Laura Chappell

Advanced Packet Filtering

Editor's Note: This article assumes that you are familiar with the basics of packet filtering. You can review the basics by reading "Basic Packet Filtering Using Your Network Analyzer" (Novell Connection, Jan. 2001, pp. 40–41). You can download this article from www.ncmag.com/past. You can also find information about basic packet filtering at www.packet-level.com.

Advanced packet filtering requires a thorough knowledge of packet structures and protocols—it's not for the faint of heart. After you master the art of creating packet filters, however, you will be better equipped to protect networks from cyber attacks, troubleshoot complex networking problems, and optimize networks to increase performance and enhance users' workflow.

This article outlines the steps required to build an advanced filter using Network Associates' Sniffer Pro 4.5. Most quality network analyzers offer some way to build advanced filters with the following characteristics:

- Bit or byte value patterns, which enable you to capture traffic that contains a specific value at a specific location within a packet
- Boolean operations that enable you to combine these patterns with AND/OR operands

BIT OR BYTE VALUE PATTERNS

The practice of specifying values at exact bit and/or byte locations in a packet can turn your network analyzer into an extremely powerful tool. For example, you may want to match the following patterns:

- **TCP Handshakes.** TCP headers with the SYN bit set to 1 (indicating that someone is attempting to make a TCP connection on the network).
- **Hidden FTP File Transfers.** TCP packets that contain the value RETR[space] above the TCP header (indicating that a device is transferring a file using FTP).
- **Destination Unreachable Packets.** Internet Control Message Protocol (ICMP) packets with the type 0 (indicating that a device is sending destination unreachable packets onto the network).

Illustration: Norman Feldble

TCP Handshakes

Connection-oriented services, such as FTP and HTTP, require an initial TCP handshake to establish a connection and exchange a starting sequence number. This sequence number increases according to the amount of data received, thereby offering reliable, guaranteed service for TCP data.

In the first packet of the TCP handshake, the SYN (SYNchronize sequence number) bit in the TCP header is set to 1. What information does this packet give you? Suppose that you have configured an extremely secure network for your company. For example, you may have decided that no one on the Internet should be able to connect to the computers on your company's network.

Because you have taken this security measure, no SYN packets would make it through the firewall, right? To determine if any SYN packets make it through the firewall, you can build a filter to capture all SYN traffic that crosses the wire on the inside of the firewall.

The bit sequence in TCP headers is structured as follows:

r	r	U	A	P	R	S	F	
0	0	0	0	0	0	1	0	=0x02

In the TCP header, the bit sequence is coded as follows:

r = reserved bits—set to 0
U = Urgent bit (value 0x20)
A = Acknowledgment bit (value 0x10)
P = Push bit (value 0x08)
R = Reset bit (value 0x04)
S = Synchronize bit (value 0x02)
F = Finish bit (value 0x01)

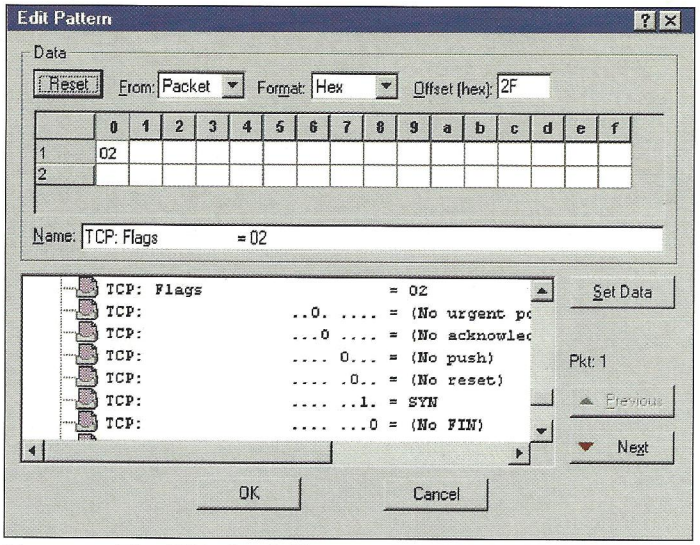


Figure 1. *The TCP flag fields are at offset 2F (hex) from the protocol layer (after the MAC header).*

Figure 1 shows the filter I built to capture all packets that have the SYN bit set.

