



**National Institute of
Standards and Technology**

Technology Administration
U.S. Department of Commerce

Special Publication 800-45

Guidelines on Electronic Mail Security

Recommendations of the National Institute of Standards and Technology

Miles Tracy, Wayne Jansen, and Scott Bisker

NIST Special Publication 800-45

Guidelines on Electronic Mail Security

*Recommendations of the National Institute of
Standards and Technology*

Miles Tracy, Wayne Jansen, and Scott Bisker

C O M P U T E R S E C U R I T Y

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8930

September 2002



U .S. Department of Commerce
Donald L. Evans, Secretary

Technology Administration
Phillip J. Bond, Under Secretary for Technology

National Institute of Standards and Technology
Arden L. Bement, Jr., Director

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analysis to advance the development and productive use of information technology. ITL's responsibilities include the development of technical, physical, administrative, and management standards and guidelines for the cost-effective security and privacy of sensitive unclassified information in Federal computer systems. This Special Publication 800-series reports on ITL's research, guidance, and outreach efforts in computer security, and its collaborative activities with industry, government, and academic organizations.

National Institute of Standards and Technology Special Publication 800-45

Natl. Inst. Stand. Technol. Spec. Publ. 800-45, xx pages (Mon. 2002)

CODEN: XXXXX

<p>Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.</p>

**U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 2002**

For sale by the Superintendent of Documents, U.S. Government Printing Office
Internet: bookstore.gpo.gov — Phone: (202) 512-1800 — Fax: (202) 512-2250
Mail: Stop SSOP, Washington, DC 20402-0001

Acknowledgements

The authors, Wayne Jansen of NIST and Scot Bisker and Miles Tracy of Booz Allen Hamilton (BAH), wish to express their thanks to colleagues at both organizations who reviewed drafts of this document. In particular, their appreciation goes to John Wack, Murugiah Souppaya, and Tim Grance from NIST, and Steve Allison, Alexis Feringa, Jonathan Holleran, Kevin Kuhlkin, and Mark McLarnon from BAH, for their research, technical support, and written contributions to this document. The authors would also like to express their thanks to all those who contributed input during the public comment period and who assisted with our internal review process.

Table of Contents

EXECUTIVE SUMMARY	ES-1
1. INTRODUCTION	1
1.1 AUTHORITY	1
1.2 PURPOSE AND SCOPE	1
1.3 AUDIENCE AND ASSUMPTIONS	2
1.4 DOCUMENT ORGANIZATION	3
2. BACKGROUND AND STANDARDS	4
2.1 BACKGROUND	4
2.2 MULTIPURPOSE INTERNET MAIL EXTENSIONS	5
2.3 MAIL TRANSPORT STANDARDS	6
2.4 SIMPLE MAIL TRANSFER PROTOCOL	6
2.5 SIMPLE MAIL TRANSFER PROTOCOL EXTENSIONS	8
2.6 PROPRIETARY MAIL TRANSPORTS	10
2.7 CLIENT ACCESS STANDARDS	10
2.8 POST OFFICE PROTOCOL	10
2.9 INTERNET MESSAGE ACCESS PROTOCOL	11
2.10 PROPRIETARY MAILBOX ACCESS MECHANISMS	13
3. EMAIL-RELATED ENCRYPTION STANDARDS	14
3.1 PRETTY GOOD PRIVACY	15
3.2 S/MIME	17
3.3 CHOOSING AN APPROPRIATE ENCRYPTION ALGORITHM	18
3.4 KEY MANAGEMENT	19
3.5 CHOOSING BETWEEN PGP AND S/MIME	20
4. PLANNING AND MANAGEMENT OF MAIL SERVERS	21
4.1 PLANNING THE INSTALLATION AND DEPLOYMENT OF THE MAIL SERVER	21
4.2 SECURITY MANAGEMENT STAFF	23
4.3 MANAGEMENT PRACTICES	25
4.4 SYSTEM SECURITY PLAN	26
4.5 HUMAN RESOURCES FOR SECURING AN EMAIL SERVER	28
4.6 GENERAL INFORMATION SYSTEM SECURITY PRINCIPLES	29
5. SECURING THE OPERATING SYSTEM	31
5.1 SECURELY INSTALLING AND CONFIGURING AN OPERATING SYSTEM	31
5.2 SECURITY TESTING THE OPERATING SYSTEM	35
5.3 RESOURCES FOR OPERATING SYSTEM SPECIFIC SECURITY PROCEDURES	35
5.4 SECURING THE MAIL SERVER OPERATING SYSTEM CHECKLIST	36
6. MAIL SERVER AND CONTENT SECURITY	38
6.1 HARDENING THE MAIL SERVER APPLICATION	38
6.2 PROTECTING EMAIL FROM MALICIOUS CODE	40
6.3 UNSOLICITED BULK EMAIL	48
6.4 AUTHENTICATED MAIL RELAY	49

6.5	SECURE ACCESS	49
6.6	ENABLING WEB ACCESS.....	50
6.7	MAIL SERVER AND CONTENT SECURITY CHECKLIST.....	51
7.	IMPLEMENTING A SECURE NETWORK FOR A MAIL SERVER.....	53
7.1	NETWORK LOCATION	53
7.2	USING MAIL GATEWAYS	56
7.3	NETWORK ELEMENT CONFIGURATION	57
7.4	NETWORK INFRASTRUCTURE CHECKLIST	64
8.	MAIL CLIENT SECURITY	66
8.1	SECURING INSTALLING, CONFIGURING, AND USING CLIENT APPLICATIONS	66
8.2	SECURE MESSAGE COMPOSITION	69
8.3	PLUG-INS	70
8.4	ACCESSING WEB-BASED EMAIL SYSTEMS.....	70
8.5	MAIL CLIENT (MUA) SECURITY CHECKLIST.....	71
9.	SECURELY ADMINISTERING A MAIL SERVER.....	73
9.1	LOGGING.....	73
9.2	MAIL SERVER BACKUP PROCEDURES.....	76
9.3	RECOVERING FROM A SECURITY COMPROMISE	78
9.4	SECURITY TESTING MAIL SERVERS.....	80
9.5	REMOTELY ADMINISTERING A MAIL SERVER	82
9.6	SECURELY ADMINISTERING A MAIL SERVER CHECKLIST	83
	APPENDIX A. GLOSSARY	A-1
	APPENDIX B. MAIL-RELATED RFCS.....	B-1
	APPENDIX C. REFERENCES	C-1
	APPENDIX D. EMAIL SECURITY TOOLS AND APPLICATIONS	D-1
	APPENDIX E. SECURING MICROSOFT EXCHANGE	E-1
E.1	EXCHANGE SERVER INSTALLATION.....	E-1
E.2	ADMINISTRATIVE PERMISSIONS	E-4
E.3	EXCHANGE CORE COMPONENT ADMINISTRATION	E-5
E.4	SECURELY CONFIGURING EXCHANGE'S INTERNET MAIL SERVICE (SMTP)	E-10
E.5	SECURELY CONFIGURING POP3.....	E-12
E.6	SECURELY CONFIGURING IMAP	E-13
E.7	SECURELY CONFIGURING LDAP	E-14
E.8	CONFIGURING AUTHENTICATED MAIL RELAY	E-15
E.9	SECURELY CONFIGURING WEB ACCESS	E-15
	APPENDIX F. SECURING LINUX AND UNIX MAIL SERVICES	F-1
F.1	SECURING SENDMAIL	F-1
F.2	CONFIGURING POP AND IMAP TO USE SSL/TLS	F-10
F.3	CONFIGURING IMP WEB BASED EMAIL SERVER	F-13
F.4	USING PROCMAIL MAIL FILTER APPLICATION	F-14
	APPENDIX G. ONLINE SECURITY RESOURCES	G-1
	APPENDIX H. PRINT RESOURCES	H-1

APPENDIX I. WEB SERVER SECURITY CHECKLIST	I-1
APPENDIX J. ACRONYM LIST	J-1
INDEX	INDX-1

Figures and Tables

Figure 2.1: Example of Message Flow	5
Figure 2.2: SMTP Commands	7
Figure 2.3: Sample SMTP Conversation	8
Figure 2.4: Sample ESMTP Conversation.....	9
Table 2.1: SMTP Extensions	9
Figure 2.5: POP 3 Commands	11
Figure 2.6: IMAP 4 Revision 1 Commands	12
Table 2.2: IMAP Extension RFC Documents.....	12
Table 3.1: PGP Internet Resources	15
Table 3.2: Recommend PGP Cipher Suites	16
Table 3.3: S/MIME CAs.....	20
Figure 6.1: Virus Scanning Implemented on Firewall	42
Figure 6.2: Virus Scanning Implemented on Mail Server	43
Figure 6.3: Virus Scanning Implemented on User Workstations.....	45
Figure 6.4: Sendmail Blacklist Configuration from sendmail.cf	49
Figure 6.5: Sendmail Access Configuration	49
Figure 6.6: Sendmail TLS Configuration Example from sendmail.mc	50
Figure 7.1: Simple Single Firewall DMZ.....	54
Figure 7.2: Two Firewall DMZ.....	55
Figure 7.3: Three Interface Firewall DMZ	55
Figure 7.4: Mail Gateway	57

Executive Summary

Electronic mail (email) is perhaps the most popularly used system for exchanging information over the Internet (or any other computer network). At the most basic level, the email process can be divided into two principal components: (1) mail servers, which are hosts that deliver, forward, and store mail; (2) clients which interface with users and allow users to read, compose, send, and store email messages. This document addresses the security issues of both mail servers and mail clients.

After Web servers, mail servers are the hosts on an organization's network that are most often targeted by attackers. Because the computing and networking technology that underpins email is ubiquitous, it is well understood and attackers are able to develop attack methods to exploit the technology. Mail servers are also targeted because they (and public Web servers) must communicate to some degree with untrusted third-parties. Additionally, email clients have been targeted as an effective means of inserting malicious code into machines and of propagating this code to other machines. As a result, mail servers, mail clients, and the network infrastructure that supports them must be protected. Examples of email security issues include the following:

- Flaws in the mail server application may be used as the means of compromising the underlying server and hence the attached network. Examples of this unauthorized access include gaining access to files or folders that were not meant to be publicly accessible or being able to execute commands and/or install software on the mail server.
- Denial of service (DoS) attacks may be directed to the mail server or its support network infrastructure denying or hindering valid users an ability to use the mail server.
- Sensitive information on the mail server may be read by unauthorized individuals or changed in an unauthorized manner.
- Sensitive information transmitted unencrypted between mail server and email client may be intercepted. For example, all popular email communication standards default to sending usernames, passwords, and the email message itself "in the clear" (i.e., unencrypted).
- Information within the email may be altered at some point between the sender and recipient.
- Malicious entities may gain unauthorized access to resources elsewhere in the organization's computer network via a successful attack on the mail server. For example, once the mail server is compromised, an attacker will be able to retrieve users' passwords, which may grant the attacker access to other hosts on the organization's network.
- Malicious entities may attack external organizations from a successful attack on a mail server host, thus concealing the intruders' identities, and perhaps making the organization liable for damages.
- Misconfiguration may allow malicious entities to use the organization's mail server to send email-based advertisements (i.e., spam).
- Viruses and other types of malicious code may be distributed throughout an organization via email.

- Users may send inappropriate, proprietary, or other sensitive information via email. This could expose the organization to legal action.

This document is intended to assist organizations in installing, configuring, and maintaining secure mail servers and mail clients. More specifically this document discusses in detail the following items:

- Email standards and their security implications
- Email-related encryption standards
- Email-specific aspects of securing the underlying operating system
- Securing mail server applications
- Filtering email content
- Email specific considerations in the deployment and configuration of network protection mechanisms:
 - Firewalls
 - Routers
 - Switches
 - Intrusion detection systems
- Securing mail clients
- Administrating the mail server in a secure manner:
 - Backups
 - Security testing
 - Updating and patching
 - Log reviews
 - Records management/archiving email.

The following key guidelines are recommended to Federal departments and agencies for maintaining a secure mail server.

Organizations should carefully plan and address the security aspects of the deployment of a mail server.

As it is much more difficult to address security once deployment and implementation have occurred, security should be considered from the initial planning stage. Organizations are more likely to make decisions about configuring computers appropriately and consistently when they develop and use a detailed, well-designed deployment plan. Developing such a plan will support

mail server administrators in making the inevitable tradeoff decisions between usability, performance, and risk.

Organizations often fail to take into consideration the human resource requirements for both deployment and operational phases of the mail server and supporting infrastructure. Organizations should address the following points in a deployment plan:

- Types of personnel required (e.g., system and mail server administrators, network administrators, information systems security officers [ISSO])
- Skills and training required by assigned personnel
- Individual (level of effort required of specific personnel types) and collective manpower (overall level of effort) requirements.

Organizations should implement appropriate security management practices and controls when maintaining and operating a secure mail server.

Appropriate management practices are essential to operating and maintaining a secure mail server. Security practices entail the identification of an organization's information system assets and the development, documentation, and implementation of policies, standards, procedures and guidelines that help to ensure confidentiality, integrity, and availability of information system resources.

To ensure the security of a mail server and the supporting network infrastructure, the following practices should be implemented:

- Organizational-wide information system security policy
- Configuration/change control and management
- Risk assessment and management
- Standardized software configurations that satisfy the information system security policy
- Security awareness and training
- Contingency planning, continuity of operations, and disaster recovery planning
- Certification and accreditation.

Organizations should ensure that the mail server operating system is deployed, configured, and managed to meet the security requirements of the organization.

The first step in securing a mail server is securing the underlying operating system. Most commonly available mail servers operate on a general-purpose operating system. Many security issues can be avoided if the operating systems underlying mail servers are configured appropriately. Default hardware and software configurations are typically set by vendors to emphasize features, functions, and ease of use at the expense of security. Because vendors are not aware of each organization's security needs, each mail server administrator must configure new servers to reflect their organization's security requirements and reconfigure them as those requirements change. Securing the operating system would generally include the following steps:

- Patch and upgrade the operating system
- Remove or disable unnecessary services and applications
- Configure operating system user authentication
- Configure resource controls
- Security testing the operating system.

Organizations should ensure that the mail server application is deployed, configured, and managed to meet the security requirements of the organization.

In many respects, the secure install and configuration of the mail server application will mirror the operating system process discussed above. The overarching principle, as before, is to install the minimal amount of mail server services required and eliminate any known vulnerabilities through patches or upgrades. If the installation program installs any unnecessary applications, services, or scripts, they should be removed immediately after the installation process completes. Securing the mail server application would generally include the following steps:

- Patch and upgrade the mail server application
- Remove or disable unnecessary services, applications, and sample content
- Configure mail server user authentication
- Configure mail server resource controls
- Security testing the mail server application.

Organizations should consider the implementation of encryption technologies to protect user authentication and mail data.

Most standard mail protocols default to unencrypted user authentication and send email data in the clear (unencrypted). Sending this data in the clear may allow an attacker to easily compromise a user account and/or intercept and alter unencrypted emails. At a minimum, most organizations should encrypt the user authentication session even if they do not encrypt the email data itself. Encrypted user authentication is now supported by most standard and proprietary mailbox protocols.

The issues involved with encrypted email data are more complex. Encrypting email places a greater load on the user's computer and the organization's network infrastructure, may complicate virus scanning and mail content filtering, and often requires significant administrative overhead. However, for many organizations the benefits of email encryption will outweigh the costs.

Organizations should employ their network infrastructure to protect their mail server(s).

The network infrastructure (e.g., firewalls, routers, intrusion detection systems) that supports the mail server plays a critical role in the security of the mail server. In most configurations, the network infrastructure will be the first line of defense between the Internet and a mail server. Network design alone, however, cannot protect a mail server. The frequency, sophistication, and

even variety of mail server attacks perpetrated today, support the idea that mail server security must be implemented through layered and diverse protection mechanisms (layered protection).

Maintaining the security of a mail server is an ongoing process.

Maintaining a secure mail server requires constant effort, resources, and vigilance from an organization. Thus securely administering a mail server on a daily basis is an essential aspect of mail server security. Maintaining the security of a mail server will usually involve the following steps:

- Configuring, protecting, and analyzing log files
- Backing up data frequently
- Protecting against malicious code (e.g., viruses, worms, Trojans)
- Establishing and following procedures for recovering from compromise
- Testing and applying patches in a timely manner
- Testing security periodically.

1. Introduction

1.1 Authority

This document has been developed by the National Institute of Standards and Technology (NIST) in furtherance of its statutory responsibilities under the Computer Security Act of 1987 and the Information Technology Management Reform Act of 1996, specifically 15 United States Code (U.S.C.) 278 g-3 (a)(5)). This is not a guideline within the meaning of 15 U.S.C 278 g-3 (a)(3).

These guidelines are for use by Federal organizations that process sensitive information. They are consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Appendix III.

This document may be used by nongovernmental organizations on a voluntary basis. It is not subject to copyright.

Nothing in this document should be taken to contradict standards and guidelines made mandatory and binding on Federal agencies by the Secretary of Commerce under his statutory authority. Nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, the Director of the Office of Management and Budget, or any other Federal official.

1.2 Purpose and Scope

The purpose of the Guidelines on Electronic Mail Security is to recommend security practices for designing, implementing, and operating email systems on public and private networks. While intended as recommended guidance for federal departments and agencies, it may be used in the private sector on a voluntary basis.

Mail servers are often the most targeted and attacked servers on an organization's network – second only to Web servers. Various types of mail content and attachments have also proven to be effective in introducing malicious code into a network through the email client. This document may be used by organizations interested in enhancing security on existing and future email systems to reduce the number and frequency of email related security incidents. This document presents generic principles that apply to all systems and provides specific examples for two mail product implementations (i.e., Microsoft Exchange and sendmail¹) common to Windows and Unix operating environments.²

This guideline does NOT cover the following aspects relating to securing a mail server:

- Securing other types of network servers.

¹ Please note this document addresses Exchange and sendmail as these are two popular mail server applications. However, that does not mean these are the best choice or most secure for all organizations in all instances. Some organizations may wish to use other mail servers that are not directly addressed in this document.

² Although this document does not provide an appendix on the specific security configuration settings for Lotus Notes, some suggestions can be found at <http://domino-11.prominic.com/A55876/Rwebv6ASC.nsf?opendatabase>.

- Firewalls and routers used to protect mail servers beyond a basic discussion in Section 7.3.1.
- Special considerations for high traffic mail servers with multiple hosts.
- Securing backend servers that may support the mail server (e.g., syslog hosts, file servers).
- Security of the X.400 standard messaging protocol.

1.3 Audience and Assumptions

The document, while technical in nature, provides the background information to help readers understand the topics that are discussed. The intended audience for this document includes the following:

- **Users** when setting up email clients and accessing email.
- **System engineers** and **architects** when designing and implementing email systems.
- **System administrators** when administering or upgrading email systems.
- **Program managers** and **information technology (IT) security officers** to ensure that adequate security measures have been considered for all phases of the system's life cycle.

The practices recommended in this document are designed to help mitigate the risks associated with email and other known security problems. They build on and assume the implementation of practices described in the following NIST guidelines as appropriate:

- NIST Special Publication 800-3, *Establishing a Computer Security Incident Response Capability*
- NIST Special Publication 800-18, *Guide to Developing Security Plans for Information Technology System*
- NIST Special Publication 800-26, *Security Self-Assessment Guide for Information Technology Systems*
- NIST Special Publication 800-27, *Engineering Principles for Information Technology Security*
- NIST Special Publication 800-28, *Guidelines on Active Content and Mobile Code*
- NIST Special Publication 800-31, *Intrusion Detection Systems*
- NIST Special Publication 800-32, *Introduction to Public Key Technology and the Federal PKI Infrastructure*
- NIST Special Publication 800-34, *Contingency Planning Guide for Information Technology Systems*
- NIST Special Publication 800-37, *Federal Guidelines for the Security Certification and Accreditation of Information Technology Systems*

- NIST Special Publication 800-40, *Procedures for Handling Security Patches*
- NIST Special Publication 800-41, *Guide to Firewall Selection and Policy Recommendations*
- NIST Special Publication 800-42, *Guideline on Network Security Testing*
- NIST Special Publication 800-43, *Guide to Securing Windows 2000 Professional*
- NIST Special Publication 800-44, *Guidelines on Securing Public Web Servers*
- NIST Special Publication 800-46, *Security for Telecommuting and Broadband Communications*
- NIST Special Publication 800-52, *Guidelines for the Selection and Use of Transport Layer Security Implementations*

All these guidelines and others can be found at the NIST Computer Security Resource Web site at <http://csrc.nist.gov/publications/nistpubs/index.html>.

1.4 Document Organization

- Section 1 provides the purpose, scope, and intended audience.
- Section 2 includes background information and standards relating to email.
- Section 3 contains information on two popular email encryption standards.
- Section 4 presents an overview on securing the underlying operating system of the mail server.
- Section 5 discusses securing the mail server application.
- Section 6 addresses protecting the mail server with the supporting network infrastructure.
- Section 7 provides information regarding email client security.
- Section 8 discusses the basics of securely administering a mail server on a daily basis.
- Appendix A defines terms used in this document.
- Appendix B lists relevant Request for Comment (RFC) documents.
- Appendix C lists references used in this document.
- Appendix D identifies email security tools and applications.
- Appendix E provides specific information regarding securing Microsoft Exchange.
- Appendix F contains specific information regarding securing sendmail and other Unix mail applications.
- Appendix G lists online mail server security resources.

2. Background and Standards

As of January 2000, there were an estimated 242 million Internet users worldwide³. Most of these 242 million users have electronic mail (email) accounts on one or more email systems. That is a huge leap since 1971 when Ray Tomlinson, a Department of Defense (DoD) researcher, sent the first ARPANET email message to himself.

The ARPANET, precursor to the Internet, was a United States (US) Advanced Research Project Agency (ARPA) project intended to develop a set of communications protocols to transparently connect computing resources in various geographical locations. Messaging applications were available on ARPANET systems; however, they could only be used for sending messages to users with local system accounts. Tomlinson modified the existing messaging system so that users could send messages to users on other ARPANET connected systems. After Tomlinson's modification was available to other researchers, email quickly became the most heavily used application on the ARPANET.

As the ARPANET evolved into the Internet, email remained one of the most heavily used applications for personal and business users. Since the ARPANET was initially a small and trusted community, there was little need for security. As the Internet grew in popularity, the need for security increased greatly. Unfortunately, the needed security was lacking because early email standards and implementations placed little emphasis on security. Maintaining compatibility with these standards presents a great challenge in securing email today.

2.1 Background

An understanding of how email messages are composed, delivered, and stored is helpful in understanding email security. For most email users, once a message is composed and sent, it leaves their system and magically appears in the intended recipient's inbox. This may seem simple but the handling and delivery of an email message can be as complex as that involving physical mail – with processing and sorting occurring at several intermediary locations before arriving at the final destination.

The process starts with message composition. The most basic email clients typically ask the user to provide the following: subject line, message body, and intended recipients. When these fields are completed and the user sends the message, the message is transformed into a specific standard format specified by RFC 822, *Standard for the Format of ARP Internet Text Messages*. At the most basic level, the two primary message sections are the header and the body. The header section contains the vital information about the message including origination date, sender, recipient(s), delivery path, subject, and format information. The body of the message contains the actual content of the message. Refer to RFC 822 for information on message headers (see Appendix B).

Once the message is translated into an RFC 822 formatted message, it can be transmitted. Using a network connection, the mail client, referred to as a mail user agent (MUA), connects to a mail transport agent (MTA) operating on the mail server. After initiating communication, the mail client provides the sender's identity to the server. Next, using the mail server commands, the

³ Current Internet population statistics, <http://www.commerce.net/research/stats/wwstats.html>.

client tells the server who the intended recipients are. Though the message contains a list of intended recipients, the mail server does not examine the message for this information. Only after the complete recipient list is sent to the server does the client supply the message. From this point, message delivery is under control of the mail server.

Once the mail server is processing the message, several events occur: recipient server identification, connection establishment, and message transmission. Using Domain Name Services (DNS), the sender's mail server determines the mail server(s) for the recipient(s). Then, the server opens up a connection(s) to the recipient mail server(s) and sends the message employing a process similar to that used by the originating client. At this point, one of two events could occur. If the sender's and recipient's mailboxes are located on the same mail server, the message is delivered using a local delivery agent (LDA). If the sender's and recipients mailboxes are located on different mail servers, the send process is repeated from one MTA to another until the message reaches the recipient's mailbox.

When the LDA has control of the message, a number of possible events may occur. Depending on the configuration, the LDA could deliver the message or process the message based on a predefined message filter before delivery (filtering can be based on a number of message properties and will be discussed in detail in Section 6.2.2). Once the message is delivered, it is placed in the recipient's mailbox where it is stored until the recipient performs some action on it (e.g., read, delete) using the MUA. Figure 2.1 illustrates the flow of the message through the various mail components discussed previously. This is the general process of sending an email.

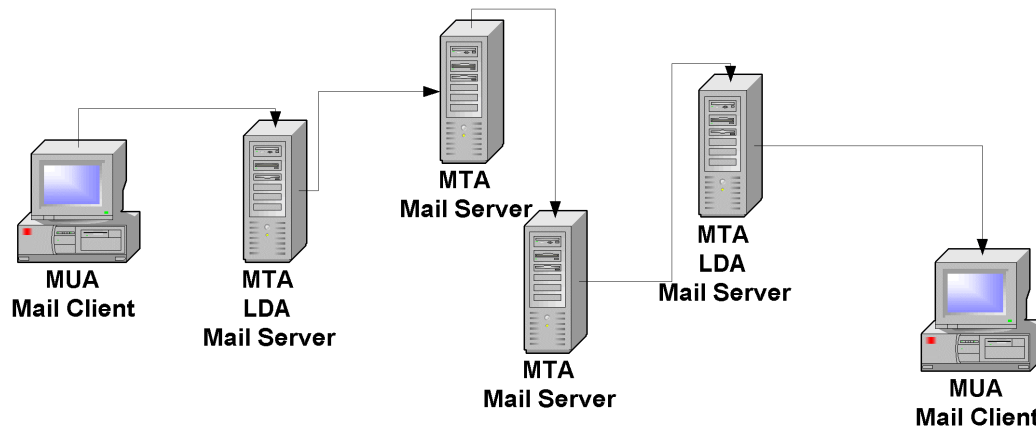


Figure 2.1: Example of Message Flow

2.2 Multipurpose Internet Mail Extensions

RFC 822 provides a standard for transmitting messages containing textual content; however, it does not address messages that contain attachments (such as a mail message with a word processing document or photo included). Instead of redefining RFC 822, Multipurpose Internet Mail Extensions (MIME) were developed. Making use of the headers in an RFC 822 message, the MIME extensions provide almost endless possibilities for message content. MIME uses the convention of content-type/subtype pairs to specify the native representation or encoding of associated data. Examples of content types include the following:

- **Audio** – for transmitting audio or voice data.

- **Application** – used to transmit application data or binary data.
- **Image** – for transmitting still image (picture) data.
- **Message** – for encapsulating another mail message.
- **Multipart** – used to combine several message body parts, possibly of differing types of data, into a single message.
- **Text** – used to represent textual information in a number of character sets and formatted text description languages in a standardized manner.
- **Video** – for transmitting video or moving image data, possibly with audio as part of the composite video data format.

The current MIME standards include five parts: RFCs 2045, 2046, 2047, 2048, and 2049 (see Appendix B). They address message body format, media types, non-American Standard Code for Information Exchange (non-ASCII) message header extensions, registration procedures, and conformance criteria. With this added functionality, email features such as message attachments and inline hypertext markup language (HTML) are possible. Note that though MIME extensions allow for binary message content, such content is incorporated into an RFC 822 message using a Base64 encoding, which provides a textual representation of binary data.⁴

2.3 Mail Transport Standards

To ensure reliability and interoperability among various email applications, mail transport standards were established. In the simplest scenario, an email message is sent from one local user to another local user. For this case, an LDA is responsible for placing the message in the appropriate mailbox. When a message is sent to non-local recipients, an MTA is needed to send the message from the local mail server to the remote mail server. Depending on the type of systems involved, different MTAs may be used, which in turn may support different implementations of a particular message transfer protocol or more than one distinct transfer protocol.

The most common MTA transfer protocol is the Simple Mail Transfer Protocol (SMTP). SMTP is the de-facto Internet standard for sending email messages. Thus, any Internet messaging system must support SMTP to facilitate communication with other email messaging applications. Other messaging systems exist that use different MTA transfer protocols between similar or clustered messaging systems. For the most part, these MTAs are proprietary and work only with specific systems. Sections 2.4 and 2.5 will provide the necessary background information on SMTP, SMTP extensions, and other proprietary MTAs.

2.4 Simple Mail Transfer Protocol

Mr. Jon Postel of the University of Southern California developed SMTP in August 1982. As RFC 821 states, “SMTP was developed to ensure a more reliable and efficient way to transport

⁴ Base64 encoding was originally derived from RFC 1421 for Privacy Enhanced Mail (PEM).

messages.” At the most basic level, SMTP is a minimal language that defines a communications protocol for delivering email messages. Figure 2.2 lists the SMTP commands and syntax.

```
HELO <SP> <domain> <CRLF>
This command (hello) is used to connect to the server as specified in <domain>.
MAIL <SP> FROM:<reverse-path> <CRLF>
This command is used to tell the server the sender's identity as specified in <reverse-path>.
RCPT <SP> TO:<forward-path> <CRLF>
This command (recipient) is used to tell the server the identity of an intended recipient as
specified in <forward-path>.
DATA <CRLF>
This command is used to convey the message body to the server.
RSET <CRLF>
This command (reset) is used to reset the server connection.
SEND <SP> FROM:<reverse-path> <CRLF>
This command is used to tell the mail server the return address of the sender as specified in
<reverse-path>.
SOML <SP> FROM:<reverse-path> <CRLF>
This command (send or mail) is used to deliver email to one or more workstations or
recipients if the user is not active.
SAML <SP> FROM:<reverse-path> <CRLF>
This command (send and mail) is used to deliver email to one or more workstations and
recipients if the user is not active.
VRFY <SP> <string> <CRLF>
This command (verify) is used to ask the receiver to confirm that a user has been identified.
EXPN <SP> <string> <CRLF>
This command (expand) is used to ask the receiver to confirm that a mailing list has been
identified.
HELP [<SP> <string>] <CRLF>
This command is used to obtain help information.
NOOP <CRLF>
This command is used to indicate no operation, but signify the sender is still connected
(i.e., “alive”).
QUIT <CRLF>
This command is used to close the server connection.
TURN <CRLF>
This command is used to ask the receiver to send a valid reply and then become the SMTP
sender, or else ask the receiver to send a refusal reply and remain the SMTP receiver.
```

Figure 2.2: SMTP Commands

When a user sends an email, the client, or MUA contacts its SMTP server and conducts a “conversation” using the SMTP language. A MUA is typically part of the mail client application (e.g., Outlook, Eudora). If a MUA is unavailable, mail messages can be sent using a telnet client connected to the SMTP service. Figure 2.3 depicts a sample SMTP conversation using telnet. The telnet and SMTP commands entered by the user for this session are shown in bold. During a manual SMTP telnet session, the HELP command can be used to determine which of the SMTP commands are enabled on the server.

```

telnet mail.nowhere.com 25
Connected to mail.nowhere.com.
Escape character is '^]'.
220 test.mail.com SMTP Service (Sample Mail Server String)
HELO test.mail.com
250 test.mail.com
MAIL FROM: jdoe@nowhere.com
250 Sender <jdoe@nowhere.com> Ok
RCPT TO: jsmith@somewhere.com
250 Recipient <jsmith@somewhere.com> Ok
DATA
354 Ok Send data ending with <CRLF>.<CRLF>
Hello World!
.
250 Message received: GM1BAR00.F4M
QUIT
221 mail.nowhere.com SMTP server closing connection.
Connection closed by foreign host.

```

Figure 2.3: Sample SMTP Conversation

2.5 Simple Mail Transfer Protocol Extensions

As the number of email users grew, additional functionality was sought in mail clients and SMTP servers. For SMTP servers to support this additional functionality, extensions were added to SMTP. In 1993, RFC 1425 introduced the concept of SMTP service extensions. Subsequently RFC 1425, was superseded by RFC 1651 in 1994 and RFC 1869 in 1995. These RFCs added three pieces to the SMTP framework:

- New SMTP commands (RFC 1425)
- Registry for SMTP service extensions (RFC 1651)
- Additional parameters for SMTP MAIL FROM and RCPT TO commands (RFC 1869)

To be compatible with older SMTP servers, there needed to be a method to allow the mail client application to determine whether the server supported extensions. This was accomplished through the “enhanced hello” (EHLO) command. When connecting to a server, a mail client could issue the EHLO command. If the server supported SMTP extensions, it would give a successful response and list the extensions that were supported. If the server did not support SMTP extensions, it would issue a command failure response prompting the MUA to respond with the standard HELO command. Servers that support SMTP extensions, also known as Extended SMTP (ESMTP),

typically respond with ESMTP in their banner.⁵ Figure 2.4 is an excerpt from an ESMTP server transaction involving the EHLO command.

```
telnet mail.nowhere.com 25
Connected to mail.nowhere.com.
Escape character is '^]'.
220 test.mail.com ESMTP Service (Sample Mail Server String)
EHLO test.mail.com
250 test.mail.com says hello
250-HELP
250-EXPN
250 SIZE 20971520
...
```

Figure 2.4: Sample ESMTP Conversation

As depicted in Figure 2.4, the sample server supports only one extension – SIZE. Numerous extensions are supported by a variety of SMTP servers. Table 2.1 lists some of the more common SMTP extensions and their associated RFCs. In particular, RFC 2554 specifies a new command and protocol for identifying and authenticating a user. The default configuration of most mail servers typically does not have authenticated relay enabled.

Table 2.1: SMTP Extensions

SMTP Extension	Associated RFC(s)
SMTP Service Extension for Message Size Declaration	1870
SMTP Service Extension for Command Pipelining	2920
SMTP Service Extensions for Transmission of Large and Binary MIME Messages	3030
SMTP Service Extension for Authentication	2554
SMTP Service Extension for Secure SMTP over TLS	2487
SMTP Service Extension for Returning Enhanced Error Codes	2034
SMTP Service Extension for Remote Message Queue Starting	1985
SMTP Service Extension for Delivery Status Notifications	1891

⁵ Many application services (e.g., email, Web, File Transfer Protocol) that operate on a server respond to a client request with a banner. This banner is a text message that contains information on the server such as application and operating system type and version. This information can be useful to attackers and should be changed as discussed later in the document. Most mail clients do not display this banner to the end user.

2.6 Proprietary Mail Transports

As mentioned previously, some messaging systems use MTAs that do not support either SMTP or ESMTP. These types of MTAs are designed to work within a closed messaging environment. Many large-scale government, academic, and private organizations have messaging systems that use these types of MTAs. However, these organizations still rely on SMTP or ESMTP capable MTAs for communicating with external messaging systems. Some examples of messaging systems that use proprietary protocols are MTAs for Lotus Notes, cc:Mail, and Microsoft Exchange.⁶ Discussion on the benefits and disadvantages of using MTAs that support only proprietary message transfer protocols is outside the scope of this document.

2.7 Client Access Standards

Once a message is delivered by the LDA, users need to access the mail server to retrieve the message. Mail clients (MUAs) are used to access the mail server and retrieve mail messages. Several methods exist for users to access their mailboxes, the simplest being direct access.

The simplest scenario for a messaging system would be one in which all users have direct access to their mailbox (common on hosts employing the Unix operating system). For each account that exists on the system, there is a corresponding mailbox in that user's home directory. When messages are received, users can use command-line based mail programs, such as "mail" or "pine," to directly access the mailbox. Although this method is straightforward, it requires all users accessing the mail server to receive messages to have a user account and a command-line interface on the host operating system.

