# NAVAL POSTGRADUATE SCHOOL
# Monterey, California

# THESIS

## EXAMINATION OF THE INTERNET MESSAGE ACCESS PROTOCOL (IMAP) TO FACILITATE USER-FRIENDLY MULTILEVEL EMAIL MANAGEMENT

by

Theresa Everette

September 2000

| | |
|---|---|
| Thesis Advisor: | Cynthia E. Irvine |
| Second Reader: | David J.Shifflett |

Approved for public release; distribution is unlimited

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

| | | |
|---|---|---|
| **1. AGENCY USE ONLY** ***(Leave blank)*** | **2. REPORT DATE**<br>September 2000 | **3. REPORT TYPE AND DATES COVERED**<br>Master's Thesis |

| | |
|---|---|
| **4. TITLE AND SUBTITLE:** Examination of the Internet Message Access Protocol (IMAP) to Facilitate User-Friendly Multilevel Email Management | **5. FUNDING NUMBERS** |
| **6. AUTHOR(S)**<br>Everette, Theresa M. | |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**<br>Naval Postgraduate School<br>Monterey, CA 93943-5000 | **8. PERFORMING ORGANIZATION REPORT NUMBER** |
| **9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)** | **10. SPONSORING / MONITORING AGENCY REPORT NUMBER** |

**11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

| | |
|---|---|
| **12a. DISTRIBUTION / AVAILABILITY STATEMENT**<br>Approved for public release; distribution is unlimited. | **12b. DISTRIBUTION CODE** |

ABSTRACT (maximum 200 words)

Information systems within the Department of Defense (DoD) need trustworthy enforcement of critical security policies against sophisticated attackers. Data, such as email, is processed on these systems on a daily basis. Since this data may contain sensitive information, special handling is required to prevent unauthorized disclosure. For these reasons, a high assurance Multilevel secure (MLS) Local Area Network (LAN) was developed to control the sharing of information at different security levels.

A challenge in multilevel environments is to provide a usable and meaningful interface to users via the email clients. These email clients interact with the high assurance server running on the MLS LAN. The high assurance server returns information at security levels at or below those of the client. An email client is only able to write and manipulate mail at its level. Therefore, client systems should provide users with feedback regarding operations they are able to perform.

In this research, six criteria were established to examine email clients. These criteria evaluated messages displayed to users via the email clients. All of the email clients was able to satisfy at least one of the established criteria.

| | |
|---|---|
| **14. SUBJECT TERMS**<br>Multilevel Secure (MLS), Local Area Network (LAN) Discretionary Access Control (DAC) policy, Mandatory Access Control (MAC) policy, Commercial-Off-The-Shelf (COTS), Internet Access Message Protocol (IMAP), POP (Post Office Protocol) | **15. NUMBER OF PAGES** 86 |
| | **16. PRICE CODE** |

| | | | |
|---|---|---|---|
| **17. SECURITY CLASSIFICATION OF REPORT**<br>Unclassified | **18. SECURITY CLASSIFICATION OF THIS PAGE**<br>Unclassified | **19. SECURITY CLASSIFICATION OF ABSTRACT**<br>Unclassified | **20. LIMITATION OF ABSTRACT**<br>UL |

THIS PAGE INTENTIONALLY LEFT BLANK

# EXAMINATION OF THE INTERNET MESSAGE ACCESS PROTOCOL (IMAP) TO FACILITATE USER-FRIENDLY MULTILEVEL EMAIL MANAGEMENT

Theresa M. Everette
Lieutenant, United States Navy
B.S., Florida Agricultural & Mechanical University, 1991

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN COMPUTER SCIENCE**

from the

**NAVAL POSTGRADUATE SCHOOL**
**September 2000**

Author: Theresa M. Everette
Theresa M. Everette

Approved by: Cynthia E. Irvine
Cynthia E. Irvine, Thesis Advisor

David J.Shifflett, Second Reader

Neil C. Rowe
Dan Boger, Chairman
Computer Science Department

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

Information systems within the Department of Defense (DoD) need trustworthy enforcement of critical security policies against sophisticated attackers. Data, such as email, is processed on these systems on a daily basis. Since this data may contain sensitive information, special handling is required to prevent unauthorized disclosure. For these reasons, a high assurance Multilevel secure (MLS) Local Area Network (LAN) was developed to control the sharing of information at different security levels.

A challenge in multilevel environments is to provide a usable and meaningful interface to users via the email clients. These email clients interact with the high assurance server running on the MLS LAN. The high assurance server returns information at security levels at or below those of the client. An email client is only able to write and manipulate mail at its level. Therefore, client systems should provide users with feedback regarding operations they are able to perform.

In this research, six criteria were established to examine email clients. These criteria evaluated messages displayed to users via the email clients. All of the email clients was able to satisfy at least one of the established criteria.

