Security And Peer-To-Peer Applications

David Piscitello

You probably can't keep P2P apps off your network—and you might not want to. But you will need to redouble your security efforts.

Accurately or not, Napster will be remembered for proving that the Internet could support decentralized information-sharing applications alongside traditional "client/server" applications like the World Wide Web. Millions of Internet users downloaded Napster's software to share new as well as copyrighted music files. From May 1999 until July 2001, when operations were ceased in compliance with a court injunction filed by the record industry, nearly 2 billion files were exchanged.

Similarly, America Online will be remembered for creating instant messaging (IM). Purists may argue that AOL's Instant Messenger (AIM) wasn't the first interactive messaging service, but AIM has captured the attention of more than 40 million users.

Napster and AIM represent paradigm shifts in Internet use and user behavior. Both satisfy an "instant gratification" society and, moreover, both have demonstrated that every computer in the Internet is neither strictly client nor server, but potentially both. Not remarkably, dozens of Napster alternatives—from Kazaa to Gnutella to Morpheus—and AIM wannabes from Yahoo!, MSN (.NET), Netscape and ICQ—have seized on this peer to peer (P2P) model for both consumer and enterprise applications and networking.

David Piscitello, president of Core Competence, Inc., is an internationally recognized expert in security technology and founder of the Internet Security Conference. He has chaired the Networld+ Interop Conference Program committee since 1996.

Characterizing Peer To Peer

Columnist, consultant and author Clay Shirky characterizes P2P as "a class of applications that takes advantage of resources…available at the edges of the Internet." Moreover, Clay explained, "accessing these decentralized resources means operating in an environment of unstable connectivity," and that "nodes must have significant or total autonomy from central servers." Let's look a little more closely at some of these terms.

Decentralized resources include file storage, content, processing cycles and the interfaces and media that engage the person reading, typing, speaking and listening at every PC, irrespective of location and means of connection to the Internet. Music and messaging currently dominate P2P, but desktop collaboration, remote administration, voice and video are also potential forms of P2P interaction. Peer to peer should and will embrace every form of communication that can conceivably be made interactive.

By **unstable connectivity**, we mean that P2P applications make no assumptions about the "always connected" nature of any element in the distributed system; neither do the applications make any assumptions about bandwidth or latency characteristics of the connections.

In fact, P2P participants today are commonly dialup, which means they typically have temporary or unpredictable IP addresses. And the situation will only get more complicated: As wireless use and support for persistent sessions independent of the underlying transport grow—in short, as mobility increases—the IP address may actually change in the middle of an interactive P2P session, as the mobile user "roams" from one location to another.

Lastly, many P2P applications for the public sector seek autonomy not only from servers in the traditional sense of client/server as we understand it from the World Wide Web, but from *any* centralized server. Those who concentrate on the civil liberties aspects of connectivity worry about centralized servers that might contain information which compromises the freedom of speech and anonymity of the application users; the fear is that these servers might be seized by a government and/or ordered to cease operations.

Examining these characteristics in detail tells us a great deal of what P2P applications are and do, and gives us insight into the potential security issues P2P applications introduce.

P2P Security Threats

P2P applications are popular, and instant messaging, like wireless GSM text messaging and alphanumeric paging, appears to have a legitimate business application. While appealing in many

"consumer" respects, however, P2P applications can be disruptive and dangerous to your business organization. The most worrisome security threats include:

■ **Copyrights and intellectual property infringements**. Music, software, professional photos, books and e-books, even motion pictures, can be shared and downloaded to your organization's computers, in violation of some copyright law.

■ **Bandwidth misuse**. P2P application file transfers can be queued and run simultaneously. Many simultaneous downloads by P2P users can easily sap bandwidth away from legitimate business applications.

■ **Violations of criminal law**. Some argue that P2P applications, especially those like FreeNet that advocate free speech and anonymity, facilitate and even encourage the sharing of pornography, especially child pornography. The outcome of such arguments notwithstanding, every organization must consider whether it can be held accountable for storage and distribution of material deemed illegal by government authorities. Organizations should also be cognizant of how the recently-enacted Patriot Act makes it a federal crime to provide computing and network resources used in the commission of a terrorist act, and the potential for P2P applications to enable criminal activity as outlined in this law.