Allowing users, particularly external users, to have access to a command-line interface is a significant security risk. To mitigate this risk, mailbox access protocols were devised. The two most widely supported mailbox access protocols are Post Office Protocol (POP) and Internet Message Access Protocol (IMAP). As with message transfer protocols for MTAs, other proprietary mailbox access protocols exist that are regularly used by commercial software vendors. It is important to understand that POP, IMAP, and indeed most proprietary protocols, in their default configuration use cleartext passwords for authentication, which can be intercepted by other hosts attached to the network.

2.8 Post Office Protocol

The POP was first developed in 1984. At its core, POP was nothing more than a way to copy messages from the mail server mailbox to the mail client. It worked much like a traditional post office mailbox. The mail client opens a connection to the mail server mailbox, downloads the mail messages, and then closes the connection. As described in RFC 918, only nine commands were originally available for POP (see "Basic Commands" in Figure 2.5).

⁶ Note: All these product families have SMTP MTAs as well as their proprietary MTAs.

Basic Commands	
USER <name>	Set username
PASS <password>	Set password
STAT	Check the status of the mailbox, typically gets number of messages
LIST [msg]	Lists messages in the mailbox; Optional argument for message [msg]
RETR <msg>	Retrieve message <msg>
DELE <msg>	Delete message <msg>
QUIT	Quit
NOOP	No operation
RSET	Reset
Optional Commands	
TOP <msg> <n>	Retrieve the top <n> lines of message <msg>
UIDL [msg]	Retrieve unique id for [msg]
APOP <name> <digest>	A more robust form of authentication than USER/PASS; see RFC 1939

Figure 2.5: POP 3 Commands

Since 1984, POP has gone through several changes and is now in its third iteration as defined in RFC 1939. The basic command set is very similar to the command set of 1984; however, POP version 3 offers a few new optional commands, listed in Figure 2.5. From a security standpoint, the addition of the APOP was important, since it avoids transmitting a user's password in the clear. Instead, a challenge/response mechanism is used, by which the client responds with a cryptographic hash of the combined challenge sent from the server and the user's password, for verification by a POP mail server performing the same operation for the user in question.

The POP mailbox access standard has some significant limitations. Typically, when users retrieve their mail, copies of the messages that reside on the server are deleted. This means that the user has the sole responsibility of maintaining message archives. Although this may be acceptable for personal accounts, it is generally unacceptable for most commercial or governmental organizations that have to meet certain legal requirements. In addition, if a user employs several workstations for retrieving mail, the messages are dispersed on multiple systems. POP may be configured so that the original messages are not deleted from the server. However, the user will have to either download all of the messages previously viewed as well as the new messages when accessing a mailbox from another system, or set up a retention period after which messages are automatically deleted from the server.

2.9 Internet Message Access Protocol

To address the above-mentioned issues with POP, IMAP was developed in 1988. The IMAP protocol was developed as a functional superset of the POP version 2 protocol. At the most basic level, IMAP was designed so user mailboxes could be centrally located and accessed from multiple email clients or MUAs.

Initially, IMAP offered very little functionality beyond that of POP, but since 1988, it has evolved into a robust mailbox access protocol. The most current edition of the IMAP standard is RFC 2060: *Internet Message Access Protocol – Version 4, Revision 1* (4rev1). Because IMAP 4rev1 supports many different features, it has a much wider command set than that of POP. Figure 2.6 provides a list of IMAP 4rev1 commands. Additionally,

with the CAPABILITY command, the IMAP server can be queried to determine if other IMAP extensions are supported.

NOOP	Perform no operation
AUTHENTICATE <type>	Choose authentication method
LOGIN <user> <passwd>	Login with username and password
LOGOUT	Logout the current user
SELECT <mailbox>	Select the desired mailbox to access
EXAMINE <mailbox>	Same as SELECT except opens mailbox read-only
CREATE <mailbox>	Create a mailbox with the name <mailbox>
DELETE <mailbox>	Delete selected mailbox
RENAME <mailbox> <newmailbox>	Rename mailbox
SUBSCRIBE <mailbox>	Subscribe to selected mailbox
UNSUBSCRIBE <mailbox>	Unsubscribe from selected mailbox
LIST <reference> [pattern]	List contents of current reference based on an optional pattern
LSUB <reference> [pattern]	Lists a set of mailboxes matching the pattern
STATUS <mailbox> <item>	Shows the status of specific items in the selected mailbox
APPEND <mailbox> [flags] <msg>	Append a message to the selected mailbox
CHECK	Performs a checkpoint on the currently selected mailbox
CLOSE	Closes the currently selected mailbox
EXPUNGE	Expunges deleted messages from the mailbox
SEARCH <criteria>	Searches the mailbox based on a certain criteria
FETCH <message> <item>	Fetches the specified item from the selected message
STORE <message> <item> <newvalue>	Updates the selected item in a message
COPY <message> <mailbox>	Copies a message to the provided mailbox
UID <command> [args]	Perform an operation on a message based on its UID
CAPABILITY	Query the server for its capabilities

Figure 2.6: IMAP 4 Revision 1 Commands

Table 2.2 lists the associated RFCs for the noted IMAP extensions. As with POP, IMAP has been extended with a challenge/response mechanism comparable to APOP, which is called the Challenge-Response Authentication Mechanism (CRAM). CRAM requires the client to make note of the challenge data sent by the server and respond with a string consisting of the user's name, a space, and a digest computed by applying the keyed Message Digest 5 (MD5) algorithm⁷ against the timestamp sent with the challenge, using a shared secret as the key.

Table 2.2: IMAP Extension RFC Documents

IMAP Extension	Associated RFC(s)
IMAP4 QUOTA extension	2087
IMAP4 non-synchronizing literals	2088
IMAP4 IDLE command	2177
IMAP4 Multi-Accessed Mailbox Practice	2180
IMAP URL Scheme	2192
IMAP4 Mailbox Referrals	2193

⁷ A popular one-way hash function developed by Ronald Rivest that is used to create a message digest for digital signatures. MD5 is faster than SHA-1, but is considered less secure.

IMAP Extension	Associated RFC(s)
IMAP/POP AUTHorize Extension for Simple Challenge/Response	2195
IMAP4 Login Referrals	2221
IMAP4 Namespace	2342
IMAP4 UIDPLUS extension	2359
IMAP4 ID extension	2971

2.10 Proprietary Mailbox Access Mechanisms

Proprietary mailbox access protocols are designed to work within closed messaging environments. Exchange, Lotus Notes, and cc:Mail are some examples of messaging systems that use proprietary mailbox access protocols. These proprietary protocols offer additional functionality when used with their associated clients. Nearly all proprietary messaging systems support standard protocols, including SMTP, POP, and IMAP, in order to interoperate with other types of MTAs and MUAs. Organizations must decide for themselves whether it is appropriate to support proprietary protocols in their email clients and servers. As mentioned earlier, regardless of whether they are standard or proprietary, most access protocols default to weak authentication mechanisms (unencrypted authentication information). Therefore, organizations need to configure the access protocols to support stronger forms of authentication.

3. Email-Related Encryption Standards

The two primary mechanisms for securing email content end-to-end are Pretty Good Privacy (PGP) and Secure Multipurpose Internet Mail Extensions (S/MIME). Both are based, in part, on the concept of public key cryptography, which involves a user having a pair of related keys: a public key that anyone can hold, and a private key that is held exclusively by its owner. The recipient's public key is used for sending encrypted information that can be decrypted only with the private key. The sender's private key is used for sending digitally signed information whose authenticity can be verified by anyone holding the corresponding public key. Digital signature techniques rely on the creation of a digest or fingerprint of the information (i.e., the message being sent) using a cryptographic hash, which can be signed more efficiently than the entire message.⁸

Because of the computationally intensive operations involved in public key cryptography, more efficient, symmetric key cryptography is also used in securing email. Symmetric key cryptography requires a single key to be shared between communicating parties, the sender and recipient of an email message. The process typically requires the sender to generate a random key and encrypt the message with it using a symmetric key encryption algorithm. Then, the sender encrypts the symmetric key, using the recipient's public key with a corresponding public key encryption algorithm, and sends both the encrypted message and encrypted symmetric key to the recipient. This hybrid process uses public key encryption only to encrypt the symmetric key. Because only the intended message recipient holds the corresponding private key that is needed to recover the symmetric key, no other party can decrypt the message and read it.

Although S/MIME and PGP are two of the most prevalent email encryption mechanisms used today, many mechanisms have been proposed since the invention of email. Two of these mechanisms were PEM, first developed in 1987, and MIME Object Security Services (MOSS). Since neither method is broadly used today, they are not discussed in this document.

Although encrypting email provides additional security, it does come at a cost and many organizations should carefully weigh the issues associated with encrypting email:

- Virus scanning and mail content filtering at the firewall and mail server is made significantly more complicated by encryption. If the firewall or mail server does not have a method for decrypting the email, it cannot read the contents for viruses or content filtering.
- Encryption and decryption requires processor time. Organizations that have marginal computers may not want to institute encryption until their computers have been updated.

⁸ For more detailed information on public key cryptography, please refer to NIST Special Publication 800-32, *Introduction to Public Key Technology and the Federal PKI Infrastructure*, (<http://csrc.nist.gov/publications/nistpubs/>).

- Encrypted emails are often larger and therefore require more network bandwidth. The actual increase will depend on a number of factors including the encryption algorithm used, the key size, message size, and the number of recipients.
- Organizational wide encryption can require significant ongoing administrative overhead. Examples of this would include key distribution, key recovery, and revocation of encryption keys.

3.1 Pretty Good Privacy

PGP was first released in June 1991. Originally freeware, both free and commercial versions of PGP have become available. PGP can be downloaded or purchased from a variety of Web sites, including those shown in Table 3.1. OpenPGP is now defined by the Internet Engineering Task Force (IETF) OpenPGP Working Group standard RFC 2440.

Table 3.1: PGP Internet Resources

Organization	URL
International PGP Site	http://www.pgpi.org/
MIT PGP Freeware Distribution	http://web.mit.edu/network/pgp.html
PGP Site (Commercial Version)	http://www.pgp.com/
OpenPGP Site	http://www.openpgp.org

The current commercial version of PGP is 7.0, from the PGP corporation. This version supports a number of cryptographic algorithms recommended by NIST to the Federal Government, including the following:

- Data Encryption Standard (DES) in triple DES mode (3DES)⁹ for data encryption
- Advanced Encryption Standard (AES)¹⁰ for data encryption
- Digital Signature Algorithm (DSA)¹¹ for digital signatures
- RSA for digital signatures
- Secure Hash Algorithm (SHA-1)¹² for hashing.

⁹ For more information about DES and 3DES see <http://csrc.nist.gov/cryptval/>.

¹⁰ For more information about AES <http://csrc.nist.gov/encryption/aes/>.

¹¹ For more information about the DSA and the associated Digital Signature Standard (DSS) see <http://www.itl.nist.gov/fipspubs/fip186.htm>.

¹² For more information about the SHA and the associated Secure Hash Standard (SHS) see <http://csrc.nist.gov/cryptval/shs.html>.

Other versions of PGP may support other encryption schemes not addressed here. Federal organizations are generally required to use algorithms that have been approved by the Federal government. Other organizations may wish to choose encryption schemes approved by the Federal Government¹³ because these are well tested and secure. Many non-approved algorithms have been broken. Thus, non-approved algorithms (whether successfully broken or not) may represent a significant vulnerability to an organization that employs them. If an organization chooses PGP, they should apply the following guidelines as show in Table 3.2:

Table 3.2: Recommend PGP Cipher Suites

Recommended Use	Cipher Suites
Highest Security:	Encryption: Advanced Encryption Standard (AES) 256-bit encryption Authentication & Digest: Digital Signature Standard (DSS) with a key size of 1024 bits or higher and Secure Hash Algorithm-1 (SHA-1)
Security and Performance:	Encryption: AES 128-bit encryption Authentication & Digest: DSS with a key size of 1024 bits or higher and SHA-1
Security and Compatibility:	Encryption: Triple Data Encryption Standard (3DES) 168/112-bit encryption (note: 3DES is considerably slower than AES) ¹⁴ Authentication & Digest: DSS with a key size of 1024 bits or higher and SHA-1
Authentication and Tamper Detection	Authentication & Digest: DSS with a key size of 1024 bits or higher and SHA-1

Although certain aspects of PGP do use public key cryptography, such as digitally signed message digests, the actual encryption of the message body is performed with a symmetric key algorithm as outlined earlier. The following is a brief description of signing and encrypting a message with PGP (some steps may occur in a different order):

- PGP creates a random session key (in some implementations of PGP, users are required to move their mouse at will within a window to generate random data)
- Message is encrypted using the random session key, and a symmetric algorithm (e.g., 3DES, AES)
- Session key is encrypted using the recipient's public key
- SHA algorithm generates a message digest (cryptographic hash); and this hash is "signed" with the sender's private key creating a digital signature

¹³For a complete list of FIPS-approved algorithms see <http://csrc.nist.gov/cryptval/>

¹⁴ Please note that "single"(56-bit) DES is now considered insecure for most applications.

- Encrypted session key is attached to the message
- Message is sent to the recipient.

The recipient reverses the steps to recover the session key and decrypt the message. Popular email clients such as Netscape Messenger, Eudora, and Microsoft Outlook require the installation of plug-ins to enable the user to send and receive PGP encrypted messages. The PGP distribution sites listed in Table 3.1 contain instructions on how to use PGP with various email client applications.

3.2 S/MIME

S/MIME, which was originally proposed in 1995 by RSA Data Security, Inc., is based on their proprietary (although widely supported) Public Key Cryptography Standard (PKCS) #7 for data format of encrypted messages, and the X.509 version 3 standard for digital certificates. For more information about the RSA PKCS standards, consult the PKCS home page (see <http://www.rsasecurity.com/rsalabs/pkcs/index.html>).

S/MIME version 2 achieved wide adoption throughout the Internet mail industry. Although it is not a recognized (IETF) standard, it is specified by the following informational RFCs:

- S/MIME Version 2 Message Specification (RFC 2311)
- S/MIME Version 2 Certificate Handling (RFC 2312)
- PKCS #1: RSA Encryption Version 1.5 (RFC 2313)
- PKCS #10: Certification Request Syntax Version 1.5 (RFC 2314)
- PKCS #7: Cryptographic Message Syntax Version 1.5 (RFC 2315)
- Description of the RC2 Encryption Algorithm (RFC 2268)

S/MIME version 3 was developed by the IETF S/MIME Working Group and adopted as an IETF standard in July 1999. S/MIME version 3 is specified by the following RFCs:

- Cryptographic Message Syntax (RFC 2630)
- S/MIME Version 3 Message Specification (RFC 2633)
- S/MIME Version 3 Certificate Handling (RFC 2632)
- Diffie-Hellman Key Agreement Method (RFC 2631).
- Enhanced Security Services for S/MIME (RFC 2634)

The IETF S/MIME Working Group now coordinates all current development of the S/MIME standard. The homepage for the S/MIME Working Group can be found at <http://www.ietf.org/html.charters/smime-charter.html>.

Because S/MIME was originally developed in 1995, the S/MIME standard had to conform to the existing U.S. export controls for cryptography code. This meant that S/MIME implementations were forced to support the insecure 40-bit RC2 algorithm. These controls have since been relaxed. However, because of the standing requirement to support 40-bit RC2, S/MIME is often criticized as being “cryptographically weak.” This is only accurate if a weak algorithm is chosen. S/MIME is compatible with a number of encryption algorithms that allow it to support secure encryption. The actual process by which S/MIME-enabled mail clients send messages is similar to that of PGP.¹⁵

The most significant feature of S/MIME is its built-in and nearly “automatic” nature. Because of heavy industry involvement from organizations like Netscape and Microsoft, S/MIME functionality exists with default installations of popular email clients such as Netscape Messenger and Outlook/Outlook Express. Similar to PGP, no flaws have been discovered in the actual S/MIME protocol. However, as the following URL describes, S/MIME clients that encrypt data using 40-bit RC2 can now be cracked using the brute force method by Windows machines: <http://www.counterpane.com/smime.html>.

The version 3 of S/MIME supports two cryptographic algorithms recommended by NIST: DES and 3DES. There are also now drafts in the IETF for incorporating AES. In order to meet previous export restrictions and for backward compatibility, S/MIME supports: RC2 40 bit and RC2 64 bit.

In general, organizations should not use RC2 40 bit (weakest) or DES (weak) for sensitive mail or other transactions. Both of these encryption algorithms are very weak in today’s environment and should only be used when no other alternative exists (e.g., export restrictions, compatibility issues, etc.). RC2 64 bit is stronger than either DES or RC2 40 bit and it is faster than either DES or 3DES. However, it is significantly weaker than 3DES so its use should be limited to instances where compatibility is paramount. Algorithm performance is rarely an issue with S/MIME, since encryption and decryption usually takes place on a desktop. When security is paramount, 3DES is the strongest algorithm currently supported by S/MIME, though AES will soon be incorporated. Organizations that adopt S/MIME and wish maximum security should ensure that their S/MIME infrastructure is configured to use 3DES or AES.

3.3 Choosing an Appropriate Encryption Algorithm

Choosing an appropriate encryption algorithm depends on several factors that will vary with each organization. Although it might appear that the strongest encryption available should always be used, that is not always true. The higher the level of the encryption, the greater impact it will have on the mail client resources and communications speed (encryption can increase the size of an email considerably). In addition, a number of countries still maintain restrictions on the export, import, and/or use of encryption. In addition, patents and licensing issues may impact which encryption schemes can be used in a particular country. Finally, the choice of email encryption standard (PGP, S/MIME, etc.) may limit the choice of encryption algorithms. Fortunately, for Federal organizations, the choice is simple and clear – either 3DES or AES.

¹⁵ The following IBM Redbook provides a detailed example of how S/MIME works: <http://www.redbooks.ibm.com/pubs/pdfs/redbooks/sg245341.pdf>.

Overall, common factors that can influence the choice of an encryption algorithm include the following items:

- Required security
 - Value of the data to either the organization and/or other entities. The more valuable the data, the stronger the required encryption.
 - Time value of data. If data are valuable for only a short time period (e.g., days as opposed to years), then a weaker encryption algorithm can be used. An example would be passwords that are changed on a daily basis because the encryption needs to protect the password for only a 24-hour period.
 - Threat to data. The higher the threat level, the stronger the required encryption.
 - Other protective measures may reduce the need for stronger encryption. An example would be using protected methods of communications such as dedicated circuits instead of the public Internet.
- Required performance. Higher performance requirements may necessitate weaker encryption, but this is not normally a consideration with email.
- System resources. Less resources such as processor speed and memory size may necessitate weaker encryption, but are not typically a factor in email.
- Import, export, or usage restrictions.
- Encryption schemes supported by mail client applications and operating systems.

3.4 Key Management

The biggest difference between PGP and S/MIME is the key management model. The default and traditional model that PGP uses for key management is referred to as the “circle of trust,” which has no central key issuing or approving authority. The circle of trust relies on users for management and control. While this is suitable for individual users and very small organizations, the overhead of such a system is unworkable in medium to large organizations.

Conversely, S/MIME and some newer versions of PGP work on a classic, more hierarchical design. Typically, there is a master registration and approving authority, referred to as a Certificate Authority (CA), with subordinate local registration authorities. Table 3.3 shows a sample of third-party S/MIME CAs.

Table 3.3: S/MIME CAs

CA Name	URL
Baltimore	http://www.baltimore.com
Entrust	http://www.entrust.com
Verisign	http://www.verisign.com

By default, S/MIME enabled mail clients depend on the trust of their immediate master CA when processing S/MIME transactions. This authority can be either a third-party CA, such as those provided in Table 3.3, or a CA that is controlled by the organization issuing the certificates.

3.5 Choosing Between PGP and S/MIME

Choosing between PGP and S/MIME will depend on a number of factors. The newer commercial versions of PGP have added considerable functionality in order to compete with S/MIME, making the choice less clear-cut than before. Products implementing both standards also provide additional functionality such as disk or file encryption that can be used to protect information besides email on a host

The advantages of PGP are as follows:

- Suitable for small groups or single users
- More secure with support for AES, though S/MIME is not far behind
- Freeware versions available
- Does not require (but supports, if required) an external public key infrastructure (PKI) (S/MIME requires an organization to purchase certificates or setup their own certificate authority)
- Can be used, albeit with more steps, with any mail client application.

The advantages of S/MIME are as follows:

- Suitable for large groups and organizations
- Most widely compatible mail encryption standard
- Support is built into most major email client applications
- More transparent to the end-user.

4. Planning and Management of Mail Servers

The most critical aspect of deploying a secure mail server is careful planning prior to installation, configuration, and deployment. Careful planning will ensure that the mail server is as secure as possible and in compliance with all relative organizational policies. Many mail server security and performance problems can be traced back to a lack of planning or management controls. The importance of management controls is difficult to overstate. In many organizations, the information technology support structure is highly fragmented. This fragmentation leads to inconsistencies and these inconsistencies can lead to security vulnerabilities and other issues.

4.1 Planning the Installation and Deployment of the Mail Server

Security should be considered from the initial planning stage and the beginning of the systems development life cycle in order to maximize security and minimize costs. It is much more difficult and expensive to address security once deployment and implementation have occurred. Organizations are more likely to make decisions about configuring hosts appropriately and consistently when they begin by developing and using a detailed, well-designed deployment plan. Developing such a plan enables organizations to make informed tradeoff decisions between usability and performance, and risk. A deployment plan allows an organization to maintain a secure configuration and aids in identifying security vulnerabilities, which often manifest themselves as deviations from the plan.

In the planning stages of a mail server, the following items should be considered [SANS00a]:

- Identify the purpose(s) of the mail server.
 - What information categories will be processed on or transmitted through the mail server?
 - What are the security requirements for this information?
 - What other service(s) will be provided by the mail server (in general dedicating the host to being only a mail server is the most secure option)?
 - What are the security requirements for these additional services?
 - Where on the network will the mail server be located (see Section 7.1)?
- Identify the network services that will be provided on the mail server, such as those supplied through the following protocols.
 - SMTP
 - POP
 - IMAP

- Identify any network service software, both client and server, to be installed on the mail server and any other support servers.
- Identify the users or categories of users of the mail server and any support hosts.
- Determine the privileges that each category of user will have on the mail server and support hosts.
- Decide if and how users will be authenticated and how authentication data will be protected.
- Determine how appropriate access to information resources will be enforced.
- Determine which mail server application(s) will meet the organization's requirements. Consider less known servers that may offer greater security albeit with less functionality in some instances. Some issues to consider include:
 - Cost
 - Compatibility with existing infrastructure
 - Knowledge of existing employees
 - Existing vendor relationship
 - Past vulnerability history
 - Functionality
- Work closely with vendor(s) in the planning stage.

The choice of mail server application may determine the choice of operating system. However, to the degree possible, mail server administrators should choose an operating system that provides the following [CERT00]:

- Minimal exposure to vulnerabilities (which can be identified on all operating systems)
- Ability to restrict administrative or root level activities to authorized users only
- Ability to deny access to information on the server other than that intended to be available
- Ability to disable unnecessary network services that may be built into the operating system or server software
- Ability to log appropriate server activities to detect intrusions and attempted intrusions

In addition, organizations should consider the availability of trained experienced staff to administer the server and server products. Many organizations have learned the difficult lesson that a capable and experienced administrator on one type of operating environment is not automatically as effective on another.

Given the sensitive nature of the mail server, it is critical that it be located in an area that provides a secure physical environment. When planning the location of the mail server the following items should be considered:

- Does the proposed location offer the appropriate physical security protection mechanisms? Examples include:
 - Locks
 - Card-reader access
 - Security guards
 - Physical intrusion detection systems (e.g., motion sensors, cameras)
- Does the proposed location offer the appropriate environmental controls so that the necessary humidity and temperature are maintained?
- Is there a backup power source?

4.2 Security Management Staff

Because mail server security is tightly intertwined with the organization's general information system security posture, a number of IT and system security staff may be interested in mail server planning, implementation and administration. This section provides a list of those roles and identifies their responsibilities as related to mail server security. These roles may vary with the organization, however, and not all organizations will have the identical roles described here.

4.2.1 Senior IT Management/Chief Information Officer (CIO)

The Senior IT Management/CIO ensures that the organization's security posture is adequate. The Senior IT Management provides direction and advisory services for the protection of information systems for the entire organization. The Senior IT Management/CIO is responsible for the following activities that are associated with mail servers:

- Coordinating the development and maintenance of the organization's information security policies, standards, and procedures
- Coordinating the development and maintenance of the organization's change control and management procedures
- Ensuring the establishment of, and compliance with, consistent IT security policies for departments throughout the organization
- Coordinating with upper management, public affairs, and other relevant personnel to produce a formal policy and process for email usage guidelines (e.g., personal use, monitoring, encryption)

4.2.2 Information Systems Security Program Managers

The Information Systems Security Program Managers oversee the implementation of, and compliance, with the standards, rules, and regulations specified in the organization's security policy. The ISSMs are responsible for the following activities associated with mail servers:

- Developing and implementing standard procedures (security policy)
- Complying with security policies, standards, and requirements
- Ensuring that all critical systems are identified and that contingency planning, disaster recovery plans, and continuity of operations plans exist for these critical systems
- Ensuring that critical systems are identified and scheduled for periodic security testing according to the security policy requirements of each respective system.

4.2.3 Information Systems Security Officers

Information Systems Security Officers (ISSOs) are responsible for overseeing all aspects of information security within a specific organizational entity. They ensure that the organization's information security practices comply with organizational and departmental policies, standards, and procedures. ISSOs are responsible for the following activities associated with mail servers:

- Developing internal security standards and procedures for the mail server(s) and supporting network infrastructure
- Cooperating in the development and implementation of security tools, mechanisms, and mitigation techniques
- Maintaining standard configuration profiles of the mail servers and supporting network infrastructure controlled by the organization, including but not limited to, operating systems, firewalls, routers, and mail server applications
- Maintaining operational integrity of systems by conducting security tests and ensuring that designated IT professionals are conducting scheduled testing on critical systems.

4.2.4 Mail Server and Network Administrators

Mail server administrators are system architects responsible for the overall design, implementation, and maintenance of a mail server. Network administrators are responsible for the overall design, implementation and maintenance of a network. Daily, mail server and network administrators must address the security requirements of the specific system(s) for which they are responsible. Security issues and solutions can originate from either outside (e.g., security patches and fixes from the vendor or computer security incident response teams) or within the organization (e.g., the Security Office). The administrators are responsible for the following activities, associated with mail servers:

- Install and configure systems in compliance with the organizational security policy(ies) and standard system/network configurations
- Maintain systems in a secure manner, though frequent backups, timely application of patches
- Monitoring system integrity, protection levels, and security related events
- Following-up with detected security anomalies associated with their information system resources
- Conducting security tests as required.

4.3 Management Practices

Appropriate management practices are the most critical to operating and maintaining a secure mail server. Security practices entail the identification of an organization's information system assets and the development, documentation, and implementation of policies, standards, procedures and guidelines that ensure confidentiality, integrity, and availability of information system resources.

To ensure the security of a mail server and the support network infrastructure, the following practices should be implemented:

- **Organizational Information System Security Policy**—A security policy should outline who in the organization is responsible for particular areas of information security (e.g., implementation, enforcement, audit, review). The policy should also specify what the basic information system security policies are and their intended internal purpose. The policy must be enforced consistently throughout the organization in order to be effective. Generally, the CIO and upper management are responsible for drafting the organization's security policy.
- **Configuration/Change Control and Management**—Is the process of controlling modification to a system's design, hardware, firmware and hardware which provides sufficient assurance the system is protected against the introduction of an improper modification prior to, during, and after system implementation. . Configuration control leads to consistency with the organization information system security policy. Configuration control is traditionally overseen by a configuration control board that is the final authority on all proposed changes to an information system.
- **Risk Assessment and Management**—Risk assessment is the process of analyzing and interpreting risk. It involves determining the assessment's scope and methodology, collecting and analyzing risk related data, and interpreting the risk analysis results. Collecting and analyzing risk data requires identifying assets, threats, vulnerabilities, safeguards, consequences, and the probability of a successful attack. Risk management is the process of selecting and implementing of controls to reduce risk to a level acceptable to the organization.
- **Standardized Configurations**—Organizations should develop standardized secure configurations for widely used operating systems and applications. This will provide guidance to mail server and network administrators on how to securely configure their

systems and ensure consistency and compliance with the organizational security policy. Because it only takes one insecurely configured host to compromise a network, organizations with a significant number of hosts are especially encouraged to apply this recommendation.

- **Security Awareness and Training**—A security training program is critical to the overall security posture of an organization. Making users and administrators aware of their security responsibilities and teaching the correct practices helps them change their behavior to conform to security best practices. Training also supports individual accountability, which is an important method for improving information system security.
- **Contingency Planning, Continuity of Operations and Disaster Recovery Planning** – Contingency planning, continuity of operations and disaster recovery planning are plans setup in advance to allow an organization or facility to maintain operations in the event of a disruption to their organization.¹⁶
- **Certification and Accreditation** – Certification in the context of information systems security means that a system has been analyzed as to how well it meets all of the security requirements of the organization. Accreditation occurs when the organization's management accepts that the system meets the organization's security requirements.¹⁷

4.4 System Security Plan

The objective of computer security planning is to protect information assets (i.e., information and information resources).¹⁸ Plans that adequately protect information assets require managers and information owners – directly affected by and interested in the information and/or processing capabilities – to be convinced that their information assets are adequately protected from loss, misuse, unauthorized access or modification, unavailability, or undetected activities.

The system security plan provides a basic overview of the security and privacy requirements of the subject system and the organization's plan for meeting those requirements. The system security plan is also perceived as way of documenting the structured process of planning adequate, cost-effective security protection for a system. Consequently, the system security plan should reflect input from various managers with responsibilities concerning the system, including functional end-users or information owners, system operations, and system security manager.

¹⁶ For more information see NIST Special Publication 800-34, *Contingency Planning Guide for Information Technology Systems* (<http://csrc.nist.gov/publications/>)

¹⁷ For more information on certification and accreditation see NIST Special Publication 800-37, *Federal Guidelines for the Security Certification and Accreditation of Information Technology Systems* (<http://csrc.nist.gov/publications/>)

¹⁸ For more information on system security plans, see NIST Special Publication 800-18, *Guide for Developing Security Plans for Information Technology Systems* (<http://csrc.nist.gov/publications/nistpubs/index.html>).

All important information systems should be covered by a system security plan. For Federal agencies the completion of system security plans is a requirement of the Office of Management and Budget (OMB) Circular A-130, "Management of Federal Information Resources," Appendix III, "Security of Federal Automated Information Resources," updated in 1996, and of Public Law 100-235, "Computer Security Act of 1987." Other organizations should strongly consider the completion a system security plan for each of their systems as well [NIST98].

For Federal agencies, OMB Circular A-130, Appendix III, does not distinguish between sensitive and non-sensitive systems. Rather, consistent with the Computer Security Act of 1987, the Circular recognizes that federal automated information systems have varied sensitivity and criticality. All federal systems have some level of sensitivity and require protection as part of good management practice.

The purposes of system security plans are to [NIST98]:

- Provide an overview of the security requirements of the system and describe the controls in place or planned for meeting those requirements; and
- Delineate responsibilities and expected behavior of all individuals who access the system.

The System Owner¹⁹ is generally responsible for ensuring that the security plan is prepared and for implementing the plan and monitoring its effectiveness. Security plans should reflect input from various individuals with responsibilities concerning the system, including functional "end users," Information Owners²⁰, the System Administrator, and the System Security Manager.

In general, an effective system security plan should include the following [NIST98]:

- **System Identification** – The first section of the system security plan provides basic identifying information about the system. It contains general descriptive information regarding who is responsible for the system, the purpose of the system, and the sensitivity level of the system.
- **Management Controls** – This section, describes the management control measures (in place or planned) that are intended to meet the protection requirements of the information system. Management controls focus on the management of the computer

¹⁹ The System Owner is responsible for defining the system's operating parameters, authorized functions, and security requirements. The information owner for information stored within, processed by, or transmitted by a system may or may not be the same as the System Owner. In addition, a single system may utilize information from multiple Information Owners.

²⁰ The Information Owner is responsible for establishing the rules for appropriate use and protection of the subject data/information (rules of behavior). The Information Owner retains that responsibility even when the data/information are shared with other organizations.

security system and the management of risk for a system. The types of control measures shall be consistent with the need for protection of the system.²¹

- **Operational Controls** – This section addresses security methods that focus on mechanisms that primarily are implemented and executed by people (as opposed to systems). These controls are put in place to improve the security of a particular system (or group of systems). They often require technical or specialized expertise – and often rely upon management activities as well as technical controls. This section describes the operational control measures (in place or planned) that are intended to meet the protection requirements of the information system.
- **Technical Controls** – Technical controls focus on security controls that the computer system executes. The controls can provide automated protection from unauthorized access or misuse, facilitate detection of security violations, and support security requirements for applications and data. The implementation of technical controls, however, always requires significant operational considerations and should be consistent with the management of security within the organization. This section describes the technical control measures (in place or planned) that are intended to meet the protection requirements of the major application.

4.5 Human Resources for Securing an Email Server

The greatest challenge and expense in developing and securely maintaining an email server is providing the necessary human resources to adequately perform the required functions. Many organizations fail to fully realize the amount of expense and skills required to field a secure email server. This failure often results in overworked employees and insecure systems. From the initial planning stages, an organization needs to determine the necessary human resource requirements. Appropriate and sufficient human resources are the single most important aspect of email server security. Organizations should also consider the fact that in general technical solutions do NOT substitute for skilled and experienced personnel.

When considering the human resource implications of developing and deploying an email server, organizations should consider the following:

- **Required Personnel** – What types of personnel are going to be required? This would include such positions as system and email server administrators, network administrators, ISSOs, etc.
- **Required Skills** – What are the required skills to adequately plan, develop and maintain the email server in a secure manner? Examples would include, operating system administration, network administration, active content expertise, programming, etc.
- **Available Personnel** – What are the available human resources within the organization? In addition, what are their current skill sets and are they sufficient for

²¹ For more detail on management controls, see NIST Special Publication 800-12, *An Introduction to Computer Security: The NIST Handbook*.

supporting the email server? Often an organization will discover that their existing human resources are not sufficient and will have to consider the following options:

- **Hire Additional Staff**, – If there is not enough staff available or they do not have the requisite skills, it may be necessary to hire additional personnel.
- **Train Current Staff**, – If there are personnel available but they do not have the requisite skills, the organization may choose to train the existing staff in the skills required. While this is an excellent option, the organization should ensure that employees meet the necessary prerequisites for training.

Once the organization has staffed the project and the email server is active it will be necessary to ensure the number and skills of the personnel are still adequate. The threat and vulnerability levels of IT systems including email servers are constantly changing, as is the technology. This means that what is adequate today may not be tomorrow.

4.6 General Information System Security Principles

When addressing mail server security issues it is an excellent idea to keep some general information security principles in mind [Curt01 and Salt75]:

- **Simplicity**—Security mechanisms (and the information systems in general) should be as simple as possible. Complexity is at the root of many security issues.
- **Fail-Safe**—If a failure occurs, the system should fail in a secure manner. That is if a failure occurs, security should still be enforced. It is usually better to lose functionality rather than losing security.
- **Complete Mediation**—Rather than providing direct access to information, mediators that enforce access policy should be employed. Common examples include file system permissions, proxies, firewalls, and mail gateways.
- **Open Design**—System security should not depend on the secrecy of the implementation or its components. “Security through obscurity” is not reliable.
- **Separation of Privilege**—Functions, to the degree possible, should be separate and provide as much granularity as possible. The concept can apply to both systems and operators/users. In the case of systems, such functions such as read, edit, write, and execute should be separate. In the case of system operators and users, roles should be as separate as possible. For example if resources allow, the role of system administrator should be separate from that of the security administrator.
- **Least Privilege**—The ability to perform one function should never imply another. In other words, access should be granted as explicitly as possible.
- **Psychological Acceptability**—Users should understand the necessity of security. This can be provided through training and education. In addition, the security mechanisms in place should present users with sensible options that give them the usability they require on a daily basis. If users find the security mechanisms too cumbersome, they may devise ways to work around or compromise them. The

objective is not to weaken security so it is understandable and acceptable, but to train, educate, and design security mechanisms and policies that are usable and effective.

