



Malicious Threats of Peer-to-Peer Networking

by *Eric Chien*, Symantec Security Response

INSIDE

- › Background Protocols
- › New Vector of Delivery
- › Malicious Uses of Peer-to-Peer Networks
- › Detection of Threats
- › Privacy Concerns
- › Future

Contents

Abstract	3
Background Protocols	3
Gnutella	3
Napster	4
Freenet	6
New Vector of Delivery	8
Malicious Uses of Peer-to-Peer Networks.	8
Detection of Threats	9
Privacy Concerns	10
Future	10
Summary	11
About the Author	11

> **Abstract**

Peer-to-peer networking allows communication between two systems, in which each system is considered equal. Peer-to-peer networking is an alternative to the client-server model. Under the peer-to-peer model, each system is both a server and a client, commonly referred to as a *servent*.

Peer-to-peer networking has existed since the birth of computing networks. Recently, however, peer-to-peer networks have gained momentum with searchable peer-to-peer network file databases, increased network connectivity, and content popularity.

This paper will discuss the malicious threats, privacy concerns, and security risks of three common peer-to-peer network systems that are gaining popularity today. The malicious threats discussed will include how malicious threats can harness existing peer-to-peer networks, and how peer-to-peer networking provides an additional (potentially unprotected) vector of delivery for malicious code.

Each protocol will be discussed, as well as the pros and cons of such models in regard to privacy and potential security risks by their usage.

The systems discussed include the Napster, Gnutella, and Freenet protocols. These protocols will be examined due to their popularity and different methods of achieving peer-to-peer networking.

Many other peer-to-peer networking systems exist (for example, Microsoft Networking), and while not explicitly discussed, conclusions can be applied to these systems as well.

> **Background Protocols**

GNUTELLA

Gnutella does not utilize a centralized server. Each computer is a client as well as a server, hereinafter called a servent. Such a true peer-to-peer networking model decreases reliability, speed, and search capabilities, and increases network traffic. Figure 1 illustrates the standard communication process involved in obtaining a file.

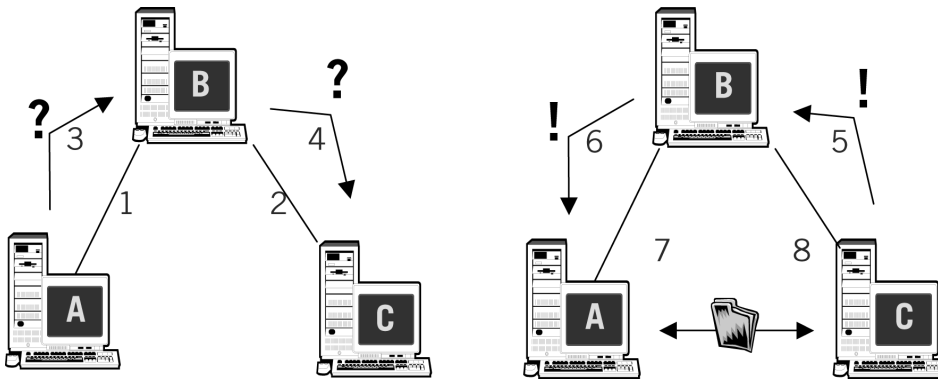


Figure 1 Standard Communication Process: Obtaining a File

1. Servent A connects to Servent B.
2. Servent C connects to Servent B.
3. Servent A sends a file name query.
4. Servent B searches for local data query matches. With no matches, Servent B forwards the query to Servent C.
5. Servent C searches for local data query matches. With a found match, Servent C responds with a query hit to Servent B.
6. Servent B passes back the query hit to Servent A.
7. Servent A connects directly with Servent C to download the file.
8. Servent C passes the file to Servent A.

If Servent C is behind a firewall and unable to receive uninitiated connections, a message can be passed via the network to have Servent C initiate the connection to Servent A, effectively “pushing” the file.

In the Gnutella protocol, you must have information on the location of a servent before you can provide the initial link into a Gnutella network. Such servent listings are not part of the Gnutella protocol. Many directory services exist for finding nearby servents.

NAPSTER

The Napster peer-to-peer networking model involves a centralized directory server. Clients primarily communicate with a directory server that passes messages among, and maintains particular states of, clients. Figure 2 illustrates the standard communication process involved in downloading a file in the Napster protocol.