The features that make P2P appealing to the masses also fly in the face of best security practices

■ **Spyware and adware**. Certain P2P client software gathers usage and behavior patterns from the computer where it is installed. This information is reported back to vendors for advertising purposes. Since many users simply "OK" default settings and licenses during installation, it is quite likely that your users have installed adware without realizing it.

■ **Indiscriminate file sharing**. P2P applications rely on the majority of participating computers to make music and other files available for retrieval by other participants. Whether from mis-configuration or user *naïveté*, unrestricted file sharing privileges can lead to unauthorized disclosure of sensitive information and unauthorized access (for example, when files reveal accounts and passwords).

Inappropriate file sharing can also lead to violations of privacy rights of individuals in your organization. For example, it's not difficult to imagine a receptionist in a medical practice accidentally sharing a folder containing patient files.

Users not only make files available, but receive them from others as well. A hidden danger here is that an attacker overwrites your shared music, video or document with malicious code (Trojan programs, viruses, remote administration and keystroke logging tools). The next time you go to listen, watch or view your own resource, you unwittingly execute the malicious code instead. As you know, the most dangerous mail attachments are those that somehow get you to execute them. With P2P, the files being shared often have defined "open" actions that create convenient vectors for execution. Unobserved and unmonitored, attackers can more easily distribute malicious code than through email attachments.

■ **Information and identity disclosure**. Certain P2P applications (AIM, Yahoo! Messenger) have weak or easily cracked measures to protect user identities and passwords. If the identity or password is used for other access purposes, e.g., a Windows OS or extranet login, identity theft of this nature can be quite serious. MSN Instant Messenger users may *think* they're being safe by entrusting identity management to Microsoft's Passport services, but Mark Slemko's highly publicized exposure of the flaws in Microsoft's Wallet function strongly suggests users may not be as safe as they think.

Other threats can be directly associated with the P2P application protocols themselves. The protocols for the major instant messenger clients, and "peering" software for Kazaa, Gnutella, Morpheus, *et. al.*, are either publicly available or have been reverse engineered. Dozens of versions of peering software have been written—much of it non-commercial shareware. Security features are rare and generally weak: Thorough vulnerability testing by the author/vendor is rare, none of the major IMs encrypt communications and all have weak or no authentication. The complexity and frequency of version changes supports the security community's concern that numerous vulnerabilities exist in much of this software.

So countering these threats becomes a major challenge. The ease with which software can be downloaded makes P2P applications extremely difficult to ban outright in enterprise networks that don't implement near-draconian policies concerning downloadable executable content. Many organizations find it difficult to keep client/peering software from working its way into corporate networks through employee home networks.

Designs That Confound Security

The very design objectives that make P2P applications appealing to the masses—decentralization and anonymity—fly in the face of best security practices for enterprise networks. The

decentralized operation of many P2P applications makes them difficult to block at Internet firewalls; the traffic that many P2P applications generate is often indistinguishable from "Web traffic." Yahoo! Messenger, for example, can be configured to operate over the same port and even protocol as the Web (HTTP Port 80), which is commonly allowed "outbound" or to the Internet through corporate firewalls.

Unlike email, where service providers and enterprises can set upper bounds on attachment sizes at the mail server, the file-transfer capabilities of P2P applications set no restrictions whatsoever. Such file transfers can bypass any antivirus measures an organization may have associated with email (and Web) servers.

P2P's decentralized data storage paradigm compounds an already frustrating data management situation for many enterprises. Too much authorized data already resides on the laptops of mobile employees and teleworker computers, outside the network manager's direct control. Then, the resource-sharing paradigm employed by many P2P applications buries important data amid dozens if not hundreds of files of questionable value and origin, and finally the prospect of archiving files that infringe on intellectual property and copyrights confounds an already Herculean task.

