GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials (Security 401)"
at http://www.giac.org/registration/gsec

Peer-to-Peer (P2P) File Sharing Applications

and their

Threat to the Corporate Environment

GSEC Version 1.4b Option 1

Stephen Farquhar

© SANS Institute 2003, Author retains full rights.

Abstract:

Every system administrator will tell you that P2P applications should be banned in the corporate environment and CIOs will agree. But how many have examined the breadth of P2P technology or potential risks beyond the loss of bandwidth?

This paper examines the threat P2P file sharing applications pose to the corporate environment by looking at the evolution of P2P technology, the scope of threats posed by P2P applications, methods used to hide the presence of P2P activity, how to detect P2P activity, and how to prevent it.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Table of Contents

Introduction to P2P 3
P2P Methodologies 3
  Centralised Server – Napster 3
  Serverless – Gnutella 4
  Ultrapeers and Multi-Source Downloads – eDonkey 4
  Closed and Semi-closed networks – Direct Connect 6
Corporate Impact 7
  Confidentiality 7
  Integrity 9
  Availability 10
  Increased Expenditure 11
  Lost Productivity 12
Detection 12
Prevention 14
Conclusion 15
References 16

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Introduction to P2P

P2P stands for Peer to Peer and, at its simplest, a P2P application enables two users to exchange files without the aid of an intermediary server. The sharing process is one of giving as well as receiving so P2P applications must act as both clients and servers in the exchange process.

Prior to file exchange, users need to discover each other and the method of discovery is a large differentiator between the different P2P applications and their methods of operation.

The discovery process can include:

- One-to-One Direct: Knowing the IP address of another user and directly connecting to them.
- Serverless: Knowing the IP address of another user and connecting to them. Then learning from that user all the other users they know about. Then learning from each of those other users all of the users they also know about. This process continues recursively until a defined threshold is reached.
- Decentralised Server: In which some clients voluntarily act as discovery servers.
- Centralised Server: In which one or several static servers are the only servers that record information about the files each user is sharing.

P2P Methodologies

To appreciate the threat that P2P applications present to the corporate environment it is important to understand their methodologies and their increasingly complex transfer methods.

Centralised Server – Napster

The P2P application that has received the most media attention and raised public awareness of P2P technology, and incidentally MP3 as an audio format, is Napster.

Napster became an instant hit because it was very easy to use. The Napster application sent a list of the MP3 files a user shared to a central server. When any user made a file search request, the request was sent to the central server, and a list of users (and their IP addresses) with matching results was sent back. The searching user then connected directly to the matching file owner's IP address and requested the file, and transfer occurred.

Napster's centralisation was it's weakest link. Knowing the IP addresses of the Napster servers made it a trivial exercise to block the address on the simplest firewall, or even via a dummy hosts file entry. Napster's central listing of users, IP addresses, and shared files also made it the easiest target for the Recording Industry Association of America (RIAA) to pursue in it's efforts to curb the explosion of music piracy facilitated by Napster.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

On 3 June, 2002, Napster filed for Chapter 11 bankruptcy protection, and subsequently ceased operations in September. In mid November 2002, Napster's assets and intellectual property were sold to Roxio (- BBC News 15 Nov 02) however this has not eliminated the Napster application.

As at 9 July, 2003, the Napster network was offering 6,923 TB of MP3 files in 219.8 million files. (- Napigator 9 July 03)

The Napster network is alive and well since the threat to the original application encouraged reverse engineering of the transfer protocol specification and the development of similar client applications and servers. (- OpenNap)

Serverless – Gnutella

Gnutella is a protocol, not an application. In total opposition to the Napster methodology, Gnutella clients create a P2P network of users without (theoretically) any central servers, so each client acts as both a server and a client (dubbed a "servant").

Since the Gnutella protocol is freely available, a number of clients have been written for the Windows / Unix / Linux / Mac platforms including: Shareaza, BearShare, Gnucleus, LimeWire, Morpheus, Phex, Swapper, XoloX, Gtk-Gnutella, Mutella, and Qtella (- Gnutella)

As a serverless network, Gnutella is almost impossible to shut down since any two users can create a network simply by connecting to each other, then subsequently invite others into the network by supplying any one of the IP addresses of an existing member of the network.