- **Least Common Mechanism**—When providing a feature to the system, it is best to have a process or service gain some function without granting the same function to other parts of the system. The ability for the mail server process to access a back-end database, for instance, should not also enable other applications on the system to access the back-end database.
- **Defense in Depth**—Organizations should understand that a single security mechanism would generally prove insufficient. Security mechanisms (defenses) need to be layered so that compromise of a single security mechanism is insufficient to compromise a host or network. There is no “silver bullet” for information system security.
- **Work Factor**—Organizations should understand what it would take to break the system or network’s security features. The amount of work necessary for an attacker to break the system or network should exceed the value that the attacker would gain from a successful compromise.
- **Compromise Recording**—Records and logs should be maintained so that if a compromise does occur, evidence of the attack is available to the organization. This information can assist in securing the network and host after the compromise and assist in identifying the methods and exploits used by the attacker. This information can be used to better secure the host or network in the future. In addition, this can assist organizations in identifying and prosecuting attackers.

5. Securing the Operating System

The first step in securing a mail server is securing the underlying operating system. All commonly available mail servers operate on a general-purpose operating system. Many security issues can be avoided if the operating systems underlying the mail servers are configured appropriately. Default hardware and software configurations are typically set by vendors to emphasize features, functions, and ease of use at the expense of security. Because vendors are unaware of each organization's security needs, each mail server administrator must configure new servers to reflect their organization's security requirements and reconfigure them as those requirements change. The practices recommended here are designed to help mail server administrators configure and deploy mail servers that satisfy their organization's security requirements. Mail server administrators managing existing mail servers should confirm that their systems address the issues discussed.

The techniques for hardening different operating systems vary greatly; therefore, this section will include the generic procedures common in securing most operating systems. References for securing specific operating systems are provided in Section 5.3. In addition, many organizations maintain their own guidelines specific to their requirements. Some automated tools also exist for hardening operating system and their use is strongly recommended (see Appendix D).

Four basic steps are necessary to maintain basic operating system security:

- Planning the installation and deployment of the host operating system and other components for the mail server (see Section 4.1)
- Configuring the host operating system to adequately address security
- Patching and updating the host operating system as required
- Testing the host operating system to ensure that the previous three steps adequately address all security issues.

5.1 Securely Installing and Configuring an Operating System

5.1.1 Patch and Upgrade Operating System

Once an operating system is installed, it is critical to apply any patches or upgrades to correct for known vulnerabilities. All operating systems released today have some known vulnerabilities that should be corrected before using the operating system to host a mail server. To adequately detect and correct for these vulnerabilities, mail server administrators should:

- Create and implement a patching process
- Identify vulnerabilities and applicable patches²²
- Mitigate vulnerabilities (until patches are available, tested, and installed)

²² To check for operating system or Web server application vulnerabilities, see the NIST ICAT Metabase at <http://icat.nist.gov>.

- Install permanent fixes (often called patches, hotfixes, service packs, or updates).²³

5.1.2 Remove or Disable Unnecessary Services and Applications

Ideally, a mail server should be on a dedicated, single-purpose host. When configuring the operating system, disable everything except that which is expressly permitted – that is, disable or preferably remove all services and applications and then selectively enable those required by the mail server. If possible, install the minimal operating system configuration that is required for the mail server application. Choose the “minimal installation” option, if available, to minimize the effort required in removing unnecessary services. In addition, many uninstall scripts or programs are far from perfect in completely removing all components of a service; therefore, it is always better to not install unnecessary services. Some common services and applications that should usually be disabled include the following:

- Windows NetBIOS/file and printer sharing (may not be possible in all installations)
- Network File System (NFS)
- Telnet
- Simple Network Management Protocol (SNMP) (if not required)
- FTP
- Network Information System (NIS)
- Language compilers and libraries
- System development tools
- Network management tools and utilities (if not required).

Removing unnecessary services and applications is preferable to simply disabling them through configuration settings, because attacks that attempt to alter settings and activate a disabled service cannot succeed when the functional components are completely removed.

Eliminating or disabling unnecessary services enhances the security of a mail server in several ways [CERT00]:

- Other services cannot be compromised and used to attack the host or impair the mail server services. Each service added to a host increases the risk of compromise for that host because each service is another possible avenue of access for an attacker. Less is more secure in this case.
- The host can be configured to better suit the requirements of the particular service. Different services might require different hardware and software configurations, which could lead to unnecessary vulnerabilities or negatively affect performance.

²³ For more information on vulnerabilities and patching, see NIST Special Publication 800-40, *Procedures for Handling Security Patches* (<http://csrc.nist.gov/publications/nistpubs/index.html>).

- By reducing services, the number of logs and log entries is reduced; therefore, detecting unexpected behavior becomes easier (see Section 9.1).

Organizations should determine the services to be enabled on a mail server. Services in addition to the mail server service that might be installed include directory protocols to access the organization's user directory and remote administration services. These services may be required in certain instances, but they may increase the risks to the server. Whether the risks outweigh the benefits is a decision for each organization to make.

5.1.3 Configuring Operating System User Authentication

For mail servers, the authorized users who can configure the operating system are limited to a small number of designated mail server administrators. The users who can access the mail server, however, may range from unrestricted to restricted subsets of the organization's employees. To enforce policy restrictions, if required, the mail server administrator must configure the operating system to authenticate a prospective user by requiring proof that the user is authorized for such access.

Enabling authentication by the host computer involves configuring parts of the operating system, firmware, and applications, such as the software that implements a network service. In special cases, for high-value/high-risk sites, organizations may also use authentication hardware, such as tokens or one-time password devices. Use of authentication mechanisms where authentication information is reusable (e.g., passwords) and transmitted in the clear over a network is strongly discouraged, because the information can be intercepted and used by an attacker to masquerade as an authorized user.

To ensure the appropriate user authentication is in place, take the following steps [CERT00]:

- **Remove or disable unneeded default accounts and groups.** The default configuration of the operating system often includes guest accounts (with and without passwords), administrator or root level accounts, and accounts associated with local and network services. The names and passwords for those accounts are well known. Remove or disable unnecessary accounts to eliminate their use by intruders, including guest accounts on computers containing sensitive information. If there is no requirement to retain a guest account or group, severely restrict its access and change the default password in accordance with the organizational password policy.

For default accounts that need to be retained, change the names (where possible and particularly for administrator or root level accounts) and passwords to be consistent with the organizational password policy. Default account names and passwords are commonly known in the hacker community.

- **Disable non-interactive accounts.** Disable accounts (and the associated passwords) that need to exist but do not require an interactive login. For Unix systems, disable the login shell or provide a login shell with NULL functionality (e.g., /bin/false).
- **Create the user groups.** Assign users to the appropriate groups. Then assign rights to the groups, as documented in the deployment plan. This approach is preferable to assigning rights to individual users because the latter will become unwieldy with large numbers of users.

- **Create the user accounts.** The deployment plan identifies who will be authorized to use each computer and its services. Create only the necessary accounts. Discourage or prohibit the use of shared accounts.
- **Check the organization's password policy.** Set account passwords appropriately. This policy should address the following areas:
 - **Length** – a minimum length for passwords. Specify at least a minimum length of eight characters.
 - **Complexity** – the mix of characters required. Require passwords to contain both uppercase and lowercase letters, at least one non-alphanumeric character, and is not a “dictionary” word.²⁴
 - **Aging** – how long a password may remain unchanged. Require users to change their passwords periodically. Administrator or root level password should be changed every 30 to 120 days. Users passwords should also be changed periodically, with the period determined by the enforced length and complexity of the password combined with the sensitivity of the information protected. When considering the appropriate aging duration the exposure level of user passwords should also be taken into account.
 - **Reuse** – whether a password may be reused. Some users try to defeat a password-aging requirement by changing the password to one they have used before. If possible, ensure that users cannot change their password by merely appending characters to the beginning or end of their original password (e.g., original password was “mysecret” and is changed to “1mysecret” or “mysecret1”).
 - **Authority** – who is allowed to change or reset passwords and what sort of proof is required before initiating any changes.
- **Configure computers to prevent password guessing.** It is relatively easy for an unauthorized user to try to gain access to a computer by using automated software tools that attempt all passwords. If the operating system provides the capability, configure it to increase the period between logon attempts with each unsuccessful attempt. If that is not possible, the second alternative is to deny login after a limited number (e.g. three) of failed attempts. Typically, the account is “locked out” for a period of time (such as 30 minutes) or until a user with appropriate authority reactivates it.

The choice to deny login is another situation that requires the mail server administrator to make a decision that balances security and convenience. Implementing this recommendation can help prevent some kinds of attacks, but it can also allow a malicious intruder to make failed login attempts to prevent user access, a denial of service (DoS) condition.

Failed network login attempts should not prevent an authorized user or administrator from logging in at the console. Note that all failed login attempts whether via the network or console should be logged. If remote administration is not to be implemented (see Section 9.5), disable the ability for the administrator or root level accounts to log in from the network.

²⁴ This would include any personal names, place names, terms, or words in any language that could be found in dictionary or word list.

- **Install and configure other security mechanisms to strengthen authentication.** If the information on the mail server requires it, consider using other authentication mechanisms such as biometrics, smart cards, client/server certificates or one-time password systems. They can be more expensive and difficult to implement, but they may be justified in some circumstances. When such authentication mechanisms and devices are used, the organization's policy should be changed accordingly.

As mentioned earlier, intruders using network sniffers can easily capture passwords passed across a network in clear text. However, passwords are economical and appropriate if properly protected while in transit. Consider implementing authentication and encryption technologies, such as Secure Socket Layer (SSL)/Transport Layer Security (TLS), Secure Shell (SSH) or Virtual Private Networks (VPNs) (for remote users), to protect passwords during transmission.

5.1.4 Configure Resource Controls Appropriately

Many operating systems provide the capability to specify access privileges individually for files, directories, devices, and other computational resources. By carefully setting access controls and denying personnel unauthorized access, the mail server administrator can reduce intentional and unintentional security breaches. For example, denying read access to files and directories helps to protect confidentiality of information, and denying unnecessary write (modify) access can help maintain the integrity of information. Limiting the execution privilege of most system-related tools to authorized system administrators can prevent users from making configuration changes that could reduce security. It also can restrict the intruder's ability to use those tools to attack the system or other systems on the network.

5.2 Security Testing the Operating System

Periodic security testing of the operating system is a vital way to identify vulnerabilities and to ensure that the existing security precautions are effective. Methods for testing operating systems include vulnerability scanning and penetration testing. Vulnerability scanning usually entails using an automated vulnerability scanner to scan a host or groups of hosts on a network for application, network, and operating system vulnerabilities. Penetration testing is a testing process designed to compromise a network using the tools and methodologies of a "hacker." It is an iterative testing process to identify and exploit the weakest areas of the network to gain access to the remainder of the network. This process eventually results in compromising the overall security of the network. Vulnerability scanning should be conducted periodically, at least weekly to monthly, and penetration testing should be conducted at least annually. Because both of these testing techniques also are applicable to testing the mail server application, they are discussed in detail in Section 9.4.²⁵

5.3 Resources for Operating System Specific Security Procedures

The following Web sites provide detailed information about securing specific operating systems:

- **Unix** – CERT *Unix Security Checklist Version 2.0*
(http://www.cert.org/tech_tips/usc20_full.html)

²⁵ For information on these and other testing techniques, see NIST Special Publication 800-42, *Guideline on Network Security Testing*, (<http://csrc.nist.gov/publications/nistpubs/index.html>).

- **Windows NT** – *NSA Guide to Securing Microsoft Windows NT Networks* (<http://nsa1.www.conxion.com/winnt/guides/wnt-1.pdf>)
- **Windows 2000** – NIST Special Publication 800-43, *Guide to Securing Windows 2000 Professional* (<http://csrc.nist.gov/publications/nistpubs/index.html>)
- **Windows 2000** – *NSA Guide to Securing Microsoft Windows 2000* (including 2000 Server) (<http://nsa1.www.conxion.com/win2k/index.html>).
- **Windows 2000**—*Connecting to the Internet Securely: Windows 2000*, (http://www.ciac.org/ciac/documents/CIAC-2321_Connecting_to_the_Internet_Securely_Windows_2000.pdf)

5.4 Securing the Mail Server Operating System Checklist

Completed	Action
	Plan the configuration and deployment of mail server
<input type="checkbox"/>	Identify functions of the mail server
<input type="checkbox"/>	Identify categories of information that will be stored, processed, and transmitted through the mail server
<input type="checkbox"/>	Identify security requirements of information
<input type="checkbox"/>	Identify a dedicated host to run the mail server
<input type="checkbox"/>	Identify network services that will be provided or supported by the mail server
<input type="checkbox"/>	Identify users and categories of users of the mail server and determine privilege for each category of user
<input type="checkbox"/>	Identify user authentication methods for the mail server
	Choose appropriate operating system for mail server
<input type="checkbox"/>	Minimal exposure to vulnerabilities
<input type="checkbox"/>	Ability to restrict administrative or root level activities to authorized users only
<input type="checkbox"/>	Ability to deny access to information on the server other than that intended to be available
<input type="checkbox"/>	Ability to disable unnecessary network services that may be built into the operating system or server software
<input type="checkbox"/>	Acceptable costs for insurance and liability (some insurers charge more for certain operating systems)
<input type="checkbox"/>	Availability of experienced staff to install, configure, secure, and maintain operating system
	Patch and upgrade operating system
<input type="checkbox"/>	Identify and install all necessary patches and upgrades to the operating system
<input type="checkbox"/>	Identify and install all necessary patches and upgrades to applications and services included with the operating system
	Remove or disable unnecessary services and applications
<input type="checkbox"/>	Disable or remove unnecessary services and applications
	Configure operating system user authentication
<input type="checkbox"/>	Remove or disable unneeded default accounts and groups

Completed	Action
<input type="checkbox"/>	Disable non-interactive accounts
<input type="checkbox"/>	Create the user groups for the particular computer
<input type="checkbox"/>	Create the user accounts for the particular computer
<input type="checkbox"/>	Check the organization's password policy, and set account passwords appropriately (e.g., length, complexity)
<input type="checkbox"/>	Configure computer to deny login after a limited number of failed attempts
<input type="checkbox"/>	Install and configure other security mechanisms to strengthen authentication
	Test the security of the operating system
<input type="checkbox"/>	Test operating system after initial install to determine vulnerabilities
<input type="checkbox"/>	Test operating system periodically (e.g. quarterly) to determine new vulnerabilities

6. Mail Server and Content Security

Email security has three main objectives. One objective is protecting the mail server from compromise. This involves hardening the operating system (see Section 5), mail server application (see Section 6.1), and network (see Section 7) to prevent malicious entities from directly attacking the mail server. Another objective is protection of the email client (see Section 8). The third objective is the security of the email content that traverses the server (see Section 6.2). This process often includes content filtering, virus scanning, and prevention of unsolicited mail.

6.1 Hardening the Mail Server Application

6.1.1 Securely Installing the Mail Server

In many respects, the secure install and configuration of the mail server application will mirror the operating system process discussed in the Section 5. The overarching principle, as before, is to install the minimal amount of mail server services required and eliminate any known vulnerabilities through patches or upgrades. Any unnecessary applications, services, or scripts that are installed should be removed immediately once the installation process completes. During the installation of the mail server, the following steps should be performed:

- Install the server software on a dedicated host
- Install minimal internet services required
- Apply any patches or upgrades to correct for known vulnerabilities
- Create a dedicated physical disk or logical partition (separate from operating system and server application) for mail boxes
- Remove or disable all services installed by the mail server application but not required (e.g., Web based mail, FTP, remote administration, etc)
- Remove all vendor documentation from server
- Remove any example or test files from server
- Apply appropriate security template or hardening script to server (see Appendix E)
- Reconfigure SMTP, POP, and IMAP service banner (and others as required) NOT to report mail server and operating system type and version (this may not be possible with all mail servers).
- Disable dangerous or unnecessary mail commands (e.g., VRFY and EXPN)

6.1.2 Securely Configuring Operating System and Mail Server Access Controls

Most mail server host operating systems provide the capability to specify access privileges individually for files, devices, and other computational resources on that host. Any information that the mail server can access using these controls can potentially be distributed to all users accessing the mail server. The mail server software is likely to provide additional file, device,

and resource access controls specific to its operation. It is important to set identical permissions for both the operation system and mail server application; otherwise, too much or too little access may be granted to users. Mail server administrators should consider how best to configure access controls to protect information stored on their public mail server from two perspectives:

- Limit the access of the mail server application to a subset of computational resources
- Limit the access of users through additional access controls enforced by the mail server, where more detailed levels of access control are required.

The proper setting of access controls can help prevent the disclosure of sensitive or restricted information that is not intended for public dissemination. In addition, access controls can be used to limit resource use in the event of a DoS attack against the mail server.

Typical files to which access should be controlled are as follows:

- Application software and configuration files
- Files directly related to security mechanisms:
 - Password hash files and other files used in authentication
 - Files containing authorization information used in controlling access
 - Cryptographic key material used in confidentiality, integrity, and non-repudiation services.
- Server log and system audit files
- System software and configuration files.

Ensure that the mail server application executes only under a unique individual user and group identity with very restrictive access controls. Thus, new user and group identities to be used exclusively by the mail server software need to be established. Make this new user and new group independent and unique from all other users and groups. This is a prerequisite for implementing the access controls described in the following steps. Although the server may initially have to run as root (Unix) or system/administrator (Windows NT/2000/XP) to bind to the necessary TCP ports, do not allow the server to continue to run in at this level of access.

In addition, use the mail server operating system to limit files accessed by the mail service processes. These processes should have read-only access to those files necessary to perform the service and should have no access to other files, such as server log files. Use mail server host operating system access controls to enforce the following:

- Temporary files created by mail server application are restricted to a specified and appropriately protected subdirectory (if possible).
- Access to any temporary files created by mail server application is limited to the mail server process(es) that created these files (if possible).

It is also necessary to ensure that the mail server cannot save files outside the specified file structure dedicated to the mail server. This may be a configuration choice in the server software or it may be a choice in how the server process is controlled by the operating system. Ensure that

such directories and files (outside the specified directory tree) cannot be accessed, even if users know the locations of those files.

On Linux and Unix hosts, consider using a “chroot jail” for the mail server application. Using chroot changes the mail servers “view” of the host file system such that the apparent root directory is not the real file system root directory but rather one its subparts. Thus if the mail server is successfully compromised, the attacker only gains access to the limited subpart of the file system accessible via chroot. This is a very powerful security measure.

To mitigate the effects of certain types of DoS attacks, configure the mail server to limit the amount of operating system resources it can consume. Some examples include:

- Install users’ mail boxes on a different hard drive or logical partition than the operating system and mail server application.
- Limit the size of attachments that are allowed.
- Ensure log files are stored in a location that is sized appropriately.

These actions will protect to some degree against attacks that attempt to fill the file system on the mail server host operating system with extraneous and incorrect information that may cause the system to crash. This will also protect against attacks that attempt to fill primary random access memory with unnecessary processes to slow down or crash the system thus limiting mail server availability. Logging information generated by the mail server host operating system may help in recognizing such attacks (see Section 9.1).

6.2 Protecting Email from Malicious Code

Email has increasingly been used as a means for sending binary files in the form of attachments. Initially, this did not pose much of a security risk because attachments were mostly small word processing documents or photos. As more organizations began using email for day-to-day collaboration, the size and types of email attachments increased. Today, many email messages are sent with attachments such as program executables, pictures, music, and sounds. The main issue an organization needs to address is what types of attachments to allow, if any.

Deciding whether to allow attachments can be a difficult decision. Not allowing attachments would definitely simplify a system and make it more secure; however, it would likely reduce its usefulness to an organization and users may just employ encoding tricks to work around the restriction in order to “get the job done”. Ultimately, allowing attachments is based on organizational needs, and it is likely that most organizations will choose to allow at least some email attachments. Based on the assumption that email attachments are desired, the mail server administrator should determine which types of attachments to allow. The simplest approach is to allow all types of attachments. If this is the case, then some sort of virus scanner should be installed in the mail transit path to filter out known malicious code, and perhaps even some behavior blocking utility at the client to prevent any unwanted operations by executable attachments from occurring. A better approach is to filter potentially dangerous attachment types (e.g., .vbs, .ws, .wsc file extensions) at the mail server or mail gateway, while conducting virus scans on allowed file types.

Viruses or worms can be transmitted in emails in the form of email viruses, or attachment viruses. If a mail server does not have anti-virus software installed, or the software is ineffective, this

potential security threat increases for the end user. Some of the more popular mail clients are highly susceptible to infection and transmission of email borne viruses. These types of viruses are typically a result of the mail client supporting active content, such as HTML messages. The choice of allowing or blocking active content must be made on an organizational basis.²⁶

Several types of content could be considered active content. Typically, the active content comes in the form of a client side scripting language, or control object. The most popular types of active content are ActiveX, Java, JavaScript, and Visual Basic Script. Active content viruses and malicious code can affect an MUA. Consequently, administrators should configure active content features with caution (see Section 8).

6.2.1 Virus Scanning

Protection from active content is only the first step for protecting end users. The next step is protection from attachment borne viruses. To protect against viruses and other malicious code, it will be necessary to implement scanning at one or more points within the email delivery process. Virus scanning can be implemented on the firewall as the mail data enters the organization's network, on the mail server or mail relay and/or on the end user's host. Each option has its own strengths and weaknesses. If resources permit, using more than one of these options can provide greater security while minimizing some of the weaknesses.

Scanning for viruses at the firewall (application proxy) or mail relay is a popular option (see Figure 6.1). In this instance, the firewall or mail relay intercepts messages before they reach the organization's mail server. The firewall or mail relay scans each message and if no viruses are found, forwards the message on to the organization's mail server for delivery. The firewall or mail relay listens on the TCP port 25 for SMTP connections, receives the message, scans the message, then forwards the message on to the mail server, which is configured to listen on an unprivileged, unused port, rather than the usual port 25. A disadvantage to this approach is that constant scanning of the SMTP stream can reduce firewall or mail relay performance. Whether this performance hit is significant depends on mail load and quality of service requirements. One remedy to improve performance is to offload virus scanning to a dedicated server.

²⁶ For more information on active content, see NIST Special Publication 800-28, *Guidelines on Active Content and Mobile Code*, (<http://csrc.nist.gov/publications/nistpubs/>).

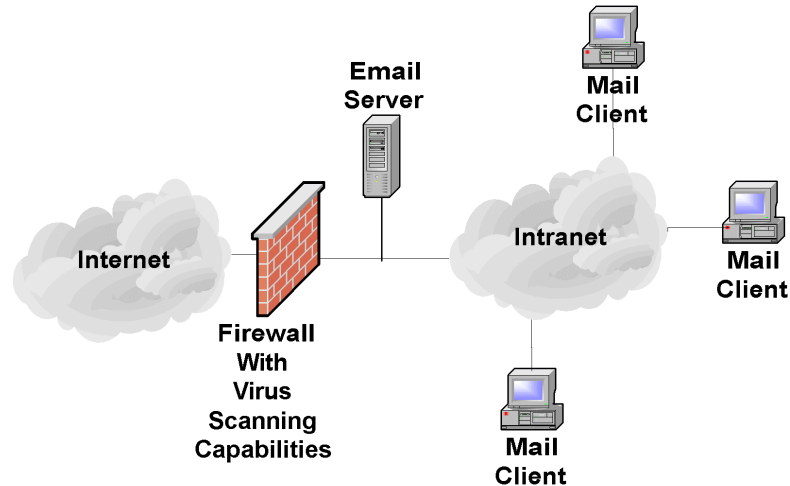


Figure 6.1: Virus Scanning Implemented on Firewall

The benefits of scanning mail at the firewall or mail relay are as follows:

- Mail can be scanned in both directions (inbound and outbound from the organization's network)
- Viruses can be stopped before entering the network
- Scanning inbound mail can be implemented with minor changes to the existing mail server configuration
- Scanning can be centrally managed to ensure compliance with organization's security policy and regular application of updated virus and malicious code signatures
- Firewalls support multiple protocols so the scanner application can also be used to scan other protocols as well (e.g., Hypertext Transfer Protocol [HTTP], FTP).

Scanning for viruses at the firewall or mail relay has a number of weaknesses:

- Can require significant modification of the existing mail server configuration when scanning mail in the outbound direction
- Cannot scan encrypted emails
- Offers no protection to internal users once a virus is on the organization's internal network, unless the network is configured so that SMTP traffic get routed through a dedicated scanner before reaching the mail server
- May require powerful (expensive) server(s) to handle the load of a large organization.

The second option for placement of a mail virus scanner is on the mail server itself (see Figure 6.2). This option is useful for protecting against mail viruses sent by internal users to other internal users because those messages would not normally be scanned by the firewall. This can offer much better protection against an internal outbreak of a mail virus.

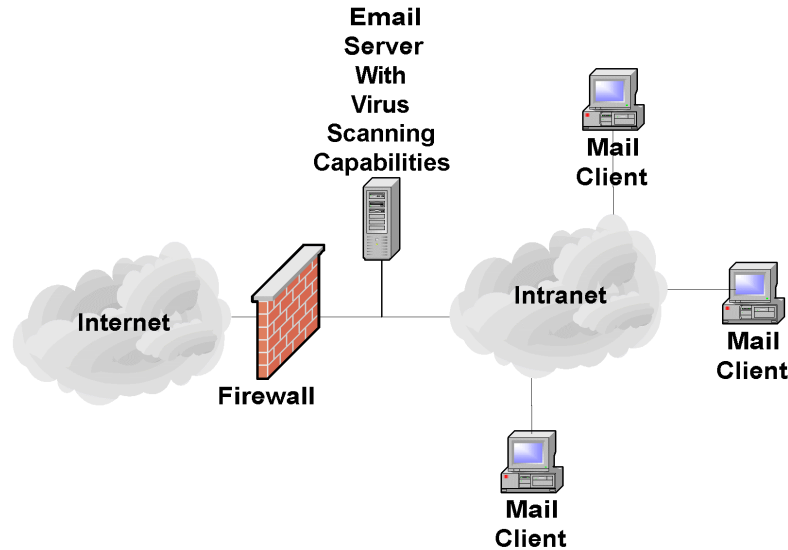


Figure 6.2: Virus Scanning Implemented on Mail Server

The major disadvantage of implementing virus scanning on the mail server is the negative effect on performance of the mail server caused by the requirement to scan all messages. Also disadvantageous, is the fact that virus scanning on the mail server often requires significant modifications to the existing mail server configuration. However, this option provides a number of advantages:

- Mail can be scanned in both directions (inbound and outbound)
- Can be centrally managed to ensure compliance with organization's security policy and that updates are applied regularly
- Offers protection to internal users once a virus is on the organization's internal network.

Scanning for viruses at the mail server has a number of weaknesses:

- Scanning may require significant modification of the existing mail server configuration (less true for most newer mail servers)
- Cannot scan encrypted emails
- May require a more powerful (expensive) server(s) to handle the load of a large organization.

Mail servers such as Microsoft Exchange and newer versions of sendmail support the integration of virus scanning at the mail server. Beginning with Exchange 5.5, Service Pack 3, and Microsoft Exchange 2000, Microsoft created an anti-virus application programming interface (AVAPI) that was designed to support the operation of virus scanning plug-ins. These virus scanning plug-ins are designed to extend the capabilities of Microsoft Exchange to provide functions like: dynamic attachment scanning, mailbox scanning, virus detection and potential removal, and clustering support for high demand networks. Many virus scanning plug-ins for Microsoft Exchange include an ability to block attachment types based on exact file names or file extensions. For

example, to limit the possibility of infection from macro viruses, an organization may block all common Microsoft Office file types such as, .doc, .dot, and .xls.

Sendmail versions 8.10 and later provide a Content Management API that offers an ability to integrate virus scanning and content filtering software within the MTA. As with desktop scanning software, mail server administrators should ensure that virus definition files are current.

When considering firewall or mail server based virus scanners, look for the following qualities:

- Detects and cleans all known viruses and other types of malicious code
- Provides heuristic scanning (provides some protection from new and unknown viruses)
- Provides content filtering (see Section 6.2.2)
- Incorporates mechanisms to help prevent email from circumventing the system
- Provides the inability for email to circumvent the system
- Provides ease of management
- Provides automated downloading and installation of updates
- Provides frequent updates (critical)
- Can identify and apply rules to different types of content
- Provides a robust and configurable alert mechanism
- Provides detailed logging capabilities (see Section 9.1).

Virus scanners can also be located on client hosts (see Figure 6.3). This type of virus scanner is installed on user workstations. Emails are scanned as they are opened by the user. The primary advantage of this type of configuration is that scanning is distributed across many hosts and therefore has minimal effect on performance of each individual system.

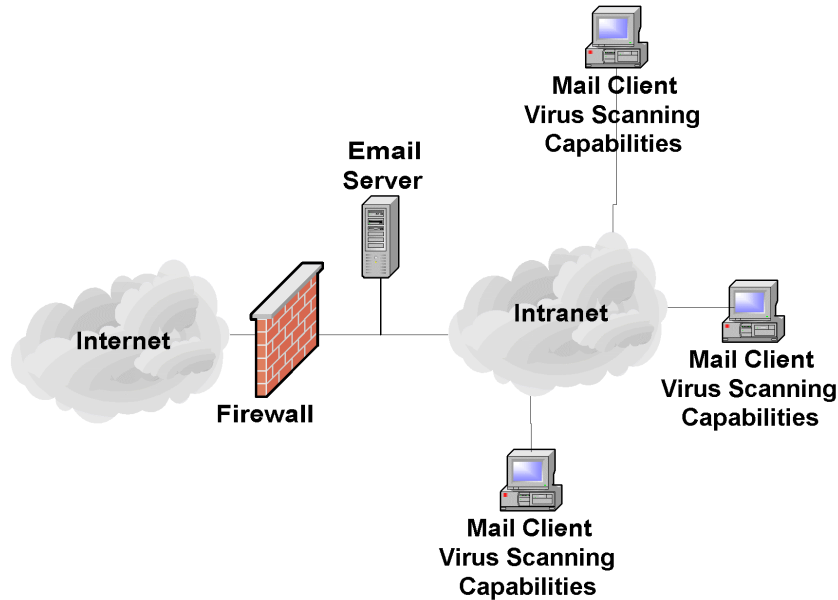


Figure 6.3: Virus Scanning Implemented on User Workstations

The greatest challenge of implementing virus scanning on user workstations is the difficult in managing the distributed virus scanners. The virus scanners can be much more difficult to centrally manage and update on a regular basis. However, enterprise-level solutions for virus scanning provide a means of central administration of virus scanners on individual hosts. Another weakness is that to the degree that users have control of the virus scanner; end users may disable some or all of its functionality (whether accidentally or intentionally).

The benefits of client side virus scanning are as follows:

- Does not require any modification to mail server
- Can scan encrypted emails when they are decrypted by user
- Distributes virus scanning and thus minimizes the impact of scanning on any one host
- Offers protection to internal users, even when a virus is received from an internal user.

The disadvantages of client side virus scanning are as follows:

- Difficult to centrally manage
- Users may be slow to update virus scanners, resulting in the organization being more susceptible to an outbreak
- Users may intentionally or accidentally disable or weaken the protection offered by the virus scanner
- Typically scans only incoming messages
- Virus is not contained at the firewall or even a central mail server.

It is useful to implement at least two levels of virus scanning. The safest option is to implement a centralized virus scanner (either at the firewall, mail relay, or mail server) and include some level of virus protection on the end users' hosts. This will provide layered protection and combines the strengths of these options.

Perhaps the most important step is to educate users about the danger of email borne viruses, malicious code, and the following actions:

- Never open attachments from unknown senders.
- Never open attachments with suspicious or known suspect file extensions (e.g., attachment.txt.vbs, attachment.exe).
- Be suspicious of emails from known senders in which the subject line or content appears to be inappropriate for the existing relationship (e.g., an email with the subject "I love you" from a professional colleague) or generic subjects (e.g., "Look at this, it's interesting").
- Scan all attachments with a virus scanner before opening, preferably by configuring the virus scanner to automatically perform this task.
- Update the virus signature database of the virus scanner on a daily to weekly basis or when there is an outbreak of a virus.
- Warn users about virus outbreaks and how to identify emails that might contain a virus.

Another consideration that should be addressed regarding attachments is the size of the attachment itself. Because of the varying storage and processing requirements for larger messages, mail servers should restrict the maximum accepted message size. When a binary file, such as a picture, is attached to an email message, it is not sent in its native format; rather, it is encoded. As mentioned in Section 2.2, binary attachments are represented as blocks of Base64 encoded text. This type of encoding imposes a 33-percent increase in the message size. This would make a message composed of basic headers and a 1 megabyte attachment become a message that is roughly 1.33 megabytes.

Implementing a size restriction benefits the mail server in the several ways:

- Decreased mail queue latency
- Decreased storage requirements
- Decreased server processor requirements.

The combination of these three benefits decreases the likelihood of the mail server being put into a DoS state resulting from a deluge of over-sized messages.

6.2.2 Content Filtering

For years, information systems that processed classified information employed the concept of content filtering. Only recently has this concept reached the Internet. In essence, content filtering works in a similar manner to virus scanning at the firewall or mail server except that it takes this concept in a different direction. It looks at the content of emails for characteristics other than

malicious code that might be of interest to the organization. When implementing file-type restrictions and virus scanning, only a certain level of security is provided. The contents of an email message or its attachments could prove much more damaging to an organization than a virus or rogue executable. For this case, some sort of content filtering mechanism should be employed. Often content filtering and virus scanning are conducted on the same server located in the network demilitarized zone (see Section 7.1.2).

In general, rules are defined to forward, quarantine, park, clean, block or delete any data passing through the server depending the results of the scan. Typical items that would be caught by the filter and possible action taken on them could be as follows:

- Email that contains suspicious active content (e.g., ActiveX, JavaScript) is stripped of the active code and forwarded to recipient.
- Spam email may be deleted (see Section 6.3).
- Extra large files might be parked for delivery at off peak hours.

Another key feature of content filtering packages is to allow the scanning of outbound data. A lexical analysis can be performed that scans email messages for words and phrases that might be viewed as inappropriate for use in organizational email. The lexical analysis can also save possible litigation against an organization by preventing inappropriate content, including hoaxes and “spam” (see Section 6.3), from leaving the organization. In addition, a lexical analysis might include searches for key words and phrases indicating that sensitive data is leaving the enterprise.

Before implementing any filtering solution, it is imperative to determine how the existing network and applications actually work. This will entail running network analyzers (sniffers); analyzing router, firewall, and server log files; and interviewing all appropriate system and network administrators. It is also imperative to analyze the existing organization information system security policy (or draft one if one does not exist). Clearly defined security policies are critical to translating the organization’s security goals into filter rules. Great care must be taken in crafting the rules because an incorrectly configured filter may fail to filter inappropriate content or may accidentally filter appropriate content. These steps will make it easier to choose the appropriate filtering software and determine the types of rules that need to be configured.

Many different content filtering applications are available for most messaging systems (see Appendix D). For maximum the effectiveness content filtering, should be employed for all incoming and outgoing messages. Many newer products incorporate content filtering, virus scanning, and file-type restriction. Incorporation of these features into one product can reduce the administration of security controls.

