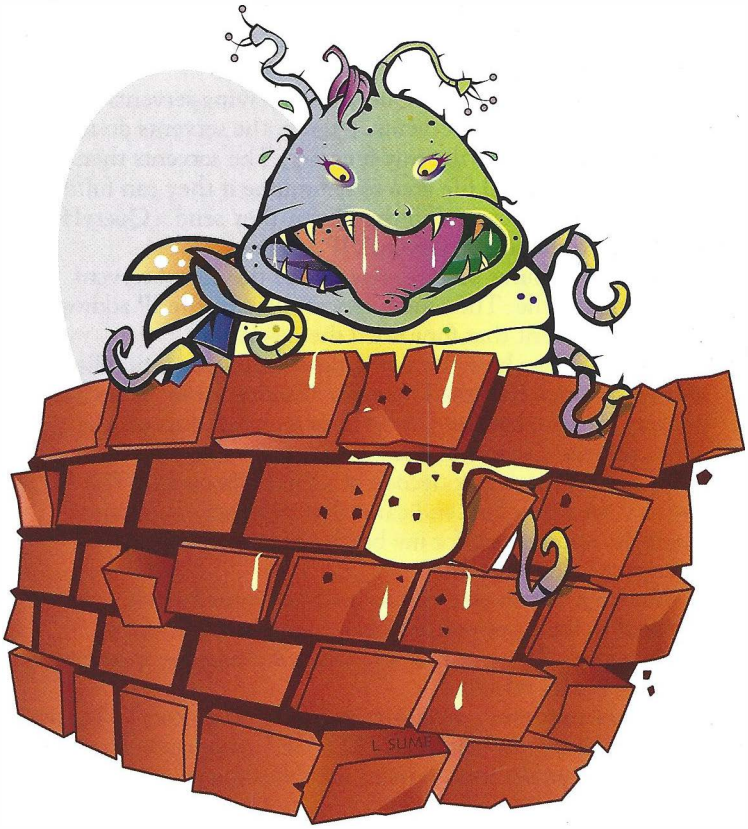
Laura Chappell

Security Alert

Just Say Gno!



On March 14, 2000, Nullsoft, the developer of Winamp and now owned by America Online (AOL), released an "open-source Napster clone" called *Gnutella* (pronounced new-tell-a). Gnutella is a peer-to-peer file-sharing protocol that has gained a worldwide audience similar to the Napster audience. Simple to load and use, Gnutella offers a quick way to share files (such as MP3, software, image, audio, and video files) across the Internet.

In a recent onsite security and network analysis, I was confronted with this Gnutellian beast. While enabling peer-to-peer file sharing, Gnutella creates significant threats to any corporate network, including the following:

- Bandwidth overload
- Security holes large enough to fly an F-18 through

This article examines Gnutella traffic on the wire and outlines the associated bandwidth and security issues. This article also explains how to use port filtering and advanced analyzer filtering to identify and stop Gnutella traffic.

SPOTTING A GNUTELLIAN IN THE WILD

During a recent network security audit, the client's IS team and I noticed a strange matrix traffic pattern. This pattern indicated that a single IP host was receiving packets from more than 82 IP devices through the PIX firewall. (See Figure 1 on p. 34.) We knew it was unlikely that the user was connected to 82 devices through manual interaction. We investigated further to identify the software that created the TCP connections.

We immediately began using a simple IP address filter to capture all of the traffic to and from the local host. After examining the packets, we noticed that all of the traffic was being sent to and from port 6346. Neither Network Associates' Sniffer nor WildPackets' EtherPeek could identify the type of traffic above the TCP layer. (See Figure 2 on p. 34.) We then checked the port number list, which indicated that port 6346 was a Gnutella communication. (The Internet Assigned Numbers Authority [IANA] maintains a list of assigned port numbers. You can download the port number list from www.iana.org.)

Gnutella hosts are called *servents*. To exchange files, one Gnutella servent first connects with another Gnutella servent.

Illustration: Lori Sume

The method these servents use to learn each other's address is not defined in the Gnutella specification. In one of our traces, however, we noticed that the Gnutella web site (www.bearshare.com) sent a list of the active Gnutella servents' IP addresses.

Next, the servents send protocol descriptors to each other. These protocol descriptors describe the purpose of the packet. Five protocol descriptors are defined for Gnutella 0.4:

0x00	Ping	Used to probe for other servents
0x01	Pong	Response to ping (includes number of files/KB to share)
0x40	Push	Request to send using different port number
0x80	Query	A request for a file
0x81	QueryHit	Response indicating sender has file meeting search criteria

The servent then attempts to establish a TCP connection with each of the other Gnutella servents. This could be hundreds or even thousands of other devices.

