WILEY eBOOK

Hack Attacks Revealed

A Complete Reference with Custom Security Hacking Toolkit

John Chirillo

Wiley Computer Publishing

John Wiley & Sons, Inc.

NEW YORK • CHICHESTER • WEINHEIM • BRISBANE • SINGAPORE • TORONTO

Publisher: Robert Ipsen

Editor: Carol A. Long

Assistant Editor: Adaobi Obi

Managing Editor: Micheline Frederick

New Media Editor: Brian Snapp

Text Design & Composition: Thomark Design

Published by John Wiley & Sons, Inc.

This title is also available in print as ISBN 0-471-41624-X

For more information about Wiley products, visit our website at www.wiley.com.

Contents

Acknowledgments

Foremost I would like to thank my wife for for her continued support and patience during this book's development, as well as for proofing this book. Next I want to thank my family and friends for their encouragement, support, and confidence. I am also grateful to Mike Tainter and Dennis Cornelius for some early ideas. I also want to express my admiration for programming guru Michael Probert for his participation on coding fundamentals.

Thanks also to the following: Shadowlord, Mindgame, Simple Nomad, The LAN God, Teiwaz, Fauzan Mirza, David Wagner, Diceman, Craigt, Einar Blaberg, Cyberius, Jungman, RX2, itsme, Greg Miller, John Vranesevich, Deborah Triant, Mentor, the FBI, The National Computer Security Center, 2600.com, Fyodor, Muffy Barkocy, Wintermute, dcypher, manicx, Tsutomu Shimomura, humble, The Posse, Jim Huff, Soldier, Mike Frantzen, Tfreak, Dan Brumleve, Arisme, Georgi Guninski, Satanic Mechanic, Mnemonic, The Grenadier, Jitsu, lore, 416, all of the H4G1S members, everyone at ValCom, and to Bruce Schneier, who inspired me.

Someone once told me in order to be successful, one must surround oneself with the finest people. With that in mind, I thank David Fugate from Waterside Productions, and Carol Long, Mathew Cohen, Adaobi Obi, Micheline Frederick, and anyone else I forgot to mention from John Wiley & Sons, Inc.

A Note to the Reader

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. We cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

This book is sold for information purposes only. Without written consent from the target company, most of these procedures are illegal in the United States and many other countries as well. Neither the author nor the publisher will be held accountable for the use or misuse of the information contained in this book.

Introduction

We are the technologically inclined and normality spurned, or at least, this is how we perceive (or perhaps want) things to be. We are adept at dealing with machines, and manipulating things. Everything comes easy to us, and when things always come to you without any failure, you begin to feel nothing matters...that the world is rigged. Perhaps, this is why we always look for conspiracies, and when they don't exist, we create them ourselves. Maybe I will tap another military switch...

Why are we like this?

We are different from other people, and those others cannot always accept this. We ourselves are not racists, or sexists, or idealists. We do not feel that other people will understand us. Those of us electronically gathered here are alike, but in the real world we are so few and far between that we do not feel comfortable in normal society.

We quickly grasp concepts, and, because of our manipulative nature, quickly see through those who are lying. They cannot deceive us. We don't care. There are systems to hack. In reality, we care about much more, but can't very well affect it.

We are dazed and confused technological mall rats waiting for the apocalypse. When will it come? We are ready, and want it. If it doesn't show up...we will be jilted at our millennial altar. Maybe we will create it. Or at least dream about it. Anarchy?

Dark visions, from an apathetic crowd.

