
Network Role-Based Security

by Barak Weichselbaum

A network role is computer software, hardware, or a device that serves one or more other users, hardware, and devices. For example, a network fax server is a network role, because it is accessed from the network and serves many people as opposed to a regular analog fax that may serve multiple users but it's not accessible from the network.

This chapter covers various network roles, such as e-mail servers, DNS servers, web servers, and more. Each network role uses one or more protocols. For some protocols it's important to go over their inner workings in order to better understand their security problems; for others, it may be unimportant for the discussion, it may be very complex, or it may fill a book of its own and therefore be out of scope. An important thing to remember is that not all network roles are Internet related, but they do carry security risks that need to be considered.

NOTE *Throughout this chapter, a network is assumed to be a TCP/IP network only.*

In this chapter we may refer to some other network terminology or applications, such as the following:

- Virtual private network (VPN)
- TCP spoofing
- Intrusion-detection system (IDS)
- Intrusion-prevention system (IPS)
- Secure Sockets Layer (SSL)
- Demilitarized zone (DMZ)
- Firewall
- Network gateway

RFC Clarification

This chapter refers to one or many RFCs. A Request For Comment (RFC) is a series of documents that discusses many aspects of the Internet: communication, protocols, procedures, programs, and concepts. (The official homepage for RFCs is <http://www.ietf.org>.) The RFCs are linked in this chapter to point the reader to further resources. If you feel you want to read how the protocol works on a very technical level (far deeper than this chapter goes), the RFC is your next reading stop; however, the RFCs are not mandatory for understanding this chapter.

- TCP port
- Digital signature
- Web application security
- MD5

These items are defined in other chapters and, therefore, are not explained in this chapter.

E-Mail

Not every corporate user is computer savvy. About a year ago, a client called one of the authors telling him his computer didn't work anymore. A few seconds after asking the client a few questions to try to pinpoint what went wrong, it all became clear in his response: "I opened an e-mail attachment, and my antivirus program asked me something. Without really reading it, I clicked Yes." (The client is a licensed acupuncturist and not a computer specialist, as you may have guessed.) "You opened a virus," the author said. "Why did you do that?" Of course, his answer didn't prevent this author from reinstalling the client's computer and charging him for the labor.

An end user may consider e-mail as a means to an end (for example, he uses it to communicate with other Internet users); however, as IT pros, we know it's not that simple—e-mails harbor spam, viruses, hackers, eavesdroppers, and more. One single misconfiguration can spell bankruptcy for the firm we work for (or own). This chapter will cover the following issues in detail:

- E-mail protocols
- E-mail distribution
- Spam and spam control
- Virus and virus control

Protocols and Security Issues

We use the word *e-mail* freely without giving it much thought, if any: "Please e-mail me the documents," "I've received the new contract by e-mail." But how does e-mail actually work? Another common issue with e-mail is the illusion that it is nonrefutable: "I've received an e-mail from John; therefore, I'm sure it came from John" or "John, I sent you that contract yesterday—what do you mean you don't have it?"

Users that receive e-mails from a colleague or a friend (whom they trust) will most likely open the attachment. One of those attachments may be a virus that will be unleashed into the computer, and maybe the corporate network, devastating it and leaving us with the task of salvaging what we can—if we can. This chapter explains the inner workings of e-mail, its protocols, its flaws, and security issues such as these:

- Protocols such as SMTP, POP3, and IMAP4
- Server configuration
- Server vulnerabilities

SMTP

E-mail is a two-part system—one part for sending and one for receiving. The Simple Mail Transfer Protocol (SMTP), defined in RFC 2821 ([<ftp://ftp.rfc-editor.org/in-notes/rfc2821.txt>]) handles sending and relaying of e-mail. (SMTP can only send e-mail. To retrieve e-mail there are others protocols such as POP3 and IMAP4, which are discussed later in this chapter.) There are two types of SMTP protocols: regular SMTP and extended SMTP (also called ESMTP). ESMTP supports authentication, whereas regular SMTP does not. SMTP communicates via plaintext (you can use Telnet to send e-mail if you know the protocol “by heart”) and has a special assigned port of TCP 25.

SMTP is “request/response”-based, which means the client (the originator of the session) sends a command and the server (contacted by the client) replies with a three-digit numeric code followed by a descriptive message. (Each command has its own assigned response code that denotes success or failure.)

Manually Connecting to a SMTP Server

If you want to test yourself, or you just want to find out which software an SMTP server runs, here are all the SMTP commands we will discuss:

- Run `telnet.exe`
- Type `open mailserv 25`

In the preceding command, *mailserv* represents the DNS name, or IP, of an SMTP mail server. For example, *komodia.com* is the address of my SMTP server.

If you managed to connect to the mail server and it is, indeed, a mail server, you should see the SMTP welcome banner. This information is used by hackers to identify the mail server and choose the appropriate exploits. Let’s look at the following response:

```
220 odin.inter.net.il ESMTP Mirapoint 3.3.3-GR; Thu, 29 May 2003 22:51:17
+0300 (IDT)
```

As you can see, it’s easy to see the server product name (Mirapoint). With this knowledge, a hacker goes to his favorite online security or exploit portal (my favorite one is <http://neworder.box.sk>) and looks up this server, searching the appropriate exploit. (Of course, the administrator of this server may use a product that spoofs the welcome banner to display this message while running another vendor’s SMTP server.)

Now imagine you're the hacker and you telneted another SMTP server that replied like this:

```
220 DNS name -- Server ESMTP (MSG)
```

This banner gives us less information than the previous one. (Of course, we can search the Internet for what server gives this message, but it's not as straightforward as the previous banner.)

SMTP Character Limitation Although SMTP uses plaintext, the protocol allows only ASCII characters, and ASCII characters are 7-bit). This "feature," therefore, requires an attachment that uses 8-bit binary data to be converted into 7-bit representation. (There are many encoding/decoding methods, such as BASE64, UUENCODE/UUDECODE, and BinHex, but they're beyond the scope of this book.)

NOTE *Converting regular text messages, although they don't require conversion, may sometimes help spammers evade spam filters. Some spam filters don't decode the converted data and therefore can't detect spam key words, while others do. For example, suppose you have a message that says "Generic Viagra for \$2.50" and you encode it using BASE64, which yields "R2VuZXJpYyBWaWFncmEgZm9yIDluNSQ=".* Unless the spam filter decodes it, a regular key word search will yield nothing! More discussion on spam can be found later in this chapter in the section "Spam and Spam Control."

SMTP Command Sequence

An SMTP session begins after the initial connection between the client and server. After the connection is established, the SMTP server sends its command code and identifying message, which usually looks like this:

```
220 Server name ESMTP Mirapoint 3.2.2-GA; Mon, 12 May 2003 00:54:59 +03
```

The 220 response code denotes the server is ready to work with us. (As you can see, the server exposes information about the software its running—in this case, "Mirapoint" SMTP server.) We will go over the common commands (first in a table, and then elaborating one by one), their meaning, and codes. (We will use uppercase spelling for the commands in order to distinguish them from the lowercase parameters; however, in practice they are case insensitive.)

See Figure 16-1 and Tables 16-1 and 16-2 for a complete SMTP session captured by Ethereal.

```
220 binkey.iticom.net ESMTP
EHLO komodia
250-binkey.iticom.net
250-PIPELINING
250 8BITMIME
RSET
250 flushed
MAIL FROM:<barak@komodia.com>
250 ok
RCPT TO:<barak@komodia.com>
250 ok
DATA
354 go ahead
Message-ID: <200305180159420654.21136680@komodia.com>
X-Mailer: Calypso Version 3.30.00.00 (4)
Date: sun, 18 May 2003 01:59:42 +0200
Reply-To: barak@komodia.com
From: "Barak weichselbaum" <barak@komodia.com>
To: barak@komodia.com
Subject: Demo of SMTP
Mime-Version: 1.0
Content-Type: multipart/alternative; boundary="=====_105321598217448=_"

-----_105321598217448=_
Content-Type: text/plain; charset="us-ascii"

Demo of SMTP

-----_105321598217448=_
Content-Type: text/html; charset="us-ascii"

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD>
<META http-equiv=Content-Type content="text/html; charset=iso-8859-1">
<META content="MSHTML 6.00.2800.1170" name=GENERATOR></HEAD>
<BODY style="FONT-FAMILY: Arial" text=#000000 bgColor=#ffffff><FONT size=2>Demo
of SMTP</FONT></BODY></HTML>

-----_105321598217448=--
.
250 ok 1053212462 qp 18507
QUIT
221 binkey.iticom.net
```

FIGURE 16-1 Screen shot of a complete SMTP session

NOTE *Ethereal is a very popular open source sniffer that works both on *nix and Windows machines (see <http://www.ethereal.com>).*

Command	Description	Success Code	Common Failure Codes
HELO <i>domain</i>	Sent with the client's ID to start the session	250	553, 554
MAIL FROM: < <i>e-mail</i> >	Sent to set the current e-mail's sender	250	
RCPT TO: < <i>e-mail</i> >	Sent to add a recipient to the current e-mail (can be used multiple times per message)	250	251, 450, 550, 551
DATA	Sent to indicate the client is ready to send the e-mail's data	354	
QUIT	Sent to the server telling it to gracefully disconnect the connection	221	

TABLE 16-1 Common SMTP Commands**TABLE 16-2**
Common SMTP
Reply Codes

Code	Meaning
221	Service is closing transmission channel. (SMTP closing.)
250	Action completed OK.
251	User not local. (User is part of another domain.)
354	Server is ready to receive data.
450	Mailbox is unavailable.
550	Mailbox is unavailable.
551	User not local. (User is part of another domain.)
553	Requested action not taken. (Usually spammers receive this error but only if an SMTP server is configured to fight spam. This will be elaborated later in the section, "How ISPs Fight Spam.")
554	Transaction failed. (Again usually for spammers. See parenthetical note for code 553.)

HELO HELO is the command that opens the SMTP session. It is used to send the client's identifying name to the mail server (the name has no special meaning, and it's usually the machine name; however, it can be used to identify spam applications, which will be discussed later in this chapter in the section "Spam and Spam Control".)

Example

```
HELO demo
```

Common Response

```
250 Server name Hello User DNS name [User's IP], pleased to meet you
```

MAIL FROM MAIL FROM is the command sent to set the current e-mail sender's address. Some mail servers check the source address to see if it has a valid DNS entry in order to reject fake source addresses or to compare it with spam lists. These methods are used to block spammers from using the SMTP server services. A problem with this configuration is that for each e-mail received the mail server has to check the source address and "waste" network resources.

Example

```
MAIL FROM: <barak@komodia.com>
```

Common Response

```
250 barak@komodia.com... Sender ok
```

RCPT TO RCPT TO is the command sent to add one more recipient to the current e-mail. RCPT TO can be invoked multiple times for multiple recipients. The e-mail's "To" field doesn't have to match the RCPT TO that was sent. That is, the client sent three RCPT TO commands (three recipients) but indicated only two e-mail addresses in the "To" field, which means that the two recipients won't know the e-mail was also sent to a third person. This is how "BCC" is implemented.

A problem that arises with the RCPT TO command is that you don't know if your e-mail header shows an accurate list of the recipients. It may state that this e-mail was also sent to other recipients, but in reality it was sent to more or different recipients than the ones listed on the e-mail.

NOTE Don't trust the mailing list information you see in the e-mail header; it can be faked.

Example

```
RCPT TO: <barak@komodia.com>
```

Common Response

```
250 barak@komodia.com... Recipient ok
```

DATA DATA is the command that tells the server you are ready to send the e-mail data. The client indicates it finished transmitting by sending a line containing only a period. An ASCII e-mail's data is structured according to RFC 822 and Multipurpose Internet Mail Extension (MIME), which allows you to put attachments inside messages structured according to RFCs 2045, 2046, 2047, 2048, and 2049 (see [ftp://ftp.rfc-editor.org/in-notes/rfc2045.txt, ftp://ftp.rfc-editor.org/in-notes/rfc2046.txt, ftp://ftp.rfc-editor.org/in-notes/rfc2047.txt, ftp://ftp.rfc-editor.org/in-notes/rfc2048.txt, and ftp://ftp.rfc-editor.org/in-notes/rfc2049.txt]).

Example

```
Return-Path: barak@komodia.com
Received: (qmail 31172 invoked by uid 0); 26 Apr 2003 17:53:50 -0000
Received: from unknown (HELO Server name) (Server's IP) by www.komodiam.com
with SMTP; 26 Apr 2003 17:53:50 -0000
Received: from Barak (My DNS name [My IP]) by my ISP's mail server
(Mirapoint Messaging Server MOS 3.2.2-GA) with ESMTP id AXT20233; Sat, 26
Apr 2003 20:53:45 +0300 (IDT)
Message-Id: <Some ID>
From: barak@komodia.com
To: test@komodia.com
Subject: demo
Date: Saturday, April 26, 2003 20:52:48 +0200
Content-Type: text/plain
```

This is a message to show our readers.

OK, now let's decipher this mess and inspect it field by field! (Of course, some e-mails will have more or fewer fields; not all fields are mandatory.) Table 16-3 lists the fields, what they mean, and where they come from.

The list in Table 16-3 can be quite helpful when you receive an e-mail and you want to take a quick peek at its header in order to look for forgeries or at the original IP of the sender. (The most important field is the Received field, which states the e-mail server and IP that sent the e-mail. If you see that a spam came from, lets say, China, it will make more sense to delete it, as you probably won't have much luck if you try to complain about it.) Sometimes spammers add fields that include bogus IPs to make it harder to track the original IP. For example, suppose a hacker adds three Received fields containing three different IPs, and the e-mail contains the original IPs, as well, which sums up to four IPs. Which IP is real? Always use the first appearing IP because it can't be forged.

The origination of the spam matters because in some countries the authorities just don't care about spam—countries such as China, Russia, and other third-world countries. How do we know which IP belongs to which country? The site <http://ip-to-country.com/> offers a database listing IPs for different countries and tools to assist in pinpointing an IP to its source country. Analyzing the headers of spam e-mails sometimes can be difficult and tricky. The site <http://www.stentorian.com/antispam/> is a very good guide to how to combat spam and analyze spam headers.

Field	Meaning	What Puts It There?
Return-Path	The reply address for error messages	Mail server
Received	The date and time the mail server received the message	Mail server
Received (second field)	Which server sent the message	Mail server
Received (third field)	Who sent the message	Mail server
Message-ID	The ID the server assigned to the message	Mail server
From	The Sender of the message	Mail client
To	The message recipients (Note: The CC field can include message recipients as well.)	Mail client
Subject	The subject of the message	Mail client
Date	The date and time this message was sent	Mail client
Content-Type	The type of message data	Mail client

TABLE 16-3 Common E-Mail Header Fields

NOTE A common misconception we encounter (or are asked about) quite often is e-mail IP spoofing. It's not possible to spoof the IP of the e-mail because of the nature of TCP sessions; the IP you see is the actual IP of the computer that initiated the session to the mail server. Of course, the IP you see could also be a proxy and not the actual machine; however, the IP is legitimate and can't be spoofed.

EXPN and VRFY The `EXPN` command is used to view the content of a mailing list, and the `VRFY` command is used to verify that a user exists (`VRFY`). The mail server should be configured to ignore these commands because they can be used to gather information about users on the server.

Extended SMTP (ESMTP)

The lack of SMTP authentication forced the industry to find new solutions to allow users to authenticate with a mail server. SMTP authentication is particularly useful for employees who need to use their corporate e-mail server while connected outside the organization. (For example, employees working at a client's office, employees working from home without a VPN, or roaming users that connect to different ISPs while traveling.) The server can be configured to require authentication from every IP outside the IP of the organization in order to prevent non-employees from using the corporate resources.

ESMTP, defined in RFC 1869 ([ftp://ftp.rfc-editor.org/in-notes/rfc1869.txt]), uses the same commands as regular SMTP with a few exceptions:

- The session begins when the client sends the EHLO command (rather than the HELO command, as with SMTP).
- After the server's response, the client may authenticate with the mail server, but this is not mandatory.

ESMTP Authentication Types

ESMTP offers a number of authentication methods. A mail server doesn't have to support all of them; when a session begins, the server notifies the client as to which authentication methods it supports. As you can see in Figure 16-2, in lines 11 and 12 the server announces it uses AUTH LOGIN PLAIN and AUTH LOGIN.

AUTH LOGIN PLAIN This method is the simplest authentication method. It requires the client to send its username and password in plaintext. As you can see, this method is quite straightforward; however, it has a major drawback—the username and password can be sniffed while someone is working on the corporate LAN, or on a wireless connection, by governments that monitor the Internet or hackers that compromise a computer along the way.

AUTH LOGIN Unlike the AUTH LOGIN PLAIN, this method sends the username and password using *BASE64 encoding*, which transfers 8-bit data into 7-bit ASCII characters. The only advantage over the previous method is the cryptic look of the username and password; however, most sniffers will decode it on the fly, and it is weak (because it can be sniffed) just like the previous authentication method.

CRAM-MD5 After requesting this authentication scheme, the mail server sends the client a challenge; and the client uses this challenge and its password to calculate a hash value to send to the server. (The client uses MD5, which is discussed in the encryption chapter.) Unlike the BASE64 and the plain logins, even if a hacker is sniffing the network and sees the hash value being sent, it has no way of knowing the original password. The exact algorithm is described in RFC 2095 ([ftp://ftp.rfc-editor.org/in-notes/rfc2095.txt]).