Note. If someone is really sneaky, he or she may try to bypass your company's firewall by sending a packet that has both the SYN flag and another flag set. To detect this type of packet, you may have to build a more advanced filter—a Boolean filter that uses the OR operand to filter on various flag-setting patterns. (Boolean filters will be discussed later in this article.)

Hidden FTP File Transfers

Although Request for Comments (RFC) 959 defines port 21 for FTP commands, many FTP server packages allow you to select your own port number to run FTP services. (For more information about RFC 959, visit www.ietf.org/rfc/rfc959.) This ability to choose another port number is certainly a security hazard.

For example, consider what would happen if a disgruntled employee set up his or her office desktop with FTP services on port 80. The employee could go home, connect to the company's network, and send packets through the firewall. (Naturally, the firewall would support port 80, which is typically used for HTTP operations.)

To prevent anyone from bypassing your company's firewall in this way, you can create a packet filter. For example, Figure 2 shows a simple filter set up to capture all RETR[space] commands (indicating an FTP file transfer), regardless of the port number that is being used. Packet filters such as this one do not change the traffic—they simply check to see if a particular type of traffic is being transmitted.

Note. Ephemeral ports are ports that are temporary. As you might guess, ephemeral ports are used only for a short period of time. For example, when an FTP client issues an ls command to view the contents of a directory, this FTP client sets up a temporary port number to be used for the transfer of directory contents. The FTP client software then translates the ls command to the NLST command.

Ephemeral ports irritate most protocol analysts. After all, just when you think you have set up a nice FTP filter by using the checkbox method, you find out that the filter won't capture even half of the actual FTP communications. (For more information about using the checkbox method to create packet filters, see "Basic Packet Filtering Using Your Network Analyzer," *Novell Connection*, Jan. 2001, pp. 40–41. You can download this article from www.ncmag.com/past.)

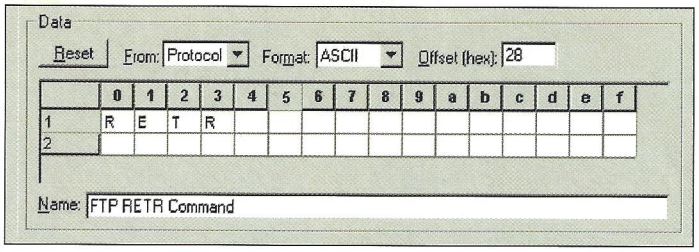


Figure 2. *By switching to ASCII format, you can enter the commands as they are listed above.*

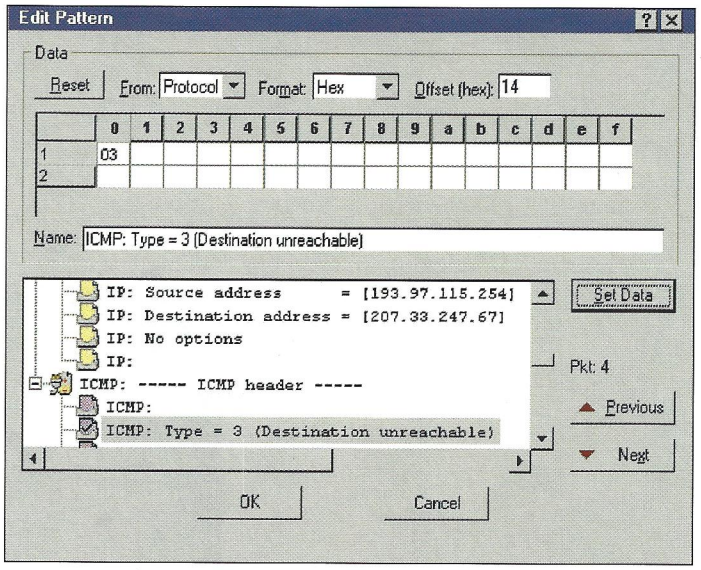


Figure 3. *This filter, which is easy to set up, captures only traffic with the value 0x03 in the Type field.*

Destination Unreachables

Hackers can use ICMP packets to discover information about active devices on your company's network. For example, when a device is being questioned about a particular service (such as FTP or telnet), the device can respond to the request with a Destination Unreachable ICMP packet. (For more information about how hackers use ICMP, see "You're Being Watched: Cyber-Crime Scans," *Novell Connection*, Mar. 2001, pp. 20–31. You can download this article from www.ncmag.com/past.)

Any decent network analyzer should have a predefined ICMP filter. However, a predefined filter does not look for specific types of ICMP traffic. The value 0x03 in the ICMP header's Type field identifies the packet as a Destination Unreachable packet. For example, in Figure 3, I have set up a filter that captures only traffic with the value 0x03 in the Type field.

