Hack Attacks Denied

A Complete Guide to Network Lockdown

Hack Attacks Denied

A Complete Guide to Network Lockdown

John Chirillo

Wiley Computer Publishing

John Wiley & Sons, Inc.

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

Disclaimer:
This eBook does not include the ancillary media that was packaged with the original printed version of the book.

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

Designations used by companies to distinguish their products are often claimed as trademarks. In all instances where John Wiley & Sons, Inc., is aware of a claim, the product names appear in initial capital or ALL CAPITAL LETTERS. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

Published by John Wiley & Sons, Inc.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold with the understanding that the publisher is not engaged in professional services. If professional advice or other expert assistance is required, the services of a competent professional person should be sought.

This title is also available in print as ISBN 0-471-41625-8

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

Contents

Acknowledgments

Foremost I would like to thank my wife for not only proofing this book, but for her continued support and patience during its development. Next in line would be my family and friends for their encouragement and confidence. Following in the wake, I find myself grateful to Neil Ramsbottom, Mike G., Mike Down, Shadowlord, Mindgame, John Fenton, Philip Beam, J.L. du Preez, Buck Naked, SteRoiD, no()ne, National Institute of Standards Technology and Marianne Swanson, 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.

As always, in order to be successful, one must surround oneself with the finest people. With that in mind, I must 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.

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

An increasing number of users on private networks are demanding access to Internet services such as the World Wide Web, email, telnet and File Transfer Protocol (FTP). Corporations want to offer Internet home pages and FTP servers for public access via the Internet. As the online world continues to expand, so too do concerns about security. Network administrators and managers worry about exposing their their organizations' confidential and or proprietary data, as well as their networking infrastructures, to the growing number and variety of Internet hackers, crackers, cyberpunks, and phreaks. In short, online security has become one of the primary concerns when an organization develops a private network for introduction to the Internet. To provide the required level of protection, an organization needs more than just a robust security policy to prevent unauthorized access; its managers need a complete and thorough understanding of all the elements involved in erecting solid fortification against hack attacks. And even those organizations not connected to the Internet need to establish internal security measures if they are to successfully manage user access to their networks, and protect sensitive or confidential information.

Hack Attacks Denied: A Complete Guide to Network Lockdown addresses all those concerns, and defines the procedures required to successfully protect networks and systems against security threats. By introducing a phased approach, which correlates to my previous book, *Hack Attacks Revealed*, this volume outlines the security steps to take to formulate and implement an effective security policy.

To begin, readers are made aware of security dangers, by introducing secret tiger team routines, complete with examples and illustrations. The book is divided into four logical phases. Phase 1 covers system infrastructure engineering, explaining the processes essential to protect vulnerable ports and services. Phase 2 details how to protect against the secret vulnerability penetrations itemized in *Hack Attacks Revealed*. Phase 3 introduces the necessary hack attack countermeasures to use on popular gateways, routers, Internet server daemons, operating systems, proxies, and firewalls. Phase 4 puts these security measures into perspective by compiling an effective security policy.

Who Should Read This Book

Hack Attacks Denied will enlighten anyone and everyone interested in or concerned about online security today, and lead to an understanding of how to best make their systems and networks as safe as they need to be.

More specifically, however, *Hack Attacks Denied* was written for these audiences:

- 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

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.

PHASE One

Securing Ports and Services

Hack Attacks Revealed, the predecessor to this book, defined and described computer ports and their services, and explained what makes certain of them so vulnerable. For those who did not read that book, and as a general reminder, computer ports are essentially doorways through which information comes into and goes out from a computer. Hackers use tools such as port scanners (also described in *Hack Attacks Revealed*) to search these ports, to find those that are open, or "listening," hence, vulnerable to penetration.

For all practical purposes, of the 65,000 or so ports on a computer, the first 1,024 are referred to and regarded as the *well-known ports*. The rest can be described as *concealed ports*. The purpose of Phase 1 is to introduce the techniques used to secure these ports and services. First we explore methods to protect well-known ports and to fortify those concealed ports. From there, we delve into discovery and scanning countermeasures. Discovery, as explained in *Hack Attacks Revealed*, is the initial "footprinting" or information gathering that attackers undertake to facilitate a plan that leads to a successful hack attack. Target port scanning is typically the second step in this discovery process.

This book is designed to to form a solid security foundation. To that end, and in keeping with the Tiger Team approach described in the first book, the phases of this book are divided into what I call "Tiger Team procedures" series of steps (phases), presented in an order that makes the most sense for successful fortification against security breaches.

CHAPTER 1

Common Ports and Services

The purpose of this chapter is to introduce the techniques used to secure the most vulnerable ports from the list of well-known ports, which includes TCP and UDP services. When two systems communicate, TCP and UDP ports become the ends of the logical connections that mandate these service "conversations." These ends specify the port used by a particular service daemon process as its contact port, that is, the "well-known port." A TCP connection is initialized through a three-way handshake, whose purpose is to synchronize the sequence and acknowledgment numbers of both sides of the connection (commonly referred to as connection-oriented or reliable service). UDP, on the other hand, provides a connectionless datagram service that offers unreliable, best-effort delivery of data.