FIGURE 16-2
Capture of a
ESMTP session
using AUTH LOGIN
method

```
220 mxout4.netvision.net.il -- Server ESMTP <MSG>
250-mxout4.netvision.net.il
250-8BITMIME
250-PIPELINING
250-DSN
250-XDFLG
250-ENHANCEDSTATUSCODES
250-HELP
250-TURN
250-XLOOP BB1271AA9203BDE4915F02C688BAC3B9
250-AUTH LOGIN PLAIN
250-AUTH=LOGIN
250-ETRN
250-RELAY
250 SIZE 0
```

Is Internet Sniffing Fact or Myth?

“Internet sniffing” is a term we often hear. Internet users and even sometimes IT pros believe it’s possible to sniff the Internet traffic of a corporation or a private user quite easily, but that is really a fallacy. In order to sniff “targeted” traffic—meaning the traffic of a specific user or IP rather than random sniffing—the hacker needs access to the first or second router, which would be an ISP router. Although it’s possible to gain such access, it’s not easy to do.

Of course, the government can easily sniff “targeted” traffic by obtaining a warrant to do so; but aside from that, what can be easily sniffed? The easiest topology to sniff is wireless networks (discussed in Chapter 13). Another possibility is sniffing on a LAN. Some topologies (computers connected to a hub, for example) can be sniffed easily, while others (such as computers connected to a switch) are harder but still feasible to sniff. See Chapter 14 for more information on sniffing LAN traffic.

Other Authentication Types There are other authentication methods than those already listed; however, they are far less common, so we’ll mention them and point to sites where you can get more information about them:

- GSSAPI (RFC 2078 and RFC 2743—[<ftp://ftp.rfc-editor.org/in-notes/rfc2078.txt> and <ftp://ftp.rfc-editor.org/in-notes/rfc2743.txt>])
- Kerberos_V4 (RFC 1411—[<ftp://ftp.rfc-editor.org/in-notes/rfc1411.txt>])
- SCRAM-MD5 (<http://josefsson.org/cgi-bin/viewcvs.cgi/libgsasl/doc/specification/draft-newman-auth-scram-03.txt?rev=1.1&content-type=text/vnd.viewcvs-markup>)

Authentication Recommendation Plaintext authentication can be sniffed easily on open networks, such as LANs and wireless networks, and therefore is highly discouraged. The most preferred ways are to use either CRAM-MD5, which even if sniffed will not reveal your password, or SSL. The big difference between CRAM-MD5 and SSL is that SSL encrypts the entire session, whereas CRAM-MD5 protects the password but leaves the content wide open for eavesdroppers to snoop on.

POP3

Post Office Protocol 3 (POP3), RFC 1460, 1725, and 1939 ([<ftp://ftp.rfc-editor.org/in-notes/rfc1460.txt>, <ftp://ftp.rfc-editor.org/in-notes/rfc1725.txt>, and <ftp://ftp.rfc-editor.org/in-notes/rfc1939.txt>]) is the protocol used to retrieve e-mail from the mail server. (It uses TCP port 110.)

POP3 is built almost like SMTP, but POP3 is used only to retrieve e-mail. It uses plaintext to communicate (without the 7-bit limit), and it mimics the SMTP answer/reply mechanism as well. (See Figure 16-3 for a capture of a failed POP3 session.) To denote success, the POP3 server sends plus (+) at the beginning of the response, as opposed to a minus (–) to denote failure. (Unlike SMTP, which uses error codes, POP3 describes the error in the text following the – character.)

```

Contents of TCP stream
+OK <20867.1053212879@binkey.iticom.net>
USER barak
+OK
PASS youwish
-ERR authorization failed
QUIT

```

FIGURE 16-3 Capture of a failed POP3 session

POP3 Command Sequence

A POP3 session begins when the POP3 server sends its identifying message, which is composed of the process ID, a timestamp, and the server ID (can be a domain name or any other arbitrary name). A common response can look like this:

```
+OK 20750.1052874132@binkey.iticom.net
```

Let's analyze the response:

- 20750 is the process ID.
- 1052874132 is the timestamp.
- binkey.iticom.net is the domain ID.

We will go over POP3 common commands, just as we did with SMTP. Again, although the commands are listed in uppercase, reserving lowercase to indicate parameters, POP3 is case insensitive.

Command	Description
USER <i>username</i>	Sends the login username to the POP3 server
PASS <i>password</i>	Sends the password of the login username to the POP3 server
LIST	Gets the list of all the available e-mails
RERT <i>message number</i>	Gets a specific e-mail
DELE <i>message number</i>	Deletes a specific e-mail
QUIT	Tells the server to close the session

USER USER is the command that sends the user's username. Most POP3 implementations will always return + (which denotes success of the command) even if that user doesn't exist. (A legitimate user may be willing to waste a few seconds for a mistaken login, but that same amount of time may discourage hackers from trying to brute force the server, because they try thousands of login attempts and it sums up to a vast amount of time.)

Example

```
USER barak
```

Common Response

```
+OK Name is a valid mailbox
```

PASS PASS is the command that sends the password for the username sent before with the USER command.

Example

```
PASS imsureyouwouldliketoknow
```

Common Response (first for success; then for failure)

```
+OK Maildrop locked and ready  
-ERR Message Server said: Invalid login
```

A good mail server will always wait a few seconds after a failed login in order to slow down brute-force attack attempts.

NOTE *The PASS command sends the username and password in plaintext, which allows a hacker to sniff them.*

LIST LIST is the command that retrieves the list of all the messages that reside on the server. The POP3 server assigns each message a unique per-session ID number (starting from 1); it also supplies the message size (in octets). The reason the RFC uses an octet, an 8-bit number, is because on some operating systems a byte is not always 8-bit.

Example

```
List
```

Common Response

```
+OK scan listing follows  
1 118414  
2 29116  
3 30405  
4 61154  
.
```

A response for no messages would look like this:

```
+OK  
.
```

Note that the period marks the end of the list.

RETR RETR is the command used to retrieve a specific message. (For the current session, the only valid numbers are the ones specified by the LIST command.) The server will send

back the message in plaintext. (If you know the POP3 protocol by heart, you can use Telnet to check your e-mail.)

Example

RETR 1

Common Response

```
+OK 1302 octets
Return-Path: <barak@komodia.com>
Received: from frigg.inter.net.il (frigg.inter.net.il [IP])
by sauron.inter.net.il (Mirapoint Messaging Server MOS 3.2.2-GA)
with ESMTTP id AWC88752;
Wed, 14 May 2003 21:29:18 +0300 (IDT)
Received: from komodia (DNS name [IP])
by frigg.inter.net.il (Mirapoint Messaging Server MOS 3.2.2-GA)
with ESMTTP id CIY86793;
Wed, 14 May 2003 21:29:17 +0300 (IDT)
Message-ID: 200305142128060210.10A7A885@out.zahav.net.il
X-Mailer: Calypso Version 3.30.00.00 (4)
Date: Wed, 14 May 2003 21:28:06 +0200
Reply-To: barak@komodia.com
From: "Barak Weichselbaum" barak@komodia.com
To: barakwe@zahav.net.il
Subject: test
Mime-Version: 1.0
Content-Type: multipart/alternative; boundary="=====_105294048613864=_"

-----_105294048613864=_
Content-Type: text/plain; charset="us-ascii"

test

-----_105294048613864=_
Content-Type: text/html; charset="us-ascii"

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD>
<META http-equiv=Content-Type content="text/html; charset=iso-8859-1">
<META content="MSHTML 6.00.2800.1170" name=GENERATOR></HEAD>
<BODY style="FONT-FAMILY: Arial" text=#000000
bgColor=#ffffff><FONTsize=2>test</FONT></BODY></HTML>

-----_105294048613864=_--
```

FBI as Black Hats

Two Russian hackers, Vasily Gorshkov and Alexey Ivanov, were indicted by a U.S. district court, having been found guilty on charges of hacking into American companies. At first glance, nothing is special about this story, but the details of how the perpetrators were caught is quite interesting and relevant to this chapter.

The hackers stole more than 300,000 credit cards from online web sites. After the hack, they tried to extort money from the owners. After the FBI got involved, its agent found an online resume for Alexey Ivanov and invited him to consult a bogus computer company they had created just for the sting. The two hackers, believing their Russian nationality protected them, flew to the United States for the interview. (The company was located in Seattle.) During the interview, the agents asked the hackers to demonstrate their abilities using the company computers. (During that time, the agents ran key loggers and sniffers in order to obtain the hackers' passwords.)

After the demonstration was over, the FBI agents used the retrieved passwords to access the hackers' computers (in Russia) and extracted relevant information to press charges against them. (Russian authorities blamed the agents for hacking, but that's another story. For more information, visit www.securityfocus.com/columnists/105.)

The moral of this story is that passwords can be sniffed over unsecured protocols (such as SMTP). We can compare it to the ease which hackers can sniff data and/or passwords of an employee that connects to the corporate SMTP server using a wireless Internet at a coffee house. Most hackers will be able to sniff plaintext passwords quite easily.

Advanced POP3 (APOP3)

POP3 passwords are insecure, and everyone using the corporate LAN can sniff them; it is for this reason that Advanced POP3 (APOP3) was introduced. It allows a user to send their password MD5 hashed using a challenge (just like ESMTP MD5 logins). The syntax for APOP3 is shown here:

```
APOP username MD5 digest
```

The MD5 digest is calculated out of the process-ID and clock (which we mentioned earlier at "POP3 Command Sequence"), followed by a shared secret (pre-agreed by the client and server). The server checks the MD5 digest, and if it's correct the client is authenticated. (The session state is the same as after successful `USER` and `PASS` commands.)

IMAP4

Internet Message Access Protocol (IMAP4), defined in RFCs 1731, 2060, and 2061 ([[ftp://ftp.rfc-editor.org/in-notes/rfc1731.txt](http://ftp.rfc-editor.org/in-notes/rfc1731.txt), [ftp://ftp.rfc-editor.org/in-notes/rfc2060.txt](http://ftp.rfc-editor.org/in-notes/rfc2060.txt), and [ftp://ftp.rfc-editor.org/in-notes/rfc2061.txt](http://ftp.rfc-editor.org/in-notes/rfc2061.txt)]), is a plaintext mail protocol that combines aspects of both POP3 and SMTP. That is, it allows the user to send outgoing mail, but it

requires an SMTP server to do so. The user connects to the IMAP4 server (TCP port 143), authenticates itself, and can then start working. Unlike POP3 and SMTP, IMAP4 can work in two persistency modes: it can store all the data (all the incoming and outgoing mails) on the server or allow the user to work offline by storing the data locally, although remote storage is the default mode. Another difference between POP3 and IMAP4 is that IMAP4 allows users to create directories and to catalog their e-mail into these directories.

Because of its complexity, the details of IMAP4 are not covered here. If you want to read about its inner workings, you can refer to the RFCs listed in the preceding paragraph or to <http://www.imap.org>.

IMAP4 Authentication Method IMAP4 authentication options are almost like the options for POP3:

- It uses a plaintext username and password.
- CRAM-MD5 for encrypted logins (doesn't encrypt the data)

Comparison of POP3 and IMAP4 Support POP3 has traits similar to IMAP:

- Both can work while offline (not connected to the Internet).
- Mail is delivered to a server that always runs (or should run).
- E-mail can be retrieved using multiple clients from different vendors.
- Both protocols are “open” (that is, they are defined by RFCs).
- Both protocols need SMTP to send mail.

Here are specific features of POP3:

- It is a simple protocol and easy to implement.
- It works with a large variety of client software.

Here are specific features of IMAP4:

- It is optimized for speed (if you are using a slow network connection such as dial-up).
- It can store e-mails on the server or retrieve them locally.

SSL Support for POP3, SMTP, and IMAP4

POP3, SMTP, and IMAP4 all have SSL support. (SSL is discussed in Chapter 12.) Unlike regular POP3, SMTP, and IMAP4, in which even after the added authentication a hacker can still sniff the content of the mail, SSL encrypts the session from start to end making it almost impossible (encryption is all about statistics) to sniff either the login credentials or the session data. (SMTP and POP3 MD5 authentications protect from password stealing/sniffing only.) POP3 SSL (POP3S) uses TCP port 995, SMTP SSL (SMTPS) uses TCP port 465, and IMAP SSL (IMAPS) uses TCP port 993. These protocols are less used because SSL is a resource hog (encryption consumes most of the CPU), and a standard computer can have around 40–200 concurrent SSL sessions. (It's possible to add encryption hardware that supports more sessions, and most hardware encryption cards add support for as many as 300–500 concurrent sessions. Another solution is to add an SSL router that performs all

the encryption/decryption on behalf of the web server; those devices can reach 30,000 concurrent connections.) The following web site contains SSL benchmarks: www.webperformanceinc.com/products/performance/performancerealistic.html.

Hotmail and Web-Based E-Mail

Hotmail is the most-used free web-based e-mail service. What makes Hotmail or any other web-based e-mail so popular? The best answer is that it's accessible from anywhere. Even if you leave the office or commute to another country, you can access your e-mail from any computer connected to the Internet.

Hotmail appears in the news from time to time. (For example, hackers found another bug in it, and Microsoft quickly patched it; the process of hackers finding a bug and Microsoft fixing it seems to run in an endless loop.) In spite of such fixes, web-based e-mail programs have one major flaw: since most people use public computers, even if the username and password are encrypted (using SSL), the owner of the computer can log the user's keystrokes and capture their password.

Hotmail logging is a common problem for people using public terminals to log onto the Internet. Even if you use another kind of Internet-based e-mail, such as Outlook Web Access (OWA)—a package of ASP pages that allows an administrator to give its users the ability to access their e-mail from any web browser on the Internet), Yahoo mail, or other web-accessed e-mail systems that may use SSL, you still have to type your password. That means the owner could put key loggers and a screen capturer to intercept the user's username, password, and data.

Security Problems and Solutions

Now, after covering basic SMTP, IMAP4, and POP3, let's explore the realms of security (or perhaps insecurity is more precise, in this case):

- Most SMTP servers offer no authentication whatsoever; anyone can connect to the mail server and send e-mail.
- E-mail is sent using plaintext, so it's possible for anyone to use a sniffer on the corporate LAN or wireless network and see your e-mail.
- Anyone who wants to put my e-mail address (or any other address) as the source address (MAIL FROM) can do so and impersonate you or any address he desires.
- Vulnerabilities are discovered for SMTP, POP3, and IMAP4 server, but the percentage of administrators actually applying them is not so high.

No Authentication Even without authentication, the administrator of the SMTP server can distinguish between legitimate users and spammers in the following ways:

- Configure the SMTP server to allow internal users to connect to the server without any authentication, and accept mail from an external address only if it is destined to the local domain. (The server software uses either an IP range or a different server to separate internal and external users; for example, an ISP internal IP range might be x.x.x.0–x.x.x.255, so a request coming from y.y.y is treated as an outside user.)
- Use “flag an IP” to grant a specific IP full access to the SMTP server only if the IP logged in successfully to the POP3 server in the last *x*-minute interval.

Hazards of Using a Public Internet Access

One of the authors of this book had his first computer job developing an application to track down calling card frauds. His company wanted to find out how calling cards were compromised. One time he and his associates discovered that a video camera was installed to record people typing their calling card number. In another case, they found a telephone booth in India that travelers were using to make phone calls. The owner of this booth monitored all dialed numbers. By doing that, he stole the calling cards and sold calls to other travelers, defrauding the card holders. One can easily see the connection between this story and web-based e-mail services. When traveling, some people use Internet café services (coffee houses that offer Internet access). The problem is that the café owners can do the same as the owner of the telephone booth in India and install key loggers and a screen capturer to discover a customer's username and password.

NOTE *The fact that there is no authentication doesn't mean that the legitimate users have a "get out of jail card." If legitimate users use their ISP's servers to spam, most likely the ISP will terminate their accounts.*

Plaintext E-Mail

We surf the Internet daily, sending and receiving e-mail without fearing for our privacy. Should we? Privacy is a major subject that has powerful advocates, such as users and freedom activists that want to protect our privacy, and organizations (for example, banks and hospitals) that are required by law to protect our sensitive information. But privacy also has its foes. For example, some governments want to limit encryption usage in the name of security.

Some of you may have read George Orwell's novel *1984*, where "big brother" is watching, but does big brother watch us? The frightening answer is yes!

The FBI has a system named Carnivore installed at major American ISPs. (The actual number and locations are prone to speculation.) This system analyzes e-mail traffic (for which the coverage area is not publicly known) and looks for suspected keywords such as "terror," "bomb," and "suicide." The suspected e-mails are reviewed by agents to decide if they pose any danger. The FBI agents are required, however, to acquire a warrant to analyze such information. (The FBI provides some information about Carnivore, which can be found at www.cdt.org/security/carnivore/000724fbi.shtml and www.fbi.gov/hq/lab/carnivore/carnlrgmap.htm.)

The National Security Agency (NSA) has a system deployed in Europe called Echelon. As with its sibling, Carnivore, the exact location and coverage area of Echelon are unknown. This system tracks phone calls, trying to pinpoint suspects based on audio analysis in real time. More information about Echelon, published by NSA, can be found at www.nsa.gov/programs/tech/factshts/infosort.html.

These systems are not 100 percent reliable and some "data" may slip through the cracks, either due to bad design or bad coverage. In addition to Carnivore and Echelon, it is speculated that governments other than the U.S. government also have systems deployed and sniffing anything from e-mail, ICQ, and Kazaa to video chats and more.

Privacy seems harder to achieve with each passing day, and there are few reasons that may compel any one of us to encrypt our e-mails:

- Our government wants to limit our options and read every single e-mail we write or receive. (Some undemocratic governments are doing it already.)
- Some companies monitor their employees' e-mail.
- Protect our data from the prying eyes of hackers.

