

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/273830243>


SMTP (Simple Mail Transfer Protocol)

Chapter · December 2007
DOI: 10.1002/9781118256114.ch26

CITATIONS
8

READS
20,918

1 author:




Vladimir Riabov


Rivier University

203 PUBLICATIONS 698 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

- 

Modern Networking Technologies [View project](#)
- 

Effective Algorithms [View project](#)

SMTP (Simple Mail Transfer Protocol)

Vladimir V. Riabov, Ph.D., *Rivier College*

Introduction	1	Content-Transfer-Encoding	11
SMTP Fundamentals	1	Content-Id	12
SMTP Model and Protocol	2	Content-Description	12
User Agent	3	Security Scheme for MIME	12
Sending E-Mail	3	Mail Transmission Types	13
Mail Header Format	3	Mail Access Modes	13
Receiving E-Mail	5	Mail Access Protocols	14
The SMTP Destination Address	5	POP3	14
Delayed Delivery	5	IMAP4	14
Aliases	5	SMTP Vulnerabilities	15
Mail Transfer Agent (MTA)	5	Standards, Organizations, and Associations	15
SMTP Mail Transaction Flow	5	Internet Assigned Numbers Authority	15
SMTP Commands	7	Internet Engineering Task Force Working Groups	16
Mail Service Types	7	Internet Mail Consortium	16
SMTP Service Extensions	7	Mitre Corporation	16
SMTP Responses	9	Conclusion	16
SMTP Server	9	Glossary	16
On-Demand Mail Relay	9	Cross References	17
Multipurpose Internet Mail Extensions (MIME)	9	References	17
MIME-Version	11	Further Reading	19
Content-Type	11		

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 mail delivery is a two-stage process that provides for mail instances when the network connection or the remote machine has failed (Comer 2005). For example, when a user does not have a permanent Internet connection, it should have a mailbox on a computer that does have such a connection. That computer must run the SMTP server and be able to always receive incoming mail. The mail messages can then be retrieved by the user from the mailbox with an e-mail client (installed on the user's machine) using either post office protocol (POP) or Internet message access protocol (IMAP). The computer with the permanent mailbox must run two servers: SMTP for accepting mail and POP/IMAP for retrieving mail. 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, multipurpose 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, some vulnerability and security issues, standards, associations, and organizations.

SMTP FUNDAMENTALS

SMTP, an application layer protocol, is used as the common mechanism for transporting electronic mail among different hosts within the transmission control protocol/Internet protocol (TCP/IP) suite. The history of SMTP has been described by Kozierok (2006). 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 2006). When the TCP connection is successful, the two processes execute a simple request-response dialogue, defined by the SMTP protocol (see RFC 2821 and 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 message. The message must contain a message header and message text ("body") formatted in accordance with RFC 2822 and RFC 822.

In February 1993, the SMTP Service Extensions standard (RFC 1425), which describes a process for adding new capabilities to extend how SMTP works while maintaining backward-compatibility with existing systems, was published. The extended SMTP (ESMTP) standard (RFC 1425) was revised in RFC 1651 in July 1994 and then RFC 1869 in November 1995. Particular SMTP extensions,

such as message size declaration (RFC 1653 and RFC 1870), authentication (RFC 2554), and pipelining (RFC 2920), were defined later. In April 2001, revisions of RFC 821 and RFC 822 were published, as RFCs 2821 and 2822 respectively. The current base standard protocol for SMTP (RFC 2821) incorporates the base protocol description (RFC 821) and the latest SMTP extensions (RFC 1869) and updates the description of the e-mail communication model to reflect changes of TCP/IP networks, especially the e-mail features built into the domain name system (DNS) (Kozierok 2006).

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 to download mail that is stored on the local server. The delivery of e-mail to a user's mailbox typically takes place via a mail delivery agent (MDA). The MDA software accepts incoming e-mail messages and distributes them to recipients' individual mailboxes (if the destination account is on the local machine), or forwards back to an SMTP server (if the destination is on a remote server) (Wikipedia 2006). On UNIX systems, /bin/mail is the most popular MDA. Many mail transfer agents (MTAs) have basic MDA functionality built in, but a dedicated MDA like procmail can provide more sophistication.

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?", 2006). The MIME protocol defines one way files can be attached to SMTP messages. Microsoft Outlook and Netscape/Mozilla Communicator are some popular mail-agent programs on Window-based systems. The other functional and capable method of file attachment includes uuencode and uudecode techniques that are no longer in widespread use.

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?" 2006).

SMTP MODEL AND 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 being transmitted until it has been successfully copied to the recipient's SMTP server queue. This is different from the store-and-forward principle that is common in many other electronic mailing systems, in which 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] 2004).

The RFC 821 standard defines a client-server protocol. The client SMTP is the one, which initiates the session (that is, the sending SMTP) and the server is the one that responds (the receiving SMTP) to the session request. Because the client SMTP frequently acts as a server for a user-mailing program, however, it is often simpler to refer to the client as the sender-SMTP and to the server as the receiver-SMTP.

An SMTP-based process can transfer electronic mail to another process on the same network or to another network via a relay or gateway process accessible to both networks (Sheldon 2001). An e-mail message may pass through a number of intermediate relay or gateway hosts on its path from a sender to a recipient. A simple model of the components of the SMTP system is shown in Figure 1.