## TABLE OF CONTENTS

# LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

Many Department of Defense (DoD) information systems contain critical information. A large amount of this information is considered sensitive and requires special handling. Therefore, both single-level and multilevel systems are required. Additionally, good user interfaces are needed so that these systems will be acceptable to users.

In the sections that follow, various aspects of single-level and multilevel systems will be discussed and the objectives of this research will be described.

## A. SINGLE-LEVEL AND MULTILEVEL SYSTEMS

A single-level system is one in which all users of the system are cleared for the highest level of information on that system. All information regardless of its original classification will be classified at the level of the system. For example, if a computer was classified as top secret, then all of the information contained on that system will also be classified as top secret.

In general, a collection of single-level systems is used to manage data classified at different sensitivity levels. In the case of networks, each level is allocated the Department of Defense are part of such networks. For example, systems used to read or write military messages operate as single-level systems. Systems used to transfer highly sensitive correspondence between DoD agencies are also single-level.

When single-level systems are employed, users are given accounts on as many networks as they are authorized to have access. Thus users are forced to logon at different workstations to access information, such as email, at different sensitivity levels. For example, if a user wanted to read both top secret and secret email, access to separate networks at these levels would be required.

Single-level systems are neither efficient nor cost effective. Several problems are associated with single-level systems. The first is the requirement to purchase redundant systems and networks for each sensitivity level. Another problem is the requirement for secure environments such as vaults, cipher-locked rooms and guarded rooms which must be used to house single-level systems containing highly sensitive information.

With the emergence of new joint environments and the needs of dynamic coalitions, single-level systems are inadequate to meet the requirements of the Department of Defense (DoD) and Intelligence Communities. Multilevel systems offer a viable solution for these communities.

A multilevel system allows users to logon to a single computer system and access their information at different sensitivity levels thus eliminating the need to purchase numerous machines or networks. The user can read information equal to or below his or her current session level. These types of systems enforce a mandatory security policy called a mandatory access control (MAC) policy. These systems support subjects and objects each of which is assigned a sensitivity level. A subject is an "active entity, generally in the form of a person, process, or device that causes information to flow among objects or changes the state of the system"[1]. An object is defined as a "passive entity that contains or receives information. Examples of objects are records, blocks, pages, files and directories"[1]. When addressing secrecy in mandatory policy enforcement systems, two rules ensure that a subject will not gain access to data or a file unless the subject has the proper sensitivity level. First, enforcement of a MAC policy does not allow a subject to write down from a higher sensitivity level to an object at a lower sensitivity level (i.e. top secret to secret). Second, these systems do not allow a subject at a lower sensitivity level to read from objects at higher sensitivity levels (i.e. secret to top secret). By enforcing these rules, systems that implement MAC policies are able to constrain the access to objects by subjects "even in the face of Trojan Horses and other malicious software"[1].

## B. PURPOSE OF THIS RESEARCH

When any type of software is written and released, the software must provide usability. Usability "is a combination of the following user-oriented characteristics: ease of learning, high speed of user task performance, low user error rate, subjective user satisfaction and user retention over time"[17]. The problem with multilevel secure systems is their inability to provide a "user-friendly" interface. In the MLS LAN, the multilevel secure base causes the IMAP server to return messages and error codes to clients in response to unauthorized access requests. The ability of a client to provide meaningful feedback to users when such error codes are returned will affect the perceived usability of the secure system.

The purpose of this thesis is to examine user interface issues for COTS software in a multilevel secure environment. Specifically, we will examine the interfaces provided by email clients such as Microsoft Outlook, Netscape Messenger, Lotus Notes, Pine, and Postal when used in conjunction with an Internet Message Access Protocol (IMAP) server executing on and constrained by a high assurance multilevel base.

In this research, the interfaces presented by each of the email clients operating in a multilevel context will be evaluated against six usability criteria.

### C. RESEARCH QUESTIONS

To determine whether email clients in a multilevel environment present users with a "user-friendly" interface, the following questions are addressed in this thesis.

1. Do any of the email clients properly capture the "read only" and "read/write" responses sent by the IMAP server to the email client and properly display the responses to the user?

2. Does the email client disregard the responses ("read only", "read/write") that are forwarded by the IMAP server?

3. Can clients be configured to receive the "read only" and "read/write" responses? If so, what is presented to the users by the clients when such responses are received?

4. If clients cannot be reconfigured to receive the "read only" and "read/write" responses, what possible changes to the source code of the email clients would be required to improve their usability?

## D. THESIS OVERVIEW

This chapter has given a brief overview of single-level and multilevel systems. Additionally, this chapter has stated that the problem with many MLS systems is their inability to provide a "user-friendly" interface. Chapter II gives an overview of security policies and secure system terminology. Chapter III describes the components of the MLS LAN. Chapter IV explains why IMAP was chosen over Post Office Protocol (POP) for the email server. Chapter V explains how well the email clients performed during the examination. Chapter VI gives the conclusions and the possible future work of this thesis.