6.2.3 Content Filtering Issues

Although email content filtering is critical to most organizations security posture, a number of legal implications that should be addressed before deployment’. Content filtering needs to be backed up by a clearly defined, written security policy. The email policy should include an explicit statement that email will be monitored for compliance, any administrative or disciplinary actions that could result if the policy is violated, and a requirement for employees to acknowledge reading and understanding the policy. Although the policy should outline the organization’s thinking, expectations, and restrictions regarding security, due regard should also be given to employee and individual rights. For instance, under some circumstances employees may have a

right to privacy when it comes to their own correspondence; however, when representing their organization, the organization may be held legally responsible for what they say or do. Without an established policy, such issues often lead to misunderstanding and problems that can be difficult to resolve.

Similarly, in some situations, email messages may be deemed to carry the same legal weight as written documents, especially when digitally signed. This can mean having to store messages to comply with record-keeping requirements, including employees' personal messages. As such, all employees should be made aware of the security policy. To the extent possible, the security policy should be broadly distributed to employees [PCMAG01]. Moreover, it may be advisable to require employees to acknowledge the policy as part of their contract of employment or as a condition of working on contract. Many email filter applications can add a legal disclaimer to all incoming and outgoing messages, ensuring recipients understand the legal weight (or lack thereof) of the emails received from or through the organization.

Appropriate legal, privacy, personnel, and human resources authorities should be consulted when forming the policy. Inevitably, this means having the policy reviewed by experts to ensure that it is legally correct and does not infringe the rights of employees. Additionally, it is important to investigate all areas of an organization to determine how workers go about their work and what level of security is most appropriate. Completely restricting access to Internet resources might solve most security issues at a stroke, but this is usually an unacceptable tradeoff. This is where email filtering tools can help, enabling security policies to be more easily converted from theory into practice.

6.3 Unsolicited Bulk Email

Regardless of the communication medium, there are always entities that attempt to exploit any means of communication to publicize their ideas or products. Email is no exception. The most common terms for these messages are unsolicited commercial email (UCE) or spam. Most email users receive at least a few of these solicitations on a regular basis. Because email is largely unregulated, system administrators must police email traffic that traverses the servers they operate. An added benefit of implementing server-based UCE control is that it will reduce mailbox sizes that, in turn, reduce server storage requirements.²⁷

To control UCE messages, administrators must address two concerns: (1) ensure that UCE cannot be sent from mail servers they control (see Section 6.4); and (2) implement inbound message control, which is the topic of this section.

Because the Internet has no centralized policing authority, mail server administrators have created lists of mail servers that are often used to send unsolicited email messages. These lists are often referred to as open relay blacklists (ORBs). Many popular mail server applications can reject messages based on multiple ORBs. These lists are updated regularly; therefore, a properly configured server can drastically reduce spam messages to its users. Figure 6.4 shows excerpts from a sendmail configuration file in which ORB controls are implemented.

²⁷ For more information on UCE see RFC 2505 "Anti-Spam Recommendations for SMTP MTAs" (<http://ietf.org/rfc/rfc2505.txt>).

```
...
FEATURE(`dnsbl',`relays.mail-abuse.org')
FEATURE(`dnsbl',`inputs.orbs.org')
...
```

Figure 6.4: Sendmail Blacklist Configuration from sendmail.cf

Additionally, most mail servers can be configured to reject messages from an explicitly defined set of domains. Figure 6.5 shows excerpts from a sendmail access configuration file that implements partial UCE control by explicitly denying or allowing mail relay from a set of domains. For more information about UCE control using sendmail or other mail servers, refer to the vendor-supplied documentation.

local.com	RELAY #allow relay from local.com
spammers.net	REJECT #reject all mail from spammers.net
[127.0.0.1]	OK #relay mail from this specific host
10.	REJECT #reject all mail from this IP domain

Figure 6.5: Sendmail Access Configuration

ORBs spam control is not foolproof. Open relays are connected and disconnected regularly. Mail server administrators who wish to participate in the policing effort may submit UCE complaints to Web sites provided in Appendix G: Spam Resources.

6.4 Authenticated Mail Relay

As mentioned previously, configuring mail relay authentication decreases the likelihood of bulk email being sent through a mail server. An added benefit of implementing authenticated relay is increased security and usability.

Two methods are available for controlling mail relay. The first is to control the subnet or domain from which messages are being sent. This method is effective if the perimeter of the messaging system resides within known address ranges. However, if remote users have systems with different address ranges, this method is not useful. To accommodate remote users, a more robust configuration is needed.

The second method is to require users to authenticate themselves before sending any messages. This is commonly referred to as authenticated relay, or SMTP AUTH, which is the SMTP extension that supports user authentication. Unfortunately, the default configuration of most mail servers does not implement authenticated relay; therefore, mail server administrators must configure the server appropriately. Requiring authenticated relay is one of the least used but most powerful security features of mail servers. (Refer to vendor documentation for configuring SMTP AUTH.)

6.5 Secure Access

In Section 2, different mail transport and mailbox access protocols were discussed. Like many Internet protocols, most of these protocols did not initially incorporate any form of encryption or cryptographic authentication. These deficits posed three problems for mail users. First, for users sending messages, the contents could be intercepted and read at any host on the path between the

sender and recipient or even forged or modified. An adept paradigm from “regular” mail would be a postcard. Any person who handles the postcard can read the message on the back. Second, the recipients cannot verify that messages were actually originated by the sender or unmodified by others when in transit. These problems were addressed in Section 3, which discussed ways to protect messages. Third, rather than supplying non-reusable authentication information, a user accessing a mailbox would send a password over the network in the clear, which could be easily observed and reused by an attacker. Unfortunately, in most default configurations, mail clients are set up to send the user’s password in the clear, allowing it to be intercepted by other computers on the local network segment of the client or any host responsible for forwarding the password to the mail server.

This last problem can be resolved by applying the same method normally used to secure World Wide Web (WWW) traffic – the Transport Layer Security (TLS) protocol. TLS is similar to the Secure Socket Layer (SSL) protocol, upon which it is based, and can be used in combination with POP, IMAP, and SMTP to encrypt communication between mail clients and servers. RFC 2595 defines how to use TLS to combat communications eavesdropping, to implement secure mailbox access and to further strengthen SMTP MTAs that incorporate SMTP AUTH. Figure 5.3 shows a sample configuration that enables TLS support for newer versions of sendmail.

```
define(`CERT_DIR', `MAIL_SETTINGS_DIR`certs')dnl
define(`confCACERT_PATH', `CERT_DIR')dnl
define(`confCACERT', `CERT_DIR/CACert.pem')dnl
define(`confSERVER_CERT', `CERT_DIR/MYcert.pem')dnl
define(`confSERVER_KEY', `CERT_DIR/MYkey.pem')dnl
define(`confCLIENT_CERT', `CERT_DIR/cert.pem')dnl
define(`confCLIENT_KEY', `CERT_DIR/MYkey.pem')dnl
```

Figure 6.6: Sendmail TLS Configuration Example from sendmail.mc

6.6 Enabling Web Access

Increasingly, organizations are providing Web-browser based access to their messaging systems. Enabling this type of access could introduce security issues at both the client (see Section 8.4) and server. Although secure Web site security is outside the scope of this document, there are several key concepts that should be followed. Avoid placing the Web server on the same machine as the mail server. The authentication mechanism of the Web front-end should employ encryption. The Web server should use SSL/TLS to encrypt all communications with clients. For some organizations, the processing requirements of dedicating a Web server to SSL/TLS communication may not be possible to accommodate. In cases such as this, the initial authentication should be encrypted. As with any public server, the Web server should be hardened before connecting it to the network²⁸.

There are also some client security issues that need to be considered before an organization approves the deployment of Web access to email. These issues are discussed in Section 8.4.

²⁸ For more information on securing Web servers, please refer to NIST Special Publication 800-44, *Securing Public Web Servers*, (<http://csrc.nist.gov/publications/nistpubs/index.html>).

6.7 Mail Server and Content Security Checklist

Completed	Action
	Hardening the mail server application
<input type="checkbox"/>	Install the server software on a dedicated host
<input type="checkbox"/>	Install minimal Internet services required
<input type="checkbox"/>	Apply any patches or upgrades to correct for known vulnerabilities
<input type="checkbox"/>	Remove or disable all services installed by the mail server application but not required (e.g., Web based mail, FTP, remote administration)
<input type="checkbox"/>	Remove all vendor documentation from server
<input type="checkbox"/>	Apply appropriate security template or hardening script to server
<input type="checkbox"/>	Reconfigure SMTP, POP and IMAP service banner (and others as required) NOT to report mail server and operating system type and version.
<input type="checkbox"/>	Disable dangerous or unnecessary mail commands (e.g., VRFY and EXPN)
	Configuring operating system and mail server access controls
<input type="checkbox"/>	Limit the access of the mail server application to a subset of computational resources
<input type="checkbox"/>	Limit the access of users through additional access controls enforced by the mail server, where more detailed levels of access control are required
<input type="checkbox"/>	Configure mail server application to execute only under a unique individual user and group identity with restrictive access controls
<input type="checkbox"/>	Ensure the mail server is not running as root or system/administrator
<input type="checkbox"/>	Configure host operating system so that mail server can write log files but not read them
<input type="checkbox"/>	Configure host operating system so that temporary files created by mail server application are restricted to a specified and appropriately protected subdirectory
<input type="checkbox"/>	Configure host operating system so that access to any temporary files created by mail server application is limited to the mail server process(es) that created these files
<input type="checkbox"/>	Ensure that mail server cannot save files outside of the specified files structure dedicated to the mail server
<input type="checkbox"/>	Configure mail server to run in chroot jail on Linux and Unix hosts
<input type="checkbox"/>	Install users mail boxes on a different hard drive or logical partition than the operating system and mail server application
<input type="checkbox"/>	Limit the size of attachments that are allowed
<input type="checkbox"/>	Ensure log files are stored in a location that is sized appropriately
	Coping with harmful attachment and content
<input type="checkbox"/>	Implement a centralized virus scanner (either on mail gateway, firewall and/or mail server)
<input type="checkbox"/>	Install virus scanners on all client hosts
<input type="checkbox"/>	Update all virus databases on all scanners on a daily or a weekly basis or when there is a particular virulent outbreak
<input type="checkbox"/>	Educate users on the dangers of viruses and how to minimize those dangers
<input type="checkbox"/>	Notify users when an outbreak occurs
<input type="checkbox"/>	Configure content filtering to block suspicious messages

Completed	Action
<input type="checkbox"/>	Configure content filtering to block UCE messages
<input type="checkbox"/>	Configure lexical analysis if required
<input type="checkbox"/>	Create a content filtering policy
<input type="checkbox"/>	Add legal disclaimer to emails, if required
<input type="checkbox"/>	Configure mail server to block mail from open relay blacklists
<input type="checkbox"/>	Configure mail server to block mail from specific domains, if required
<input type="checkbox"/>	Configure authenticated mail relay on server
<input type="checkbox"/>	Configure mail server to use encrypted authentication
<input type="checkbox"/>	Configure mail server to support Web access only via SSL/TLS and only if such access is deemed necessary

7. Implementing a Secure Network for a Mail Server

The network infrastructure that supports the mail server plays a critical role in the security of the mail server. In most configurations, the network infrastructure will be the first line of defense between the Internet and a mail server. Although considerations of network infrastructure are influenced by many factors other than security (e.g., cost, performance, reliability), this section will primarily address security issues.

Network design alone, however, cannot protect a mail server. The frequency, sophistication, and even variety of attacks perpetrated today, lend support to the idea that mail security must be implemented through layered and diverse protection mechanisms (layered protection). This section discusses those network components that can support and protect mail servers to further enhance their overall security.

7.1 Network Location

An organization has many choices when selecting a networking location and security may not be the principal factor in deciding between those options. Network location is the first and in many respects most critical networking decision that affects mail server security. Network location is important for several reasons. Network location determines what network infrastructure can be used to protect the mail server. For example, if the mail server is located before the organization's firewall, then the firewall cannot be used to control traffic to and from the mail server. Network location also determines what other portions of the network are vulnerable if the mail server is compromised. For example, if the mail server is located on the internal production network, then the internal network is subject to attack from the compromised mail server.

7.1.1 Unadvisable Network Locations

Some organizations choose to locate their public mail servers on their internal production networks, that is, they locate their mail server on the same network as their internal users and servers. This location is not recommended because it exposes the internal network to unnecessary risk of compromise. The principal weakness of this configuration is that mail servers are often targets of attackers. If they manage to compromise the mail server, they will be on the internal network and can more easily compromise internal hosts.

Another network location that is not generally recommended is placing the mail server before an organization's firewall or router that provides IP filtering. In this type of the configuration, the network can provide little, if any, protection to the mail server. All security has to be provided by the mail server itself, which provides a single point of failure. To be even somewhat secure in this location, the mail server operating system and application has to be well hardened (all unnecessary and insecure services disabled) and with all necessary security patches applied. To maintain the "security" of the setup, the mail server administrator must stay up-to-date on all vulnerabilities and related patches. Another limitation of this location is that it is difficult in this type of configuration to provide any sort of secure remote administration capability.

7.1.2 Demilitarized Zone

A DMZ can be defined as a host or network segment inserted as a ‘neutral zone’ between an organization’s private network and the Internet. It prevents outside users of the mail server from gaining direct access to an organization’s internal network (intranet). A DMZ mitigates the risks of locating a mail server on an internal network or exposing it directly to the Internet. It is a compromise solution that offers the most benefits with the least amount of risk for most organizations. The DMZ allows access to the resources located within it to both internal and external users. There are a wide variety of DMZ configurations, each with its own strengths and weaknesses.

Creating a DMZ places a firewall between an organization’s border router and its internal network (in some configurations the border router itself may act as a basic firewall). The new segment of network that is created by this action is where the mail server or mail gateway is placed along with other network infrastructure components and servers that need to be externally accessible. The new segment of network created by this action forms the DMZ where a mail server is placed. Figure 7.1 illustrates an example of this simple DMZ using a router with access controls lists (ACLs) to restrict certain types of network traffic to and from the DMZ.

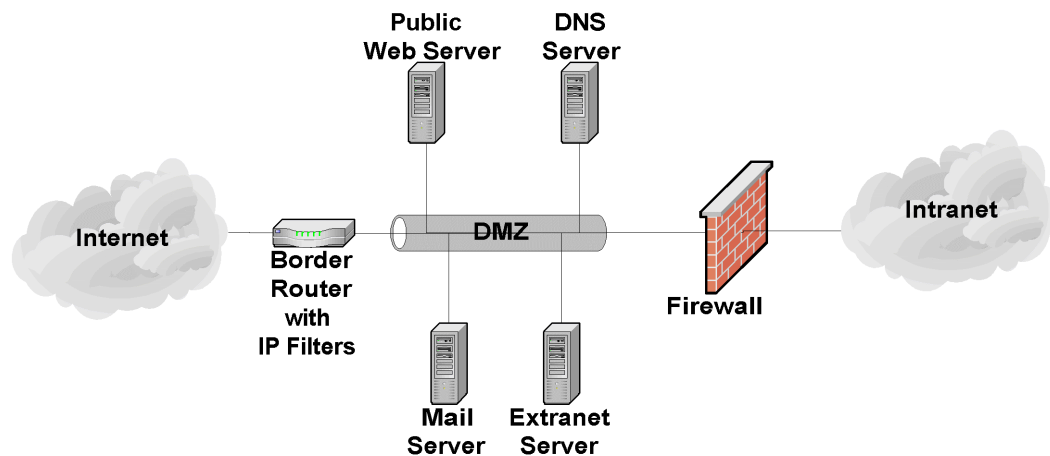


Figure 7.1: Simple Single Firewall DMZ

A single firewall DMZ is a low cost approach since the organization only needs to purchase a single firewall and can use their existing border router to provide protection to the DMZ. It is usually appropriate only for small organizations that face a minimal threat. The basic weakness in the approach is that while the router is able to protect against most network attacks it is not “aware” of the mail server application layer protocols (e.g., SMTP, POP, IMAP) and thus cannot protect against application layer attacks aimed at the mail server. In addition, a router cannot provide any virus scanning of incoming email. A superior approach is to add a second firewall between the Internet and the DMZ as shown in Figure 7.2.

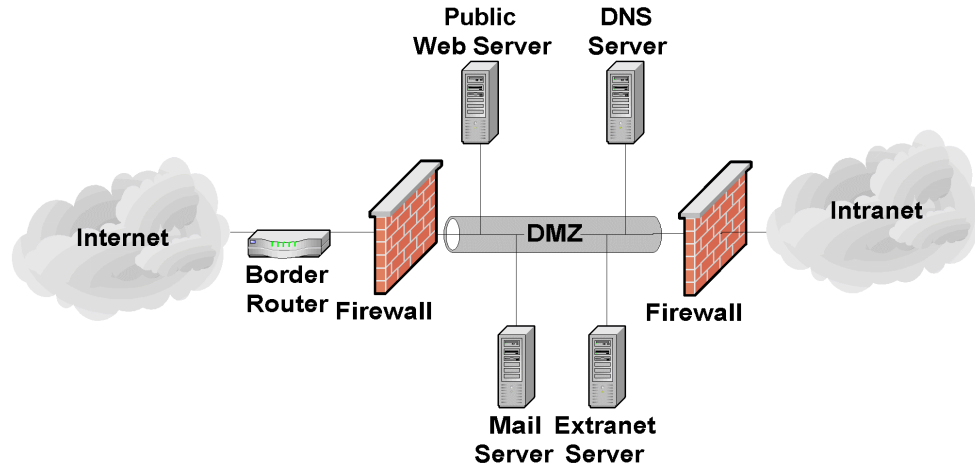


Figure 7.2: Two Firewall DMZ

A two firewall DMZ configuration offers superior protection over a router-firewall DMZ since the dedicated firewalls can have a more complex and powerful security rule set. In addition, the dedicated firewall is often able to analyze incoming and outgoing mail traffic, it can detect and protect against application layer attacks aimed at the mail server. Depending on the configuration of the firewalls and the level of traffic the DMZ receives; this type of DMZ may result in some performance degradation.

For organizations who desire the security of the two firewall DMZ but who do not have the resources to purchase two firewalls, there exists another option called the “service leg” DMZ. In this configuration, a firewall is constructed with three (or more) network interfaces. One network interface attaches to the border router, another interface attaches to the internal network, and a third network interface connects to the DMZ (see Figure 7.3).

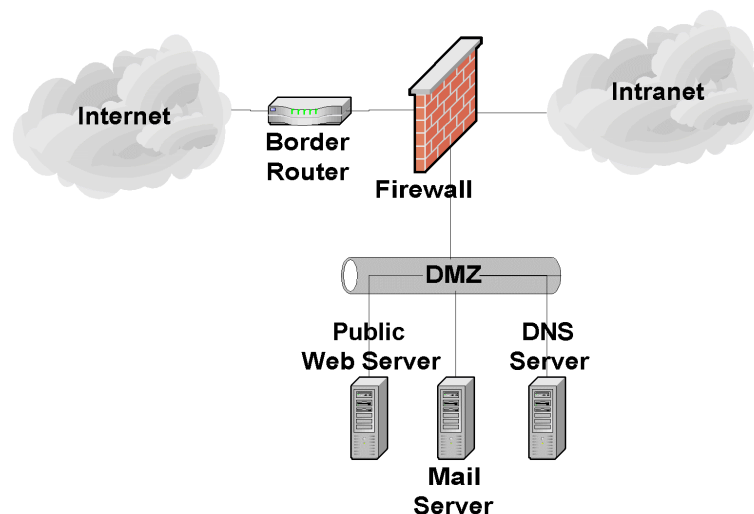


Figure 7.3: Three Interface Firewall DMZ

This configuration subjects the firewall to an increased risk of service degradation during a DoS attack aimed at the DMZ. In a standard DMZ network configuration (discussed above), a DoS

attack against the mail server will generally only affect the mail server. In a service-leg DMZ network configuration, the firewall bears the brunt of any DoS attack because it must examine any network traffic before the traffic reaches the mail server (or any other DMZ or internal network resource). This processing can overwhelm the firewall and slow all traffic, including that destined for the internal network [NIST02a].

The advantages of a DMZ from a security standpoint are as follows:

- Mail server may be better protected and network traffic to and from the mail server can be monitored.
- Compromise of the mail server does not directly threaten the internal production network.
- Greater control can be provided over the security of the mail server since traffic to and from the mail server can be controlled.
- DMZ network configuration can be optimized to support and protect the mail server(s).

The disadvantages of a DMZ from a security standpoint are as follows:

- DoS attacks aimed at the mail server may have an effect on the internal network.
- Depending on the firewall configuration controlling traffic between the DMZ and internal network, it may be possible that the mail server can be used to attack or compromise hosts on the internal network. In other words, protection offered by the DMZ depends in large part on the firewall configuration.

7.2 Using Mail Gateways

To add further protection to a mail server, organizations should consider using a mail gateway in the DMZ in order to better protect the mail server. This adds an additional layer of protection and makes the mail server significantly more difficult to attack. When a mail server is located in the DMZ, it still must communicate with untrusted third-parties. This communication is an access point for attackers. A mail gateway acts as a proxy between the real mail server and the Internet. All messages and communications must go through proxy before they are forwarded to the mail server. This breaks the direct line of communication between the Internet and the mail server making it much more difficult to attack the mail server. Since the mail gateway generally requires only limited functionality, it is much easier to harden and secure than a fully functional mail server. An example of using a mail gateway is provided in Figure

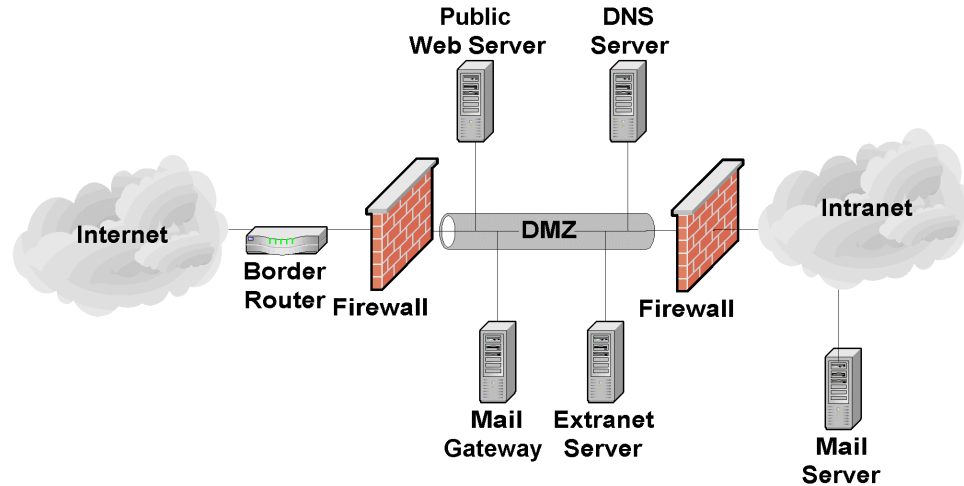


Figure 7.4: Mail Gateway

7.3 Network Element Configuration

Once the mail server has been located in the network, the network infrastructure elements should be configured to support and protect the mail server. The elements of network infrastructure that affect mail server security are firewalls, routers, intrusion detection systems (IDS), and network switches. Each has an important role to play and is critical to the overall strategy of protecting the mail server through layered protection. Unfortunately, when it comes to securing a mail server there is no single “silver bullet” solution. A firewall or IDS alone cannot adequately protect a public mail server from all threats or attacks.

7.3.1 Router/Firewall Configuration

Firewalls and routers are devices or systems that control the flow of network traffic between networks. They protect mail servers from vulnerabilities inherent in the Transport Control Protocol/Internet Protocol (TCP/IP) suite. They also help reduce the security issues associated with insecure applications and operating systems. There are several types of firewalls. The most basic are routers that can provide access control on IP packets. In the middle are stateful firewalls that can provide access control based on TCP and User Datagram Protocol (UDP) as well as IP protocols. The most powerful firewalls that are application layer or proxy firewalls that are able to understand and filter mail content and commands.²⁹

A common misperception of firewalls and routers (acting as a firewall) is that they eliminate all risk and can protect against the misconfiguration of the mail server or poor network design. Unfortunately, this is not the case. Firewalls and routers themselves are vulnerable to misconfiguration and software vulnerabilities. Mail servers in particular are vulnerable to many attacks, even when located behind a secure, well-configured firewall. A firewall or router (acting as a firewall) that is protecting a mail server should be configured to block all access to the mail server from the Internet except TCP port 25 (SMTP), TCP port 110 (POP3), TCP port 143

²⁹ For more information about Firewalls, see NIST Special Publication 800-41, *Guide to Firewall Selection and Policy Recommendations*, (<http://csrc.nist.gov/publications/nistpubs/index.html>).

(IMAP), TCP port 398 (Lightweight Directory Access Protocol [LDAP]), and TCP port 636 (secure LDAP). A firewall is the first line of defense for a mail server; however, to be truly secure, the organization will need to implement layered protection for its mail server (and network). Most importantly, organizations should strive to maintain all systems in a secure manner and not depend solely on routers, firewall(s) or any single component to stop attackers.

A router is able to function as network layer (also called packet filter) firewall. A router functioning as a network layer firewall can provide filtering based on several pieces of information [NIST02a]:

- Source IP address
- Destination IP address
- Traffic type
- TCP/UDP port number (sometimes).

The strengths of routers are as follows:

- Speed
- Lower cost (most organizations already have a border router than can be configured to provide network layer firewall capabilities)
- Mature technology.

The weaknesses of routers are as follows:

- Susceptible to application layer attacks (e.g., cannot examine mail content)
- Does not protect against attacks via allowed ports
- Difficult to configure
- Limited logging capabilities
- Susceptible to IP spoofing attacks
- Limited rule set and filtering capabilities.

In addition to packet filtering routers, the only pure network layer firewalls that are available today are small office home office (SOHO) firewall appliances and personal firewalls [NIST02a] that may only perform packet-level filtering.

Stateful inspection firewalls are transport layer devices that incorporate “awareness” of the state of a TCP connection. Stateful inspection firewalls maintain internal information such as the state of connections passing through them, and the contents of some of the data streams. This allows for better and more accurate rules sets and filtering. Stateful inspection firewalls add the capability to enforce rules based upon connection state to the capabilities of a filtering router.

Application layer firewalls (sometimes called application-proxy gateway firewalls) are advanced firewalls that combine network and transport layer access control with application layer functionality. Application layer firewalls permit no traffic directly between the Internet and the internal network, or between two networks. These components can usually perform extensive logging and access control.

Application layer firewalls are considered the most secure type of firewall and have numerous advantages over packet filtering routers and stateful inspection firewalls:

- Logging capabilities
- Filtering capabilities (can filter specific types of mail content and specific SMTP, POP and IMAP commands)
- Ease of configuration
- Resistance to IP spoofing attacks
- User authentication capabilities.

Application layer firewalls also have some disadvantages as compared to packet filtering routers and stateful inspection firewalls:

- Slower
- Limited support for obscure and new protocols.

Although not strictly a limitation, application layer firewalls tend to be implemented on a workstation running a general-purpose operating system (e.g., Windows, Linux, and Unix). This introduces an added layer of complexity because that general-purpose operating system must also be secured in addition to the firewall software itself. Typically, routers and stateful inspection firewalls run on specialized operating systems, thus reducing this risk.

To successfully protect a mail server using a firewall, ensure that it is capable of and configured to support the following:

- Control all traffic between the Internet and the mail server
- Block all inbound traffic to the mail server except that traffic which is required. This usually includes one or more of the following protocols:
 - TCP port 25 (SMTP)
 - TCP port 110 (POP3)
 - TCP port 143 (IMAP)
 - TCP port 398 (LDAP)
 - TCP port 636 (secure LDAP)

- Block (in conjunction with the intrusion detection system [see Section 7.3.2]) IP addresses or subnets that the IDS reports are attacking the organizational network
- Notify the network or mail server administrator of suspicious activity through an appropriate means (e.g., page, email, network trap)
- Provide content filtering
- Provide virus scanning
- Protect against DoS attacks
- Log critical events, including the following details:
 - Time/date
 - Interface IP address
 - Vendor specific event name
 - Standard attack event (if one exists)
 - Source and destination IP address
 - Source and destination port numbers
 - Network protocol used by attack.
- Patched to the latest or most secure level (firewall application and underlying operating system).

Most firewall devices available in hardware and software perform some type of logging of the traffic they receive. For most firewalls, the default-logging configuration is suitable, provided logging is enabled. Administrators should consult their vendor documentation if they believe they require additional information logged. Certain brands of hardware-based firewalls include an ability to track and log information for each firewall policy. This ability enables accountability to a very specific extent.

Many firewalls support the ability to selectively decide what information to log. If a firewall receives a series of similar packets from the same location, it may decide not to log any additional packets after the first one. Although this is a valuable feature, consider the consequences: each packet that is dropped and not logged is potential evidence of a malicious intent. The principle of logging, which is a fundamental aspect of accountability, is discussed in detail in Section 9.1.

As with operating systems and other security-enforcing elements, a firewall requires updates. Although more prevalent in software implementations of firewall technology, hardware and router firewalls are capable of updating their firmware. Specific instructions on how to update a firewall are found within the vendor documentation. Administrators should check for firewall updates frequently.

7.3.2 Intrusion Detection Systems

An IDS is an application that monitors system and network resources and activities, and uses information gathered from these sources to notify the network administrator when it identifies a possible intrusion or penetration attempt.³⁰

The two principal types of IDSs are host-based and network-based. Host-based IDSs must be installed on each individual computer that is to be monitored or protected. Host-based IDSs are very closely integrated with the operating system of the host computer they protect. Thus, a host-based IDS must be designed specifically for each operating system (and often each version of that operating system). These types of IDSs monitor network traffic to and from the host, the use of system resources, and the system log files.

Host-based IDSs are useful when most of the network traffic to and from the mail server is encrypted (e.g., SSL/TLS, S/MIME, are in use) because the functionality and capability of network-based IDSs (see below) is severely limited when network traffic is encrypted. Host-based IDSs, because they are located on the server, can detect some attacks and penetration attempts not recognized by network-based IDSs.

Host-based IDS can have a negative impact on host performance. In general, the greater the detection capabilities, the greater the negative impact on the performance of the host. Host-based IDSs may not detect certain network-based attacks such as certain DoS attacks [NIST01b]. If a host-based IDS is on a mail server that is compromised, it is very likely that the attacker will also compromise the IDS itself.

Network-based IDS are implemented as protocol analyzers with the capability to recognize particular events. These devices monitor all network traffic on a network segment scrutinizing it for signs of attack or penetration attempts. Most network IDSs rely on predefined “attack signatures” to detect and identify attacks. Attack signatures are a series of events that usually indicate that a particular attack or penetration attempt is in progress. When the IDS detects a series of events that match one of its attack signatures, it assumes that an attack is in progress and notifies the network administrator.

Unlike host-based IDSs, network-based IDSs can monitor multiple hosts and network segments simultaneously. They can usually detect more network-based attacks and can more easily provide a comprehensive picture of the current attacks against a network. Since network-based IDSs are installed on a dedicated host, they do not have a negative effect on the performance of the mail server host and are not immediately compromised by a successful attack on the mail server.

Network-based IDSs do have some limitations. The timing of an attack can have a significant effect on the ability of a network-based IDS to detect an attack. For example, if an intruder spreads out the timing of his attack, the attack may not be detected by the IDS. In addition, the attacker can format the method of the attack (e.g., fragment packets, alter attack pattern, exploit differences in various TCP/IP stacks) so that it is not recognized by the network-based IDS.

³⁰ For more information on IDSs, see NIST Special Publication 800-31, *Intrusion Detection Systems*, (<http://csrc.nist.gov/publications/nistpubs/index.html>).

Network configuration, especially the use of switches (see Section 7.3.3), can have a negative effect on the ability of a network-based IDS to detect attacks. Network-based IDS are also more susceptible to being disabled by DoS attacks (even those not directly targeted at the IDS).

Most host-based and network-based IDSs require frequent updates to their attack signature databases so that they can recognize new attacks. An IDS that is not updated frequently will fail to recognize the latest (and often most popular) attacks.

The following applications have some IDS capabilities and are a useful complement to an IDS although they are not considered to be IDSs.

- **Honey Pot** – is a host(s) that is (are) placed on a network for the strict purpose of attracting and detecting intruders. A honey pot may divert an attacker’s attention from the “real” information system resources and allow an organization to monitor the attacker’s actions without risking “real” organizational information and resources. The real benefit of a honey pot is that since, by definition, it is not used for anything other than detecting attackers, there is a high probability that any network traffic or log entries are an indicator of malicious activity. The danger of a honeypot is that if it is compromised, it may be used to attack other networks potentially making the honeypot host organization liable for damages.³¹
- **File Integrity Checker** – A file integrity checker computes and stores a checksum for every guarded file and establishes a database of file checksums. It provides a tool for the system administrator to recognize changes to files, particularly unauthorized changes. File integrity checkers are often included with host-based IDSs. See Appendix D for a listing of commonly available standalone file integrity checkers.

To successfully protect a mail server using an IDS, ensure that it is capable of and configured to:

- Monitor network traffic to and from the mail server
- Monitor changes to critical files on mail server (host-based or file-integrity checker)³²
- Monitor the system resources available on the mail server host (host-based)
- Block (in conjunction with the firewall) IP addresses or subnets that are attacking the organizational network
- Notify the network administrator or mail server administrator of attacks through appropriate means
- Detect as wide a variety of scanning and attacks as possible with an acceptable level of false positives

³¹ Honey pots and related technologies should be used conservatively and only by organizations with a highly skilled technical staff that are willing to experiment with leading-edge technology. Furthermore, such techniques should be used only after seeking guidance from legal counsel given the possible liability issues. For more information, see NIST Special Publication 800-31 *Intrusion Detection Systems* (<http://csrc.nist.gov/publications/>).

³² Note: certain critical files will change regularly and thus should not be protected by a file integrity checker. This will vary depending on the mail server and operating system employed.

- Log events, including the following details:
 - Time/date
 - Sensor IP address
 - Vendor specific attack name
 - Standard attack name (if one exists)
 - Source and destination IP address
 - Source and destination port numbers
 - Network protocol used by attack.
- Update with new attack signatures frequently (e.g., on a weekly basis).

In addition, it is critical that network-based IDS and their underlying operating system be hardened, as network-based IDSs are often a target of attackers. In particular, the network-based IDSs should not respond to any type of system interrogation through their monitoring interfaces. If remote management is desired, it should be conducted through an out-of-band means (e.g., separate isolated network). Although typically difficult to administer and interpret, IDSs are a critical early warning system that can provide the mail server administrator with the necessary information to defend the mail server from attack.

7.3.3 Network Switches

Network switches are devices that provide connectivity between two or more hosts located on the same network segments. They are similar to hubs in that they allow communications between hosts except that unlike hubs, switches have more “intelligence” and send communications to only those hosts to which the communications are addressed. The benefit of this from a security standpoint is that when switches are employed on a network, it is much more difficult to eavesdrop on communications between other hosts on the network segment. This is extremely important when a mail server is on a network segment that is used by other hosts. For example, if a hub is used and a host on the DMZ is compromised, an attacker may be able to eavesdrop on the communications of other hosts on the DMZ possibly leading to the compromise of those hosts or the information they communicate across the network. A primary example of this would be public Web servers, which are located with mail servers and, if compromised, able to sniff unencrypted mail traffic and passwords within the DMZ.

Many switches include specific security settings that further enhance the security of the network by making it difficult for a malicious entity to “defeat” the switch. Some examples include the ability to minimize the risk of Address Resolution Protocol (ARP) spoofing and ARP poisoning attacks.³³ If a switch has these security capabilities, they should be enabled (see appropriate vendor documentation).

³³ ARP poisoning occurs when an attacker successfully updates the ARP cache on a target host with a forged ARP entry. This is generally used to redirect network traffic for malicious purposes.