The obstacle for new users of a Gnutella client is finding another Gnutella client to connect to. To solve this problem Host Cache servers have been established to record lists of Gnutella clients to connect to as a starting point. (- Gnutella Hosts)

The decentralised nature of Gnutella means simply blocking access to a known list of servers cannot prevent access. Blocking needs to occur at the port or protocol level.

Also, a weakness of Gnutella (and most P2P applications) is the extra configuration required for them to work behind a Network Address Translation (NAT) based firewall or Internet connection. Since the user must act as both a server and a client, for the server aspect to work data connections need to be initiated from outside the network. To work correctly this usually means port forwarding must be established from the NAT interface to the internal destination machine.

Ultrapeers and Multi-Source Downloads – eDonkey

The continued evolution of P2P and user demand for faster file transfer and transparent interaction with the P2P network has led to the development of two important concepts: Ultrapeers and Multi-Source downloads.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Ultrapeers exist as semi-permanent servers that assume the centralised indexing role similar to the Napster servers. Clients discover lists of ultrapeer servers from websites and upon connection to a server they learn of other ultrapeer servers that the current server knows about. Also upon connection, the client uploads a list of shared files and a hash value calculated from the file contents. The hash is significant because it allows identical files shared by different people to be identified, regardless of the filename.

Ultrapeers are created by a user running specific P2P server software separate to their client, or activating an ultrapeer function within their client software.

Multi-Source Downloads represent the current peak of efficient P2P file transfer. Since a server knows a shared file by its hash value, the file is guaranteed to be identical to other files carrying the same hash, regardless of filename.

When a client requests a file, the request is sent to the server, which subsequently passes it to every server and every client carrying the same hash. Since the file, regardless of filename or even file completeness, is identical to every other file of the same hash, the client can request different chunks of the file from each sharing user. The chunking capability is very important to efficient file transfer from multiple sources.

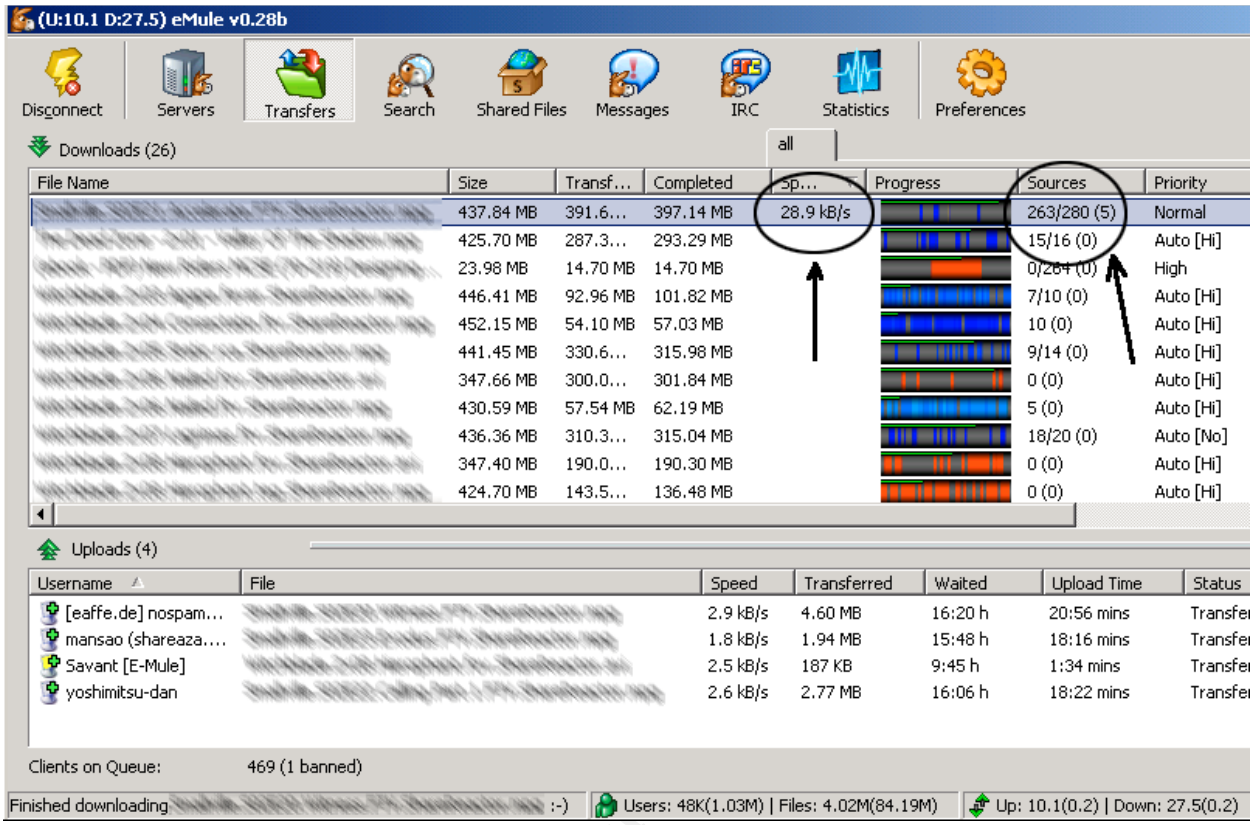
Hashing and chunking enable the client to receive multiple concurrent streams of data to the limit of the available bandwidth. Twenty file sharers offering the same file over a modem link (e.g. conservatively 4 kilobytes per second on a 56k link) can provide an aggregate download capability of 4 x 20 = 80 Kilobytes per second or 640 kilobits per second. This is 30% of a branch office 2 Megabit link.