In this chapter, we'll focus on the ports defined in *Hack Attacks Revealed* as those most vulnerable. These include Port 7: echo, Port 11: systat, Port 15: netstat, Port 19: chargen, Port 21: FTP, Port 23: telnet, Port 25: SMTP, Port 53: domain, Port 67: bootp, Port 69: TFTP, Port 79: finger, Port 80: http, Port 109: pop2, Port 110: pop3, Port 111: portmap, Port 135: loc-serv, Port 137: nbname, Port 138: nbdatagram, Port 139: nbsession, Port 161: SNMP, Port 512: exec, Port 513: login, Port 514: shell, Port 514: syslog, Port 517: talk, Port 518: ntalk, Port 520: route, and Port 540: uucp.

Securing Well-Known Ports

Keep in mind that the well-known ports are defined as the first 1,024 ports that are reserved for system services. Hence, outgoing connections will usually have port numbers higher than 1023. This means that all incoming packets that communicate via ports higher than 1023 are replies to connections initiated by internal requests. These incoming connections communicate via well-known ports that are listening to particular services. System processes or *service daemons* control these "services." However, while these services are listening for legitimate incoming connection requests, they are also open to malicious exploitation. With that in mind, let's look at methods used to "lock down" these well-known ports and to consecutively secure their services.

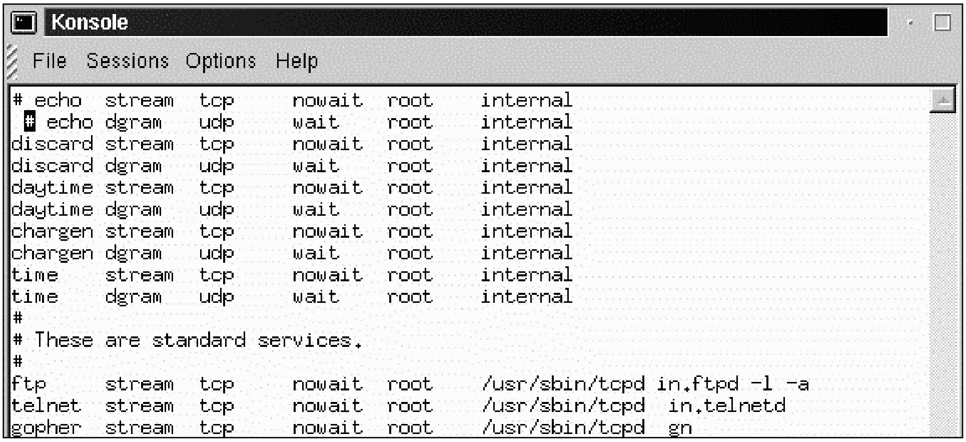
Before we delve into the specific ports, a brief explanation of the Windows Registry and the UNIX Internet Servers Database (inetd) daemon is in order. Inetd is actually a *daemon control process* that handles network services operating on a UNIX System. Using file /etc/inetd.conf for configuration, this daemon controls service activation, including ftp, telnet, login, and many more. Though this book refers to the inetd.conf file as it is implemented on the Linux system in directory /etc/, it is important to be aware that each flavor of UNIX may have a different location for this file; for example, AIX uses directory /usr/sbin, Digital uses /usr/sbin, HP-UX 9 and 10 use /etc and /usr/lbin, respectively, IRIX uses /usr/etc, Solaris uses /usr/sbin, and SunOS uses /usr/etc.

In Windows systems, the system Registry is somewhat comparable to the UNIX inetd daemon as a hierarchical database where all the system settings are stored. It has replaced all of the .ini files that controlled Windows 3.x. All system configuration information from *system.ini*, *win.ini*, and *control.ini* are all contained within the Registry. All Windows programs store their initialization and configuration data there as well.

Tiger Note **Remember to always make a backup of the *inetd.conf* file and the Windows Registry before making any adjustments.**

