Home > My courses > Network Defense and Remote Access Configuration > 04 [Enter Module Title Here] > Lesson Proper for Week 4

Lesson Proper for Week 4

TCP/IP II

IP Header Structure

The data in an IP packet is the part that end users see, but the header is the part that computers use to communicate, and it plays an important role in network security and intrusion detection. An IP header contains a number of fields and is similar to a TCP header, as you will learn in the “TCP Headers” section later in this chapter. Figure 2-1 shows a common way of depicting information in an IP header, which is divided into sections of 32-bit layers.

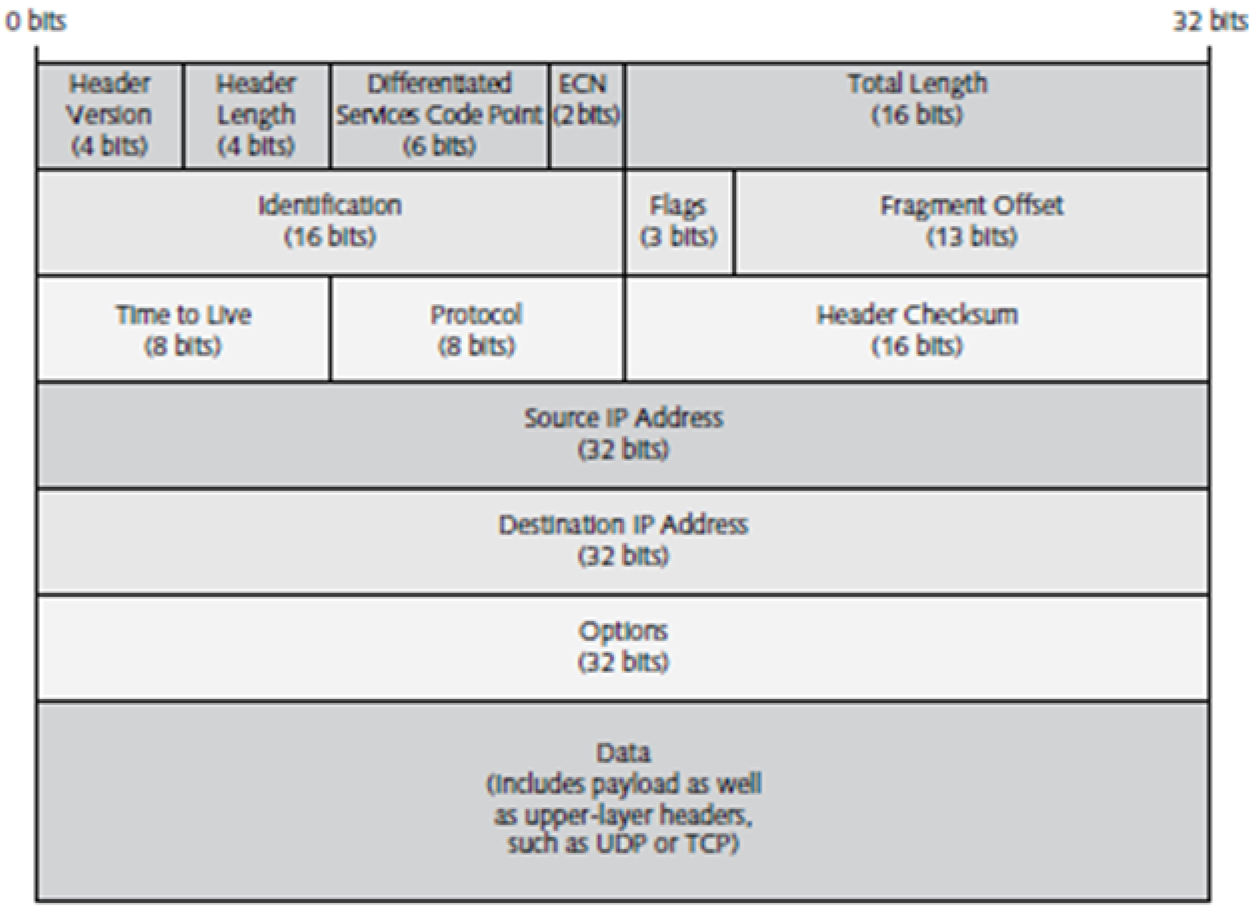

0 bits
32 bits
Header Version (4 bits)
Header Length (4 bits)
Differentiated Services Code Point (6 bits)
ECN (2 bits)
Total Length (16 bits)
Identification (16 bits)
Flags (3 bits)
Fragment Offset (13 bits)
Time to Live (8 bits)
Protocol (8 bits)
Header Checksum (16 bits)
Source IP Address (32 bits)
Destination IP Address (32 bits)
Options (32 bits)
Data (Includes payload as well as upper-layer headers, such as UDP or TCP)

Figure 2-1 IP header structure

It is important to understand the different fields in an IP header so you can configure packet filters properly. Each field has varying importance to attackers, so you must know what each one does to protect against different types of attacks. The following fields make up the IP header structure:

- *Header Version*—This 4-bit field identifies the IP version used to generate the datagram. Because this is an IPv4 header, the value should indicate version 4.

- *Header Length*—This field describes the length of the header in 32-bit words, and is a 4-bit value. The default value is 20 bytes. Thus, a header length that has a binary size of 0101, which equals 5 in decimal, indicates 5 × 32 bits or 160 bits.Next, 160 bits/8 = 20 bytes.
A "word" is the width of a computer processor' s registers. As you should recall from your hardware classes, registers are the storage areas where the processor does calculations. Most computers since the 386 processor have had a 32-bit word size, although most new computers now support a 64-bit word size and processors like the AMD Athlon or the Intel Core i7.

- *Differentiated Services Code Point (DSCP)*—This 6-bit field expresses the quality of service in the datagram's transmission through the network. This field is used by services that are sensitive to latency issues, such as Voice over IP (VoIP).

- *Explicit Congestion Notification (ECN)*—This 2-bit field allows ECN-compliant routers that are operating on ECN-compliant network infrastructures to signal congestion and thus minimize dropping of packets.

- *Total Length*—This 16-bit field specifies the datagram's total length to a maximum of 65,535 bytes.

- *Identification*—This 16-bit value helps divide the data stream into packets of information. The receiving computer (possibly a firewall) uses each packet's identification number to reassemble the packets that make up the data stream in the correct order.

- *Flags*—This 3-bit value indicates whether the datagram is a fragment—one datagram within a sequence of datagrams that make up an entire communication—and whether it is the last fragment or more will follow.