After the connections are established, the other Gnutella servents immediately begin sending their list of searches through the Gnutella network. This traffic rate runs between 4,500 and 5,300 bytes per second—unnecessary overhead.

A Gnutella servent can share any local files the user makes available. In addition, if a user lists a drive that is mapped to a server, the Gnutella servent can share files from your company's server. This is a huge security risk!

When the servent receives a ping request, that servent routes the ping to all other connected servents. The servent does not send the ping back the way it came. When the desired servent receives the ping, however, that servent sends a pong back the way the ping came. In this way, all servents act as routers for Gnutella protocol descriptors.

When a servent sends a Query, all receiving servents route the packet to all other servents. (Again, the servents do not route the packet back the way it came.) The servents then examine their shared file area to determine if they can fulfill the Query. If they can fulfill the Query, they send a QueryHit descriptor to the source servent's IP address.

Note. The source servent may not be the actual servent requesting the file. The source servent may be the IP address of the last servent that forwarded the descriptor.

Figure 3 shows how the traffic flows through a Gnutella network. Notice that the Query for "beatles" never flows back the way it came, but the QueryHit is returned through the same path the Query was sent.

Figure 4 shows a hex decode of a Gnutella packet. The payload descriptor field indicates that this packet contains QueryHit information for my beatles request. This payload descriptor field is circled.

Warning. When you start to analyze Gnutella traffic, be prepared to see some pretty raunchy searches. Currently, approximately 25 percent of the Gnutella searches appear to be pornography related.

Files are not transferred using the Gnutella protocol. Files are transferred using the standard HTTP GET command. The GET command Gnutella uses contains the following parameters:

```
GET /get/<file index>/<file name>/HTTP/1.0
User-Agent: <Gnutella network>
```

For example, in my download of the Beatles song "Blue Jay Way," the HTTP packet contained the following GET parameters:

```
GET /get/<69>/Beatles - Blue Jay Way.mp3/HTTP/1.0
User-Agent: LimeWire 1.6b
```

When a file transfer begins, the traffic level immediately spikes. (See Figure 5.)

SLIDING THROUGH FIREWALLS

The Gnutella protocol specification can bypass firewalls that are set up to block the file downloads. For example, if the servent that has the desired file does not permit an incoming file download connection on the Gnutella default port, that servent can send a Push request that lists an allowed IP address and a port. Users inside a company can also get around a firewall by setting up the Gnutella application to use ports that are allowed through the firewall, such as port 80 (HTTP), 443 (HTTPS), 23 (Telnet), 25 (SMTP), or 110 (POP3).

ADDING INSULT TO INJURY

Just when you thought it was bad enough, here's a little salt for that network wound. In February 2001, a worm named *Mandragore* posing as a regular media file infected the Gnutella network. (Mandragore is also known as *W32/Gnuman*.) After this executable file is downloaded and run, however, it infects the