Note. I strongly recommend that you learn ICMP from the inside out—today! Read RFC 792 (www.ietf.org/rfc/rfc792), get the "Packet-Level ICMP" course from www.podbooks.com, and capture the ICMP traffic on your company's network. You will be amazed at how you can optimize a network by capturing and analyzing ICMP traffic.

BOOLEAN OPERATIONS

To effectively use some bit or byte value patterns, you have to logically combine them in a single filter using a Boolean operand such as AND. For example, suppose you are trying to capture all of the fragments on a network. If you filter on all packets that have the "more fragments" bit set to 1, you would miss the last fragment of the fragment set.

Likewise, if you filter on the RETR FTP command to detect anyone who is running a nonstandard implementation of FTP, you would miss anyone who is using the STOR command to send files onto your company's network. These examples show why you need to use Boolean operands to combine and differentiate patterns.

The standard Boolean operands include the following:

- AND
- OR
- AND NOT

To take full advantage of advanced Boolean filtering, you must have a thorough understanding of how protocols work. After you put in the time and effort to understand protocols, however, you will find that your network analyzer has become ten times as useful as it was before.

The following sections provide an example of each Boolean operand and explains how you define these Boolean filters with the Sniffer Pro 4.5.

AND (Catching Port Unreachables)

This filter allows you to capture a specific type of ICMP Destination Unreachable packet. With the following filter, you are looking for Port Unreachable packets that indicate someone is requesting a service that does not exist at that location:

Packets with the ICMP type field value of 3 (Destination Unreachable)
AND
Packets with the ICMP code value 3 (Port Unreachable)

By using AND to filter on these operations, you can look for packets that have both the type 3 value and the code 3 value at the correct offset. Figure 4 shows how you specify patterns for this filter in Sniffer Pro 4.5; Figure 5 shows the final data pattern filter.

The code numbers for Destination Unreachable packets are listed below:

0 Net Unreachable
1 Host Unreachable
2 Protocol Unreachable
3 Port Unreachable
4 Fragmentation Needed and Don't Fragment was Set
5 Source Route Failed
6 Destination Network Unknown
7 Destination Host Unknown
8 Source Host Isolated
9 Communication with Destination Net is Administratively Prohibited
10 Communication with Destination Host is Administratively Prohibited
11 Destination Network Unreachable for Type of Service

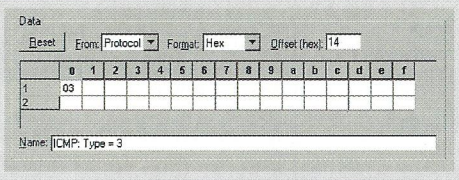
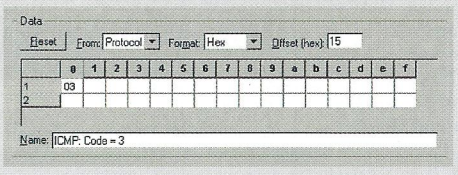
Patterns	Sniffer Pattern Window
Pattern 1: Packets with the ICMP type field value of 3 (Destination Unreachable)	
Pattern 2: Packets with the ICMP code value 3 (Port Unreachable)	

Figure 4. *This pattern allows you to capture two types of packets—packets with the ICMP type field value of 3 and packets with the ICMP code value 3.*

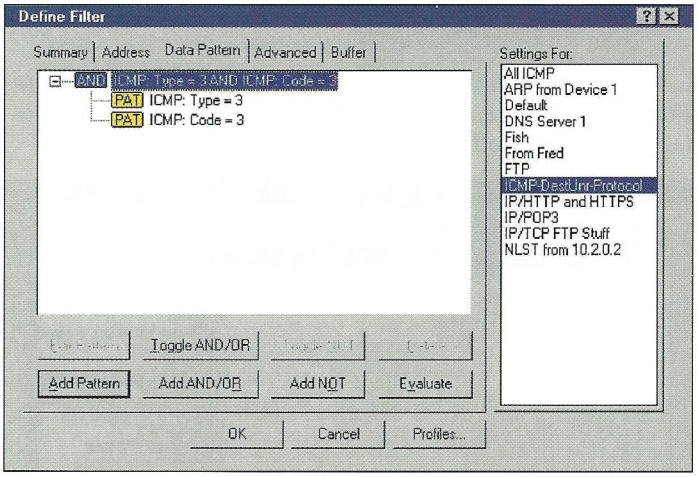


Figure 5. *The AND operand enables you to detect packets that are using two values.*

12 Destination Host Unreachable for Type of Service
13 Communication Administratively Prohibited (See RFC 1812 at www.ietf.org/rfc/rfc1812.)
14 Host Precedence Violation (See RFC 1812.)
15 Precedence cutoff in effect (See RFC 1812.)