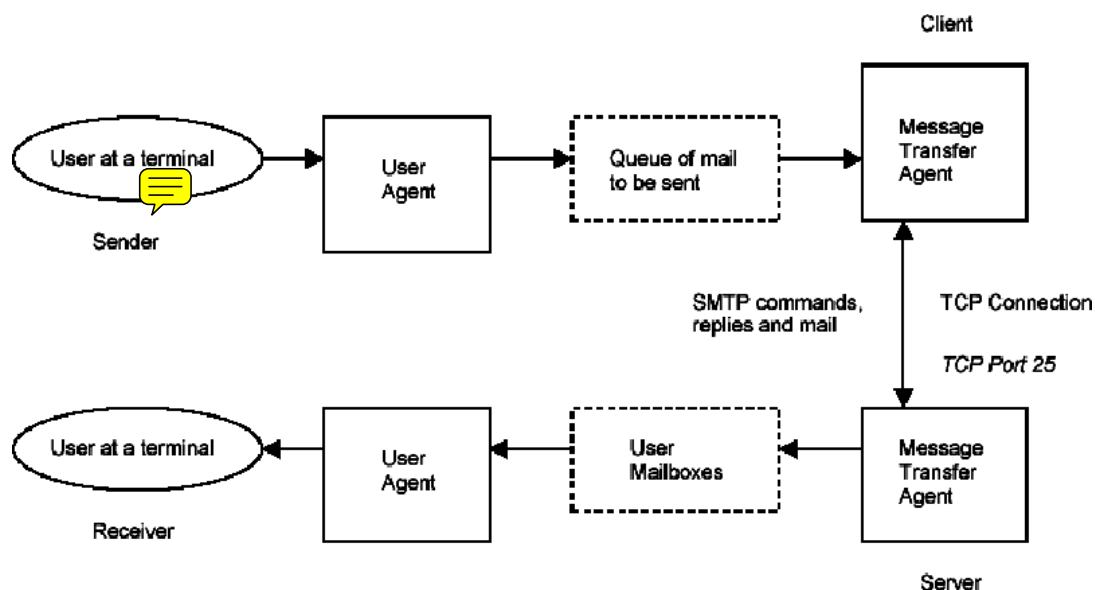


Figure 1: The basic simple mail transfer protocol (SMTP) model

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 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 repeat 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 821 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 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 most common way in the early days of SMTP was through a process called *relaying* (Kozierok 2006). SMTP routing information was included along with the e-mail address, to specify a sequence of SMTP servers that the mail should be relayed through to get to its destination. 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 2005). 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 is located beyond the TCP/IP network. When a mail gateway is used, the SMTP end-to-end transmission is host-to-gateway, gateway-to-host or gateway-to-gateway; the behavior beyond the gateway is not defined by SMTP.

The creation of domain name system (DNS) radically changed the e-mail delivery approach. DNS includes support for a special *mail exchanger record* that allows easy mapping from the domain name in an e-mail address to the IP address of the SMTP server that handles mail for that domain (Kozierok 2006). The sending SMTP server uses DNS to find the mail exchanger record of the domain to which the e-mail is addressed. The sender uses this information for identifying the DNS name of the recipient's SMTP server and for resolving an IP address that can be used for a direct connection between the sender's SMTP server and the recipient's one to deliver the e-mail (RFC 2821).

USER AGENT

Introduced in RFC 821 and RFC 822, the SMTP defines user agent functionality, but not the implementation details. 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 interface (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 transaction 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 envelope 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 differently and accept a blank line as a terminator (SMTP 2004). Everything after the null (or blank) line is the message body, which is a sequence of lines containing ASCII characters. The message body contains the actual information that can be read by the recipient.

Mail Header Format

The header includes a number of key words and values that define the sending date, sender's address, where replies should go, and some other information.

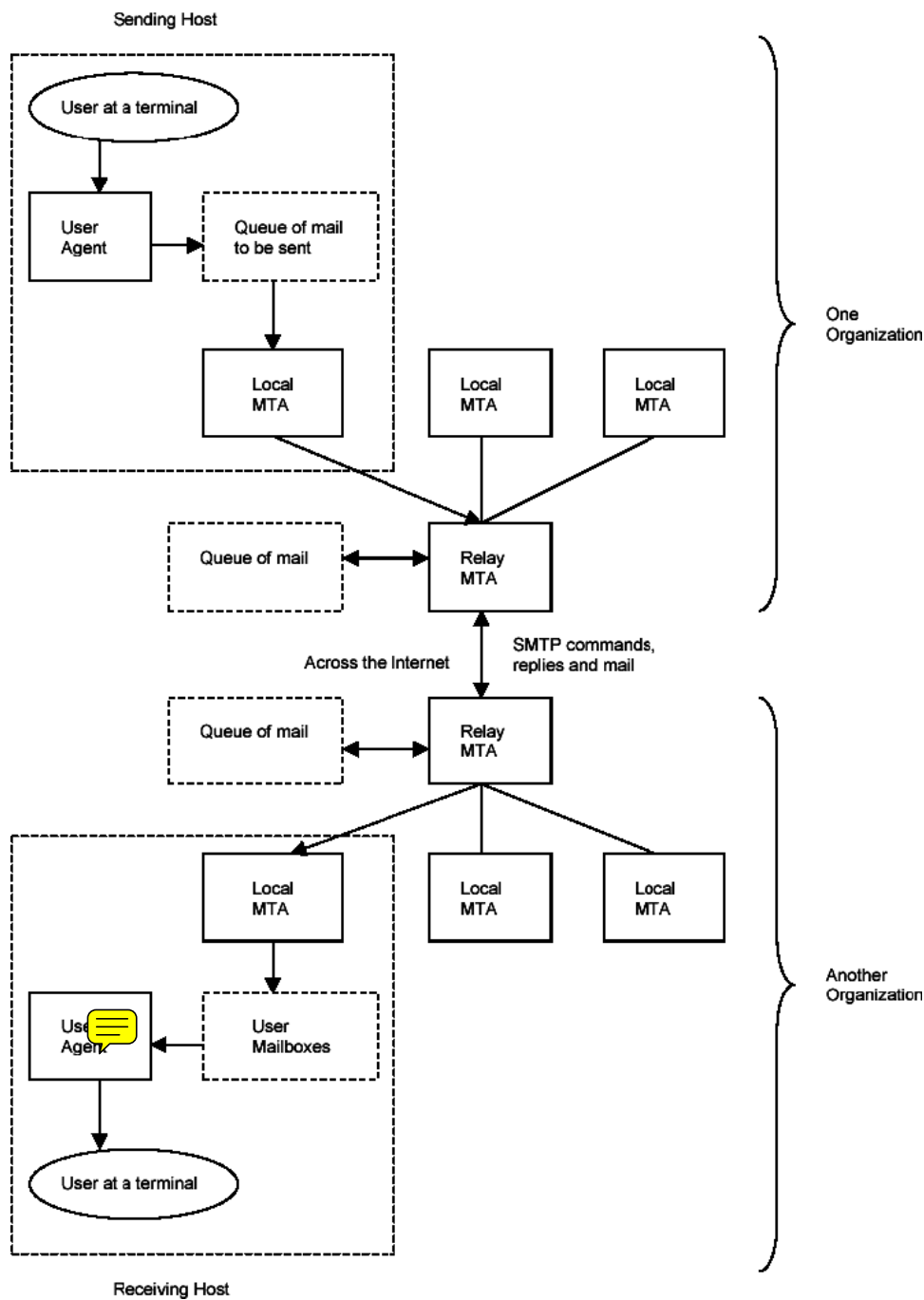


Figure 2: The simple mail transfer protocol (SMTP) model with relay mail transfer agents

The header is a list of lines, of the form (SMTP 2004):

field-name: field-value

Fields begin in column 1: Lines beginning with white space characters (SPACE or TAB) are continuation lines, which are unfolded to create a single line for each field in the canonical representation. Strings enclosed in ASCII quotation marks indicate single tokens within which special 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:

