SMTP (Simple Mail Transfer Protocol)

Vladimir V. Riabov, Rivier College

INTRODUCTION

Electronic mail (e-mail) is one of the most popular network services nowadays. Most e-mail systems that send mail over the Internet use simple mail transfer protocol (SMTP) to send messages from one server to another. The messages can then be retrieved with an e-mail client using either a post office protocol (POP) or Internet message access protocol (IMAP). SMTP is also generally used to send messages from a mail client to a mail server in “host-based” (or Unix-based) mail systems, where a simple mbox utility might be on the same system (or via Network File System (NFS) provided by Novell) for access without POP or IMAP.

This chapter describes the fundamentals of SMTP, elements of its client-server architecture (user agent, mail transfer agent, ports), request-response mechanism, commands, mail transfer phases, SMTP messages, multi-purpose Internet mail extensions (MIME) for non-ASCII American Standard Code for Information Interchange data, e-mail delivery cases, mail access protocols (POP3 and IMAP4), SMTP software, vulnerability and security issues, standards, associations, and organizations.

SMTP FUNDAMENTALS

SMTP is used as the common mechanism for transporting electronic mail among different hosts within the transmission control protocol/Internet protocol (TCP/IP) suite. It is an application layer protocol. Under SMTP, a client SMTP process opens a TCP connection to a server SMTP process on a remote host and attempts to send mail across the connection. The server SMTP listens for a TCP connection on a specific port (25), and the client SMTP process initiates a connection on that port (Cisco SMTP, 2005). When the TCP connection is successful, the two processes execute a simple request-response dialogue, defined by the SMTP protocol (see RFC 821 for details), in which the client process transmits the mail addresses of the originator and the recipient(s) for a message. When the server process accepts these mail addresses, the client process transmits the e-mail instant message. The message must contain a
SMTP (Simple Mail Transfer Protocol)

The SMTP model (RFC 821) supports both end-to-end (no intermediate message transfer agents [MTAs]) and store-and-forward mail delivery methods. The end-to-end method is used between organizations, and the store-and-forward method is chosen for operating within organizations that have TCP/IP and SMTP-based networks.

A SMTP client will contact the destination host's SMTP server directly to deliver the mail. It will keep the mail item from being transmitted until it has been successfully copied to the recipient's SMTP. This is different from the store-and-forward principle that is common in many other electronic mailing systems, where the mail item may pass through a number of intermediate hosts in the same network on its way to the destination and where successful transmission from the sender only indicates that the mail item has reached the first intermediate hop (“Simple Mail Transfer Protocol” [SMTP], 2005).

The RFC 821 standard defines a client–server protocol. Both the SMTP client and server should have two basic components: UA and local MTA. Users deal with a user agent (UA). Popular user agents for UNIX include Berkeley Mail, Elm, MH, Pine, and Mutt. The user agents for Windows include Microsoft Outlook/Outlook Express and Netscape/Mozilla Communicator. The exchange of mail using TCP is performed by an MTA. The most common MTA for UNIX systems is Sendmail, and MTA for Windows is Microsoft Exchange 2000/2003. In addition to stable host-based e-mail servers, Microsoft Corporation has developed LDAP/Active-directory servers and B2B-servers that enhance mail-delivery practices. Users normally do not deal with the MTA. It is the responsibility of the system administrator to set up the local MTA. Users often have a choice, however, for their user agent (Stevens, 1993). The MTA maintains a mail queue so that it can schedule repeated delivery attempts in case a remote server is unable. Also the local MTA delivers mail to mailboxes, and the information can be downloaded by the UA (see Figure 1).

The RFC 822 standard specifies the SMTP protocol, which is a mechanism of communication between two MTAs across a single TCP connection. The RFC 822 standard specifies the format of the electronic mail message that is transmitted using the SMTP protocol (RFC 821) between the two MTAs. As a result of a user mail request, the sender-SMTP establishes a two-way connection with a receiver-SMTP. The receiver-SMTP can be either the ultimate destination or an intermediate one (known as a mail gateway). The sender-SMTP will generate commands, which are replied to by the receiver-SMTP (see Figure 1).

Both the SMTP client and server should have two basic components: UA and local MTA. There are few cases of sending electronic-mail messages across networks. In the first case of communication between the sender and the receiver across the network (see Figure 1), the sender’s UA message header and message text (“body”) formatted in accordance with RFC 822.

Mail that arrives via SMTP is forwarded to a remote server, or it is delivered to mailboxes on the local server. POP3 or IMAP allow users download mail that is stored on the local server. Most mail programs such as Eudora allow the client to specify both an SMTP server and a POP server. On UNIX-based systems, Sendmail is the most widely used SMTP server for e-mail. Sendmail includes a POP3 server and also comes in a version for Windows NT (“What is SMTP?”, 2005). The MIME protocol defines the way files are attached to SMTP messages. Microsoft Outlook and Netscape/Mozilla Communicator are the most popular mail-agent programs on Window-based systems.

The X.400 International Telecommunication Union standard (Tanenbaum, 2003) that defines transfer protocols for sending electronic mail between mail servers is used in Europe as an alternative to SMTP. Also, the message handling service (MHS) developed by Novell is used for electronic mail on Netware networks (“What is SMTP?”, 2005).

Figure 1: The basic simple mail transfer protocol (SMTP) model.
prepares the message, creates the envelope, and puts message in the envelope. The MTA transfers the mail across the network to the TCP-port 25 of the receiver’s MTA. In the second case of communication between the sending host (client) and the receiving host (server), relaying could be involved (see Figure 2). In addition to one MTA at the sender site and one at the receiving site, other MTAs, acting as client or server, can relay the electronic mail across the network.

The system of relays allows sites that do not use the TCP/IP protocol suite to send electronic mail to users on other sites that may or may not use the TCP/IP protocol suite. This third scenario of communication between the sender and the receiver can be accomplished through the use of an e-mail gateway, which is a relay MTA that can receive electronic mail prepared by a protocol other than SMTP and transform it to the SMTP format before sending it. The e-mail gateway can also receive electronic mail in the SMTP format, change it to another format, and then send it to the MTA of the client that does not use the TCP/IP protocol suite (Forouzan, 2003). In various implementations, there is the capability to exchange mail between the TCP/IP SMTP mailing system and the locally used mailing systems. These applications are called mail gateways or mail bridges. Sending mail through a mail gateway may alter the end-to-end delivery specification, because SMTP will only guarantee delivery to the mail-gateway host, not to the real destination host, which

Figure 2: The simple mail transfer protocol (SMTP) model with relay mail transfer agents.
is located beyond the TCP/IP network. When a mail gate-
way is used, the SMTP end-to-end transmission is host-to-
gateway, gateway-to-host or gateway-to-gateway; the be-

havior beyond the gateway is not defined by SMTP.

USER AGENT

Introduced in RFC 821 and RFC 822, the SMTP defines user agent functionality, but not the implementation de-
tails. A survey of the SMTP implementations can be found
in RFC 876. The UA is a program that is used to send
and receive electronic mail. The most popular user agent
programs for UNIX are Berkeley Mail, Elm, MH, Mutt,
Mush, and Zmail. Some UAs have an extra user inter-
face (e.g., Eudora) that allows window-type interactions
with the system. The user agents for Windows include
Microsoft Outlook/Outlook Express and Netscape/Mozilla
Communicator.

Sending e-Mail

Electronic mail is sent by a series of request-response
transactions between a client and a server. An SMTP trans-
action consists of the envelope and message, which is
composed of header (with From and To fields) and body
(text after headers sent with the DATA command). The en-
velope is transmitted separately from the message itself
using MAIL FROM and RCPT TO commands (see RFC
1123). A null line, that is, a line with nothing preceding
the <CRLF> sequence, terminates the mail header. Some
implementations (e.g., VM, which does not support zero-
length records in files), however, may interpret this dif-

ferently and accept a blank line as a terminator (SMTP,
2005). Everything after the null (or blank) line is the mes-

sage body, which is a sequence of lines containing ASCII
characters such as the colon are not significant. Many
quotation marks indicate single tokens within which spe-

cial characters such as the colon are not significant. Many
important field values (such as those for the "To" and
"From" fields) are "mailboxes." The most common forms
for these are the following:

- \texttt{johnsmith@mail.it.rivier.edu}
- \texttt{jsmith@mail.it.rivier.edu}
- \texttt{johnsmith<jsmith@mail.it.rivier.edu>}

The string "john Smith" is intended for human recipi-

ents and is the name of the mailbox owner. The string
"jsmith@mail.it.rivier.edu" is the computer-readable ad-
dress of the mailbox (the angle brackets are used to de-

limit the address but are not part of it). One can see that
this form of addressing is closely related to the domain
name system (DNS) concept (Internet Assigned Numbers
Authority [IANA], 2005). In fact, the client SMTP uses
the DNS to determine the IP address of the destination
mailbox.

Some frequently used fields (key words) are the follow-
ing:

- \texttt{to} Primary recipients of the message.
- \texttt{cc} Secondary ("carbon-copy") recipients of the mes-
  
  sage.
- \texttt{from} Identity of sender.
- \texttt{reply-to} The mailbox to which responses are to be
  sent. This field is added by the originator.
- \texttt{return-path} Address and route back to the origina-
  tor. This field is added by the final transport system that
delivers the mail.
- \texttt{subject} Summary of the message. The user usually
  provides the summary.

Receiving e-Mail

The UA periodically checks the content of the mailboxes
(see Figure 1). It informs the user about mail arrival by
giving a special notice. When the user tries to read the
mail, a list of arrived mail packages is displayed. Each
line of the list contains a brief summary of the information
about a particular package in the mailbox. The summary
may include the sender mail address, the subject, and the
time the mail was received or sent. By selecting any of the
packages, the user can view its contents on the terminal
display.