And yet, we are not technogoths, waiting for some distant, terrible, cyberdistopia. We have lives, and want to live. We are sick of hearing from a select few that we are "different." To us, the young generation going into the next millennium, the young generation brought together by technology and in technology,

> the word "different" shouldn't matter. We are all "different," all abnormal...but it should have no impact.
>
> Those of us on the brink of technology, falling over, laugh at those who do not understand technology. They embody the Old World, driven by race and prior position in society. We laugh at them for being "different," because they refuse to be apathetic about difference. Why can't they be different like us?
>
> Microsoft asked where I want to go today. The only place I want to go is straight to tomorrow. I am a hacker of the future and this is my manifesto…
>
> —Mindgame

As the world becomes increasingly networked through the Internet, competitors, spies, disgruntled employees, bored teens, and hackers more frequently invade others' computers to steal information, sabotage careers, and just to make trouble. Together, the Internet and the World Wide Web have opened a new backdoor through which a remote attacker can invade home computers or company networks and electronically snoop through the data therein. According to my experiences, approximately 85 percent of the networks wired to the Internet are vulnerable to such threats.

The continued growth of the Internet, along with advances in technology, mean these intrusions will become increasingly prevalent. Today, external threats are a real-world problem for any company with connectivity. To ensure that remote access is safe, that systems are secure, and that security policies are sound, users in all walks of life need to understand the hacker, know how the hacker thinks—in short, become the hacker.

The primary objective of this book is to lay a solid foundation from which to explore the world of security. Simply, this book tells the truth about hacking, to bring awareness about the so-called Underground, the hacker's community, and to provide the tools for doing so.

The book is divided into six parts:

Part 1: In the Beginning

- Chapter 1: Understanding Communication Protocols
- Chapter 2: NetWare and NetBIOS Technology

Part 2: Putting It All Together

- Chapter 3: Understanding Communication Mediums

Part 3: Uncovering Vulnerabilities

- Chapter 4: Well-Known Ports and Their Services
- Chapter 5: Discovery and Scanning Techniques

Part 4: Hacking Security Holes

- Chapter 6: The Hacker's Technology Handbook

Chapter 7: Hacker Coding Fundamentals

Chapter 8: Port, Socket, and Service Vulnerability Penetrations

Part 5: Vulnerability Hacking Secrets

Chapter 9: Gateways and Routers and Internet Server Daemons

Chapter 10: Operating Systems

Chapter 11: Proxies and Firewalls

Part 6: The Hacker's Toolbox

Chapter 12: TigerSuite: The Complete Internetworking Security Toolbox

The difference between this book and other technical manuscripts is that it is written from a hacker's perspective. The internetworking primers in Parts 1 and 2, coupled with Chapter 6, "The Hacker's Technology Handbook, will educate you about the technologies required to delve into security and hacking. These chapters can be skimmed if your background is technically sound, and later used as references. Part 3 reviews in detail the tools and vulnerability exploits that rule "hackerdom." Part 4 continues by describing covert techniques used by hackers, crackers, phreaks, and cyberpunks to penetrate security weaknesses. Part 5 reveals hacking secrets of gateways, routers, Internet server daemons, operating systems, proxies, and firewalls. Part 6 concludes with the software and construction necessary for compiling a TigerBox, used by security professionals and hackers for sniffing, spoofing, cracking, scanning, spying, and penetrating vulnerabilities. Throughout this book you will also encounter Intuitive Intermissions, real-life interludes about hacking and the Underground. Through them you'll explore a hacker's chronicles, including a complete technology guide.

Who Should Read This Book

The cliché "the best defense is a good offense" can certainly be applied to the world of network security. Evaluators of this book have suggested that this book it may become a required reference for managers, network administrators (CNAs, MCPs), network engineers (CNEs, MCSEs), internetworking engineers (CCNA/P, CCIEs), even interested laypeople. The material in this book will give the members in each of these categories a better understanding of how to hack their network vulnerabilities.

More specifically, the following identifies the various target readers:

- The home or small home office (SOHO) Internet Enthusiast, whose web browsing includes secure online ordering, filling out forms, and/or transferring files, data, and information
- The network engineer, whose world revolves and around security

- The security engineer, whose intent is to become a security prodigy
- The hacker, cracker, and phreak, who will find this book both educational and entertaining
- The nontechnical manager, whose job may depend on the information herein
- The hacking enthusiast and admirer of such films as *Sneakers*, *The Matrix*, and *Hackers*
- The intelligent, curious teenager, whose destiny may become clear after reading these pages

As a reader here, you are faced with a challenging "technogothic" journey, for which I am your guide. Malicious individuals are infesting the world of technology. My goal is to help mold you become a virtuous hacker guru.