- jsmith@mail.it.rivier.edu
- John Smith <jsmith@mail.it.rivier.edu>
- "John Smith" <jsmith@mail.it.rivier.edu>

The string "John Smith" is intended for human recipients and is the name of the mailbox owner. The string "jsmith@mail.it.rivier.edu" is the computer-readable address of the

mailbox (the angle brackets are used to delimit 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], 2006). 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 following:

- **to** Primary recipients of the message.
- **cc** Secondary (“carbon-copy”) recipients of the message.
- **from** Identity of sender.
- **reply-to** The mailbox to which responses are to be sent. This field is added by the originator.
- **return-path** Address and route back to the originator. This field is added by the final transport system that delivers the mail.
- **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 *local-part@domain-name*, can take several forms (SMTP 2004):

- **user@host**—For a direct destination on the same TCP/IP network.
- **user%remote-host@gateway-host**—For a user on a non-SMTP destination remote-host, via the mail gateway gateway-host.
- **@host-a,@host-b:user@host-c**—For a relayed message. This form contains explicit routing information. The message will first be delivered to host-a, who will resend (relay) the message to host-b. Host-b will then forward 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 message can be delayed at the sender site, the receiver site, or the intermediate servers (Forouzan 2005).

In the case of delaying at the sender site, the client has to accommodate a spooling system, in which e-mail messages are stored before being sent. A message created by the user agent is delivered to the spool storage. The client mail transfer agent periodically (usually every 10 to 30 minutes) checks the spool to find the mail that can be sent. The mail will be sent only if the receiver is ready and the IP address of the server has been obtained through DNS. If a message cannot be delivered in the timeout period (usually about 3 to 5 days), the mail returns to the sender.

Upon receiving the message, the server-MTA stores it in the mailbox of the receiver (see Figure 1). In this case, the receiver can access the mailbox at any convenient time.

Finally, the SMTP standard procedures allow intermediate MTAs to serve as clients and servers. Both intermediate clients and servers can receive mail, store mail messages in their mailboxes and spools, and send them later to an appropriate destination.

Aliases

The SMTP mechanism allows one name, an alias, to represent several e-mail addresses (this feature is known as “one-to-many alias expansion”; Forouzan 2005). Additionally, a single user can also be defined by several e-mail addresses (this is called “many-to-one alias expansion”). The system can handle these expansions by including an alias expansion facility (connected to the alias databases) at both the sender and receiver sites.

MAIL TRANSFER AGENT (MTA)

MTAs transfer actual mail. The system must have the client MTA for sending e-mail and the server MTA for receiving mail (see Figure 1). The SMTP-related RFCs do not define a specific MTA. The UNIX-based MTA uses commonly the Sendmail utility. The most common MTA for Windows is Microsoft Exchange 2000/2003.

The “mta-name-type” and “address-type” parameters (e.g., dnc and rfc822 for the Internet mail, respectively) are defined for use in the SMTP delivery status notification document (see RFC1891). An identification of other mail systems can also be used. One of the identification methods has been described in “The COSINE and Internet X.500 Schema” (section 9.3.18) in the RFC1274 document. The mail system names listed here are used as the legal values in that schema under the “otherMailbox” attribute “mailboxType” type, which must be a PrintableString. The “Mapping between X.400 (1988)/ISO 10021 and RFC 822” is described in the section 4.2.2 of the RFC1327 document. The names listed here are used as the legal values in that schema under the “std-or-address” attribute “registered-dd-type” type, which must be a “key-string” (for details, see Mail Parameters 2006).

SMTP Mail Transaction Flow

The SMTP protocol (RFC 821) defines how commands and responses must be sent by the MTAs. The client sends commands to the server, and the server responds with numeric reply codes and optional human-readable strings.

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 Responses” later in this chapter), the exchange can easily be followed in Figure 3.

In this scenario (Comer 2005; SMTP 2004), the user jsmith at host sun.it.rivier.edu sends a note to users darien, steve and bryan at host mail.unh.edu. Here the lines sent by the server (receiver) are preceded by S, and the lines sent by the client (sender) preceded by C. 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 2004):