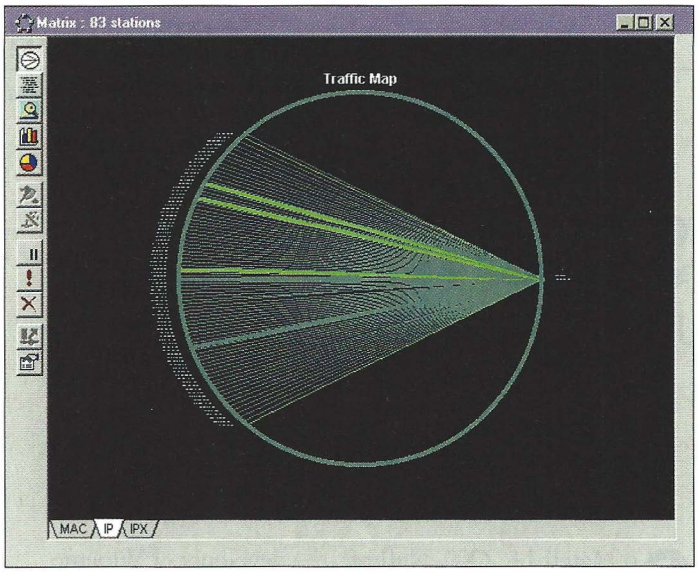


Figure 1. *The typical IP communications pattern of a Gnutella connection*

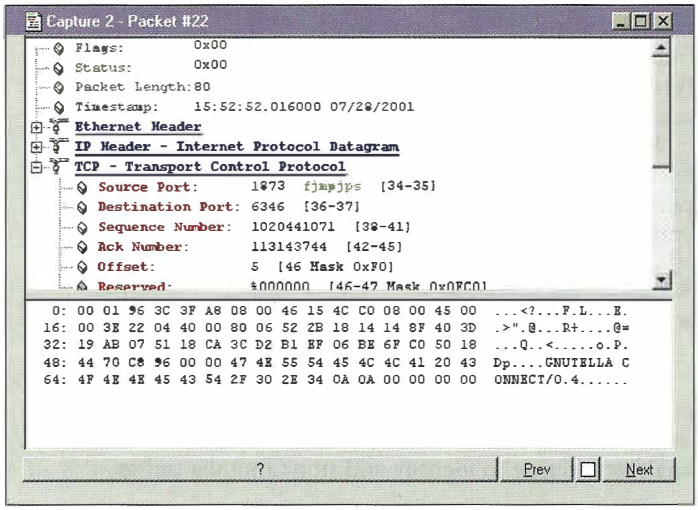


Figure 2. *Analyzers may not be able to decode Gnutella traffic.*

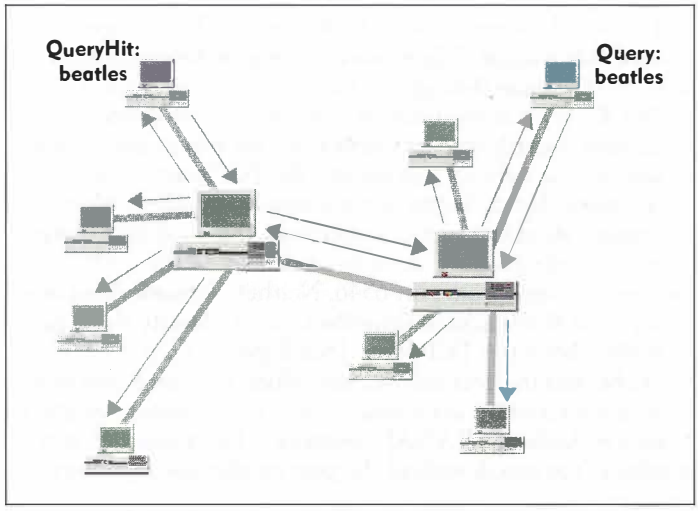


Figure 3. *Gnutella responses follow the original request path through the network.*

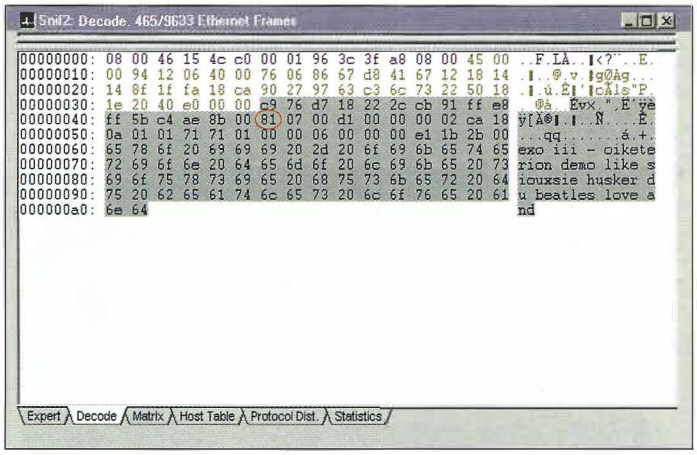


Figure 4. *Gnutella traffic contains the Query in plain text.*

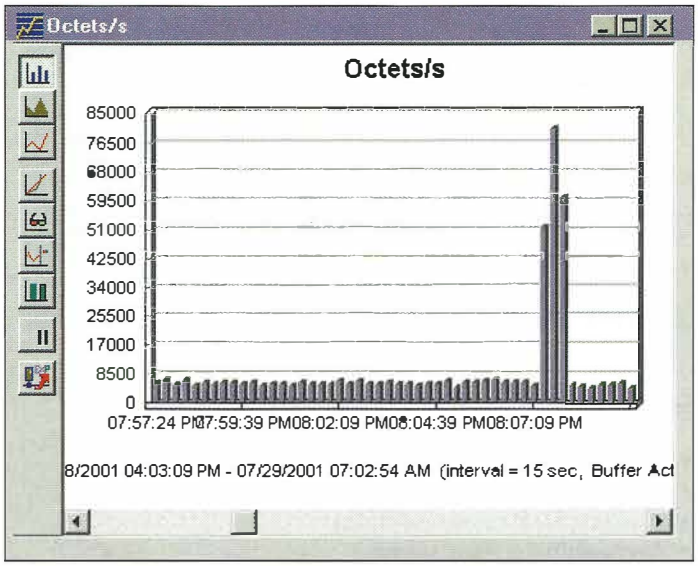


Figure 5. *The file transfer process creates a sudden spike in the traffic level.*