Switches can have a negative impact on network-based IDSs (see Section 7.3.2). Network-based IDSs rely on seeing all network traffic on a particular segment and network switches by “hiding” this traffic can impact the ability of the IDS to see suspicious activity. To address this issue most network switches allow the network administrator to configure a specific port on the switch to “mirror” all network traffic. This allows a network-based IDS to see all traffic on a particular network segment.

7.4 Network Infrastructure Checklist

Completed	Action
	Network location
<input type="checkbox"/>	The mail server located on the internal network and protected by a mail gateway and/or firewall, or The mail server is located in a DMZ
	Firewall configuration
<input type="checkbox"/>	Mail server is protected by a firewall
<input type="checkbox"/>	Mail server, if it faces a higher threat or if it is more vulnerable, is protected by an application layer firewall
<input type="checkbox"/>	Firewall controls all traffic between the Internet and the mail server
<input type="checkbox"/>	Firewall blocks all inbound traffic to the mail server except TCP port 25 (SMTP), TCP port 110 (POP3), TCP port 143 (IMAP), TCP port 398 LDAP and TCP port 636 (secure LDAP), if required
<input type="checkbox"/>	Firewall blocks (in conjunction with the intrusion detection system) IP addresses or subnets that the IDS reports are attacking the organizational network
<input type="checkbox"/>	Firewall notifies the network administrator or mail server administrator of suspicious activity through an appropriate means
<input type="checkbox"/>	Firewall provides content filtering (application layer firewall)
<input type="checkbox"/>	Firewall is configured to protect against DoS attacks
<input type="checkbox"/>	Firewall logs critical events
<input type="checkbox"/>	Firewall and firewall operating system patched to latest or most secure level
	Intrusion detection systems
<input type="checkbox"/>	IDS configured to monitor traffic network traffic to and from the mail server after firewall
<input type="checkbox"/>	IDS configured to monitor changes to critical files on mail server (host-based or file-integrity checker)
<input type="checkbox"/>	IDS blocks (in conjunction with the firewall) IP addresses or subnets that are attacking the organizational network
<input type="checkbox"/>	IDS notifies the network or mail server administrator of attacks through appropriate means
<input type="checkbox"/>	IDS configured to maximize detection with an acceptable level of false positives
<input type="checkbox"/>	IDS configured to log events
<input type="checkbox"/>	IDS updated with new attack signatures frequently (e.g., on a weekly basis)
<input type="checkbox"/>	IDS configured to monitor the system resources available on the mail server host (host-based)
	Network switches
<input type="checkbox"/>	Network switches are used to protect against network eavesdropping

Completed	Action
<input type="checkbox"/>	Network switches are configured in high-security mode to defeat ARP spoofing and ARP poisoning attacks
<input type="checkbox"/>	Network switches are configured to send all traffic on network segment to IDS host (network-based)

8. Mail Client Security

Hundreds to thousands of mail clients access every operational mail server. Regardless of whether server security has been improved, it is important to secure the client side. In many respects, the client side represents a greater risk to security than the mail server. Numerous issues need to be carefully considered and addressed to provide an appropriate level of security for email clients. Specific recommendations for securing particular email client applications are not included in this document. This section provides general recommendations that apply to most client applications.

8.1 Securing Installing, Configuring, and Using Client Applications

8.1.1 Patching and Updating Mail Clients

The most important step in securing an email client is to ensure that all users are using the latest and/or most secure version of the mail client with all necessary patches applied.³⁴ Most major email clients have had significant vulnerabilities. To identify the vulnerabilities of a particular mail client, see the NIST ICAT Metabase (<http://icat.nist.gov>). The best resource for patches is the appropriate vendor's Web site:

- Eudora: <http://www.eudora.com/>
- Lotus Notes: <http://www.lotus.com/home.nsf/welcome/downloads>
- Microsoft Outlook: <http://www.microsoft.com/office/outlook/default.htm>
- Microsoft Outlook Express: <http://windowsupdate.microsoft.com/>
- Netscape: <http://home.netscape.com/smartupdate/>

Updating Outlook is made slightly more complicated because it operates in conjunction with Microsoft Internet Explorer. Many of the configuration settings and vulnerabilities of Internet Explorer can have an effect on Outlook; therefore, it also is critical to keep Internet Explorer updated. Failure to run a secure version of a mail client reduces the effectiveness of the rest the security measures discussed below.

8.1.2 Mail Client Security

Mail client applications are rarely configured securely in their default configuration. Manufacturers tend to emphasize usability and performance over security. Although this may be appropriate in many circumstances, it is not appropriate for organizations that wish to maximize their overall security stance. Mail clients should be configured to:

- Disable automatic message preview.

³⁴ When and how to apply patches includes many complex issues that are beyond the scope of this document. For a more detailed discussion of security patches, see NIST Special Publication 800-40, *Applying Security Patches*, (<http://csrc.nist.gov/publications/nistpubs/index.html>).

- Disable automatic opening of next message.
- Disable processing of active content. This may cause problems for email applications that are bundled with Web browsers because disabling this functionality affects the Web browser where such functionality may be required. In those cases, some selective and careful disabling/enabling of active content may be needed. Another work around is to define separate Security Zones for Outlook and Explorer. This allows Explorer to have less restrictive security settings than Outlook.
- Configure “security zones” for Outlook so that [NSA01]:
 - Download signed ActiveX controls is disabled
 - Download unsigned ActiveX controls is disabled
 - Java permissions are disabled
 - Launching programs within an IFRAME is disabled
 - Active scripting is disabled
 - Scripting of Java applets is disabled.

Note that the above settings are for Outlook and Internet Explorer 5.5. Other versions should have similar settings. In addition, these settings will affect both the Outlook mail client and Internet Explorer.

- Configure Eudora so that:
 - “Allow executables in HTML content” is disabled
 - Microsoft viewer is disabled
 - MAPI is disabled.
- Configure Netscape so that:
 - “Enable Java” is unchecked
 - “Enable JavaScript for Mail and News” is unchecked
 - “Send email address as anonymous FTP Password” is unchecked
 - “Microsoft ActiveX Portability Container for Netscape” is removed as are any other plug-in that support ActiveX.

8.1.3 Authentication and Access

Early mail client applications did not require user authentication because mailbox access was restricted by the local file-system and the user owned the mailbox file. As MUAs evolved and provided the functionality to access mailboxes remotely via POP and/or IMAP (see Section 2), user authentication became a requirement. Typically, this was accomplished with users inputting

a username and a password when accessing their mailbox. To be more “user-friendly,” mail clients incorporated configuration files that contained (e.g., “remembered”) usernames and passwords with which to access the mail server. Although this provides ease of use for users, it introduces security weaknesses insofar as a remote intruder or a local intruder (having physical access) to the mail client host may gain access to the authentication information and thus the mailbox contents. In addition, if automatic completion of user input is enabled, a local intruder may be able to use the feature to discover passwords systematically.

To increase mail client security, it is important to disable this functionality. If this functionality cannot be disabled, then it is imperative that these configuration files be kept secure. Many operating systems provide some level of file permissions and access controls that will offer some protection. There are a few popular operating systems such as Windows 95/98/ME that do not. With systems that do provide these controls, ensure that the mail client configuration files are restricted so that they are accessible to only the file owner. Additionally, ensure that the file is located in a directory controlled by the owner. In cases in which a system’s file permissions and access controls are unavailable, the best resolution is to remove the user passwords from the configuration files, unless some form of file encryption is available.

Another area to address is the actual communication between the mail client and mail server. As mentioned in Section 3, all network communication, with the default configurations of SMTP, POP, or IMAP, occurs unencrypted. This makes usernames, passwords, and message content subject to interception and alteration by malicious entities. To increase client to server security, this communication can be encrypted using SSL/TLS. Many of the more commonly used mail clients support SSL/TLS. If SSL/TLS is available, it should be used. At a minimum SSL version 3 should be used, TLS version 1 is preferred.

8.1.4 Client Host Operating System Security

Many host operating systems provide a number of configuration settings and other measures for increasing the security of the email client either directly or indirectly. The host operating system is a critical component of the overall security of client host. The host operating system should be:

- Updated to the most secure patch level.
- Configured to allow access to locally stored messages and mail client configuration files only to appropriate user(s).
- Configure (Windows hosts only) the Windows Scripting Host (WSH) [NSA01]:
 - Remove WSH or allow only administrator to access
 - Change the default action of the following files extensions from execute to edit:³⁵
 - WSC (Windows Script Component)
 - WSH (Windows Script Host Settings File)

³⁵ Note: Not all email clients will accurately interpret these settings. For example certain versions of Netscape will execute files with these extensions, even when the operating system is configured securely.

- WS (Windows Script File)
 - WSF (Windows Script File)
 - VBS (Visual Basic Script)
 - VBE (VBScript Encoded File)
 - JS (JavaScript)
 - JSE (JavaScript Encoded File)
- On Windows hosts, ensure that they are configured to display the full file extensions (this will ensure that an email attachment such as iloveyou.txt.vbs is displayed instead of iloveyou.txt).
 - Install an anti-virus application and configure it to automatically scan all incoming messages and any attachments as they are opened.
 - Ensure that the operating system enforces the concept of least privilege because malicious code runs in the security context on which it was launched (i.e., the user's access level).
 - Ensure that critical components of the operating system are protected from malicious code.³⁶
 - Use a file encrypting application to protect the mail stored locally on the user's hard drive (this is especially important for laptop computers, which are more likely to be stolen).
 - Configure client operating system to automatically lock out after a fixed period of inactivity.

8.2 Secure Message Composition

As with the Internet, email is increasingly being used to conduct business and to transmit sensitive information. Encryption should be used to securely send a message. Two primary methods for encrypting email are S/MIME and PGP. These methods are discussed in Section 3. Both offer similar levels of protection, but their inherent architectures are different. Most mail clients support S/MIME natively, whereas PGP usually comes in the form of a plug-in. Ultimately, the choice comes down to which solution meets the requirements of the organization. As a general rule, unencrypted email should be treated as a postcard – anyone can read and modify.

For a mail client that is configured to send and receive encrypted messages, all received messages should be stored in their encrypted format. The mail client may also be configured to send and receive unencrypted, but authenticated, messages, where integrity is the primary concern. Depending on the sensitivity of the message, the mail client should be set to require a password each time the message is opened for reading.

³⁶For more information, see NIST Special Publication 800-28, *Guidelines on Active Content and Mobile Code*, (<http://csrc.nist.gov/publications/nistpubs/>)

8.3 Plug-ins

A number of different plug-ins are available for mail clients. These plug-ins offer additional functionality beyond that of the basic configuration. Mail encryption and anti-virus plug-ins are a small subset of the types of plug-ins available. Some provide advanced filtering capabilities, whereas others provide audible notification of new messages. Regardless of the type of plug-in, care needs to be taken when installing them. As the general rule, only install plug-ins from trusted sources. Be wary of any plug-in that has not been distributed from the manufacturer in a digitally signed archive. Following these rules will reduce the likelihood of installing a malicious plug-in.

8.4 Accessing Web-Based Email Systems

From a user standpoint, accessing a mail server using a Web server can be efficient and convenient. Unfortunately, a number of security concerns should be carefully considered before implementing Web-based access to mail servers. Many of those are the same as those for standard mail clients. For example, Web-based access in its default configuration sends passwords and data in the clear like POP and IMAP. For greater security, organizations should configure the mail server to accept Web connections only via 128-bit SSL/TLS connections.³⁷ This action will encrypt both user authentication and email content as it is transmitted between the mail Web server and the remote user's Web browser. Note, however, that data between the mail server and the receipt(s) not encrypted, and some form of email encryption, such as S/MIME or PGP would have to be used if message confidentiality is required. Unfortunately, most Web-based email systems do not directly support their use. One solution would be to encrypt the data and then paste it into the browser for transmission (this is easily done with PGP).

Enabling Web-based access often requires a weakening in the overall security posture of the mail server (this is particular true of Microsoft Exchange). Organizations must be aware of the risks and carefully consider whether to implement Web-based access to their mail server (see Section 6.6).

A great risk of Web-based email systems stems is their access from public computers (e.g., college computing lab, Internet café, and public library). In these situations, the browser may be configured to remember (store) usernames and passwords. If it is configured this way, another unauthorized user may obtain access to the organization's mail server using these credentials. Another danger is that a public computer may have a keystroke logger. This logger will register and save all keystrokes, including the username and password entered by the mail user. Again, this data could be used to compromise the organization's mail server. Web browsers will also temporarily cache a user's credentials for a fixed period after the user logs in. If the user fails to empty the browser cache and close the browser after completing access to the mail server, it is possible for an unauthorized user to employ the cached credentials to access the organization's mail server. The use of SSL/TLS will not generally protect against these dangers.

The security of Web-based email is based in large part on the expertise of users. Thus, users should be carefully trained prior to being granted access to Web-based email. Organization's

³⁷ For more information on SSL/TLS and its use with Web servers, see NIST Special Publication 800-44, *Guidelines on Securing Public Web Servers*, and NIST Special Publication 800-52, *Guidelines for the Selection and Use of Transport Layer Security Implementations*, (<http://csrc.nist.gov/publications/nistpubs/index.html>).

should also consider creating a “Rules of Behavior” agreement that each user signs in which he or she acknowledges their responsibility and accountability for the appropriate use of Web-based email.

8.5 Mail Client (MUA) Security Checklist

Completed	Action
	Patching and updating mail clients
<input type="checkbox"/>	Update email client to most secure version
<input type="checkbox"/>	Apply any necessary patches to email client (in conformance with organizational policies and configuration management)
<input type="checkbox"/>	Apply any necessary patches to browser (for email clients that are integrated with browser (e.g., Outlook and Netscape))
	Mail client security
<input type="checkbox"/>	Disable automatic message preview
<input type="checkbox"/>	Disable automatic opening of next message
<input type="checkbox"/>	Disable processing of active content (if appropriate)
<input type="checkbox"/>	Enable secure authentication and access
<input type="checkbox"/>	Disable ability of email client to store username and passwords
<input type="checkbox"/>	Configure client to use data encryption (S/MIME, PGP)
<input type="checkbox"/>	Configure client to store all received message only in encrypted form
<input type="checkbox"/>	Enable and install only absolutely necessary plug-ins from trusted sources
<input type="checkbox"/>	Configure Web-based email access to only use 128-bit SSL/TLS
<input type="checkbox"/>	Educate users on the vulnerabilities of Web-based email access
	Microsoft Outlook Specific Configuration Settings
<input type="checkbox"/>	Disable download signed ActiveX controls
<input type="checkbox"/>	Disable download unsigned ActiveX controls
<input type="checkbox"/>	Disable Java permissions
<input type="checkbox"/>	Disable launching programs within an IFRAME
<input type="checkbox"/>	Disable active scripting
<input type="checkbox"/>	Disable scripting of Java applets
	Eudora Specific Configuration Settings
<input type="checkbox"/>	Disable “Allow executables in HTML content”
<input type="checkbox"/>	Disable Microsoft viewer
<input type="checkbox"/>	Disable MAPI
	Netscape Specific Configuration Settings
<input type="checkbox"/>	Uncheck “Enable Java”
<input type="checkbox"/>	Uncheck “Enable JavaScript for Mail and News”
<input type="checkbox"/>	Uncheck “Send email address as anonymous FTP Password”
<input type="checkbox"/>	Remove Microsoft ActiveX Portability Container for Netscape

Completed	Action
	Mail client host operating system security
<input type="checkbox"/>	Ensure operating system is updated to most secure patch level
<input type="checkbox"/>	Configure operating system to allow access to locally stored messages and mail client configuration files only to appropriate user(s)
<input type="checkbox"/>	Secure or remove Windows Scripting Host (Windows hosts only)
<input type="checkbox"/>	Change the default action on files associated with the Windows Scripting Host from execute to edit (Windows hosts only)
<input type="checkbox"/>	Ensure that operating system is configured to show full file extensions (Windows hosts only)
<input type="checkbox"/>	Ensure the operating system enforces the concept of least privilege because malicious code runs in the security context on which it was launched (i.e., the user's access level)
<input type="checkbox"/>	Ensure that critical components of the operating system are protected from malicious code
<input type="checkbox"/>	Use a file encrypting system to protect the mail stored locally on the user's hard drive (especially important for laptop computers)
<input type="checkbox"/>	Configure client operating system to automatically lock out after a fixed period of inactivity

9. Securely Administering a Mail Server

9.1 Logging

Logging is a cornerstone of a sound security posture. Logging the correct data and then monitoring those logs is critical. Host logs are important, especially in the case of S/MIME or PGP (see Section 3) enabled mail servers, where network monitoring is less effective. Unfortunately, reviewing logs is mundane and reactive, and many mail server administrators do not know where to locate the appropriate log files. These log files are often the only record of suspicious behavior. It is important to enable the mechanisms to record this information and use them to initiate alert mechanisms to detect failed and successful intrusion attempts. Similar problems can result from not having the necessary procedures and tools in place to process and analyze the log files.

System and network logs can alert the mail administrator that a suspicious event has occurred and that further investigation is needed. Mail server software can provide additional log data relevant to mail-specific events.

Mail server logs provide:

- Alerts to suspicious activities that require(s) further investigation
- Tracking of an intruder's activities
- Assistance in the recovery of the system
- Assistance in the post-event investigation
- Required information for legal proceedings.

The selection and implementation of specific mail server software will determine which set of detailed instructions the mail administrator should follow to establish logging configurations. Some of the guidance contained in the steps below may not be fully applicable to all vendors' mail server software products.

9.1.1 Recommended Generic Logging Configuration

Although the logging capabilities of a mail server will vary for each product, the following generic configuration is recommended. Set logging on the mail server to the most detailed level available (e.g., "maximum," "detailed," "level 9").³⁸ Once the most detailed log level is set, ensure the following events are logged (if supported by the mail server software):

- Local host related logging

³⁸ Note: setting the logging to the most detailed level as recommended here may dramatically increase the amount space required for the log files. It is recommend that organizations closely monitor the size of the log files when they implement this level of logging to ensure that the log files do not fill up the hard drive. Due to the size of the log files, it may also be necessary to remove and archive the logs more frequently or carefully reduce the logging level of detail.

- IP stack setup errors
- Resolver configuration problem (e.g., DNS, NIS, Windows Internet Naming Service [WINS])
- Mail server configuration errors (e.g., mismatch with DNS: local configuration error, out of date alias database)
- Out of date alias database(s)
- Lack of system resources (disk space, memory, CPU)
- Alias database rebuilds
- Connection related logging
 - Logons (successful and failed)
 - Security problems (e.g., spamming)
 - Lost communications (network problems)
 - Protocol failures
 - Connection timeouts
 - Connection rejections
 - Use of VRFY and EXPN commands
- Message-related logging
 - Send on behalf of
 - Send as
 - Malformed addresses
 - Message collection statistics
 - Creation of error messages
 - Delivery failures (permanent errors)
 - Messages being deferred (transient errors).

Some mail server software provides a capability to enforce or disable the checking of specified access controls during program startup. This level of control may be helpful, for example, to avoid inadvertent alteration of log files because of errors in file access administration. Mail server administrators should determine the circumstances under which they may wish to enable such checks (assuming the mail server software supports this feature).

9.1.2 Reviewing and Retaining Log Files

Reviewing log files can be time-consuming and mundane. Log files are an inherently reactive security measure: they inform of events that have already occurred. Accordingly, they are often useful for corroborating other evidence, whether it is a CPU utilization spike or anomalous network traffic reported by an IDS. When a log is used to corroborate other evidence, a focused review is in order. For example, if an IDS reported an inbound connection to the mail server at 8:17 a.m. that attempted to use the VRFY command, then a review of the logs generated just before 8:17 a.m. is appropriate. Mail server logs should also be reviewed for indications of attacks or spamming. The frequency of the review will depend on the following factors:

- Traffic the server receives
- General threat level (certain sites, in particular the Federal Government and certain commercial institutions' receive many more attacks than other sites and thus should review their logs more frequently)
- Specific threats (at certain times specific threats arise that may require more frequent log file analysis as a result)
- Vulnerability of the mail server
- Value of data and services provided by mail server.

Reviews should take place on a daily to weekly basis and when a suspicious activity has been noted or a threat warning has been issued. Obviously, the task could quickly become burdensome to a mail administrator. To reduce this burden, automated log file analysis tools have been developed (see Section 9.1.3).

In addition, a long-term and more in-depth analysis of the logs is needed. Because a typical mail server attack can involve hundreds of unique requests, an attacker may attempt to disguise a mail attack by increasing the interval between requests. In this case, reviewing a single day's or week's logs may not show recognizable trends. However, when trends are analyzed over a week, month, or quarter, multiple attacks from the same host or subnet are more easily recognized.

Log files should be protected to ensure that if an attacker does compromise a mail server, the log files cannot be altered to cover the attack. Although encryption can be useful in protecting log files, the best solution is to store log files on a host separate from the mail server(s). This is often called a log or syslog host.

Log files should be backed up and archived regularly. Archiving log files for a period of time is important for several reasons. They can be important for certain legal actions. They are often useful in troubleshooting problems with the mail server. The retention period for archived log files depends on a number of factors, including:

- Legal requirements
- Organizational requirements
- Size of logs (which is directly related to the traffic of the site and the number of details logged)

- Value of mail server data and services
- Threat level.

9.1.3 Automated Log File Analysis Tools

Most mail servers receive significant amounts of traffic, and the log files quickly become voluminous. Automated log file analysis tools should be installed to ease the burden on the mail server administrator. These tools analyze the entries in the mail server log file and identify suspicious and unusual activity.

Many commercial and public domain tools are available to support regular analysis. The automated log file analyzer should forward any suspicious log file events to the responsible mail administrator or security incident response team as soon as possible for follow-up investigation. Some organizations may wish to use two or more log file analyzers, which will reduce the risk of missing an attack or other significant events in the log files. A list of some commonly used log file analyzers is provided in Appendix D.

9.2 Mail Server Backup Procedures

One of the most important functions of a mail server administrator is to maintain the integrity of the data on the mail server. This is important because mail servers are often one of the most exposed and vital servers on an organization's network. Mail servers are often the targets of malicious actions in addition to possible hardware and software failures.

9.2.1 Mail Server Backup Policies and Strategies

The mail administrator needs to perform backups of the mail server on a regular basis for several reasons. A mail server fails as a result of a malicious or unintentional act or a hardware or software failure. In addition, Federal and state governmental organizations are governed by regulations on the back up and archiving of mail server data. Commercial organizations should also backup their mail server data on a regular basis for legal and/or financial reasons.

All organizations need to create a mail server data backup policy. The contents of this policy will be influenced by three factors:

- Legal requirements
 - Applicable laws and regulations (Federal, state, and international)
 - Litigation requirements
- Mission requirements
 - Contractual
 - Common practices
 - Criticality of data to organization
- Organizational guidelines and policies.

Although each organization's mail server backup policy will be different to reflect its particular environment, it should address the following issues:

- Purpose of the mail server backup policy
- Who is affected by the mail server backup policy
- Which mail servers are covered by the backup policy
- Define key terms, especially legal and technical
- Describe the requirements in detail from the legal, business, and organization's perspective
- Outline frequency of backups
- Outline the procedures for ensuring data is properly retained and protected
- Outline the procedures for ensuring data is properly destroyed or archived when no longer required
- Clearly document the litigation exception process and how to respond to discovery requests
- List the responsibilities of those involved in data retention, protection and destruction activities
- Build a table showing the information type and its corresponding retention period
- Document the specific duties of a central/organizational data backup team if one exists.

Three primary types of backups exist. Full backups include the operating system, applications, and data stored on the mail server (i.e., an image of every piece of data stored on the mail server hard drive[s]). The advantage of a full backup is that it is easy to restore the entire mail server back to the state (i.e., configuration, patch level, data) it was in when the backup was performed. The disadvantage of full backups is that they take considerable time and resources to perform. Incremental backups reduce the impact by backing up only data that has changed since the previous backup (either full or incremental). Differential backups reduce the number of backup sets that must be accessed to restore a configuration by backing up all changed data since the last full-backup. However, each differential backup will become ever larger as time lapses from last full backup, taking more time and storage than an incremental backup. Generally, full backups are performed less frequently (weekly to monthly or when a significant change occurs) and incremental or differential backups performed more frequently (daily to weekly). The frequency of backups will be determined by several factors:

- Volatility of information on and configuring of the mail server
- Amount of data to be backed up
- Backup device and media available
- Time available for dumping backup data
- Criticality of data

- Threat level faced by mail server
- Effort required to data reconstruction without data backup
- Other data backup or redundancy features of mail server (e.g., Redundant Array of Inexpensive Disks (RAID), mirroring, etc.).

When archiving or backing up e-mail data, organizations should generally conform to the following guidelines:

- Employ write once, read many media (to prevent the alteration or accidental erasure of archived information).
- Contain a verification capability to ensure that data is being correctly backed up or archived.
- Include the capability to serialize and time-date the information stored.
- Provide for easy retrieval of indexes and records preserved on the backup media.
- Maintain at least two copies in two geographically distinct locations.
- Accurately organize and index all information maintained on both original and any duplicate storage media.

9.3 Recovering From a Security Compromise

Most organizations will eventually face a successful compromise of one or more hosts on their network. The first step in recovering from a compromise is to create and document the required policies and procedures for responding to successful intrusions *prior* to an intrusion.³⁹ The response procedures should outline the actions that are required to respond to a successful compromise of the mail server and the appropriate sequence of these actions (sequence can be critically important). These response procedures would be contained within the organization's security policy. In addition, the organization may wish to ensure that some or all of its system administrators are knowledgeable in the field of computer forensics.

A mail server administrator should take the following steps after discovering a successful compromise:

- Report incident to organization's computer incident response capability
- Consult the organization's security policy

³⁹ For more information on this area, see NIST Special Publication 800-3, *Establishing a Computer Security Incident Response Capability* and NIST Special Publication 800-18, *Guide to Developing Security Plans for Information Technology Systems*, (<http://csrc.nist.gov/publications/nistpubs/index.html>).

- Isolate compromised system(s) or take steps to contain attack so additional evidence can be collected⁴⁰
- Consult, as appropriate, with management, legal counsel, and law enforcement expeditiously
- Investigate “similar”⁴¹ hosts to determine if the attacker also has compromised other systems
- Analyze the intrusion, including:
 - Modifications made to the system’s software and configuration
 - Modifications made to the data
 - Tools or data left behind by intruder
 - Review system logs, intrusion detection, and firewall log files.
- Restore the system
 - One of the following two options:
 - Install clean version of operating system, or
 - Restore from backups (this option can be more risky, as the backups may have been made after the compromise and restoring from a compromised back may still allow the attacker access to the system).
 - Disable unnecessary services
 - Apply all patches
 - Change all passwords (even on uncompromised hosts as required)
 - Reconfigure network security elements (firewall, router, IDS) to provide additional protection and notification.
- Test system to ensure security
- Reconnect system to network
- Monitor system and network for signs that the attacker is attempting to access the system or network again
- Document lessons learned.

⁴⁰ Isolating the system must be accomplished with great care if the organization wishes to collect evidence. Many attackers now configure compromised systems to erase evidence if a compromised system is disconnected from the network or rebooted. One method to isolate a system would be to reconfigure the nearest upstream switch or router.

⁴¹ “Similar” would include hosts in the same IP address range, that have the same or similar passwords, that share a trust relationship, and/or that have the same operating system and/or applications.

System administrators should consider the following when deciding whether to reinstall the operating system of a compromised system as opposed to restoring from a backup:

- Level of access that the intruder gained (e.g., root, user, guest, system)
- Type of attacker (internal or external)
- Purpose of compromise (e.g., mail spoofing, illegal software repository, platform for other attacks)
- Method of system compromise
- Actions of hacker during and after compromise (e.g., log files, intrusion detection reports)
- Duration of compromise
- Extent of compromise on network (i.e., the number of hosts compromised)
- Results of consultation with management and legal counsel.

The lower the level of access gained by the intruder and the more the mail server administrator understands about the hacker's actions, the less risk there is in restoring from a backup and patching the vulnerability. For incidents in which there is less known about the intruder's actions and/or in which the intruder gains high-level access, it is recommended that the operating system and applications be reinstalled from the manufacturer's original distribution media and that the mail server data be restored from a known good backup.

If legal action is pursued, system administrators need to be aware of the guidelines for handling a host after a compromise. Consult legal counsel and relevant law enforcement authorities as appropriate.

9.4 Security Testing Mail Servers

Periodic security testing of public mail servers is critical. Without periodic testing, there is no assurance that current protective measures are working or that the security patch just applied by the mail server administrator is functioning as advertised. Although a variety of security testing techniques exists, vulnerability scanning is the most common. Vulnerability scanning assists a mail server administrator in identifying vulnerabilities and verifying whether the existing security measures are effective. Penetration testing is also used, less frequently and usually only as part of an overall penetration test of the organization's network.⁴²

9.4.1 Vulnerability Scanning

Vulnerability scanners are automated tools that are used to identify vulnerabilities and misconfiguration of hosts. Many vulnerability scanners also provide information about mitigating discovered vulnerabilities.

⁴² For information about other testing techniques, see NIST Special Publication 800-42, *Guideline on Network Security Testing* (<http://csrc.nist.gov/publications/nistpubs/index.html>).

Vulnerability scanners attempt to identify vulnerabilities in the hosts scanned. Vulnerability scanners can help identify out-of-date software versions, missing patches or system upgrades, and validate compliance with, or deviations from, the organization's security policy. To accomplish this effort, vulnerability scanners identify operating systems and major software applications running on hosts and match them with known vulnerabilities. Vulnerability scanners employ large databases of vulnerabilities to identify vulnerabilities associated with commonly used operating systems and applications.

However, vulnerability scanners have some significant weaknesses. Generally, they identify only surface vulnerabilities and are unable to address the overall risk level of a scanned mail server. Although the scan process itself is highly automated, vulnerability scanners can have a high false positive error rate (reporting vulnerabilities when none exist). This means an individual with expertise in Web server security and administration must interpret the results. Furthermore, vulnerability scanners cannot identify vulnerabilities in custom code or applications.

Vulnerability scanners rely on periodic updating of the vulnerability database to recognize the latest vulnerabilities. Before running any scanner, mail server administrators should install the latest updates to its vulnerability database. Some vulnerability scanner databases are updated more regularly than others (the frequency of updates should be a major consideration when choosing a vulnerability scanner).

Vulnerability scanners are often better at detecting well-known vulnerabilities rather than more esoteric ones because it is impossible for any one product to incorporate all known vulnerabilities in a timely manner. In addition, manufacturers want to keep the speed of their scanners high (more vulnerabilities detected requires more tests, which slows the overall scanning process). Therefore, vulnerability scanners may be of little use to Web administrators operating less popular Web servers, operating systems or custom coded applications.

Vulnerability scanners provide the following capabilities:

- Identifying active hosts on network
- Identifying active services (ports) on hosts and which of these are vulnerable
- Identifying applications and banner grabbing
- Identifying operating systems
- Identifying vulnerabilities associated with discovered operating systems and applications
- Testing compliance with host application usage/security policies.

Organizations should conduct vulnerability scanning to validate that operating systems and mail server applications are up to date on security patches and software versions. Vulnerability scanning is a labor-intensive activity that requires a high degree of human involvement to interpret the results. It may also be disruptive to network operations by taking up bandwidth and slowing response times. However, vulnerability scanning is extremely important for ensuring that vulnerabilities are mitigated as soon as possible, before they are discovered and exploited by adversaries. Vulnerability scanning should be conducted on a weekly to monthly basis. Many organizations also run a vulnerability scan whenever a new vulnerability database is released for the organization's scanner application.

Organizations should also consider running more than one vulnerability scanner. As previously discussed, no scanner is able to detect all known vulnerabilities, however using two scanners will generally increase the number of vulnerabilities detected. A common practice is to use one commercial and one freeware scanner.

Vulnerability scanning results should be documented and discovered deficiencies corrected. The following corrective actions may be necessary as a result of vulnerability scanning: Network- and host-based vulnerability scanners are available for free or for a fee. Appendix D contains a list of readily available vulnerability scanning tools.

9.4.2 Penetration Testing

“Penetration testing is security testing in which evaluators attempt to circumvent the security features of a system based on their understanding of the system design and implementation” [NISS99]. The purpose of penetration testing is to exercise system protections (particularly human response to attack indications) by using common tools and techniques developed by hackers. This testing is highly recommended for complex or critical systems.

Penetration testing can be an invaluable technique to any organization's information security program. However, it is a very labor-intensive activity and requires great expertise to minimize the risk to targeted systems. At a minimum, it may slow the organization's network response time resulting from network mapping and vulnerability scanning. Furthermore, the possibility exists that systems may be damaged in the course of penetration testing and may be rendered inoperable. Although this risk is mitigated by the use of experienced penetration testers, it can never be fully eliminated.

Penetration testing does offer the following benefits [NIST02b]:

- Tests the network using the same methodologies and tools employed by hackers
- Verifies whether vulnerabilities exist
- Goes beyond surface vulnerabilities and demonstrates how these vulnerabilities can be exploited iteratively to gain greater access
- Demonstrates that vulnerabilities are not purely theoretical
- Provides the “realism” necessary to address security issues
- Allows for testing of procedures and the susceptibility of the human element to social engineering.

9.5 Remotely Administering a Mail Server

It is strongly recommended that remote administration of a mail server be allowed only after careful consideration of the risks. The most secure configuration is to disallow any remote administration. However, that may not be viable for all organizations. The risk of enabling remote administration varies considerably depending on the location of the mail server on the network (see Section 7.1). For a mail server that is located behind a firewall, remote administration or content updating can be implemented relatively securely from the internal

network, but not without added risk. Remote administration should generally not be allowed from a host located outside the organization's network.

If an organization determines that it is necessary to remotely administer or update content on a mail server, following these steps should ensure that it is implemented in as secure manner as possible:

- Use a strong authentication mechanism (e.g., public/private key pair, two factor authentication).
- Restrict hosts that can be used to remotely administer or update content on the mail server:
 - Restrict by authorized users
 - Restrict by IP address (not hostname)
 - Restrict to hosts on the internal network.
- Use more secure protocols (e.g., secure shell, Secure Hypertext Transfer Protocol (HTTPS)) and not less secure protocols (e.g., Telnet, FTP, NFS, or HTTP). Secure in this instance is defined as those protocols that provide encryption of both passwords and data.
- Enforce the concept of least privilege on the remote administration and content updating (i.e., attempt to minimize the access rights for the remote administration/update account(s)).
- Do not allow remote administration from the Internet through the firewall unless accomplished via strong mechanisms such as virtual private networking (VPN) tunneling.
- Change any default accounts or passwords from the remote administration utility or application.
- Do not mount any file shares on the internal network from the mail server or vice versa.