About the Author

Now a renowned superhacker who works on award-winning projects, assisting security managers everywhere, John Chirillo began his computer career at 12, when after a one-year self-taught education in computers, he wrote a game called Dragon's Tomb. Following its publication, thousands of copies were sold to the Color Computer System market. During the next five years, John wrote several other software packages including, The Lost Treasure (a game-writing tutorial), Multimanger (an accounting, inventory, and financial management software suite), Sorcery (an RPG adventure), PC Notes (GUI used to teach math, from algebra to calculus), Falcon's Quest I and II (a graphical, Diction-intensive adventure), and Genius (a complete Windows-based point-and-click operating system), among others. John went on to become certified in numerous programming languages, including QuickBasic, VB, C++, Pascal, Assembler and Java. John later developed the PC Optimization Kit (increasing speeds up to 200 percent of standard Intel 486 chips).

John was equally successful in school. He received scholarships including one to Illinois Benedictine University. After running two businesses, Software Now and Geniusware, John became a consultant, specializing in security and analysis, to prestigious companies, where he performed security analyses, sniffer analyses, LAN/WAN design, implementation, and troubleshooting. During this period, John acquired numerous internetworking certifications, including Cisco's CCNA, CCDA, CCNP, pending CCIE, Intel Certified Solutions Consultant, Compaq ASE Enterprise Storage, and Master UNIX, among others. He is currently a Senior Internetworking Engineer at a technology management company.

In the Beginning

CHAPTER
1

Understanding Communication Protocols

Approximately 30 years ago, communication protocols were developed so that individual stations could be connected to form a local area network (LAN). This group of computers and other devices, dispersed over a relatively limited area and connected by a communications link, enabled any station to interact with any other on the network. These networks allowed stations to share resources, such as laser printers and large hard disks.

This chapter and Chapter 2 discuss the communication protocols that became a set of rules or standards designed to enable these stations to connect with one another and to exchange information. The protocol generally accepted for standardizing overall computer communications is a seven-layer set of hardware and software guidelines known as the Open Systems Interconnection (OSI) model. Before one can accurately define, implement, and test (hack into) security policies, it is imperative to have a solid understanding of these protocols. These chapters will cover the foundation of rules as they pertain to TCP/IP, ARP, UDP, ICMP, IPX, SPX, NetBIOS, and NetBEUI.

A Brief History of the Internet

During the 1960s, the U.S. Department of Defense's Advanced Research Projects Agency (ARPA, later called DARPA) began an experimental wide area

network (WAN) that spanned the United States. Called ARPANET, its original goal was to enable government affiliations, educational institutions, and research laboratories to share computing resources and to collaborate via file sharing and electronic mail. It didn't take long, however, for DARPA to realize the advantages of ARPANET and the possibilities of providing these network links across the world.

By the 1970s, DARPA continued aggressively funding and conducting research on ARPANET, to motivate the development of the framework for a community of networking technologies. The result of this framework was the Transmission Control Protocol/Internet Protocol (TCP/IP) suite. (A *protocol* is basically defined as a set of rules for communication over a computer network.) To increase acceptance of the use of protocols, DARPA disclosed a less expensive implementation of this project to the computing community. The University of California at Berkeley's Berkeley Software Design (BSD) UNIX system was a primary target for this experiment. DARPA funded a company called Bolt Beranek and Newman, Inc. (BBN) to help develop the TCP/IP suite on BSD UNIX.