- *Fragment Offset*—If data is received in the form of a fragment, this value indicates where the fragment belongs in the sequence so that a packet can be reassembled.

- *Time to Live (TTL)*—This 8-bit value identifies the maximum amount of time the packet can remain in a network before it is dropped. Each router or device through which the packet passes (hops) reduces the TTL by a value of one. The TTL avoids congestion that results from corrupted packets infinitely looping through the network.

- *Protocol*—This field identifies the type of protocol being carried. For example, 1 = ICMP, 2 = IGMP, 6 = TCP, 17 = UDP, 47 = GRE (Generic Routing Encapsulation), 50 = ESP (Encapsulating Security Payload), and 51 = AH (Authentication Header).

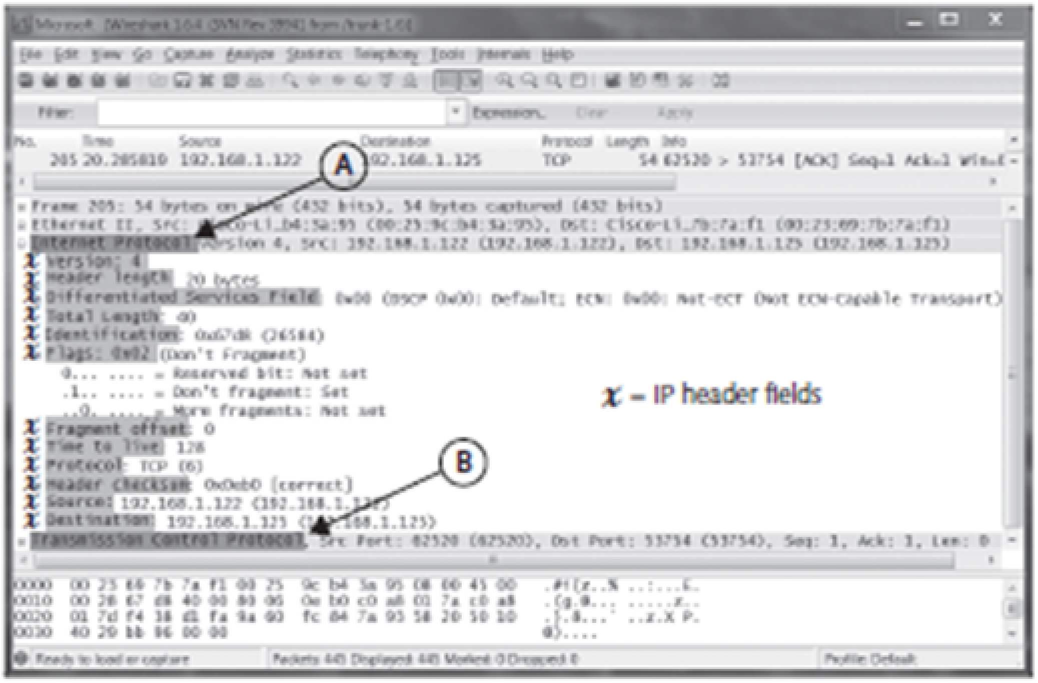
- *Header Checksum*—This field is the sum of the 16-bit values in the datagram header; it is calculated at every hop to ensure accuracy of the header.

- *Source IP Address*—This field is the address of the computer or device that sent the IP datagram.

- *Destination IP Address*—This field is the address of the computer or device that received the IP datagram.

- *Options*—This field can include items such as a security field and several source routing fields that the packet sender uses to supply routing information. Gateways can then use this routing information to send the packet to its destination.

Programs that capture packets as they pass through a network interface give you another way to view packet header information. Most network operating systems (NOSs) have some type of built-in or add-on program to monitor network activity, such as Windows Network Monitor. Many security administrators, however, prefer third-party applications for their versatility and extra features. For example, Wireshark, which was formerly Ethereal, is an open-source network analysis utility that tracks packets and supplies detailed information about them. (For more information, see www.wireshark.org and Hands-On Project 2-2 at the end of this chapter.) Figure 2-2 shows a portion of a Wireshark capture of a TCP/IP packet. The "A" indicates the beginning of the IP header. The fields are indicated by an "X." The "B" shows the start of the TCP header, the upper-level protocol that IP was transporting in this example.


File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply
A
B
X – IP header fields
Ready to load or capture
Packets: 445 Displayed: 445 Marked: 0 Dropped: 0
Profile: Default

```
No.  Time          Source           Destination      Protocol  Length  Info
  205 20.285810    192.168.1.122    192.168.1.125    TCP          54 62520 > 53754 [ACK] Seq=1 Ack=1 Win=

Frame 205: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)
Ethernet II, Src: Cisco-Li_b4:5a:95 (00:25:9c:b4:5a:95), Dst: Cisco-Li_7b:7a:f1 (00:23:69:7b:7a:f1)
Internet Protocol Version 4, Src: 192.168.1.122 (192.168.1.122), Dst: 192.168.1.125 (192.168.1.125)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
  Total Length: 40
  Identification: 0x67d8 (26584)
  Flags: 0x02 (Don't Fragment)
    0... .... = Reserved bit: Not set
    .1.. .... = Don't fragment: Set
    ..0. .... = More fragments: Not set
  Fragment offset: 0
  Time to live: 128
  Protocol: TCP (6)
  Header checksum: 0x0eb0 [correct]
  Source: 192.168.1.122 (192.168.1.122)
  Destination: 192.168.1.125 (192.168.1.125)
Transmission Control Protocol, Src Port: 62520 (62520), Dst Port: 53754 (53754), Seq: 1, Ack: 1, Len: 0

0000  00 23 69 7b 7a f1 00 25  9c b4 5a 95 08 00 45 00   .#i{z..% ..Z...E.
0010  00 28 67 d8 40 00 80 06  0e b0 c0 a8 01 7a c0 a8   .(g.@... .....z..
0020  01 7d f4 38 d1 fa 9a 60  fc 84 7a 93 58 20 50 10   .}.8...` ..z.X P.
0030  40 29 bb 86 00 00                                  @)....
```