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 himself 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 TO:<forward-path> 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.

```
S: 220 mail.unh.edu Simple Mail Transfer Service Ready
C: HELO it.rivier.edu
S: 250 mail.unh.edu

C: MAIL FROM:<jsmith@it.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:<bryan@mail.unh.edu>
S: 550 No such user here

C: DATA
S: 354 Start mail input, end with <CRLF>.<CRLF>
C: Date: 26 Jan 2004 11:02:34 EST
C: From: John Smith <jsmith@it.rivier.edu>
C: Subject: Important meeting
C: To: <darien@mail.unh.edu>
C: To: <steve@mail.unh.edu>
C: cc: <bryan@mail.unh.edu>
C:
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 (“sender” C) and the server (“receiver” S)

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 <CRLF>.<CRLF> message.
6. The client now sends the data line by line, ending with the sequence <CRLF>.<CRLF> line on which the receiver acknowledges with a 250 OK or an appropriate error message if anything went wrong.
7. The following actions (SMTP 2004) 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. Table 1 lists commands and their description and formats. 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 EHLO command is strongly preferred to HELO when the server will accept the former (RFC 2821). Servers must continue to accept and process HELO in order to support older clients.

The other two commands (RSET and NOOP) are often used and highly recommended. The VRFY and EXPN commands are often disabled. This technique allows reducing spam validation of email addresses. The next five programs (TURN, HELP, SEND FROM, SOML FROM, and SAML FROM) are seldom used. The TURN command raises security issues (RFC 2821), because, in the absence of strong authentication of the host requesting that the client and server switch roles, it can easily be used to divert mail from its correct destination. SMTP systems should not use this command unless the server can authenticate the client. The SEND, SAML, and SOML commands were originally introduced in RFC 821 to provide additional, optional mechanism of delivering messages directly to the user's terminal screen. They were rarely implemented, and changes in workstation technology and the introduction of other protocols may have rendered them obsolete even where they are implemented (RFC 2821). SMTP clients should not provide SEND, SAML, or SOML as services.

For a full list of commands, see the RFC 821 "Simple Mail Transfer Protocol," RFC 1123 "Requirements for

Internet Hosts—Application and Support," and RFC 2821 "Simple Mail Transfer Protocol." For details of SMTP service extensions, see the RFC 1651 "SMTP Service Extensions," RFC 1652 "SMTP Service Extension for 8bit-MIMEtransport," 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 2006):

- HELO for Simple Mail (see RFC821)
- EHLO for Mail Service Extensions (see RFC1869)
- LHLO for Local Mail (see RFC2033).

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 2006). 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)
- 8BITMIME—Use 8-bit data; it defines support for the 8-bit content transfer encoding type in MIME (see RFC1652)
- AUTH—Uses to implement an authorization mechanism for servers requiring enhanced security (see RFC2554)
- SIZE—Message size declaration, which allows information about the size of a message to be declared by an SMTP sender prior to transmitting it, so the SMTP receiver can decide if it wants the message or not (see RFC1870)
- CHUNKING—Chunking (see RFC3030)
- BINARYMIME—Binary MIME (see RFC3030)
- CHECKPOINT—Checkpoint/restart (see RFC1845)
- PIPELINING—Command pipelining, which allows multiple commands to be transmitted in batches from the SMTP sender to the receiver, rather than sending

Table 1: Simple Mail Transfer Protocol (SMTP) Commands. Adapted from SMTP Specifications 2006

Command	Description	Format	References
ATRN	Authenticated TURN		RFC 2645
AUTH	Authentication		RFC 2554
BDAT	Binary data		RFC 3030
DATA	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	DATA Best wishes.	RFC 821, RFC 2821
EHLO	Extended Hello		RFC 1869, RFC 2821
ETRN	Extended TURN		RFC 1985
EXPN	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	EXPN: a b c	RFC 821, RFC 2821
HELO	Hello; used by the client to identify itself	HELO: sun.it.rivier.edu	RFC 821, RFC 2821
HELP	Help; requests the recipient to send information about the command sent as the argument	HELP: mail	RFC 821, RFC 2821
MAIL FROM	Mail; used by the client to identify the sender of the message; the argument is the e-mail address of the sender	MAIL FROM: jsmith@sun.it.rivier.edu	RFC 821, RFC 2821
NOOP	No operation; used by the client to check the status of the recipient; requires an answer from the recipient	NOOP	RFC 821, RFC 2821
QUIT	Quit; terminates the message	QUIT	RFC 821, RFC 2821
RCPT	Recipient; used by the client to identify the intended recipient of the message; if there are multiple recipients, the command is repeated	RCPT TO: steve@unh.edu	RFC 821, RFC 2821
RSET	Reset; aborts the current e-mail transaction; the stored information about the sender and recipient is deleted; the connection will be reset	RSET	RFC 821, RFC 2821
SAML	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	SAML FROM: jsmith@sun.it.rivier.edu	RFC 821
SEND	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	SEND FROM: jsmith@sun.it.rivier.edu	RFC 821
SOML	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	SOML FROM: jsmith@sun.it.rivier.edu	RFC 821
STARTTLS	Extended Hello with transport layer security		RFC 3207
TURN	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)	TURN	RFC 821
VRFY	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.	VRFY: steve@unh.edu	RFC 821, RFC 2821

- one command at a time and waiting for a response code (see RFC2920)
- DSN—Delivery status notification, which allows an SMTP sender to request that the SMTP receiver notify if a problem occurs in delivering a message the sender gives to it (see RFC1891)
 - ETRN—Extended turn (see RFC1985)
 - ENHANCEDSTATUSCODES—Enhanced status codes; it extends the traditional 3-digit SMTP reply code format with extra codes that provide more information (see RFC2034 and RFC1893)
 - STARTTLS—Start TLS (see RFC3207).

Some of these key words have parameters (for details, see Mail Parameters 2006).

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 (e.g., see RFC 1846).

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

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” (2006).

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), 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 2006). 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 2006).

MULTIPURPOSE INTERNET MAIL EXTENSIONS (MIME)

The RFC 821/ 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 multipurpose Internet mail extensions (MIME) supplementary protocol was 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 (2006).

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 supports. New procedures modifies the RFC 821 standard to allow a client SMTP agent to request that the server responds with a list of the service extensions that it supports 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 2006); 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

Table 2: Simple Mail Transfer Protocol (SMTP) Reply Codes. Adapted from SMTP Specifications 2006.

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

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 messages, which are declared as consisting of 8-bit data rather than 7-bit data, when both the client and the server conform 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 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 can be considered as a set of software functions that transforms non-ASCII data to ASCII characters and vice versa, as shown in Figure 4.

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 (see Figure 5). 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 (2005).

Content-Transfer-Encoding

The Content-Transfer-Encoding:<type> header line defines the method to encode the messages into a bit-stream

Figure 5: MIME header

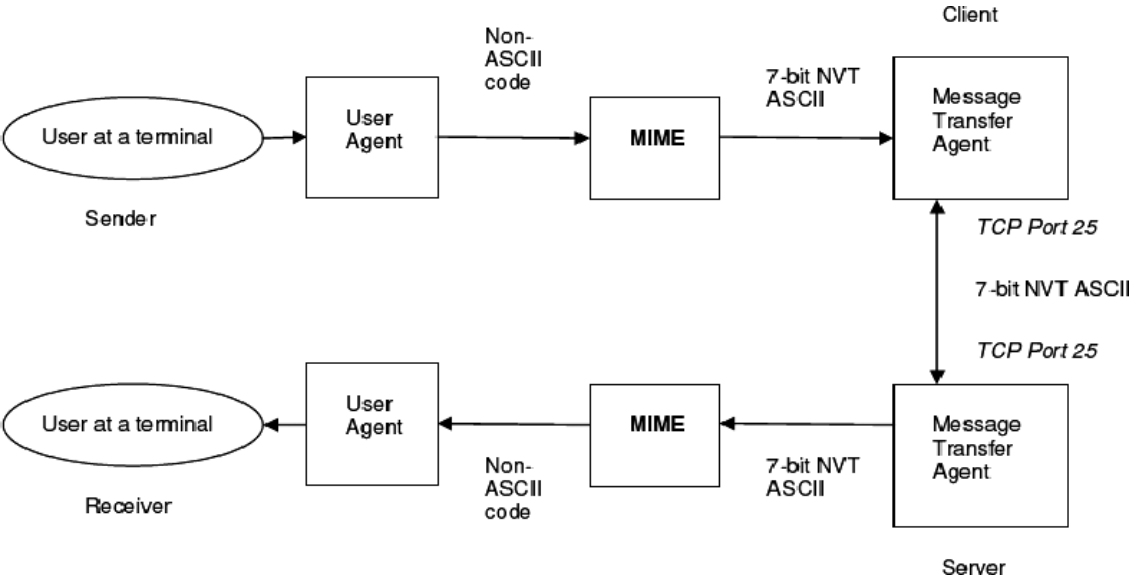
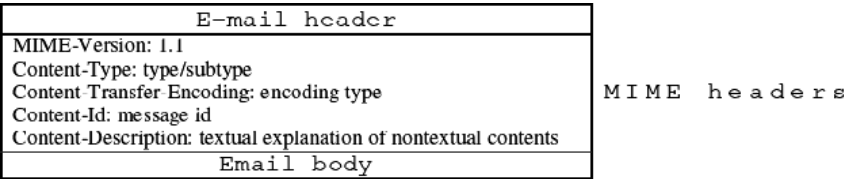


Figure 4: MIME functionality

Table 3: Data Types and Subtypes in a Multipurpose Internet Mail Extensions (MIME)Content-Type Heaser Declaration

Type	Subtype	Description
Text	Plain	Unformatted 7-bit ASCII text; no transformation by MIME is needed
	HTML	HTML format
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Body contains no-ordered parts of different data types
	Digest	Body contains ordered parts of different data types, but the default is message/RFC822
	Alternative	Parts are different versions of the same message
Message	RFC822	Body is an encapsulated message
	Partial	Body is a fragment of a bigger message
	External-Body	Body is a reference to another message
Image	JPEG	Image is in JPEG format
	GIF	Image is in GIF format
Video	MPEG	Video is in MPEG format
Audio	Basic	Single channel encoding of voice at 8 KHz
Application	PostScript	Adobe PostScript
	Octet-stream	General binary data (eight-bit bytes)

GIF = Graphics Interchange Format; HTML = Hypertext Markup Language; JPEG = Joint Photographic Experts Group; MPEG = Motion Picture Experts Group.

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 Tschabitscher 2006; 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).

Figure 6 shows an example of a multi-part message in MIME format with mixed subtypes.

Content-Id

The header line Content-Id: id=<content-id> uniquely identifies the whole message in a multiple message environment.

Content-Description

The header line Content-Description:<description> defines whether the body is image, audio, or video.

Security Scheme for MIME

The S/MIME is a security scheme for the MIME protocol. It was developed by RSA Security and is an alternative to the pretty good privacy (PGP) encryption and digital signature scheme that uses public-key cryptography. The S/MIME scheme was standardized by IETF. According to “Report of the IAB Security Architecture Workshop” (RFC 2316), the designated security mechanism for adding secured sections to MIME-encapsulated e-mail is security/multipart, as described in “Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted” (RFC 1847).

The S/MIME is widely used by large companies that need to standardize e-mail security for both interorganization and intraorganization mail exchange (Internet Engineering Task Force [IETF] SMIME, 2006). It requires establishing a public-key infrastructure either in-house or by using any of the public certificate authorities (Sheldon 2001).