Port 7: Echo

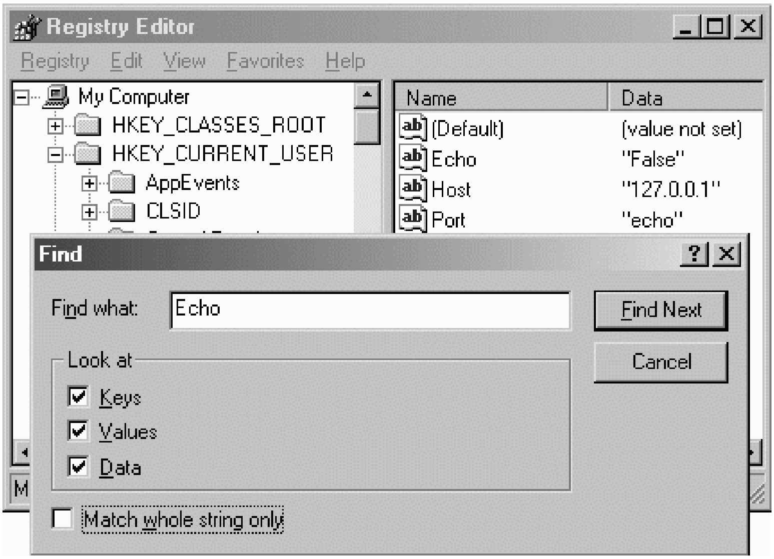
Standard communication policies may not necessitate the echo service, as it simply allows replies to data sent from TCP or UDP connection requests. In this case, it is advisable to disable this service to avoid potential denial-of-service (DoS) attacks. Before attempting to disable this service, however, you should check to see if any proprietary software—for example, system-monitoring suites or custom troubleshooting packages—requires it.



```
# echo  stream  tcp     nowait  root    internal
 # echo dgram   udp     wait    root    internal
discard stream  tcp     nowait  root    internal
discard dgram   udp     wait    root    internal
daytime stream  tcp     nowait  root    internal
daytime dgram   udp     wait    root    internal
chargen stream  tcp     nowait  root    internal
chargen dgram   udp     wait    root    internal
time    stream  tcp     nowait  root    internal
time    dgram   udp     wait    root    internal
#
# These are standard services.
#
ftp     stream  tcp     nowait  root    /usr/sbin/tcpd in.ftpd -l -a
telnet  stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
gopher  stream  tcp     nowait  root    /usr/sbin/tcpd  gn
```

Figure 1.1 Disabling services on UNIX systems.

- To disable the echo service in UNIX, simply edit the /etc/inetd.conf file and comment out the echo entry, as illustrated in Figure 1.1. At that point, restart the entire system or just the inetd process.
- To render the echo service inoperative in Windows systems, you must edit the system Registry by running regedit.exe from the Start/Run command prompt. From there, search for TCP/UDP Echo entries, and change their values to "false," or zero (see Figure 1.2). Upon completion, reboot the system and verify your modifications.



Name	Data
(Default)	(value not set)
Echo	"False"
Host	"127.0.0.1"
Port	"echo"

Find what: Echo

Look at
☑ Keys
☑ Values
☑ Data

□ Match whole string only

Figure 1.2 Editing the Windows system Registry to disable services in Windows systems.

If you are unsure or uneasy with making modifications to the Windows system Registry, refer to Appendix A for details on custom security software. In this case, with TigerWatch, you can proactively monitor and lock down system ports and services without interfering with the Registry or manually disabling a service. Later, we'll review TigerWatch, among other programs, in illustrative detail.

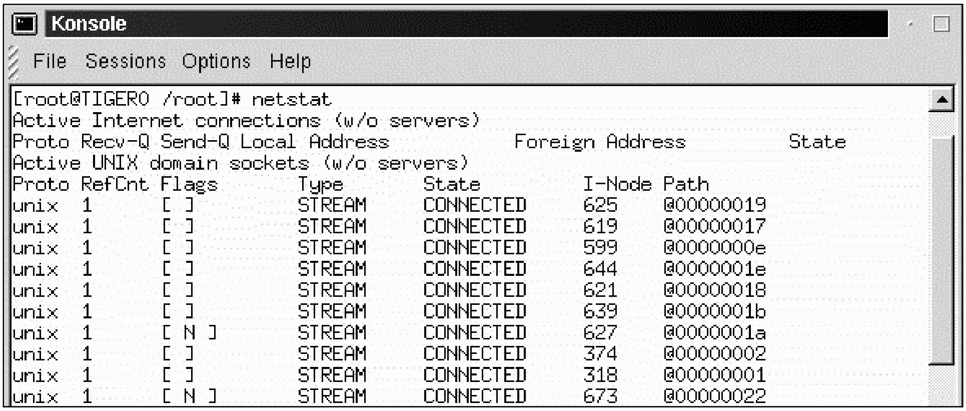
Port 11: Systat and Port 15: Netstat

By remote initiation, systat provides process status and user information, and therefore, should be disabled. To disable the systat service in UNIX, simply edit the /etc/inetd.conf, and comment out its entry for the echo service, as illustrated in Figure 1.1. At that point, restart the entire system or just the inetd process.

Not unlike systat, netstat can provide an attacker with active network connections and other useful information about the network's subsystem, such as protocols, addresses, connected sockets, and MTU sizes (refer to Figure 1.3). To disable the netstat service in UNIX, simply edit the /etc/inetd.conf file and comment out its entry, as shown in Figure 1.1 for the echo service. At that point, restart the entire system or just the inetd process.