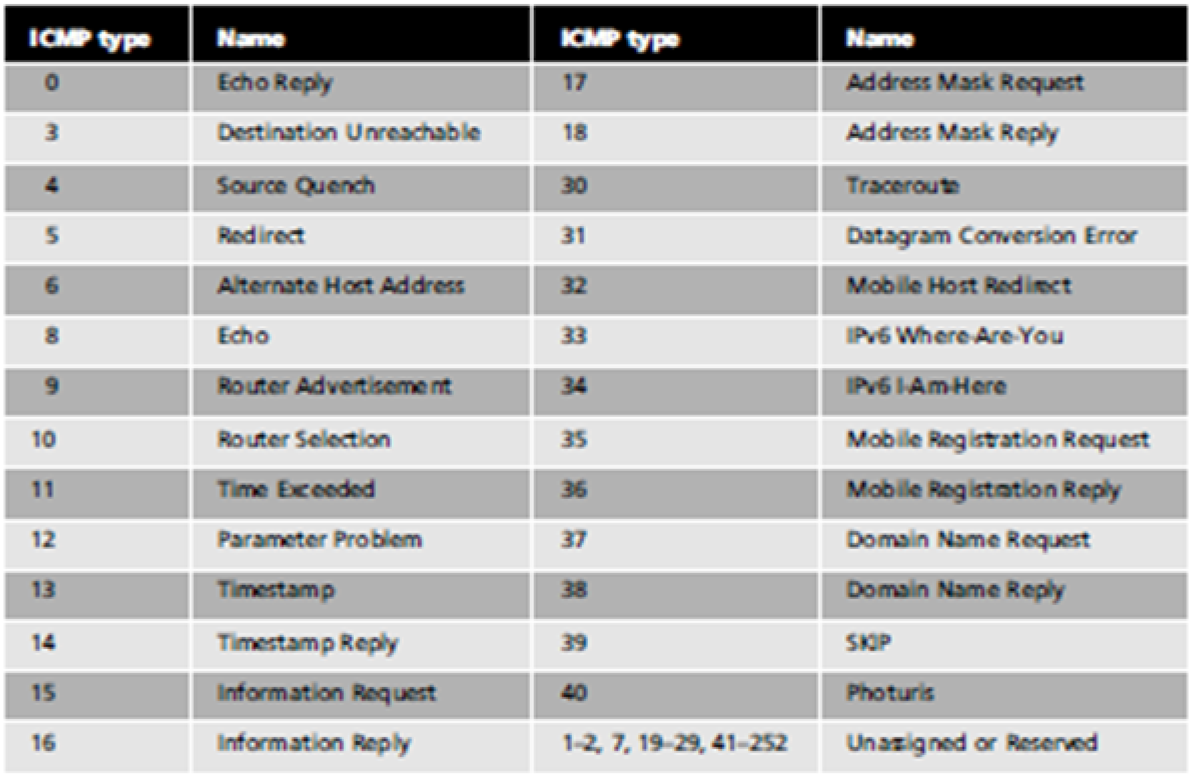
Figure 2-2 IP header structure as seen in a Wireshark packet capture

ICMP Messages

Internet Control Message Protocol (ICMP) is designed to assist TCP/IP networks with troubleshooting communication problems. For example, using the ping command, ICMP produces messages that indicate whether a host has connectivity with another host. Like IP, ICMP is processed at the Network layer of the OSI model.

Based on a packet's message type, a firewall or packet filter must be able to determine whether an ICMP packet should be allowed to pass. An administrator might want to allow specific ICMP packet types for network diagnostics while wanting to block other types that could be used as part of an attack. Table 2-7 lists some common ICMP types.

Many ICMP types have codes associated with them. Some common codes are used for type 3 (Destination Unreachable) ICMP messages. For example, a type 3 ICMP message with a code of 13 indicates that the message was administratively prohibited, which usually means that an access list or firewall rejected the message. You can find a complete list of ICMP types and their codes at www.iana.org/assignments/icmpparameters/icmp-parameters.xml.



ICMP type	Name	ICMP type	Name
0	Echo Reply	17	Address Mask Request
3	Destination Unreachable	18	Address Mask Reply
4	Source Quench	30	Traceroute
5	Redirect	31	Datagram Conversion Error
6	Alternate Host Address	32	Mobile Host Redirect
8	Echo	33	IPv6 Where-Are-You
9	Router Advertisement	34	IPv6 I-Am-Here
10	Router Selection	35	Mobile Registration Request
11	Time Exceeded	36	Mobile Registration Reply
12	Parameter Problem	37	Domain Name Request
13	Timestamp	38	Domain Name Reply
14	Timestamp Reply	39	SKIP
15	Information Request	40	Photuris
16	Information Reply	1-2, 7, 19-29, 41-252	Unassigned or Reserved

Table 2-7 ICMP types

TCP Headers

TCP/IP packets do not contain just IP header information. They might also contain TCP headers (shown in Figure 2-3) that provide hosts with a different set of flags—and that give attackers a different set of components they can misuse in an attempt to attack networks. TCP headers are processed at the Transport layer of the OSI model. The TCP portion of a packet is called a TCP segment.

The Flags section of a TCP header is the set of nine 1-bit fields identified in Figure 2-3. From a security standpoint, the flags are important because you can specify them when you create packet-filtering rules. For example, the TCP header portion of a TCP packet that has an acknowledgement (ACK) flag set to 1 rather than 0 indicates that the destination computer received the packets that were sent. The first three flags—NS, CWR, and ECE—are related to Explicit Congestion Notification (ECN). The next six flags—URG, ACK, PSH, RST, SYN, and FIN—are of particular importance from a security perspective.

- NS (Nonce Sum)—Associated with ECN
- CWR (Congestion Window Reduced)—Associated with ECN
- ECE (ECN Echo)—Associated with ECN
- URG (Urgent)—When set to 1, data should be considered significant
- ACK (Acknowledgement)—Indicates that the previous packet was received

