Laura Chappell

Inside the TCP Handshake

Editor's Note: For more information about network analysis techniques and uses, attend Laura Chappell's "Introduction to Network Analysis" (session 164) at BrainShare 2000 in Salt Lake City. You can also visit http://www.netanalysis.org.

Many applications rely on the connection-oriented services (such as HTTP, FTP, and pure IP) which are offered by TCP. When these applications are launched, the TCP stack on the local device must establish a connection with the TCP stack on the destination device.

The handshake process is based on three steps. (In this article, the device that initiates the handshake process is called *Device 1*, and the destination device, or the target of the connection, is called *Device 2*.)

1. Device 1 sends its TCP sequence number and maximum segment size to Device 2.
2. Device 2 responds by sending its sequence number and maximum segment size to Device 1.
3. Device 1 acknowledges receipt of the sequence number and segment size information.

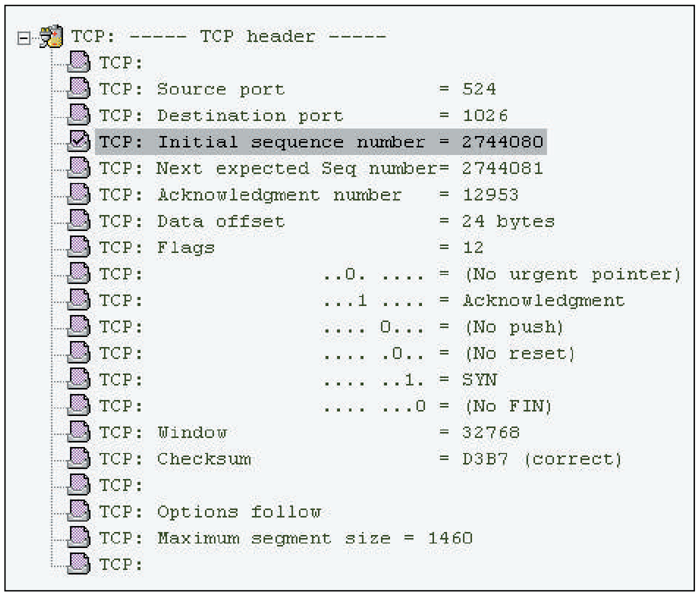
That's it. Pretty simple. By looking inside the handshake packets, you can see what a TCP header actually contains during the handshake process and understand how to troubleshoot this process.

Why is it so important to recognize a healthy TCP handshake? Unfortunately, hackers can alter this handshake to overload a destination device. Remember last month's denial of service attacks on popular Internet sites?

INSIDE PACKET 1

In the initial packet, Device 1 inserts a self-assigned initial sequence number in the TCP header. (See Figure 1.) This number is used to track the sequence of data sent to Device 2, ensuring that no packets are missing. The SYN flag, which is set on packets, is used to synchronize sequence numbers.

Note: This packet contains a hidden field—the Acknowledgment Number field. The Acknowledgment Number field contains the next-expected sequence number from the other side of the communication. Since this packet is the first in the handshake process, however, Device 1 does not know what sequence number Device 2 will use. As a result, the 4-byte field is left at 0x0000-0000, and this field is not displayed by the network analyzer.


```
TCP: ----- TCP header -----
TCP:
TCP: Source port              = 524
TCP: Destination port         = 1026
TCP: Initial sequence number  = 2744080
TCP: Next expected Seq number= 2744081
TCP: Acknowledgment number    = 12953
TCP: Data offset              = 24 bytes
TCP: Flags                    = 12
TCP:               ..0. .... = (No urgent pointer)
TCP:               ...1 .... = Acknowledgment
TCP:               .... 0... = (No push)
TCP:               .... .0.. = (No reset)
TCP:               .... ..1. = SYN
TCP:               .... ...0 = (No FIN)
TCP: Window                   = 32768
TCP: Checksum                 = D3B7 (correct)
TCP:
TCP: Options follow
TCP: Maximum segment size = 1460
TCP:
```
Figure 1. The initial TCP handshake packet

The Data Offset field simply defines the length of the TCP header. The header is typically 24 bytes during the first two packets of the handshake and 20 bytes for regular data exchange.