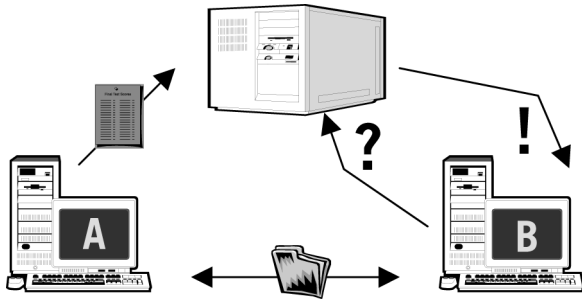


Figure 2 Standard Communication Process: Downloading a File in Napster Protocol

1. Client A logs on to the Server.
2. The Server responds with a success message.
3. Client A sends the file names of files available for sharing.
4. Client B logs on to the Server.
5. The Server responds with a success message.
6. Client B sends a search message for a particular file name to the Server.
7. The Server responds with a listing of available clients with file name matches.
8. Client B sends a download request to the Server for a file located on Client A.
9. The Server responds with detailed information about Client A, including the IP address and listening port.
10. Client B connects to Client A and sends the file request.
11. Client A responds with the file.

Note: If Client A is unable to receive uninitiated direct connections due to a firewall, the steps differ starting at step 8.

8. Client B sends an alternative download request to the Server for a file located on Client A.
9. The Server sends a message to Client A to initiate the file transfer to Client B.
10. Client A connects to Client B and transfers the file.

New versions of Napster also have client-to-client browsing.

1. Client A logs on to the Server.
2. The Server responds with a success message.
3. Client B logs on to the Server.
4. The Server responds with a success message.
5. Client B sends a browse request to the Server for Client A.
6. The Server sends a message from Client B to Client A, requesting browse ability.
7. Client A responds with a browse request accepted message.
8. The Server sends detailed information about Client A, including the IP address and port information, to Client B.
9. Client B sends a browse request to Client A.
10. Client A responds with a shared file listing.
11. Client B connects to Client A and sends the file request.
12. Client A responds with the file.

This potentially limits Napster's liability and prevents central server blocking or filtering by file name. However, clients still must register themselves with the central directory service.

FREENET

The Freenet model of exchange is similar to Gnutella, being a true peer-to-peer model. However, users do not have control over what content is held in their shared space, known as a DataStore. A user inserts a file into the Freenet network, where it is encrypted and propagated along the network to an appropriate node determined by a unique key, which identifies the file.

This allows data with close keys to be sorted to the same nodes on the network, which causes the clustering of close key data. This allows a fast response to search queries. Since all data is encrypted, users only have control of the amount of space they wish to make available on their systems, not the content that resides on their systems.

To insert a file into the Freenet network, users need only to provide the unique key, which identifies the file.

The encryption of all data on the Freenet network prevents the identification, filtering, or blocking of content that is propagated throughout the network. In addition, blocking certain content on your system is difficult, as no one has control over the content residing on users' systems.

Figure 3 illustrates the standard communication process involved in obtaining a file in the Freenet protocol.

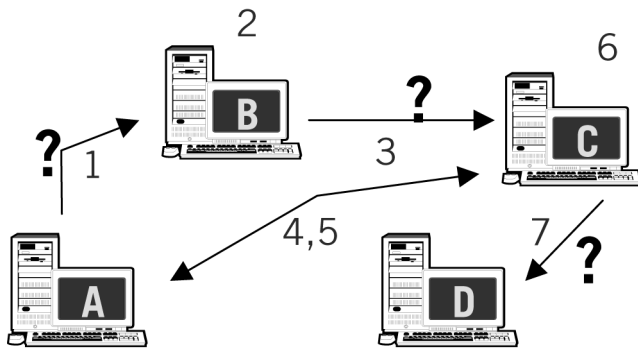


Figure 3 Standard Communication Process: Obtaining a File in Freenet Protocol

1. Servent A requests a file from Servent B.
2. Servent B checks which nearby servent has the closest key. Servent B discovers that Servent C has the closest key.
3. Servent B requests the file from Servent C.
4. Servent C checks which nearby servent has the closest key. Servent C discovers Servent A has the closest key.
5. Servent A, which originally initiated the request, notifies Servent C.
6. Servent C checks which nearby servent has the next closest key. Servent C discovers that Servent D has the closest key.
7. Servent C requests the file from Servent D.

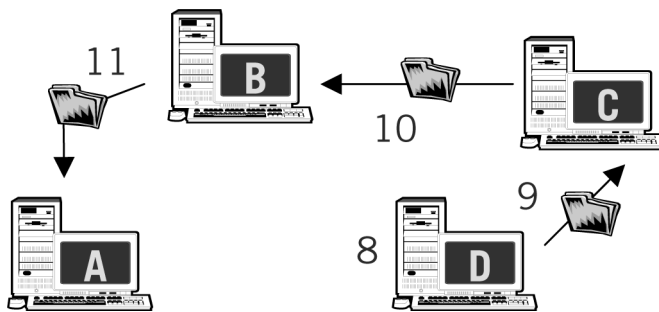


Figure 4 Standard Communication Process

8. Servent D has a matching key for the requested file.
9. Servent D responds to Servent C with the file.
10. Servent C responds to Servent B with the file.
11. Servent B responds to Servent A with the file.