eDonkey 2000 (- eDonkey) was the first of the new breed of P2P applications to use both of these methods. The Gnutella specification was also recently updated to include support for ultrapeers, hashing, and multi-source downloads.

In the screenshot below, eMule (a variant of eDonkey) is being used to download a number of files. These are not small MP3 files. The majority of files are in excess of 340 MB and are typical of the video files being transferred that make the speed available on corporate bandwidth very attractive.

The screenshot shows the first file being downloaded at 28.9 kB/s (230 kilobits/sec) from 5 sources (the Sources column). The Sources column also shows that 263 people have been identified as positively having some parts of this file.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



Closed and Semi-closed networks – Direct Connect

As legal action against file sharers of copyright material increases, sharers have sought means to protect themselves from scrutiny. Some P2P applications have facilitated this by adding password control for entry to the network. Direct Connect (- NeoModus) is one of these.

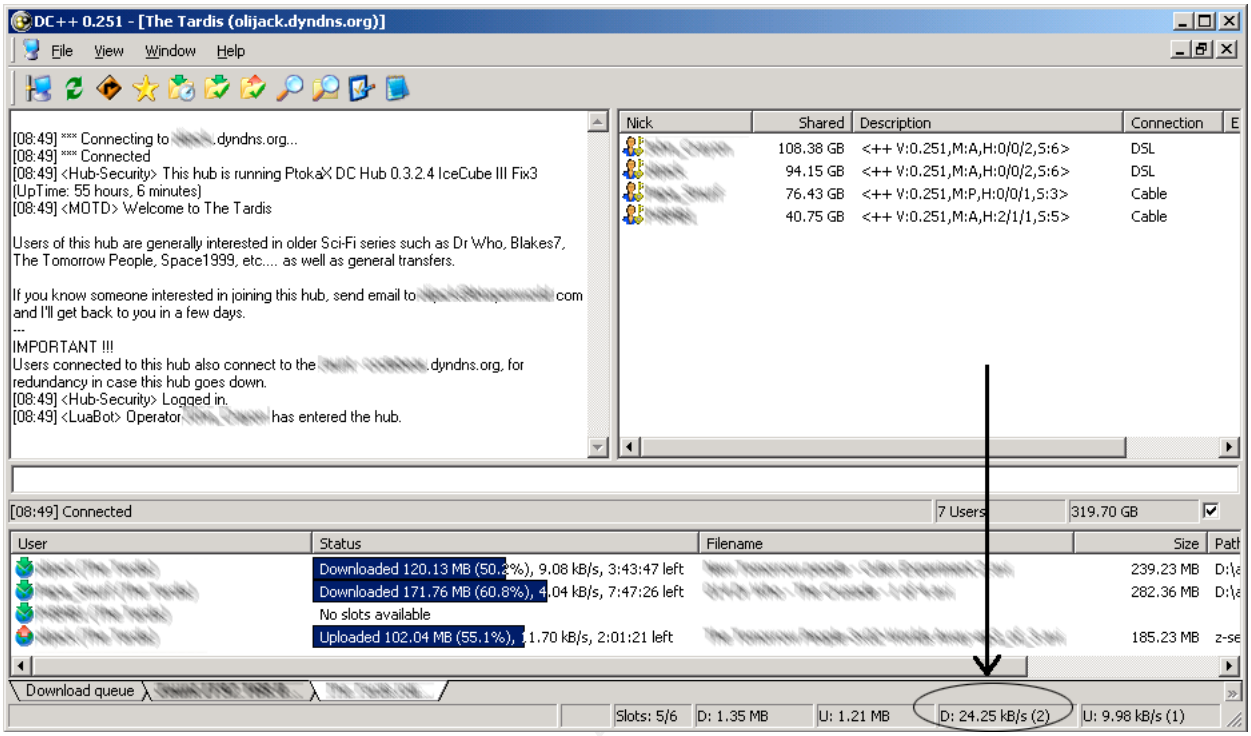
For Direct Connect (DC) to operate, at least one user in the network must run a server application (a Hub) that becomes the connection point for users. When a user connects to a DC Hub, all other users connected to that hub become visible and their files become searchable. In fact, each user's entire file list becomes downloadable.

An interesting point of differentiation from other server based P2P software is that the Hub does not store any file information about users. A Hub simply facilitates users finding and connecting to each other. The power of the Hub lies in the ability for the Hub operator to assign users passwords. A password controlled Hub effectively becomes a closed network that enables members to coordinate with a greater degree of efficiency than users of open networks.