Ironically, P2P applications wind up creating the exact opposite effect from what companies hope to implement through their deployment of network-attached storage and storage area networks: Where SANs and NAS management applications seek to consolidate information and to eliminate redundant, irrelevant and inaccurate copies of documents, P2P applications replicate files at any location, without an audit trail and without validity or integrity checks.

Noteworthy P2P Security Incidents

Many of the security incidents associated with P2P applications involve viruses and worms (Table 1). Worms have spread through the Gnutella, Kazaa and Grokster networks, including VBS.gnutella, W32.HLLW.Electron and W32.Shermnar.Worm. Mandragore, aka W32.gnuman, was significant enough to garner attention from CNN. Worms such as these demonstrate how infected files could be spread through peer-to-peer networks as quickly as via email, if not quicker.

More worrisome are worms that carry Trojan or back-door programs. W32.HLLW.Kazmor (Kazaa) and W32.Evala.Worm (Kazaa, Gnutella, and Grokster) aided attackers in gaining control of compromised computers. W32.Elem.Trojan (Kazaa) destroyed files on a compromised computer.

Antivirus experts at Symantec, Trend Micro and McAfee have also identified viruses designed to hijack user identities and passwords, inflict denial of service (DoS) attacks or create mass mailings for several IMs. The archives of security mailing lists where vulnerability advisories are reported (e.g., SecuriTeam, Bug-Traq) identify scores of vulnerabilities that include arbitrary file

Worms have spread through P2P networks

TABLE 1 P2P Security Incidents

Vulnerability/P2P Application	AIM	MSN Messenger	Yahoo! IM	Kazaa	Morpheus	ICQ
Buffer Overflow Allows Remote Code Execution		May 2002				
Trojan Horse Causes Malfunction		August 2002				
Mass Mailing Worm	April 2002	April 2002 January 2002	June 2002	July 2002	July 2002	June 2002
Trojan Horse Allows Attacker to Gain Control of a Computer	August 2002			June 2002		
Internet Worm Displays Bogus Error Messages		August 2001				
Internet Worm Deletes Files, Folders, Formats Drives		April 2002		July 2002		
Internet Worm Deletes and Modifies Files, Causes System Instability		April 2002		Nov 2002 August 2002		August 2002
Trojan Horse Steals Passwords		August 2002 Dec 1997				
Trojan Attempts to Disable Antivirus Software				July 2002		
Trojan Attempts to Disable Personal Firewall Software			June 2002			June 2002
Denial of Service Attack Zombie				July 2002		
Worm Propagates Through File Sharing Service				June 2002	August 2002	

creation, arbitrary email reading, message sniffing and execution of arbitrary code.

One example should suffice: Internet Security Systems' X-Force reported that "a bug in the AOL Instant Messenger (AIM) client allowed an attacker to overflow internal buffers by sending URLs constructed in a certain manner. In this situation, it was possible to crash the client, obtain control of the AIM program, execute arbitrary code and add random buddies to a buddy list via a Web page or email." The majority of these IM clients are written with a "haste-to-market" mentality, undergo no secure code review and are full of flaws.

P2P Security Measures

Security-minded individuals may react to risks associated with P2P applications by saying, "There's nothing radically new and threatening here." This is partly true. None of the vulnerabilities and the exploits against P2P applications are *new*. The problems lie in the nature of P2P applications themselves.

Security for P2P applications is not only woefully inadequate, but difficult to retrofit

Journalist Michael Hurwicz aptly sums up the P2P security situation: "A major focus for P2P technology has been defeating security restrictions. Its success on this front demands reassessment of corporate security infrastructure, particularly perimeter security based on firewalls." P2P applications are a new avenue into your corporation that attackers will exploit if left unattended.

Many companies are attempting to legitimize P2P applications for enterprises (Groove, Authentica, Microsoft, Sun Microsystems, IBM, WebEx) by implementing stronger authentication, authorization, access controls and encryption in desktop collaboration, document access and other "emerging" business-enabling P2P applications. Such steps are encouraging, but they don't address the problem that security administrators face today, where employees use public P2P networks and software. To deal with the most immediate concerns, security managers should concentrate on the following areas:

■ **Policy—**There are measures every organization should consider to protect against exploits via P2P applications. Like every security initiative, this one begins with policy and awareness.