9.6 Securely Administering a Mail Server Checklist

Completed	Action
	Logging
<input type="checkbox"/>	Log IP stack setup errors
<input type="checkbox"/>	Log resolver configuration problem (e.g., DNS, NIS, WINS)
<input type="checkbox"/>	Log mail server configuration errors (e.g., mismatch with DNS, local configuration error, out-of-date alias database)
<input type="checkbox"/>	Log out of date alias database(s)
<input type="checkbox"/>	Log lack of system resources (e.g., disk space, memory, CPU)
<input type="checkbox"/>	Log alias database rebuilds
<input type="checkbox"/>	Log logons (successful and failed)
<input type="checkbox"/>	Log security problems (e.g., spamming)
<input type="checkbox"/>	Log lost communications (network problems)
<input type="checkbox"/>	Log protocol failures

Completed	Action
<input type="checkbox"/>	Log connection timeouts
<input type="checkbox"/>	Log connection rejections
<input type="checkbox"/>	Log use of VRFY and EXPN commands
<input type="checkbox"/>	Log send on behalf of
<input type="checkbox"/>	Log send as
<input type="checkbox"/>	Log malformed addresses
<input type="checkbox"/>	Log message collection statistics
<input type="checkbox"/>	Log creation of error messages
<input type="checkbox"/>	Log delivery failures (permanent errors)
<input type="checkbox"/>	Log messages being deferred (transient errors)
<input type="checkbox"/>	Store logs on a separate (syslog) host
<input type="checkbox"/>	Archive logs according to organizational requirements
<input type="checkbox"/>	Review logs daily
<input type="checkbox"/>	Review logs weekly (for more long-term trends)
<input type="checkbox"/>	Use automated log file analysis tool(s)
	Mail server backups
<input type="checkbox"/>	Create a mail server backup policy
<input type="checkbox"/>	Back up mail server differentially or incrementally on a daily to weekly basis
<input type="checkbox"/>	Back up mail server fully on a weekly to monthly basis
<input type="checkbox"/>	Periodically archive backups
	Recovering from a compromise
<input type="checkbox"/>	Report incident to organization's computer incident response capability
<input type="checkbox"/>	Consult the organization's security policy
<input type="checkbox"/>	Isolate compromised system(s) or take steps to contain attack so additional evidence can be collected
<input type="checkbox"/>	Consult, as appropriate, with management, legal counsel, and law enforcement officials expeditiously
<input type="checkbox"/>	Investigate other "similar" hosts to determine if the attacker has also compromised other systems
<input type="checkbox"/>	Analyze the intrusion
<input type="checkbox"/>	Restore the system
<input type="checkbox"/>	Test system to ensure security
<input type="checkbox"/>	Reconnect system to network
<input type="checkbox"/>	Monitor system and network for signs that the attacker is attempting to access the system or network again
<input type="checkbox"/>	Document lessons learned
	Security testing
<input type="checkbox"/>	Periodically conduct vulnerability scans on mail server and network supporting network
<input type="checkbox"/>	Update vulnerability scanner before testing
<input type="checkbox"/>	Correct any deficiencies identified by the vulnerability scanner
<input type="checkbox"/>	Conduct penetration testing on the mail server on the support network infrastructure

Completed	Action
<input type="checkbox"/>	Correct deficiencies identified by penetration testing
	Remote administration
<input type="checkbox"/>	Use a strong authentication mechanism (e.g., public/private key pair, two factor authentication)
<input type="checkbox"/>	Restrict hosts that can be used to remotely administer on the mail server by IP address and to the internal network
<input type="checkbox"/>	Use secure protocols (e.g., secure shell, HTTPS)
<input type="checkbox"/>	Enforce the concept of least privilege on the remote administration (i.e., attempt to minimize the access rights for the remote administration/update account[s])
<input type="checkbox"/>	Change any default accounts or passwords for the remote administration utility or application
<input type="checkbox"/>	Do not allow remote administration from the Internet unless mechanisms such as VPN are used

Appendix A. Glossary

Address Resolution Protocol—A protocol used to obtain a node's physical address. A client station broadcasts an Address Resolution Protocol (ARP) request onto the network with the Internet Protocol (IP) address of the target node it wishes to communicate with, and the node with that address responds by sending back its physical address so that packets can be transmitted.

Host – The term can refer to almost any kind of computer, from a centralized mainframe that is a host to its terminals, to a server that is host to its clients, to a desktop personal computer (PC) that is host to its peripherals. In network architectures, a client station (user's machine) is also considered a host because it is a source of information to the network in contrast to a device such as a router or switch that directs traffic.

Host Operating System – The host operating system is the software “master control application” that runs the computer. It is first program loaded when the computer is turned on, its principal component, the “kernel,” resides in memory at all times. The operating system sets the standards for all application programs (such as the mail server) that run in the computer. The applications communicated the operating system for most user interface and file management operations.

Hotfix – Microsoft's term for a bug fix, which is accomplished by replacing one or more existing files in the operating system or application with revised versions.

Mail Server – A host that provides “electronic post office” facilities. It stores incoming mail for distribution to users and forwards outgoing mail. The term may refer to just the application that performs this service, which can reside on a machine with other services but for this document mail server refers to the entire host including mail server application, host operating system and the supporting hardware.

Mail Server Administrator – The mail server equivalent of a system administrator. Mail server administrators are system architects responsible for the overall design and implementation of a mail server.

Mail Transfer Agent (MTA) – It receives messages from mail user agents or other MTAs and either forwards them to another MTA or, if the recipient is on the MTA, it delivers the message to the local deliver agent (LDA) for delivery to the recipient. Common MTAs include Microsoft Exchange and sendmail.

Mail User Agent (MUA) – An application used by an end user to access his or her mail server to read, compose, and send email messages. Common MUAs include Microsoft Outlook and Netscape Messenger.

Network Administrator – A person who manages a local area network (LAN) within an organization. Responsibilities include network security, installing new applications, distributing software upgrades, monitoring daily activity, enforcing licensing agreements, developing a storage management program, and providing for routine backups.

Patch – A patch (sometimes called a “fix”) is a “repair job” for a piece of programming. A patch is the immediate solution that is provided to users; it can sometimes be downloaded from the software maker's Web site. The patch is not necessarily the best solution for the problem, and the product developers often find a better solution to provide when they package the product for its

next release. A patch is usually developed and distributed as a replacement for or an insertion in compiled code (that is, in a binary file or object module). In larger operating systems, a special program is provided to manage and track the installation of patches.

Service Pack – A software patch that is applied to an installed application. It is either downloaded from the vendor's Web site or distributed via Compact Disc Read-Only Memory (CD-ROM). When executed, it modifies the application in place.

System – See host.

System Administrator – A person who manages a multi-user computer system. Responsibilities are similar to that of a network administrator. A system administrator would perform systems programmer activities with regard to the operating system and other network control programs.

Vulnerability – A security exposure in an operating system or other system software or application software component. A variety of organizations maintain publicly accessible databases of vulnerabilities based on version number of the software. Each vulnerability can potentially compromise the system or network if exploited.

Web Server – A computer that provides World Wide Web (WWW) services on the Internet. It includes the hardware, operating system, Web server software, Transport Control Protocol (TCP)/Internet Protocol (IP), and Web site content (Web pages). If the Web server is used internally and not by the public, it may be known as an "intranet server."

Appendix B. Mail-Related RFCs

- 1) RFC 822, *Standard for the Format of ARP Internet Text Messages*, <http://www.ietf.org/rfc/rfc0822.txt>
- 2) RFC 2045, *MIME, Part One: Format of Internet Message Bodies*, <http://www.ietf.org/rfc/rfc2045.txt>
- 3) RFC 2046, *MIME, Part Two: Media Types*, <http://www.ietf.org/rfc/rfc2046.txt>
- 4) RFC 2047, *MIME, Part Three: Message Header and Extensions for Non-ASCII Text*, <http://www.ietf.org/rfc/rfc2047.txt>
- 5) RFC 2048, *MIME, Part Four: Registration Procedures*, <http://www.ietf.org/rfc/rfc2048.txt>
- 6) RFC 2049, *MIME, Part Five: Conformance Criteria and Examples*, <http://www.ietf.org/rfc/rfc2049.txt>
- 7) RFC 821, *Simple Mail Transport Protocol*, <http://www.ietf.org/rfc/rfc0821.txt>
- 8) RFC 1869, *SMTP Service Extensions*, <http://www.ietf.org/rfc/1869.txt>
- 9) RFC 1870, *SMTP Service Extension for Message Size Declaration*, <http://www.ietf.org/rfc/1870.txt>
- 10) RFC 2920, *SMTP Service Extension for Command Pipelining*, <http://www.ietf.org/rfc/2920.txt>
- 11) RFC 3030, *SMTP Service Extensions for Transmission of Large and Binary MIME Messages*, <http://www.ietf.org/rfc/3030.txt>
- 12) RFC 2554, *SMTP Service Extension for Authentication*, <http://www.ietf.org/rfc/2554.txt>
- 13) RFC 2487, *SMTP Service Extension for Secure SMTP Over TLS*, <http://www.ietf.org/rfc/2487.txt>
- 14) RFC 2034, *SMTP Service Extension for Returning Enhanced Error Codes*, <http://www.ietf.org/rfc/2034.txt>
- 15) RFC 1985, *SMTP Service Extension for Remote Message Queue Starting*, <http://www.ietf.org/rfc/1985.txt>
- 16) RFC 1891, *SMTP Service Extension for Delivery Status Notifications*, <http://www.ietf.org/rfc/1891.txt>
- 17) RFC 1939, *Post Office Protocol Version 3*, <http://www.ietf.org/rfc/1939.txt>
- 18) RFC 2060, *IMAP4 Revision 1*, <http://www.ietf.org/rfc/2060.txt>
- 19) RFC 2087, *IMAP4 QUOTA Extension*, <http://www.ietf.org/rfc/2087.txt>

- 20) RFC 2088, *IMAP4 Non-Synchronizing Literals*, <http://www.ietf.org/rfc/2088.txt>
- 21) RFC 2177, *IMAP4 IDLE Command*, <http://www.ietf.org/rfc/2177.txt>
- 22) RFC 2180, *IMAP4 Multi-Accessed Mailbox Practice*, <http://www.ietf.org/rfc/2180.txt>
- 23) RFC 2192, *IMAP URL Scheme*, <http://www.ietf.org/rfc/2192.txt>
- 24) RFC 2193, *IMAP4 Mailbox Referrals*, <http://www.ietf.org/rfc/2193.txt>
- 25) RFC 2195, *IMAP/POP AUTHorize Extension for Simple Challenge/Response*,
<http://www.ietf.org/rfc/2195.txt>
- 26) RFC 2221, *IMAP4 Login Referrals*, <http://www.ietf.org/rfc/2221.txt>
- 27) RFC 2342, *IMAP4 Namespaces*, <http://www.ietf.org/rfc/2342.txt>
- 28) RFC 2359, *IMAP4 UIDPLUS Extension*, <http://www.ietf.org/rfc/2359.txt>
- 29) RFC 2971, *IMAP4 ID Extension*, <http://www.ietf.org/rfc/2971.txt>
- 30) RFC 2311, *S/MIME Version 2 Message Specification*, <http://www.ietf.org/rfc/2311.txt>
- 31) RFC 2312, *S/MIME Version 2 Certificate Handling*, <http://www.ietf.org/rfc/2312.txt>
- 32) RFC 2633, *S/MIME Version 3 Message Specification*, <http://www.ietf.org/rfc/2633.txt>
- 33) RFC 2632, *S/MIME Version 3 Certificate Handling*, <http://www.ietf.org/rfc/2632.txt>
- 34) RFC 2634, *Enhanced Security Services for S/MIME*, <http://www.ietf.org/rfc/2634.txt>
- 35) RFC 1991, *PGP Message Exchange Formats*, <http://www.ietf.org/rfc/1991.txt>
- 36) RFC 2015, *MIME Security With Pretty Good Privacy*, <http://www.ietf.org/rfc/2015.txt>
- 37) RFC 3156, *MIME Security With OpenPGP*, <http://www.ietf.org/rfc/3156.txt>

Appendix C. References

- [Allmn] Eric Allman and Greg Shapiro, *Securing Sendmail*, <http://www.sendmail.net/000705securitygeneral.shtml>
- [CERT00] *Securing Network Servers*, 2000, <http://www.cert.org/security-improvement/modules/m10.html>
- [Curt01] Matt Curtin, *Developing Trust: Online Privacy and Security*, November 2001
- [Hard02] John Harden, *Enhancing E-Mail Security With Procmail the E-Mail Sanitizer*, <http://www.impsec.org/email-tools/procmail-security.html>
- [NIST01a] Wayne A. Jansen, NIST Special Publication 800-28, *Guidelines on Active Content and Mobile Code*, October 2001, <http://csrc.nist.gov/publications/nistpubs/index.html>
- [NIST01b] Rebecca Bace and Peter Mell, NIST Special Publication 800-31, *Intrusion Detection Systems*, August 2001, <http://csrc.nist.gov/publications/nistpubs/index.html>
- [NIST02a] John Wack, et al., NIST Special Publication 800-41, *Guidelines on Firewalls and Firewall Policy*, January 2002, <http://csrc.nist.gov/publications/nistpubs/index.html>
- [NIST02b] John Wack, et al., NIST Special Publication 800-42, *Guideline on Network Security Testing*, February 2002, <http://csrc.nist.gov/publications/nistpubs/index.html>
- [NSA01] Trent Pitsenbarger, *Email Security in the Wake of Recent Malicious Code Incidents*, 2001, <http://nsa1.www.conxion.com/>
- [NSA02] Trent Pitsenbarger, *NSA Guide to the Secure Configuration and Administration of Microsoft Exchange*, 2002, <http://nsa1.www.conxion.com/>
- [NISS99] *National Information System Security Glossary*, NSTISSI No. 4009, January 1999
- [PCMAG01] Ashley McKinnon, *Web and Email Filtering*, *PC Magazine*, December 2001, <http://www.zdnet.co.uk/pcmag/labs/2001/12/filt/01.html>
- [Salt75] Jerome H. Saltzer and Michael Schroeder, "The Protection of Information in Computer Systems," *Proceedings of the IEEE*, Volume 63, pages 1278-1308
- [Seif01] Kurt Sifried, *Best Practices Advisory*, September 2001, http://seifried.org/security/best-practices/ksbpa-001-ssl_imap_pop_linux.html

Appendix D. Email Security Tools and Applications

Email Content Filters

Tool	Capabilities	Web site	Linux/ Unix	Win32	Cost
eSafe Mail	Mail filtering and virus scanning	http://www.ealaddin.com/		✓	\$\$\$
Description	<i>Can be used to filter messages with Microsoft Exchange, Lotus Domino, and SMTP-based mail servers. Includes a virus scanner.</i>				
MailMarshal	Mail filtering	http://www.webmarshall.com/		✓	\$\$\$
Description	<i>Can work beyond the normal scanning of words and phrases to the interpretation of context (lexical scanning). Other functions include the blocking of spam and the addition of legal disclaimers. Compatible with Exchange and Lotus Notes. No virus scanner included.</i>				
Procmail	Mail content filtering	http://www.impsec.org/email-tools/procmail-security.html	✓		Free
Description	<i>Freeware mail content filter. Supports sendmail.</i>				
SecureIQ Suite	Mail content filtering and virus scanning	http://www.group-software.com/	✓	✓	\$\$\$
Description	<i>A suite of server-based email security tools including rule-based content checking of images and text, comfortable and automatic encryption, personalized legal disclaimer and modular virus protection.</i>				
SuperScout	Mail and Web content filtering	http://www.surfcontrol.com/		✓	\$\$\$
Description	<i>Supports Microsoft Exchange, Lotus Domino/Notes and SMTP mail servers. No anti-virus software included.</i>				

\$\$\$=This product involves a fee.

File Integrity Checkers

Tool	Capabilities	Web site	Linux/ Unix	Win32	Cost
Aide	Unix and Linux	http://www.cs.tut.fi/~rammer/aide.html	✓		Free
Description	<i>Advanced Intrusion Detection Environment (AIDE) is a free replacement for Tripwire. It does file integrity checking and supports a number of Unix and Linux platforms.</i>				
LANGuard	Windows 2000/NT	http://www.gfi.com/languard/		✓	Free
Description	<i>LANGuard File Integrity Checker is a utility that provides intrusion detection by checking whether files have been changed, added or deleted on a Windows 2000/NT system.</i>				
Tripwire	Windows, Unix, Linux, and Routers	http://www.tripwiresecurity.com/	✓	✓	Free to \$\$\$
Description	<i>Tripwire monitors file changes, verifies integrity, and notifies the administrator of any violations of data on network hosts.</i>				

\$\$\$=This product involves a fee.

Log File Analysis Tools

Tool	Capabilities	Web site	Linux/ Unix	Win32	Cost
Analog	Automated log file analysis tool for send mail	http://anteater.drzoom.ch/	✓		Free
Description	<i>Automated log file analysis tool that will analyze sendmail log files.</i>				
MELIA	Log file analysis for Windows hosts running Exchange	http://www.pixel.com.au/products/melia/melia.htm		✓	\$\$\$
Description	<i>Microsoft Exchange Log File Analysis (MELIA) is a powerful automated log file analysis tool for Microsoft Exchange servers. Generates 30 different reports and can be used with Microsoft Access and SQL server.</i>				
NetTracker	Most Web servers, mail servers, and operating systems	http://www.sane.com/products/NetTracker/	✓	✓	\$\$\$
Description	<i>Automated Web server and mail server log file analysis tool.</i>				

\$\$\$=This product involves a fee.

Network Sniffers

Tool	Capabilities	Web site	Linux/ Unix	Win32	Cost
Dsniff	Unix sniffer	http://www.monkey.org/~dugsong/dsniff/	✓	✓	Free
Description	<i>Dsniff is a collection of tools for network auditing and penetration testing. Dsniff, filesnarf, mailsnarf, msgsnarf, urlsnarf, and webspay passively monitor a network for interesting data (passwords, email, files, etc.). Arpspoof, dnsspoof, and macof facilitate the interception of network traffic normally unavailable to an attacker (e.g., due to layer-2 switching). Sshmitm and webmitm implement active monkey-in-the-middle attacks against redirected SSH and HTTPS sessions by exploiting weak bindings in ad-hoc PKIs.</i>				
Ethereal	Unix/Windows sniffer with GUI	http://www.ethereal.com/	✓	✓	Free
Description	<i>Ethereal is a free network protocol analyzer for Unix and Windows. It allows users to examine data from a live network or from a capture file on disk. It can interactively browse the capture data, viewing summary and detail information for each packet. Ethereal has several powerful features, including a rich display filter language and an ability to view the reconstructed stream of a TCP session.</i>				
Sniffit	Unix sniffer	http://reptile.rug.ac.be/~coder/sniffit/sniffit.html http://www.symbolic.it/Prodotti/sniffit.html (Windows)	✓	✓	Free
Description	<i>Sniffit is a freeware general-purpose sniffer for various versions of Linux, Unix, and Windows.</i>				
Snort	Unix sniffer and IDS	http://www.snort.org	✓	✓	Free
Description	<i>Snort is a freeware lightweight IDS and general-purpose sniffer for various versions of Linux, Unix and Windows.</i>				
TCPDump	Unix sniffer	http://www-nrg.ee.lbl.gov/	✓		Free
Description	<i>TCPDump is a freeware general-purpose sniffer for various versions of Linux and Unix.</i>				

Scanning and Enumeration Tools

Tool	Capabilities	Web site	Linux/ Unix	Win32	Cost
DUMPSec	Windows enumeration tool	http://www.systemtools.com		✓	Free
Description	<i>DumpSec is a security auditing program for Microsoft Windows. It dumps the permissions (DACLS) and audit settings (SACLs) for the file system, registry, printers and shares in a concise, readable listbox format, so that holes in system security are readily apparent. DumpSec also dumps user, group, and replication information.</i>				
Firewalk	Firewall filter rule mapper	http://www.packetfactory.net/firewalk/	✓		Free
Description	<i>Firewalk is an application that employs traceroute-like techniques to analyze IP packet responses to determine gateway ACL filters and map networks. This allows Firewalk to determine the filter rules in place on packet-forwarding devices.</i>				
Nmap	Port scanner OS detection	http://www.insecure.org/nmap/	✓	✓	Free
Description	<i>Nmap ("Network Mapper") is an open source utility for network exploration or security auditing. It was designed to rapidly scan large networks, although it also works against single hosts. Nmap uses raw IP packets to determine what hosts are available on the network, what services (ports) they are offering, what operating system (and version) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics.</i>				
Solarwinds	Network enumeration	http://www.solarwinds.net/		✓	\$\$\$
Description	<i>Solarwinds is a collection of network and management and discovery tools.</i>				
SuperScan	Port scanner, OS detection, and banner enumeration	http://www.foundstone.com/		✓	Free
Description	<i>SuperScan GUI network mapper. It will rapidly scan large networks to determine what hosts are available on the network, what services, they are offering, the version of these services and the type and version of the operating system. It will also perform reverse DNS lookup.</i>				

\$\$\$=This product involves a fee.

Virus Scanning Applications

Tool	Capabilities	Web site	Linux/ Unix	Win32	Cost
F-Secure Anti-Virus	Server and host-based virus scanners	http://www.f-secure.com/	✓	✓	\$\$\$
Description	<i>Offers a variety of anti-virus software applications for firewalls, mail servers, client computers and handheld computers.</i>				
Interscan	Server and host-based virus scanners	http://www.datatel-systems.com/virus.htm	✓	✓	\$\$\$
Description	<i>Offers a variety of anti-virus software applications for firewalls, mail servers, and client computers.</i>				
McAfee Anti-Virus	Server and host-based virus scanners	http://www.nai.com/	✓	✓	\$\$\$
Description	<i>Offers a variety of anti-virus software applications for firewalls, mail servers, client computers and handheld computers.</i>				
Panda Antivirus	Server and host-based virus scanners	http://www.pandasoftware.com/	✓	✓	\$\$\$
Description	<i>Offers a variety of anti-virus software applications for firewalls, mail servers, and client computers.</i>				
Sophos Anti-Virus	Server and host-based virus scanners	http://www.altcomp.com/	✓	✓	\$\$\$
Description	<i>Offers a variety of anti-virus software applications for firewalls, mail servers, and client computers.</i>				
Norton AntiVirus	Server and host-based virus scanners	http://www.symantec.com/product/	✓	✓	\$\$\$
Description	<i>Offers a variety of anti-virus software applications for firewalls, mail servers, client computers and handheld computers.</i>				

\$\$\$=This product involves a fee.

Vulnerability Scanning Tools

Tool	Capabilities	Web site	Linux/ Unix	Win32	Cost
CyberCop Scanner	Vulnerability scanner	http://www.pgp.com/products/		✓	\$\$\$
Description	<i>CyberCop Scanner is a network-based vulnerability-scanning tool that identifies security holes on network hosts.</i>				
ISS Internet Scanner	Vulnerability scanner	http://www.iss.net/		✓	\$\$\$
Description	<i>ISS Internet Scanner is a network-based vulnerability-scanning tool that identifies security holes on network hosts.</i>				
Nessus	Vulnerability scanner	http://www.nessus.org/	✓	✓	Free
Description	<i>Nessus a freeware network-based vulnerability-scanning tool that identifies security holes on network hosts.</i>				
Retina	Vulnerability scanner	http://www.eeye.com		✓	\$\$\$
Description	<i>Retina a general-purpose network security scanner that identifies a large number of Web server vulnerabilities.</i>				
SAINT	Vulnerability scanner	http://www.wwdsi.com/saint/	✓		\$\$\$
Description	<i>SAINT is a network-based vulnerability-scanning tool that identifies security holes on network hosts.</i>				
SARA	Vulnerability scanner	http://www-arc.com/sara/	✓		Free
Description	<i>SARA is a freeware network-based vulnerability-scanning tool that identifies security holes on network hosts.</i>				

\$\$\$=This product involves a fee.

Mail Server Hardening Tools

Tool	Capabilities	Web site	Linux/ Unix	Win32	Cost
Bastille Hardening System	Hardens Linux	http://www.bastille-linux.org/	✓		Free
Description	<i>Attempts to "harden" or "tighten" the Linux operating system. It currently supports Red Hat and Mandrake systems. It attempts to provide the most secure, yet usable, system possible</i>				
IIS Lockdown Tool	Hardens IIS	http://www.microsoft.com/downloads/		✓	Free
Description	<i>Assists administrators in locking down IIS versions 4.0 and 5.0.</i>				
Microsoft Network Security Hotfix Checker	Windows 2000/NT	http://www.microsoft.com/downloads/		✓	Free
Description	<i>Allows administrators to assess the patch status for Windows NT 4.0 and Windows 2000 operating systems, as well as the status of hotfixes for IIS 4.0 and 5.0, SQL Server 7.0 and 2000, and Internet Explorer 5.01 and later.</i>				
Windows Update	Assists in the update of most versions of Windows	http://www.microsoft.com/downloads/		✓	Free
Description	<i>Allows administrators to scan their servers to find any updates that are available at that time from Microsoft and other participating vendors.</i>				

Appendix E. Securing Microsoft Exchange

DISCLAIMER

This Appendix is only a guide on recommended product security settings and is meant to complement not replace well-structured organizational policy or sound judgment. Moreover, its contents do not address site-specific issues, which typically must be taken into account when applying this material to actual system environments involving local operational and policy concerns. This appendix is intended to serve as an example of how the generic guidelines given in the body of the main document can be interpreted and applied to specific classes of products. NIST recommends that experienced technical personnel, familiar with the organizational operational and policy aims, review these settings, experiment with them on a suitably configured test system, and judge the suitability of their effect on an operational system before attempting deployment.

This section provides the basics on Securing Microsoft Exchange. Much of this information is based on information from the Microsoft Web site (<http://www.microsoft.com/exchange/>) and the national Security Agency (NSA) *Guide to the Secure Configuration and Administration of Microsoft Exchange* [NSA02]. Because new vulnerabilities are discovered on a regular basis for all mail server applications, it is recommended that mail server administrators consult one or more of the online resources provided in Appendix G in addition to the procedures documented here.

Exchange security is tightly coupled to the underlying Windows operating system. For example, the Exchange logon can be coupled to the Windows operating system logon so that a user does not have to log on separately to Exchange. File permissions, registry settings, password usage, user rights, and other issues associated with Windows security have a direct impact on Exchange security. Therefore, it is absolutely necessary to configure the underlying Windows operating system securely to have any security for Exchange. For more information on operating system security, see the following resources:

- **Windows NT** – *NSA Guide to Securing Microsoft Windows NT Networks* (<http://nsa1.www.conxion.com/winnt/guides/wnt-1.pdf>)
- **Windows 2000** – *NSA Guide to Securing Microsoft Windows 2000* (<http://nsa1.www.conxion.com/win2k/index.html>).

This section addresses both Exchanges 5.0 and 5.5; where differences exist, they are clearly noted.

E.1 Exchange Server Installation

E.1.1 Create the Windows Exchange Services Account

The Exchange mail server application will require an account on the host Windows operating system. This account is commonly referred to as the “Exchange services

account.” The Exchange Server’s access rights are as defined by that account using Windows access control mechanisms. For example, if the name of the account established for Exchange services is “Exchange_Primary,” the Exchange server will be able to access only files and directories for which it has been granted the appropriate access permissions.

The following are recommended when creating this account:

- Create a unique account as the Exchange services account. The Exchange services account has carte blanche to access and manipulate the various components that comprise an Exchange environment. Creating a unique account will insure that these rights are not shared with processes or individuals that do not need such access.
- Set a strong and complex password for this account.
- Use an unpredictable name for the account.
- Do not enter a description for the account.

It is important to create this account before installation, because the installation routine will ask the installer to enter the Exchange Services Account name and password.

E.1.2 Create Windows Exchange Administrator’s Group

To simplify the assignment of administrative rights to the Exchange Server, it is recommended that a separate Windows Exchange Administrators Group be established. It is strongly recommended that the Windows operating system administrator group not be used, because it is not necessary to have Windows operating system administrative rights for many Exchange administration functions.

Having a separate Exchange Administration Group, or Groups will preclude the need for Exchange administrators to log in unnecessarily as a Windows operating system administrator – something that should be avoided for security reasons.

Consider partitioning Exchange Administrative rights through the use of multiple Exchange Administrative groups.

E.1.3 Exchange Application Install

When installing the Exchange Server, the following guidelines are recommended in regard to file location and the installation service packs and hot fixes.

- Do not install the Exchange Server on the same partition as the operating system. If necessary to install the Exchange Server on the same partition as the operating system, simply create the destination directory before beginning and give the Exchange services account “FullControl.”
- The information store and directory service log files should be on a physical drive separate from the information stores and directory service themselves. These log files can serve as a record of all transactions made since the last backup. In the event of a loss of the drive holding the Information Store or directory service, having the logs on

a separate physical drive will help ensure an ability to restore all lost data. If the use of a separate physical drive is not feasible, using a separate partition will provide a level of protection. The location of these files can be changed through use of the Exchange optimizer program, which can be run as an option during the installation routine or can be executed separately after installation is complete.

- **[Exchange 5.0]** Install latest service pack, because some of the security-related settings detailed here cannot be set on the base installation of Exchange Server 5.0; instead they require the prior application of the service pack.
- **[Exchange 5.0]** At the time of this writing, Microsoft had released several security relevant patches or hot fixes for Exchange Server 5.0. It is recommended that the mail server administrator review the security bulletins at <http://www.microsoft.com/technet/security/current.asp> for the latest information. It is critical to install security related fixes as soon as possible.
- **[Exchange 5.5]** Install latest service pack, which offers a variety of bug and security fixes. The service pack is cumulative (in other words, SP4 contains all the fixes and features of SP1 through SP3).
- **[Exchange 5.5]** At the time of this writing, Microsoft had released several security relevant patches or hot fixes for Exchange Server 5.5. It is recommended that the mail server administrator review the security bulletins at <http://www.microsoft.com/technet/security/current.asp> for the latest information. It is critical to install security related fixes as soon as possible.

Very few items within the Exchange Server directory require general user access; however, access rights are liberally granted by default. To revoke unnecessary access permissions, the following permissions are recommended for the directories in which the Exchange Server is installed. It is also necessary to change the rights associated with the mapisvc.inf file, give the Exchange Administrator Group(s) permission to execute Exchange administrative and diagnostic programs from the Start menu, and tighten the default permissions on a registry key.

- Give the following accounts Full Control access to all directories, subdirectories, and files within the directories where the Exchange Server was installed:
 - CREATOR OWNER
 - Domain Admins
 - Exchange_Primary
 - SYSTEM
 - <All Exchange Administrator Groups>
- Make certain that no other accounts are given access. It is particularly important to make certain that the group “Everyone” is not allowed access.

- Modify the permissions associated with the file %SystemRoot%\SYSTEM32\mapisvc.inf to allow the “Authenticated Users” group Modify access.
- **[Exchange 5.0]** To share files from the sampapps\clients directory, add “Authenticated Users” with read access.

E.2 Administrative Permissions

In addition to the file and directory permissions established at the operating system level, Exchange introduces application-level permissions. The rights associated with a given user are a combination of rights established at the application level and the operating system level. For example, a Windows user account with Administrative rights to Windows does not necessarily have the appropriate permission within Exchange to administer the Exchange server. These rights must be granted expressly through the Exchange Administrator tool.

It is impossible for these guidelines, which are intended for general usage, to expressly detail the exact permissions that should be applied to each organization’s exchange server. Instead, this section will focus on some key concepts that should be kept in mind when assigning administrative privileges.

E.2.1 Exchange Administrator Account(s)

To simplify the assignment of administrative rights to the Exchange Server, it is recommended that a separate Windows Exchange Administrators Group be established. It is strongly recommended that the mail server administrator not use the Windows administrator group, because it is not necessary to have Windows administrative rights for many Exchange administration functions.

Having a separate Exchange Administration Group, or groups, offers several benefits. First, it will preclude the need for Exchange administrators to log in unnecessarily as a Windows administrator – something that should be avoided for security reasons. Second, it will allow an organization to better partition administrative rights and responsibilities. Third, having a dedicated Exchange administrator group(s) will simplify the process of managing administrative rights because adding a new Exchange administrator is as simple as making them part of the Exchange Administrator Group.

E.2.2 Understand Exchange Administrator Roles

The Exchange Administrator tool allows various degrees of administrative rights to be applied in fine detail to the various levels of the Exchange hierarchy. Microsoft Exchange has numerous predefined roles to assist in assigning administrative privileges. These predefined roles are identical in concept to the roles defined under Windows (such as giving “Read” access to a file, which is a package of rights that gives the user Read and Execute permission on the file).

These predefined roles are well defined in the Exchange Server help facility. A few of these roles are somewhat confusing and their misapplication could result in security concerns, most notably the “permissions admin” role and “admin” role. An individual with admin rights has the capability to perform day-to-day administration on an

Exchange server. They can add mailboxes and manipulate numerous Exchange settings. The permission admin right includes all these rights plus the ability, as the name implies, to change the permission rights on the various objects within the Administrator tool. Permission admin rights can be dangerous because a rogue administrator with those rights could give themselves “send as” rights to a mailbox and effectively be able to masquerade as another user.

E.2.3 Understanding Inheritance

Permissions can be set on every object with the Exchange Administrator tool – in large organizations with many users, the total number of objects could be significant. Fortunately, under Exchange, permissions are, mostly inherited from the parent container, which greatly simplifies the task of assigning permissions. It is important to understand how permissions are inherited with the Exchange Administrator tool to ensure that the permissions are set up properly.

Generally, the effective permissions granted a user on a directory object are the sum of two types of permissions:

- The permissions the user account has on that object
- The permissions the user account inherits from above. The account inherits only the permissions assigned to the same user account on object(s) above it in the hierarchy. The inheritance does not end at the immediate parent. It continues up the directory tree to the top level of the hierarchy.

The only exception to this general description of inheritance is the configuration container. Because of its critical role, the configuration container does not inherit permissions.

E.3 Exchange Core Component Administration

Figure E.1 illustrates the basic components of Microsoft Exchange Server 5.0/5.5. These components work in concert to process information from client software packages, to synchronize servers in multi-server environments, and to perform general Exchange housekeeping.

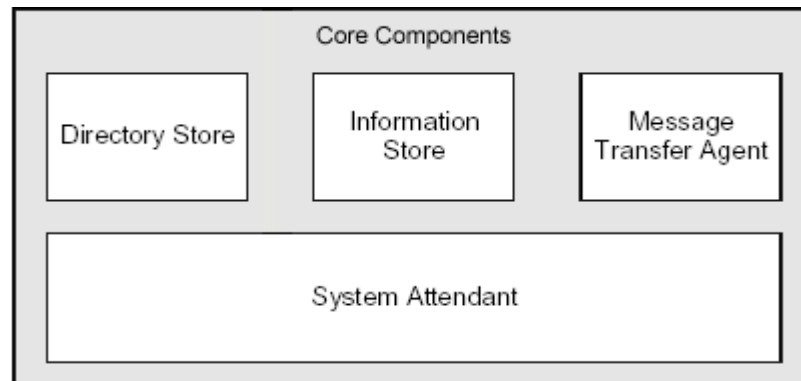


Figure E.1: Exchange Server Core Components

E.3.1 Directory Store

The Microsoft Exchange Server's Directory Store contains all the information about a site that is required to process data delivery. This includes addresses, distribution lists, details about public folders and mailboxes (but not the public folders and mailboxes themselves), and configuration information about the Exchange environment. The Directory Store provides a single, central location where administrators, users, and applications can look up and configure information about a variety of objects, such as user mailboxes. The directory also generates address books that contain information about users, such as email addresses and other related information.

The Directory Store is also responsible for enforcing security on all the directory objects, such as user mailboxes.

The Directory Store is managed at both the site and server level where, from a security perspective, two items are of interest: Lightweight Directory Access Protocol (LDAP) access and diagnostic logging.

Site Level

LDAP is a protocol that allows a client to query the Exchange directory for a variety of information related to the Exchange users. Directory store settings at the site level allows control over the information that is exported to LDAP clients under three scenarios:

- Anonymous requests
- Authenticated requests
- Inter-site replication.

It may be desirable, for example, to allow fully authenticated users (those who log on using a Windows account) full access to all user attributes as they browse the Exchange Directory Store. However, it may not be desirable to allow anonymous users, who by definition are not authenticated, to be able to access a complete listing of the users on the organization's Exchange network. It is recommended that careful consideration be given

to the information enabled for export via LDAP, particularly in relation to anonymous requests.

LDAP export settings are administered at the site level in the Exchange Administrator tool:

- Select the DS Site Configuration object under the Configuration object. Then select File/Properties and the “Attributes” tab. Consider carefully the attributes available for export via LDAP, particularly in relation to anonymous users.

Server Level

Diagnostic logging is a feature that is primarily intended to allow the administrator to log any of a plethora of events to aid in diagnosing system problems – it is recommended that a few of the events be logged as standard practice.