So what are our options? We said before that the standard authentication methods can keep our password safe but not the content of our e-mail:

- Use Pretty Good Privacy (PGP) to encrypt all your messages. (Note, however, that PGP is not bulletproof when dealing with the authorities, as you'll in the sidebar "Virus Semantics".)
- Use SSL to encrypt SMTP, POP3, and IMAP4 sessions. (Because SSL generates per-session keys and uses strong encryption, brute-forcing the session will take approximately 25 years.)
- Use *steganography*, which is when you hide data inside other data, such as a message inside a picture. Steganography is a good option when you fear that the government may force you to relinquish your key, because when you use steganography, the government first has to know the encrypted data exists!
- Use proprietary e-mail products or secure web e-mail programs, such as HushMail, that offer encryption as part of their solution.

NOTE *When using encryption for any purpose, make sure the key is strong enough. For example, note that a 40-bit key can be broken in a timeframe of several hours to one week.*

The Fight for Encryption

Encryption had become a very big concern for governments because users can encrypt their e-mail without any ability for authorities to read it, allowing the users to write about their next crime, illegal plans, money laundering, you name it. In most western countries, you can refuse to give the key based on the right to keep silent in order to avoid self-incrimination. However, some other countries may just throw you into a dungeon to make sure you'll reveal your key. (The issues surrounding the proper and improper use of encryption and how a government seeks to deal with them are summarized in the article "Cryptography and Public Key Encryption, What's the Big Deal," by Simon Baker; <http://simonbs.com/diz/diz.html>.)

The Regulation of Investigatory Powers Act gives the British government the power to require you to relinquish your encryption keys, so privacy activists developed software positions to defeat this. The software, called m-o-o-t, stores no data or encryption keys locally. (The article about the British regulation can be found at www.theregister.co.uk/content/55/25499.html, and the software can be found at www.m-o-o-t.org/.)

Impersonation As we stated earlier, anyone can forge your source address in their e-mail, so how do you know if senders of e-mail are who they proclaim to be? The sad answer is that if you're using only SMTP and POP3, there's no way. The solution is to use a digital signature. There are numerous products that allow you either to sign a document (that will be sent as an attachment) or e-mail or to verify the document (attachment) or e-mail you received. The problem is that you must use the same product on both ends; that is, you have to make sure whoever sent you an e-mail uses the same digital signature software as you. The most known product that allows you to sign a document is PGP—and it's free. (To see other commercial products, you can visit Yahoo's directory at http://dir.yahoo.com/Business_and_Economy/Business_to_Business/Computers/Security_and_Encryption/Software/Digital_Signatures/.)

Mail Server Vulnerabilities

Just like other network servers (web servers, for example), ways to exploit mail servers (all discussed protocols) have been discovered. The most common exploit found is buffer overflow (explained in detail in the "Buffer Overflow" section, later in this chapter). The overflow can be used either to crash the server or to run arbitrary code in the context of the exploited server. For example, an exploit discovered for IMate web mail is carried out by sending a HELO string that contains 1,119 characters or more (<http://www.securityfocus.com/bid/1286/exploit/>). A less common exploit, which cripples e-mail servers, is to send malformed e-mail headers that may cause the CPU to reach 100 percent utilization, thus slowing down the computer and slowing all e-mail delivery. (A similar exchange exploit that does the same thing can be found at <http://news.com.com/2100-1001-928055.html>.) The solution for these kinds of exploits is to make sure to patch the server with the latest security patches.

A hacker may wish to target your mail server (POP3, SMTP, or IMAP4) for various reasons:

- To use your server to send spam
- To read the e-mail of one or more users on the server
- To damage your server

Mail Distribution

Now that you've had a primer on SMTP, let's go over how SMTP "knows" how to deliver its data to the required destination, how mail servers interact, and what the differences are between direct connection and an open relay.

Mail DNS Entry

How does one mail server know where to send an e-mail? Suppose it receives an e-mail addressed to `barak@komodia.com`. Doing a regular DNS lookup (called "A record") is just like resolving a web address and will return the address of a web site; but the mail server's address doesn't have to be the same as the web address. To complicate things, each domain may have several mail servers. The solution is quite simple. You need to look up a Mail Exchangers (MX) DNS entry for this domain. (MX lookup is done under the hood and doesn't need user intervention.)

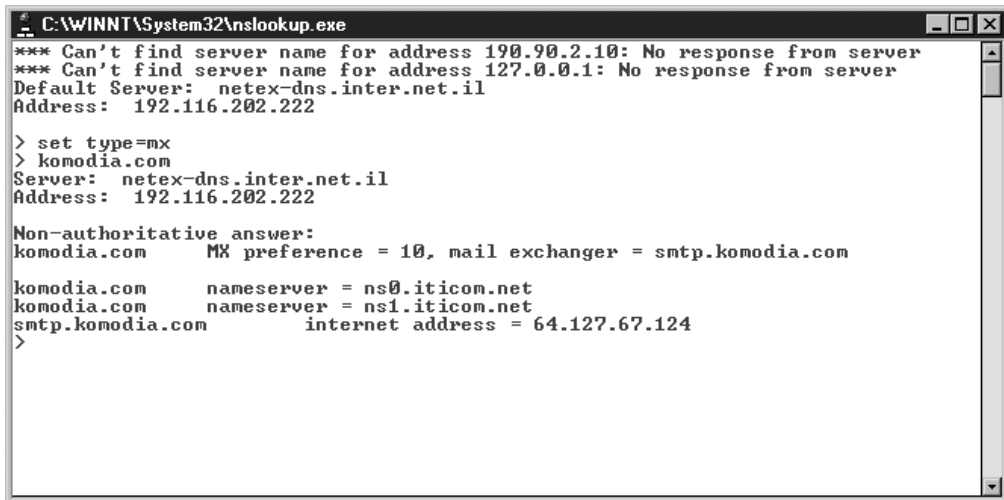
Here is an example that simulates what a server does (for Windows):

1. Run **nslookup.exe** from the command prompt. Normally it would point to a running DNS, but if it doesn't you can type **server IP** (where *IP* is the IP of the DNS server).
2. Type **set type=mx** and press ENTER. (This sets the program to look up MX addresses.)
3. Enter the name of the domain (**komodia.com**) and press ENTER.
4. If the address exists, it will be returned to you (see Figure 16-4).

The MX entry has a preference field that ranks the mail server priority, and other mail servers should strive to use the lowest preference. Now let's get back to the mail server. A mail server receives an e-mail for domain *x.com*. The first step it takes is to resolve its MX record. Assuming the server receives an answer with an IP of *x.x.x.x*, it will try to connect this address (TCP port 25) and will behave like a normal SMTP client and will send the e-mail.

The mail server may not be able to resolve the domain's MX record. If this is the case, it will send an error message to the sender, saying it can't resolve that address. (Usually it will add a copy of the original e-mail.)

Another possibility is that the server may not be able to connect to the given IP. If this is the case, it will try to connect to lower priority servers on the list at every arbitrary interval (defined per mail server) until it manages to send the e-mail or fails, resulting in a letter to the sender saying it wasn't able to deliver the message.



```
C:\WINNT\System32\nslookup.exe
*** Can't find server name for address 190.90.2.10: No response from server
*** Can't find server name for address 127.0.0.1: No response from server
Default Server: netex-dns.inter.net.il
Address: 192.116.202.222

> set type=mx
> komodia.com
Server: netex-dns.inter.net.il
Address: 192.116.202.222

Non-authoritative answer:
komodia.com      MX preference = 10, mail exchanger = smtp.komodia.com

komodia.com      nameserver = ns0.iticom.net
komodia.com      nameserver = ns1.iticom.net
smtp.komodia.com internet address = 64.127.67.124
>
```

FIGURE 16-4 NSLookup screen capture

Direct Mail

An advanced user can mimic a mail server and directly connect to the appropriate mail server. Suppose you want to send mail to someone@x.com and you don't want to use your mail server (or you don't have access to such a server). You can take the following steps:

1. Resolve the MX record for the domain x.com.
2. Connect to the IP you resolved (assuming it is resolved).
3. Send the message like a normal SMTP server.

Abusing Direct Mail What will happen if you add `RCPT TO` for domain y.com when you are connected to x.com's mail server? There are two options:

- The mail server will tell you it can't relay to a user outside the domain.
- The mail server will allow you to enter this address and act as an "open relay," which a spammer can use to send spam freely. (Open relays are discussed later in this chapter in the section "Open Relays.")

Spam and Spam Control

Internet is the de facto standard of communications today: e-mail, web sites, and instant messaging are taken for granted and are part of our daily lives, just like telephones. Direct marketing phone calls and spam share a lot in common. Just as we hate those pesky phone calls we receive at the most inconvenient times trying to sell us knives, life insurance, or some other gizmo we're not interested in, we hate spam that fills up our mailboxes and wastes valuable resources and time.

A common question we're asked over and over again is "Why do I get spam? I didn't submit my e-mail to any advertisement company." Spam is one of the Internet's largest plagues, but unlike unsolicited phone calls (which are far easier to trace, and are limited by laws.)

This section will discuss the following aspects of spam:

- Its origins
- How to fight it
- How to configure servers correctly to keep spammers off your network

Definition

First, let's start with the etymology of spam taken from Webster's dictionary: "from a skit on the British television series *Monty Python's Flying Circus* in which chanting of the word *Spam* (trademark for a canned meat product) overrides the other dialogue."

Spam® is a canned meat product that is made from leftovers of the animal (in other words junk). Spam is, therefore, a type of electronic "junk mail," or unsolicited e-mail attempting to sell commodities or services. The problems with spam are these:

- It wastes a great amount of bandwidth both for the ISP from which the spam is sent and for the ISPs whose mail servers receive the spam.
- It wastes user time to read and delete spam.
- Most spam source addresses are forged and may use the e-mail of a legitimate user. In this case, this user will receive all the bounced e-mail, as well as replies from other users.

Where Spam Comes From

Spam is the cheapest way to advertise a product, as it costs only \$120 (more or less) for a disc of 30 million e-mail addresses. With that and a fast Internet connection, voila! You're a bonafide spammer. When you do decide to purchase goods or services from a spam ad, make sure it's a legitimate business and not a scammer. (Check for information such as the company's physical address, as well as other contact information, and visit its web site. You might even make a phone call and see who answers.)

NOTE *It's difficult to understand the people or corporations that purchase goods and services from these spam ads because doing so makes spam an effective way of marketing and therefore contributes to the problem indirectly.*

A good example of a bulk mailing scam would be the Nigerian sting (the scammers are usually Nigerian). The victim receives a letter saying that the sender has \$20 million in his account (in Nigeria or in another third-world country), but for some reason he can't retrieve it.

The person asks for your help. This request varies, but in the end it says that you need to deposit \$5,000 (this amount varies as well) and that after they get the money you will be compensated with a large amount of money. (Of course, they welcome you to come to Nigeria and check it out yourself firsthand.) As you guessed by now, the victim will never see his money, or if he flew down there would either be robbed or kidnapped for ransom. (For a complete overview of this sting, visit <http://www.securiteam.com/securitynews/5RP01159FS.html>.)

How ISPs Fight Spam

ISPs strive to prevent their users from receiving spam; the easiest way to do so is to contract the services of a spam-fighting company that manages a Realtime Blocking List (RBL). The spam-fighting company maintains a real-time list of open proxies, open relays, and spammer IPs. Every connection coming from one of these IPs is treated as spam and is rejected. (More information about RBL services can be found at <http://spamcop.net/bl.shtml>.)

Another option is to use a spam-blocking software that resides on the mail server and blocks spam according to the following criteria (common settings):

- Repetitive source IP addresses
- Source IPs having no MX record
- DNS records having "dial-up" inside them
- Recurring subjects
- Body text having spam keywords such as "Viagra" or "sex"
- Recurring body text content
- Recurring source addresses
- More than one user per message
- Invalid message structure

Scam and Fraud

Howard Carmack, also known as the “buffalo spammer,” was arrested and charged with forgery and identity theft. It appears that Mr. Carmack used stolen credit cards to open accounts at Earthlink, an Internet service provider, to send spam.

All in all, Mr. Carmack allegedly sent 825 million spam e-mails from accounts he purchased using stolen credit cards, causing \$16.4 million in damages to the ISP. This story shows the degree to which spam is causing damage to ISPs (because it consumes a large amount of bandwidth) and that at least some scammers are getting arrested. For more information, visit <http://www.msnbc.com/news/913505.asp>.

Mail servers have a couple of ways to reject a spam session if it is identified in a search during or before the session:

- If the spam is identified because of the IP, the mail server can refuse to answer or respond with error code 553 or 554 after the spammer sends the HELO command.
- The mail server can allow the spammer to complete the session, giving the spammer indications that the mail was sent when actually it was discarded.

Because they have only partial success in blocking spam, these products are not silver bullets; it's true they block spam, but they can also block legitimate e-mail. (For example, if you were to send e-mail to Pfizer, the manufacturer of Viagra, to ask a question, most likely the reply would be blocked by the spam protection software.) A recent test by ICSA labs (<http://www.icsa.com>), an authority in security certification that certifies a range of security products such as personal and corporate firewalls, IDSs, and more, showed that because of the low performance of antispam solutions (they manage to block up to 60 percent of spam e-mails, but they can reach 80 percent in the expense of blocking legitimate e-mails), it's too soon to try to certify them. To read this story, you can visit <http://www.theregister.com/content/55/30546.html>.

Open Relays

Open relay servers are usually misconfigured servers that allow anyone on the Internet to use their services and send mail to all domains. Such relays exist because of misconfigured server settings and hacked computers that have a mail server installed. To secure a mail server so it won't be an open relay:

- Make sure the application is up-to-date.
- Set up a rule that limits outside users so that they can send e-mail only to users on your domain.

Open relays will be discussed further in the following three sections.

How Spammers Hide and Why

Why do spammers hide? The answer is quite simple: They hide because ISPs don't like them; they consume bandwidth and are a legal liability. What are the options for our average spammer? One of the most popular options is to use an open proxy. *Open proxies*

Testing for Open-Relay

A simple test to see if your server is an open relay is to set up an e-mail client with a bogus username outside your domain; for example, if your domain is x.com, choose a user test@y.com. Set up the mail server to be your mail server and try to send an e-mail to a user outside your domain (for example, test@z.com). If your mail server allows you to send the message, your server is an open relay

are computers that allow Internet users to relay data through them, showing the destination host the IP address of the proxy and not of the connecting computer. The spammer loads its spamming software with a large list of open proxies (usually up to 1,000, which can be acquired from various sites that maintain updated proxy lists). The software checks the proxies and generates a working list. The software then cycles the list and uses the proxies as relays for spam. (The user sees the proxy's IP and has no way of knowing the spammer's original IP.) On the other hand, when a spammer uses either direct mail or open relay to send its spam, the spammer's original IP appears in the header, as opposed to the proxy's IP.

Why do these proxies exist? Here are some reasons:

- People run proxies to acquire e-mails for creating a mailing list of their own. (The lists cost money, and this is one of the fastest ways to build one.)
- Governments want to monitor people who try to “anonymize” and guard their privacy.
- RBL companies want an early warning system for spam, so they add the proxies to their list before the competition, and they can log the source IP.
- Computers can run *spam zombies* (an application controlled by another party to use the host computer as a source for spam).

How to Fight Spam

When you receive spam (users often receive in the neighborhood of around 40 spam messages daily), you have a few options for combating this plague. You can reply to the sender; however, this is highly discouraged because the spammer will know the address is a working address, and your address will go from the spam list to the “sure working” spam list, which (assuming the source address isn't faked) results in—guess what? More spam! (To make life harder for spammers and help combat the plague, some mail servers don't report invalid addresses to the sender software, i.e. if I connect to such a server and try to send mail to a non-existent local account the server will report it was sent successfully.)

Another way spammers lure users is by adding an option to be removed from their “mailing list” (something like “Click here to be removed from this list”), which results in your address being shifted into the “sure working” list and in your receiving even more spam.

NOTE *When you receive spam, it's best just to delete it or add it to your e-mail program's junk filter. (Junk filters sort out e-mail according to source addresses, however, so they are not always very effective because source addresses are usually random and faked.)*

How can a corporation shield its users from this plague?

- Install a spam filter on the company's mail server.
- Make sure the company's mail servers aren't "open relays."
- Educate users how to handle spam (not replying or clicking anything).
- An administrator that has free time can complain about spam, as we outlined for single users.

How can you, an individual, protect yourself from this plague? Here are some suggestions:

- Complain to an online spam-combating service such as SpamCop (<http://spamcop.net/anosignup.shtml>).
- Track down the spammer's original IP address and send a complaint to the ISP that owns that address. (Sometimes the address will be a proxy, so you may not reach the spammer; but if the ISP decides to close the proxy, it will reduce the number of resources for all the spammers.)
- If the ISP doesn't resolve the issue, feel free to complain to its uproot ISP. (You may have heard in the media that a big ISP closed a smaller ISP.)
- The spammer has to give you some way to contact them (usually a web site address or phone number). Complain to the ISP that hosts this web site.

Usually when you contact the ISP, the ISP will shut down the spammer's account and add that individual to a black list. While that doesn't prevent the spammer from getting an account elsewhere, it will result in lost time and maybe the loss of potential clients that were lured into the spammer's web.

NOTE Most ISPs have an e-mail address called *abuse@ispdomain.com* that allows other Internet users to report abuses or spam coming from the ISP. You can send complaints to the address for the ISP to handle. Don't expect a reply, however.