The SMTP Destination Address

The SMTP destination address (a mailbox address), in its
general form \texttt{local-part@domain-name}, can take several
forms (SMTP, 2005):

- \texttt{user@host}—For a direct destination on the same TCP/IP
  network.
- \texttt{user%remote-host@gateway-host}—For a user on a non-
  SMTP destination remote-host, via the mail gateway
  gateway-host.
- \texttt{host-a@host-burer@host-c}—For a relayed message.
  This form contains explicit routing information. The
  message will first be delivered to host-a, who will re-
  send (relay) the message to host-b. Host-b will then for-
  ward the message to the real destination host-c. Note
  that the message is stored on each of the intermediate
  hosts; therefore, there is no end-to-end delivery in this
case. This address form is obsolete and should not be
used (see RFC 1123).

Delayed Delivery

The SMTP protocol allows delayed delivery, and the mes-
cage can be delayed at the sender site, the receiver site, or
the intermediate servers (Forouzan, 2003).
There are a small number of commands (less than a dozen) that the client can send to the server. An example of sending a simple one-line message and an interpretation of the SMTP connection can be found in Stevens (1993). Although mail commands and replies are rigidly defined (see “Commands and replies” later in this chapter), the exchange can easily be followed in Figure 3.

In this scenario (Corner, 1995; SMTP, 2005), the user sam at host sun.it.rivier.edu sends a note to users darian, steve, and bryan at host mail.unh.edu. Here the lines sent by the server (receiver) are preceded by C, and the lines sent by the client (sender) preceded by S. Note that the message header is part of the data being transmitted. All exchanged messages (commands, replies, and data) are text lines, delimited by a ‘CRLF’. All replies have a numeric code at the beginning of the line.

The scenario includes the following steps (SMTP, 2005):

1. The client (sender-SMTP) establishes a TCP connection with the destination SMTP and then waits for the server to send a 220 Service ready message or a 421 Service not available message when the destination is temporarily unable to proceed.

2. The HELO command is sent, and the receiver is forced to identify itself by sending back its domain name. The client (sender-SMTP) can use this information to verify if it contacted the right destination SMTP. If the sender-SMTP supports SMTP service extensions as defined in the RFC 1651, it may substitute an EHLO command in place of the HELO command. A receiver-SMTP, which does not support service extensions, will respond with a 500 Syntax error, command unrecognized message. The client (sender-SMTP) should then retry with HELO, or if it cannot transmit the message without one or more service extensions, it should send a QUIT message. If a receiver-SMTP supports service extensions, it responds with a multiline 250 OK messages that include a list of service extensions, which it supports.

3. The client (sender) now initiates the start of a mail transaction by sending a MAIL command to the receiver. This command contains the reverse-path, which can be used to report errors. Note that a path can be more than just the user-mailbox@host-domain-name pair. In addition, it can contain a list of routing hosts. Examples of this are when the mail passes a mail bridge or when the user provides explicit routing information in the destination address. If accepted, the server (receiver) replies with a 250 OK message.

4. The second step of the actual mail exchange consists of providing the server SMTP with the destinations for the message (there can be more than one recipient). This is done by sending one or more RCPT commands. Each of them will receive a 250 OK reply if the destination is known to the server or a 550 No such user here reply if it is not.

5. When all RCPT commands are sent, the client (sender) issues a DATA command to notify the server (receiver) that the message contents are following. The server replies with the 354 Start mail input, end with <CR><LF> message.
6. SMTP (SIMPLE MAIL TRANSFER PROTOCOL)

   S: 220 mail.unh.edu Simple Mail Transfer Service Ready
   C: HELO It.rivier.edu
   S: 250 mail.unh.edu
   C: MAIL FROM:<jsmith8it.rivier.edu>
   S: 250 OK
   C: RCPT TO:<darien@mail.unh.edu>
   S: 250 OK
   C: RCPT TO:<steve@mail.unh.edu>
   S: 250 OK
   C: RCPT TO:<bryen@mail.unh.edu>
   S: 550 No such user here
   C: DATA
   S: 354 Start mail input, end with \r\n.\n.
   C: From: John Smith <jsmith8it.rivier.edu>
   C: Subject: Important meeting
   C: To: <darien@mail.unh.edu>
   C: cc: <steve@mail.unh.edu>
   C: Date: 26 Jan 2004 11:02:34 EST
   C: Best wishes
   C: See you soon...
   C: .
   S: 250 OK
   C: QUIT
   S: 221 mail.unh.edu Service closing transmission channel

   Figure 3: An example of the interactive session between the client (C) and the server (S).

   6. The client now sends the data line by line, ending with the sequence \r\n.\n.\n. on which the receiver acknowledges with a 250 OK or an appropriate error message if anything went wrong.

   7. The following actions (SMTP, 2005) are possible after that:
      - The sender has no more messages to send; he will end the connection with a QUIT command, which will be answered with a 221 Service closing transmission channel reply (see Figure 3).
      - The client (sender) has another message to send and simply goes back to Step 3 to send a new MAIL command.

   In this description, only the most important commands that must be recognized in each SMTP implementation (see RFC821) have been mentioned. Other optional commands (the RFC 821 standard does not require them to be implemented everywhere) implement several important functions such as forwarding, relaying, mailing lists, and so on.

   SMTP Commands

   The commands formed with ASCII (text) are sent from the client to the server. The simple structure of the commands allows for building mail clients and servers on any platform. The list of commands and their description and formats are shown in Table 1. The command consists of a key word followed by zero or more arguments. Five commands (HELO, MAIL FROM, RCPT TO, DATA, and QUIT) are mandatory, and every implementation must support them. The other three commands (RSET, VRFY, and NOOP) are often used and highly recommended. The next six programs (TURN, EXPN, HELP, SEND FROM, SOML FROM, and SAML FROM) are seldom used.

   For a full list of commands, see the RFC 821 "Simple Mail Transfer Protocol" and RFC 1123 "Requirements for Internet Hosts—Application and Support." For details of SMTP service extensions, see the RFC 1651 "SMTP Service Extensions," RFC 1652 "SMTP Service Extension for 8bit-MIME transport," RFC 1653 "SMTP Service Extension for Message Size Declaration," and RFC 2554 "SMTP Service Extension for Authentication." The commands normally progress in a sequence (one at a time). The advanced pipelining feature introduced in the RFC 2920 document allows multiple commands to be sent to a server in a single operation of the TCP-send type.

   Mail Service Types

   The set of services desired from a mail server are sometimes characterized by the "hello" key word. The various mail service types are as follows (Mail Parameters, 2002):
      - HELO for Simple Mail (see RFC821)
      - EHLO for Mail Service Extensions (see RFC1869)
      - LHELO for Local Mail (see RFC2033).
## Table 1: Simple Mail Transfer Protocol (SMTP) Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Format</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATRN</td>
<td>Authenticated TURN</td>
<td></td>
<td>RFC2645</td>
</tr>
<tr>
<td>AUTH</td>
<td>Authentication</td>
<td></td>
<td>RFC2554</td>
</tr>
<tr>
<td>BDAT</td>
<td>Binary data</td>
<td></td>
<td>RFC3030</td>
</tr>
<tr>
<td>DATA</td>
<td>Data, used to send the actual message; all lines that follow the DATA command are treated as the e-mail message; the message is terminated by a line containing just a period</td>
<td>DATA</td>
<td>Best wishes. RFC2821, RFC2B21</td>
</tr>
<tr>
<td>EHLO</td>
<td>Extended Hello</td>
<td></td>
<td>RFC1869, RFC2B21</td>
</tr>
<tr>
<td>ETN</td>
<td>Extended TURN</td>
<td></td>
<td>RFC1985</td>
</tr>
<tr>
<td>EXPN</td>
<td>Expand; asks the receiving host to expand the mailing list sent as the arguments and to return the mailbox addresses of the recipients that comprise the list</td>
<td>EXPN: a b c</td>
<td>RFC2B21, RFC2B21</td>
</tr>
<tr>
<td>HELO</td>
<td>Hello; used by the client to identify itself</td>
<td>HELO: sun.it.rivier.edu</td>
<td>RFC2B21, RFC2B21</td>
</tr>
<tr>
<td>HELP</td>
<td>Help; requests the recipient to send information about the command sent as the argument</td>
<td>HELP: mail</td>
<td>RFC2B21, RFC2B21</td>
</tr>
<tr>
<td>MAIL FROM</td>
<td>Mail; used by the client to identify the sender of the message; the argument is the e-mail address of the sender</td>
<td>MAIL FROM: <a href="mailto:jsmith@sun.it.rivier.edu">jsmith@sun.it.rivier.edu</a></td>
<td>RFC2B21, RFC2B21</td>
</tr>
<tr>
<td>NOOP</td>
<td>No operation; used by the client to check the status of the recipient</td>
<td>NOOP</td>
<td>RFC2B21, RFC2B21</td>
</tr>
<tr>
<td>QUIT</td>
<td>Quit; terminates the message</td>
<td>QUIT</td>
<td>RFC2B21, RFC2B21</td>
</tr>
<tr>
<td>RCPT</td>
<td>Recipient; used by the client to identify the intended recipient of the message; if there are multiple recipients, the command is repeated</td>
<td>RCPT TO: <a href="mailto:steve@unh.edu">steve@unh.edu</a></td>
<td>RFC2B21, RFC2B21</td>
</tr>
<tr>
<td>RSET</td>
<td>Reset; aborts the current e-mail transaction; the stored information about the sender and recipient is deleted; the connection will be reset</td>
<td>RSET</td>
<td>RFC2B21, RFC2B21</td>
</tr>
<tr>
<td>SAML</td>
<td>Send to the mailbox or terminal; specifies that the mail have to be delivered to the terminal or the mailbox of the recipient; the argument is the address of the sender</td>
<td>SAML FROM: <a href="mailto:jsmith@sun.it.rivier.edu">jsmith@sun.it.rivier.edu</a></td>
<td>RFC2B21</td>
</tr>
<tr>
<td>SEND</td>
<td>Send; specifies that the mail is to be delivered to the terminal of the recipient and not the mailbox; if the recipient is not logged in, the mail is bounced back; the argument is the address of the sender</td>
<td>SEND FROM: <a href="mailto:jsmith@sun.it.rivier.edu">jsmith@sun.it.rivier.edu</a></td>
<td>RFC2B21</td>
</tr>
<tr>
<td>SOML</td>
<td>Send to the mailbox or terminal; it specifies that the mail is to be delivered to the terminal or the mailbox of the recipient; the argument is the address of the sender</td>
<td>SOML FROM: <a href="mailto:jsmith@sun.it.rivier.edu">jsmith@sun.it.rivier.edu</a></td>
<td>RFC2B21</td>
</tr>
<tr>
<td>STARTTLS</td>
<td>Extended Hello with transport layer security</td>
<td></td>
<td>RFC3207</td>
</tr>
<tr>
<td>TURN</td>
<td>Turn; it lets the sender and the recipient switch positions whereby the sender becomes the recipient and vice versa (most SMTP implementations today do not support this feature; see RFC2821)</td>
<td>TURN</td>
<td>RFC2B21</td>
</tr>
<tr>
<td>VRFY</td>
<td>Verify; it verifies the address of the recipient, which is sent as the argument; the sender can request the receiver to confirm that a name identifies a valid recipient</td>
<td>VRFY: <a href="mailto:steve@unh.edu">steve@unh.edu</a></td>
<td>RFC2B21, RFC2B21</td>
</tr>
</tbody>
</table>