The following screenshot shows users connected to a DC Hub. Because the Hub is closed to unauthorised users, these members have much greater control about what files they share to which people, and how the bandwidth is allocated. In this instance, the user who took the screenshot is achieving an average 24.25 kB/s (195 kilobits/sec)

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

download speed from only 2 users. On a more populated Hub this user could be downloading concurrently from as many people it takes to saturate the available bandwidth.



Corporate Impact

P2P applications running unchecked in a corporate environment have a direct impact on the foundations of IT and Security: The Confidentiality, Integrity, and Availability of data and services.

Additionally they impact corporate expenditure and productivity.

Confidentiality
To participate in a P2P network many P2P applications require a minimum amount of data to be shared before they will be functional.

Sharing the File Server in one easy step - Astute users will selectively share files, but many users accept application defaults or blindly tick the first checkbox they see. This can result in the entire contents of their hard drive being shared or worse, all drives including network drives to be shared. Hence, unwittingly exposing the contents of the corporate file server to the public becomes a minor task.

During the writing of this paper, a search was made using the Kazaa P2P software for the 81kb file “clock.avi”, and the results yielded multiple finds. This file is a standard file found in the c:\WINNT directory of Windows 2000 Professional and the positive search

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

results indicate users sharing their operating system directories. Searches on words such as “resume”, “cv”, and “policy” also produce results showing users carelessly sharing files they would otherwise consider confidential.

To compound the problem, some P2P applications also load a local web server with their client, exposing all shared files via a web interface as well as the regular P2P application interface.