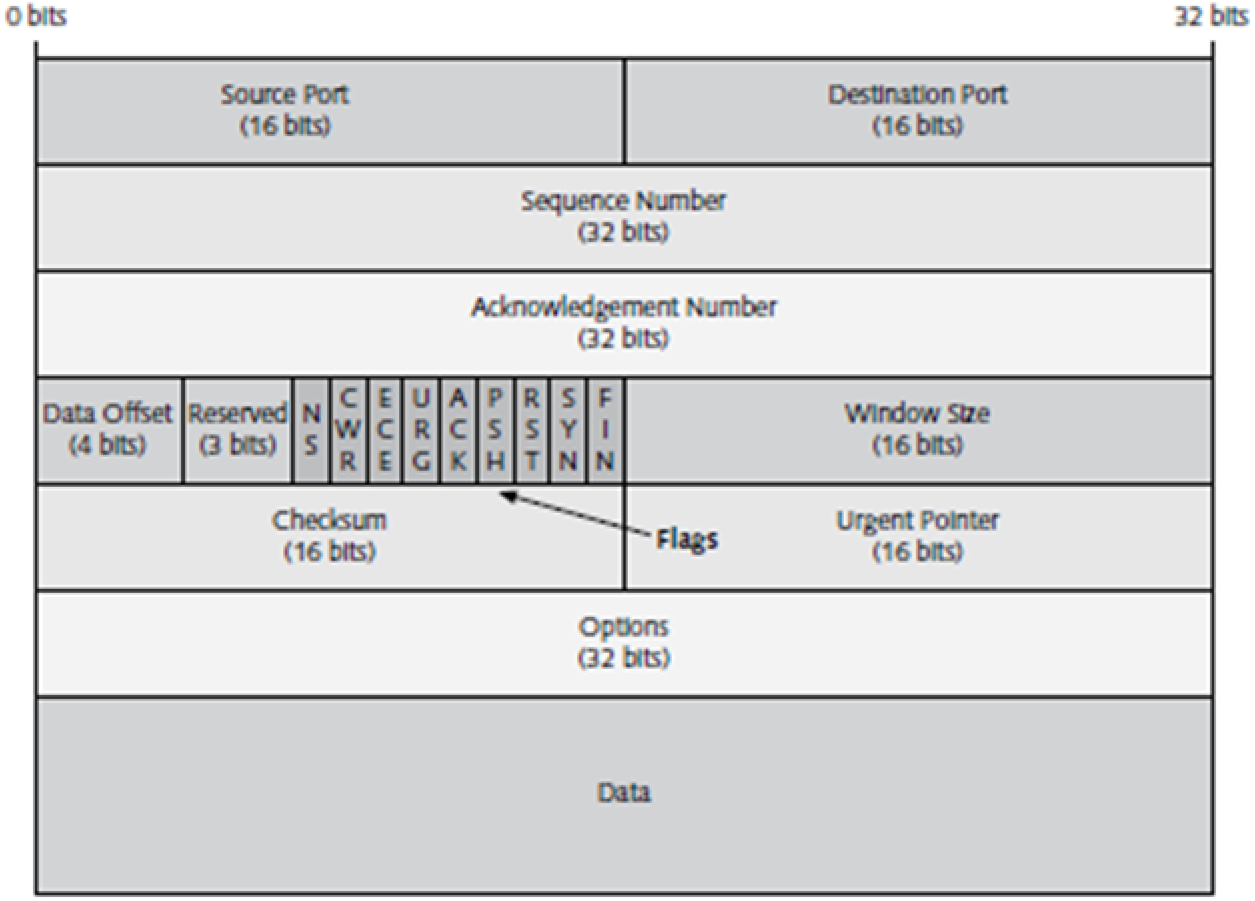


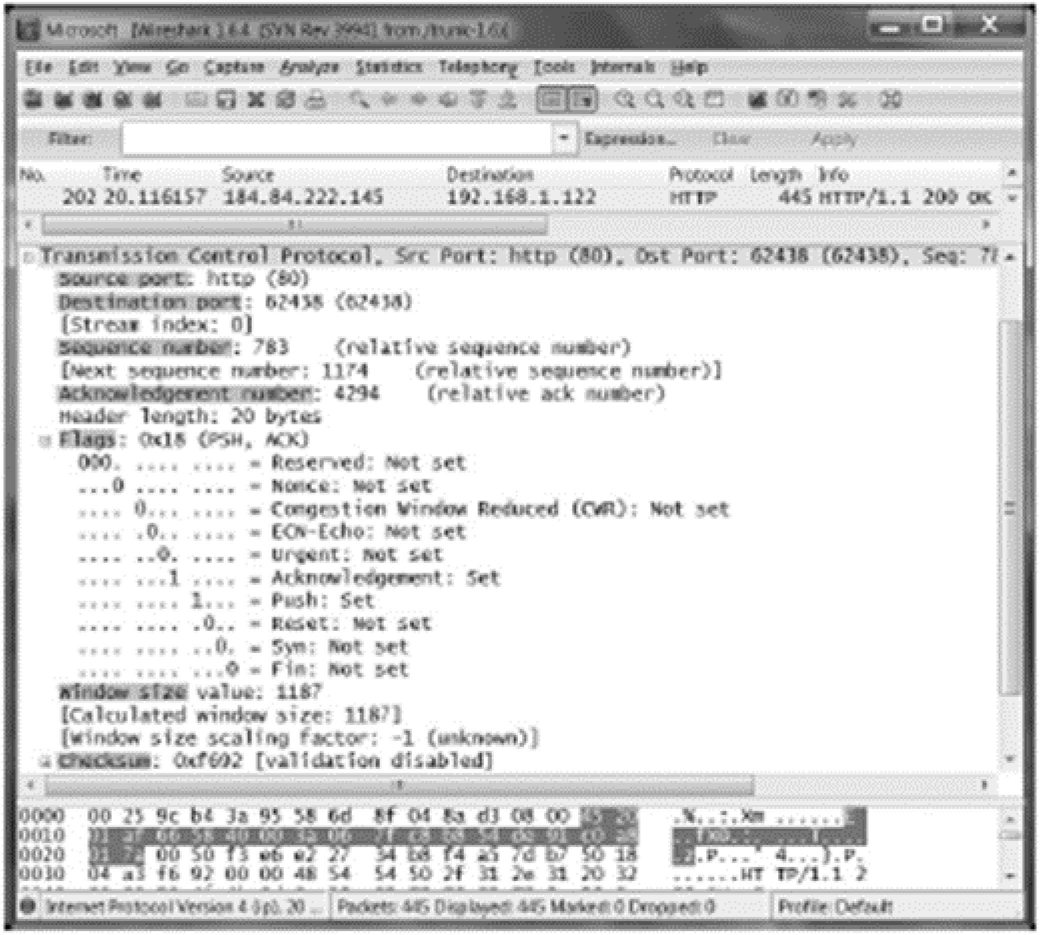
Figure 2-3 TCP header structure

- PSH (Push)—Forces TCP to deliver data rather than buffer it on the receiver
- RST (Reset)—Resets the connection
- SYN (Synchronize)—Synchronizes the sequence numbers
- FIN (Finish)—Indicates that no more data will come from the sender

Figure 2-4 shows how a TCP packet is displayed in a packet analyzer.