Note. The Internet Assigned Numbers Authority (IANA) maintains this list of code numbers. As a protocol analyst, you should stay up-to-date on the latest numbers assigned for various protocols. To get the latest information, visit www.iana.org.

OR (Catching Non-Standard FTP Operations)

As mentioned earlier in this article, you can create a filter that captures all of the packets that contain the value RETR[space] (which is used when someone retrieves a file via FTP). However, what if you want to capture specific FTP traffic? For example, you may want to capture FTP traffic that is used to put files onto local systems, to retrieve files from local systems, or to view file lists.

Understanding how FTP works will help you create a filter that captures specific FTP traffic. FTP uses a series of commands directly after the TCP header. These commands are following:

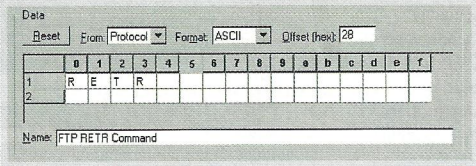
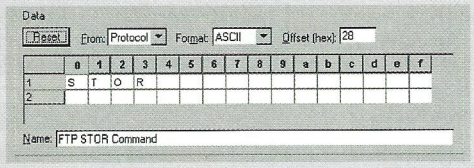
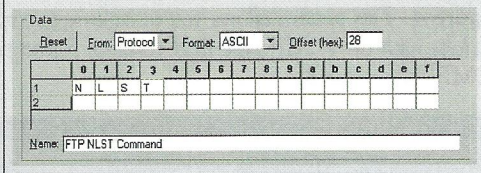
Patterns	Sniffer Pattern Window
Pattern 1: Packets with the RETR pattern (used to get files using FTP)	
Pattern 2: Packets with the STOR pattern (used to put files using FTP)	
Pattern 3: Packets with the NLST pattern (used to list files using FTP)	

Figure 6. *The pattern shown here allows you to capture the following types of FTP traffic: RETR, STOR, and NLST packets.*

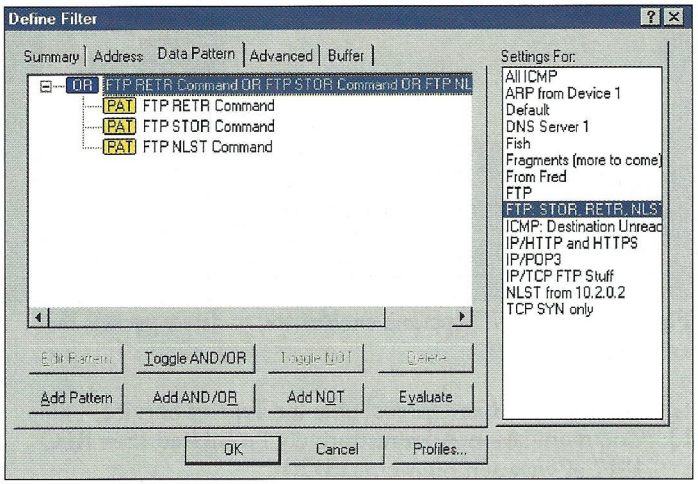


Figure 7. *When you are building a filter, you can use the OR operand to widen the number of possible data matches.*

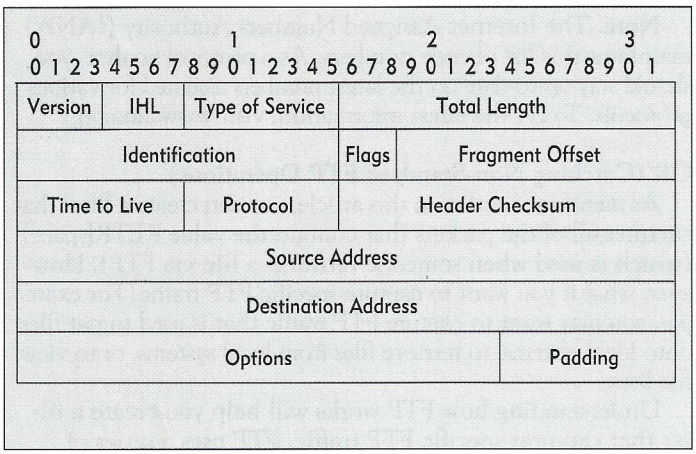


Figure 8. *If you are managing an IP network, you should understand the format of the IP header.*