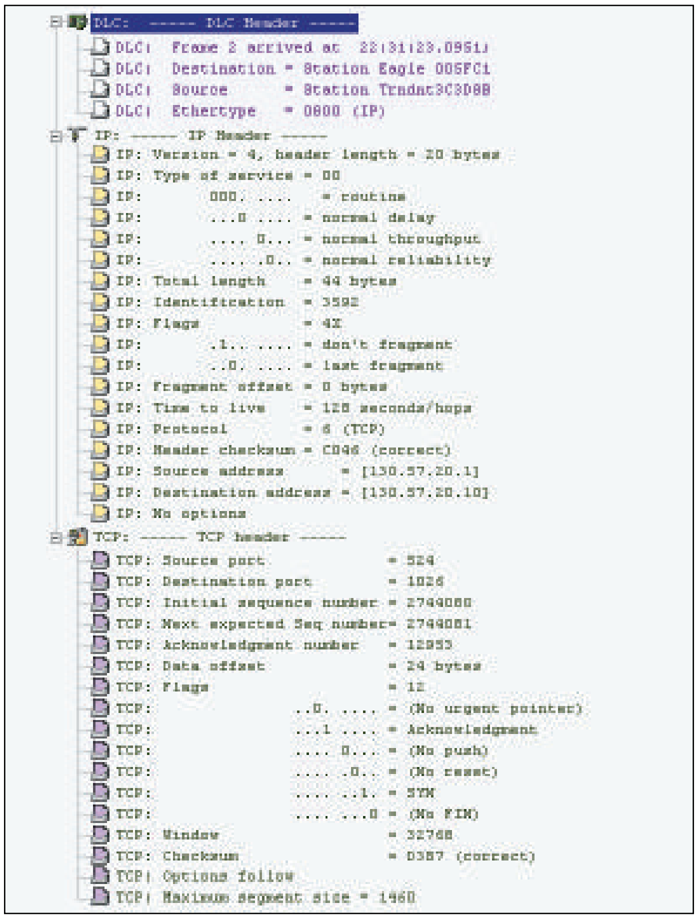
In this packet, you will also notice that Device 1 has defined a Maximum Segment Size (MSS), which indicates the amount of data that can be contained in the packet after the TCP header. Here's why 1,460 bytes makes sense for this Ethernet packet:

1,460	Data after TCP header (the payload)
20	Typical TCP header size
20	Typical IP header size
14	Typical Ethernet header size
4	CRC size
1,518	Maximum Ethernet packet size

The TCP header also defines the desired process or application for this connection: port 524. NetWare 5 uses this port for pure IP (NetWare Core Protocol [NCP] over IP) communications.

INSIDE PACKET 2

Figure 2 shows the TCP header inside Device 2's response to the connection request. (See p. 36.) As you can see, Device 2



```
DLC:  ----- DLC Header -----
   DLC:  Frame 2 arrived at  22:31:23.0951;
   DLC:  Destination = Station Eagle 005FC1
   DLC:  Source      = Station Trndnt3C3D8B
   DLC:  Ethertype   = 0800 (IP)
IP: ----- IP Header -----
   IP: Version = 4, header length = 20 bytes
   IP: Type of service = 00
   IP:       000. ....   = routine
   IP:       ...0 .... = normal delay
   IP:       .... 0... = normal throughput
   IP:       .... .0.. = normal reliability
   IP: Total length    = 44 bytes
   IP: Identification  = 3592
   IP: Flags           = 4X
   IP:       .1.. .... = don't fragment
   IP:       ..0. .... = last fragment
   IP: Fragment offset = 0 bytes
   IP: Time to live    = 128 seconds/hops
   IP: Protocol        = 6 (TCP)
   IP: Header checksum = C046 (correct)
   IP: Source address      = [130.57.20.1]
   IP: Destination address = [130.57.20.10]
   IP: No options
TCP: ----- TCP header -----
   TCP: Source port             = 524
   TCP: Destination port        = 1026
   TCP: Initial sequence number = 2744080
   TCP: Next expected Seq number= 2744081
   TCP: Acknowledgment number   = 12953
   TCP: Data offset             = 24 bytes
   TCP: Flags                   = 12
   TCP:               ..0. .... = (No urgent pointer)
   TCP:               ...1 .... = Acknowledgment
   TCP:               .... 0... = (No push)
   TCP:               .... .0.. = (No reset)
   TCP:               .... ..1. = SYN
   TCP:               .... ...0 = (No FIN)
   TCP: Window                  = 32768
   TCP: Checksum                = D387 (correct)
   TCP: Options follow
   TCP: Maximum segment size = 1460
```

Figure 2. The second TCP handshake packet

has selected the sequence number 2744080.

Now the network analyzer—in this case, Network Associates's Sniffer—shows the value of the Acknowledgment Number field—12953. This field contains the next sequence number that Device 2 expects to receive from Device 1.

This packet has two flags set: SYN (which synchronizes sequence numbers) and ACK (which acknowledges receipt of Device 1's first packet).

This packet also indicates that Device 2 has an MSS of 1,460 bytes as well. Does this mean that the communications between the two devices would never be fragmented? No. If there is an intervening link that does not support this packet size, the packets would be fragmented.

INSIDE PACKET 3

To view packet 3, you can download the TCPSHAKE.CAP file from the trace file area at http://www.netanalysis.org. You can then use your network analyzer to view the contents of packet 3. You can probably guess what Device 1's sequence number is (12953). The final packet in the handshake process has the ACK flag set.

CONCLUSION

A basic knowledge of the typical communication processes will save you time and effort as you troubleshoot your company's network. This knowledge can also help you identify perpetrators of malicious attacks on your company's network.

Laura Chappell writes technical training books for podbooks.com (http://www.podbooks.com) and is a senior protocol analyst at NetAnalysis Institute. Ms. Chappell also makes a pretty mean margarita. (For more information about NetAnalysis Institute, visit http://www.netanalysis.org.)