Note: From “SMTP Specifications,” 2005.
The EHLO key word has a numerical parameter SIZE for specifying the new format of e-mail messages (see RFC1870).

SMTP Service Extensions

SMTP (RFC821) specifies a set of commands or services for mail transfer. A general procedure for extending the set of services is defined in the STD11/RFC1869 document. The service extensions are identified by key words sent from the server to the client in response to the EHLO command (Mail Parameters, 2002). The set of service extensions are as follows:

- SEND—Send as mail (see RFC821)
- SOML—Send as mail or to terminal (see RFC821)
- SAML—Send as mail and to terminal (see RFC821)
- EXPN—Expand the mailing list (see RFC821)
- HELP—Supply helpful information (see RFC821)
- TURN—Turn the operation around (see RFC821)
- BITTME—Use 8-bit data (see RFC1652)
- SIZE—Message size declaration (see RFC1870)
- CHUNKING—Chunking (see RFC3030)
- BINARYMIME—Binary MIME (see RFC3030)
- CHECKPOINT—Checkpoint Restart (see RFC1845)
- PIPELINING—Command Pipelining (see RFC2920)
- DSN—Delivery Status Notification (see RFC1891)
- ETLN—Extended Turn (see RFC1985)
- ENHANCEDSTATUSCODES—Enhanced Status Codes (see RFC2034)
- STARTTLS—Start TLS (see RFC3207).

Some of these key words have parameters (for details, see Mail Parameters, 2002).

SMTP Responses

Responses are sent from the server to the client. A response is a three-digit code that may be followed by additional textual information. The meanings of the first digit are as follows:

- 2bc—positive completion reply; the requested command has been successfully completed and a new command can be started.
- 3bc—positive intermediate reply; the requested command has been accepted, but the server needs some more information before completion can occur.
- 4ab—transient negative completion reply; the requested command has been rejected, but the error condition is temporary, and the command can be sent again.
- 5ab—permanent negative completion reply; the requested command has been rejected, and the command cannot be sent again.

The second (b) and the third (c) digits provide further details about the responses. The list of typical reply codes and their description are shown in Table 2.

SMTP SERVER

The SMTP server sends and receives mail from other Internet hosts using the SMTP. The SMTP server processes all incoming and outgoing mail. Outgoing mail is spooled until the SMTP server can confirm it has arrived at its destination; incoming mail is spooled until users access it by using a POP3 or IMAP4 mail client. Spooling allows the transfer from client and server to occur in the background. The instructions on how to configure the SMTP server in the Windows NT environment and how to set options to provide security for the SMTP server are described in “How to Set SMTP Security Options” (2000).

ON-DEMAND MAIL RELAY

On-demand mail relay (ODMR), also known as authenticated TURN (ATRN), is an e-mail service that allows a user to connect to an Internet service provider (ISP) to authenticate and request e-mail using a dynamic IP address (instead of static IP addresses used in a "traditional" SMTP model) from any Internet connection (see RFC 2645). The initial client and server roles are short-lived, because the point is to allow the intermittently connected host to request mail held for it by a service provider. The customer initiates a connection to the provider, authenticates, and requests its mail. The roles of client and server then reverse, and the normal SMTP scenario proceeds. The provider has an ODMR process listening for connections on the ODMR port 366 (SMTP Specifications, 2005). On the server, this process implements the EHLO, AUTH, ATRN, and QUIT commands. Also, it has to be an SMTP client with access to the outgoing mail queues. An MTA normally has a mail client component, which processes the outgoing mail queues, attempting to send mail for particular domains, based on time or events, such as new mail being placed in the queue or receipt of an ETRN command by the SMTP server component. The ODMR service processes the outgoing queue on request. The ISP provider side has normal SMTP server responsibilities, including generation of delivery failure notices (SMTP Specifications, 2005).

MULTIPURPOSE INTERNET MAIL EXTENSIONS (MIME)

The RFC 822/STD 10 standard specifies that data sent via SMTP is 7-bit ASCII data, with the high-order bit cleared to zero. This is adequate in most instances for the transmission of English text messages but is inadequate for non-English text or nontextual data.

There are two approaches to overcoming these limitations. In the first approach, the MIME were defined in RFC 1521 and RFC 1522, which specify a mechanism for encoding text and binary data as 7-bit ASCII within the mail envelope defined by the RFC 822 standard. MIME is also described in SMTP (2003).

In the second approach, the SMTP service extensions (RFC 1651, RFC 1652, and RFC 1653) define a mechanism to extend the capabilities of SMTP beyond the limitations imposed by the RFC 821 standard. The RFC 1651 document introduces a standard for a receiver-SMTP to
inform a sender-SMTP, which service extensions it sup-
pports. New procedures modify the RFC 821 standard to
allow a client SMTP agent to request that the server re-
sponds with a list of the service extensions that it sup-
pports at the start of an SMTP session. If the server SMTP
does not support the RFC 1651, it will respond with an
error and the client may either terminate the session or
attempt to start a session according to the rules of the RFC
821 standard. If the server does support the RFC 1651, it
may also respond with a list of the service extensions that
it supports. A registry of services is maintained by the
Internet Assigned Numbers Authority (IANA, 2005); the
initial list defined in the RFC 1651 document contains
those commands listed in RFC 1123 as optional for SMTP
servers.

Specific extensions are defined in RFC 1652 and RFC
1653. A protocol for 8-bit text transmission (RFC 1652)
allows an SMTP server to indicate that it can accept data
consisting of 8-bit bytes. A server, which reports that this
extension is available to a client, must leave the high-order
bit of bytes received in an SMTP message unchanged if
requested to do so by the client.

The MIME and SMTP service extension approaches are
complementary. Following their procedures (RFC 1652),
nontraditional SMTP agents can transmit mes-
sages, which are declared as consisting of 8-bit data rather
than 7-bit data, when both the client and the server con-
form to the RFC 1651 or RFC 1652 options (or both).
Whenever a client SMTP attempts to send 8-bit data to
a server, which does not support this extension, the client