Directory Store diagnostic logging levels are administered from the server level in the Exchange Administrator tool:

- Select the appropriate server under the Servers object. Next select File/Properties. Then, select the “Diagnostic Logging” tab and highlight the MSExchangeDS entry. It is recommended that the following be logged at the “maximum” level:
 - LDAP Interface
 - Security.

Use the Windows event viewer to view logged events.

E.3.2 Information Store

The Information Store is responsible for maintaining and accessing messages in response to client requests. The Information Store consists of two components, a Private Information Store and a Public Information Store.

The private store is primarily for user mailboxes and consists of messages sent from user to user. The mailboxes are accessible by the mailbox owner and others for whom access has been allowed. The public store is used for newsgroups and other objects for which wide access is typically defined. Each store can hold almost out any kind of object, including mail, files, voice mail, and links to other files.

The Information Store is managed in the Exchange Administrator at the site and server levels where, from a security perspective, two items are of interest at both levels.

Site Level

At the site level, message tracking and top-level folder creation are of interest.

Enabling message tracking instructs the Exchange server to create a daily log file of all messages that are handled by the Information Store. That log can be used to track messages through the Exchange Server environment. This could be useful in various

security contexts. For example, if a user inadvertently got a Microsoft Word macro virus, message tracking could be used to determine just how far the infected document has spread.

To enable message tracking on the Information Store, from the Exchange administrator:

- Select the Information Store Site Configuration object under the configuration object and then select File/Properties. Enabled message tracking from the “General” tab.

Public folders are created via clients, not the Exchange Administrator Tool. The top-level folder creation settings allow the administrator to control who has that right. Note that the default condition is that everyone can create public folders. Depending on the sensitivity of the data and the manner in which public folders are used, it may be desirable to curtail the rights of individuals to create public folders. (For example, public folders may be used to hold newsgroups that are available for access remotely via newsreaders.)

To control who can create public folders, from the Exchange Administrator tool:

- Select the Information Store Site Configuration object under the configuration object and then select File/Properties and the “Top Level Folder Creation” tab. Depending on the specific usage of public folders, it may be desirable to restrict this right.

Server Level

At the server level, the logons feature and diagnostic logging are of interest.

There are no specific security settings in relation to the logons feature. The feature provides an easy way to determine who is logged onto the Information Store at any given time.

To determine who is logged onto the Private Information Store via the Exchange Administrator tool:

- Select the Private Information Store object under the server object. Then select File/Properties and the “Logons” tab.

To determine who is logged onto the Public Information Store:

- Select the Public Information Store object under the server object. Then select File/Properties and the “Logons” tab.

The diagnostic logging feature of the Information Store is identical in function to that of the Directory Store, as described above. It is recommended that diagnostic logging be enabled for a number of events related to both the private and Public Information Stores.

To enable diagnostic logging for the Private Information Store, via the Exchange Administrator tool:

- Select the Private Information Store object under the server object. Then select File/Properties and the “Diagnostics Logging” tab. Highlight the

MSExchangeIS/Private object. It is recommended the mail server administrator log the following at the “maximum” level:

- Logons
- Access Control
- Send On Behalf Of
- Send As
- Download.

To enable diagnostic logging for the Public Information Store, from the Exchange Administrator tool:

- Select the Public Information Store object under the server object. Then select File/Properties and the “Diagnostics Logging” tab. Highlight the MSExchangeIS/Public object. It is recommended the mail server administrator log the following at the “maximum” level:
 - Logons
 - Access Control
 - Send On Behalf Of
 - Send As
 - Download.

Use the Windows event viewer to view logged events.

E.3.3 Message Transfer Agent

The Message Transfer Agent (MTA) routes messages between mail servers.

The MTA is used anytime a message needs to be sent to another mail server. The MTA is managed at both the site and server levels in the Exchange Administrator where, from a security perspective, two items are of interest: message tracking and diagnostic logging. Message tracking and diagnostic logging for the MTA are identical in concept to that of the Directory Store and Information Store.

Site Level

Message tracking is enabled at the site level in the Exchange Administrator:

- Select the MTA Site Configuration object from within the configuration object, and then select File/Properties. Message tracking is enabled from the “General Tab.”

Server Level

MTA diagnostic logging levels are administered from the server level in the Exchange Administrator:

- Select the MTA object from the appropriate server object, and then select File/Properties and the “Diagnostic Logging” tab. It is recommended that the mail server log the following at the “maximum” level:
 - Security
 - Configuration.

Use the Windows event viewer to view logged events.

E.4 Securely Configuring Exchange’s Internet Mail Service (SMTP)

This section discusses securely configuring the Exchange Internet Mail Service (IMS). From a security perspective six areas need to be addressed when configuring IMS:

- **Limiting message size** – controlling the maximum size of incoming and outgoing messages. From a security perspective, this is of interest given the restrictions that can be placed on the size of incoming messages. If an incoming message from another Simple Mail Transfer Protocol (SMTP) host exceeds the set limit, the IMS does not write any data beyond the set limit. This prevents large messages from filling up the disk space on the Exchange server, reducing the threat of denial of service attacks.
- **Message tracking** – enabling message tracking instructs Exchange to create a daily log file of all messages handled by the IMS. This can be useful in a variety of security contexts when it is desirable to understand the flow of a message through the Exchange environment.
- **Disabling auto-replies** – disabling auto-replies for messages received via the IMS. Users can set up out-of-office messages that are sent automatically on receipt of a message. In some cases, it is possible that the information a user might include in this message should not be shared outside the organization. This could create a problem if the out-of-office messages were sent in response to emails from the Internet. In addition, by default, the IMS includes the sender’s display name, in addition to the sender’s address, in outbound messages. This can be disabled as well.
- **Restricting user access** – Controlling which Exchange users can or cannot send outgoing messages through IMS. IMS can be set up to accept or reject messages from any sender listed in the Microsoft Exchange Server address book. This feature is of limited value. It is of utility only for restricting individuals who are logging into the Exchange Server natively. These settings do not apply to individuals who access the Exchange Server through SMTP.
- **Accepting/rejecting SMTP connections** – Controlling from which IP addresses incoming SMTP messages are accepted. If an Exchange server is not intended for universal access, the IMS can be set to accept or reject messages based on IP address. Exchange Server 5.5 offers some additional capabilities for controlling SMTP connections. With version 5.5, it is possible to:

- Limit SMTP connections to those that are authenticated, encrypted, or both. This restriction applies to both connections with other hosts and client connections. The method used by a client for logging into the SMTP service is controlled by the user at the client end. There are two options: authenticating via an account name and password entered by the user and transmitted as a base64 encoded message (which is not secure), and logging in using Secure Password Authentication (SPA).
- Restrict access to clients that are homed on the server. This option requires the user to have a mailbox on the Exchange Server before a connection will be allowed. This can be used to restrict SMTP connectivity to users served by the Exchange Server.
- Accept clients only if authentication account matches the submission address. This option precludes a user from masquerading as another when dropping off a message via SMTP. Given the issue with the lack of a robust mechanism to protect user passwords in transit, this feature is of limited value in countering a determined adversary.
- Exchange Server 5.5 supports the Secure Multi-Purpose Internet Mail Extension (S/MIME) for message confidentiality, and integrity (see Section 3.2). As part of the S/MIME standard, clients can add a signature to a message that is used by the recipient to verify the identity of the user. To preserve these signatures as messages pass through the IMS it is necessary to enable this option.

To configure the Internet Mail Service, first select the Internet Mail Service object under the connections object in the Exchange Administrator tool, select File/Properties, and then:

- Select the “General” tab to set a message size limit. It is recommended that a message size limit that is reasonable for the environment be set. Note that whatever limit is set applies to incoming and outgoing messages.
- Select the “Internet Mail” tab to set message tracking. It is recommended that message tracking be enabled.
- From the “Internet Mail” tab, click on “Interoperability” [**Exchange Server 5.0**] or “Advanced Options” [**Exchange Server 5.5**]. It is recommended that the following be disabled:
 - Out-of-office responses
 - Automatic replies
 - Display names
- Select the “Delivery Restrictions” tab. These settings are advertised to restrict which Exchange users can or cannot send outgoing messages through IMS; however, they have no effect on users accessing the Exchange Server through SMTP.

- **[Exchange Server 5.0]** Select the “Connections” tab; check “Accept or reject by host” and select “Specify Hosts.” Unless an Exchange server is intended for universal access, it is recommended to restrict access.
- **[Exchange Server 5.5]** Select the “Connections” tab. Unless an Exchange server is intended for universal access, it is recommended that access be restricted by use of one or more of the following:
 - Use the “Accept Connections” option to require authentication, encryption, or both on all SMTP connections (applies to both client connections and connections to other hosts).
 - Use the “Clients can only submit if homed on this server” option to require the user to have a mailbox on the Exchange Server before a connection will be allowed.
 - Use the “Clients can only submit if authentication accounts matches submission address” option to help preclude a user from masquerading as another when dropping off a message via SMTP.
- Select the “Internet Mail” tab. Enable (check) “Clients support S/MIME signatures” if S/MIME will be used.

E.5 Securely Configuring POP3

POP3 is a mail access protocol typically used to access mail via the Internet. It actually works in conjunction with the SMTP for message transfer. SMTP is used to send messages from a client, and POP3 is used to retrieve messages. Exchange Server supports the use of POP3, where the primary security concern relates to the manner in which user authentication is performed. There are four options:

- **Basic (clear text).** When this option is selected, passwords are passed in the clear. The potential security concerns are obvious.
- **Basic (clear text) with Secure Socket Layer (SSL)/Transport Layer Security (TLS).** This option is identical to the first option, except that SSL/TLS is used to encrypt the link between the client and the server. SSL/TLS encryption is enabled via the key manager that is accessible via the Internet Information Server (IIS) Internet Service Manager.
- **Windows NT Challenge/Response.** This option uses cryptographic processes to ensure that passwords are not sent in the clear. However, once the client is authenticated, all other communications are sent unencrypted.
- **Windows NT Challenge/Response with SSL/TLS.** Same as the above option except that all communications between the client and server are encrypted.

Although the Windows NT Challenge/Response with SSL/TLS is the most secure method for accessing an Exchange server via POP, it is supported only by Microsoft email clients (e.g., Outlook). This option is not useful for organizations that support or use email clients other than those produced by Microsoft. For organizations with a

heterogeneous client environment, the most secure option is basic authentication with SSL/TLS. Sending passwords in the clear should never be used for the mail server accessible from the Internet.

To set the allowed authentication mechanisms for POP3, from the Exchange Administrator:

- Select the Protocols container under the site Configuration container. Select POP3 (Mail) Site Defaults and File/Properties and the “Authentication” tab. Select the allowed authentication mechanisms appropriate for the Exchange installation remembering the risk associated with Basic (clear text) passwords.
- If using Basic (clear text) passwords, grant the Exchange services account the “bypass traverse checking” right on the computer where the Exchange Server is installed.

E.6 Securely Configuring IMAP

The Internet Message Access Protocol (IMAP) is a successor to POP3. Like POP3, it works in conjunction with the SMTP. SMTP is used by the client for message uploads, IMAP is used for downloads. The primary security consideration is the type of authentication that is required. As with POP, there are four authentication options:

- **Basic (clear text).** When this option is selected, passwords are passed in the clear. The potential security concerns are obvious.
- **Basic (clear text) with SSL/TLS.** This option is identical to the first option with the exception that SSL/TLS is used to encrypt the link between the client and the server. SSL/TLS encryption is enabled via the key manager that is accessible via IIS Internet Service Manager.
- **Windows NT Challenge/Response.** This option uses cryptographic processes to ensure that passwords are not sent in the clear. However once the client is authenticated, all other communications are sent unencrypted.
- **Windows NT Challenge/Response with SSL/TLS.** Same as the above option except that all communications between the client and server are encrypted.

Although the Windows NT Challenge/Response with SSL/TLS is the most secure method for access an Exchange server via IMAP, it is supported only by Microsoft email clients (e.g., Outlook). This option is not useful for organizations that support or use mail clients other than those produced by Microsoft. For organizations with a heterogeneous client environment, the most secure option is basic authentication with SSL/TLS. Sending passwords in the clear should never be used for mail server accessible from the Internet.

To set the allowed authentication mechanisms for IMAP, from the Exchange Administrator:

- Select the Protocols container under the site Configuration container. Select IMAP (Mail) Site Defaults and File/Properties and the “Authentication” tab. Select the

allowed authentication mechanisms appropriate for the Exchange installation remembering the risk associated with Basic (clear text) passwords.

- If using Basic (clear text) passwords, grant the Exchange services account the “bypass traverse checking” right on the computer where the Exchange Server is installed.

E.7 Securely Configuring LDAP

LDAP is used by clients to access information stored in a DS component of Microsoft Exchange (e.g., an organizational email address list). It allows the client to read, sort, and/or delete objects stored in the DS.

Two items are of interest in relation to LDAP. As with POP3 and IMAP, the authentication mechanism is important, as well as the choice to allow or disallow anonymous access. The concerns with the latter relate to the fact that with anonymous access there is the potential for information stored in the DS to be accessed by anyone.

As with POP and IMAP, four authentication options exist:

- **Basic (clear text).** When this option is selected, passwords are passed in the clear. The potential security concerns are obvious.
- **Basic (clear text) with SSL/TLS.** This option is identical to the first option with the exception that SSL/TLS is used to encrypt the link between the client and the server. SSL/TLS encryption is enabled via the key manager that is accessible via the IIS Internet Service Manager.
- **Windows NT Challenge/Response.** This option uses cryptographic processes to ensure that passwords are not sent in the clear. However, once the client is authenticated, all other communications are sent unencrypted.
- **Windows NT Challenge/Response with SSL/TLS.** Same as the above option except that all communications between the client and server are encrypted.

Although the Windows NT Challenge/Response with SSL/TLS is the most secure method for access an Exchange server via LDAP, it is supported only by Microsoft email clients (e.g., Outlook). This option is not useful for organizations that support or use mail clients other than those produced by Microsoft. For organizations with a heterogeneous client environment, the most secure option is basic authentication with SSL/TLS. Sending passwords in the clear should never be used for mail server accessible from the Internet.

To set the allowed authentication mechanisms for LDAP, from the Exchange Administrator:

- Select the Protocols container under the site Configuration container. Select LDAP Site Defaults and File/Properties and the “Authentication” tab. Select the allowed authentication mechanisms appropriate for the Exchange installation remembering the risk associated with Basic (clear text) passwords.

- If using Basic (clear text) passwords, grant the Exchange services account the “bypass traverse checking” right on the computer where the Exchange Server is installed.
- If it is necessary to use Basic (Clear Text) authentication or Basic (Clear Text) using SSL/TLS, the client must specify in the user profile the account name in the following format to establish a successful connection:

dc=[domain name], cn=[account name]

or

cn=[account name], cn=[domain name]

To control anonymous access:

- Select the Protocols container under the site Configuration container. Select LDAP (Directory) Site Defaults and File/Properties and the “Anonymous” tab. The decision to enable or disable anonymous access will vary depending on the installation.

Note: If users are allowed to access mailboxes or public folders via a Web browser, the LDAP must be enabled for anonymous access or Web access will not function.

E.8 Configuring Authenticated Mail Relay

Microsoft Exchange 5.5 Service Packs 2 and 3 provides a capability to combat spam in accordance with the recommendations made of RFC 2505. Administrators have an option of installing Service Pack 2 and a related relaying hotfix, or installing Service Pack 3 alone. The Exchange IMS provides two configuration options to control mail relaying and spam:

- Do not reroute incoming SMTP mail
- Reroute incoming SMTP mail.

If an administrator chooses the first configuration option, malicious unauthorized entities could use the Exchange server as a launching point for spamming another organization. This is referred to as unsolicited commercial email (UCE). With this configuration, it may also appear that the organization that controls the subverted Exchange server was the cause of the spam in the first place. The ideal solution is to configure Exchange to return the SMTP 550 error code that indicates relaying is prohibited. One solution to prevent relaying and spam is to select the second option; reroute incoming mail. In addition to doing this, the administrator must specify all Internet domains for which the Exchange IMS service handles incoming mail. Without doing this, the server has been completely opened up to relaying.

E.9 Securely Configuring Web Access

Exchange provides a capability for users to access their mailboxes and public folders via a Web browser using the Web-standard Hypertext Transfer Protocol (HTTP). Web access to Exchange is provided through an IIS Active Server Page (ASP). Exchange

Server 5.0 is only supported with IIS 3.0, and Exchange Server 5.5 is supported with IIS 3.0 or IIS 4.0. “Outlook Web Access” is the term Microsoft uses for the component that provides this connectivity.

A number of security-related considerations are associated with allowing HTTP access to an Exchange server. First, a hot fix is necessary for those installations that are using IIS 3.0. Second, for Web access to function, it is necessary to weaken certain security settings. Third, selection of the proper mechanism for authenticating access via the Web is vitally important. The final security issue relates to anonymous (unauthenticated) access that Exchange supports for access to public folders and the global address book. If an organization is to allow anonymous access, care should be taken to restrict anonymous users to only that data they are authorized to see. In general, enabling Web access is not risk free, and the risks should be considered very carefully before implementation.

E.9.1 Changes to the Default Operating System Settings

The following changes (which will weaken overall security) will be required to implement Exchange’s Web-based email option:

- On the Exchange Server, give Authenticated Users Modify access to all files and subdirectories to the directory or directories where the Exchange Server was installed.
- On client machines, give Authenticated Users Modify access to %SystemRoot%\System32\Mapisrv.inf.
- Ensure that Authenticated Users have a right to log on locally to the Exchange Server computer.

E.9.2 Authentication

Installation of Outlook Web Access automatically installs the ASP and creates the appropriate virtual directory within IIS to access it. Configuration of IIS is beyond the scope of this document; note: however, that authentication mechanisms for Web access are controlled by IIS, not the Exchange Administrator tool. There are three options:

- **Allow Anonymous Access.** This allows users to access public folders and the global access list via the Web without authentication.
- **Basic Authentication.** This option will result in the user being prompted for a user name and password as a means of authentication. It is not secure because the password is sent in the clear; however, combining this option with SSL/TLS encryption would provide protection.
- **Windows NT Challenge/Response.** This is the native Windows NT authentication mechanism. Like SSL/TLS encrypted basic authentication, this option provides a protected means of authentication. Only the Internet Explorer browser supports this option.

If IIS and Exchange Server are on the same computer, then it is possible to use any of the above authentication methods. If IIS and Exchange Server are on separate computers,

then it is not possible to use Windows NT LAN Manager Challenge/Response (NTLM) authentication; however, using SSL/TLS will ensure that all information passing between the client and the IIS host is encrypted.

Select the appropriate authentication mechanism for the environment. Note: that anonymous access provides no authentication, basic authentication results in passwords being sent in the clear unless combined with SSL/TLS, and Windows NT Challenge/Response works only with Internet Explorer. Windows NT Challenge/Response or Basic Authentication combined with SSL/TLS encryption are the recommended authentication mechanisms.

E.9.3 Anonymous Access

HTTP access can be controlled for all servers in a site at the site level in the Exchange Administrator tool or can be controlled individually on a mailbox-by-mailbox basis. To control HTTP access at the site level, go to the Protocols container under the site Configuration container. Select HTTP (Web) Site Settings and File/Properties and the “General” tab.

- To allow anonymous access to public folders, enable “Allow anonymous users to access the anonymous public folders.”
- To allow anonymous access to the global address list, enable “Allow anonymous users to browse the global address list.”

If anonymous access is allowed, Exchange Administrators can also control which public folders are enabled for anonymous access. To do so:

- Select the HTTP (Web) Site Settings object as described above. Select the “Folder Shortcuts” tabs. Use the “new” and “properties” buttons to enable or disable public folders for anonymous access. Note: that users can manipulate the access controls on folders they own from the client. Once again, it is important to consider who should be allowed to create public folders.
- To control access via HTTP on a mailbox-by-mailbox basis, select the Recipients container and then the mailbox. Select File/Properties and the “Protocols” tab. Highlight HTTP (Web) and click on “settings” to enable or disable HTTP for the mailbox. Note: that disabling HTTP access to a mailbox does not restrict that user from accessing public folders that allow anonymous access.

E.9.4 Data Confidentiality

Note that the advanced security features previously discussed will not function when accessing messages via HTTP. A user can neither decrypt messages in his/her inbox nor encrypt or sign messages being sent to others. SSL/TLS encryption can be used to provide protection for the messages in transit. SSL/TLS is set up via IIS settings that are covered in NIST 800-44, *Guidelines on Securing Public Web Servers*: <http://csrc.nist.gov/publications/nistpubs/>.

Appendix F. Securing Linux and Unix Mail Services

DISCLAIMER

This Appendix is only a guide on recommended product security settings and is meant to complement not replace well-structured organizational policy or sound judgment. Moreover, its contents do not address site-specific issues, which typically must be taken into account when applying this material to actual system environments involving local operational and policy concerns. This appendix is intended to serve as an example of how the generic guidelines given in the body of the main document can be interpreted and applied to specific classes of products. NIST recommends that experienced technical personnel, familiar with the organizational operational and policy aims, review these settings, experiment with them on a suitably configured test system, and judge the suitability of their effect on an operational system before attempting deployment.

F.1 Securing Sendmail

This section discusses the basics of securing sendmail. Much of this information is based on information from several sendmail Web sites (<http://www.sendmail.net>, <http://www.sendmail.org> and <http://www.sendmail.com>) and, in particular, sendmail.net's guideline on securing sendmail [Allmn]. Since new vulnerabilities are discovered on a regular basis for all mail server applications, it is recommend that mail server administrators consult one or more of the online resources provided in Appendix G in addition to the procedures documented here.

Sendmail is a powerful and popular email application that is available in both freeware and commercial versions. Sendmail, like many similar applications, has had its share of security concerns. However, the newer versions of sendmail combined with appropriate configuration can be made far more secure. The most important first step in securing sendmail is to ensure that the latest or most secure version is installed. The latest versions of sendmail are configured to be especially "paranoid." Unlike previous versions, sendmail 8.10 and later are configured to be fairly secure even with a default install.

Sendmail security also relies on the security of the underlying operating system. The underlying operating system should be upgraded and patched to the most secure level. File permissions, password usage, user rights, and other issues associated with the underlying operating system security also have a direct impact on sendmail security. Therefore, it is absolutely necessary to securely configure the underlying operating system in order to have any security for sendmail.

F.1.1 Sendmail File Permissions

File permissions are one of the most critical and overlooked aspects of securing sendmail. Files and directories often have more open permissions than is prudent from a security standpoint. This often occurs because default installs tend to emphasize performance and

usability over security. In addition, mail server administrators often ease permissions in order to make their jobs easier.

The most critical files and directories that are a concern to a sendmail administrator are those with root-impact information. These include the:

- Mail queue directory
- `/etc` directory and all subdirectories
- Sendmail configuration file (`sendmail.cf`).

To secure these files and directories, the following steps should be taken:

- The mail queue directory `/var/spool/mqueue` should be made mode 700 and owned by `RunAsUser`.
- The path to the mail queue also needs to be protected. Therefore, the following directories need to be made mode 755 and owned by root:
 - `/`
 - `/var`
 - `/var/spool`.
- The `/etc` directory (and all subdirectories) need to be made mode 755 owned by root.
- The `sendmail.cf` (which is located in `/etc` for older versions of sendmail and `/etc/mail` in newer versions) should be mode 644 and owned by `TrustedUser` (which is usually root). For even more security, consider mode 600.
- Sendmail maps (the sendmail databases) should all be stored in the `/etc/mail` directory for both security and administrative ease.
- The alias files should be owned by root and have permissions of 644, or more restrictive.
- Ensure that the default files access restrictions that are included with sendmail versions 8.10 and later are enabled. This includes the following restrictions on sendmail and sendmail users:
 - Unable to read files that are writable by group or other.
 - Do not read files in directories that are writable by group or other.
 - Do not read `.forward` files that are links.
 - Do not write files in directories that are writable by group or other.

F.1.2 Sendmail Configuration Settings

There are a number of sendmail configuration settings that should be checked and changed if necessary to ensure security. Some of these steps may be too restrictive for some installations but organizations should carefully consider the risks of not taking the suggested actions. Configure sendmail to:

- Ensure that no application or script that trusts all input is “aliased” (i.e., included in the aliases file).
- Ensure that the aliases file is only writable by root.
- Ensure that directory and directory path of the aliases file is protected.
- Consider using smrsh.
- Enable the SafeFile Environment option, which will chroot user mailboxes. Note: this does not protect system mailboxes. This option is configured in the `sendmail.mc` file. An example would be: `define(`confSAFE_FILE_ENV',
`/var/mailjail')dnl`
- Disable the SMTP VRFY and EXPN commands. These two commands, EXPN (expand) and VRFY (verify), are used somewhat like the finger command to provide information about users. The EXPN command can be used to expand a user’s address to show the complete path to where the account is maintained. The VRFY command can be used to verify that a user has an account on the specific host. These commands are available, on an interactive basis, after connecting to a system on Transport Control Protocol (TCP) port 25.
- Configure sendmail to operate in a chroot jail. Note: this procedure requires a high-level of skill and varies significantly among operating systems.
- For sendmail 8.11 and below, configure the sendmail logging level in the `sendmail.cf` file a value no lower than 9. This can be accomplished by editing the `sendmail.cf` file and changing the OL entry to read OL9 or higher.
- For sendmail 8.12 and above, configure the sendmail logging level in the `sendmail.cf` file a value no lower than 9. This can be accomplished by editing the `sendmail.cf` file and changing the O LogLevel entry to read O LogLevel=9 or higher.
- Critical-level sendmail messages will be logged to a system log file that is owned by root and has permissions of 644, or more restrictive.
- Configure sendmail to authenticate clients via an encrypted means (e.g., Secure Socket Layer [SSL], Simple Authentication Security Layer [SASL]).

F.1.3 Configuring Anti-Spam Features of Sendmail

Mail relaying is disabled by default on sendmail versions 8.9 and newer. For example, if an organization called alice.com has a mail server, sendmail, in the default configuration

will not accept messages from bob.com destined for christina.com. If an organization wishes to permit some level of relaying, they will need to reconfigure the default install of sendmail.⁴³

The simplest approach is to list the domains the organization is willing to relay in the file `/etc/mail/relay-domains`. Anything listed in this file will be accepted for relaying. Note: sendmail must be restarted after this file is modified in order for the changes to take effect.

For more precise tuning, several features have been added to control relaying:

- **FEATURE(relay_hosts_only).** Normally, domains are listed in `/etc/mail/relay-domains`; any hosts in those domains match. With this feature, each host in a domain must be listed.
- **FEATURE(relay_entire_domain).** Setting this feature allows relaying of all hosts within the organization's domain. For example, on the host `gateway.organization.com`, this feature allows mail to or from any host in the `organization.com` domain. More precisely, this relays any host listed in the `$=m` class. This is equivalent to listing the name of the domain in `/etc/mail/relay-domains` (see above).
- **FEATURE(access_db).** This enables the hash database `/etc/mail/access` to enable or disable access from individual domains (or hosts, if **FEATURE(relay_hosts_only)** is set). The database format is described below.
- **FEATURE(blacklist_recipients).** If set, this feature looks up recipients as well as senders in the access database. The database format is described below.
- **FEATURE(rbl).** On sendmail versions 8.11 and below enables rejection of mail based on the Realtime Blackhole List maintained at `mail-abuse.org` or other such lists.
- **FEATURE(dnsbl).** On sendmail versions 8.12 and above enables rejection of mail based on the Realtime Blackhole List maintained at `mail-abuse.org`.
- **FEATURE(accept_unqualified_senders).** Normally, sendmail will not accept mail from a sender without a domain attached – for example, `user` instead of `user@somedomain.com`. This feature allows such users, but in general it should not be used.
- **FEATURE(accept_unresolvable_domains).** Normally, sendmail will refuse to accept mail that has a return address with a domain that cannot be resolved using the regular host lookups (a technique commonly used by spammers). This feature permits acceptance of such addresses. Unresolvable domains can be selectively accepted using the access database. Under normal circumstances, this feature should not be implemented.

⁴³ Readers are strongly recommended to refer the sendmail `cf/README` file for specific configuration options available their version of sendmail.

- **FEATURE(relay_based_on_MX).** Setting this feature permits relaying for any domain that is directed to the mail server host.

Several other features are not recommended, unless the mail server is inside a firewall, because they make the system vulnerable to abuse by spammers:

- **FEATURE(relay_local_from).** This feature allows relaying if the message claims to originate at the organization's domain. Since forging this address is trivial, this is usually a bad idea.
- **FEATURE(loose_relay_check).** This turns off checking for explicit routing through the mail server host, such as target%C.ORG@ORGANIZATION.COM.
- **FEATURE(promiscuous_relay).** This setting will turn off all checking for relaying and will make it very likely that the organization's mail server will be exploited for spam.
- **FEATURE(delay_checks).** This setting allows checks of the defined rule set to be skipped if a sender has been authenticated using a "trusted" mechanism, i.e., one that is defined via TRUST_AUTH_MECH().

The access database (normally in /etc/mail/access) allows a mail administrator to administratively allow access to the mail server by individual domains. Each database entry consists of a domain name or network number as the key and an action as the value.

Keys can be a fully or partly qualified host or domain name such as host.subdomain.domain.com, subdomain.domain.com, or domain.com. The last two forms match any host or subdomain under the specified domain. (If FEATURE(relay_hosts_only) is set, only the first form works.) Keys can also be a network address or subnetwork, e.g., 205.199.2.250, 205.199.2, or 205.199. The latter two forms match any host in the indicated subnetwork. Lastly, keys can be user@host.domain to reject mail from a specific user

Values can be REJECT to refuse connections from this host, DISCARD to accept the message but silently discard it (the sender will think it has been accepted), OK to allow access (overriding other built-in checks), RELAY to allow access including relaying SMTP through the mail server, or an arbitrary message to reject the mail with the customized message.

For example, a database might contain:

To:cyberpromo.com	REJECT
Connect:subsidiary.com	RELAY
Connect:spam@spammer.com	550 Please do not spam us
128.32	RELAY

This configuration will reject all mail from any host in the cyberpromo.com domain, allow any relaying to or from any host in the subsidiary.com domain, reject mail from spam@spammer.com with a specific message, and allow relaying for the 128.32.*.* network.

Action	Description
OK	Accept mail even if other rules in the running rule set would reject it, for example, if the domain name is unresolvable
RELAY	Accept mail addressed to the indicated domain or received from the indicated domain for relaying through your SMTP server RELAY also serves as an implicit OK for the other checks
REJECT	Reject the sender/recipient with a general purpose message
DISCARD	Discard the message completely using the <code>discard</code> mailer. For sender addresses it indicates that you should discard anything received from the indicated domain. If it is used for recipients, it affects only the designated recipients, not the whole message
### any text	Where ### is an RFC 821 compliant error code and “any text” is a message to return for the command
ERROR:### any text	As above, but useful to mark error messages as such
ERROR:D.S.N:###	Where D.S.N is an RFC 1893 compliant error code and the rest as above.

Note that the access database is a map and just as with all maps, the database must be generated using `makemap`. For example:

```
makemap hash /etc/mail/access < /etc/mail/access
```

F.1.4 Configuring SASL SMTP Authentication

SMTP authentication (AUTH) (see RFC 2554, <ftp://ftp.isi.edu/in-notes/rfc2554.txt>) provides two significant benefits. It protects user identifications and passwords by encrypting the SMTP AUTH. It allows users to access a mail server from anywhere in the world; while still denying access to bulk mail abusers. It does this by using the Simple Authentication and Security Layer (SASL) (see RFC 2222, <ftp://ftp.isi.edu/in-notes/rfc2222.txt>). It can also be configured using Transport Layer Security (TLS) and/or Secure Socket Layer (SSL) (see Section F.1.5).

SASL defines two terms that are important in the context of SMTP authentication:

- **Authorization Identifier** (user identification)—The user identification is the item that SMTP AUTH uses to check whether operations are allowed (authorized).
- **Authentication Identifier** (password)—Is the identifier that is being used to authenticate the client. That is, the authentication credentials of the client contain the authentication identifier.

To configure sendmail to use SASL it is necessary to download, compile, and install an SASL application. This example uses Cyrus SASL (<ftp://ftp.andrew.cmu.edu/pub/cyrus-mail/>). Ensure that when compiling the program that the Cyrus SASL libraries are installed in a location that sendmail uses on the system by default. For maximum security, the libraries should be owned and only writable by root.

It will be necessary to create a `sasl` password file using `saslpasswd`. Sendmail requires `sasl` to be owned by root or the trusted user and not be readable by anyone else since the file contains sensitive data (shared secrets).

Once SASL is installed and configured, it is necessary to compile sendmail with the correct options. Generally, it requires the following options:

```
APPENDDEF(`confENVDEF',`-DSASL')
```

```
APPENDDEF(`conf_sendmail_LIBS', `-lsasl')
```

The following options may need to be set in the `site.config.m4` (or OS specific) file, which is generally located in `devtools/Site` directory:

```
APPENDDEF(`confLIBDIRS', `-L/PATH/TO/SASL/lib')
APPENDDEF(`confINCDIRS', `-I/PATH/TO/SASL/include')
```

Note: it is necessary to replace the “/PATH/TO/SASL” with the appropriate path on the system.

To test the install first use the following command:

```
sendmail -d0.1 -bv root | grep SASL
```

Ensure that SASL appears in the output. If it does not, it is necessary to check the configuration of sendmail and Cyrus. If SASL does appear in the output, start the sendmail daemon. Once the daemon is started, connect to send mail with a telnet client. The output from telnet should appear similar to this example:

```
% telnet localhost 25
Trying 127.0.0.1...
Connected to localhost
Escape character is '^]'.
220 local.sendmail.ORG ESMTP Sendmail 8.10.0/8.10.0; Thu, 9
Sep 1999 10:48:44 -0700 (PDT)
ehlo localhost
250-local.sendmail.ORG Hello localhost [127.0.0.1], pleased to
meet you
250-ENHANCEDSTATUSCODES
250-DSN
250-AUTH DIGEST-MD5 CRAM-MD5
250 HELP
quit
```

If the output does not look like this it is necessary to search for security related problems (unsafe files) in the log files. If this does not reveal any problems, it is necessary to increase the LogLevel to 13 or higher and try again.

There are some options in the `sendmail.cf` files, that may need to be changed from their default values depending on the needs of the organization:

- **AuthMechanisms (confAUTH_MECHANISMS)**—Defines a list of mechanisms that are offered for authentication. This list is intersected with the list of available (i.e., installed) mechanisms, and the result of the intersection is listed in the AUTH keyword value for the EHLO response. The default values are: GSSAPI KERBEROS_V4 DIGEST-MD5 CRAM-MD5.
- **C{TrustAuthMech} (TRUST_AUTH_MECH())**—Defines a list of mechanisms that are used to allow relaying.
- **DefaultAuthInfo (confDEF_AUTH_INFO)**—Specifies a file in which the authorization identity, the authentication identity, the secret, and the realm to be used

for authentication are stored. This file must be in a safe directory and unreadable by everyone except root (or TrustedUser). It is used when sendmail acts as a client to authenticate itself to a server. Examples of this include:

- admin
- MySecretPassword
- example.domain

All data is usually case sensitive and the entire line is used in each case (including any spaces).

An example of SMTP AUTH entries in the `sendmail.cf` files are provided below:

```
TRUST_AUTH_MECH(`GSSAPI DIGEST-MD5')dnl
define(`confAUTH_MECHANISMS', `GSSAPI DIGEST-MD5')dnl
define(`confDEF_AUTH_INFO', `/etc/mail/auth/auth-info')dnl
FEATURE(`no_default_msa')dnl turn off default entry for MSA
DAEMON_OPTIONS(`Port=587, Name=MSA, M=E')dnl
```

Appropriately configured, SMTP AUTH will now allow relaying for those senders who have successfully authenticated themselves. Per default, relaying is allowed for any user who authenticated via a trusted mechanism, i.e., one that is defined via:

```
TRUST_AUTH_MECH(`list of mechanisms')
```

Now that sendmail provides SMTP AUTH, it will be necessary to configure the user clients to employ it. To do this, see the appropriate vendor documentation.