An additional nuisance and threat is the installation of Adware and Spyware. Many P2P applications (e.g. Kazaa) are funded through Adware that is present as either an intrinsic part of the application, or installed as an additional application.

Kazaa’s End User Licence Agreement states:

> 7.1 Sharman reserves the right to run advertisements and promotions on the Kazaa Media Desktop.
>
> 7.2 By accepting the terms of this Licence, you agree that we have the right to run such advertisements and promotions without compensation to you.
>
> 7.3 The timing, frequency, placement and extent of advertising by us within the pages comprising kazaa.com or the Kazaa Media Desktop is subject to change and shall be determined by us at our sole discretion
>
> (- Kazaa)

Adware, especially the type installed as a separate application and runs independently of the P2P applications, is a nuisance because it downloads and displays advertising material to the user, consuming bandwidth as it downloads advertisements, and consumes user time as it interrupts the user and displays the ads.

Of greater concern is Spyware that collects and reports information on user activity back to a central source. Spyware is often installed without alerting users to its presence or asking for their consent to install.

Spyware Guide (http://www.spywareguide.com/product_list_full.php) maintains a regularly updated list of Spyware applications.

Spyware may collect information such as sites visited, applications installed, directory contents, and monitor keystrokes. Once installed, Spyware can be difficult to detect – some Spyware applications report their findings using a HTTP form submission style request. This means their reports on user activity pass straight through the corporate firewall on port 80, or via the system web proxy.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Gator offers the following about its capabilities:

> Fills in FORMS with no typing! Remembers PASSWORDS automatically. Protects and encrypts your data on YOUR computer. Only your first name, postal/zip code, and country are sent to the Gator Corporation. (- Gator)

These "features" are extremely undesirable in the corporate environment and regardless of what it does or does not collect, the fact that it collects anything at all threatens the confidentiality of corporate information.

Application Vulnerabilities – P2P applications provide a direct connection from the user machine to the outside world. The user relies on the application to ensure external requests for files are limited to those files specifically shared. If the application fails to limit access as requested then external parties can potentially access all the files the user can access including corporate data. To date, most P2P applications have had some type of vulnerability exposed including:

- Direct Connect (- NeoModus 29 Sep 01)
- Gnut Gnutella Client (- SaferMag 30 Aug 01)
- eDonkey (- SecurityFocus 6 Jun 02)
- Emule (- SecuriTeam 21 Mar 03)
- Kazaa, Grokster and Morpheus (- SecuriTeam 27 Feb 02)

Integrity

P2P applications threaten file and system integrity through direct and indirect means.

Viral Vector – P2P applications are typically created by one or several individuals working voluntarily on the application as a personal project. The application development cycle rarely includes extensive quality control and beta testing is achieved by general public release, often through download via the P2P network the application supports. The opportunity for viral infection exists at the developer's site and subsequent public release. The opportunity also exists every time the file is reshared to the network since every sharer is also a potential source of infection.

Similarly, every file in the P2P network is a potential source of infection. Since the files are transferred in either sequential or non-sequential chunks, the only opportunity to scan for infection is on the user's workstation once the file is complete. Any gateway based virus checking is avoided.

Application stability – Since many P2P applications are a personal project of the developer, the developer is often learning and experimenting with each version release. By their very nature, P2P applications are network orientated applications handling data transactions at fundamental network and file levels. The process of chunking for example involves building the file piece by piece in a non-linear manner requiring file

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

read/writes to specific sections of a file on disk. Poorly handled file locking and packet assembly can lead to data corruption on disk and in network traffic.

Additional stability problems can be caused by the third party programs that P2P applications load. When Aureate advertising Adware software is installed, it binds itself to Internet Explorer's browser activity and subsequently makes the browser unstable causing system crashes. (- OptOut)

Availability
P2P applications' use of network and files has a significant impact on resource availability.

Border Bandwidth loss – The loss of bandwidth due to P2P activity has an immediate impact on business operations and affects users in two ways: loss of Internet related bandwidth and loss of internal bandwidth.