Port 19: Chargen

The chargen service can be exploited to pass data to the echo service and back again, in an endless loop, causing severe system congestion. As a character stream generator, it is unlikely that standard communication policies would necessitate this service; therefore, it is advisable to disable this service to avoid attacks.



```
[root@TIGER0 /root]# netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node Path
unix  1      [ ]         STREAM     CONNECTED     625    @00000019
unix  1      [ ]         STREAM     CONNECTED     619    @00000017
unix  1      [ ]         STREAM     CONNECTED     599    @0000000e
unix  1      [ ]         STREAM     CONNECTED     644    @0000001e
unix  1      [ ]         STREAM     CONNECTED     621    @00000018
unix  1      [ ]         STREAM     CONNECTED     639    @0000001b
unix  1      [ N ]       STREAM     CONNECTED     627    @0000001a
unix  1      [ ]         STREAM     CONNECTED     374    @00000002
unix  1      [ ]         STREAM     CONNECTED     318    @00000001
unix  1      [ N ]       STREAM     CONNECTED     673    @00000022
```

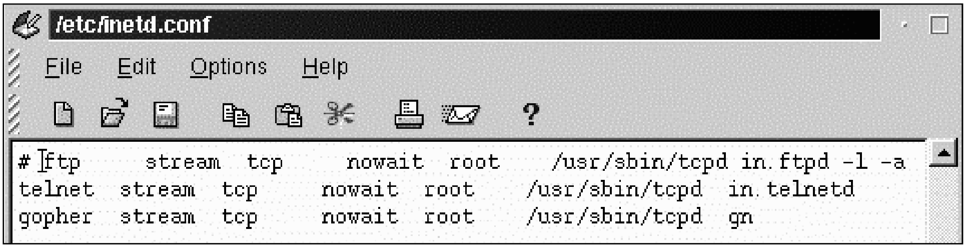
Figure 1.3 Some of the information revealed with Netstat.

- To disable the service in UNIX, simply edit the /etc/inetd.conf file, and comment out the chargen entry, as illustrated in Figure 1.1 for the echo service. At that point, restart the entire system or just the inetd process.
- Although the chargen service is not inherent to Windows, it may have been installed nonetheless. To render this service inoperative in Windows systems, you must edit the system Registry by running regedit.exe from the Start/Run command prompt. From there, search for chargen entries, and change their values to “false,” or zero (see Figure 1.2 for the same procedures performed for the echo service). Upon completion, reboot the system and verify your modifications.

Port 21: FTP

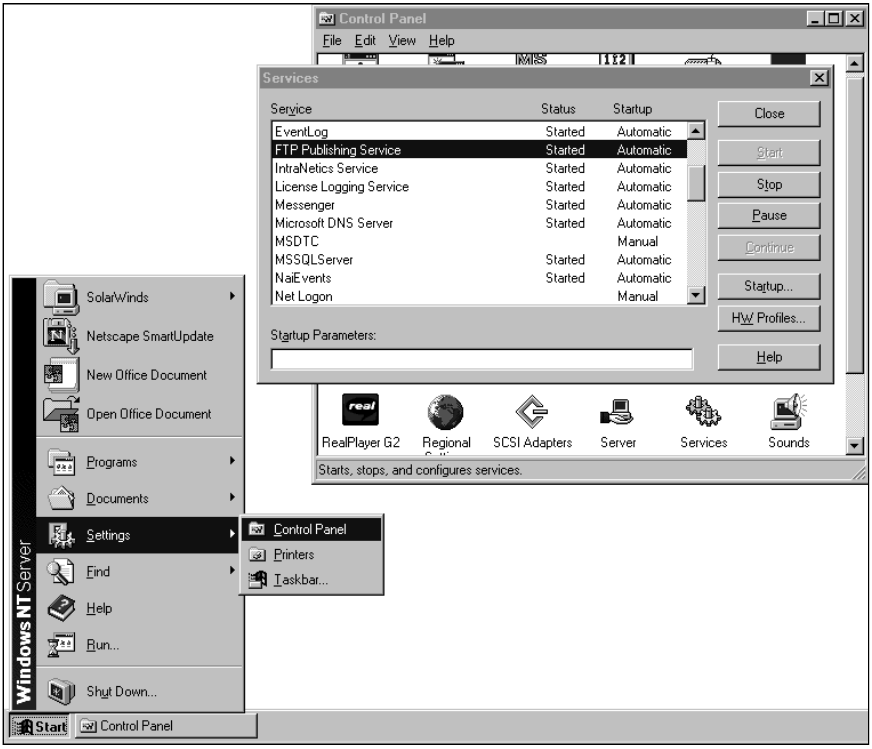
Unless your standard communication policies require the file transfer protocol (FTP), it is advisable to disable it. However, if FTP is a necessity, there are ways to secure it. For that reason, we’ll examine these scenarios, including lockdown explanations. Let’s begin with rendering FTP inoperative, obviously the most secure state.

