If a company decides that this is against the determined security policy or any other internal rule, the company can block it easily at the firewall. All that need to be done is to define a tight as possible rule set on the firewall for both inbound and outbound traffic. Companies have a given rule set on their firewalls based on their business needs. Since needs change firewall rule sets are subject to change on reasonable requests. However, some people try to get around firewall rule set by using sophisticated tricks. P2P data sharing tools have been developed which are aimed at getting data somehow illegally through a firewall. Some do that by tunnelling downloads through HTTP or by providing at least the option to do it. This is very effective, since HTTP on port 80 is open on almost every network. The Gnutella protocol offers "push routes" to fool firewalls: if an administrator restricted the download from a client by setting a firewall rule then the client is told to send the data from his side.
Since these tools show the purpose to work illegally through firewalls, I rate them as a security risk. Still I want to blame neither the P2P community nor the developer of these tools. It just shows how easy it is to trick some security devices, so the security devices have to become better.
In my test the ability to block P2P tools was relatively high since the firewall I've used was a protocol-aware proxy firewall. With the protocol awareness non-RFC conform traffic (like some tunnelled traffic) could be blocked by default. The proxy feature also stopped some P2P tools by default. By adding pattern matching http traffic was filtered. So this is a pretty easy way to limit P2P tools.

Still there remains an unpleasant feeling. If somebody finds it worth doing he will develop a P2P tool using another technique that also works behind a proxy firewall. While firewalls have to improve to keep pace with security needs it's a good idea to look at an additional method for blocking unwanted P2P. Intrusion detection systems with signatures for P2P tools would be a good extra protection. Since IDS are built for analysing traffic and signature update is a common process on these systems, it would be a more dynamic solution.

References

[01] Steve Gibson. "The Code of Backchannel Conduct ". Gibson Research Corporation
URL: http://grc.com/oo/cbc.htm (10 May 2000)

[02] Bill Beesley. "Viruses, Trojans and Spyware, OH MY". eMonitor
URL: http://www.okcpcug.org/articles/virus_trojans_and_spyware.htm (14 April 2003)

[03] Bradley Mitchell. "P2P – Peer to Peer". Computer Networking
URL: http://compnetworking.about.com/cs/peertopeer/index.htm?terms=p2p (2003)

[04] Katherine S. Mangan, "Colleges Could Face Lawsuits Over Illegal File-Sharing". The Chronicle
URL: http://mailshare.nmu.edu/listserv/network-users/msg00106.html (14. October 2002)

[05] Eileen Rivera. "Universities Stopping Students' Swapping". TechLive.
URL: http://www.techtv.com/news/internet/story/0,24195,3405877,00.html (31.October 2002)

[06] Kevin Townsend. "Spyware, Adware, and Peer-to-Peer Networks: The Hidden Threat to Corporate Security". Pestpatrol
URL: http://www.pestpatrol.com/Whitepapers/CorporateSecurity0403.asp (2 April 2003)

[07] "Spyware". Duke University
URL: http://www.oit.duke.edu/ats/support/spyware (22 March 2003)

[08] "Sandeep's Peer-to-Peer Resources Page". University of Washington
URL: http://faculty.washington.edu/sandeep/p2p/ (2001)

[09] "PacketHound". Palisade Systems
URL: http://www.palisadesys.com/products/packethound/index.shtml (2003)

[10] Scholl. "Napster Protocol Specification".
http://opennap.sourceforge.net/napster.txt (2001)

[11] "The Gnutella Protocol Specification v0.4". Griffith University.
URL: http://www.cit.gu.edu.au/teaching/6102CIT/GnutellaProtocol04.pdf
(2000)

[12] Jerome Kuptz. "Independence Array". Wired Digital
URL: http://www.wired.com/wired/archive/8.10/architecture.html (2002)

[13] "FAQ for Bearshare". Free Peers,Inc.
URL: http://www.bearshare.com/help/faq.htm (2003)

[14] "Technical FAQ". Free Peers,Inc.
URL: http://www.bearshare.com/help/faqtechnical.htm (2003)

[15] Gnutella Glossary. Lime Wire LLC.
URL: http://www.limewire.com/index.jsp/glossary (2002)

[16] "Documentation for Freenet". The Freenet Project.
URL: http://freenetproject.org/cgi-bin/twiki/view/Main/Documentation

[17] "Tech Info". Filetopia Inc.
URL: http://www.filetopia.org/index6.htm (1999)

[18] Symantec Enterprise Firewall Configuration Guide. Symantec Corporation
URL:ftp://ftp.symantec.com/public/english_us_canada/products/symantec_enterprise_firewall/manuals/7.0/sef_sevpn_70_config.pdf (2001)

[19] Symantec Enterprise Firewall Reference Guide. Symantec Corporation
URL:ftp://ftp.symantec.com/public/english_us_canada/products/symantec_enterprise_firewall/manuals/7.0/sef_sevpn_70_ref.pdf (2001)

[20] "Technical Details of the Symantec Enterprise/Raptor Firewall HTTP URL Pattern Matching Function (httpurlpattern)". Firetower, Inc.
URL: http://www.firetower.com/forum/regex.html (2003)