The loss of Internet bandwidth slows email transmission and reduces web page access speed. Additionally real-time data intensive activities such as Video Conferencing and Voice-over-IP suffer if bandwidth is restricted.

An example:
Consider a 10 megabit/second corporate link serving a thousand users. Let's say one user downloading several files can achieve a download rate of 20 kilobytes per second as demonstrated in our earlier screenshots:

20 kilobytes = 160 kilobits
10 megabits = 10 000 kilobits
10 000 / 160 = 62.5

i.e. 62 P2P users (or 6% of our example user base) can easily saturate 100% of the available bandwidth.

Internal Bandwidth loss – Users don't have to be accessing the Internet to use P2P applications. Employees can easily establish P2P networks inside the network for transferring files amongst each other using Gnutella based applications or Direct Connect. This is particularly popular for employees who are geographically separated but appear on the same internal network.

Internal P2P networks can consume the bandwidth of inter-office Wide Area Networks as easily as external Internet links. Similarly, internal file exchange on the 10 or 100 megabit LAN can consume the bandwidth required for day-to-day business activities.

Hash time – For chunking to work correctly, the hash value of a file must be calculated. Remember, the hash is an identifier that uniquely identifies a particular file as having exactly the same contents of another file with the same hash. To calculate the hash, every byte of the file must be examined and processed through a hashing function. This process is extremely processor and disk intensive. Also, while it is occurring, the

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

hashing computer becomes unusable for other activities, and the file being hashed is locked.

Furthermore, if a user has inadvertently shared a network drive, hashing the contents of the network drive is equivalent to a sustained download of the entire contents of the drive to the local machine. This process consumes user time, network bandwidth, and file server resources.

Storage loss – Where is the P2P user storing his files? Downloading video files of movies and television episodes requires a lot of storage space. Consider a typical TV episode in a compressed format is approximately 250 megabytes, and a typical movie is around 650 megabytes. Corporate file server space becomes attractive for storing such large files once the user has filled their local hard drive.

Increased Expenditure

Through their consumption of bandwidth, business interruption, and potential for legal liability, P2P applications are costly.

Bandwidth costs – Depending on where you live, Internet access frequently has a cost associated with the actual amount of data downloaded. In Australia for example, most data carriers charge a fixed monthly fee for Internet access including a specific amount of data (the data cap), and an additional download fee for every megabyte of data in excess of the cap.

An example:

In this example for an organisation that has exceeded its cap, the data carrier charges an excess download fee of AUD $0.12 per megabyte.

A P2P user achieving a reasonable 20 kilobytes/second average download speed can download in one hour:

20 kilobytes x 60 seconds x 60 minutes = 72 000 kilobytes or 70 megabytes.

70 megabytes x $0.12 per megabyte = $8.40

If the user sustains his download speed, during 8 business hours per day, for 20 working days per month by queuing for download all the episodes of “Buffy the Vampire Slayer”, the business incurs a cost of:

$ 8.40 x 8 x 20 = $ 1 344 per month or approx $ 16 000 per year.

Copyright violation and liability – Few files downloaded using P2P applications are legitimate. The files are usually pirated music, video, or software. Any organisation found to be housing these files, transmitting them, or using unlicensed software is exposed to expensive legal liability.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Lost Productivity

P2P applications reduce the productivity of the users that use them, and the productivity of staff who are required to cleanup after them.

User Productivity – User productivity is adversely affected through time wasted searching for files. Additional time is wasted handling files once they have been downloaded. Most users will want to take music and video files home after downloading, which means a transport mechanism such as a Laptop, CD, USB Keydrive, or memory card (e.g. in a PDA) needs to be used.

Adware time and interruption – Adware associated with P2P applications can interrupt user tasks even if the P2P application is not running, and further distract the user if the advertisement is of interest.

Cleanup costs – When P2P applications are identified, they need to be uninstalled along with accompanying Adware products. Rather than uninstalling, the best solution is often to reinstall the Standard Operating Environment image if one exists. In either case, staff time is unnecessarily wasted.