- As with most of the vulnerable services in UNIX, commenting out the FTP service in the /etc/inetd.conf file should disable the daemon altogether (see Figure 1.4). To finalize the modification, don’t forget to stop and restart the inetd daemon—or, better yet, reboot the entire operating system.
- In Windows systems, there are two basic techniques for disabling FTP: modifying the startup configuration, and terminating the active process for Windows NT and 9x/2K, respectively. Modifying the startup configuration in Windows NT is as easy as it sounds, but you must be logged on with privileges to do so. From Start/Settings/Control Panel, double-click the Services icon, then scroll down to find the FTP Publishing Service, as illustrated in Figure 1.5.



```
# ftp     stream  tcp    nowait  root    /usr/sbin/tcpd in.ftpd -l -a
telnet  stream  tcp    nowait  root    /usr/sbin/tcpd  in.telnetd
gopher  stream  tcp    nowait  root    /usr/sbin/tcpd  gn
```

Figure 1.4 Disabling the FTP service under UNIX.

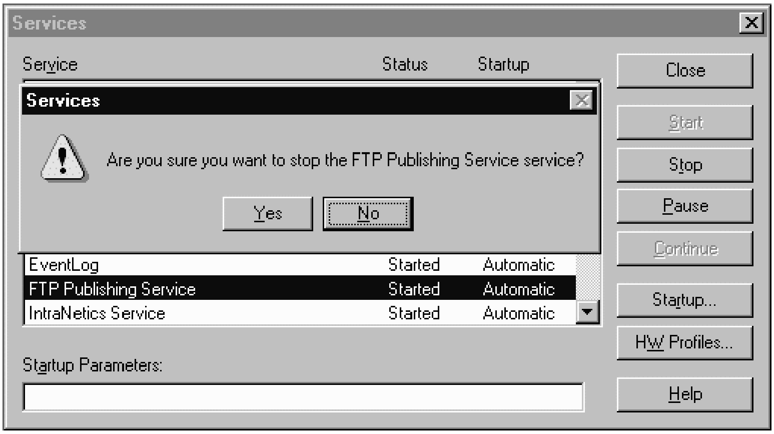


Service	Status	Startup
EventLog	Started	Automatic
FTP Publishing Service	Started	Automatic
IntraNetics Service	Started	Automatic
License Logging Service	Started	Automatic
Messenger	Started	Automatic
Microsoft DNS Server	Started	Automatic
MSDTC		Manual
MSSQLServer	Started	Automatic
NaiEvents	Started	Automatic
Net Logon		Manual

Starts, stops, and configures services.

Figure 1.5 Locating the FTP service daemon in Windows NT.

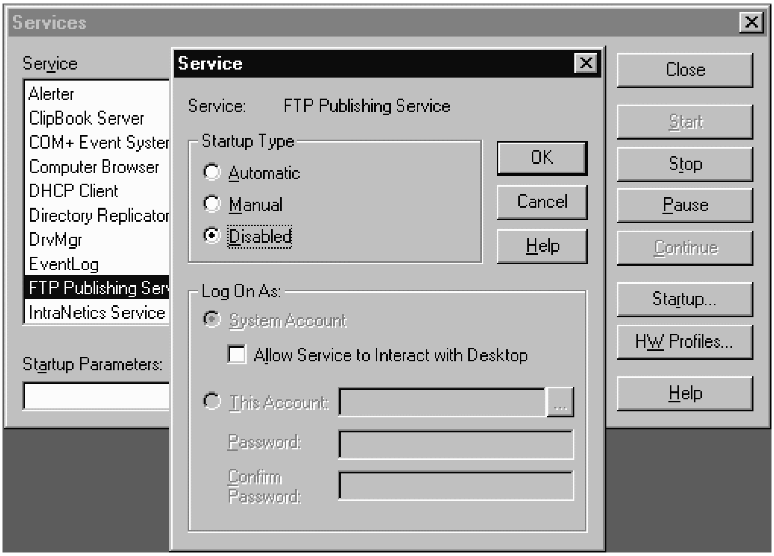
- At this point, highlight the FTP Publishing Service by pointing and clicking with the mouse; then click the Stop button option to the right of the services window (see Figure 1.6). After permitting Windows to stop the



Are you sure you want to stop the FTP Publishing Service service?

Service	Status	Startup
EventLog	Started	Automatic
FTP Publishing Service	Started	Automatic
IntraNetics Service	Started	Automatic

Figure 1.6 Manually disabling the FTP service daemon in Windows NT.