UDP Headers

The **User Datagram Protocol (UDP)**, like TCP, is processed at the Transport layer of the OSI model. This portion of a packet is called a UDP datagram and it provides a transport service for IP, but this protocol is considered unreliable because it is connectionless. In other words, a UDP packet does not contain the sequence number/acknowledgement number mechanism that enables TCP to guarantee delivery of the packet. (This connection-oriented mechanism is explained in detail later.) UDP is much faster than TCP because of the relative lack of overhead information in the header, and is appropriate when delivery does not need to be guaranteed. UDP simply sends the packets and relies on other protocols to ensure delivery perform error checking, and so on. It does not provide errors and depends on the application in use to notify the user of any errors in transmission. UDP is especially useful for real-time applications, multimedia, or anything that requires speed over reliability.


Microsoft [Wireshark 1.6.4 (SVN Rev 39941 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply

```
No.   Time          Source           Destination     Protocol  Length  Info
  202 20.116157  184.84.222.145   192.168.1.122   HTTP        445 HTTP/1.1 200 OK

Transmission Control Protocol, Src Port: http (80), Dst Port: 62438 (62438), Seq: 7
  Source port: http (80)
  Destination port: 62438 (62438)
  [Stream index: 0]
  Sequence number: 783    (relative sequence number)
  [Next sequence number: 1174    (relative sequence number)]
  Acknowledgement number: 4294    (relative ack number)
  Header length: 20 bytes
  Flags: 0x18 (PSH, ACK)
    000. .... .... = Reserved: Not set
    ...0 .... .... = Nonce: Not set
    .... 0... .... = Congestion Window Reduced (CWR): Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... ...1 .... = Acknowledgement: Set
    .... .... 1... = Push: Set
    .... .... .0.. = Reset: Not set
    .... .... ..0. = Syn: Not set
    .... .... ...0 = Fin: Not set
  Window size value: 1187
  [Calculated window size: 1187]
  [Window size scaling factor: -1 (unknown)]
  Checksum: 0xf692 [validation disabled]

0000  00 25 9c b4 3a 95 58 6d  8f 04 8a d3 08 00 [illegible]   .%..:.Xm ......[illegible]
0010  [illegible]                                                [illegible]
0020  [illegible] 00 50 f3 a6 e2 27  34 b8 f4 a5 7d b7 50 18   [illegible].P...' 4...}.P.
0030  04 a3 f6 92 00 00 48 54  54 50 2f 31 2e 31 20 32   ......HT TP/1.1 2
```

Internet Protocol Version 4 (ip), 20 ...
Packets: 445 Displayed: 445 Marked: 0 Dropped: 0
Profile: Default

Figure 2-4 TCP header structure as seen in a Wireshark packet capture

UDP is used for broadcasting messages or for protocols that do not require the same level of service as TCP. For example, Simple Network Management Protocol (SNMP) and Trivial File Transfer Protocol (TFTP) are normally used on LANs, where packet loss is not considered a serious problem. On the other hand, attackers can scan for open UDP services to exploit by sending empty UDP datagrams to a suspected open port. If the port is closed, the system sends back an ICMP Destination Unreachable message (type 3). UDP packets have their own headers, as shown in Figure 2-5. As you can see, the UDP header is much smaller and simpler than a TCP header. In Figure 2-6, you can see how a UDP packet is displayed in a packet analyzer.

Packet Fragmentation

Fragmentation of IP packets was originally developed as a means of allowing large packets to pass through routers that had frame size limitations. Routers were then able to divide packets into multiple fragments and send them along the network, where receiving routers reassembled them in the correct order and passed them along to their destination.

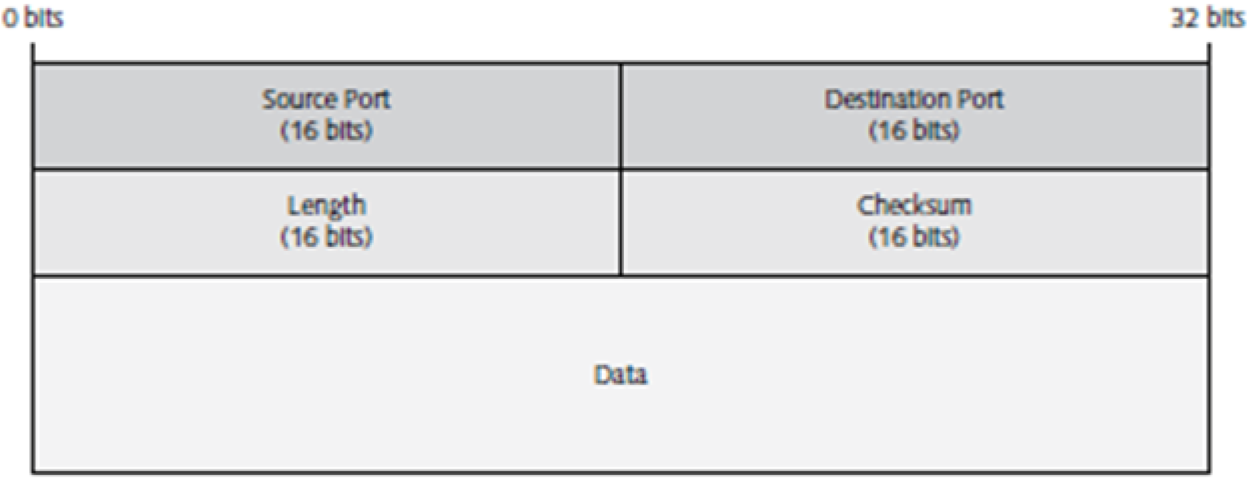
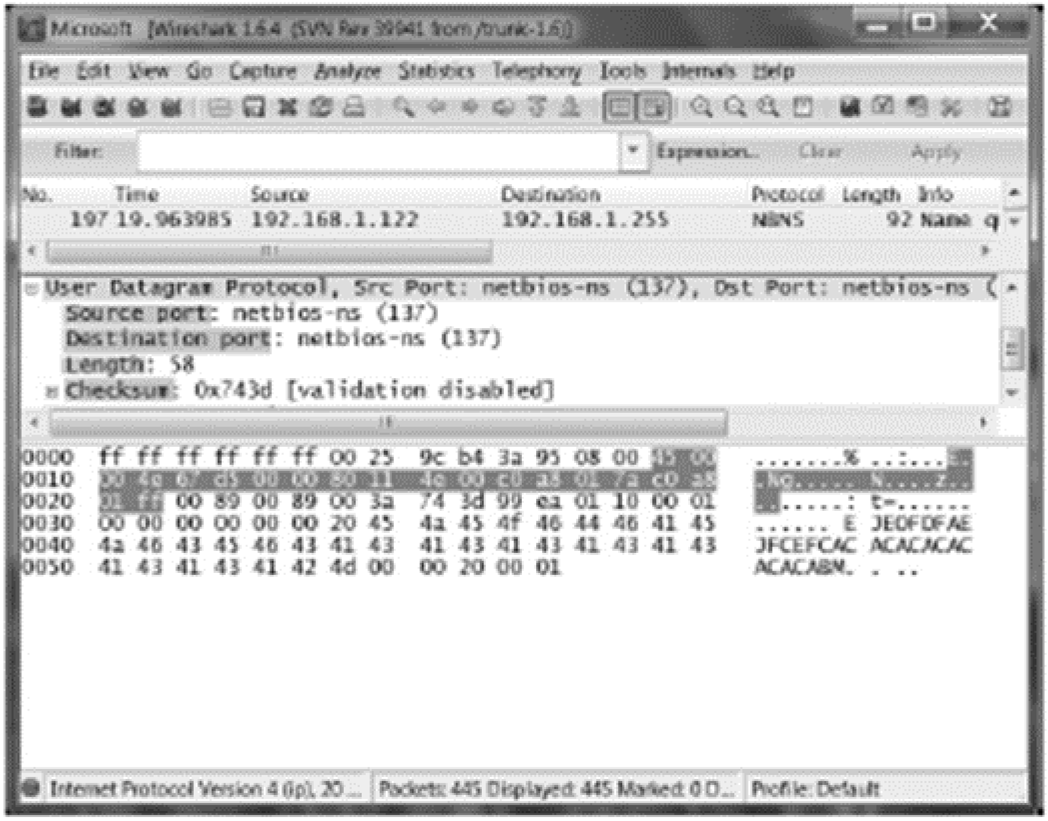