This new technology came about during a time when many establishments were in the process of developing local area network technologies to connect two or more computers on a common site. By January 1983, all of the computers connected on ARPANET were running the new TCP/IP suite for communications. In 1989, Conseil Europeén pour la Recherche Nucléaire (CERN), Europe's high-energy physics laboratory, invented the World Wide Web (WWW). CERN's primary objective for this development was to give physicists around the globe the means to communicate more efficiently using *hypertext*. At that time, hypertext only included document text with command tags, which were enclosed in <angle brackets>. The tags were used to markup the document's logical elements, for example, the title, headers and paragraphs. This soon developed into a language by which programmers could generate viewable pages of information called Hypertext Markup Language (HTML). In February 1993, the National Center for Supercomputing Applications at the University of Illinois (NCSA) published the legendary browser, Mosaic. With this browser, users could view HTML graphically presented pages of information.

At the time, there were approximately 50 Web servers providing archives for viewable HTML. Nine months later, the number had grown to more than 500. Approximately one year later, there were more than 10,000 Web servers in 84 countries comprising the World Wide Web, all running on ARPANET's backbone called the Internet.

Today, the Internet provides a means of collaboration for millions of hosts across the world. The current backbone infrastructure of the Internet can carry a volume well over 45 megabits per second (Mb), about one thousand

times the *bandwidth* of the original ARPANET. (Bandwidth is a measure of the amount of traffic a media can handle at one time. In digital communication, this describes the amount of data that can be transmitted over a communication line at bits per second, commonly abbreviated as bps.)

Internet Protocol

The Internet Protocol (IP) part of the TCP/IP suite is a four-layer model (see Figure 1.1). IP is designed to interconnect networks to form an Internet to pass data back and forth. IP contains addressing and control information that enables *packets* to be routed through this Internet. (A packet is defined as a logical grouping of information, which includes a header containing control information and, usually, user data.) The equipment—that is, routers—that encounter these packets, strip off and examine the *headers* that contain the sensitive routing information. These headers are modified and reformulated as a packet to be passed along.

Packet headers contain control information (route specifications) and user data. This information can be copied, modified, and/or spoofed (masqueraded) by hackers.

One of the IP's primary functions is to provide a permanently established connection (termed connectionless), unreliable, best-effort delivery of *datagrams* through an Internetwork. Datagrams can be described as a logical grouping of information sent as a network layer unit over a communication medium. IP datagrams are the primary information units in the Internet. Another of IP's principal responsibilities is the fragmentation and reassembly of datagrams to support links with different transmission sizes.

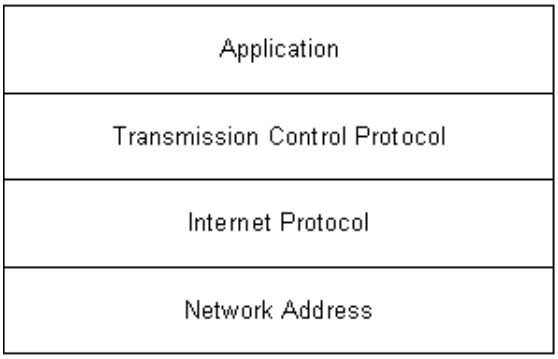

Figure 1.1 The four-layer TCP/IP model.