Service
Alerter
ClipBook Server
COM+ Event Syster
Computer Browser
DHCP Client
Directory Replicator
DrvMgr
EventLog
FTP Publishing Ser
IntraNetics Service

Startup Parameters:

Service: FTP Publishing Service

Startup Type
○ Automatic
○ Manual
● Disabled

Log On As:
● System Account
□ Allow Service to Interact with Desktop
○ This Account:
Password:
Confirm Password:

Figure 1.7 Permanently disabling the FTP service daemon in Windows NT.

service, the FTP daemon should remain inactive until the next reboot, depending on the next step. This step includes clicking the Startup button (to the right of the Services window), again with the FTP Publishing Service highlighted. In the new Startup Configuration window, select disabled and click OK to permanently disable the service (as shown in Figure 1.7).

Typically, on Windows 9x/2K systems, in order to permanently disable an FTP service daemon, you would do so from the service's proprietary administration module. An alternative is to permanently remove the service from the system via Start/Settings/Control Panel by selecting the Add/Remove Programs icon. However, if you are uncomfortable with these options, or prefer to temporarily disable the service, you can always press the Ctrl+Alt+Del keys together to pull up the Close Program Task Manager. At that point, simply scroll down, locate, and then highlight the FTP process. From there, depress the End Task button to terminate the FTP service until the system is restarted (Figure 1.8).

As previously mentioned, when disabling the FTP service is not an option, there are ways to secure it. Let's investigate some of these FTP exploit countermeasures:

FTP Banner Alteration. It is advisable to modify your FTP daemon banner, as it may potentially divulge discovery data to an attacker. The extent of this information varies from program to program, but may include daemon type, version, and residing platform. For example, take a look at Fig-

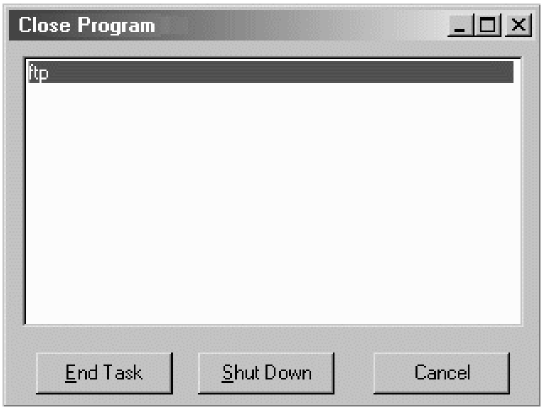
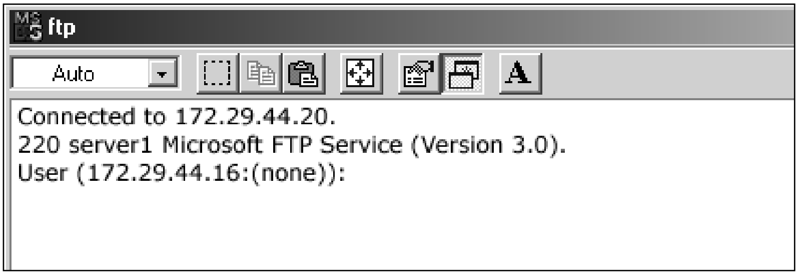


Figure 1.8 Terminating the FTP service daemon in Windows 9x/2K.

ure 1.9: some important discoveries have been made with this simple FTP request, such as the target system name, FTP daemon type, and version. For all practical purposes, all an attacker has to do now is search for known exploits for this version and then attack.

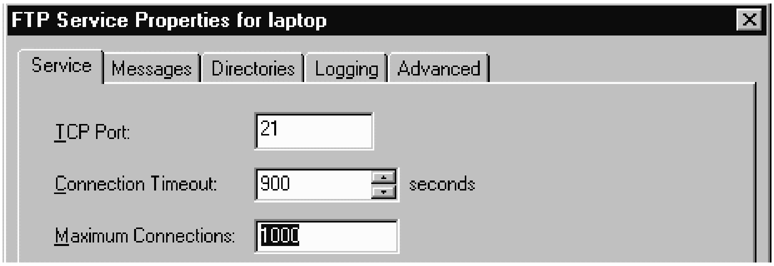
Tiger Note **Some packages may not permit banner alterations.**

FTP Connection Limitation. The FTP maximum connection limit poses an interesting threat. Many programs, by default, set this option to a high amount (see Figure 1.10). When modifying the connection limit, be realistic in your calculations. For example, consider how many connection streams the server really can handle. In this example, even 200 simultaneous sessions would bring my NT test server to its virtual knees. Some hackers like to do just that by spoofing multiple session requests.



```
Connected to 172.29.44.20.
220 server1 Microsoft FTP Service (Version 3.0).
User (172.29.44.16:(none)):
```

Figure 1.9 FTP banner discovery.