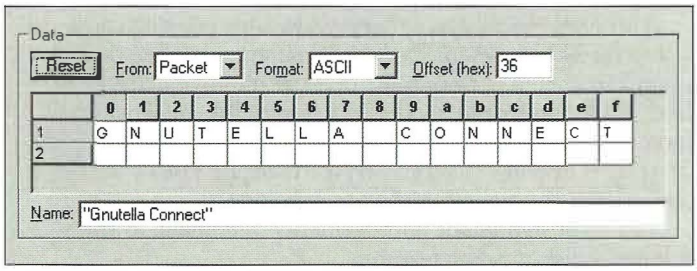


Figure 6. *The Sniffer filter for Gnutella-Connect patterns*

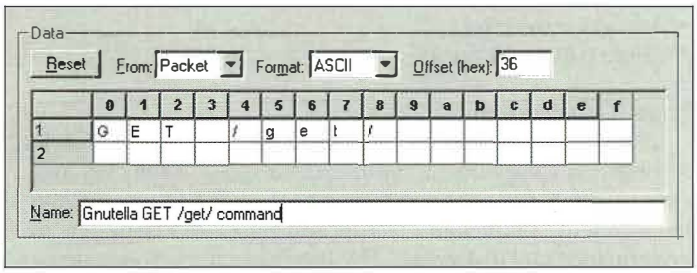


Figure 7. *The Sniffer filter for the "GET /get/" pattern*

Gnutella victim and spreads to other computers, constantly changing its name to match all media search requests. (For more information about the Mandragore virus, visit www.mcafee.com.)

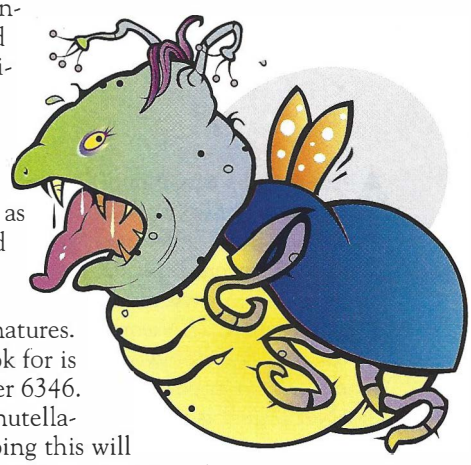
DEALING WITH GNUTELLA

After reading this article, you may feel a little threatened. However, I believe we are just seeing the tip of the iceberg. Gnutella will offer a special set of challenges for large and small networks. Given the serious security threat Gnutella offers, I believe you should completely eradicate Gnutella from your corporate network.

I can think of three steps you should take immediately to deal with Gnutella traffic on the network:

Step 1: Set up port filtering. For example, set up BorderManager to filter on incoming and outgoing ports 6346 (Gnutella) and 6347 (Gnutella router).

Step 2: Define corporate policy regarding shared network access. Ensure all users understand the security and performance issues associated with peer-to-peer networks such as Gnutella. Educate users on the various forms of Gnutella networks, such as LimeWire, Gnotella, and BearShare.



Step 3: Analyze all traffic looking for Gnutella signatures. The first signature to look for is the Gnutella port number 6346. Second, look for the "Gnutella-Connect" text string. Doing this will catch any Gnutella traffic using a nondefault port number. Also, build and use a filter that looks for the "GET /get/" string that is used to start a Gnutella file transfer. Figures 6 and 7 show the filters I created to capture these three signatures.

Don't wait until your network suffers from Gnutella overhead, security leaks, or worms. Start shutting down Gnutella ports and begin educating users.

For more information about Gnutella, visit the following web sites:

- http://dss.clip2.com
- www.limewire.com
- www.bearshare.com
- www.gnutellanews.com
- www.gnutelliums.com
- www.gnutella.co.uk
- www.gnutella.wego.com
- www.zeropaid.com

Laura Chappell performs onsite network analysis sessions for troubleshooting, optimization, and security checks and teaches hands-on courses on protocol analysis. For more information about these services or courses, send an e-mail message to lchappell@packet-level.com.