USER	Log in with this username
PASS	Use this password for login
NLST	List files on remote system
CWD	Change working directory (on remote system)
PORT	Use the following ephemeral port number
RETR	Retrieve a file
STOR	Put a file on the remote system
QUIT	Log out

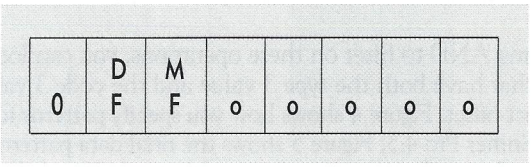
For example, you can use the following filter to capture RETR[space], STOR, or NLST packets:

Packets with STOR following the TCP header (for FTP put commands)
OR
Packets with RETR[space] following the TCP header (for FTP get commands)
OR
Packets with NLST following the TCP header (for FTP file listings) (consider adding AND NOT port 21)

By using OR to filter on these three operations, you can look for packets that have either the RETR[space], STOR, or NLST commands. Can we delete value? Figure 6 shows how you specify patterns for this filter in Sniffer Pro 4.5, and Figure 7 shows the final data pattern filter.

AND NOT (Catching All Fragmented Packets)

IP can fragment packets if those packets must cross a network that supports a smaller Maximum Transmission Unit (MTU) size. Fragmented packets cause additional overhead and inefficient use of the network bandwidth. In some cases, hackers can use fragmented packets to cross a firewall with data that should be filtered from the network. Figure 8 shows the format of an IP header. The breakdown of the Flags and Fragment Offset fields is shown below:



In this Flags and Fragment Offset field, the following apply:

- "0" is always set to zero.
- DF identifies the "don't fragment" bit, which means "don't fragment" when set to 1 and "OK to fragment" when set to 0.
- MF identifies the "more fragments" bit, which means "more fragments to come" when set to 1 and "last fragment" when set to 0.
- The "o" identifies the offset field, which defines the location of this packet in the entire datastream.

A simple filter based on "1 in the more to come" bit captures all fragments except the last fragment in the set. The filter does not capture the last fragment because this fragment contains the value 0 in the "more to come" bit field. As you can see, creating a

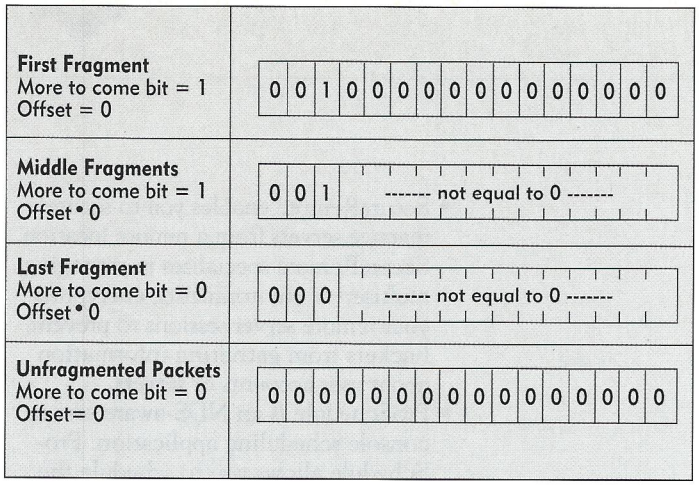


Figure 9. *By studying the various packets transmitted across the network, you can define a filter to capture all fragment packets.*

filter to capture this fragment requires some extra thinking.

Figure 9 shows characteristics of the various packets seen on a network. Can you find the defining patterns that would enable you to capture all fragments?

All of the packets in a fragment set have either "1" in the "more fragments" bit, or the packets have some value other than "0" in the offset field. Using the AND NOT operand, you can build the following filter to capture all fragment packets:

Packets with the "more fragments" bit set to 1 (part of a fragment set)
AND NOT
The IP "more to come" bit set to 0
AND
The IP fragment offset of 0

You can use Sniffer Pro 4.5's binary format to make your filter a bit more clear instead of jumping to hex translations. Figure 10 shows how you specify patterns for this filter in Sniffer Pro 4.5, and Figure 11 contains the summary of the final filter that uses the AND NOT operand.