FTP Service Properties for laptop

TCP Port: 21

Connection Timeout: 900 seconds

Maximum Connections: 1000

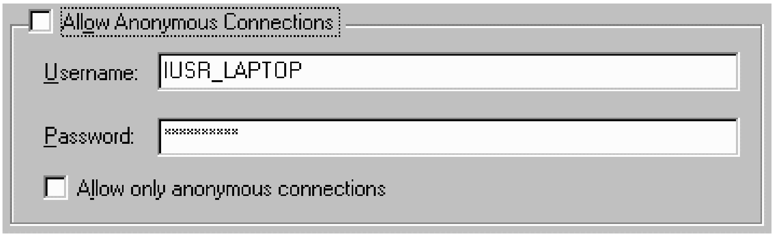
Figure 1.10 FTP connection limit on an NT server.

Anonymous Connection Status. It is important to avoid permitting anonymous FTP connections (Figure 1.11), unless your personal/business policy requires it. Also be aware that many FTP packages, especially UNIX, allow such connectivity by default. If you decide you have to sanction anonymous connections, be sure to strictly secure file and directory permissions. On UNIX platforms, be sure to strip down the FTP /etc/passwd file as well.

Permissions. It is crucial to modify file, directory, upload, and download FTP permissions, per user. Always check and double-check your settings for reliability. Depending on the number of users, this may take some time; but it is time well spent. Also, on UNIX platforms in particular, disable chmod options, along with directory browsing. On Windows systems, be cognizant of the potentially wily Guest account—in most cases, it should be disabled.

Tiger FTP

FTP software daemons usually come packaged with UNIX operating systems. However, home and/or private Windows users who seek FTP provisioning and who are partial to full control need not fret. Following is an FTP compilation that can be used at your discretion. With it, you can control the functionality to provide secure FTP access to friends and family members. Functions



☐ Allow Anonymous Connections

Username: IUSR_LAPTOP

Password: **********

☐ Allow only anonymous connections

Figure 1.11 Anonymous FTP connection status.

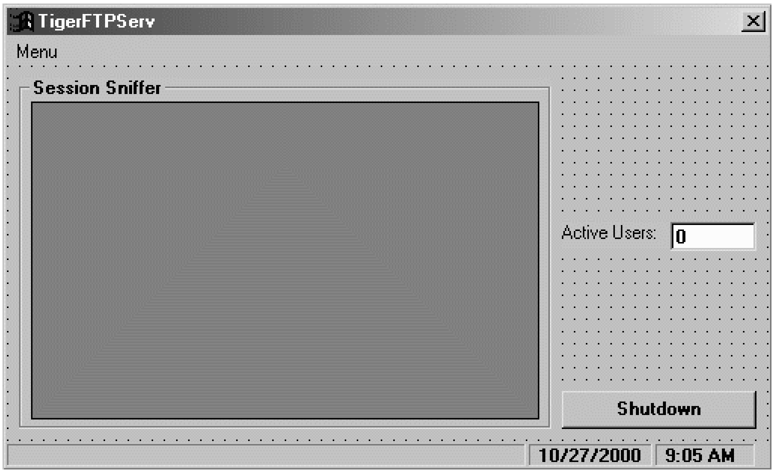


Figure 1.12 TigerFTPServ primary form and program interface.

include available command options, file and directory permissions, and session stream options. TigerFTPServ (see Figure 1.12) is yours to modify, distribute, and utilize in any fashion. The program also includes a session sniffer, whereby all connection requests and transaction status are displayed in real time. To avoid confusion and ensure security, all user permissions are controlled via TFTPServ.ini:

```
[Settings]
Version=1.0.0
[Users]
Users=1
Name1=test
Pass1=tester
DirCnt1=2
Home1=C:\
Access1_1=c:\,RWXLMS
Access1_3=d:\,RWXLMS
```

You can modify the main form, *FrmFTP.frm*, to control user connections and to customize the look and feel of the main program module.

FrmFTP.frm

```
Public MainApp As MainApp

Private Sub Form_Unload(Cancel As Integer)
  MainApp.Closing
  Set MainApp = Nothing
End Sub
```

```
Private Sub EndCmd_Click()
  Dim i As Integer
  For i = 1 To MAX_N_USERS
    If users(i).control_slot <> INVALID_SOCKET Then
      retf = closesocket(users(i).control_slot)
      Set users(i).Bash = Nothing
    End If
    If users(i).data_slot <> INVALID_SOCKET Then
      retf = closesocket(users(i).data_slot)
    End If
  Next
  retf = closesocket(ServerSlot)
  If SaveProfile(App.Path & "\tftpserv.ini", True) Then
  End If
  Unload Me
End Sub

Private Sub mEndCmd_Click()
Dim i As Integer
  For i = 1 To MAX_N_USERS
    If users(i).control_slot <> INVALID_SOCKET Then
      retf = closesocket(users(i).control_slot)
      Set users(i).Bash = Nothing
    End If
    If users(i).data_slot <> INVALID_SOCKET Then
      retf = closesocket(users(i).data_slot)
    End If
  Next
  retf = closesocket(ServerSlot)
  If SaveProfile(App.Path & "\tftpserv.ini", True) Then
  End If
  Unload Me
End Sub

Private Sub mSetup_Click()
  UserOpts.Show 1
End Sub
```

The form *AddEditDir.frm* is used to add listings to the available FTP directories for file downloading.