Figure 2-5 UDP header structure


Microsoft [Wireshark 1.6.4 (SVN Rev 39941 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply

```
No.   Time       Source          Destination     Protocol Length Info
  197 19.963985  192.168.1.122   192.168.1.255   NBNS         92 Name q
```

```
User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (
    Source port: netbios-ns (137)
    Destination port: netbios-ns (137)
    Length: 58
  Checksum: 0x743d [validation disabled]
```

```
0000  ff ff ff ff ff ff 00 25  9c b4 3a 95 08 00 45 00   .......% ..:...E.
0010  00 4e 67 d5 00 00 80 11  4e 00 c0 a8 01 7a c0 a8   .Ng..... N....z..
0020  01 ff 00 89 00 89 00 3a  74 3d 99 ea 01 10 00 01   .......: t=......
0030  00 00 00 00 00 00 20 45  4a 45 4f 46 44 46 41 45   ...... E JEOFDFAE
0040  4a 46 43 45 46 43 41 43  41 43 41 43 41 43 41 43   JFCEFCAC ACACACAC
0050  41 43 41 43 41 42 4d 00  00 20 00 01              ACACABM. . ..
```

Internet Protocol Version 4 (ip), 20 ... | Packets: 445 Displayed: 445 Marked: 0 D... | Profile: Default

Figure 2-6 UDP header structure as seen in a Wireshark packet capture

Fragmentation creates a number of security problems, however. Because the TCP or UDP port number is supplied only at the beginning of a packet, it appears only in fragment number 0. Fragments numbered 1 or higher pass through the filter without being scrutinized because they contain no port information. An attacker simply has to modify the IP header to make all fragment numbers start at 1 or higher. All fragments then go through the filter and can access internal resources.

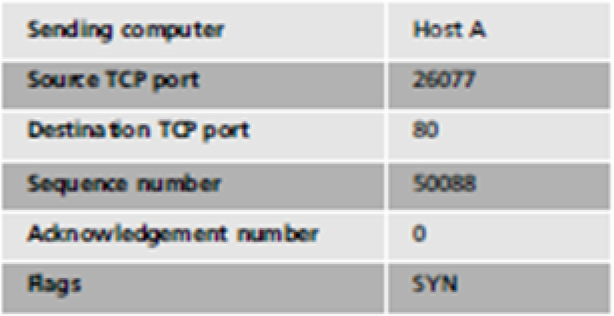
To improve security, you should configure the firewall or packet filter to drop all fragmented packets. You could also have the firewall reassemble fragmented packets and allow only complete packets to pass through. Note that fragmentation is seldom used today because of the improvements in routers.

The TCP Life Cycle and the TCP Three-Way Handshake

Before a client initiates a connection-oriented TCP session with another computer, it must establish which of its own ports it will use as a source of communication and establish the destination port on the other computer. Typically, a client, such as Host A in the following example, will dynamically assign itself a source port on which it will communicate and will know what port is appropriate for the destination port on the other computer. For example, Web servers listen for requests for HTTP service on their port 80; therefore, if Host A were trying to access a Web page on Host B, Host A would set port 80 as the destination port.

To establish connection-oriented communication, each computer needs a way to know that the other computer received the packets sent. Sequence and acknowledgement numbers perform this function, as demonstrated in the way that two hosts first establish the TCP connection: the TCP three-way handshake.

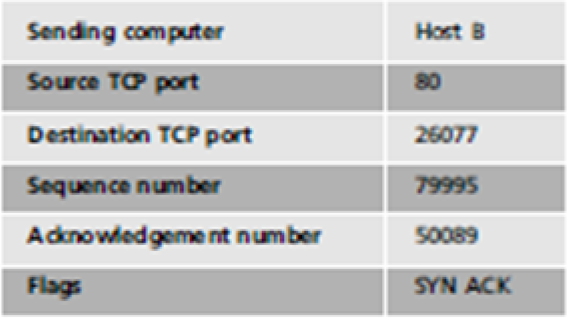
- Host A includes a randomly generated initial sequence number in its first packet to Host B. This packet is called a SYN packet because the TCP SYN flag is set. The acknowledgement number is zero because the SYN packet is the first in the session and there is no previous packet for Host A to acknowledge. Table 2-8 shows parts of the SYN packet configuration for this example.