---

Table 2 Simple Mail Transfer Protocol (SMTP) Reply Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>211</td>
<td>System status or system help reply</td>
</tr>
<tr>
<td>214</td>
<td>Help message</td>
</tr>
<tr>
<td>220</td>
<td>Domain service ready; ready to start TLS</td>
</tr>
<tr>
<td>221</td>
<td>Domain service closing transmission channel</td>
</tr>
<tr>
<td>250</td>
<td>OK, queueing for node node started; requested command completed</td>
</tr>
<tr>
<td>251</td>
<td>OK, no messages waiting for node node; user not local, will forward to forwardpath</td>
</tr>
<tr>
<td>252</td>
<td>OK, pending messages for node node started; cannot VRFY user (e.g., information is not local) but will take message for this user and attempt delivery</td>
</tr>
<tr>
<td>253</td>
<td>OK, messages pending messages for node node started; cannot VRFY user (e.g., information is not local) but will take message for this user and attempt delivery</td>
</tr>
<tr>
<td>254</td>
<td>OK, node node started; requested command completed</td>
</tr>
<tr>
<td>255</td>
<td>OK, no messages waiting for node node; user not local, will forward to forwardpath</td>
</tr>
<tr>
<td>354</td>
<td>Start mail input; end with &lt;CRLF&gt;,&lt;CRLF&gt;</td>
</tr>
<tr>
<td>355</td>
<td>Octet-offset is the transaction offset</td>
</tr>
<tr>
<td>421</td>
<td>Domain service not available, closing transmission channel</td>
</tr>
<tr>
<td>432</td>
<td>A password transition is needed</td>
</tr>
<tr>
<td>450</td>
<td>Requested mail action not taken; mailbox unavailable; ATRN request refused</td>
</tr>
<tr>
<td>451</td>
<td>Requested action aborted: local error in processing; unable to process ATRN request now</td>
</tr>
<tr>
<td>452</td>
<td>Requested action not taken: insufficient system storage</td>
</tr>
<tr>
<td>453</td>
<td>You have no mail</td>
</tr>
<tr>
<td>454</td>
<td>TLS not available due to temporary reason; encryption required for requested authentication mechanism</td>
</tr>
<tr>
<td>458</td>
<td>Unable to queue messages for node node</td>
</tr>
<tr>
<td>459</td>
<td>Node node not allowed: reason</td>
</tr>
<tr>
<td>500</td>
<td>Command not recognized; command. Syntax error</td>
</tr>
<tr>
<td>501</td>
<td>Syntax error in parameters or arguments; no parameters allowed</td>
</tr>
<tr>
<td>502</td>
<td>Command not implemented</td>
</tr>
<tr>
<td>503</td>
<td>Bad sequence of commands</td>
</tr>
<tr>
<td>504</td>
<td>Command parameter temporarily not implemented</td>
</tr>
<tr>
<td>521</td>
<td>Machine does not accept mail</td>
</tr>
<tr>
<td>530</td>
<td>Must issue a STARTTLS command first; encryption required for requested authentication mechanism</td>
</tr>
<tr>
<td>534</td>
<td>Authentication mechanism is too weak</td>
</tr>
<tr>
<td>538</td>
<td>Encryption required for requested authentication mechanism</td>
</tr>
<tr>
<td>550</td>
<td>Requested action not taken (command is not executed); mailbox unavailable</td>
</tr>
<tr>
<td>551</td>
<td>User not local; please try forwardpath</td>
</tr>
<tr>
<td>552</td>
<td>Requested mail action aborted: exceeded storage allocation</td>
</tr>
<tr>
<td>553</td>
<td>Requested action not taken: mailbox name not allowed</td>
</tr>
<tr>
<td>554</td>
<td>Transaction failed</td>
</tr>
</tbody>
</table>

Note: From “SMTP Specifications,” 2005.
SMTP must either encode the message contents into a 7-bit representation compliant with the MIME standard or return a permanent error to the user.

The SMTP service extension has the limitation on maximum length of a line (only up to 1,000 characters as required by the RFC 821 standard). The service extension also limits the use of non-ASCII characters to message headers, which are prohibited by the RFC 822 regulations.

The RFC 1653 document introduces the protocol for message size declaration that allows a server to inform a client of the maximum size message it can accept. If both server and client support the message size declaration extension, the client may declare an estimated size of the message to be transferred, and the server will return an error if the message is too large. Each of these SMTP service extensions is a draft standard protocol and each has a status of elective.

The MIME protocols define five header lines that can be added to the original header section to define the transformation parameters: MIME-version, content-type, content-transfer-encoding, content-id, and content-description. Each header line is described in detail in the following sections.

**MIME-Version**

The header line `MIME-Version: 1.1` declares that the message was composed using the (current) version 1.1 of the MIME protocol.

**Content-Type**

The header line `Content-Type: <type/subtype; parameters>` defines the type of data used in the body of the message. The identifiers of the content type and the content subtype are separated by a slash. Depending on the subtype, the header may contain other parameters. The MIME standard allows seven basic content types of data, the valid subtypes for each, and transfer encodings, which are listed in Table 3. Examples of the content-type headers can be found in Forouzan (2003).

**Content-Transfer-Encoding**

The `Content-Transfer-Encoding: <type>` header line defines the method to encode the messages into a bit-stream of 0s and 1s for transport. The five types of encoding are as follows:

- **7bit**—for NVT ASCII characters and short lines of less than 1,000 characters.
- **8bit**—for non-ASCII characters and short lines of less than 1,000 characters; the underlying SMTP protocol must be able to transfer 8-bit non-ASCII characters (this type is not recommended).
- **binary**—for non-ASCII characters with unlimited-length lines; this is 8-bit encoding. The underlying SMTP protocol must be able to transfer 8-bit non-ASCII characters (this type is not recommended).
- **base64**—for sending data made of bytes when the highest bit is not necessarily zero; 6-bit blocks of data are encoded into 8-bit printable ASCII characters (for details, see Forouzan, 2003; Stevens, 1993), which can then be sent as any type of character set supported by the underlying mail transfer mechanism.
- **quoted-printable**—for sending data that consist of mostly ASCII characters with a small non-ASCII portion; if a character is not ASCII, it is sent as three characters: the first character is the equal sign, and the next two are the hexadecimal representation of the byte.

Although the content type and encoding are independent, the RFC 1521 document recommends `quoted-printable` for text with non-ASCII data, and `base64` for image, audio, video, and octet-stream application data. This allows maximum interoperability with RFC 821 conformant MTAs (Stevens, 1993).
MAIL ACCESS PROTOCOLS

MAIL TRANSMISSION TYPES

The SMTP (RFC821) and the Standard for the Format of Advanced Research Project Agency (ARPA) Internet Text Messages (RFC822) specify that a set of "Received" lines will be prepended to the headers of electronic mail messages that are transported through the Internet (Mail Parameters, 2002). The received line may optionally include either or both a "via" phrase or a "with" phrase (or both). The legal value for the "via" phrase is intended to indicate the link or physical medium over which the message was transferred (e.g., the UDP link type should be specified for the Unix-to-Unix Copy Program). The "with" phrase is intended to indicate the protocol or logical process that has been used to transfer the message (e.g., SMTP or ESMTTP parameters are used respectively for SMTP [RFC821] or SMTP with service extensions [RFC1869] protocol types).

MAIL ACCESS MODES

To reach its final destination, an e-mail message should be handled by a mail server, the mail access protocol, and the mail client. A general concept of how these components work together is described in "Accessing Your Mail" (1997).

An Internet mail server (known as the mail transfer agent, described earlier) is the software responsible for transmitting and receiving e-mail across the Internet. The MTA software is run on a computer that has a connection to the Internet and is managed, monitored, and backed up by ISPs or a company's information services staff. Some mail servers store mail only until the user retrieves it, whereas others store user mail permanently. An e-mail user typically uses a mail client program to interact with the mail server (Rose, 1993).

A mail client (known as the mail user agent, described earlier) is the software that a user employs to read, send, file, and otherwise process the electronic mail. Usually running on a user's desktop computer, the mail client also manages related e-mail data (address books, spelling dictionaries, and stationery). The mail client connects to a mail server to retrieve new mail. Some mail clients also use the mail server to store all e-mail (Rose, 1993).

The communication between the mail client and mail server is regulated by the mail access protocol, a standardized set of transmitted commands and responses sent over many different types of network connections. The protocol commands (created for managing access to the Internet e-mail only) depend on a design approach that can significantly affect the manner, modes, characteristics, and capabilities of the interaction between the mail client and mail server ("Accessing Your Mail," 1997). The SMTP Protocol handles the task of the actual sending of e-mail on the Internet.

A mail access protocol operates in three common modes that differ in where and how a user stores and processes his or her mail ("Accessing Your Mail," 1997):

- Offline mode—e-mail is downloaded from a temporary storage on the mail server to the user's computer. After download, the mail is deleted from the server.
- Online mode—user's e-mail, his or her inbox, and all filed mail remains permanently on the mail server. By connecting to the server and establishing an e-mail session, the user can download a temporary copy of his or her e-mail and read it, or send e-mail. Once the connection is finished, the copy is erased from the user's computer, and only the original remains on the server.
- Disconnected/reynchronization mode—combines both offline and online modes. A copy of the user's e-mail is downloaded to his or her computer(s), and the original message remains on the mail server. The user can change a local copy of his or her e-mail on any computer, then resynchronize all copies, including the original e-mail message on the server and copies on additional computers.

All three modes offer multiprotocol support. This includes support for existing platforms such as UNIX, Microsoft Windows, and Apple Macintosh, and future platforms such as Java Mail Service-based network computers. All three modes, including their advantages and disadvantages, are discussed in detail in "Accessing Your Mail" (1997).

MAIL ACCESS PROTOCOLS

POP3

POP is used on the Internet to retrieve e-mail from a mail server. There are two versions of POP. The first, known as POP2 (RFC 937), became a standard in the mid-1980s and requires SMTP to send messages. Nowadays it has a
status of “not recommended.” The newer version, POP3 (RFC 1725), can be used with or without SMTP.

POP was designed primarily to support the offline access mode (RFC 1323). Typically, e-mail arrives from the user's mailbox on the server to the user's computer. POP is designed so that mail client software can determine which messages have been previously downloaded from the server. The mail client can then download only new messages. POP also provides the ability to selectively delete messages from the server. It can be used by a mail client to perform basic resynchronization of the mailbox on the server and on the user's computers. The client can leave the most recent messages on the server after they have been downloaded. These messages can then be downloaded a second time to a second computer. Additionally, some POP implementations provide optional features, such as allowing users to download only headers at one session, to review the topics, and then download selected bodies and attachments in a subsequent session to minimize connection times over slow links (“Accessing Your Mail,” 1997).

POP servers are widely available both commercially and as freeware on a number of operating systems. Moreover, there are almost no interoperability issues between POP servers and mail clients, and users can use any POP mail client with any POP server. All ISPs support and use POP.

In the end-to-end application related to SMTP, the server must be available whenever a client (sender) transmits mail. If the SMTP server resides on an end-user PC or workstation, that computer must be running the server when the client is trying to send mail. For some operating systems (e.g., when a server program is activated on the VM SMTP service virtual machine or the MAIL program on DOS), the server becomes unavailable and unreachable by the SMTP client (SMTP, 2005). The mail-sending process will fail in these cases. Especially, it is important for single-user systems that the client has an accessible mailbox on various types of server (RFC 1725).