AddEditDir.frm

```
Option Explicit

Private Sub AddEditCnx_Click()
  UserOpts.Tag = ""
  Unload Me
End Sub

Private Sub AddEditDone_Click()
```

```
  UserOpts.Tag = DirPath.Text
  Unload Me
End Sub

Private Sub BrowseDir_Click()
  AddEditDir.Tag = DirPath.Text
  FindFolder.Show 1
  DirPath.Text = AddEditDir.Tag
End Sub
```

The next form, *FindFolder.frm*, is used as the user interface for searching available directories for downloadable files.

FindFolder.frm

```
Option Explicit
Dim DrvS(32) As String
Dim LastStr As String
Dim DrvC As Integer

Private Sub FldrDone_Click()
  Form_Terminate
End Sub

Private Sub FolderList_Click()
Dim s As String, t As String, s2 As String
Dim i As Integer
  i = FolderList.ListIndex + 1
  s2 = FolderList.Text
  If Mid(s2, 1, 1) = "[" Then
    s2 = Mid(s2, 2, 2) & "\"
    DirPath = s2
  Else
    If FolderList.Text = ".." Then
      s = Left(LastStr, Len(LastStr) - 1)
      Do Until Right(s, 1) = "\"
        s = Left(s, Len(s) - 1)
      Loop
      s2 = s
      DirPath = s2
    Else
      s2 = DirPath & FolderList.Text & "\"
      DirPath = s2
    End If
  End If
  LastStr = s2
  FolderList.Clear
  s = FindFile("*.*", s2)
  Add_Drives
End Sub
```

```
Private Sub Form_Load()
Dim s As String
  GetSystemDrives
  If AddEditDir.Tag <> "" Then
    LastStr = AddEditDir.Tag
    DirPath = LastStr
    s = FindFile("*.*", AddEditDir.Tag)
  End If
  Add_Drives
End Sub

Private Sub Add_Drives()
Dim x As Integer
  For x = 1 To DrvC
    FolderList.AddItem "[" & DrvS(x) & "]"
  Next
End Sub

Private Sub Form_Terminate()
  AddEditDir.Tag = DirPath.Text
  Unload Me
End Sub

Private Sub GetSystemDrives()
Dim rtn As Long
Dim d As Integer
Dim AllDrives As String
Dim CurrDrive As String
Dim tmp As String
  tmp = Space(64)
  rtn = GetLogicalDriveStrings(64, tmp)
  AllDrives = Trim(tmp)
  d = 0
  Do Until AllDrives = Chr$(0)
    d = d + 1
    CurrDrive = StripNulls(AllDrives)
    CurrDrive = Left(CurrDrive, 2)
    DrvS(d) = CurrDrive
    DrvC = d
  Loop
End Sub

Private Function StripNulls(startstr) As String
Dim pos As Integer
  pos = InStr(startstr, Chr$(0))
  If pos Then
    StripNulls = Mid(startstr, 1, pos - 1)
    startstr = Mid(startstr, pos + 1, Len(startstr))
    Exit Function
  End If
End Function
```

UserOpts.frm can be customized as the administrative module for adding, deleting, and setting user preferences.

UserOpts.frm

```
Option Explicit
Dim uItem As Integer
Dim aItem As Integer
Dim tStrng As String
Dim uUser As Integer
Dim Pcnt As Integer

Private Type Priv
  Path As String
  Accs As String

End Type
Private Privs(20) As Priv

Private Sub FDAdd_Click()
  tStrng = Get_Path("")
  If tStrng <> "" Then
    AccsList.AddItem (tStrng)
    Pcnt = Pcnt + 1
    UserIDs.No(uUser).Priv(Pcnt).Path = tStrng
    FDUpdate.Enabled = True
    FDRemove.Enabled = True
  End If
  AccsList_False
End Sub

Private Sub FDEdit_Click()
  tStrng = Get_Path(AccsList.Text)
  If tStrng <> "" Then
    AccsList.List(aItem) = tStrng
    UserIDs.No(uUser).Priv(aItem + 1).Path = tStrng
  End If
  AccsList_False
End Sub

Private Sub FDRemove_Click()
Dim z As Integer
  For z = (aItem + 1) To UserIDs.No(uUser).Pcnt
    UserIDs.No(uUser).Priv(z).Path = UserIDs.No(uUser).Priv(z + 1).Path
    UserIDs.No(uUser).Priv(z).Accs = UserIDs.No(uUser).Priv(z + 1).Accs
  Next
  UserIDs.No(uUser).Pcnt = UserIDs.No(uUser).Pcnt - 1
  AccsList.RemoveItem (aItem)
  AccsList_False
End Sub
```