Sending computer	Host A
Source TCP port	26077
Destination TCP port	80
Sequence number	50088
Acknowledgement number	0
Flags	SYN

Table 2-8 TCP three-way handshake: SYN

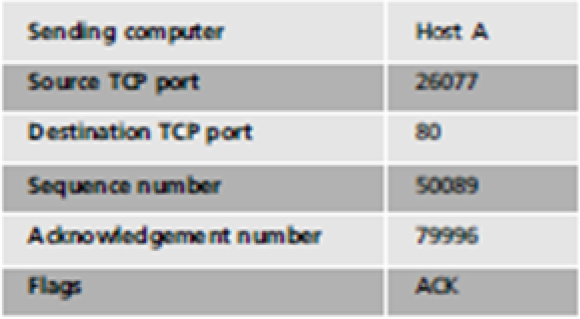
- Host B receives the SYN packet and responds with a SYN ACK packet. This packet includes a randomly generated initial sequence number for Host B. As a way of proving that Host B received the SYN packet from Host A, the acknowledgement number is set to the number that Host B expects to receive in the second packet from Host A. The first packet's sequence number is incremented by one and placed as the acknowledgement number. This configuration is shown in Table 2-9.



Sending computer	Host B
Source TCP port	80
Destination TCP port	26077
Sequence number	79995
Acknowledgement number	50089
Flags	SYN ACK

Table 2-9 TCP three-way handshake: SYN ACK

The final packet in the three-way handshake is the ACK packet that Host A sends in response to the SYN ACK from Host B. Now Host A increments its initial sequence number by one and sets the acknowledgement number to be one more than the initial sequence number that Host B sent in the SYN ACK (see Table 2-10).



Sending computer	Host A
Source TCP port	26077
Destination TCP port	80
Sequence number	50089
Acknowledgement number	79996
Flags	ACK

Table 2-10 TCP three-way handshake: ACK

More goes on in this three-way handshake than acknowledgements. For example, sliding window size is negotiated. After a connection is established, TCP sliding windows control the flow and efficiency of communications. During the transfer of large amounts of data, it would be very inefficient if each packet had to be received and acknowledged before the next packet could be sent. TCP/IP is designed for packet-switching networks where packets might arrive out of order. Sliding window size determines the number of packets that can be sent before ACKs must be received. The sender controls the size of the sliding window.

Once data has been exchanged, either party can end the session by sending a packet with the FIN flag set. The station that receives this initial flag sends a response packet with the ACK flag and its own FIN flag set to acknowledge receipt, and to show that it is also ready to end communications. If the receiving side still has data to send, however, it sends only an ACK flag back and continues sending data until it is done.

After all data has been sent, the side that first received the FIN flag sends its own FIN flag to show that termination can begin. During this lapse between the two flags, the sender of the first FIN flag has a status of FIN WAIT 2, and the recipient of the first FIN flag has a CLOSE WAIT status. Some applications, such as Web browsers, often use this type of half-closed connection to avoid having to initiate the session again. Figure 2-7 summarizes the TCP three-way handshake.

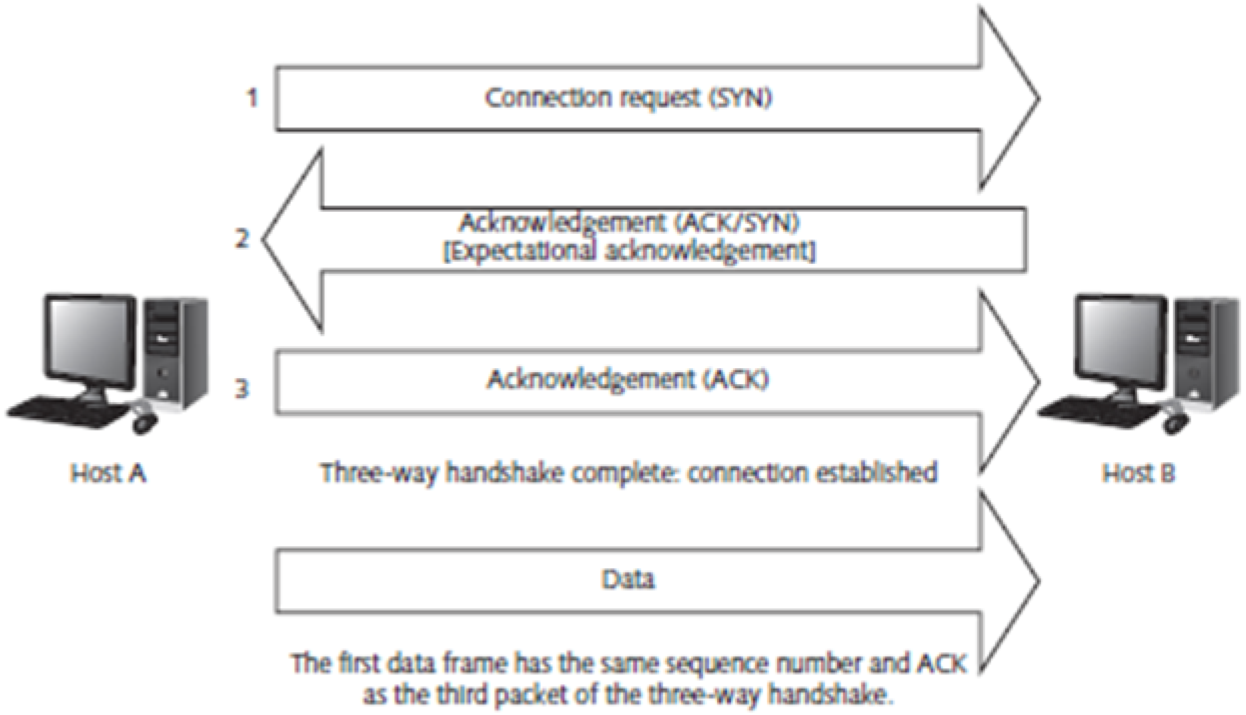

1
Connection request (SYN)
2
Acknowledgement (ACK/SYN)
[Expectational acknowledgement]
3
Acknowledgement (ACK)
Host A
Three-way handshake complete: connection established
Host B
Data
The first data frame has the same sequence number and ACK
as the third packet of the three-way handshake.

Figure 2-7 Summary of the TCP three-way handshake

Domain Name System

Domain Name System (DNS) is a general-purpose service used both on the Internet and on organizations' internal networks. DNS servers translate **fully qualified domain names (FQDNs)** to IP addresses that identify the host computer. For example, to connect to Web sites, users need access to an authoritative DNS server for the system domain they are trying to reach. If a user enters the FQDN www.cengage.com, the authoritative DNS server for cengage.com needs to respond with the correct IP addresses for the computer whose hostname is www.

In terms of network security, DNS is important because it gives network administrators another tool for blocking unwanted communication. With firewalls, Web browsers, and proxy servers, administrators can block DNS names of Web sites and other sites that contain offensive or unsuitable content. Proxy servers are devices that protect internal clients through the use of network address translation. In addition, networks that use DNS servers need to allow DNS traffic when packet filtering is set up.