```

From: "Smith, John" <jsmith@college.edu>
To: "Peter Adams" <peter_adams@hotmail.com>
Subject: CS Mid Term
Date: Fri, 3 Mar 2006 09:08:10 -0500
MIME-Version: 1.0
Content-Type: multipart/alternative;
    boundary="----=_NextPart_001_01C3AF7F.35820A9B"

Received: from EXCHANGEMAIL.COLLEGE.EDU ([67.251.112.30]) by
bay0-mc10-f2.bay0.hotmail.com with Microsoft SMTPSVC(6.0.3790.211); Fri, 3
Mar 2006 06:08:12 -0800
X-Message-Info: JGTYoYF78jEHjJx36Oi8+Z3TmmkSEdPtfpLB7P/ybN8=
Content-class: urn:content-classes:message
X-MimeOLE: Produced By Microsoft Exchange V6.5.7226.0
X-MS-Has-Attach: X-MS-TNEF-Correlator: Thread-Topic: CS Mid Term
Thread-Index: AcY+qoksOm67V+jHSd+jwoRpQ5vCCwAICufp
References: <BAY108-F243FC3114EBD6165216203E9EA0@phx.gbl>
Return-Path: jsmith@college.edu
X-OriginalArrivalTime: 03 Mar 2006 14:08:12.0085 (UTC)
FILETIME=[E827E650:01C63ECB]

This is a multi-part message in MIME format.

-----=_NextPart_001_01C3AF7F.35820A9B
Content-Type: text/plain;
    charset="utf-8"
Content-Transfer-Encoding: base64

RGVhciBTdHVkZW50LA0KIA0KVG9uaWdodCBhZnRlciBvdXIgQ1M1NTMgY2xhc3MgKGF0IDc6MzMzAg
UE0pIHlvdSBhcmUgaW52aXRlZCB0byB2aXNpdCBvdXIgY29sbGVnZSBjVCBMQU4gQ2VudGVyIGxv
Y2F0ZWQgaW4gdGhlIFNUSCBldWlsZGluZyBvbiB0aGUgZmlyc3QgZmxvb3IgbmVhci==

-----=_NextPart_001_01C3AF7F.35820A9B
Content-Type: text/html;
    charset="utf-8"
Content-Transfer-Encoding: base64