One of the simplest approaches to resolve this problem is to allow the end user to run a client program, which communicates with a server program on a host. This server program acts as both a sender and a receiver SMTP (SMTP, 2005). Here the end-user mailbox resides on the server, and the server system is capable of sending mail to other users.

In another approach, the SMTP server function has to be off-loaded from the end-user workstation, but not the SMTP client function. In this case, the user has a mailbox that resides on a server system, and he can send mail directly from the workstation. To collect mail from the mailbox, the user must connect to the mail server system.

The current post office protocol version 3 (RFC 1725) is a draft standard protocol, and its status is effective. POP3 extensions are described in RFC 2449. POP3 security options are introduced in RFC 2595. The RFC 1734 describes the optional AUTH command for indicating an authentication mechanism to the POP3 server, performing an authentication protocol exchange, and optionally negotiating a protection mechanism for subsequent protocol interactions (Sheldon, 2001).

**IMAP4**

IMAP is a protocol for retrieving e-mail messages (RFC 1066). The IMAP4 version is similar to POP3 but supports some additional features. For example, with IMAP4, the user can search through his or her e-mail messages for key words while the messages are still on the mail server. The user can then choose which messages to download to his or her machine.

IMAP uses SMTP as its transport mechanism. Following the simple analogy (Sheldon, 2001), IMAP servers are like post offices, whereas SMTP is like the postal carriers. IMAP uses TCP to take advantage of its reliable data delivery services, which are allocated on the TCP port 143. The latest IMAP version 4, revision 1 (IMAP4rev1) is defined in RFC 2060. IMAP has many advanced features, such as the ability to address mail not by arrival number, but by using attributes (e.g., “Download the latest message from Smith”). This feature allows the mailbox to be structured more like a relational database system rather than a sequence of messages (Tanenbaum, 2003).

Authentication mechanisms are described in RFC 1731. Security issues have been introduced in “IMAP/POP Authorization for Simple Challenge/Response” (RFC 2156), “IMAP4 Login Referrals” (RFC 2221), and “IMAP4 Implementation and Best Practices” (RFC 2683).

**SMTP SECURITY ISSUES**

**SMTP Vulnerabilities**

The processes of retrieving e-mail from servers and managing data communication through the Internet are vulnerable to various attacks. A review of vulnerabilities can be found in “Vulnerability Tutorials” (2005) or SMTP Vulnerability Tutorials and other references. The Common Vulnerabilities and Exposures (CVE) organization provides a list of standardized names for SMTP vulnerabilities and other information security exposures. All CVE references (CVE entries and CAN candidates) cited in this text can be found at the CVE Web site, provided in the references (CVE, 2005).

Summaries of major SMTP vulnerability problems are given in Table 4.

A security audit of selected SMTP problems has been provided by the U.S. Computer Emergency Readiness Team (CERT) Coordination Center operated by Carnegie Mellon University, and E-Soft. Details about information about vulnerability problems, possible actions of an attacker or spammers, recommendations for downloading updated versions of software, examples of code modification, and test results can be found on the CERT (2005) and Security Space (“SMTP Problems,” 2005) Web sites.

The vulnerability problems can be grouped into several general high-risk categories: buffer overflow; redirection attacks through the firewall; bounced "piping" attacks; and host-shell-gaining attacks (see Table 4).

The medium-to-high risk category includes denial-of-service attacks. Low-medium-risk categories include mail relaying on the remote SMTP server, mail-queue manipulation attacks; debug-mode-leak category; and crashing antivirus-software attack (“SMTP Problems,” 2005). Most SMTP-specific vulnerabilities occur from...
<table>
<thead>
<tr>
<th>CVE Name</th>
<th>Type of Vulnerability</th>
<th>Possible Attacker Intrusive Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE-2004-1049</td>
<td>Stack-based buffer overflow in the SMTP service support in zmsm.exe in Zone Labs ZoneAlarm before v. 4.5.3.38, ZoneLabs Integrity client v. 4.0.</td>
<td>It allows remote attackers to execute arbitrary code via a long RCPT TO argument.</td>
</tr>
<tr>
<td>CVE-2002-309</td>
<td>SMTP proxy in Symantec Enterprise Firewall v. 6.5.x includes the firewall's physical interface name and address in an SMTP exchange when NAT translation is made to an address other than the firewall.</td>
<td>It allows remote attackers to determine certain firewall configuration information.</td>
</tr>
<tr>
<td>CVE-2002-0055</td>
<td>SMTP service in Microsoft Windows 2000, Windows XP Professional, and Exchange 2000 to cause a DoS via a command with a malformed data transfer (BDAT) request.</td>
<td>An attacker may disrupt the SMTP service and, depending on the system configuration, potentially SS and other Internet services as well. See also MS02-012.</td>
</tr>
<tr>
<td>CVE-2002-0054</td>
<td>SMTP service in Microsoft Windows 2000 and Internet Mail Connector (IMC) in Exchange Server 5.5 does not properly handle responses to NTLM authentication.</td>
<td>It allows remote attackers to perform mail relaying via an SMTP AUTH command using null session credentials.</td>
</tr>
<tr>
<td>CVE-2001-0894</td>
<td>Vulnerability in Postfix SMTP server that is configured to e-mail the postmaster: SMTP errors cause the session to terminate.</td>
<td>It allows remote attackers to cause a DoS (memory exhaustion) by generating a large number of SMTP errors, which forces the SMTP session log to grow too large.</td>
</tr>
<tr>
<td>CVE-2001-0692</td>
<td>Vulnerability in SMTP proxy in WatchGuard Firebox (2500 and 4500) v. 4.5.4-6.</td>
<td>A remote attacker may bypass firewall filtering via a base64 MIME encoded e-mail attachment whose boundary name ends in two dashes.</td>
</tr>
<tr>
<td>CVE-2001-0653</td>
<td>Format string vulnerability in Exim (v. 3.22-10 in Red Hat, v. 3.16 in Debian, and v. 3.16 in Conectiva) in batched SMTP mode.</td>
<td>It allows a remote attacker to execute arbitrary code via format strings in SMTP mail headers.</td>
</tr>
<tr>
<td>CVE-2001-0690</td>
<td>Local buffer overflow on Sendmail (v.8.11.x).</td>
<td>A local user may gain root privileges.</td>
</tr>
<tr>
<td>CVE-2001-0634</td>
<td>The authentication error on the remote SMTP server</td>
<td>An attacker may exploit this flaw to use the SMTP server as a spam relay.</td>
</tr>
<tr>
<td>CVE-2001-1203</td>
<td>Lotus Domino SMTP server (v. 4.63-5.08) is vulnerable to a DoS (central processing unit consumption) attack by forging an e-mail message with the sender as bounce[@127.0.0.1] (localhost).</td>
<td>It allows remote attackers to cause a DoS: the server enters a mail loop.</td>
</tr>
<tr>
<td>CVE-2000-1047</td>
<td>The Lotus Domino SMTP server (v.5.0.5.x) is vulnerable to buffer overflow when supplied a too long ENVID variable within a MAIL FROM command.</td>
<td>An attacker may use this flaw to prevent Domino services from working properly, or to execute arbitrary code on the host.</td>
</tr>
<tr>
<td>CVE-2000-1022</td>
<td>The mailguard feature in Cisco Secure PIX Firewall (v. 5.2.1) does not properly restrict access to SMTP commands.</td>
<td>It allows remote attackers to execute restricted commands by sending a DATA command before sending the restricted commands.</td>
</tr>
<tr>
<td>CVE-2000-0507</td>
<td>The remote Imate SMTP server crashes when it is issued a HELO command with an argument longer than 1,200 characters.</td>
<td>An attacker may shut down the SMTP server.</td>
</tr>
<tr>
<td>CVE-2000-0488</td>
<td>Buffer overflow on the ITHouse mail server (v.1.04).</td>
<td>Remote attackers may execute arbitrary commands via a long RCPT TO mail command.</td>
</tr>
<tr>
<td>CVE-2000-0435</td>
<td>Buffer overflow in the remote Lotus SMTP server when the server is issued a too long argument to the MAIL FROM command.</td>
<td>An attacker may prevent the host from acting as a mail host and may execute arbitrary code on the system.</td>
</tr>
</tbody>
</table>
| CVE-2000-0319  | mail.local in the remote Sendmail server does not properly identify the 
              | A remote attacker may cause a DoS or corrupt mailboxes via a message line that is 2047 characters long and ends as \n.                                                                                                                                 |

(Continued)
Along this pass, messages get “stamped” by the intermediate servers (including the sender’s and receiver’s servers) before reaching the destination host. This results in slowing down servers. To fix the problem, the origin of a spam e-mail should be identified. An e-mail message typically transports through a set of SMTP servers (including the sender’s and receiver’s servers) before reaching the destination host. Along this pass, messages get “stamped” by the intermediate SMTP servers. The stamps release tracking information that can be identified in the mail headers. Mismatches between the IP addresses and the domain names in the header could unveil the real source of spam mail. The real domain names that correspond to the indicated IP addresses can be found out by executing a reverse DNS lookup. Modern mail programs have incorporated this functionality, which generates a received: header line that includes the identity of the attacker (see examples in Campbell et al., 2003).

Antispoofing measures are under active development. Mail Abuse Prevention System (MAPS) and Open Relay Behavior-Modification System (ORBS) provide testing, reporting and cataloging of e-mail servers configured for SMTP relay. These organizations maintain real-time blackhole lists (RBL) of mail servers with problematic histories. For protection and security purposes, companies may configure their SMTP servers and other e-mail service systems in such manner that any mail coming from RBL-blacklisted mail servers is automatically rejected (Campbell, 2003). Other initiatives for restricting the sender address spoofing include SPF, Hotmail domain cookies, and Microsoft’s caller ID. Also see “E-Mail Threats and Vulnerabilities”.