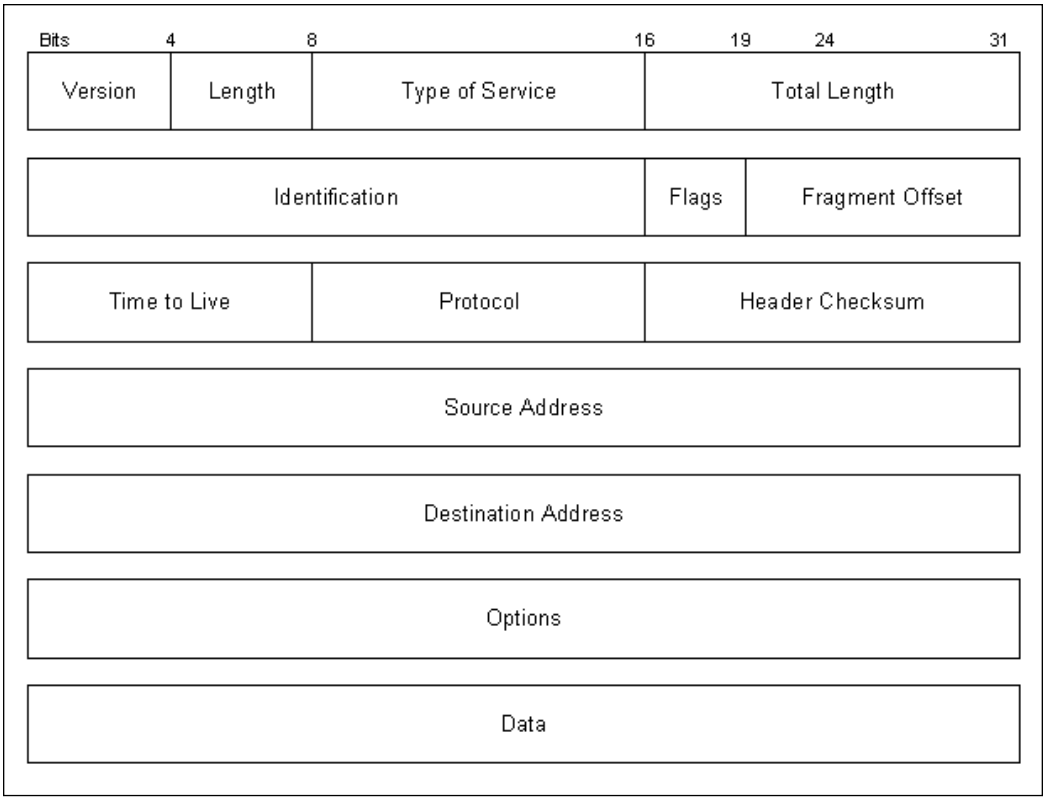

Figure 1.2 An IP packet.

During an analysis session, or *sniffer capture*, it is necessary to differentiate between different types of packet captures. The following describes the IP packet and the 14 fields therein, as illustrated in Figure 1.2.

Version. The IP version currently used.

IP Header Length (Length). The datagram header length in 32-bit words.

Type-of-Service (ToS). How the upper-layer protocol (the layer immediately above, such as transport protocols like TCP and UDP) intends to handle the current datagram and assign a level of importance.

Total Length. The length, in bytes, of the entire IP packet.

Identification. An integer used to help piece together datagram fragments.

Flag. A 3-bit field, where the first bit specifies whether the packet can be fragmented. The second bit indicates whether the packet is the last fragment in a series. The final bit is not used at this time.

Fragment Offset. The location of the fragment's data, relative to the opening data in the original datagram. This allows for proper reconstruction of the original datagram.

Time-to-Live (TTL). A counter that decrements to zero to keep packets from endlessly looping. At the zero mark, the packet is dropped.

Protocol. Indicates the upper-layer protocol receiving the incoming packets.

Header Checksum. Ensures the integrity of the IP header.

Source Address/Destination Address. The sending and receiving nodes (station, server, and/or router).

Options. Typically, contains security options.

Data. Upper-layer information.

Key fields to note include the Source Address, Destination Address, Options, and Data.

Now let's look at actual sniffer snapshots of IP Headers in Figures 1.3a and 1.3b to compare with the fields in the previous figure.

```
  ----- IP Header -----
IP:
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    = 60 bytes
IP: Identification  = 59136
IP: Flags           = 0X
IP:       .0.. .... = may fragment
IP:       ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live    = 32 seconds/hops
IP: Protocol        = 1 (ICMP)
IP: Header checksum = 0376 (correct)
IP: Source address      = [172.29.44.14]
IP: Destination address = [172.29.44.2]
IP: No options
```

Figure 1.3a Extracted during the transmission of an Internet Control Message Protocol (ICMP) ping test (ICMP is explained later in this chapter).

```
  ----- IP Header -----
IP:
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    = 112 bytes
IP: Identification  = 4864
IP: Flags           = 4X
IP:       .1.. .... = don't fragment
IP:       ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live    = 32 seconds/hops
IP: Protocol        = 6 (TCP)
IP: Header checksum = FA8F (correct)
IP: Source address      = [10.55.28.117]
IP: Destination address = [10.55.28.22]
IP: No options
```

Figure 1.3b Extracted during the transmission of a NetBIOS User Datagram Protocol (UDP) session request (these protocols are described later in this chapter and in Chapter 2).