PCFET0NUWVBFIEhUTUwgUFVCTElDIChlLy9XM0MvL0RURCBIVElMIDQuMCMBUcmFuc210aW9uYWwv
L0VOIj48SFRNTD48SEVBRD48TUVUQSBIVFRQLUVVRVU1WPSJDb250ZW50LVR5cGUiIENPTlRFTlQ9
InRleHQvaHRtbDsgY2hhcnNldD1ldGYtOCI+PC9IRUFEPjxCT0RZPjxESVY+RGVhciBTdHVkZW50
LDwvREL=

```

Figure 6: An example of a multi-part message in MIME format with mixed subtypes

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 as they are transported through the Internet (Mail Parameters 2006). 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 UUCP 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 ESMTP 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 user's computer, and only the original remains on the server.
- **Disconnected/resynchronization 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 multiplatform 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). Two dedicated protocols (POP3 and IMAP4) of retrieving e-mail are considered in the next section.

Instead of using POP3 or IMAP4, on some systems it is possible for a user to have direct server access to e-mail. This is most commonly done on UNIX systems, where protocols like TELNET or NFS (Network File System) can give a user shared access to mailboxes on a server (TCP/IP Guide 2006). Being the oldest method of e-mail access, it provides the user (who must be on the Internet to read e-mail) with the most control over his or her mailbox, and is well-suited to those who must access mail from many locations.

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 1939). Typically, e-mail arrives from the network and is placed in the user's inbox on the server. POP is then used to transfer the mail from the user's inbox 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 inbox 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, 2004). 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 2004). 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 elective. 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 1064). 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 "IMAP4/POP Authorization for Simple Challenge/Response" (RFC 2195), "IMAP4 Login Referrals" (RFC 2221), and "IMAP4 Implementation and Best Practices" (RFC 2683).

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" (2006) released by the Saint Corporation. The Common Vulnerabilities and Exposures (CVE) organization provides a list of standardized names for SMTP vulnerabilities (for both CVE entries and CAN candidates) and other information security exposures (CVE 2006). Summaries of major SMTP vulnerability problems are discussed in (Riabov 2006).

SMTP was designed in an era when security of the internet was not an issue. As a result, the SMTP protocol includes no robust security mechanism. For example, someone can use the TELNET protocol to connect directly to an SMTP server on port 25. The SMTP commands and replies can all be sent as text, and, therefore, a person can manually perform a mail transaction. This is useful for debugging, but also makes abuse of a wide open SMTP server trivially easy. Since spammers often do not want to be identified, they employ spoofing techniques to make it more difficult to identify them (Kozierok 2006). Nowadays, most modern SMTP servers incorporate several security features to avoid vulnerability problems.

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. Detailed information about vulnerability problems, possible actions of an attacker or spammer, recommendations for downloading updated versions of software, examples of code modification, and test results can be found on the CERT (2006) and Security Space ("SMTP Problems," 2006) 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. The medium-to-high risk category includes denial-of-service attacks. Low-to-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" 2006). Most SMTP-specific vulnerabilities occur from misapplied or unapplied patches related to Sendmail installations or misconfigured Sendmail daemons on the SMTP servers (Campbell et al. 2003).

ISPs restrict access to their outgoing mail servers to provide better service to their customers and prevent spam from being sent through their mail servers. There are several methods for establishing restrictions that could result in denying users' access to their outgoing mail server.

Originally (see RFC 821), e-mail servers (configured for SMTP relay) did not verify the claimed sender identity and would simply pass the mail on with whatever return address was specified. Bulk mailers have taken advantage of this to send huge volumes of mail with bogus return addresses. 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 et al. 2003). Other initiatives for restricting the sender address spoofing include SPF, Hotmail domain cookies, and Microsoft's caller ID. The analysis of various SMTP security issues can be found in (Riabov 2006).

STANDARDS, ORGANIZATIONS, AND ASSOCIATIONS

Internet Assigned Numbers Authority

The IANA (2006) 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 2006).

The IETF Message Tracking Protocol (MSGTRK) Working Group (IETF MSGTRK, 2006) 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 "Deliver 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, 2006).

Internet Mail Consortium

The Internet Mail Consortium Web site (IMC 2006) 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 2006).

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 MAIL and RCPT commands. MIME, 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 SMTP's lack of security is a problem for businesses. The security in the SMTP transactions can be supported by S/MIME and other methods.

GLOSSARY

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

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, mails sent to an IMS will use encapsulated SMTP as their 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 mail server. The user can then choose which messages to download to his or her computer. While 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.

Mail delivery agent (MDA): The software that runs mail-delivery processes on the machine where a users' mailbox is located. Often, that delivery is performed directly by the mail transfer agent (MTA), which then serves a secondary role as an MDA. Examples of separate mail delivery agents include Procmail, Deliver, and Cyrdeliver.

Mailbox: A file where e-mail messages are stored.

Mail relaying: 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 try 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 nontext 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.

Network virtual terminal (NVT): A set of facilities for establishing communication by using the TCP/IP protocols