Legal Issues

In the United States, there is a law to help people prevent direct marketing. If you receive a phone call and you ask the caller not to call you anymore, that caller is barred from calling you again for five years. If the caller does call you during that five-year period, you have the right to sue for damages.

NOTE There are people who have sued direct marketers and won substantial amounts of money. To read about actual cases that went to trial, visit <http://www.stopjunkcalls.com/savvy.htm>; to read rules on how to behave when a direct marketing call occurs, visit <http://junk.ro.nu/10.html>.

Spam is currently legal, but there are plans to make it illegal and make life harder for spammers. A proposed new bill would make it illegal to hide behind a proxy server (meaning the spammer's original IP would be shown). This bill was legislated to help

counter cable theft, but it can also be used against spammers that hide behind proxy servers. To read about this bill, visit <http://www.securityfocus.com/news/3912>.

A recent new spamming technique is to send a virus that takes over the victim's computer and turns it into a spam zombie. The zombie sends e-mail to its operator, supplying it the computer's IP while turning the computer into either a proxy server or an open relay mail server to be used by spammers. Of course, this is illegal and is considered hacking. To read more, visit <http://www.securityfocus.com/news/4217>.

Governments are starting to take action against open relay servers. In addition, some ISPs already maintain their own blacklists of open relay servers and ignore all mail coming from them. For example, the leading ISP, America Online (AOL), has one of the toughest spam policies. For further information, visit <http://www.msnbc.com/news/914094.asp>.

Viruses and Virus Control

The first Internet virus to succeed in causing damage was called "Morris," but it wasn't terribly lethal. To read more about the Morris virus, visit <http://www.wbglinks.net/pages/reads/misc/morrisworm.html>. The beginning of May 2000, on the other hand, signaled a major change in the computing industry where viruses are concerned. A Philippine national wrote a virus with a concept that wasn't very well known. It was a script that spread itself further via your e-mail. This virus was not the first Internet worm, but it was the first to cause so many headlines. The name of the virus was "Love Letter," but it was also known as the "I Love You" virus. To read more about the outbreak of this virus, visit <http://news.com.com/2100-1001-240132.html?tag=rn>.

Although the Love Letter virus wasn't deadly (that is, it didn't devastate the infected computer), it was serious because it was activated after the user opened the infected attachment. Once the virus was running, it sent itself to all the addresses in the user's e-mail address list. It wasted a lot of resources (some mail servers crashed because they couldn't handle the load) and signaled the beginning of a new era of viruses, each growing more dangerous and complex, but all having common characteristics. The reason those viruses are deadly is because unsuspecting users see an e-mail from a friend and they don't fear opening it, but once they do the virus uses their computers to spread even further. This section covers the following virus concepts:

- The evolution and history of viruses
- Antivirus solutions
- Other solutions for combating viruses

Evolution of Viruses

Viruses have grown more and more sophisticated and can now cause worldwide damage. The virus known as "Sircam" was the first to take documents from users' computers, infect them, and send them to the users' address list. The impact of this new plague was that it compromised corporate and personal data. Just imagine, for example, the damage that could be inflicted if an employee from Sun Microsystems were to receive a document from a Microsoft employee describing Microsoft's new marketing strategy. To read more about the impact of Sircam, visit <http://www.vnunet.com/News/1125834>.

The viruses of today are quite a leap if you remember the "old" viruses that spread via floppy disks and CD-ROMs.

Virus Nostalgia

Comparing today's viruses with old ones makes them look like a walk in the park. One of the authors remembers playing one of his first computer games (Art of War by Broderbund) and suddenly seeing ping-ponging all over his computer screen (a four-color CGA monitor). Looking back, he realizes he witnessed the first computer virus.

A *worm* is malicious code that propagates itself through known exploits, making it unnecessary for it to spread via e-mail. The Code Red worm was one of the first to exploit a bug in Microsoft's Internet Information Server (IIS) web server software, and it broke the record for speed of infection (the patch for this exploit was available for two years). Code Red was designed to attack the White House web site on a specific date, and it was suspected that the Chinese government was behind it. As a precaution, the site was brought down by its administrator that day to dodge the attack. To read more about this worm, visit <http://www.pcworld.com/news/article/0,aid,56504,00.asp>.

The Nimda virus (which spells "Admin" in reverse), also a worm, was the first to spread using all possible exploits: it exploited e-mail, SQL servers, web servers, and NetBIOS. After the virus infected a computer, it started to scan the Web and continue to infect other vulnerable machines. To read more about Nimda, visit http://searchsecurity.techtarget.com/qna/0,289202,sid14_gci770749,00.html.

Although those viruses and worms were the pioneers, there were, of course, other notorious viruses and worms, such as Klez, SQL Slammer, Goner, and B. Trans. The list is long.

One moral of this story is that if people had been more security-aware and kept their systems patched and up-to-date, most of the latest outbreaks could have been prevented.

NOTE *It's important to understand that personal users (moms and dads, for example, who buy computers, connect them to the Internet, and surf without any shred of security awareness) are hacked more easily than corporate users; and such hacked computers cause problems for corporations as well. Personal users' hacked computers are used to propagate malicious viruses even further, and because most users are connected via broadband, which is a very fast connection, the rate of infection is very fast as well. We had an idea how to prevent this kind of issue quite easily: ISPs could block any incoming connections for personal users (except maybe popular file-sharing software ports). In this scenario, if users wanted their ports open, they would need to sign a document stating that they're aware of the implications of such an action and that they take full legal liability if their computers should spread viruses or be used for hacking. This way, 95 percent of all connected personal consumers will not suffer from viruses that spread using network exploits. Note that it wouldn't prevent e-mail infection however.*

People Behind the Viruses

Most people tend to think that a geek with eyeglasses is in charge of the latest worm outbreak, but as IT pros, we know that's not completely true!

Hackers Hackers, in fact, write the majority of viruses; however, they are not the only culprits, as you will see in the following two subsections. Usually, hackers write viruses to gain prestige or to show that they're smarter than the people in charge of protecting against viruses.

Not All Security Teams Learn from Their Mistakes

While one of the authors was working as an R&D manager in a security startup company, his team deployed the product it developed as the company's main protection device. The team had two networks—one for the production, and one for testing. The testing network contained nonpatched computers, but those computers were protected by the product. One of the product's security policies was to disable Kazaa; but two of the programmers found the Kazaa blocking policy unacceptable, so they disabled all security policies, leaving the test network exposed to hackers and viruses.

The next day, the company was without Internet for a few hours. After investigating, it was learned that Nimda had infected one of the test computers and had started to attack and scan the network. If you think those two programmers learned their lesson, think again. They continued to run Kazaa and eDonkey in spite of what they had done.

NOTE *My personal opinion is that hackers do manage to outsmart the industry experts because it is evident how much damage the latest viruses cause in spite of all the security measures taken to prevent them.*

Governments Governments have been known to write viruses, too. Consider the previous discussion of the “Code Red” virus (see “Evolution of Viruses,” earlier in this chapter) or a more known fact—that the “Magic Lantern” virus was created by the FBI. It has been suggested that the American infrastructure is at risk for cyber attacks. For example, did one of the “evil axis” regimes plant a dormant Trojan to attack critical facilities at times of conflict? And even if it did, would your government tell you about it?

Why is the American infrastructure such a popular target? One obvious reason is that America isn't so popular nowadays because of its latest interventions in world affairs; another is just pure jealousy. In addition, America's infrastructure relies heavily on the Internet, which makes the Internet a perfect target for sabotage. (Note that the Internet didn't have as many hackers 10 and 20 years ago, so a decision to adapt the Internet as a communication medium for infrastructure at that time was quite logical.) Because of the existent motivation and ability to create viruses to cause damage, many experts expect infrastructure attacks in the next few years. (Although it is far from proven, North Korea, part of what is considered the “evil axis,” is suspected of having a school for cyber warfare. To read more about it, visit www.wired.com/news/politics/0,1283,59043,00.html.)

Disgruntled Employees Disgruntled employees make up the minority of virus writers; however, they have their share in the pie. Usually these employees don't wish to sabotage their workplace network directly, fearing they will be caught; instead, employees write viruses and spread them inside the workplace, relying on the assumption that the viruses will be blamed and that nobody will be able to trace them back to them.

NOTE *Writing a virus (or obtaining a customized virus) isn't as hard as one might suspect. Many generators of viruses are available on the Web. In addition, there are companies that offer viruses with customized abilities. Such companies even warrant that if an antivirus detects their virus in a one-year period, they will change the code to overcome it.*

Hackers from Within

Four years ago, a disgruntled female soldier at an instructional army base in Israel infected the base computer on purpose with a virus her boyfriend wrote. A member of the investigation team quite quickly discovered that this virus was tailor made. What gave the virus away?

- The computer had no network, but for some reason most computers were infected.
- The virus printed messages that corresponded with names of base officials.

The soldier was caught rather quickly and was sentenced by an army court to eight months in prison. Her boyfriend was arrested by the police, but the public and the media lost interest in this case, and it's unknown if he was indicted or not.

Crude Virus Removal Techniques

The best way to stop viruses from reaching the corporate network is simply to delete all incoming attachments. This prevention measure comes at a price, however—although the organization may never be hit by a virus, it won't be able to accept legitimate attachments such as documents and contracts.

Another option, which is available in the latest versions of Microsoft Outlook, is to deny access to files, such as EXE and DOC files, that may contain viruses. Most users find this option very annoying and feel that it hampers their productivity. Some users get around the option simply by sending the file with a different extension. For example, suppose someone wants to send you a file called contract.doc but because of this restriction sends it with another extension such as .do (contract.do). When the person on the other end receives the file, they can easily save it and rename it back to contract.doc. This is risky behavior. This example demonstrates that the arbitrary blocking of specific attachments can help reduce the threat of virus infection but can't prevent it totally because users can outsmart the system.

NOTE *It is possible that quite shortly we will see viruses that will attach themselves with a .do or .ex extension and will contain a message saying to save and rename the file back to an .exe or .doc extension.*

Antiviruses

After each wave of virus infections, antivirus (AV) companies are accused of not being able to address the virus in a timely fashion. (There is a joke security experts like to tell to customers who ask where viruses come from. They like to reply that AV companies produce AV software on the first floor and the viruses on the second floor.)

At present, there is no silver bullet for combating viruses. Every new virus infects freely until a way is found to counter it; for this reason, even if an update is available a few hours after the virus starts to spread, it won't help you if your system has already been hit.

- **Personal AVs** filter e-mail and scan local files
- **Corporate AVs** are managed centrally and offer virus signature updates

Virus Semantics

How do you know when a virus is a virus? This looks like a simple question, but the following story will make you think twice:

The FBI suspected Nicodemo S. Scarfo, Jr. of loan sharking and illegal gambling, but the feds couldn't prove it because the guy protected his documents using PGP. To overcome this obstacle, FBI agents installed a little FBI-made Trojan named "Magic Lantern." To make a long story short, they retrieved the passwords and convicted the guy. (To read more about this story, visit www.pcworld.com/news/article/0,aid,87084,00.asp.)

Is Magic Lantern a virus or not? In our opinion it is, but what did the AV companies think? Each company had pro and con arguments, but in the end the Russian AV company Kaspersky added the Trojan's signature into its AV signature database.

- **Gateway virus filters** are installed on mail servers and scan e-mail messages before the user receives them
- **Network gateway virus filters** filter all sessions that use a POP3 or IMAP4 port

All flavors of AVs need to be updated on a daily basis (or need to be updated on some other arbitrary interval determined by the administrator). It is the vendor, however, that decides how often to release new signatures. Out-of-date AV software is ineffective against new viruses.

The following table summarizes the advantages and disadvantages of different AV solutions:

Type	Location	Advantages	Disadvantages
Personal AV	Desktop	Easy to deploy Good for home users	Useful mostly for home and not corporate users because it can't be centrally managed
Corporate AV	Desktops with a central server	Easy to deploy Centrally managed	Can be disabled by the user
Gateway AV	Mail server	Filters viruses before the user can download them	Expensive Protects only the mail on the specific mail server on which it's installed
Network gateway AV	Corporate network gateway	Protects users that connect to any mail server	Expensive Can be bypassed if downloading mail from a different port

Protection Against Viruses

Viruses and worms take advantage of many security holes, so each infection method needs to be addressed differently.

Infection Method	How to Protect Against It
Sociable e-mail asking to open/read this document or view a nude picture of a famous actor/singer	User education AV software (all three flavors) Up-to-date patches on all the machines
Virus spread via known web and SQL exploits	Up-to-date patches on all the machines Personal or corporate antivirus on machines that run those services
Virus spread via NetBIOS	NetBIOS users always having passwords (a common practice at many corporations)
Unknown viruses spread via unknown exploits	Special products that reside on the web server and analyze all web requests, filtering them if an anomaly exists (although they don't offer 100 percent protection)

Recommendations for Securing E-Mail Servers

In order to secure your mail server, you can take these measures:

- Make sure your OS is up-to-date.
- Make sure your mail server is up-to-date.
- Place your mail server behind a firewall.
- Purchase an IDS or IPS to increase security.
- Allow users outside your domain to e-mail only the users on your domain (to prevent your server from becoming an open relay).
- If your users require access from outside the corporation, only allow them to connect using strong authentication such as CRAM-MD5.
- Consider which type of AV you wish to install.

Sandboxing

Currently, new technologies are researched to combat viruses more efficiently. Most of them focus on sandboxing (described next), some target processes, and some target the entire OS as one sandbox.

Sandboxing is a way to confine software to well-defined boundaries. For example, Java is a sandbox. If a virus is launched inside a sandbox, the sandbox should detect that the virus is trying to infect or delete files or that it is trying to access the Internet. Most likely, the sandbox rules will not allow such activities and will treat the software as hostile by deleting or quarantining it.

Some products already use some form of sandboxing. One such example is StormWatch by Okena.

Proxy Servers

In today's environment, every business has to have Internet to exist. Employees that surf the Web don't care how the data gets from point A to point B. Is it NAT? Direct access? A proxy server? On the other hand, it's important that the network architect or administrator understand the options available when deploying the enterprise network infrastructure.

This section will cover the following topics:

- Network connectivity scenarios
- Proxy connectivity
- Proxy security

Network Connectivity

Today's Internet address space allows almost four billion addresses (some addresses are reserved, so that's why it's "almost four billion"). What will happen when there are more than four billion computers connecting to the Internet?

Direct Connection

The easiest way (from an administrator's point of view) to connect an organization to the Internet is to assign every computer, device, and server with a real physical address. Therefore, if an organization has 1,000 computers and 10 network printers (assuming one router), it will require 1,010 real IP addresses. This method is usually a poor choice because it adds insecurity to the network (every single computer has a real IP and is accessible from the Internet) and because of the cost of leasing 1,000 IP addresses per month. The network topology for this configuration is shown in Figure 16-5.

NAT and PAT

Network Address Translation (NAT) is considered the "firewall for the poor" because it adds a subset of firewall functionality but with a fraction of the cost. NAT comes in two flavors on Windows machines: Internet connection sharing, as shown in Figure 16-6, and NAT, part of the "routing and remote access" service that comes with server versions of Windows NT, 2000, and XP. NAT allows the organization to set one computer with a real IP (or more if needed) and act as a gateway to the corporation's computers.

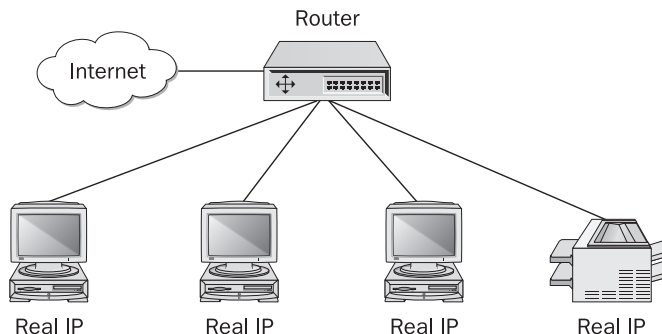
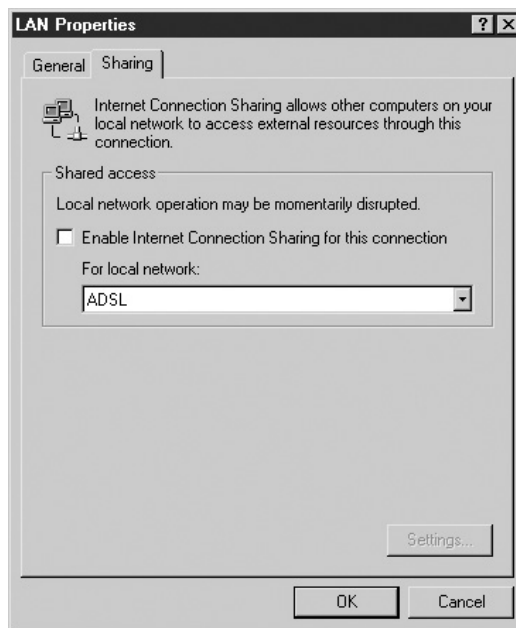


FIGURE 16-5 A direct connection to the Internet

FIGURE 16-6
Screen shot of
Windows 2000
NAT settings



Every internal computer has an internal IP, which is reserved and can't be used as an Internet address (most used addresses are 192.168.x.x), and the NAT server is defined as its

IP Address Types

What is the difference between a public address and a private address? *Private addresses* are those that can't be used on the Internet (a real Internet IP, or *public address*, can't be an address that is considered "private." Private addresses, defined in RFC 1918 ([ftp.rfc-editor.org/in-notes/rfc1918.txt]), were created to avoid situations where the corporation doesn't want its internal networks to use real IPs (for reasons stated in the "Direct Connection" section). Suppose a company assigned addresses to its internal computers that already belonged to someone else—google.com, for example. For most of the local-network Internet activities, everything would be smooth; but problems would arise when the users wanted to access google.com. Instead of reaching Google, their computers would try to access one of the computers on the LAN (because the corporation would be using Google's IP range). The solution to such a problem is simple. The Internet Assigned Numbers Authority (IANA), at <http://www.iana.org>, has the following allocated ranges, for use by private networks, that are not routable as Internet addresses:

- 10.0.0.0 – 10.255.255.255
- 172.16.0.0 – 172.31.255.255
- 192.168.0.0 – 192.168.255.255

NAT as a Firewall

NAT is considered the “firewall for the poor” because unless explicitly configured to do so, it doesn’t allow incoming connections to enter the internal network. However, unlike real firewalls, it doesn’t have stateful inspection (discussed in Chapter 11); therefore, if you know about an ongoing session, you can disrupt it. However, this scenario is very unlikely.