IP Datagrams, Encapsulation, Size, and Fragmentation

IP datagrams are the very basic, or fundamental, transfer unit of the Internet. An IP datagram is the unit of data commuted between IP modules. IP datagrams have headers with fields that provide routing information used by infrastructure equipment such as routers (see Figure 1.4).

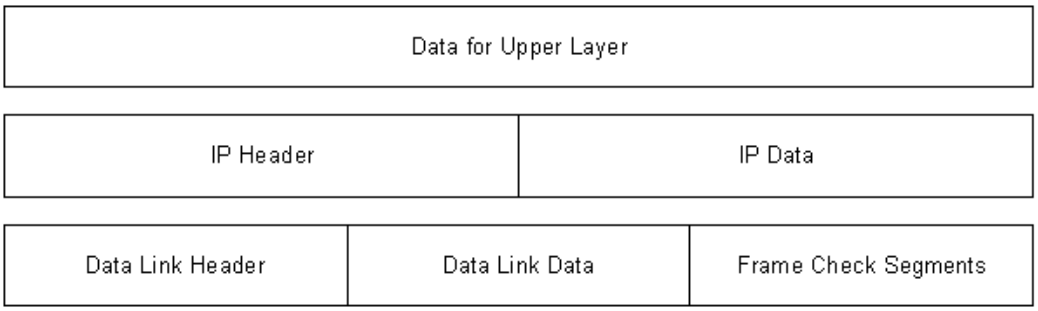

Figure 1.4 An IP datagram.

Be aware that the data in a packet is not really a concern for the IP. Instead, IP is concerned with the control information as it pertains to the upper-layer protocol. This information is stored in the IP header, which tries to deliver the datagram to its destination on the local network or over the Internet. To understand this relationship, think of IP as the method and the datagram as the means.

The IP header is the primary field for gathering information, as well as for gaining control.

It is important to understand the methods a datagram uses to travel across networks. To sufficiently travel across the Internet, over physical media, we want some guarantee that each datagram travels in a physical frame. The process of a datagram traveling across media in a frame is called *encapsulation*.

Now, let's take a look at an actual traveling datagram scenario to further explain these traveling datagram methods (see Figure 1.5). This example includes corporate connectivity between three branch offices, over the Internet, linking Ethernet, Token Ring, and FDDI (Fiber Distributed Data Interface) or fiber redundant Token Ring networks.

An ideal situation is one where an entire IP datagram fits into a frame; and the network it is traveling across supports that particular transfer size. But as