Decide whether you will allow P2P applications. If you choose to prohibit them entirely, consider how your organization can best accomplish this, and whether you are willing to make the considerable effort a complete ban may require. Document this and related decisions in your security and acceptable use policies, and identify disciplinary consequences for violations of policy. Have your policy carry some clout, and require employees to sign and acknowledge that they have read and understand the policies you've established.

■ **Software Control—**To truly prevent P2P applications from operating in your networks, you may find it necessary to block users from installing software on systems. Centrally manage and catalog all desktop software, and audit systems regularly. This is a considerable undertaking, however, so take time to determine if this is an acceptable cost when weighed against the risks identified in this and other articles on this subject.

Install antivirus software and maintain virus definitions on all systems. Many P2P applications bypass antivirus and other content inspection gateways, so even when gateways are installed, desktop antivirus measures continue to be an important line of defense. Use personal firewalls on all client systems; in particular, look for personal firewalls that allow centralized policy control and lockdown, and block access to all services not expressly permitted in your security policy. Personal firewalls that include intrusion detection, logging facilities and desktop integrity software (PestPatrol, TripWire) can be very helpful in verifying that your security policy remains implemented at all client hosts.

■ **Access Controls—**Weakly authenticated and unrestricted access to file-sharing services rank among the most prevalent sources of network-based attacks. Most users don't understand the dangers inherent in enabling file sharing. Employees should be provided with guidelines for granting access to individuals and groups. Network vulnerability scanning can help identify systems that expose sensitive and commonly attacked file shares.

If you must use P2P, then use it in conjunction with file systems that offer access controls (e.g., Windows New Technology FileSystem) and define the most stringent access possible to only those folders and files necessary.

■ **Perimeter Defenses—**Many P2P applications and networks operate with assistance from relay and rendezvous servers, and the domain names, IP addresses and ports used are well documented. Be certain that inbound and outbound access to all known P2P services and servers is blocked at Internet firewalls; if your security policy is "that which is not expressly permitted is prohibited," then your firewalls should already be blocking these services.

The noteworthy exception here is Port 80/HTTP outbound. Several P2P applications use this port. Create a "blacklist" of relay and

rendezvous servers to which all outbound and inbound traffic is prohibited (to be effective, this list must precede your default "allow HTTP outbound"). Logging denied attempts to connect to P2P applications at firewalls will help you identify unauthorized P2P application users.

If you are keenly worried about P2P applications, you may find it necessary to use LAN analysis to capture and subsequently analyze traffic passed through Port 80 (HTTP) to determine if employees are using P2P applications that tunnel traffic over HTTP.

Conclusion

Peer-to-peer applications and networks may represent a new and valuable paradigm for business applications. P2P applications used today by the general public clearly illustrate the power of this networking paradigm, but security appropriate for enterprise applications is not only woefully lacking, but difficult if not impossible to retrofit and equally difficult to remedy by applying conventional security measures. While the threats are real, only a careful risk analysis by your organization will help you determine how to deal with peer-to-peer applications in your network□

Companies Mentioned In This Article

America Online (www.aoltimewarner.com)
Authentica (www.authentica.com)
Endeavors Technology (www.endeavors.com/)
Gnutella (www.gnutella.com)
Grokster (www.grokster.com/)
Groove Desktop Collaboration Software (www.groove.net/)
IBM (www.ibm.com)
ICQ (www.icq.com)
Internet Security Systems (www.iss.net)
Kazaa (www.kazaa.com)
McAfee (www.mcafee.com)
Microsoft (www.microsoft.com)
Morpheus (www.morpheus.com/)
PestPatrol (www.pestpatrol.com)
Project FreeNet (www.sanity.uklinux.net/freenet.html)
Sun Microsystems (www.sun.com)
Symantec (www.symantec.com)
Trend Micro (www.trendmicro.com)
TripWire, Inc. (www.tripwire.com)
WebEx (www.webex.com/home/default.htm)
Yahoo! (www.yahoo.com)