### Table 4

<table>
<thead>
<tr>
<th>CVE Name</th>
<th>Type of Vulnerability</th>
<th>Possible Attacker Intrusive Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE-2000-0075</td>
<td>Super Mail Transfer Package, later called MsgCore, has a memory leak.</td>
<td>Remote attackers may cause a DoS by repeating multiple HELO, MAIL FROM, RCPT TO, and DATA commands in the same session.</td>
</tr>
<tr>
<td>CVE-1999-0203</td>
<td>The remote Sendmail’s SMTP server did not complain when issued the command (from piped program): MAIL FROM: testing</td>
<td>An attacker may send mail that will be bounced to a program that allows him to execute arbitrary commands on the host.</td>
</tr>
<tr>
<td>CVE-1999-0096</td>
<td>The remote Sendmail SMTP server seems to pipe mail sent to the “decode” alias to a program.</td>
<td>An attacker can use this “decode” flaw to overwrite arbitrary files on the remote server.</td>
</tr>
<tr>
<td>CAN-2003-0818</td>
<td>Multiple integer overflows in Microsoft ASN.1 library (MSASN1.DLL). See also MS04-007.</td>
<td>An attacker may execute arbitrary code on this host by sending a specially crafted ASN.1 encoded packet with improper lengths.</td>
</tr>
<tr>
<td>CAN-2003-0714</td>
<td>Exchange remote buffer overflow: SMTP service is vulnerable to a flaw in the XEXCH50 extended verb (command).</td>
<td>An attacker may gain root privileges.</td>
</tr>
<tr>
<td>CAN-2003-0681</td>
<td>Remote Sendmail servers (v. 8.12.9 and earlier) have prescan() overflow on a remote buffer.</td>
<td>An attacker may gain root privileges.</td>
</tr>
<tr>
<td>CAN-2003-0540</td>
<td>Remote Postfix (v. 1.1.12) daemon multiple vulnerabilities.</td>
<td>An attacker may remotely disable it, or use it as a DoS agent against arbitrary hosts.</td>
</tr>
<tr>
<td>CAN-2003-0264</td>
<td>SMTP (v. 5.1) SMTP server experiences various overflows.</td>
<td>A cracker might execute arbitrary commands on this host or to disable it remotely.</td>
</tr>
<tr>
<td>CAN-2003-0161</td>
<td>Sendmail (v. 8.12.8 and earlier) servers have buffer overflow due to type conversion.</td>
<td>An attacker may gain remotely root privileges.</td>
</tr>
<tr>
<td>CAN-2002-1137</td>
<td>Remote header buffer overflow on Sendmail servers (v. 8.12.7 and earlier).</td>
<td>A local attacker may regain the extra dropped privileges and run commands as root.</td>
</tr>
<tr>
<td>CAN-2001-0713</td>
<td>A user may supply a custom configuration file to remote Sendmail servers.</td>
<td></td>
</tr>
</tbody>
</table>

Note: The CAN number indicates a candidate for inclusion in the Common Vulnerabilities and Exposures (CVE) list of standard names for security problems. It must be reviewed by the CVE editorial board before it can be added to CVE (CVE, 2005).

DoS = denial of service; MIME = multipurpose internet mail extensions; NAT = network address translation; SMTP = Simple mail transfer protocol.
SMTP Server Buffer Overflow Vulnerability

Sendmail contains a buffer overflow in code that parses e-mail addresses. (CAN-2003-0161). When processing e-mail messages, sendmail creates tokens from address elements (user, host, domain). The code that performs this function (prescan() in parseaddr.c) contains logic to check that the tokens are not malformed or overly long. In certain cases, a variable in prescan() is set to the special control value -1, which may alter the program logic to skip the length checks. Using an e-mail message with a specially crafted address containing 0xFF, an attacker could execute arbitrary code or cause a denial of service on a vulnerable system. Upgraded versions of sendmail should be used for protection.

Another remote buffer overflow in sendmail was reported (CAN-2002-1337). This vulnerability may allow remote attackers to gain root privileges of the sendmail daemon. A properly patched sendmail server (version 8.12.8) will drop invalid headers, thus preventing down-stream servers from receiving them.

A buffer overflow in the mail server was identified as vulnerability in the Lotus Domino family of servers (Lotus, 2005) that includes an SMTP server (see Table 4, CVE-2000-0452). It supports extensions, which allow for the use of delivery-status notifications that provide information about the delivery status of an e-mail message to the sender. An e-mail client specifying an identifier for an address and an arbitrary recipient address, which greatly facilitates the spread of spam. Even SMTP servers, which generally do not allow relaying, do allow it if the session originates from a host in the server’s domain or from a host from which relaying is explicitly permitted. If the scan is performed from such a host, a false alarm may result. To resolve this issue, UNIX mail servers should be upgraded to the latest version of Sendmail, which does not allow relaying by default (Antirelay Parse, 2005).

Mail Relaying SMTP Vulnerability in Microsoft Windows 2000

A specific type of vulnerability in the default SMTP server running Microsoft Windows 2000 was discovered by Joo Gouwela (“Authentication Error”, 2001). An SMTP implementation is provided with Microsoft Windows 2000, and it is installed by default. Microsoft Exchange Server also includes an SMTP service, but the component that performs SMTP authentication is different from the basic SMTP Service in Windows 2000 and is not affected by the vulnerability. A flaw in the authentication process (CVE, 2001, No. 0504) used by the SMTP service that installs as part of Internet Information Services (IIS) could allow an unauthorized user to authenticate successfully to the service using incorrect credentials. An attacker can use this vulnerability to gain user-level privileges on the SMTP service, thereby enabling the attacker to use the service (e.g., to co-opt a server’s resources for mass mailings) but not to administer it. The service can be used by an attacker to perform SMTP mail relaying. There have been cases in which threatening e-mails were relayed to prevent the recipient from being able to trace where they came from. This vulnerability affects only stand-alone machines (e.g., Web servers), not domain members or Microsoft Exchange mail servers running Windows 2000.

Customers who need SMTP services should apply the patch (“Patch Availability,” 2005), which eliminates the vulnerability by ensuring that the SMTP service properly authenticates users before allowing them to levy requests on it. Also, proper firewalls could be used to prevent Internet users from exploiting the vulnerability. Recommendations for preventing the servers from relaying and spam can be found in Fugatt (2002, July 30).

Also see “Windows 2000 Security”.

Encapsulated SMTP Address Vulnerability

The security vulnerability in Microsoft Exchange Server 5.5 (CVE, 2002, No. 0504) could allow an attacker to perform mail relaying via an Exchange server that is configured to act as a gateway for other Exchange sites, using the Internet Messaging Service.

The vulnerability lies in the way that site-to-site relayings performed via SMTP. The SMTP server in Microsoft Windows 2000 and Internet Mail Connector in Exchange Server 5.5 does not properly handle responses to NTLM authentication, which allows remote attackers to perform mail relaying via an SMTP AUTH command using null session credentials. Encapsulated SMTP addresses could be used to send mail to any e-mail address. The method
of configuring the Exchange Internet Mail Service (IMS) (called Internet Mail Connector in prior versions of Exchange), is vulnerable to the attack. The IMS service provides encapsulated addresses, when used as a Site Connector, and uses a special form of addressing called “enveloped SMTP,” which is used to encapsulate various message types into SMTP addresses. The Exchange supports three kinds of Site Connectors: an X.400 connector, the Exchange Site Connector, and the Exchange Internet Mail Service. A malicious user could address e-mails using this format and route mail through an Exchange Server, even if mail relay has been disabled.

Any customer who has configured an IMS on an Internet-connected Exchange Server should consider installing the patch (“Patch Availability,” 2005) that eliminates the vulnerability.

**Malformed Request Denial of Service**

The SMTP service in Microsoft Windows 2000, Windows XP Professional, and Exchange 2000 is vulnerable to cause a denial of service via a command with a malformed data transfer (BDAT) request (CVE, 2002, No. 0055). By sending either a message with a corrupted timestamp or a malformed version of a particular SMTP command to the server, it is possible for a remote attacker to cause the mail service to crash and thus stop responding to legitimate requests.

**Extended Verb Request Handling Flaw**

IMS in Exchange Server 5.5 and Exchange 2000 do not require authentication before allowing a user to send a certain extended verb request. This vulnerability allows remote attackers to cause a denial of service (memory exhaustion) and to consume large amounts of memory by directly connecting to the SMTP service and possibly triggering a buffer overflow in Exchange 2000 (CVE, 2003, No. 0714). Command execution could be possible. The Microsoft Security Bulletin (2004, No. 03-046) recommends the patch to fix this vulnerability.

**Reverse DNS Response Buffer Overflow**

Microsoft Exchange does not check the length of the response from the DNS server before copying it into a fixed-length buffer (CVE, 2002, No. 0660). Therefore, a remote attacker who has control over a registered DNS server could cause a buffer overflow by creating a long, specially crafted reverse DNS entry and then issuing the smtp command to Exchange. The overflow would crash the server or even allow the attacker to execute arbitrary commands. Microsoft Exchange 5.5 is affected by this vulnerability if the patch has not been installed. At the same time, Microsoft Exchange 2000 is not affected because it runs atop the native Windows 2000 SMTP service rather than the Internet Mail Connector. To fix the reverse DNS problem, the patch (“Patch Availability,” 2005) should be applied.