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. An e-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 e-mail servers on the Internet.

Spam: 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 Domain Name System*; *TCP/IP Suite*.

REFERENCES

- QUALCOMM. 1997. Accessing your mail when and where you want on the Internet. http://www.eudora.com/pdf_docs/primer.pdf (accessed March 11, 2006).
- Campbell, P., B. Calvert, and S. Boswell. 2003. *Security+ Guide to Network Security Fundamentals*. Boston: Cisco Learning Institute.
- CERT Computer Emergency Readiness Team. 2006. Vulnerability Database. <http://www.cert.org/> (accessed March 11, 2006).
- Cisco SMTP. 2006. <http://www.cisco.com/univercd/cc/td/doc/product/software/ioss390/ios390ug/ugsmtp.htm> (accessed March 11, 2006).
- Comer, D. F. 2005. *Internetworking with TCP/IP, Vol. 1: Principles, Protocols, and Architecture*, 5th edition. Upper Saddle River, NJ: Prentice Hall.
- CVE: Common Vulnerabilities and Exposures. 2006. Mitre Corporation. Retrieved March 11, 2006, from <http://cve.mitre.org/> (accessed March 11, 2006).
- Forouzan, B. A. 2005. *TCP/IP Protocol Suite*, 3rd edition. New York: McGraw-Hill.
- How to set SMTP security options in Windows 2000. 2006. <http://support.microsoft.com/default.aspx?scid=http://support.microsoft.com:80/support/kb/articles/Q303/7/76.ASP&NoWebContent=1> (accessed March 11, 2006).
- IETF DRUMS. 2006. Internet Engineering Task Force Working Group: Detailed Revision/Update of Message Standards (DRUMS). <http://www.ietf.org/html.chapters/OLD/drums-chapter.html> (accessed March 11, 2006).
- IETF MSGTRK. 2006. Internet Engineering Task Force Working Group. Message Tracking Protocol (MSGTRK). <http://www.ietf.org/html.chapters/OLD/msgtrk-chapter.html> (accessed March 11, 2006).
- IETF SMIME. 2006. Internet Engineering Task Force Working Group. S/MIME Mail Security (SMIME). <http://www.ietf.org/html.chapters/smime-chapter.html> (accessed March 11, 2006).
- Internet Assigned Numbers Authority (IANA). 2006. <http://www.iana.org/> (accessed March 11, 2006).
- Internet Mail Consortium (IMC). 2006. <http://www.imc.org/rfcs.html> (accessed March 11, 2006).
- Kozierok, C. M. 2006. TCP/IP Electronic Mail Delivery Protocol: The Simple Mail Transfer Protocol (SMTP). http://www.tcpipguide.com/free/t_TCPIPElectronicMailDeliveryProtocolTheSimpleMailTr.htm (accessed March 11, 2006).
- Mail Parameters. 2006. <http://www.iana.org/assignments/mail-parameters> (accessed March 11, 2006).

- RFC821 (STD 10): Simple mail transfer protocol. 1982. <http://www.ietf.org/rfc/rfc821.txt> (accessed March 11, 2006).
- RFC822 (STD 11): Standard for the format of ARPA—Internet Text Messages. 1982. <http://www.ietf.org/rfc/rfc822.txt> (accessed March 11, 2006).
- RFC876: Survey of SMTP implementations. 1983. <http://www.ietf.org/rfc/rfc876.txt> (accessed March 11, 2006).
- RFC937: Post office protocol—Version 2. 1985. <http://www.ietf.org/rfc/rfc937.txt> (accessed March 11, 2006).
- RFC1064: Interactive mail access protocol—Version 2. 1988. <http://www.ietf.org/rfc/rfc1064.txt> (accessed March 11, 2006).
- RFC1123: Requirements for Internet hosts—application and support. 1989. <http://www.ietf.org/rfc/rfc1123.txt> (accessed March 11, 2006).
- RFC1274: The COSINE and Internet X.500 schema. 1991. <http://www.ietf.org/rfc/rfc1274.txt> (accessed March 11, 2006).
- RFC1327: Mapping between X.400 (1988)/ISO10021 and RFC 822. 1992. <http://www.ietf.org/rfc/rfc1327.txt> (accessed March 11, 2006).
- RFC1425: SMTP Service Extensions. 1993. <http://www.ietf.org/rfc/rfc1425.txt> (accessed March 11, 2006).
- RFC1521: MIME (multipurpose internet mail extensions), part one: Mechanisms for specifying and describing the format of Internet message bodies. 1993. <http://www.ietf.org/rfc/rfc1521.txt> (accessed March 11, 2006).
- RFC1522: MIME (multipurpose internet mail extensions), part two: Message header extensions for non-ASCII Text. 1993. <http://www.ietf.org/rfc/rfc1522.txt> (accessed March 11, 2006).
- RFC1651: SMTP service extensions. 1994. <http://www.ietf.org/rfc/rfc1651.txt> (accessed March 11, 2006).
- RFC1652: SMTP Service Extension for 8bit-MIME transport. 1994. <http://www.ietf.org/rfc/rfc1652.txt> (accessed March 11, 2006).
- RFC1653: SMTP Service extension for message size declaration. 1994. <http://www.ietf.org/rfc/rfc1653.txt> (accessed March 11, 2006).
- RFC1725: Post office protocol—version 3, RFC 1725. 1994. <http://www.ietf.org/rfc/rfc1725.txt> (accessed March 11, 2006).
- RFC1731: IMAP4 authentication mechanisms. 1994. <http://www.ietf.org/rfc/rfc1731.txt> (accessed March 11, 2006).
- RFC1734: POP3 AUTHentication command. 1994. <http://www.ietf.org/rfc/rfc1734.txt> (accessed March 11, 2006).
- RFC1845: SMTP service extension for Checkpoint/Restart. 1995. <http://www.ietf.org/rfc/rfc1845.txt> (accessed March 11, 2006).
- RFC1846: SMTP 521 reply code. 1995. <http://www.ietf.org/rfc/rfc1846.txt> (accessed March 11, 2006).
- RFC1847: Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted. 1995. <http://www.ietf.org/rfc/rfc1847.txt> (accessed March 11, 2006).
- RFC1869: SMTP service extensions. 1995. <http://www.ietf.org/rfc/rfc1869.txt> (accessed March 11, 2006).
- RFC1870: SMTP service extension for message size declaration. 1995. from <http://www.ietf.org/rfc/rfc1870.txt> (accessed March 11, 2006).
- RFC1891: SMTP service extension for delivery status notification. 1996. <http://www.ietf.org/rfc/rfc1891.txt> (accessed March 11, 2006).
- RFC1893: Enhanced Mail System Status Codes. 1996. <http://www.ietf.org/rfc/rfc1893.txt> (accessed March 11, 2006).
- RFC1939 (STD 53): Post office protocol, version 3. 1996. <http://www.ietf.org/rfc/rfc1939.txt> (accessed March 11, 2006).
- RFC1985: SMTP Service extension for remote message queue starting. 1996. <http://www.ietf.org/rfc/rfc1985.txt> (accessed March 11, 2006).
- RFC2033: Local mail transfer protocol. 1996. <http://www.ietf.org/rfc/rfc2033.txt> (accessed March 11, 2006).
- RFC2034: SMTP service extension for returning enhanced status codes. 1996. <http://www.ietf.org/rfc/rfc2034.txt> (accessed March 11, 2006).
- RFC2060: Internet message access protocol, Version 4rev1. 1996. <http://www.ietf.org/rfc/rfc2060.txt> (accessed March 11, 2006).
- RFC2195: IMAP/POP authorization for simple challenge/response. 1997. <http://www.ietf.org/rfc/rfc2195.txt> (accessed March 11, 2006).
- RFC2221: IMAP4 login referrals. 1997. <http://www.ietf.org/rfc/rfc2221.txt> (accessed March 11, 2006).
- RFC2316: Report of the IAB Security Architecture Workshop. 1998. <http://www.ietf.org/rfc/rfc2316.txt> (accessed March 11, 2006).
- RFC2449: POP3 extension mechanism. 1998. <http://www.ietf.org/rfc/rfc2449.txt> (accessed March 11, 2006).