Detection

The following list is a guide to the many items to check for the presence of P2P applications on the network.

Increased Internet Bandwidth Usage: Whether you have real time bandwidth logging, or only look at the end of month Internet charges, a sudden increase in bandwidth usage can indicate P2P activity.

Increased Internal Bandwidth Usage: Similarly, sudden and uncharacteristic increases of internal network activity can indicate P2P activity.

Unusual boundary port activity: Firewall logs should only show activity for permitted services. e.g. Port 25 for SMTP services, Port 80 for web services, etc … Logs should be processed searching for port activity outside the expected ranges.

Unusual internal port activity: A packet sniffer used inside the network can be used to detect unusual traffic related to P2P activity.

Proxy logging and detecting tunnelled traffic: To bypass the firewall, a number of applications exist that allow P2P software to tunnel via the web proxy server. They typically work by installing a local Socks firewall product that intercepts P2P traffic originating from the client, encapsulating it in a HTTP packet, or an SSL packet, and forwarding it through the web proxy to a specific receiver outside the firewall.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

The receiver opens the packet and forwards the contents to the network as though it was the originator. When a reply is received, the receiver encapsulates the reply as a HTTP/SSL response and passes it back to the corporate proxy server which in turn returns the packet to the client.

HTTP Tunnelling products include:

- HTTPort (http://www.htthost.com/index.htm)
- Socks2HTTP (http://www.totalrc.net/s2h/index.jsp)
- HTTP-Tunnel (http://www.http-tunnel.com/HT_Products_HTTPTunnelClient.asp)
- KazaaHTTP (http://www.iprisma.com/kazaahttp/)

Detecting traffic tunnelled through your web proxy can be very difficult, especially if it is encrypted. Few products readily claim to be able to identify tunnelled traffic, however Packeteer states:

> PacketShaper also identifies P2P that tunneled through HTTP tunnel gateways. (- Packeteer)

and Allot claims:

> In Q2, NetReality will deliver WiseWan functionality to thwart savvy users who try to beat corporate policies by hiding peer-to-peer transactions via a mechanism called HTTP Tunnel. (- Allot 11 Mar 02)

A more immediate method of identifying tunnelling activity is simple log analysis. The volume of data downloaded through P2P activity is generally significantly larger than regular web traffic.

Assuming you have reasonably static internal IP addresses (or long DHCP leases), a proxy log report of data downloaded by IP addresses should identify unusually large download activity by IP.

If users are required to authenticate with the proxy, then regardless of IP address, your proxy logs should be able to show data volumes downloaded per user.

Excessive data usage identified in a proxy log should always be investigated regardless of whether P2P activity is suspected.

Internal scanning: Regularly scanning your internal network for unauthorised open ports can also detect unauthorised network applications including P2P applications, web servers, and Socks proxy applications.

File type and size scanning: Scanning the file server for files of unusual types and sizes will identify suspicious files. P2P files typically have extensions of avi, mpg, mp3, zip, and rar, and are usually greater than 3 MB for mp3 files, greater than 50MB for music collection archives and video files.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Prevention
The task of preventing the use of P2P applications in the corporate environment is a subset of the task of preventing any unauthorised software usage and starts with policy, followed by a variety of techniques to form multi-layered defences.

Minimum user rights – The first place to start with the prevention of P2P application usage in your environment is to prevent users from installing them in the first place. Minimum user rights should restrict non-privileged users from installing software on their computers.

Proxy authentication – Locking down the equipment you control does not prevent staff or contractors bringing in their own equipment e.g. laptops. Ensure access to your proxy server requires authentication to prevent unauthorised access to P2P networks via http tunnelling.

Minimal firewall openings – Your firewall should prevent access to all services except those that you specifically permit. However, if your corporate policy is to allow all services and only deny those services that are deemed a threat, you will need to block many TCP and UDP ports to cover the range of popular P2P applications such as:

Direct Connect	411, 412
KaZaA	1214, 1285, 1299, 1331,1337, 3135, 3136, 3137
Gnutella clients	6346, 6347
WinMX	6257, 6699
eDonkey	4661, 4662
Filetopia	5665

This list is by no means exhaustive, and many P2P applications now enable users to define which ports they wish to use, effectively circumventing a list of specifically blocked ports.