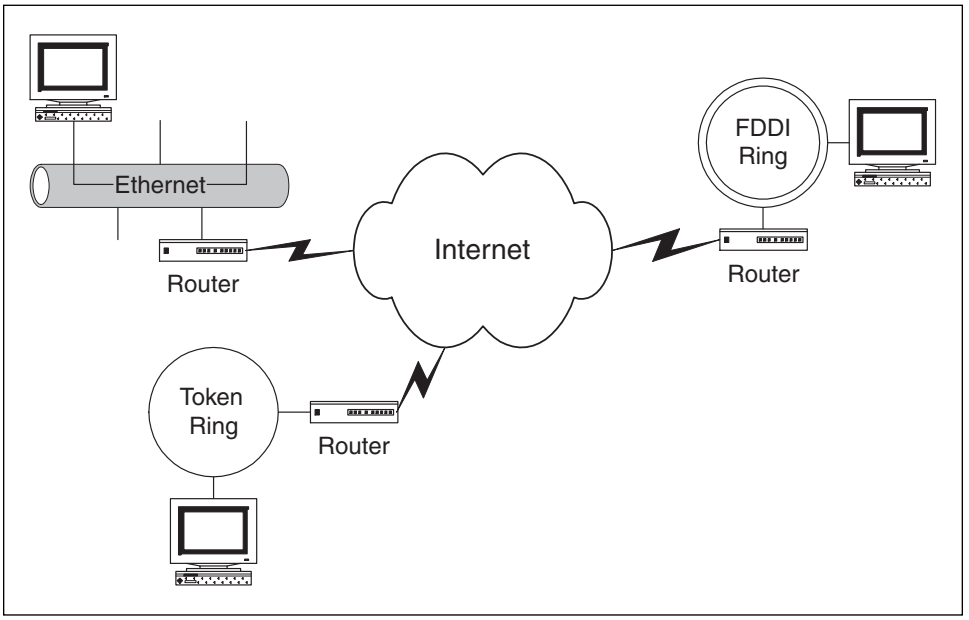

Figure 1.5 Real-world example of a traveling datagram.

we all know ideal situations are rare. One problem with our traveling datagram is that networks enforce a maximum transfer unit (MTU) size, or limit, on the size of transfer. To further confuse the issue, different types of networks enforce their own MTU; for example, Ethernet has an MTU of 1500, FDDI uses 4470 MTU, and so on. When datagrams traveling in frames cross network types with different specified size limits, routers must sometimes divide the datagram to accommodate a smaller MTU. This process is called *fragmentation*.

Routers provide the fragmentation process of datagrams, and as such, become vulnerable to passive and intrusive attacks.

IP Addresses, Classes, Subnet Masks

Communicating on the Internet would be almost impossible if a system of unique addressing were not used. To prevent the use of duplicate addresses, routing between nodes is based on addresses assigned from a pool of classes, or range of available addresses, from the InterNetwork Information Center (InterNIC). InterNIC assigns and controls all network addresses used over the Internet by assigning addresses in three classes (A, B, and C), which consist of 32-bit numbers. By default, the usable bits for Classes A, B, and C are 8, 16, and 24 respectively. Addresses from this pool have been assigned and utilized

Class	First Octet or Series	Octets as Network vs. Host	Netmask Binary
A	1 – 126	Network.Host.Host.Host	1111 1111 0000 0000 0000 0000 0000 0000 or 255.0.0.0
B	128 – 191	Network.Network.Host.Host	1111 1111 1111 1111 0000 0000 0000 0000 or 255.255.0.0
C	**192 – 223**	**Network.Network.Network.Host**	**1111 1111 1111 1111 1111 1111 0000 0000 or 255.255.255.0**
D	*Defined for multicast operation and not used for normal operation*		
E	*Defined for experimental use and not used for normal operation*		

Figure 1.6 IP address chart by class.

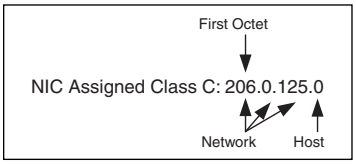

Figure 1.7 IP address example with four octets.

since the 1970s, and they include the ranges shown in Figure 1.6; an example of an IP address is shown in Figure 1.7.

The first octet (206) indicates a Class C (Internet-assigned) IP address range with the format *Network.Network.Network.Host* with a standard mask binary indicating 255.255.255.0. This means that we have 8 bits in the last octet for hosts. The 8 bits that make up the last, or fourth, octet are understood by infrastructure equipment such as routers and software in the following manner:

Bit:	1	2	3	4	5	6	7	8	
Value:	128	64	32	16	8	4	2	1	= 255 (254 usable hosts)

In this example of a full Class C, we only have 254 usable IP addresses for hosts; 0 and 255 cannot be used as host addresses because the network number is 0 and the broadcast address is 255.

With the abundant utilization of Class B address space and the flooding of requested Class C addresses, a Classless Interdomain Routing (CIR) system was introduced in the early 1990s. Basically, a route is no longer an IP address; a route is now an IP address and mask, allowing us to break a network into *subnets* and *supernets*. This also drastically reduces the size of Internet routing tables.

It is important to understand IP address masking and subnetting for performing a security analysis, penetration hacking, and spoofing. There's more information on these topics later in this chapter.