- RFC2554: SMTP service extension for authentication. 1999. <http://www.ietf.org/rfc/rfc2554.txt> (accessed March 11, 2006).
- RFC2595: Using TLS with IMAP, POP3 and ACAP. 1999. <http://www.ietf.org/rfc/rfc2595.txt> (accessed March 11, 2006).
- RFC2645: On-demand mail relay (ODMR) SMTP with dynamic IP addresses. 1999. <http://www.ietf.org/rfc/rfc2645.txt> (accessed March 11, 2006).
- RFC2683: IMAP4 implementation and best practices. 1999. <http://www.ietf.org/rfc/rfc2683.txt> (accessed March 11, 2006).
- RFC2821: Simple mail transfer protocol. 2001. <http://www.ietf.org/rfc/rfc2821.txt> (accessed March 11, 2006).
- RFC2822: Internet Message Format. 2001. <http://www.ietf.org/rfc/rfc2822.txt> (accessed March 11, 2006).
- RFC2852: Deliver by SMTP service extension. 2000. <http://www.ietf.org/rfc/rfc2852.txt> (accessed March 11, 2006).
- RFC2920: SMTP service extension for command pipelining. 2000. <http://www.ietf.org/rfc/rfc2920.txt> (accessed March 11, 2006).
- RFC3030: SMTP service extensions for transmission of large and binary MIME messages. 2000. <http://www.ietf.org/rfc/rfc3030.txt> (accessed March 11, 2006).
- RFC3207: SMTP service extension for secure SMTP over transport layer security. 2002. <http://www.ietf.org/rfc/rfc3207.txt> (accessed March 11, 2006).
- Riabov, V. V. 2006. Simple mail transfer protocol. In: *Handbook on Information Security, volume 1: Key Concepts, Infrastructures, Standards and Protocols*, edited

- by Hossein Bidgoli. Hoboken, NJ: John Wiley & Sons, pp. 878–900.
- Rose, M. T. 1993. *The Internet Message, Closing the Book with Electronic Mail*. Upper Saddle River, NJ: Prentice Hall.
- Sheldon, T. 2001. *McGraw-Hill Encyclopedia of Networking and Telecommunications*. New York: McGraw-Hill.
- Simple Mail Transfer Protocol (SMTP). (2004). <http://ulla.mcgill.ca/arts150/arts150bs.htm> (accessed September 24, 2004).
- SMTP problems. (2006). E-Soft, Inc. <http://www.securityspace.com/smysecure/catdescr.html?cat=SMTP+problems> (accessed March 11, 2006).
- SMTP specifications. 2006. <http://www.networksorcery.com/enp/protocol/smtp.htm> (accessed March 11, 2006).
- Stevens, W. R. 1993. *TCP/IP Illustrated: The Protocols, Volume I*. Boston, MA: Addison-Wesley.
- Tanenbaum, A. S. 2003. *Computer Networks*, 4th edition. Upper Saddle River, NJ: Prentice Hall PTR.
- TCP/IP Guide. 2006. TCP/IP Electronic Mail Access and Retrieval Protocols and Methods. http://www.tcpipguide.com/free/t_TCPIPElectronicMailAccessandRetrievalProtocolsandM.htm (accessed March 11, 2006).
- Tschabitscher, H. 2006. How Base64 Encoding Works. In: *Your Guide to E-mail*. http://email.about.com/cs/standards/a/base64_encoding.htm (accessed March 11, 2006).
- Vulnerability Tutorials. 2006. Saint Corporation. http://www.saintcorporation.com/demo/saint/vulnerability_tutorials.html (accessed March 11, 2006).
- What is SMTP? 2006. http://whatis.techtarget.com/definition/0,289893,sid9_gci214219,00.html (accessed March 11, 2006).
- Wikipedia. 2006. Mail Delivery Agent. http://en.wikipedia.org/wiki/Mail_Delivery_Agent (accessed March 11, 2006).
- Office Solutions. <http://www.msexchange.org/tutorials/MF005.html> (accessed March 11, 2006).
- IMAP Information Center. 2006. <http://www.washington.edu/imap/> (accessed March 11, 2006).
- Microsoft Security Bulletins. 2006. <http://www.microsoft.com/technet/security/bulletin/> (accessed March 11, 2006).
- Raynal, F. 2000. Bastille Linux, MISC Magazine. <http://www.security-labs.org/index.php3?page=103> (accessed March 11, 2006).
- RFC1090: SMTP on X.25. 1989. <http://www.ietf.org/rfc/rfc1090.txt> (accessed March 11, 2006).
- RFC1730: Internet message access protocol—Version 4. 1994. <http://www.ietf.org/rfc/rfc1730.txt> (accessed March 11, 2006).
- RFC1733: Distributed electronic mail models in IMAP4. 1994. <http://www.ietf.org/rfc/rfc1733.txt> (accessed March 11, 2006).
- RFC1830: SMTP service extensions for transmission of large and binary MIME messages. 1995. <http://www.ietf.org/rfc/rfc1830.txt> (accessed March 11, 2006).
- RFC2045: MIME, part one: Format of Internet message bodies. 1996. <http://www.ietf.org/rfc/rfc2045.txt> (accessed March 11, 2006).
- RFC2046: MIME, part two: Media types. 1996. <http://www.ietf.org/rfc/rfc2046.txt> (accessed March 11, 2006).
- RFC2047: MIME, part three: Message header extensions for non-ASCII text. 1996. <http://www.ietf.org/rfc/rfc2047.txt> (accessed March 11, 2006).
- RFC2048: MIME, part four: Registration procedures. 1996. <http://www.ietf.org/rfc/rfc2048.txt> (accessed March 11, 2006).
- RFC2049: MIME, part five: Conformance criteria and examples. 1996. <http://www.ietf.org/rfc/rfc2049.txt> (accessed March 11, 2006).
- RFC2197: SMTP service extension for command pipelining. 1997. <http://www.ietf.org/rfc/rfc2197.txt> (accessed March 11, 2006).
- RFC2442: The batch SMTP media type. 1998. <http://www.ietf.org/rfc/rfc2442.txt> (accessed March 11, 2006).
- RFC2487: SMTP service extension for secure SMTP over TLS. 1999. <http://www.ietf.org/rfc/rfc2487.txt> (accessed March 11, 2006).
- RFC2505: Anti-spam recommendations for SMTP MTAs. 1999. <http://www.ietf.org/rfc/rfc2505.txt> (accessed March 11, 2006).
- RFC3461: Simple mail transfer protocol (SMTP) service extension for delivery status notifications (DSNs). 2003. <http://www.ietf.org/rfc/rfc3461.txt> (accessed March 11, 2006).
- Setting SMTP Security. 2006. Texoma, Inc. http://help.texoma.net/imap/user/setting_smtp_security.htm (accessed March 11, 2006).
- SMTP Tutorial at RAD Data Communications. (1998). <http://www.rad.com/networks/1998/smtp/smtp.htm> (accessed March 11, 2006).
- The IMAP Connection. 2006. <http://www.imap.org/> (accessed March 11, 2006).
- What is SMTP Security? 2006. http://help.westelcom.com/faq/what_is_smtp.htm (accessed March 11, 2006).

FURTHER READING

- Antirelay Parse. 2006. Sendmail organization, antirelay rules. <http://www.sendmail.org/antirelay.Parse0.txt> (accessed March 11, 2006).
- Authentication error in SMTP service could allow mail relaying. 2001. Microsoft Security Bulletin, MS01-037. <http://www.microsoft.com/technet/security/bulletin/MS01-037.msp> (accessed March 11, 2006).
- Bastille Linux Project. 2006. Open Source Development Network. <http://sourceforge.net/projects/bastille-linux/>
- Bastille Project. 2006. <http://www.bastille-linux.org/> (accessed March 11, 2006).
- CA Vulnerability Information Center. 2000. @Work Smart-Server3 SMTP vulnerability. <http://www3.ca.com/securityadvisor/vulninfo/Vuln.aspx?ID=1972> (accessed March 11, 2006).
- Fugatt, M. 2002. Blocking incoming mail using Microsoft Exchange 2000. Tutorials: Exchange 2000, Pentech Office Solutions. <http://www.msexchange.org/tutorials/MF014.html> (accessed March 11, 2006).
- Fugatt, M. 2002. Understanding relaying and spam with Exchange 2000. Tutorials: Exchange 2000, Pentech