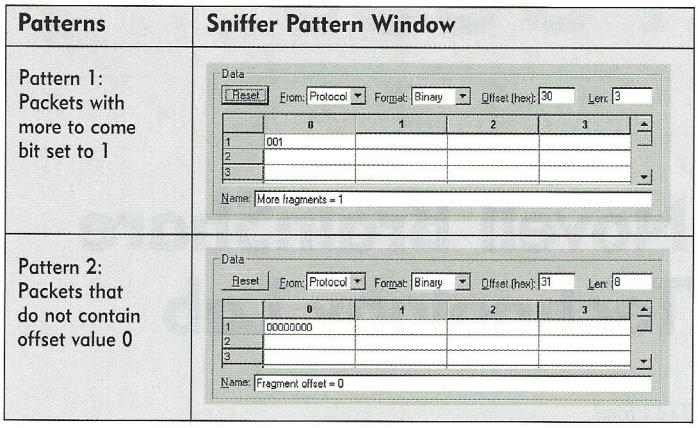


Figure 10. *This pattern allows you to capture all fragment packets that are transmitted on the network.*

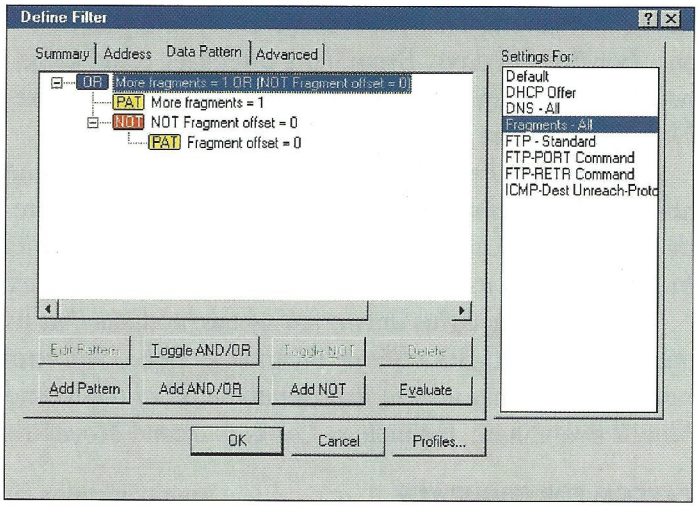


Figure 11. *By using the AND and the AND NOT operands, you can build a filter to capture just fragment packets.*

CONCLUSION

If you have heard me lecture on packet filtering, you know that I consider filtering a true art form. Creating packet filters also requires hard work: You need to understand protocols; you need to know where to get the offset and field value information; you need to know how to figure out what the possible variations are; and you need to test your filters.

ADDITIONAL RESOURCES

If this article has whetted your appetite for analyzing your company's network at packet level, check out the following *Novell Connection* articles by Laura Chappell:

- "10 Tips for Creating a Network Analysis Report," Feb. 2000, pp. 22–27
- "Fragmentation: Good, Bad, and Downright Ugly," Mar. 2000, p. 32
- "Inside the TCP Handshake," Mar. 2000, pp. 34–36
- "Cyber Crime: It Could Happen to You," Apr. 2000, pp. 36–40
- "Analyzing FTP Communications," Sep. 2000, pp. 22–34
- "It's Alarming: Broadcast and Multicast Storms," Oct. 2000, pp. 37–38
- "Using the Sniffer Capture Window or Panel," Nov. 2000, pp. 38–39
- "Basic Packet Filtering Using Your Network Analyzer," Jan. 2001, pp. 40–41
- "You're Being Watched: Cyber-Crime Scans," Mar. 2001, pp. 20–31
- "Routing Sequences for ICMP," Mar. 2001, pp. 32–35

You can download all of these articles at www.ncmag.com/past.

In addition, you can see Laura Chappell in virtual person by watching her Webcast seminar. During the seminar, she discusses topics such as wireless technologies, troubleshooting your company's network, and "living at packet-level." You can view the Webcast on the NUI web site at www.nuinet.com.

Laura Chappell has just released a four-book CD set that includes Laura's Lab Kit—a collection of Laura's favorite analysis tools and utilities, articles, video clips, and Scott Haugdahl's Network Analyst's Survival Kit. This four-book CD set is available online at www.podbooks.com.