F.1.5 Configuring SSL/TLS SMTP Authentication

This section documents how to compile and configure sendmail to support SMTP AUTH via SSL/TLS. The advantage of using SSL/TLS over SASL is that it is more secure and can be used to protect Post Office Protocol (POP) and Internet Mail Access Protocol (IMAP) (see Section F.2).

To enable SSL/TLS support it is necessary to compile sendmail with SSL/TLS support. Before starting the compile process, it is necessary to download and install some additional applications:

- **OpenSSL**—is a full-featured toolkit for implementing SSL v2/v3 and TLS v1 protocols (<http://www.openssl.org/>).
- **Sfio**—Is a library for managing I/O streams. It provides functionality similar to that of Stdio (<http://www.research.att.com/sw/tools/sfio/>).

Once these two applications have been installed, it is necessary to create or modify the sendmail `site.config.m4` so that sendmail knows to compile in support for SSL/TLS. This file generally resides in `devtools/Site/` within the sendmail source tree. Add the following lines (modifying the paths as appropriate):

```

dnl Stuff for TLS
define(`confSTDIO_TYPE', `portable')
APPENDDEF(`confENVDEF', `-DSFIO')
APPENDDEF(`confLIBS', `-lsfio')
APPENDDEF(`confINCDIRS',          `-I/usr/local/include      -
I/usr/local/include/sfio')
APPENDDEF(`confLIBDIRS',  `-L/usr/local/lib')
APPENDDEF(`conf_sendmail_ENVDEF',  `-DSTARTTLS')
APPENDDEF(`conf_sendmail_LIBS',  `-lssl -lcrypto')

```

Then a standard `./Build` and `./Build install` should provide a `sendmail` binary with `STARTTLS` support.

Once `sendmail` has been successfully compiled, it is necessary to edit the `m4` configuration file and regenerate the `sendmail.cf` (`sendmail` configuration file). The additional lines that need to be added are (change as appropriate):

```

define(`CERT_DIR', `MAIL_SETTINGS_DIR`certs')dnl
define(`confCACERT_PATH', `CERT_DIR')dnl
define(`confCACERT', `CERT_DIR/cacert.pem')dnl
define(`confSERVER_CERT', `CERT_DIR/cert.pem')dnl
define(`confSERVER_KEY', `CERT_DIR/key.pem')dnl
define(`confCLIENT_CERT', `CERT_DIR/cert.pem')dnl
define(`confCLIENT_KEY', `CERT_DIR/key.pem')dnl

```

This defines the location of the server certificate and key to `MAIL_SETTINGS_DIR/certs`, or probably `/etc/mail/certs` in most cases. Within that directory should be three files (instructions for generating them are in the next section):

- Certificate Authority (CA) certificate—“`cacert.pem`”
- X.509 certificate, signed by an CA—“`cert.pem`”
- X.509 private key—“`key.pem`.”

Although it is possible to get certificates from a third-party CA, for many organization's a local CA is sufficient. To create a local CA, it is necessary to execute the following commands and steps (on Linux):

```

mkdir CA

cd CA

mkdir certs crl newcerts private

echo "01" > serial

cp /dev/null index.txt

cp /usr/local/openssl/openssl.cnf.sample openssl.cnf

vi openssl.cnf    (set values)

```

```
openssl req -new -x509 -keyout private/cakey.pem -out
cacert.pem -days 365 -config openssl.cnf
```

Once the CA is functioning, it is necessary to create a new certificate:

```
cd CA (same directory created above)

openssl req -nodes -new -x509 -keyout newreq.pem -out
newreq.pem -days 365 -config openssl.cnf
```

Once a new certificate is generated, it is necessary to sign it with the CA installed above:

```
cd CA (same directory created above)

openssl x509 -x509toreq -in newreq.pem -signkey newreq.pem -
out tmp.pem

openssl ca -config openssl.cnf -policy policy_anything -out
newcert.pem -infiles tmp.pem

rm -f tmp.pem
```

Note: `newcert.pem` contains a signed certificate and `newreq.pem` contains an unsigned certificate and private key.

Be sure to enter the hostname of the mail server that users will be accessing in the Common Name (CN) field of the x.509 certificate (openssl suggests the operator's name when it prompts for the CN). Otherwise, most mail clients will warn the user that the server hostname does not match the name in the certificate. Edit `newreq.pem` and remove the unsigned certificate (leaving the private key). Copy the resulting `newreq.pem` to `/etc/mail/certs/key.pem` and copy `newcert.pem` to `/etc/mail/certs/cert.pem`. Set the permissions on `key.pem` to 400 and the owner to whoever sendmail runs as (RUN_AS_USER in the m4 or RunAsUser in the `sendmail.cf`; defaults to root but possibly something like mailnull on a DMZ mail server).

Once this is complete, restart sendmail and check syslog for any errors. Then telnet to port 25, issue an ehlo and make sure that STARTTLS is listed in the supported features.

F.2 Configuring POP and IMAP to use SSL/TLS

There are many POP and IMAP servers available for the Linux/Unix environment. To operate them in a secure manner, it is necessary to check appropriate resources (e.g. <http://icat.nist.gov>) for known vulnerabilities and, if necessary, download any available patches or updates (for more information see NIST Special Publication 800-40, *Procedures for Handling Security Patches* [<http://csrc.nist.gov/publications/>]). One weakness common to all POP and IMAP servers in their default configurations is that the connection between the mail client and mail server is unencrypted. This means that both the user identification and password as well as all data are more likely to be intercepted and compromised. The way to protect this data is to use SSL/TLS to encrypt the connection between the mail server and mail client. This example will deal with configuring POP and IMAP to use SSL/TLS on a Linux host. Other Unix

implementations should be similar. Much of this section was sourced from Kurt Seinfried, *Best Practices Advisory* [Seif01].

The first step is to configure xinetd, typically the files reside in `/etc/xinetd.d/`, and there will need to be either an “imap” or “imaps” file (check the `/etc/services` file with respect to port 993 and port 995) with the following:

```
service imaps
{
    socket_type          = stream
    wait                 = no
    user                 = root
    server               = /usr/sbin/stunnel
    server_args           = -l /usr/sbin/imapd -- imapd
    log_on_success        += DURATION USERID
    log_on_failure        += USERID
}
```

It is then necessary to restart xinetd. At this point, connections will not work as a certificate needs to be generated. The location of the certificate can vary, for example on Red Hat it is typically located at: `/usr/share/ssl/certs/stunnel.pem`.

If an attempt is made to use an IMAP or POP client to connect to the server prior to generating a certificate, there will be the entries similar to the following in the `/var/log/messages`:

```
Sep 21 22:11:15 vomet stunnel[14083]: SSL_read (socket):
Connection reset by peer (104)

Sep 21 22:11:15 vomet stunnel[14083]: Connection reset: 3418
bytes sent to SSL, 338 bytes sent to socket

Sep 21 22:11:15 vomet stunnel[14139]:
/usr/share/ssl/certs/stunnel.pem: No such file or directory
(2)

Sep 21 22:11:15 vomet stunnel[14140]:
/usr/share/ssl/certs/stunnel.pem: No such file or directory
(2)

Sep 21 22:11:15 vomet stunnel[14141]:
/usr/share/ssl/certs/stunnel.pem: No such file or directory (2)
```

Unfortunately, using a stock `openssl.cnf` or `stunnel.cnf` to create the certificate will not work for IMAP and POP. It is necessary to modify the configuration file, typically in one of the following locations:

```
/usr/share/ssl/openssl.cnf
/etc/stunnel.cnf
```

In the appropriate file, find the section called “`req_attributes`.” By default, this section should look something like the following:

```
[ req_attributes ]
```



```

challengePassword      = A challenge password
challengePassword_min  = 4
challengePassword_max  = 20

```

It is necessary to set the “challengePassword_min” to 0 so that it is possible to create a certificate without protection that can be opened as needed. Once the certificate has been created, it will be necessary to ensure that it is only readable by root and not writable by anyone other than root. Depending on how stunnel was compiled, it may or may not warn about improper permissions:

```

Sep 21 22:27:35 vomit stunnel[14236]: Wrong permissions on
/usr/share/ssl/certs/stunnel.pem

```

Assuming everything is configured correctly, the following entries should be in /var/log/messages:

```

Sep 21 22:12:17 vomit stunnel[14151]: stunnel 3.8 on i386-pc-
linux-gnu PTHREAD+LIBWRAP

Sep 21 22:12:17 vomit stunnel[14151]: imapd connected from
161.184.218.225:61879

```

In addition, the following entries should be in /var/log/secure:

```

Sep 21 22:12:17 vomit xinetd[2596]: START: imaps pid=14174
from=161.184.218.225

Sep 21 22:12:17 vomit xinetd[2596]: EXIT: imaps pid=14174
duration=5(sec)

```

If it is necessary to keep the certificate file (certificate.pem) in a different location and use the stunnel “-p” option to specify a different location, such as:

```

service imaps
{
    socket_type      = stream
    wait            = no
    user            = root
    server          = /usr/sbin/stunnel -p
/etc/imapd.pem
    server_args      = -l /usr/sbin/imapd -- imapd
    log_on_success   += DURATION USERID
    log_on_failure   += USERID
}

```

There are some client issues to consider when using SSL/TLS secured IMAP and POP. Older clients such as Outlook Express version 5.0 will accept any certificate for SSL/TLS wrapped IMAP or POP as long as the server name and date are correct. This makes a man in the middle attack very easy. It is strongly recommended to upgrade to Outlook Express 6.0, which partially solves this problem. Netscape Messenger supports SSL/TLS wrapped IMAP but not SSL/TLS wrapped POP.

F.3 Configuring IMP Web Based Email Server

A variety of different applications exist for Web-based email access via Unix hosts. This particular section is based on the popular Internet Message Program (IMP) Webmail Client⁴⁴; however, many of the practices and configuration settings will apply to other Web-based email applications. In general, enabling Web access is not risk free and the risks should be considered very carefully before implementation.

F.3.1 Installation Considerations

Installing IMP requires a number of additional applications. IMP requires a mail server to receive and send email. The mail server must support IMAP, as POP and other mail client access standards are not supported. IMP does not include any Web server functionality, so it requires a Web server to be installed and operating on the host running IMP. For more information on securing Web servers, see NIST Special Publication 800-44, *Guidelines on Securing Public Web Servers* (<http://csrc.nist.gov/publications/>). In general the Web server should be securely configured with the necessary security patches installed prior to the installation of IMP.

The Web server supporting IMP should be configured to support only SSL/TLS connections. This will provide a high level of protection to the user authentication process and protect the data being transmitted from IMP to the end user's Web browser. For maximum protection, consider the use of client authentication via x.509 certificates. This will provide a higher level of protection at the cost of some functionality.

IMP also requires Hypertext Preprocessor (PHP) to be installed. PHP is an application that allows for interactive or dynamic Web pages. For more information on PHP see NIST Special Publication 800-28, *Guidelines on Active Content and Mobile Code* (<http://csrc.nist.gov/publications/>). PHP has a number of known security vulnerabilities that should be corrected through the appropriate configuration and installation of the latest security patches. Ensure that when installing PHP that the PHPLIB directory is outside of the Web server's document root or configure the Web server to not serve the PHPLIB directory (e.g. on Apache use the configuration directive).

IMP also requires a database backend. This is used to support an address book functionality. A variety of databases can be used. Again, the database program that is used will need to be configured securely prior to the installation of IMP.

F.3.2 Configuring IMP

Once IMP has been installed, it is necessary to configure it in a secure manner. First check for any new updates or security patches at the IMP homepage (<http://www.horde.org/imp/download/>). Once all the necessary patches have been applied open the IMP configuration file via any text editor. The default IMP configuration file is:

```
(IMP Install Root)/config/defaults.php3
```

⁴⁴ For more information on the IMP Web mail client, see <http://www.horde.org/imp/>.

The following lines will need to be edited as appropriate:

- Line 3 specifies the email address where problem reports are sent:

```
$default->problem_email = 'sysop@mysite.com';
```

- Line 16 specifies the local hostname of the machine running IMP:

```
$default->localhost = 'webmail.mysite.com';
```

- Line 19 specifies IMAP server defaults:

```
$default->server = 'mail.mysite.com';
```

- Line 93 specifies the database password:

```
$default->db_password = '';mypassword;
```

IMP can also be configured via the Web. Due to security concerns, it is recommended that this functionality be disabled. The easiest way to disable this functionality is to:

```
chmod 000 setup.php3
```

Keep in mind that many of the security concerns that relate to accessing mail via Web are often on the client side (see Section 8).

F.4 Using Procmail Mail Filter Application

Procmail (<http://www.procmail.org/>) is a program that processes email messages looking for particular information in the headers or body of each message, and takes actions based on what it finds. This procmail rule set is specifically designed to “sanitize” email on the mail server, before users retrieve their messages. It is not intended for end users to install on their desktop systems for personal protection. The requirements for using procmail include:

- Email must be received by a system that runs or can run procmail (most Linux and Unix systems operating sendmail or a similar mail server).
- Generally, the mail server should be set up to use procmail as the local delivery agent. The following line (or similar) should be in the `/etc/sendmail.cf` file:

```
Mlocal, P=/usr/bin/procmail, F=lsDFMAw5:/|@qSPfhn9,  
S=10/30, R=20/40, A=procmail -Y -a $h -d $u
```

- In addition to changing the `sendmail.cf` file, it will be necessary to add the following lines (or similar) to the `sendmail.mc` file:
- `define('LOCAL_MAILER_PATH','/usr/bin/procmail')`
- `define('LOCAL_MAILER_ARGS','procmail -d $u')`

- Perl (<http://www.perl.org/>) may be required if procmail is configured to use a scripting capability.
- To scan attachments for Office macros, mimencode which is a part of the metamail package (<http://www.impsec.org/email-tools/>) and mktemp from OpenBSD (<ftp://ftp.openbsd.org/pub/OpenBSD/src/usr.bin/mktemp>) must be installed. For organizations that use Microsoft Office, this is necessary since most email borne malicious code are of the Office macro variety, (unless of course other measures protecting against malicious code have been taken [e.g., anti-virus programs]).

F.4.1 Implementing Procmail on a Sendmail Gateway

Implementing procmail on a sendmail gateway is an excellent way to implement procmail. It reduces the burden on the organizations mail server and allows the use of procmail to protect mail servers that do not normally support procmail (e.g., Microsoft Exchange). Much of the guidance in this section comes from John D. Harden's E-mail Sanitizer Web page (<http://www.impsec.org/email-tools/procmail-security.html>) [Hard02]:

- 1) Add the following delivery agent to the `sendmail.cf` configuration file:

```
Mlocal,      P=/usr/bin/procmail, F=lsDFMAw5:/|@qSPfhn9,
S=10/30, R=20/40,
      A=procmail -Y -a $h -d $u
```

- 2) Add “procmail” to class P with close to the top of the `sendmail.cf` configuration file:

```
CPprocmail
```

- 3) Add the following to rule set 0, between the “handle virtual users” and “short-circuit local delivery” entries:

```
# pipe through procmail for processing
R$*<@example.com>$*      $#procmail  $@/etc/procmail/filter.rc
$: $1<@example.com.procmail.>$2
R$*<@example.com.>$*      $#procmail  $@/etc/procmail/filter.rc
$: $1<@example.com.procmail.>$2
R$*<@$*.procmail.>$*      $1<@ $2.>$3
```

Note: vary the domain name and script name (`/etc/procmail/filter.rc`) for your needs.

If relaying mail for more than one domain is required, use the following entries instead:

```
R$*<@$=w>$*              $#procmail  $@/etc/procmail/filter.rc
$: $1<@ $2.procmail.>$3
R$*<@$=w.>$*              $#procmail  $@/etc/procmail/filter.rc
$: $1<@ $2.procmail.>$3
R$*<@$*.procmail.>$*      $1<@ $2.>$3
```

- 4) Here is a sample `filter.rc` file—add the appropriate local configuration settings before using it:

```
#####
##
#
# procmail rules to filter mail on a gateway
#

LOGFILE=/var/log/procmail.log
NL="
"
LOGABSTRACT=no

# Configuration settings go here...
# See the discussion of what to put in /etc/procmailrc at
#http://www.impsec.org/email-tools/sanitizer-
#configuration.html
# /etc/procmailrc is the "master procmail script" for local
# delivery, this file is the "master procmail script" for
# relay
# The settings in one DO NOT affect the other.

POISONED_EXECUTABLES=/etc/procmail/poisoned

# etc... - you NEED to put configuration settings here.
# NB: DO NOT enable RECIPIENT notification on a relay...

# run the sanitizer
INCLUDERC=/etc/procmail/local-rules.procmail
INCLUDERC=/etc/procmail/html-trap.procmail

# send the mail on to the next relay
:0
! -oi -f "$@"

#
#####
##
```

- 5) If the next hop is a Microsoft Exchange Server, ensure that it is configured so that it will accept mail addressed to its fully qualified domain name in addition to the simple domain name. For example, if the domain is "myorg.com" and the Exchange Server is running on the computer named "myexchange," configure it to accept mail addressed to "@myexchange.myorg.com" as well as just "@myorg.com."

Appendix G. Online Security Resources

Computer Crime/Incident Handling

Resource/Title	URL
CERT/CC, How the FBI Investigates Computer Crime	http://www.cert.org/tech_tips/FBI_investigates_crime.html
CERT/CC, Responding to Intrusions	http://www.cert.org/security-improvement/modules/m06.html
CERT/CC, Detecting Signs of Intrusion	http://www.cert.org/security-improvement/modules/m09.html
Computer Evidence Processing Steps	http://www.forensics-intl.com/evidguid.html
Federal Guidelines on Searching and Seizing Computers	http://www.usdoj.gov/criminal/cybercrime/searching.html
Federal Code Related to Cybercrime	http://www.usdoj.gov/criminal/cybercrime/fedcode.htm
NIST ITL Bulletin, September 1999: Securing Web Servers	http://csrc.nist.gov/publications/nistbul/09-99.pdf
NIST SP 800-3, Establishing a Computer Security Incident Response Capability	http://csrc.nist.gov/publications/nistpubs/800-3/800-3.pdf

Federal Government Security Resources

Resource/Title	URL
Defense Information System Agency (DISA) Security Checklist	http://iase.disa.mil/techguid/checklist.html
Federal Computer Incident Response Center (FedCIRC)	http://www.fedcirc.gov/
National Infrastructure Protection Center	http://www.nipc.gov/
National Information Assurance Partnership	http://www.niap.nist.gov/
National Security Agency Rainbow Series	http://www.radium.ncsc.mil/tpep/library/rainbow/index.html
National Security Agency Security Recommendation Guides	http://nsa1.www.conxion.com/
NIST Computer Security Resource Center	http://csrc.nist.gov/
NIST ICAT Vulnerability Metabase	http://icat.nist.gov/
Office of Management and Budget Circular No. A-130	http://www.whitehouse.gov/omb/circulars/a130/
U.S. Department of Energy Computer Incident Advisory Capability (CIAC)	http://www.ciac.org/ciac/

General Security Resources

Resource/Title	URL
CERIAS	http://www.cerias.purdue.edu/
Computer Emergency Response Team (CERT)	http://www.cert.org/
NIST ICAT Vulnerability Metabase	http://icat.nist.gov/
RISKS Forum	http://catless.ncl.ac.uk/Risks/
Security Administration, Networking, and Security (SANS) Institute	http://www.sans.org/
SANS Twenty Most Critical Internet Security Vulnerabilities	http://www.sans.org/top20.htm

Mail Encryption Resources

Resource/Title	URL
Securing Email Through Proxies: Smap and Stunnel	http://rr.sans.org/email/smap.php
Securing POP Mail on Windows Clients	http://sewpsc.sewp.nasa.gov/documents/pop.mail.pdf
Securing POP Mail on Windows Clients	http://csrc.nist.gov/fasp/FASPDocs/SecurPOPwSSH.htm

Miscellaneous Mail Server Security Resources

Resource/Title	URL
dominosecurity.org	http://www.dominosecurity.org/
Lotus Domino Security Page	http://www.lotus.com/home.nsf/welcome/securityzone
Netcraft	http://www.netcraft.com/
Netscape Security Page	http://home.netscape.com/security/

Spam Resources

Resource/Title	URL
Arbitrary Blackhole List	http://abl.v6net.org/
Coalition Against Unsolicited Commercial Email	http://www.cauce.org/
Distributed Server Boycott List	http://www.dsbl.org/
Not Just Another Bogus List	http://njabl.org/
Open Relay Database	http://www.ordb.org/
Spam.abuse.net	http://spam.abuse.net/
Spamhaus	http://www.spamhaus.org/
Spam Prevention Early Warning System	http://www.spews.org/

Appendix H. Print Resources

Aviel D. Rubin, *White-Hat Security Arsenal: Tackling the Threats*, Addison-Wesley Publication Corporation, 2001.

Bryan Costales and Eric Allman, *Sendmail*, O'Reilly & Associates, 1997.

Craig Hunt, *Linux Sendmail Administration*, Sybex, 2001.

Dave Wood and Mark Stone, *Programming Internet Email*, O'Reilly & Associates, 1999.

Geoff Mulligan, *Removing the Spam: Email Processing and Filtering*, Addison-Wesley Publication Corporation, 1999.

Kevin Johnson, *Internet Email Protocols: A Developer's Guide*, Addison-Wesley Publication Corporation, 2000.

Mike Daugherty, *Monitoring and Managing Microsoft Exchange 2000 Server*, Digital Press, 2000.

Paul Robichaux, *Managing Microsoft Exchange Server*, O'Reilly & Associates, 1999.

Thomas Rizzo, *Programming Microsoft Outlook and Microsoft Exchange*, Microsoft Press, 1999.

Appendix I. Web Server Security Checklist

Appendix J. Acronym List

3DES	Triple Data Encryption Standard
ACL	Access Control List
AES	Advanced Encryption Standard
ARP	Address Resolution Protocol
ARPA	Advanced Research Project Agency
ASCII	American Standard Code of Information Interchange
AVAPI	Anti-Virus Application Programming Interface
CA	Certificate Authority
CD-R	Compact Disc Recordable
CIDR	Classless InterDomain Routing
CIFS	Common Internet File System
CIO	Chief Information Officer
CGI	Computer Gateway Interface
CN	Common Name
COM	Component Object Model
CPU	Central Processing Unit
CRAM	Challenge-Response Authentication Mechanism
CRC	Cyclic Redundancy Check
DES	Data Encryption Standard
DMZ	Demilitarized Zone
DNS	Domain Name Service
DoD	Department of Defense
DoS	Denial of Service Attack
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard

ESMTP	Extended Simple Mail Transfer Protocol
FTP	File Transfer Protocol
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDS	Intrusion Detection System
IETF	Internet Engineering Task Force
IIS	Internet Information Server
IMAP	Internet Message Access Protocol
IP	Internet Protocol
ISP	Internet Service Provider
ISSO	Information Systems Security Officers
IT	Information Technology
ITL	Information Technology Laboratory
LAN	Local Area Network
LDA	Local Delivery Agent
LDAP	Lightweight Directory Access Protocol
MAC	Message Authentication Code
MD5	Message Digest 5
MIME	Multipurpose Internet Mail Extensions
MOSS	MIME Object Security Services
MTA	Mail Transport Agent
MUA	Mail User Agent
NetBIOS	Network Basic Input/Output System
NFS	Network File System

NIPC	National Infrastructure Protection Center
NIS	Network Information System
NIST	National Institute of Standards and Technology
NSA	National Security Agency
OMB	Office of Management and Budget
ORB	Open Relay Blacklist
PC	Personal Computer
PEM	Privacy Enhanced Mail
PGP	Pretty Good Privacy
PHP	Hypertext Preprocessor
PKCS	Public Key Cryptography Standard
PKI	Public Key Infrastructure
POP	Post Office Protocol
RADIUS	Remote Authentication Dial-In User Service
RAID	Redundant Array of Inexpensive Disks
RAM	Random Access Memory
RFC	Request for Comments
SHA-1	Secure Hash Algorithm-1
S/MIME	Secure Multipurpose Internet Mail Extensions
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOHO	Small Office Home Office
SSH	Secure Shell
SSL	Secure Socket Layer
TCP	Transport Control Protocol
TLS	Transport Layer

UCE	Unsolicited Commercial Email
UDP	User Datagram Protocol
US	United States
VBS	Visual Basic Script
VPN	Virtual Private Networks
WINS	Windows Internet Naming Service
WSC	Windows Script Component
WSF	Windows Script File
WSH	Windows Scripting Host
WWW	World Wide Web

Index

- 3DES, 17, 19, 21, 22
- access controls, 38, 42, 54, 57, 71, 77
- access controls lists. *See* ACLs
- access privileges, 41
- ACLs, 57
- Active Server Page. *See* ASP
- ActiveX, 44, 50, 70, 74, 75
- Address Resolution Protocol. *See* ARP.
- Advanced Encryption Standard. *See* AES
- Advanced Research Project Agency. *See* ARPA
- AES, 18, 19, 21, 22, 23
- American Standard Code of Information Interchange. *See* ASCII
- APOP, 13, 14
- application layer firewalls, 62
- application proxy, 44
- ARAPANET, 5
- ARP, 5, 67, 68
- ARP poisoning attacks, 67, 68
- ARPA, ES-5, 1
- attachments, 1, 6, 7, 43, 49, 50, 55, 72
- attackers, 33, 65, 82
- AVAPI, 46
- backup procedures, 79
- backups, 82
- Base64 encoding, 7
- binary files, 43
- CA, 22, 23, 9, 10
- CERT, 39, 53, 9
- Certificate Authority. *See* CA
- certification and accreditation, 3, 29
- Chief Information Officer. *See* CIO
- CIO, 26
- circle of trust, 22
- complete mediation, 32
- Component Object Model. *See* COM
- compromise recording, 33
- computer security, 28
- confidentiality, 3, 28, 38, 42, 73
- configuration control, 28
- content filtering applications, 50
- Content Management API, 47
- contingency planning, 3, 29
- continuity of operations, 3, 27, 29
- cryptographic hash, 13, 16, 19
- cyclic redundancy check. *See* CRC
- Data Encryption Standard. *See* DES
- default configuration, 4
- defense in depth, 33
- demilitarized zone. *See* DMZ
- Denial of service. *See* DoS
- deployment plan, 3, 24, 37
- DES, 17, 19, 21
- Digital Signature Algorithm. *See* DSA
- Digital Signature Standard. *See* DSS
- disaster recover planning, 3, 29
- DMZ, viii, 57, 58, 59, 66, 67, 11
- DNS, 6, 77, 87
- Domain Name Services. *See* DNS
- DoS, 1, 38, 42, 43, 49, 58, 59, 63, 64, 65, 67
- DSA, 18, 2
- DSS, 18, 19
- email content filtering, 50
- encryption, 18, 19
- encryption algorithm, 16, 17, 21, 22
- encryption key management, 22
- enhanced hello, 10
- ESMTP, viii, 11, 12
- Eudora, 9, 19, 69, 70, 75
- Exchange, vi, 1, 3, 7, 12, 15, 46, 73
- exploits, 33
- Extended SMTP. *See* ESMTP
- fail-safe, 32
- Federal Government, 17, 18, 78
- file integrity checker, 65
- filtering router, 62
- FIPS 112
- firewalls, 2, 45, 60
- firmware, 28, 36, 63
- general-purpose operating system, 4
- HELP command, 9
- honey pot, 65
- hotfixes, 35
- HTML, 7, 44, 70, 75
- human resources, 32
- Hypertext Transfer Protocol. *See* HTTP
- IDSs, 64-67
- IETF, 17, 20, 21
- IMAP, vi, viii, 12, 14, 15, 24, 41, 53, 54, 57, 61, 62, 67, 70, 71, 73
- Information System Security Principles, 32
- Information Systems Security Officers. *See* ISSO

Information Systems Security Program
 Managers. *See* ISSM
 integrity, 3, 27, 28, 38, 42, 65, 67, 73, 79
 Internet, iii, v, vi, viii, 1, 5, 6, 8, 14, 17, 20, 22,
 39, 49, 51, 53, 54, 56, 57, 59, 60, 61, 62,
 67, 69, 70, 72, 73, 77, 86, 88
 Internet Engineering Task Force. *See* IETF
 Internet Message Access Protocol. *See* IMAP
 intrusion detection systems. *See* IDSs
 IP filtering, 56
 ISSM, 27
 ISSO, 3, 27
 Java, 44, 70, 74, 75
 Java Virtual Machine. *See* JVM
 JavaScript, 44, 50, 70, 72, 75
 LDA, 6, 7, 12
 least common mechanism, 33
 least privilege, 33
 lexical analysis, 50, 55
 local delivery agent. *See* LDA
 log entries, 65
 logging, 38, 47, 61, 62, 63, 76, 77
 logging configuration, 76
 Lotus Notes, 1, 12, 15, 69
 mail gateways, 59
 mail relay, 44, 45, 49, 52, 55
 mail server administrator, 4, 34, 36, 38, 44, 63,
 66, 67, 68, 79, 81, 83
 mail server commands, 6
 mail servers, 1, 2, 4, 1, 2, 6, 11, 26, 27, 28, 34,
 36, 41, 43, 46, 49, 51, 52, 56, 60, 66, 73,
 76, 79, 80, 83
 mail transport agent. *See* MTA
 mail user agent. *See* MUA
 malicious code, 1, 2, 5, 1, 44, 45, 47, 49, 50,
 72, 75
 management controls, 31
 MD5, 15
 Message Digest 5. *See* MD5
 message preview, 69
 Microsoft Exchange. *See* Exchange
 MIME, v, viii, 7, 11, 16, 19, 20, 21, 22, 23, 64,
 72, 73, 74, 76
 MIME Object Security Services. *See* MOSS
 misconfiguration, 1
 MOSS, 16
 MTA, 6, 7, 8, 47
 MUA, vi, 9, 10, 44, 74
 Multipurpose Internet Mail Extensions. *See*
 MIME
 NetBIOS, 35, 3

Network Administrators, 27
 network analyzers, 50
 Network Basic Input/Output System. *See*
 NetBIOS
 Network File System. *See* NFS
 Network Information System. *See* NIS
 network infrastructure, 5, 57
 network switches, 66
 NFS, 35, 86
 NIS, 35, 77, 87
 NIST, iii, iv, 1, 2, 3, 16, 17, 21, 29, 31, 35, 39,
 44, 54, 60, 64, 65, 69, 72, 73, 81, 83
 NSA, 39, 1, 3
 Office of Management and Budget. *See* OMB
 OMB, 1, 30
 OMB Circular A-130, 1, 30
 one-time password, 36, 38
 open design, 33
 open relay blacklists. *See* ORBs
 OpenPGP, 17
 operating system, 2, 3, 4, 3, 11, 12, 25, 32, 34,
 35, 36, 37, 38, 40, 41, 42, 43, 54, 55, 56,
 62, 63, 64, 65, 66, 67, 71, 72, 75, 80, 82,
 83
 operational controls, 31
 ORBs, 52
 Outlook, 9, 19, 21, 69, 70, 74, 1, 13, 14, 15, 16,
 17
 password guessing, 37
 patches, 4, 35, 56, 84, 85
 patching, 29, 34
 PEM, 7, 16
 penetration testing, 39, 85
 PGP, v, viii, 16, 17, 18, 19, 21, 22, 23, 72, 73,
 74, 76
 PKCS, 20
 PKI, 2, 16, 23
 plug-ins, 19, 47, 73, 74
 POP, vi, viii, 12, 13, 14, 15, 24, 41, 53, 54, 57,
 62, 70, 71, 73
 Post Office Protocol. *See* POP
 Pretty Good Privacy. *See* PGP
 Privacy Enhanced Mail. *See* PEM
 psychological acceptability, 33
 public key, 16, 19
 public key cryptography, 16, 19
 Public Key Cryptography Standard. *See* PKCS
 public key infrastructure. *See* PKI
 RC2, 20, 21
 remote administration, 56
 remote users, 38, 52

- reviewing logs, 76
- RFC 1421, 7
- RFC 1425, 10
- RFC 1651, 10
- RFC 1869, 10
- RFC 2268, 20
- RFC 2311, 20
- RFC 2312, 20
- RFC 2313, 20
- RFC 2314, 20
- RFC 2315, 20
- RFC 2440, 17
- RFC 2554, 11
- RFC 2630, 20
- RFC 2631, 20
- RFC 2632, 20
- RFC 2633, 20
- RFC 2634, 20
- RFC 822, 5, 6, 7
- RFC 918, 13
- risk assessment, 29
- risk management, 29
- root, 32
- routers, 2
- RSA, 18, 19, 20
- S/MIME, 16, 20, 21, 22, 23, 72, 11, 12
- scripts, 4
- Secure Hash Algorithm. *See* SHA-1
- Secure Multipurpose Internet Mail Extensions.
 See S/MIME
- Secure Shell. *See* SSH
- Secure Socket Layer. *See* SSL
- security policy, 27, 28, 29, 84
- security requirements, 4, 27
- security testing, 27, 38, 83, 85
- security through obscurity, 33
- security training, 29
- sendmail, viii, 1, 4, 46, 52, 53
- Senior IT Management, 26
- sensitive information, 1
- sensitivity, 30
- separation of privilege, 33
- service packs, 35
- service-leg DMZ, 59
- SHA-1, 18, 19
- Simple Mail Transfer Protocol. *See* SMTP
- Simple Network Management Protocol. *See*
 SNMP
- simplicity, 32
- SMTP, vi, viii, 9, 10, 11, 12, 15, 24, 41, 44, 45,
 51, 52, 53, 54, 57, 61, 62, 67, 71
- SMTP commands, 8, 9
- SMTP extensions, 8, 10, 11
- sniffers, 38, 50
- SNMP, 35
- spam, 1, 50, 51, 52
- SSH, 38
- SSL, vi, 38, 53, 54, 55, 64, 71, 73, 74
- standardized configurations, 29
- stateful inspection firewalls, 61
- switches, 2, 67
- symmetric key, 19
- symmetric key cryptography, 16
- system security plan, 30, 31
- technical controls, 31
- telnet, 9, 10, 11
- TLS, vi, viii, 11, 38, 53, 54, 55, 64, 71, 73, 74
- Transport Layer Security. *See* TLS
- Triple Data Encryption Standard. *See* 3DES
- Trojans, 5
- UCE, 51, 52, 55
- unauthorized access, 1, 30, 31, 38
- unencrypted, 1, 4, 15, 66, 71, 72, 73
- Unix, vi, 1, 4, 12, 37, 39, 42, 43, 55, 62
- unsolicited commercial email. *See* UCE
- untrusted third-parties, 1, 59
- upgrades, 4, 84
- user authentication, 4, 36, 39, 40, 52, 70, 73
- vendors, 4
- Virtual Private Networks. *See* VPNs
- virus scanner, 44
- virus scanning, 5, 41, 45, 46, 47, 48, 49, 50, 57,
 63
- viruses, 5, 16, 44, 45, 46, 47, 49, 55
- Visual Basic Script, 44, 72
- VPNs, 38
- VERFY command, 78
- vulnerability scanner, 84, 85
- vulnerability scanners, 84
- vulnerability scanning, 39, 83, 85
- Web administrator, 56
- Web content, 61
- Web server, 4, 5, 32, 35, 56, 57, 58, 59, 84, 85
- Web-based access, 73
- Windows, 1, 3, 21, 35, 39, 42, 62, 71, 72, 75,
 77
- Windows Scripting Host, 71
- worms, 5, 44
- X.400, 2
- X.509, 20, 10