Subnetting, VLSM, and Unraveling IP the Easy Way

Subnetting is the process of dividing an assigned or derived address class into smaller, individual, but related, physical networks. Variable-length subnet masking (VLSM) is the broadcasting of subnet information through routing protocols (covered in the next chapter). A subnet mask is a 32-bit number that determines the network split of IP addresses on the bit level.

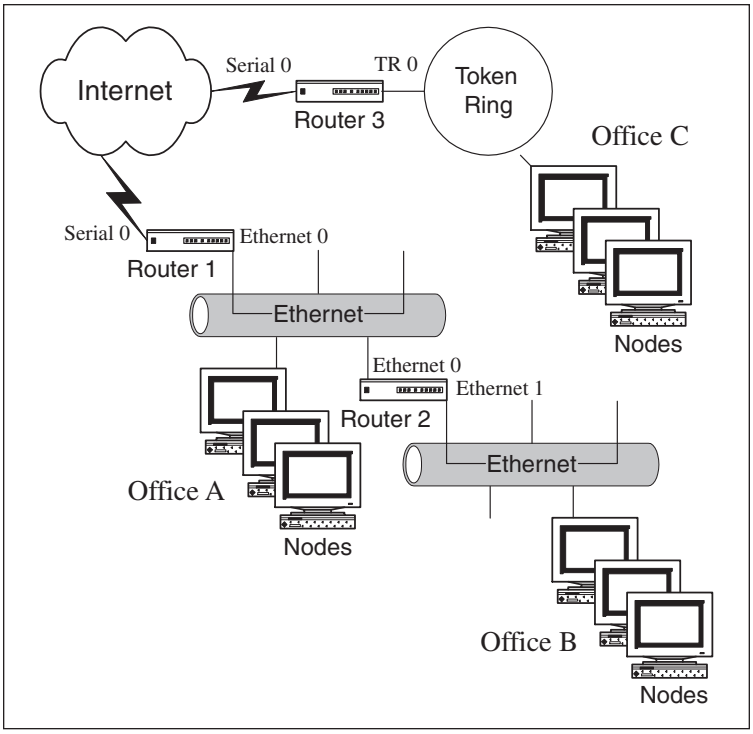

Figure 1.8 Real-world IP network example.

Example 1

Let's take a look at a real-world scenario of allocating IP addresses for a routed network (Figure 1.8).

Given: 206.0.125.0 (NIC assigned Class C). In this scenario, we need to divide our Class C address block to accommodate three usable subnets (for offices A, B, and C) and two subnets for future growth. Each subnet or network must have at least 25 available node addresses. This process can be divided into five steps.

Step 1

Four host addresses will be required for each of the office's router interfaces: Router 1 Ethernet 0, Router 2 Ethernet 0/Ethernet 1, and Router 3 Token Ring 0 (see Figure 1.9).

Step 2

Only one option will support our scenario of five subnets with at least 25 IP addresses per network (as shown in the Class C subnet chart in Figure 1.10).

Subnet	Interfaces
1	Router 1 - Ethernet 0 Router 1 - Serial 0 *(IP address will be provided by the Internet provider)* Router 2 - Ethernet 0
2	Router 2 - Ethernet 1
3	Router 3 - Token Ring 0 Router 3 – Serial 0 *(IP address will be provided by the Internet provider)*

Figure 1.9 Real-world network example interface requirement chart.

See Appendix A: "IP Reference Table and Subnetting Charts," as well as an IP Subnetting Calculator found on the CD for quick calculations. It is important to understand this process when searching for all possible hosts on a network during a discovery analysis.

Bits in Subnet Mask	Subnet Mask	# of Subnets	# of Hosts Per Subnet
2	255.255.255.192	2	62
3	**255.255.255.224**	**6**	**30**
4	255.255.255.240	14	14
5	255.255.255.248	30	6
6	255.255.255.252	62	2

Figure 1.10 Class C subnet chart by number of subnets versus number of hosts per subnet.