Note that it’s possible to count the number of computers behind a NAT server. Research elaborating on this subject can be found at www.research.att.com/~smb/papers/fnat.pdf.

gateway. The advantage is quite clear—you can lease only a limited amount of real IPs. (You can have one ADSL line serving up to 30 computers.) The problem arises when you want to deploy servers. The solution is quite simple. You can either use Port Address Translation (PAT), or you can assign real IPs to your servers. (The number of servers in an organization that need to be accessed from the Internet is limited.) The network topology in which the internal users are connected through NAT and the Internet servers are connected with real IPs is shown in Figure 16-7.

PAT Port Address Translation (PAT) allows a server to accept connections from the Internet without having a real Internet address. The NAT server is configured to forward all incoming connections into a specific port in the network to a specific IP and port. (The server can use a different port than the one used to listen on.) Figure 16-8 shows the network topology with NAT/PAT.

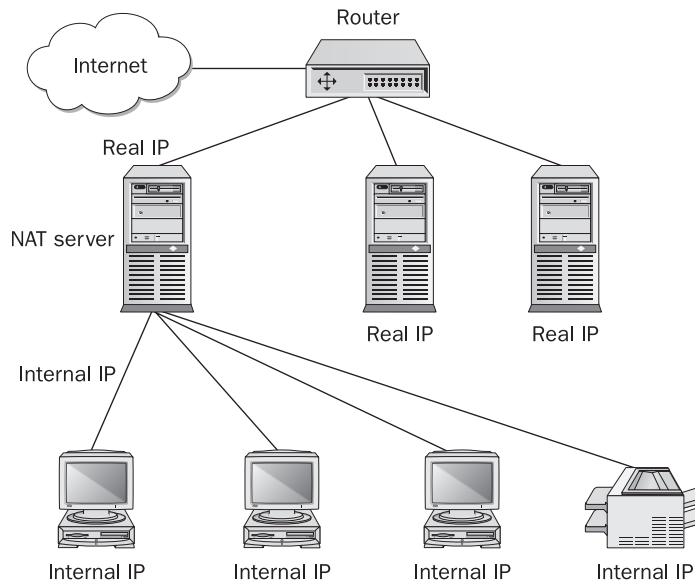


FIGURE 16-7 Internet servers connected with real IPs

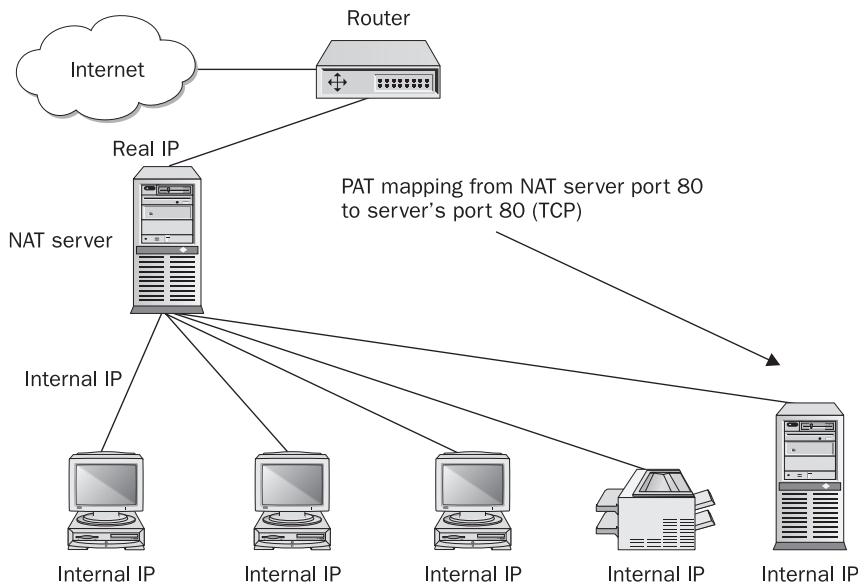


FIGURE 16-8 The network topology with NAT/PAT

A common misconception about NAT is that it allows Internet users to initiate sessions into the local network. In fact, an administrator has to explicitly map a port to accept connections destined into one of the internal computers.

Proxy Connectivity

Usually a network administrator has to choose between a proxy server and NAT. (Administrators usually choose according to preference. Most of the time, as you will see, we prefer proxy servers, although there have been scenarios where we had to deploy NAT.) The reasons for choosing NAT or proxy servers over direct connections are quite obvious: you don't need to buy an IP address per connected computer, and you gain the extra protection NAT offers. What are the reasons to choose proxy servers over NAT, or vice versa? To answer this, we first need to understand how proxies work.

Types of Proxies

Proxy server is a comprehensive term, as there are many different types of proxies. We'll go over common proxy types, giving details about each one, and provide a list of not-so-common proxies as well.

HTTP Proxy The HTTP proxy, as its name suggests, is used to return HTTP answers. The client connects to the HTTP proxy and requests the data using the HTTP protocol. The only difference between a request made to a proxy with a regular one is if the user wanted to query Google, instead of connecting to Google and sending `GET / HTTP/1.0`, the client would connect to the proxy server and send `GET http://www.google.com HTTP/1.0`. After the client has requested data from the proxy, it makes a request on the client's behalf.

(This proxy can also be used to cache results and increase surfing speed. Some ISPs deploy a transparent proxy that serves their users without any configuration at the user's end.)

FTP Proxy The FTP proxy acts just like the HTTP proxy. The client connects to the FTP proxy, and the proxy mediates between the FTP server and the client.

Direct Mapping An organization usually works with one SMTP server. SMTP has no built-in proxy support at the protocol level. An administrator can “direct map” a local IP and port to a remote IP and port—that is, the administrator can map the proxy server's IP and port 25 to the SMTP server's IP and port 25 (TCP). Users connect to the proxy's IP and are “tunneled” to the SMTP server. (Tunneling can be used for other services as well, such as UDP mapping. For example, we've created DNS tunneling, which uses port 53 UDP, when deploying a DNS server wasn't an option.)

NOTE A tunnel is a method for forwarding all connections made to a specific port at a tunnel server into another predefined computer and port. All the information received back from the final computer is forwarded into the original session initiator. Figure 16-9 shows a tunnel DNS deployment.

POP3 Proxy POP3 can be tunneled just like SMTP; however, sometimes you have to allow connections to multiple POP3 servers. (For example, it's possible to have three POP3 servers for three different e-mail accounts.) There are two options to deploy a POP3 proxy:

- Make one tunnel per server. (This can be unrealistic if you need support for a vast number of servers.)
- Use a dedicated POP3 proxy.

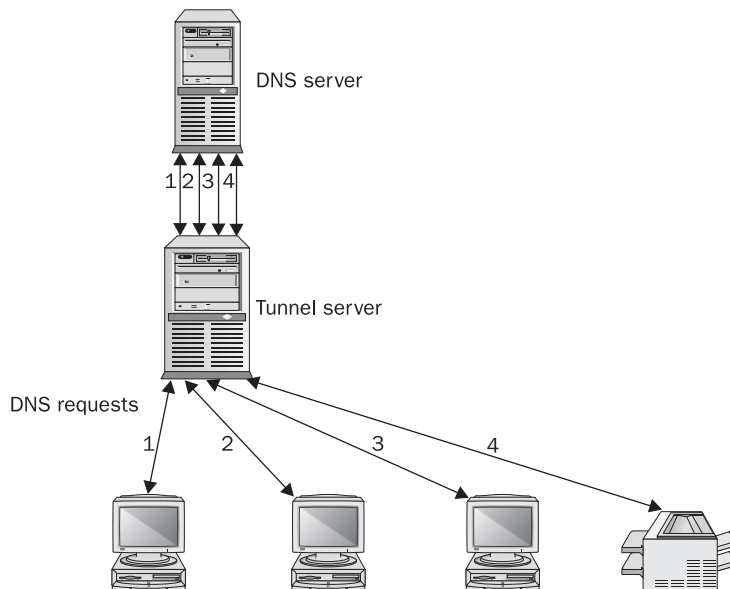


FIGURE 16-9 A tunnel DNS deployment

How does POP3 know which proxy server to connect? (Remember, the POP3 protocol doesn't support proxies.) After you deploy a POP3 proxy, you need to set the username delimiter (usually the # character). For example, if the user is barak and the POP3 server is komodia.com, you reconfigure your mail client username field to barak#komodia.com and you reconfigure the POP3 server field to be the proxy server. That way, the POP3 proxy can extract the server from the username.

SOCKS Proxy Internet applications emerge at an astonishing rate, and because it's not feasible to write a dedicated proxy for each one, there is a generic proxy type (SOCKS) that allows custom TCP protocols:

1. The application connects to the SOCKS proxy.
2. It tells the proxy it wants to connect to the IP and port. (It can also send unresolved domain names such as www.google.com.)
3. The proxy tries to do so and sends a code for success or failure.
4. In case of success, the application communicates with the requested server.

There are two types of SOCKS proxies—SOCKS4 and SOCKS5, defined in RFC 1928 ([ftp://ftp.rfc-editor.org/in-notes/rfc1928.txt]). These two protocols are quite different, and for most applications both will work just the same. (Note that the protocols are not plaintext and can't be used in Telnet.) The main difference between the versions is that SOCKS5 supports authentication and GSSAPI, whereas SOCKS4 does not.

HTTP Connect According to version 1.1 of the HTTP standard, defined in RFC 2616 and 2817 ([ftp://ftp.rfc-editor.org/in-notes/rfc2616.txt and ftp://ftp.rfc-editor.org/in-notes/rfc2817.txt]), a client may request that the web server open a Transport Layer Security (TLS) on its behalf. To do that, the client issues a CONNECT command, followed by the requested destination address and port (www.komodias.com:25), as shown in this example:

```
CONNECT www.komodias.com:25 HTTP/1.1
```

This will instruct the web server (or proxy server) to relay the connection to www.komodias.com port 25 (the server doesn't have to support the connection method), and it will reply with a standard HTTP answer, as in this example:

```
HTTP/1.1 200 OK
```

After the success command, the session will behave like a direct connection between the client and its requested destination address.

NOTE *In our experience, if your ISP deploys a transparent proxy and you need to use TLS, good luck. We've tried to resolve this issue with an ISP, requesting them either to use TLS or to allow the user to bypass their transparent proxy, but the ISP was very incompetent and didn't even understand what TLS was. (We needed TLS for testing a software application.) In the end, the solution was to use a dial-up account since there's no transparent proxy there.*

Other Proxies There are other types of proxies that are less used, but for completeness of this chapter, we'll mention them:

- Real Audio proxy (for streaming video)
- VDOLive (for streaming video)
- RealTime Streaming Protocol (RTSP) (for streaming media)

DNS and Proxies

Proxy servers can work with domain names (i.e., `www.google.com`) and perform the lookup at their end or receive a real Internet IP to connect to. Proxy servers can have DNS proxies as well.

Proxy Security Issues

Proxies are very secure because they offer a variety of security options, making them very secure, such as the following:

- Logging
- Interfaces
- Authentication
- Reverse proxies

Of course, the administrator needs to understand how to configure these options correctly!

Logging

Proxy servers allow you to log everything that is occurring in your system: connections, disconnections, login success and failure codes, and errors.

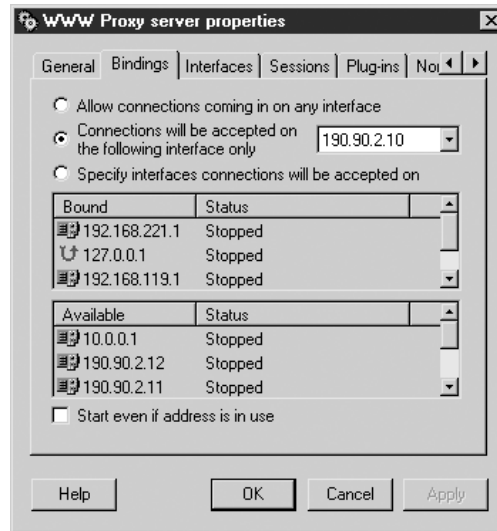
Interfaces

Proxy servers allow you to choose which interfaces you work with. A computer may have more than one network card, and you don't automatically want to serve all networks. Proxies allows you to choose, per service, which networks to serve and which networks to output the data from. (Figure 16-10 shows Deerfield's Wingate interface configuration.)

Faulty Proxy Installation "Invited" Hackers

The first time one of the authors installed a proxy server (it was seven years ago), he got burned. One day he received a call from his company's ISP saying he had a misconfigured proxy. After a quick investigation, he found that he had allowed anyone from the Internet to use the company's proxy server. (He had allowed the proxy to receive requests from all of the interfaces, including the Internet.) Hackers found the proxy and used it as a relay to try to hack other servers. Nevertheless, this was not their lucky day because he had logs of everything, logs which he then sent to his ISP. He immediately closed the breach (and learned a valuable lesson).

FIGURE 16-10
Deerfield's
Wingate WWW
proxy settings



In order to configure a proxy that serves the Internet network (unlike a reverse proxy, which is another story and is discussed shortly in the section, “Reverse Proxy”), the interfaces only need to be local. Having local interfaces instructs the proxy to receive requests only from the internal network.

Authentication

Proxy servers allow multiple methods of authentication:

- Username/password verification, which is part of the protocol itself (SOCKS5)
- An assumed username, which is verified according to a computer IP/name
- An authentication client, in which the computer runs a dedicated client that matches the specific proxy in order to authenticate with it

Reverse Proxy

One can configure a proxy to accept connections from the Internet (in this case, the interface would be the Internet and not the local network) and tunnel them back to a specific server, such as a web server or an SMTP/POP3 server. That way, if hackers should hack the proxy server, thinking it is the actual computer, they might succeed in stopping the service, but they won't be able to delete or deface the data. Figure 16-11 shows a reverse proxy configuration.

Securing a Proxy Server

When deploying a proxy server, make sure to do the following:

- Unless you are deploying a reverse proxy, never allow the proxy to accept connections from the public interface.
- Enable logging for all services.
- Require authentication for supporting services.
- Make sure the proxy server is running the most up-to-date software.

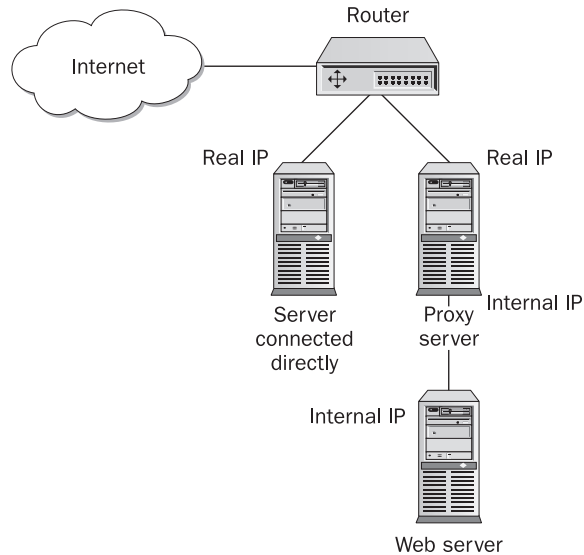


FIGURE 16-11 A reverse proxy configuration

DNS Servers

The Internet wasn't always like it is today. It started out in the late sixties and was called Arpanet. The primary motive behind Arpanet was the requirement to create a communication network for the government branches that would be reliable and unaffected by the failure of one or more gateways. (Mostly they wanted a network to withstand a nuclear attack.) The idea behind Arpanet was that if you had two points, A and B, and there were two possible routes for going from A to B (and vice versa), either through point C or point D, the network would choose the fastest route. On the other hand, if point C was the fastest, and it failed, the network would use point D instead.

Arpanet grew bigger and bigger (research institutions were granted access as well), making address administration very complex and difficult. Every site had to know the IP address of other sites, and soon it became hard to manage all those IPs. To solve this problem, one possibility was to create a host file that held the IPs and names of remote sites, but this file would require daily maintenance to add, remove, or modify entries. It became clear that a new solution was required, and the solution was DNS (Domain Name Service).

This section is an overview of DNS and DNS security. (For more information about the birth of Arpanet, you can visit www.lk.cs.ucla.edu/LK/Inet/birth.html.)

DNS Overview

The Domain Name Service (DNS), defined in RFCs 1034 and 1035 ([ftp.rfc-editor.org/in-notes/rfc1034.txt and [ftp://ftp.rfc-editor.org/in-notes/rfc1035.txt](http://ftp.rfc-editor.org/in-notes/rfc1035.txt)]), is a hierarchical naming service (i.e., it's built like a tree) that can be communicated over both TCP and

UDP (port 53). UDP is used more frequently than TCP, but TCP is required for zone transfers and long messages. Suppose you want to resolve the address `x.com`—what are the stages to do that? Most likely, you will issue a DNS query to the address’s DNS server (usually the ISP’s server), and if this domain exists, the server will return the resolved IP.