> **New Vector of Delivery**

Peer-to-peer networking introduces an additional vector of delivery. In the past, the primary method of contracting a virus was via a floppy disk. The floppy disk drive was the primary vector of delivery. Today the primary vector of delivery is email. Malicious software is commonly found as email attachments.

Peer-to-peer networking provides another method of introducing malicious code to a computer. This additional vector of delivery is currently the greatest threat that malicious software has on peer-to-peer networking.

While peer-to-peer networking systems allow you to introduce executable files to a computer, those files still require specific downloading and execution. For example, a user of Gnutella may search for ExampleVirus and a server may return a match of ExampleVirus.exe. To become infected, the Gnutella user must request a download of that file from the remote server and execute the file.

Therefore, classic peer-to-peer unaware viruses could inadvertently be transmitted via a peer-to-peer network. Viruses could also take advantage of the regular use of a peer-to-peer network. For example, viruses could specifically attempt to copy themselves to or infect files within the shared peer-to-peer space.

The first discovered Gnutella worm, VBS.GWV.A, does this by copying itself to the Gnutella-shared directory as a popular file name. For example, the worm may copy itself into the Gnutella-shared directory as Pamela Anderson movie listing.vbs. The goal is to trick users into downloading and executing the file.

Viruses could actually harness the existing peer-to-peer network infrastructure to propagate themselves. For example, a worm could set up a server on an infected system. The user with the infected system does not have to initially be part of the peer-to-peer network. Then, this server could return the exact matches for incoming search queries, and those downloading and executing the file will in turn become infected. An example of such a worm is W32.Gnuman.

> **Malicious Uses of Peer-to-Peer Networks**

The use of peer-to-peer networks allows not only the ability for malicious software to propagate, but utilization of the protocols for communication by malicious software.

In many organizations, Backdoor Trojan Horses, such as Back.Orifice, are not effective in infiltrating an organization due to a firewall. Such programs open listening connections, waiting for a client outside of the organization to connect. Because firewalls prevent incoming connections, except to particularly defined machines and ports, the machines remain uncompromised.

However, the firewall does not generally block the peer-to-peer software, as it makes outgoing connections to the centralized directory services or other servers. Outgoing connections generally are not blocked. Once an outgoing connection is made, the centralized directory service or a server can pass information to the client.

The majority of current Backdoor Trojan Horses do not make such outgoing connections because they would need to connect to a defined awaiting server. When they are discovered, this may lead to the identification of the malicious hacker. Some Backdoor Trojan Horses avoid this scenario by making outgoing connections to IRC or similar centralized services. W32.PrettyPark is an example of a worm that creates an outgoing connection to IRC (and therefore, the common firewall configuration does not block it). Once the worm is connected to IRC, hackers can join the same channel and send remote access commands.

Such methods could be conducted using a peer-to-peer network as well. For example, a malicious threat could register with the Napster centralized server and pass a specific, unique list of files. Then, a hacker would perform a search on those specific files, and when matched, he or she would be able to identify any infected systems. A request for a certain file would signal the infected machine to perform a particular task, such as taking a screen shot. Information gathering and system control of the system could then be performed in this manner, bypassing the firewall and ensuring the anonymity of the hacker.

In addition, malicious software could easily change the configurations of existing peer-to-peer clients. For example, a Trojan Horse could modify the settings so that, instead of a particular directory being opened for access, such as C:\MyMusic, the entire hard drive could be opened for browsing and downloading.

> **Detection of Threats**

Since peer-to-peer malicious threats still need to reside on the system's current desktop, a scanning infrastructure can provide protection against infection. However, desktop protection may not prove to be the best method in the future.

Should peer-to-peer networking become standard in home and corporate computing infrastructures, network scanning may become more desirable. Such scanning is not trivial since, by design, peer-to-peer transfer of data does not pass through a centralized server, such as an email server.

Systems such as network-based IDS may prove useful, as well as gateway/proxy scanning to prevent malicious threats from using peer-to-peer connections that pass inside and outside of organizations.

However, peer-to-peer networking models such as Freenet will render networking scanning useless since all data is encrypted. You will not be able to scan data that resides in the DataStore on a system. Detection of threats passed via Freenet type models will only be scanned on the unencrypted file at the desktop just prior to execution. The issue of encryption reinforces the necessity for desktop-based, antivirus scanning.

> **Privacy Concerns**

While the previous threats require a virus writer to create a malicious program, the simple usage of peer-to-peer connections can prove to be the greatest threat to the corporation.