**Firewall SMTP Filtering Vulnerability**

During expanded internal regression testing by Cisco, it was discovered that the Cisco Secure PIX Firewall feature “mailguard,” which limits SMTP commands to a specified minimum set of commands, can be bypassed (Cisco, 2001). The filtering command fixup protocol smtp(portnum), which is enabled by default on the Cisco Secure PIX Firewall, can fail. All users of Cisco Secure PIX Firewalls with software that provide access to SMTP Mail services are at risk. To exploit this vulnerability, attackers can make connections to an SMTP mail server (protected by the PIX Firewall) and can circumvent the expected filtering of the mailguard feature. If the mail server is not properly secured, an attacker may collect information about existing e-mail accounts and aliases or can execute arbitrary code on the mail server. Cisco has offered free software upgrades for all affected customers (Cisco, 2001).

**Spoofing**

On the Internet, mail is usually delivered directly from the sending host to the receiving host. This inherent “open” design of SMTP allows a host computer, which needs to deliver a message to another computer(s), to make a connection (or multiple connections) to some other SMTP server and ask that server to relay the message(s) on its behalf. Gateways can be used to bridge firewalls.

By denying access to a sending machine with a firewall, many companies and ISPs have been blocking the receipt of unwanted mail from known sources. The “blocked” senders of junk mail may attempt to deliver it through another computer by requesting the computer to route that mail for them. Senders of unsolicited e-mail can also use this method to hide their real identity by manipulating the headers in the message and then sending the message through client’s system for delivery to its final destination. This “spoofing” action gives the appearance that the message originated from the relaying server. When a bulk mailer chooses a client’s computer to deliver unsolicited mail to thousands of other people known as “spamming”), the client’s system immediately becomes busy delivering messages that did not originate with the client’s users.

The SMTP server may protect the client’s system against this type of abuse in two ways. First, the server allows administrators to configure the system to accept only mail originating from local users or destined for local users. Second, the server administrator can define systems from which the client never wants to receive mail. It blocks mail from known sources of spam mail (“Setting SMTP Security,” 2005). Also see “Networks Attacks.”

**Bounce Attack**

In the case of anonymous file transfer protocol (FTP) services, the attacker can instruct the FTP server to send a file to the SMTP service being attacked on the victim’s system (see “FTP Security Considerations, RFC 2577”). Using the FTP server to connect to the service on the attacked computer makes it difficult to track down the attacker (Campbell et al., 2003). Particularly, a client–attacker can upload a file that contains SMTP commands to an FTP server. Then, using an appropriate PORT command, the client instructs the attacked server to open a connection to a third computer’s SMTP port 25 and transfer the uploaded
file containing SMTP commands to the third computer. This action may allow the client attacker to forge mail on the third computer without making a direct connection.

**Restricting Access to an Outgoing Mail Server**

The access to an outgoing mail server can be restricted by verifying that the computer is on the ISP's local network. When the user dials the modem and connects to the ISP, his computer is given an IP address that identifies him as being a part of that network. If the user has two ISPs and dials up to one and then connects to the other's mail server, it may prevent him or her from relaying mail because the computer is not identified as being on the local network for the provider. In this case, the user should try to use the SMTP server to dial up and connect to the Internet ("What is SMTP Security?", 2005).

Another way to restrict access is to insist on a local domain return address. If users connect to the mail server for "domain.com," it may only allow them to send mail that is from "username@domain.com." Therefore, if they try to send mail from another account that has the return address of "username@anotherdomain.com," it may restrict them from relaying to another server ("What is SMTP Security?", 2005).

**Mail Encryption**

SMTP is not a secure protocol. Messages sent over the Internet are not secure unless some form of encryption is implemented. S/MIME is a widely used Internet e-mail standard. This and some other security topics (PGP, transport layer security (TLS), host-to-host encryption) are discussed in other chapters.

Also see "Encrypting E-Mail, PGP, S/MIME, TLS, and Virtual Private Networks (VPNs) Basics".

**Bastille Hardening System**

The Bastille Hardening System (Bastille Project, 2005) has been designed to "harden" or "tighten" UNIX-based operating systems. It currently supports the Red Hat Enterprise 3, Debian, Mandrake, SuSE, and TurboLinux Linux distributions along with HP-UX and Mac OS X. The Bastille Linux Hardening software (Version 2.1.2) is available from the Source Forge Web site (Bastille Linux Project, 2005) enhances the security of a Linux box by configuring daemons, system settings, and firewalling. Written in Perl, the Bastille Linux intends to improve Linux-based computer security. Among others, it has a revised sendmail module dedicated to secure holes that were discovered previously (see Table 4). A review of other service modules (Remote Access, Pluggable Authentication, DNS, Apache, FTP, SecureNet, File Permission, Patch Download, and Firewall Configuration (IPChains)) can be found in Raynal (2000).

**POP and IMAP VULNERABILITIES**

POP was designed to support offline mail processing (Rosen, 1993). The mail is deleted from the server and is handled offline (locally) on the client machine. In the implementation of this protocol on a UNIX system, the server must run with root privileges; therefore, it can access mail folders and undertake some file manipulation on behalf of the user logging in. After login, these privileges are discarded. Vulnerability exists in the way the login transaction is handled in some implementations of these procedures (CERT, 2005). This vulnerability can be exploited to gain privileged access on the server. By preparing carefully crafted test text to a system running a vulnerable version of the POP server, remote users may be able to cause a buffer overflow and execute arbitrary instructions with root privileges. They do not need access to an account on the system to do this. Vulnerable POP versions are identified in CVE, 2001, No. 0443, and "Vulnerability Tutorials," 2005.

POP servers allow non-UNIX users to access their mail on a machine without logging in. The servers give PC and Macintosh users a way to receive mail through another machine. When connecting to a POP server, the client transmits the users' userid and password in clear text. After authentication, users can access their mail. Each time the client reconnects to the POP server, the users' userid and password are transmitted. Some POP client programs check the server every few minutes to check for the arrival of new mail. These frequent checks increase the possibility of the machine, username, and password being discovered by a password sniffer "tuned" for POP mail systems.

This clear text password issue is resolved by using an optional command allowable for POP3 servers ( RFC 1725). When the initial connection is made to a POP server, the server displays a time stamp in its banner. The client uses this time stamp to create an MD5 hash string that is shared between the server and client. The next time the client reconnects to the POP server, the users' userid and password are transmitted. Some POP client programs check the server every few minutes to check for the arrival of new mail. Each time the client reconnects to the server, the user's userid and password are transmitted. Some POP client programs check the server every few minutes to check for the arrival of new mail. These frequent checks increase the possibility of the machine, username, and password being discovered by a password sniffer "tuned" for POP mail systems.

This clear text password issue is resolved by using an optional command allowable for POP3 servers (RFC 1725). When the initial connection is made to a POP server, the server displays a time stamp in its banner. The client uses this time stamp to create an MD5 hash string that is shared between the server and client. The next time the client reconnects to the POP server, the users' userid and password are transmitted. Some POP client programs check the server every few minutes to check for the arrival of new mail. These frequent checks increase the possibility of the machine, username, and password being discovered by a password sniffer "tuned" for POP mail systems.

This clear text password issue is resolved by using an optional command allowable for POP3 servers (RFC 1725). When the initial connection is made to a POP server, the server displays a time stamp in its banner. The client uses this time stamp to create an MD5 hash string that is shared between the server and client. The next time the client reconnects to the POP server, the users' userid and password are transmitted. Some POP client programs check the server every few minutes to check for the arrival of new mail. These frequent checks increase the possibility of the machine, username, and password being discovered by a password sniffer "tuned" for POP mail systems.

This clear text password issue is resolved by using an optional command allowable for POP3 servers (RFC 1725). When the initial connection is made to a POP server, the server displays a time stamp in its banner. The client uses this time stamp to create an MD5 hash string that is shared between the server and client. The next time the client reconnects to the POP server, the users' userid and password are transmitted. Some POP client programs check the server every few minutes to check for the arrival of new mail. These frequent checks increase the possibility of the machine, username, and password being discovered by a password sniffer "tuned" for POP mail systems.

This clear text password issue is resolved by using an optional command allowable for POP3 servers (RFC 1725). When the initial connection is made to a POP server, the server displays a time stamp in its banner. The client uses this time stamp to create an MD5 hash string that is shared between the server and client. The next time the client reconnects to the POP server, the users' userid and password are transmitted. Some POP client programs check the server every few minutes to check for the arrival of new mail. These frequent checks increase the possibility of the machine, username, and password being discovered by a password sniffer "tuned" for POP mail systems.

This clear text password issue is resolved by using an optional command allowable for POP3 servers (RFC 1725). When the initial connection is made to a POP server, the server displays a time stamp in its banner. The client uses this time stamp to create an MD5 hash string that is shared between the server and client. The next time the client reconnects to the POP server, the users' userid and password are transmitted. Some POP client programs check the server every few minutes to check for the arrival of new mail. These frequent checks increase the possibility of the machine, username, and password being discovered by a password sniffer "tuned" for POP mail systems.

This clear text password issue is resolved by using an optional command allowable for POP3 servers (RFC 1725). When the initial connection is made to a POP server, the server displays a time stamp in its banner. The client uses this time stamp to create an MD5 hash string that is shared between the server and client. The next time the client reconnects to the POP server, the users' userid and password are transmitted. Some POP client programs check the server every few minutes to check for the arrival of new mail. These frequent checks increase the possibility of the machine, username, and password being discovered by a password sniffer "tuned" for POP mail systems.

This clear text password issue is resolved by using an optional command allowable for POP3 servers (RFC 1725). When the initial connection is made to a POP server, the server displays a time stamp in its banner. The client uses this time stamp to create an MD5 hash string that is shared between the server and client. The next time the client reconnects to the POP server, the users' userid and password are transmitted. Some POP client programs check the server every few minutes to check for the arrival of new mail. These frequent checks increase the possibility of the machine, username, and password being discovered by a password sniffer "tuned" for POP mail systems.