Protocol Overview

DNS is quite a complex protocol, and the structure of it is beyond the scope of this book, but DNS has some important behavioral properties that should be noted:

- DNS can transmit more than one question per query.
- A DNS reply can be made up of more than one answer.
- If queried from different locations, DNS can return different answers. (For example, Akamai manages its worldwide cache this way.)

Hierarchal Structure

What are the stages a DNS server performs in order to resolve the address of domain `x.com` or `y.net`? DNS is structured like a tree, where each node has its own servers. To resolve `x.com`, the DNS server looks for the root server of `.com` and queries it for `x.com`. Assuming `x.com` exists, this server refers it to the server that hosts `x.com` addresses. The DNS server queries it for the IP of `x.com`, and this should be its final stop. Figure 16-12 depicts this scenario and also shows servers for other root addresses.

Root Servers

The Internet has 13 DNS root servers. A resolver needs to start with one of these servers (see <http://www.root-servers.org> for a complete list). Every DNS answer has a timeout value (usually two days) that tells when this record may be changed, which means that if you try to resolve `x.com` and one of the root servers gives you the list of servers that handle `.com`, you can keep their address in your cache for the specific period of time. Then, the next time you have to resolve a `.com` address, you can query one of the `.com` servers directly until the timeout expires, at which time you’ll have to requery the root server.

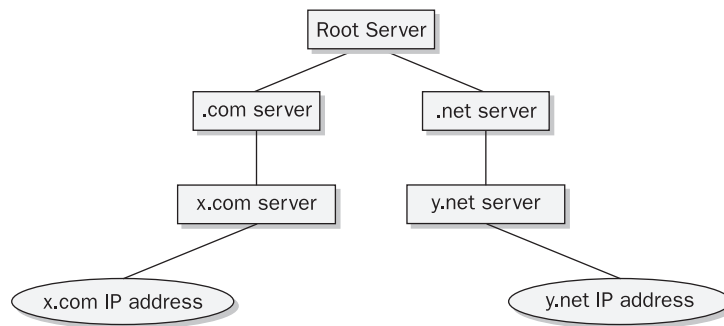


FIGURE 16-12 The stages that a DNS server performs in order to resolve the address of domain `x.com` or `y.net`

Practical Experience

In the following example using Windows 2000, let's try to mimic the DNS server way of working and try to resolve google.com like an ISP server would:

1. Run nslookup.exe.
2. Type **server a.root-servers.net** and press ENTER. (This will tell nslookup that you want to work with another server.)
3. Type **google.com** and press ENTER. (This will query the address of google.com.)
4. At this stage, the server should give you a list of other DNS servers that can resolve google.com, and it should look somewhat like this:

```
Server:  a.root-servers.net
Address: 198.41.0.4
```

```
Name:    google.com
Served by:
- A.GTLD-SERVERS.NET
  192.5.6.30
  com
- G.GTLD-SERVERS.NET
  192.42.93.30
  com
- H.GTLD-SERVERS.NET
  192.54.112.30
  com
...
```

5. Let's continue with the first server. Type **server 192.5.6.30** and press ENTER.
6. Type **google.com** and press ENTER.
7. You receive a list of additional DNS servers, which should look somewhat like this:

```
Server:  [192.5.6.30]
Address: 192.5.6.30

Name:    google.com
Served by:
- ns2.google.com
  216.239.34.10
  google.com
- ns1.google.com
  216.239.32.10
  google.com
- ns3.google.com
  216.239.36.10
  google.com
- ns4.google.com
  216.239.38.10
  google.com
```

8. From the name of the list of DNS servers, we can assume it's the DNS servers that actually host the IP, and again we will connect to one of them and try to query them. Type **server 216.239.34.10** and press ENTER.

9. Type **google.com** and press ENTER.
10. Now you receive a list of Google's IPs (finally):

```
Server: [216.239.34.10]
Address: 216.239.34.10

Name: google.com
Addresses: 216.239.33.100, 216.239.51.100
```

Something important to note is that this is how DNS servers resolve addresses. However, a client that connects to a regular DNS server will receive an answer without all this hassle.

DNS Security

DNS servers from different vendors share similar security problems, such as:

- Unpatched servers
- Misconfigured servers that allow zone transfer to unauthorized IPs
- Cache poisoning

Unpatched Servers

Berkeley Internet Name Domain (BIND), which can be found at www.isc.org/products/BIND, is a DNS server provided freely by the Internet Software Consortium (ISC) and is the most used DNS service for Unix machines. Numerous exploits have been discovered for BIND, and they are widely used to hack into systems. As with BIND, exploits have been found for other DNS servers; however, BIND exploits have been used in high-profile break-ins. To prevent security breaches, it's important to make sure you have the latest patches installed. (SecurityFocus runs a service that will notify you of any new exploits and inform you when a patch is available.)

Misconfigured Servers

Zone transfer is a DNS method for retrieving the full content of a DNS server. If the corporation maintains its internal addresses on a DNS server (like Active Directory does), a hacker can use this server to gain information about network topology and to gain computer information. The hacker connects to the corporate DNS server and requests a "zone transfer." After the transfer is complete, the hacker has a list of all of the corporation's computers and devices. This kind of attack helps a hacker to gain more information about the network that may help him carry an attack even further; however, it's not a must. To solve this problem, there is more than one solution:

- If the DNS is used for internal purposes only, block access to the DNS server from the Internet.
- Allow zone transfer to trusted IPs only.
- Block TCP DNS (zone transfers are done over TCP only, while regular DNS is usually UDP).

Inside Information and Social Engineering

Acquiring inside information can be a great asset for hackers. Suppose a hacker got the address of a network printer and managed to print something. Think of the possibilities the hacker has. He could print mountains of pages, wasting physical resources and hampering productivity. He could also forge a document to aid him in performing “social engineering,” the art of gaining information through conversation, to gain information from employees at that firm.

DNS Cache Poisoning

DNS poisoning is an old attack. It works like this: a hacker guesses the request ID of the server the hacker wants to poison and then sends it back a forged answer (with the IP the hacker wants). This type of attack is made possible for two reasons:

- Most DNSs use UDP, which is stateless and easily forged.
- The vulnerable implementation of a DNS server uses a sequential ID generator. For example, if it uses 1 as the current ID, it'll use 2 as the next, and so on. So it's easy for the hacker to guess the next ID range.

The hacker uses this attack for two purposes:

- For a denial of service (DoS) attack
- To lure users into a specially crafted site

The best solution is to make sure you run the most up-to-date DNS server. To read more about this type of attack, visit www.securityfocus.com/guest/17905.

Denial of Service A hacker can return invalid IPs (such as 127.0.0.1, which is LocalHost—an address that always refers to the local computer) to every request from the server, resulting in the corporation's inability to correctly resolve any domain names and thereby bringing productivity to a halt.

Luring Users into a Crafted Site To lure users, hackers can send back the IPs of their own crafted sites that resemble known popular sites in order to trick users into giving personal information, such as e-mail addresses, passwords, and credit cards. When users see a site they recognize (assuming the hacker did a good job imitating the original site), they have less fear submitting sensitive data. (A good example of a crafted site is www.microcrap.com, which resembles www.microsoft.com. The Microcrap site is very similar to Microsoft's and could be made to mimic Microsoft's site exactly if the owner wanted to take it further.)

Source Code Repository Access

Software companies consider their code to be their most important asset. If this asset is lost it might be going out of business. Source code repositories exist to help with code management and possess the following characteristics:

- Code is located in a central place.
- Support facilities exist to ease backup and restore.
- Version control is managed.
- Access to project limitations according to rights is enforced.

Basic Security

Most source code repositories share common security features:

- Enforce per-user access rights.
- Enforce rules to set projects as read-only for specific users. (This is a good idea because users can make mistakes on other people's projects.)
- Block user access to projects.

Because source code repositories usually are deployed in a trusted environment (that is, you have to trust your programmers to do their jobs), these security measures are usually enough. (On the other hand, if one of your programmers is an experienced hacker, they may have an easy time accessing the code repository—that is, breaking into the code repository computer.)

Advanced Security

The security problem starts when users need to work with code repositories remotely. Users need a secure way to do so, encrypting both password and data; since there's no "complete" off the shelf product that can do that, it is an advanced task.

VSS

Visual SourceSafe (VSS) is Microsoft's code repository server. It doesn't offer any encryption solutions, although it can work remotely via FTP. The passwords can be sniffed, as well as the code. Aside from purchasing third-party products, the only solution for this problem is to set up VPN connection to the corporate network. Figure 16-13 shows the VSS administration console.

CVS

Concurrent Versions System (CVS) is the most popular source repository software for *nix OS. (There's also a client version for Windows.) CVS allows the clients to be tunneled over terminal VPNs, such as SSH, solving the problem of encryption and passwords. The VPN connection is made under the hood and is initiated by the CVS client.

Other Solutions

There are other source code repository solutions that have encryption as part of their solutions. Two examples are Source Off Site (which Microsoft endorses in addition to its own SourceSafe product) and StarTeam.

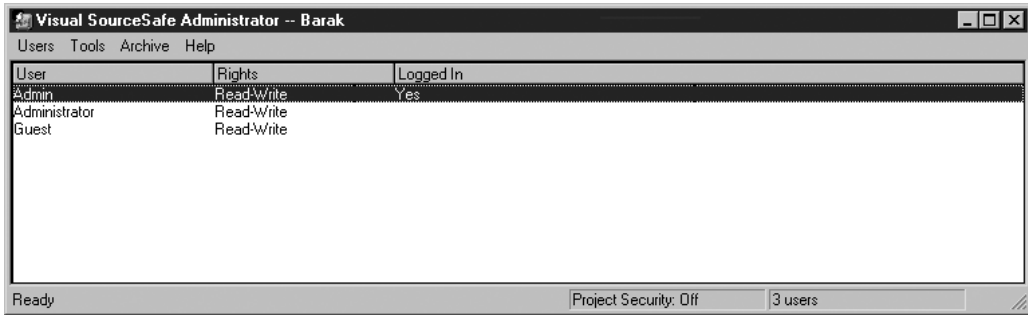


FIGURE 16-13 VSS administrator's console

NOTE *One very important issue concerning code repositories is the reliability of such systems. Poorly designed systems cause the users not to use the systems' full potential and hamper productivity.*

Liberal Security Settings May Cause Problems

When one of the authors was an R&D manager at a startup company, one of his programmers didn't know how to work with VSS. Instead of learning how to work with it, he complained about it. Because of his reluctance to learn how to operate VSS, he made fatal mistakes that caused the project to be duplicated. This duplication caused him to work with incorrect code. In the end the problem was located and the programmer's rights to create and delete projects were revoked.

Code repositories are a must for every development company. As long as a repository is configured correctly, using the correct hardware for the expected workload, it'll work rather smoothly. In addition, the administrator needs to make sure the appropriate security roles and rules are configured.

Outcome of Bad Source Control Implementation

A soldier in the Israeli army participated in the army's six-month computer programming course. During this course, the class used VSS as its code repository. The network was heavily used and very slow. In addition, VSS was misconfigured and was serving more users than the hardware was capable of. Those conditions caused VSS to "swallow" files—when users "checked out" files (taking the latest version from the server and replacing the local outdated file version), it sometimes erased them, leaving a blank copy. (Because of this behavior, VSS earned the nickname Virus SourceSafe.) In the end, users tended either to create a local backup before "giving" their files to VSS or just didn't use it. In general, VSS is a good solution, but in this specific scenario it failed because it was poorly deployed and administered.

Web Servers

Web security is divided into two parts:

- Web server security (web server configuration and software)
- Web application security (Java, ActiveX, PHP, and ASP)

This section covers web server security and solutions only. Web application security is covered in Chapter 23.

Overview of Web Server Security

A web server can be hacked into even if it runs the simplest HTML code possible:

```
<html>
<head>
<title>Hello world</title>
</head>
</html>
```

Because this HTML is so simple, a hacker can try two strategies to penetrate the web server: web application exploits and web server exploits. There is no way to exploit web application insecurities when a web server hosts only the HTML given as an example, so a hacker must try to hack the web server itself. There are far fewer web server exploits than web application exploits available; however, web server exploits are far more dangerous.

Goals of Server Attacks

Following are the goals of an attack carried against a web server:

- Web site defacement
- Data corruption
- Data theft
- Denial of service

The first three goals are covered in this section; denial of service is a common problem for most network services and is covered in Chapter 28.

Web Site Defacement Defacing is usually performed by *script kiddies*—Internet users who run known programs, checking for exploits that they can use to carry out attacks. Script kiddies often get caught, and in the security scene, being considered a script kiddy is as bad as being considered a newbie.

The web site is usually changed into something like this:

```
Box owned by SomeNickName
```

The site <http://www.safemode.org/mirror/> contains an archive of defaced versions of sites.

Data Corruption While web site defacement is actually data corruption of the HTML code, data corruption can be even deadlier. An entire web site can be deleted, as well as customer data, credit card data, and other important data.

If data is destroyed and the web site owner had a good backup policy, the damage can be minimized, but if the data is only modified (for example, if all the 1 digits in the credit cards are changed to 2), then it's a real problem.

Data Theft Is data theft more severe than web site defacement or data corruption? There's no simple answer. It really depends on the nature of the web site. Sometimes all three types of attacks can spell going out of business. Consider an online e-commerce web site that has its entire credit card database stolen. In addition to the negative press a site receives after an attack, the credit card company whose database was stolen can sue the web site owners for each credit card compromised, not to mention the angry clients that may have had their credit cards used in malice. On the other hand, consider what happens when a major online web site such as Amazon or eBay is defaced. For such high-traffic sites, an hour without net exposure means losses in millions of dollars.

Types of Attacks

Web server attacks are carried out successfully due to a couple of mistakes that can be easily avoided (which are elaborated in this section). It is much easier to secure a web server than to secure a web application. Here are some common types of server attacks:

- Exploiting known web server exploits (like buffer overflow, directory traversal, script permissions and directory browsing)
- Exploiting misconfigured related web services (like SQL server)
- Exploiting samples that are installed by default.

Buffer Overflow

Buffer overflow is a technique for injecting malicious code into applications. It works by corrupting the application *stack*—a place in memory where the application code is stored—and forcing it to do what the hacker wants (such as running Trojans or remote control applications). The simplest form of this exploit is to use C string functions in a way

DDoS Attack Spells Damages in Revenue

In February 2001, a coordinated DoS attack on eBay and Amazon caused damages in millions (users couldn't access these sites for a couple of hours) and brought web site security into the public's awareness.

Although this attack wasn't defacement, data loss, data theft, or corruption, it shows the potential impact of such attacks (DDoS) on e-commerce web sites.

To read more about this story, visit <http://abcnews.go.com/sections/tech/DailyNews/yahoo000208.html>.

the programmer doesn't anticipate. Here is some sample code, which you can skip if you don't know C:

```
char aTmp[100];
scanf ("%s", aTmp);
```

Here, the programmer declared an array that can hold 100 bytes (one char is one byte). The `scanf` method is used to read data from the console into local variables. However, it doesn't check that the size of the input can be contained in the variable supplied; and because the programmer didn't check for the size of the input string (for example, you can enter an input that is longer than 100 chars), it'll overflow into the code section. (Remember the array can hold 100 bytes.) A specially crafted input will include assembly code that will run inside the context of the vulnerable application and will have the same privilege as that application. (The code that executes will usually give a hacker remote access into the computer—for example, binding `cmd.exe` in Windows to a port allows a hacker to telnet into the computer and use the command prompt.)

A security company called eEye (<http://www.eeye.com>) discovered a buffer overflow exploit in IIS version 4. This exploit allows a hacker to take control over a web server, and from there the sky is the limit. (This same exploit was widely used for Code Red propagation.)

NOTE *Most of Apache's exploits are buffer overflows.*

Directory Traversal

Directory traversal is a method for accessing directories other than the allowed ones. The default IIS web site is located at `c:\inetpub`, assuming the OS (NT/2000/XP) is installed on drive C and the user didn't change any directory name (this is the case 99 percent of the time). Hackers may read files they weren't meant to. For example, let's assume our site name is `www.bad.com`:

```
http://www.bad.com/../../autoexec.bat
```

The `../../` tells the server to go one directory up, so if the server resides in `c:\inetpub`, the link will be transformed into `c:\autoexec.bat` (one directory up).

Unless the server is configured to allow script access on all directories (which will be discussed in the following section), the web server will return the content of `autoexec.bat` (or any other file we choose).

NOTE *We've used IIS as an example; however, this exploit was not exclusive to IIS alone.*

Script Permissions

In order to run the Common Gateway Interface (CGI), Perl, or other server-side applications the admin must grant executable permission to the directory where the server-side application resides. Some administrators grant this permission to the wrong place (usually because they don't understand the implications of it). Let's look at the following example to consider what would happen if the admin granted this privilege to all of drive C:

```
http://www.bad.com/../../winnt/system32/cmd.exe%20%2fc%20dir
```

Let's decipher what this cryptic URL actually means. Some characters, such as spaces and slashes, can't be used directly inside a URL; but sometimes you need to use them nevertheless. The solution is to represent such characters using their hexadecimal, or base 16, ASCII equivalents. (Base 16 uses the letters a, b, c, d, e, and f to represent digits greater than 9—for example, the letter *a* represents the number 10 in hex, and the letter *f* represents the number 15—and uses the number 10 to represent the digit 16.) So, in the preceding example,

- A space (), which in ASCII code is 32 in decimal notation and 20 in hex, becomes %20.
- A slash (/), which in ASCII code is 47 in decimal notation and 2f in hex, becomes %2f.