File Server quotas – implementing per-user storage limits on the file server will prevent users storing excessive P2P sourced files, and provide an alert if someone attempts to exceed a quote limit.

Proxy blacklists: Most proxy servers support blacklists or have add-on software available to enable the blacklisting of sites that the organisation never wants the user to visit or have access to. Ensure that you blacklist access to sites that support P2P services the common Adware sites such as gator.com.

If your proxy server does not directly support blacklists, a similar functionality can be achieved if you run an internal DNS server that checks its own hosts file before performing lookups. In that hosts file, add dummy entries pointing to non-existent IP addresses, or the localhost (127.0.0.1) address for the sites you want to block.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Free blacklists are available to download for blocking a large number of annoying sites. These lists usually include most Adware and Spyware sites. Check:

- A2Z Communications Blacklist - http://www.a2zcomms.com/blacklist/index.shtml
- DansGuardian - http://blacklist.dansguardian.org/
- Dan Pollock's List - http://someonewhocares.org/hosts/

Conclusion

P2P applications have evolved from simple, centralised, music sharing services to complex, decentralised, file exchange mechanisms. The current crop of P2P applications are capable of exposing corporate information, damaging data, consuming resources, and stealthily tunnelling straight through the firewall and proxy server.

The financial losses P2P applications can cause range from the calculable costs of lost productivity to the incalculable costs of civil and criminal legal liability.

Tackling the P2P threat requires detecting their presence and activity through a variety of mechanisms. These include a high level view of unusual data activity to a low level inspection of active ports on individual machines.

Prevention requires a multilayered defence starting at the user desktop and working out to the edge of the network.

P2P applications will continue to evolve their capability and efficiency of operation, seeking new ways to hide their activity and bypass corporate security measures. Like virus infections and intrusion attempts, IT staff must remain alert and informed about the ongoing threats posed by P2P applications.

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

References

BBC News. “Napster set to be sold to CD burner”. 15 November, 2002. http://news.bbc.co.uk/1/hi/business/2480047.stm

Napigator Online Servers list. 9 July, 2003. http://www.napigator.com/servers/

OpenNap: Open Source Napster Server. http://opennap.sourceforge.net/

Gnutella. “Connect”. http://www.gnutella.com/connect/

Gnutella Hosts. IPs / Hosts / Servers. http://gnutella.da.ru/

eDonkey. http://www.edonkey2000.com/

NeoModus. http://www.neo-modus.com/

Kazaa. End User License Agreement. http://www.kazaa.com/us/terms.htm

Gator.http://www.gator.com/

NeoModus. Security Update. 29 September, 2001. http://www.neo-modus.com/?page=News&start=10

SaferMag. Gnut Gnutella Client Arbitrary Script Code Execution. 30 August, 2001. http://www.safermag.com/html/safer40/alerts/06.html

SecurityFocus. eDonkey 2000 URI Handler Buffer Overflow Vulnerability. 6 June, 2002. http://www.securityfocus.com/bid/4951/discussion/

SecuriTeam. Emule Remote Crash. 21 March, 2003. http://www.securiteam.com/unixfocus/5VP0T0A9PS.html

SecuriTeam. Kazaa, Grokster and Morpheus Remote Denial of Service. 27 Feb 02 http://www.securiteam.com/exploits/5RP0R1P6AC.html

OptOut. Gibson, Steve. Aureate Crashes Windows and Browsers. 8 May, 2000 http://grc.com/oo/aureatemail.htm

Packeteer. Uncontrolled Peer-to-Peer Traffic?. http://www.packeteer.com/solutions/problems_peer-to-peer.cfm

Allot. Attention Mp3 Fans: Leave Your Inner Deejay At Home. 11 March, 2002. http://www.allot.com/html/pr_netreality_mar_11_02.shtm

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.