This clear text password issue is resolved by using an optional command allowable for POP3 servers (RFC 1725). When the initial connection is made to a POP server, the server displays a time stamp in its banner. The client uses this time stamp to create an MD5 hash string that is shared between the server and client. The next time the client reconnects to the POP server, the users' userid and password are transmitted. Some POP client programs check the server every few minutes to check for the arrival of new mail. These frequent checks increase the possibility of the machine, username, and password being discovered by a password sniffer "tuned" for POP mail systems.

This clear text password issue is resolved by using an optional command allowable for POP3 servers (RFC 1725). When the initial connection is made to a POP server, the server displays a time stamp in its banner. The client uses this time stamp to create an MD5 hash string that is shared between the server and client. The next time the client reconnects to the POP server, the users' userid and password are transmitted. Some POP client programs check the server every few minutes to check for the arrival of new mail. These frequent checks increase the possibility of the machine, username, and password being discovered by a password sniffer "tuned" for POP mail systems.

This clear text password issue is resolved by using an optional command allowable for POP3 servers (RFC 1725). When the initial connection is made to a POP server, the server displays a time stamp in its banner. The client uses this time stamp to create an MD5 hash string that is shared between the server and client. The next time the client reconnects to the POP server, the users' userid and password are transmitted. Some POP client programs check the server every few minutes to check for the arrival of new mail. These frequent checks increase the possibility of the machine, username, and password being discovered by a password sniffer "tuned" for POP mail systems.
A valid account name and password would be required to exploit this vulnerability. The second vulnerability is a buffer overflow in the processing of the user's login name (CVE, 2003, No. 0143). By supplying a name longer than 63 characters, a remote attacker could crash the service or execute arbitrary commands. The third vulnerability (CVE, 2003, No. 0143) is in the qvmpoint function, which is QPOP's own implementation of the vmpoint function. A buffer overflow occurs as a result of a failure to add a terminating null byte, when creating long strings during subsequent calls to the qvmpoint function, and allowing the execution of commands. Recommendations for resolving these issues can be found in "Common Vulnerability Tutorials," 2005). Secure versions of POP3 (RFC 2449) and IMAP4 (RFC 2950) that use the public key encryption mechanism (Tanenbaum, 2003) are also available.

STANDARDS, ORGANIZATIONS, AND ASSOCIATIONS

Internet Assigned Numbers Authority

The IANA (2005) provides the central coordinating functions of the global Internet for the public needs. The IANA organization maintains a registry of the following services:

- Domain name services
- Database of indexes by Top-Level Domains code
- "Whois" service of domain name recognition
- IP address assignment services (for both IPv4 and IPv6)
- Protocol number assignment services

Internet Engineering Task Force

Working Groups

Internet electronic mail was originally defined in the RFC821 standard as a part of the IETF project. Since August 1982, e-mail standards declared in this document were updated and revised by the IETF Detailed Revision/Update of Message Standards (DRUMS) Working Group. The group is also searching new directions in the electronic message communication through the Internet. The latest SMTP documents (including RFCs) can be found on the DRUMS Web site (IETF DRUMS, 2005).

The IETF Message Tracking Protocol (MSGTRK) Working Group is designing diagnostic protocols that a sender can use to request information from servers about the submission, transport, and delivery of a message, regardless of its status. The "Delivery by SMTP Service Extension" document (RFC 2852) specifies extensions to define message delivery time for making a decision to drop the message if it is not delivered within a specific time period. For diagnostic purposes, the "diagnostic-type" parameter (e.g., smtp for the Internet Mail) is defined for use in the SMTP delivery status notification (see RFC1891).

The IETF S/MIME Mail Security (SMIME) Working Group is developing S/MIME security standards. The latest S/MIME documents (including RFCs) can be found on the SMIME Web site (IETF SMIME, 2005).

Internet Mail Consortium

The Internet Mail Consortium Web site (IMC, 2005) publishes a complete list of electronic mail-related requests for comments documents (RFCs).

Mitre Corporation

The Mitre Corporation publishes a list of standardized names for all publicly known vulnerabilities and security exposures known as Common Vulnerabilities and Exposures (CVE, 2005).

CONCLUSION

SMTP is an application protocol from the TCP/IP protocol suite that enables the support of e-mail on the Internet. Mail is sent by a series of request-response transactions between a client and a server. The transactions pass the message, which is composed of header and body, and the envelope (SMTP source and destination addresses). The header contains the mail address(es), which consists of two parts: a local address (also known as a "user mailbox") and a domain name. Both SMTP client and SMTP server require a user agent (UA) and a mail transfer agent (MTA). The MTA function is transferring the mail across the Internet. The command-response mechanism is used by SMTP to transfer messages between an MTA client and an MTA server in three stages: connection establishment, mail transfer, and connection termination. The envelope is transmitted separately from the message itself using the RCPT and DATA commands. IMIME, which is an extension of SMTP, allows the transfer of non-ASCII (multimedia) messages. POP3 and the IMAP 4 together with SMTP are used to receive mail by a mail server and hold it for hosts. The SMTPs lack of security is a problem for businesses. The security in the SMTP transactions can be supported by SMIME and other methods described in this chapter. Vulnerabilities of SMTP, POP, and IMAP servers (buffer overflow, mail relaying, spoofing, and other attacks) have been analyzed.

GLOSSARY

Body

The text of an e-mail message. The body of a message follows the header information.

Bounce Attack

An attack that uses a third party's FTP server to hide the true source of the attack from the victim.

Client

Any application program used to retrieve information from a server. Internet clients include World Wide Web browsers, Usenet newsreaders, and e-mail programs.

Client-Server

The relationship between two application programs. One program, the server, is responsible for servicing requests from the other program, the client.

Delivery Status Notification (DSN)

An extended SMTP service that provides information about the delivery status of an e-mail message to the sender.

Disconnected-Resynchronization Mode

A mail-access mode in which mail is synchronized between a server and a client computer. By synchronizing mail
on the server, users can access their own mail from any computer that has access to the server where the mail is stored.

Domain Name System (DNS) A behind-the-scenes Internet service that translates Internet domain names to their corresponding IP addresses, and vice versa.

E-Mail Client An application that runs on a personal computer or workstation and enables the sender to send, receive, and organize e-mail. It is called a client because e-mail systems are based on a client–server architecture. Mail is sent from many clients to a central server, which reroutes the mail to its intended destination.

Encapsulated Address This address provides a way to send the e-mail to a site acting as a gateway for another site while indicating the server to which the message eventually needs to be sent. An encapsulated address consists of an address within an address; the outer address directs the mail to the gateway, which uses the inner address to determine where to send the e-mail. Because the Exchange Internet Mail Service (IMS) uses SMTP as its e-mail protocol, mail is sent to an IMS server using encapsulated SMTP as its addressing scheme.

Gateway Software that translates data from the standards of one system to the standards of another. For example, a gateway might exchange and convert Internet e-mail to X.400 e-mail.

Header Part of an e-mail message that precedes the body of the message and provides the message originator, date, and time.

Internet Message Access Protocol (IMAP) An Internet protocol used by mail clients for retrieving e-mail messages stored on servers. The latest version, IMAP4, is similar to POP3 but supports some additional features; for example, a user can search through his e-mail messages for key words while the messages are still on the server. IMAP-based applications can operate in offline mode, they typically operate in online or disconnected-resynchronization mode.

Mail Access Protocol A standardized set of commands and responses responsible for communication between the mail client and mail server.

Mail Client The software used to read, file, send, and otherwise process e-mail, typically running on a user's desktop computer.

Mailbox A file where e-mail messages are stored.

Mail Delaying A legitimate practice in which e-mail is routed to an intermediate mail server, which then delivers it to the recipient's mail server. For example, a company can have several servers and one of them is designated as a mail gateway to the Internet. Any e-mail sent to the company would arrive at the gateway server and then be relayed to the appropriate server for delivery to the recipient. Malicious users sometimes attempt to perform unauthorized mail relaying.

Mail Server A computer typically managed by an ISP or information services department that handles receipt and delivery of e-mail messages. It also may store mail for the user on a temporary or permanent basis.

Multipurpose Internet Mail Extensions (MIME) An Internet standard that provides the transfer of non-text information, such as sounds and graphics, and non-U.S. English (such as Cyrillic, Chinese, or Japanese) via e-mail.

Mail Transfer Agent (MTA) The software that is running on a mail server that relays, and delivers mail.

Mail User Agent (MUA) The software (also known as the mail client) used to read, file, send, and process e-mail, typically running on a desktop computer.

On-Demand Mail Relay (ODMR) A restricted profile of SMTP described in RFC 2645.

Post Office Protocol (POP) A protocol used to retrieve e-mail from a mail server in offline mode. A mail client that implements the POP protocol downloads all new mail from a mail server, terminates the network connection, and processes all mail offline at the client computer. The current version, POP3 can be used with or without SMTP.

Port In a software device, a port is a specific memory address that is mapped to a virtual networking cable. Ports allow multiple types of traffic to be transmitted to a single IP address. SMTP traditionally uses port 25 for e-mail communication.

Server A host computer that provides resources to client computers.

Simple Mail Transfer Protocol (SMTP) A protocol widely used to exchange e-mail between mail servers over the Internet.

Span Undesired junk e-mail or junk postings offering dubious business deals.

User Agent (UA) An SMTP component that prepares the message, creates the envelope, and puts the message in the envelope.

CROSS REFERENCES
See E-Mail and Instant Messaging, Internet E-Mail Architecture, Network Attacks, PGP (Pretty Good Privacy), S/MIME (Secure MIME).

REFERENCES
FURTHER READING