After the web server parses it, the URL will become:

```
../winnt/system32/cmd.exe /c dir
```

This will perform a `dir` command, listing all the files in the current directory, and will send the results back to the user. Of course, a hacker can perform even more complex commands in order to delete, run, or modify data on the web server.

Figure 16-14 shows the configuration screen of IIS directory permissions. The best practice is to set executable permissions to a directory that contains only the server-side application and not software that may aid hackers, such as `cmd.exe`.

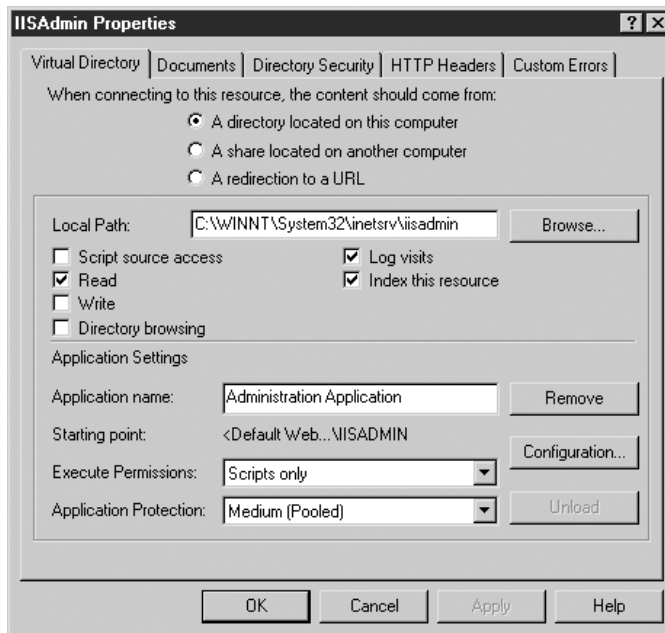


FIGURE 16-14 Screen shot of IIS script permissions console

Directory Browsing

Directory browsing is usually disabled, but if it is enabled it will show the list of all files in that directory and will allow browsing of subdirectories. Sometimes the knowledge of a file's existence can help a hacker to hack the web server. It's strongly discouraged to enable directory browsing unless you clearly understand the implications.

Default Samples

Default samples are somewhat in between web server security and web application security because some samples that are installed by default are vulnerable. So it's web application security, but because it's part of the default installation, it can be considered an aspect of web server security.

The best way to protect against this vulnerability is not to install the samples; and if they are already installed, just delete them.

Other Services

A hacker can hack a web server by hacking other services the web server is running, such as FTP, SMTP, POP3, SQL server, and NetBIOS. The best way to prevent this is to make sure that applications running on the web server are essential, and to secure them. Sometimes it's better to have one computer per service because if you place all services on one computer a hacker can compromise one service and use it to affect the other services, i.e. a computer that runs both a web server and a SMTP/POP3 server—if a hacker managed to take over the computer using an exploit in the SMTP server, he will be able to deface or take over the web site as well.

Other Exploits

Each web server has specific exploits that can't be categorized. For example, IIS has the .htr bug, which allows a hacker to see contents of files that reside on the server (usually these exploits are part of server-side scripts supplied by the vendor). To read more about this exploit, visit <http://neworder.box.sk/showme.php3?id=3772>. It's important to note that a patch is available as of April 2002, and it's noted in security checklists to delete this specific ISAPI. For more about this patch, visit support.microsoft.com/default.aspx?scid=kb%3Ben-us%3B318091.

Web Server Protection

Web server protection is easier than web application protection because you don't need to understand your programmers' code or look for insecurities inside, which sometimes is not even possible in the case of outsourced work. Taking the following measures will ensure your web server is secure:

- Set the web server service or daemon to run with the least amount of privileges possible. (That way, if a hacker takes control over the web server, there will be fewer options for hacking deeper into the computer or network.)
- Install the most recent security patches and keep track of new exploits discovered.
- Delete default samples or avoid installing them.
- Secure the computer hosting the web server by deleting unneeded applications, securing other network services on the same machine, and making sure the OS has the most up-to-date security patches.

- Make sure script permissions are given only to isolated directories that contain only the scripts in question.
- Have an index.html file for each directory so there will be no need to resort to directory browsing.

When a web server is hacked due to a known and old exploit, it is a result of pure negligence and carelessness of the administrator.

Third-Party Security Products

There are numerous product categories that help secure different aspects of web security, and some can even deal with unknown and new attacks (it's important to understand that there can never be 100 percent protection). The products that will be covered are:

- Antiviruses
- ISAPI-based products
- Secure logs
- Feedback analyzers
- Firewalls
- IDSs
- Vulnerability scanners
- Input validation

Antiviruses Antiviruses should be installed on the web server because if a hacker uses an unknown exploit and tries to inject a Trojan into the computer, most likely it's a known Trojan that the AV will detect and stop.

ISAPI-Based Products These products intercept URL requests and filter them for possible attacks such as buffer overflows. Microsoft is offering two free products for IIS: URLScan and IIS Lockdown, which increase IIS security. These products are installed by default in IIS 6. (EEye was one of the first companies to offer such a security product.)

Code Red Fiasco

One of the biggest ISPs in Israel, an ISP that is also a vendor of security products, had an unused web server displaying a default page of "Under Construction" that was taken over by Code Red. This virus scanned a user's computer a few times a day, so the user sent an e-mail to the ISP asking it to remove the virus. That helped for a few days until the virus infected the same machine again. The user sent the same e-mail again, and what do you know? The problem was solved. But to the user's surprise, the virus infected the same machine once again (for the third time). This time, the user threatened to contact the press and let them know about the incompetence of this ISP proclaiming to be the leader in security. Well, the third time was the charm; the user's threats must have scared the virus off since it didn't attack that computer anymore.

Secure Logs Secure logs are not actually used to protect the web server but serve as a repository for logs that prevents a hacker from changing the log and deleting the incriminating records.

One way to protect the logs is to configure a database to let the web server only insert records and not delete them. This way, the hacker won't have any ability to delete the server's log records.

Feedback Analyzers These products analyze the response of the web server and compare it to the original known web site. If the site is defaced, the response will not match the original web site and will be blocked. This won't prevent the down time of the site but will save the company from embarrassment.

Firewalls Firewalls are a good protection against TCP/IP protocol-level attacks (as opposed to content-based attacks) and are recommended for blocking unnecessary open ports. (Firewalls are discussed in Chapter 11.)

IDSs IDSs are good for after-the-hack investigations, but they are not 100 percent reliable even if an attack that they recognize occurred. (IDSs are discussed in Chapter 14.)

Vulnerability Scanners Administrators should run a vulnerability scanner periodically to test its web server security because if the scanner finds an exploit, so will a hacker. There are many types of vulnerability scanners; some are web-based, and some are commercial. The most popular scanner is Nessus; it's free and can be found at www.nessus.org.

Input Validation Input validation products are used to check every data submission to the web site and tests for signs of anomaly, SQL injection commands, and buffer overflows. The two leading vendors in this field are Kavado and Sanctum.

Choosing the Right Web Server

The decision of which web server to choose will affect not only the development cycle but also the security of your web site.

IIS Internet Information Server (IIS) is Microsoft's web server (the latest version is 6, part of 2003 .NET server). In the past, Microsoft was accused quite often about its security policy and the high rate of exploits found. Today, far fewer exploits are discovered in IIS than in the past. In addition, Microsoft has a better understanding of security than before, and therefore has improved its handling of new discovered exploits. Following are reasons to consider choosing IIS:

- It integrates into the Microsoft .NET framework.
- It runs ASP.
- You need commercial support.

Apache Apache is the most used web server for Linux and Unix platforms. In addition, it's free, it's open source, and has two popular versions: 1.3 and 2.0. Apache is considered to be more secure than IIS, but it does have its share of discovered exploits. In mid-2002, an

exploit was found that compromised 50,000,000 web servers running Apache: a hacker could cause the web server to execute arbitrary code. (More on this exploit can be found at <http://www.vnunet.com/News/1132708>.) Following are reasons to consider choosing Apache:

- It's free and it's open source.
- It doesn't require you to purchase an OS (although it can run on Windows as well).
- Support is available from many support companies (although not from the developers of Apache).

Other Web Servers Running web servers other than the two just discussed can be a double-edged sword because they are less used (or are used by no one else, in the event of a custom-made server). Although the chances of someone finding an exploit with less-popular servers is very slim, the fact is they are less tested, so there's a greater chance that old exploits (or exploits that use similar techniques) will be discovered than with the more popular servers.

IP Telephony and Streaming Media

In 1995, Vocaltec (www.vocaltec.com) shocked the world with a brand-new application that revolutionized the communication world. Its application allowed a voice conversation to be carried over the Internet. This application allowed consumers to save money over their normal phone calls. Instead of making expensive long-distance calls, two people could talk over the Internet without cost.

Voice over IP (VoIP) had its share of problems in the past. It had no open standard, Public Switched Telephone Networks (PSTNs) didn't support it, and its performance was poor due to the fact that routers treat all traffic as equal. As time passed, however, all of these problems were solved—an open standard was adopted, PSTNs now support VoIP, and routers support Quality of Service (QoS), a service that allows the prioritization of Internet traffic such as streaming media and VoIP.

Common Usage

VoIP and streaming media are very similar at the protocol level, but they share fewer traits when it comes to usage!

Using IP Telephony to Cut Costs

One of the authors worked for a company that had a branch in Antigua. Long-distance calls to Antigua were very expensive; so in order to reduce costs, he purchased a PC hardware card that could connect the computer to an analog phone. He connected the computer at his end to the phone system and set it up to control the computer and make a call to the company's branch overseas. The branch overseas did the same, and the company saved a substantial amount of money.

VoIP Usage

VoIP is mostly used for the following:

- Carry cheaper long distances calls. A company can lease an Internet line, set up two PBXs, and sell international calls. (In some countries such practices may be illegal.)
- Set up internal voice systems carried over the existing network infrastructure.

NOTE *Some ISPs limit the broadband upload rate. One of the reasons is to keep users from running VoIP servers and selling calls at low rates because they have only the expense of an ADSL or cable line.*

Streaming Media Usage

Streaming media is often used to conduct conferences. When a person's physical attendance is not needed, that individual can participate in a conference using a streaming media client. Another popular usage is to help students who are ill or students with disabilities to participate, or just listen to, classes they would have missed otherwise. Streaming media is also used extensively in the adult industry.

Streaming Media Protocols

The streaming media standard is comprised of a group of protocols, maintained by the International Telecommunication Union (ITU), at www.itu.int. The various protocols handle different bandwidth or connection types.

H263, defined in RFCs 2190 and 2429 (ftp.rfc-editor.org/in-notes/rfc2190.txt and ftp.rfc-editor.org/in-notes/rfc2429.txt), is one of today's streaming media standards. This protocol is layered over the Real-time Transport Protocol (RTP), defined in RFC 1889 (ftp.rfc-editor.org/in-notes/rfc1889.txt), which is built on top of UDP. It also uses RTP Control Protocol (RTCP), a subprotocol of RTP, for controlling the session. (It is built on top of TCP.)

We won't delve into the inner workings of both H263 and RTP because they are very complex and are beyond the scope of this book. (There are video and audio protocols older and newer than H263, but an entire book would be needed to cover VoIP and streaming video accurately.)

Key Features of VoIP/Streaming Media Protocols

We will quickly summarize the key features of the protocols before discussing their security aspects:

- They have no built-in authentication.
- They have no built-in encryption.
- They are layered over UDP.
- They have a feedback mechanism (RTCP) that tells the sender how many packets the receiver actually received. They use it to adjust the session quality in real time.

Security Issues of VoIP/Streaming Media Protocols

Today's VoIP/Streaming media protocols are insecure, and the reason for this is simple: any attempts to add extensive built-in security would result in the sacrificing of speed, which is critical for streaming media.

DoS Using Packet Injection

Most protocols rely on RTP, which uses UDP as its base protocol; therefore, hackers can inject malicious packets into a running session if they know the session ports and IP addresses because UDP is stateless and therefore easy to forge. This results in both sides being overloaded with data that can't be decoded.

Sniffing

VoIP/streaming media sessions can be easily sniffed (as can any other plaintext protocol) and then replayed because there's no encryption—only standard encoding.

Solutions

The following solutions are available:

- Use either H235, the security and encryption protocol for VoIP and streaming video (www.itu.int/rec/recommendation.asp?type=items&lang=e&parent=T-REC-H.235-200011-I) or tunnel VoIP/streaming media via VPN. (Note that because VPN increases Internet latency by adding overhead and encryption, it requires a very wide bandwidth that is out of reach for many clients.)
- Rely on product-specific authentication. Because authentication is not part of the standard, most products have their own proprietary authentication protocols.
- Set up your client/server to accept only specific IPs and users to establish and participate in streaming media sessions.
- Secure Internet servers to block hackers from breaking into the server using methods not directly related to the VoIP/streaming media protocols.
- Consider using dedicated VoIP switches that can help prevent eavesdropping and increase performance.

H263 Interception

A friend of one of the authors has developed a product for an international security firm that sells its product to governments. This product sniffs and records an H263 session, which can later be replayed or even transmitted to a remote client. The first time we saw this feature of the product, we were amazed. This product also analyzes other protocols, such as ICQ, FTP, SMTP, HTTP, IRC, and more.

Credit Card Security

The Internet is a “paradise” for retailers because it allows them to make their stores available 24 hours a day, 7 days a week, 365 days a year, and they know that millions of customers can log on. Most important, the “lease” for online retail is very cheap when compared to the price of operating a physical store. All of these features, along with the Internet boom, have driven many retailers to open Internet stores. Because most retailers don’t understand Internet security, they contract a company to handle security for them. Let’s view the typical flow of a service offer between a client and a contractor (where CL = client and CO = contractor):

CL: I’m looking for a company to build me an e-commerce web site.

CO: You came to the right place. We have built X number of web sites.

CL: Good. When can you finish it?

CO: How about next week?

CL: And what is the price?

CO: \$X.00

CL: Sounds good. Let’s go ahead.

Of course, business transactions usually aren’t as short as that; but we wanted to demonstrate the key topics discussed by an unwary client and a contractor. Have you ever seen a company that says “No”? Well, they are scarce—most companies just say “Yes” (even if they don’t have a clue) and then dump the problem on their programmers. We’ve seen it too many times. Note that in the preceding example, neither side talked about security. Why? Clients often don’t talk about security because they don’t realize the importance of it (can’t blame them, now can we?). And the contractor? Well, contractors may have several reasons not to discuss security: implementing security for the site may increase the price, or sadly, maybe they don’t have a clue how to implement it! Security requires quite a wide area of expertise, and most programmers just don’t care. They may do a great job designing web sites, but that work is performed often without any consideration for security. This section will list common credit card security mistakes and their solutions.

NOTE *Credit card fraud is a big problem, but there are ways to protect against it. For more information on how to avoid credit cards fraud, visit www.tamingthebeast.net/articles2/card-fraud-strategies.htm.*

Common Insecure Practices

Most of today’s insecure systems can be categorized into different mistake categories, which we will elaborate on here.

Credit Card Data Location

Some systems store credit card data on the front-end machine (the web server), so that if the computer is hacked, the hacker will have an easy time accessing the web site.

NOTE *We've seen far too many web sites that hold their entire customer data on an access database located on the front end.*

Failure to Use SSL

Secure Sockets Layer (SSL) is the de facto standard when it comes to e-commerce. Failure to add SSL support to an e-commerce site is just pure negligence. In addition, it may scare potential customers away. Quite simply, not using SSL allows hackers to steal credit card numbers by sniffing them on the LAN or wireless networks.

Application Insecurities

Web applications can have insecurities, too. If a professional writes a web application without keeping in mind security issues, most hackers will be able to exploit the application and gain access to users' accounts and personal data.

Contracting with a Company That Fails to Implement Security Properly

As previously stated, security is not a simple topic, and it requires quite awhile to become an expert in it. It's important to be sure that the deploying company is proficient at securing e-commerce services because there can be many points of failure:

- Installing a firewall without securing the web server
- Securing the web server without securing the web application
- Securing the web application without deploying SSL
- Misconfiguring the firewall
- Using SSL to protect user input, but retrieving the data using an insecure connection
- Keeping the data backup on an insecure server
- Allowing unauthorized personnel physical access to the server

And the list goes on and on.

Securing Credit Card Systems

Some basic steps can be taken to increase the security of credit card systems.

Credit Card Data Location

Credit card data should be placed on a secure server that is not connected directly to the Internet. If hackers succeed in hacking the web server, they'll nevertheless have a hard time accessing the secure server. Figure 16-15 shows a web server located at the DMZ that has access to a database server located inside the internal network.

Credit Card Accessibility

There's no reason to allow a web site to display credit card information. Users already know their credit card numbers—and if they need to remember which card they used (in cases where they have more than one), the site can display the last four digits of the card. This way, if hackers break into the system, they won't be able to retrieve the credit card information.