Attackers can exploit DNS in many ways, including buffer overflow attacks, zone transfer attacks, and cache poisoning attacks. In a DNS buffer overflow attack, an overly long DNS name is sent to the server. When the server is unable to process or interpret the DNS name, it cannot process other requests.

DNS zone files contain a list of every DNS-configured host on a network as well as their IP addresses. Microsoft DNS-enabled networks also list all services running on DNS-configured hosts. When an attacker attempts to penetrate a network, the DNS zone file can provide a list of exploitable targets on that network. When configuring DNS servers connected to the Internet, you should disable zone transfers to all hosts except those that are internal to the network. Internal hosts must be able to transfer zone information to update their records.

A DNS cache poisoning attack exploits the fact that every DNS packet contains a Query section and a Reply section. An older, more vulnerable server has stored answers sent in response to requests to connect to DNS addresses. Attackers can break into the cache to discover the DNS addresses of computers on the network. Most DNS servers, however, have been patched to eliminate this vulnerability.

A newer DNS cache poisoning exploit was discovered by Dan Kaminsky in 2008. This exploit involves the spoofing of transaction IDs, which are supposed to prevent hackers from assigning their own IP addresses to a domain. DNS uses transaction IDs in the range of 0 to 65535. If a hacker sends multiple, slightly varied requests to a name server (such as requests to resolve 1.frog.com, 2.frog.com, and so on), eventually the domain can be spoofed by matching the ID. Once the attacker correctly matches the transaction ID, he can direct all traffic for that site to a site of his choosing. An attacker can also pollute top-level domains using this vulnerability. Prior to releasing details of the exploit, Kaminsky notified vendors and allowed them time to develop a patch that focused on randomizing port numbers.

Originally, the DNS infrastructure did not have optimum security. In 2005, the first implementations of DNSSEC (DNS Security) were rolled out in Sweden in the .se domain. DNSSEC uses cryptographic techniques to enable authentication and data integrity of DNS packets, eliminating vulnerabilities that allow exploitations such as cache poisoning. Unfortunately, DNSSEC is still not widely used, partly because an enormous amount of work is required to revise existing DNS implementations and to establish the complex cryptographic infrastructure. (For more details on cryptography, see Chapter 5.)

Internet Protocol Version 6 (IPv6)

IPv4 has serious drawbacks. Although it was an engineering masterpiece in 1981, the Internet has grown at a rate that the creators of the IPv4 32-bit addressing scheme did not expect. IP addresses are now in short supply, so Internet Protocol version 6 (IPv6), which has a larger address space of 128 bits, is being deployed to allow an almost endless supply of IP addresses.

Because an IPv4 address is 32 bits long, IPv4 permits a total of 232 addresses, which is more than 4 billion. With 128 bits, IPv6 offers 2128 addresses, which is 340 undecillion. An undecillion is a 1 followed by 39 zeros.

IPv4 also presents problems with the routing system. Routers on the Internet backbone have routing tables with about 90,000 entries. Routers get the job done, but because most computers are not connected directly to the Internet backbone, a packet must traverse several extra hops along the route to its destination. In IPv6, backbone routing tables need only the entries of other routers that are connected directly to them. The information in an IPv6 header contains the rest of the information needed to get a packet to its destination, so the process is streamlined.

Security is another concern with IPv4. Although it does support IPsec (an industry standard set of encryption and authentication protocols), IPv4 has no native encryption methods. Plenty of encryption methods are available, but the lack of standardization can create compatibility problems, and encryption can increase overhead on the network. IPv6, on the other hand, has integrated support for IPsec.

Another advantage of IPv6 is that Network Address Translation (NAT) is not needed because of the vast number of IP addresses provided. While NAT has worked well enough to deal with the decreasing number of IP addresses in IPv4, NAT has security problems, as you will see in Chapter 11 when you learn about VPNs. In short, because NAT devices need to read encapsulated IP headers, it is difficult to maintain data confidentiality for end-to-end transmissions; typically, the packets are unencrypted by the NAT firewall and sent through the internal network unencrypted. IPv6 obviates this problem.

Another major advantage of IPv6 is its autoconfiguration capabilities. Instead of relying solely on Dynamic Host Configuration Protocol (DHCP) or manual configuration, IPv6 can determine its own settings based on two different models. Stateful autoconfiguration, provided by DHCPv6, is referred to as "stateful" because the DHCPv6 client and server must keep their information updated to prevent addressing conflicts. With stateless

autoconfiguration, a computer that is trying to connect to a network can determine its own IP address based on its Media Access Control (MAC) address and then can receive additional configurations without a DHCP server. This process simplifies some aspects of network administration because a server does not need to issue address configurations. In the next few sections, you learn about the core IPv6 protocols and see how the next generation of IP works.

◀ Preliminary Activity for Week 4

Jump to...

Analysis, Application, and Exploration for Week 4 ▶

Navigation

Home

Dashboard

Site pages

My courses

Capstone Project 2

Network Defense and Remote Access Configuration

Participants

General

01 [Enter Module Title Here]

02 [Enter Module Title Here]

03 [Enter Module Title Here]

04 [Enter Module Title Here]

- Preliminary Activity for Week 4
- **Lesson Proper for Week 4**
- Analysis, Application, and Exploration for Week 4
- Generalization for Week 4
- Evaluation for Week 4
- Assignment for Week 4
- 05 [Enter Module Title Here]
- OJT/Practicum 2
- Seminars and Tours
- Courses

Fair Warning

NOTICE: Please be reminded that it has come to the attention of the Publishing Team of eLearning Commons that learning materials published and intended for ***free use only by students and faculty members within the eLearning Commons network were UNLAWFULLY uploaded in other sites without due and proper permission***.

PROSECUTION: Under Philippine law (Republic Act No. 8293), copyright infringement is punishable by the following: Imprisonment of between 1 to 3 years and a fine of between 50,000 to 150,000 pesos for the first offense. Imprisonment of 3 years and 1 day to six years plus a fine of between 150,000 to 500,000 pesos for the second offense.

COURSE OF ACTION: Whoever has maliciously uploaded these concerned materials are hereby given an ultimatum to take it down within 24-hours. Beyond the 24-hour grace period, our Legal Department shall initiate the proceedings in coordination with the National Bureau of Investigation for IP Address tracking, account owner identification, and filing of cases for prosecution.

Activities

- Assignments
- Forums
- Quizzes
- Resources