Using peer-to-peer software within your environment creates an unforeseen hole in your network security. Such software easily operates within the restrictions of a configured firewall, as the software generally makes outward connections rather than relying on accepting incoming connections.

Users could easily misuse or configure such software to allow outside systems to browse and obtain files from their computers. These files can be anything from confidential data in an email inbox to proprietary design documents.

Even if the peer-to-peer network is configured properly, the network should not be used to transfer confidential information. Data is generally passed along the network unencrypted. Such data can easily be obtained by a network-sniffing program. Administrators should consider limiting the usage of peer-to-peer networks due to privacy concerns alone.

> **Future**

The current peer-to-peer model appears to be moving toward a true peer-to-peer model without a centralized server, which Microsoft Networking uses today. The current peer-to-peer model's advantage over Microsoft Networking is its ability to perform fast searches and exchange data through firewalls.

Future models of peer-to-peer networking will combine aspects of Microsoft Networking and Napster's protocols to allow for easy search capabilities and the ability of open DataStores.

For example, in Microsoft Networking you can allow for Full Control, meaning that a remote user can not only download, but also upload and modify data stored in the shared space.

Imagine departmental groups in a corporation who need to share and update each other's files. A peer-to-peer networking model that does not require that a file be downloaded in order to be executed, and allows write-ability to remote shares will increase the ability of a malicious threat to propagate.

Threats that infect network shares, such as W32.FunLove, demonstrate the difficulty of containment in environments that utilize central file servers (along with personal shares). A peer-to-peer networking model that incorporates uploading as well as downloading increases the propagation and difficulty of containment of network infectors.

Such a model allows simpler two-way communication of malicious threats. Virus writers may be able to update their threats via a peer-to-peer network. For example, an infected machine may send an update to all other nearby nodes of a peer-to-peer network.

> **Summary**

Peer-to-peer networks obviously pose a danger as an additional vector of delivery. Their impact on security will depend on the adoption of peer-to-peer networks in standard computing environments. If systems use peer-to-peer networks as email is used today, then they will be significant methods of delivery of malicious code. The use of two-way network communication also exposes the system to potential remote control.

More importantly, the usage of a peer-to-peer network creates a hole in a firewall and can lead to the exporting of private and confidential information. Therefore, administrators should begin analyzing their networks for peer-to-peer network usage and configure firewalls and systems accordingly to limit or prevent their usage.

> **About the Author**

Eric Chien joined Symantec Corporation at the Symantec Security Response headquarters in Santa Monica, California in 1997. Chien graduated from the University of California, Los Angeles with a Bachelor's of Science degree in Electrical Engineering and Molecular Genetics.

Currently, Chien heads research in the Europe, Middle East, and Africa (EMEA) regions, analyzing current virus threats and researching new threats in the world of viruses and malicious software. He has been a key developer in projects such as the Digital Immune System (DIS), Symantec's automated system of virus analysis, and the Seeker project, which proactively finds viruses on the Internet.

Chien has spoken at various conferences and published a variety of papers addressing threats to computer security via malicious software.

SYMANTEC, THE WORLD LEADER IN INTERNET SECURITY TECHNOLOGY, PROVIDES A BROAD RANGE OF CONTENT AND NETWORK SECURITY SOFTWARE AND APPLIANCE SOLUTIONS TO INDIVIDUALS, ENTERPRISES AND SERVICE PROVIDERS. THE COMPANY IS A LEADING PROVIDER OF VIRUS PROTECTION, FIREWALL AND VIRTUAL PRIVATE NETWORK, VULNERABILITY ASSESSMENT, INTRUSION PREVENTION, INTERNET CONTENT AND EMAIL FILTERING, AND REMOTE MANAGEMENT TECHNOLOGIES AND SECURITY SERVICES TO ENTERPRISES AND SERVICE PROVIDERS AROUND THE WORLD. SYMANTEC'S NORTON BRAND OF CONSUMER SECURITY PRODUCTS IS A LEADER IN WORLDWIDE RETAIL SALES AND INDUSTRY AWARDS. HEADQUARTERED IN CUPERTINO, CALIF., SYMANTEC HAS WORLDWIDE OPERATIONS IN 38 COUNTRIES.

FOR MORE INFORMATION, PLEASE VISIT WWW.SYMANTEC.COM

WORLD HEADQUARTERS

**20330 Stevens Creek Blvd.
Cupertino, CA 95014 U.S.A.
408.517.8000
800.721.3934**

www.symantec.com

**For Product Information
In the U.S., call toll-free
800.745.6054**

**Symantec has worldwide
operations in 38 countries.
For specific country
offices and contact numbers
please visit our Web site.**