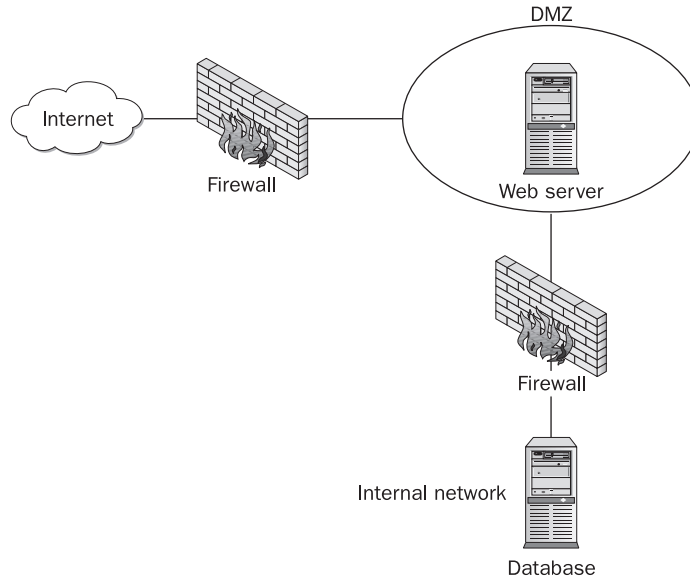


FIGURE 16-15 A web server located at the DMZ with access to a database server located inside the internal network

SSL

Deploying SSL connections gives you the advantage of security and wins you the confidence of clients. Not deploying SSL and using an e-mail post to send data from the user to the operator via e-mail is a big no-no.

Data Structure

To maintain security, keep client lists and credit card information separate. The data can be linked using a sequential ID or an MD5 hash of the credit card.

Data Encryption

Keeping credit card information encrypted solves two problems:

- If hackers gain access, they will need the key in order to extract the credit card data. (Most hackers will just give up and go on to attack other targets.)
- You protect against unauthorized persons looking for data (for example, if your server is located at an ISP and you can't physically protect it).

User-Related Security

Sometimes security-aware users demand extra protection. One thing they can do to increase their personal security is to delete their credit card information after the transaction is completed. Allowing your users to do so may keep those security-aware users coming to the site.

Printers and Faxes

Printers and faxes are an integral part of every office, but it's important to remember that hackers can exploit printers—and faxes can be “sniffed” just like regular e-mail!

Printers

Printers can be connected in the following three ways, the first two of which have their own security issues:

- Network printer
- Network computer with a printer attached
- Non-networked computer with a printer attached

Network Printer

A network printer is a network that is either connected directly to the network or has a special network hub connecting it to the network. Problems arise when administrators give the printer a real IP address (a public address). Think of the potential of this: hackers have complete access to a corporate printer without even having to scan for one exploit. In addition, local users can sniff and then view data going to the printer. (Note that the possibilities for hacking printers is discussed earlier in the chapter in the section, “Misconfigured Servers.”)

Network Computer with a Printer Attached

When a printer is connected to a computer connected to the network, the user has to have rights to print to it; however, most administrators allow full printer sharing to anyone.

Solutions

Printer security is often neglected and may be attributed to the lack of information published about the subject.

To ensure the security of network printers and printers that are connected to a computer, it is important not to assign real IP addresses to printers or computers that aren't servers! To add extra security, every user should be authenticated with the printer; most printers and printer hubs support this. Last but not least, IPSec should be used to avoid sniffing. (The latter is possible only when the printer is connected to a computer.)

Bad Printer Configuration Exposed

Vulnerability scanning performed for a client revealed that all of the office's computers were connected directly to the Internet (with real IPs), regardless of previous warnings to the client not to do so. (People just don't care until they get hacked!) Anyway, the client's printer had a real IP as well. Connecting to the printer's web interface revealed that the toner needed replacing soon. (Of course, this fact was included in the end report!)

Fax Security

This section deals with fax network servers and fax over VoIP, and will not deal with analog faxes (although it has its shares of security problems, it's not a network role).

Fax Servers

Fax servicing programs exist that can implement fax security. The most common way of working with a fax server is to send it an e-mail with an attachment and put the phone number in the header. Unless there's a good reason, the fax server shouldn't have Internet access because it can allow hackers to send faxes on their behalf or, even worse, to send faxes on the corporation's behalf (perhaps even faxes containing the corporate logo). If you find it imperative to allow your users remote access to the fax server, require the users to connect to the server using only SSL or VPN.

Fax over IP (FoIP)

When making a local or international phone call, you don't know if the call is channeled using the old, conventional method or over VoIP (nor should you care). The problem starts when you send a fax and the channeling medium is VoIP. VoIP may suit voice conversation, but it can't be used for analog signals such as faxes. The solution is to install VoIP routers that can identify a fax session and convert it into FoIP, which converts the fax into a digital image and transmits it as such over the IP-based network, decoding it on the other end and sending it as a regular fax.

Because FoIP can be transmitted over insecure lines, it's very important to install a router that can establish VPN connections to other routers, thus encrypting the data transmitted over the Internet.

Special Systems

There are many network roles that can't be categorized; that is, they are unique in their field. Two examples of such systems are SAP R/3 (by SAP) and BaanERP (by BAAN), which handle business-to-business (B2B) communications and enterprise resource planning (ERP). The security and pitfalls of each network role server should be examined carefully before deploying it. In this section, we'll list a number of subjects that should be checked for this kind of network role:

- OS security
- Intercommunication security
- Level of security support
- Auditing

Of course, a deeper check should be made for each network role server, since none are alike.

OS Security

Each network server is deployed on a specific OS (some are multiplatform) or on a hardware device. Before deploying the server, a company needs to be familiar with the OS and have the expertise to secure it and to address security problems that might arise.

Intercommunication Security

Because the server serves other clients, it uses the network as its communication medium. A secure product will allow (or may even require) a session to be encrypted or carried over a dedicated line.

NOTE *Some systems operate in a secure environment, and in such cases a secure session will not add security.*

Level of Security Support

Before purchasing a product, a corporation should make sure the vendor has a security policy and is known to supply security patches quickly after an exploit is discovered.

Auditing

Most ERP systems can reach hundreds of thousands of users. The administrator or division responsible for security should have an interface for auditing their users and looking for security breaches.

What does the administrator need to look for in logs?

- Login failures, because a few of them can indicate an attempted illegal entry
- User logins at “strange hours” (for example, weekends and after business hours)
- Excess usage, which may indicate abuse

Going over logs can be very tedious, but there are now vendors that sell software to help with log auditing.

How Not to Secure a System

A soldier was stationed with a squad that handled the manpower aspects of an infantry brigade. The squad used the army’s main manpower system. It knew the system allowed users from one unit to do transactions on behalf of other units, which is in violation of policy. Because one of the unit’s personnel failed to do their jobs and didn’t enter some necessary changes, the commander approved the use of the system breach so the soldiers could make the changes themselves. A month later, the terminal was blocked because of the incident, and the commander went to his superior officer to explain why he had approved it. (By the way, he got away with it.) The following lessons can be learned from this story:

- A system should enforce its security rules and not rely only on auditing.
- The auditing should have been done sooner. (Although the breach was used only to change records affecting the soldiers themselves and those changes needed to be made, this could have caused havoc in the system for one month without being detected.)

SCADA

Supervisory Control and Data Acquisition (SCADA) is a system used to automate and control process operations. It is mostly used in the following industries: chemical, food and beverage, glass, metals and mining, oil and gas, pharmaceutical, power (including nuclear power plants), pulp and paper, and water and wastewater.

This section includes the following topics:

- An overview of SCADA
- SCADA security problems and solutions

Overview

Suppose you have a factory with one circuit. This circuit can have two options, on and off—when the circuit is “on,” it lights a light bulb. One of our tasks is to make sure the circuit is always working. One option is to assign an employee to watch it constantly. This option may work with one circuit, but what happens if there are 2,000 circuits? The best solution is to connect a sensor to each circuit and connect them to one or more computers that monitor them, issuing alerts when one circuit or more stops. This is what SCADA is built for, although it can be more complicated than was portrayed at the beginning of the overview. SCADA systems can reach to over 100,000 sensors with different functions.

Typical SCADA Topology

SCADA is a system with no open standards. Standards are in development, but they are not yet fully adopted. Because SCADA spans a range of different industries, each with different needs, no two SCADA systems are the same. We will cover the most common topologies, but keep in mind that there may be other, less-common topologies.

SCADA Master

The SCADA master is a set of one or more computers that control and monitor all the sensors. (That is, usually it’s a computer, but it can be hardware as well.) The master has two ways it can retrieve data from its sensors: poll and push.

Poll The most common SCADA configuration is poll, in which the master connects to all the sensors at arbitrary intervals (from a few times a second to once every minute) and queries their status.

Push Less common than the polling mode is push, in which the sensor connects to the master and reports at intervals or when an anomaly occurs.

Sensors

Sensors monitor an aspect of a certain operation (for example, temperature, humidity, wind speed, or wind direction). A sensor is usually made out of small, dedicated hardware or a full-fledged computer. There are three types of sensors:

- Remote Telemetry Unit (RTU)
- Remote Terminal Unit (RTU)
- Programmable Logic Controller (PLC)

RTU

The lack of standards and the variety of industries and vendors are making RTUs incomparable. When deploying a SCADA system, most likely the RTUs need to be purchased from the same vendors because although there are converters to match the RTU of one vendor to the SCADA master of another, they are very expensive. RTU comes in two flavors, Remote Telemetry Unit and Remote Terminal Unit, that differ in functionality.

Remote Telemetry Unit The Remote Telemetry Unit is more like an I/O extension of the SCADA master. It has neither computing power nor logic. The SCADA master has to analyze its data and issue commands to it if needed.

Remote Terminal Unit This Remote Terminal Unit is an advanced version of the Remote Telemetry Unit that offers a method for programming it and has advanced logic without the intervention of the SCADA master. Most Remote Telemetry Units have the following features:

- CPU and memory
- Persistent memory to save data
- Communication capability (to connect to the SCADA master)
- Secure power supply (and usually a backup)
- Watchdog timer (to ensure that the RTU restarts if it fails)
- Electrical protection against grid instabilities

PLC

Programmable Logic Controllers (PLCs) use ladder logic. *Ladder logic* is a representation of relay logic consisting of two vertical lines with contact symbols along the rungs in between—hence the “ladder look.” PLCs can be used as stand-alone devices, but they are difficult to configure because doing so requires ladder logic. PLCs are best used as sensors and not for control operations.

NOTE *In the context of SCADA, we will use the word “device” to refer to an RTU, a PLC, or a control console.*

Control Console

A control console is one or more computers that show operators the current process flow, their status, and system alarms. The operator can control machines (that is, stop, start, and modify running values) that the SCADA system controls.

SCADA Security

The lack of a SCADA standard is a security advantage. Unlike other network roles where there are two or three major vendors, in the case of SCADA there are multiple industries each having multiple vendors and deployment topologies, making it harder for hackers to gain entry because hackers need specific knowledge of the system they want to penetrate. However, it is speculated that governments have more options for gaining this information and using it to hack other countries. (It has been speculated that Al-Qaida is trying to hack

into the American water supply. More information on this story can be found at <http://www.securityfocus.com/news/0319>.) There are various components of SCADA security:

- Hardware security
- Network security
- Identification security

Hardware Security

SCADA systems are evolving slowly; some systems still run on old platforms such as DOS, VMS, and NT4. Many exploits have been discovered for those operating systems, and unless they are patched, they can be used against those OSs. Most of the old attacks (such as land attacks or sending a specially crafted packet, modifying its source address to match the destination address) are harmless against modern OSs but can crash an old NT server. The solution is to make sure the OS is patched with the latest security patches.

Network Security

SCADA systems have multiple methods for connecting the SCADA master to its devices, and each has its pros and cons.

RS232 Some devices connect to the SCADA master via a serial port (RS232). This method is very secure because only the device connected to the port can communicate via this port. This method has many drawbacks, however, such as the inability to connect more than two devices. Unless you are installing dedicated hardware, distance from the SCADA master to the devices is very limited. In addition, RS232 wires are more expensive than regular network cables.

Modems Modems can be used as a communication medium between a device and the SCADA master. Both the SCADA master and the device must make sure that the device on the other line is in fact what it proclaims to be. For example, if a hacker learns a SCADA device phone number, that hacker can connect to the device at the end of the line and try to attack it or gain information. There are a number of solutions:

- Run a VPN session over the insecure phone line.
- Use caller ID and answer only calls from a predefined phone number.
- Use dial-back.

Wireless Wireless connections are easy to attack because a hacker can easily jam a wireless connection, rendering it useless. (See Chapter 13 for more about wireless connections.) Because interception of wireless traffic is easy, data sent over wireless connections should be encrypted to avoid snooping (VPN will usually be used).

Ethernet Although expensive, Ethernet is the easiest and most used method for connecting the SCADA master with its devices. Ethernet connections are susceptible to numerous attacks:

- Distributed denial of service (DDoS) attacks that overflow the network and may stop important data from reaching the devices or the SCADA master.
- Specific network attacks carried out against commonly used OSs such as Windows and Unix.
- SCADA systems connected to the corporate network. If the corporate network is hacked from the Internet, the hacker can continue to try to hack the SCADA system.

The solutions to these problems are as follows:

- Physically secure all network switches and ports, disallowing connections of “unauthorized” hardware that may carry an attack.
- Install an IDS and firewalls to protect the SCADA master and devices.
- Never connect the SCADA network to the corporate network! Doing so is pure negligence!
- Periodically conduct vulnerability assessments of your SCADA network.
- Make sure OSs are patched with up-to-date patches.

NOTE According to Windows XP's embedded license, you can't use it in devices that are critical and that control lives, such as nuclear power plants and life-supporting medical equipment.

Identification Security

SCADA systems can be used to identify employees and limit physical access to protected areas. There are several methods for identifying an employee, and each has its strengths and weaknesses.

Smart Card A smart card contains encrypted data, which is almost impossible to forge and can be checked by stand-alone consoles (they need only verify the encrypted data). The major drawback to smart cards is that they can be stolen. Because of this, each usage of such a card should be validated against photo or biometric data.

Airport SCADA Security

One of the authors visited an airport (we choose not to reveal which airport) with consoles scattered inside. The consoles allow passengers to check information about their flights. While examining the console, he noticed it was connected via an RJ-45 jack (a standard Ethernet connection) that he could unplug and connect to his laptop if he had wanted (of course he didn't). Think of the implications of such negligence: a hacker or terrorist could connect a device that will attack the network or could try to hack the system. It's sad that after the September 11 tragedy so much emphasis has been put on securing against such incidents at the macro level while neglecting the micro level.

Biometric Information Biometric information can be used to identify personal information using nonforged personal “features” such as the following:

- Fingerprints
- Retina fingerprints
- Hand geometry
- Face recognition
- Voice recognition

NOTE *For best security, it is advisable to use both smart cards and biometric information.*

PBX

A Private Branch Exchange (PBX) is a computer-based switch that can be thought of as a local phone company. Following are some common PBX features:

- Multiple extensions
- Voice mail
- Call forwarding
- Fax management
- Remote control (for support)

PBX has many security aspects, and this section can't cover them all. (An excellent article published by the U.S. government that details most PBX vulnerabilities can be found at <http://csrc.nist.gov/publications/nistpubs/800-24/sp800-24pbx.pdf>.)

Hacking a PBX

Hackers hack PBXs for several reasons:

- Gain confidential information (espionage)
- Place outgoing calls that are charged to the company's account (and thus free to the hacker)
- Cause damages by crashing the PBX

This section will briefly go over some common attacks, without delving into details.

Administrative Ports and Remote Access

Administrative ports are needed to control and diagnose the PBX. In addition, vendors often require remote access via a modem to be able to support and upgrade the PBX. This port is the number one hacker entry point. A hacker can connect to the PBX via the modem; or if the administrative port is shared with a voice port, the hacker can access the port from outside

the PBX by calling and manipulating the PBX to reach the administrative port. Just as with administrative privileges for computers, when hackers have remote administrative privileges, “they own the box” and can use it to make international calls or shut down the PBX.

Voice Mail

A hacker can gain information from voice mail or even make long-distance phone calls using a “through-dial” service. (After a user had been authenticated by the PBX, that user is allowed to make calls to numbers outside the PBX.) A hacker can discover a voice mail password by running an automated process that “guesses” easy passwords such as “1111,” “1234,” and so on.

Denial of Service

A PBX can be brought down in a number of ways:

- PBXs store their voice mail data on a hard drive. A hacker can leave a long message, full of random noises, in order to make compression less effective—whereby a PBX might have to store more data than it anticipated. This can result in a crash.
- A hacker can embed codes inside a message. (For example, a hacker might embed the code for message rewinding. Then, while the user listens to the message, the PBX will decode the embedded command and rewind the message in an endless loop.)

Securing a PBX

Here is a checklist for securing a PBX:

- Connect administrative ports only when necessary.
- Protect remote access with a third-party device or a dial-back.
- Review the password strength of your users.
- Allow passwords to be different lengths, and require the # symbol to indicate the end of a password, rather than revealing the length of the password.
- Disable all through dialing features.
- If you require dial through, limit it to a set of predefined needed numbers.
- Block all international calls, or limit the number of users who can initiate them.
- Block international calls to places such as the Caribbean that fraudsters tend to call.
- Train your helpdesk staff to identify attempted PBX hacks, such as excessive hang-ups, wrong number calls, and locked-out mailboxes.
- Make sure your PBX model is immune to common DOS attacks.

Each PBX model has its own specific exploits. Search hacker’s sites to find out what exploits they use against your PBX, and make sure you disable those exploits on your PBX. (A good site to start with is www.phrack.org.)

Summary

This chapter dealt with network role security and how to secure them. Throughout the chapter and various services covered you can see that the number one rule of security is to keep your system up to date and patched. By doing so you eliminate most automated attacks (script kiddies, vulnerabilities scanners, and worms). Using the rest of the security methods will protect against experienced hackers.