# II. OVERVIEW OF SECURITY POLICIES AND SECURE SYSTEM TERMINOLOGY

## A. COMPUTER SECURITY

Computer security relates to how an organization protects its information and resources. Its objectives can be described in terms of confidentiality, integrity and availability. Confidentiality is the prevention of the unauthorized disclosure of information; integrity is the prevention of the unauthorized modification of information; and availability is the prevention of the unauthorized withholding of information resources [6].

Prior to the extensive use of networking, most classified and unclassified computer systems could only be accessed by users physically proximate to the systems. Securing computer assets was often limited to guards, vaults and combination/cipher locked rooms. When systems are networked, physical security alone is inadequate to protect computer systems and data contained therein. According to the National Research Council:

> We are at risk. America depends on computers. They control power delivery, communications, aviation, and financial services. They are used to store vital information, from medical records to business plans, to criminal records. Although we trust them, they are vulnerable to the effects of poor design, and insufficient quality control,

> to accident, and perhaps most alarmingly, to deliberate attack. The modern thief can steal more with a computer than with a gun. Tomorrow's terrorist may be able to do more damage with a keyboard than with a bomb. As you can see, computer systems are vital in our everyday lives, we must therefore, become more active in protecting this vital resource [5].

Therefore, in order for computer security to be effective, it must be usable so that it is embraced not only by top management but by everyone within an organization.

## B. SECURITY POLICY

A fundamental aspect of computer security is a well-formulated security policy. A security policy is defined in terms of protecting an identified resource from unauthorized use. This identified resource must be tangible or have some form that is tangible [8].

The main objective of any security policy is to prevent or protect the identified resource from active threats. These threats include unauthorized distribution of classified information and unauthorized dissemination of an organization-related information. The objective is only meaningful if the organization for which the policy exists either owns the resource or exercises control over the resource [8].

A security policy might not state specific requirements for anyone within an organization. Instead, it might state general requirements such as the handling of sensitive information and the types of clearances held by those with certain positions within an organization. For example, the security policy could state different managers' responsibilities with respect to the enforcement of the security policy.

### 1. Integrity & Confidentiality Policies

#### *a. Integrity*

Integrity is concerned with preventing the unauthorized modification of data. It is concerned with creation, deletion, writing, and changing the status of data. Integrity is utilized in both the commercial and military sectors. When a mandatory integrity policy is enforced, a user at a low integrity level would be prevented from writing or modifying high integrity level information. The write operation would be denied because the lower integrity information might contaminate higher integrity information.

In the commercial sector, integrity is used to control fraud and error. It is required by businesses that utilize data processing for accounting and management purposes. For example, an organization's individual customer orders could be considered low integrity, so all employees might be able to modify it. However, an inventory listing might be considered high integrity data, so only managers would be able to modify it.

#### ***b. Secrecy***

Secrecy is concerned with the unauthorized disclosure of sensitive information. It can also be referred to as confidentiality. It is the primary policy used by the military to "regulate the control of classified information within the government"[7].

The military policy is enforced by adding sensitivity labels to subjects and objects. In a computer, these are reflected as sensitivity levels and may provide additional granularity by assigning access categories; all of which determine the information the users will be allowed to access.

### **2. Mandatory and Discretionary Policies**

An access control policy details the rules that are necessary to enforce the security policy. The two types of access control polices are Mandatory Access Control (MAC) and Discretionary Access Control (DAC) policies.

### a. *Mandatory Access Control (MAC) Policies*

"A mandatory policy can provide protection against unauthorized modification of information as well as protection against unauthorized disclosure"[6]. It is a policy that constrains subjects' (users/owners) access to objects (information within a computer such as files, data, and databases). A MAC policy is enforced by regulating the flow of information between sensitivity or classification levels.

A MAC policy is usually in effect in organizations, such as the DoD, that utilize background checks for personnel clearances, hierarchical classifications and security clearances (e.g. top secret, secret [6]). Corporations also use MAC policies to reflect security policies associated with proprietary information.

(1) Bell and LaPadula (BLP) model. The Bell and LaPadula model is the most widely used security model for MAC policies. The Bell and LaPadula model is used by the DoD to implement its security policy [6]. It is based on attributes of subjects and objects within a system. These attributes can be defined as security levels (top secret, secret, confidential, and unclassified). The model is based on the notion of a secure state. When the

system is in a secure state, each subject has access only to those objects to which the subject is authorized. The BLP model has defined rules that allow computer systems to transition from one secure state to another. Access in the system is determined by means of the dominance relationship between objects and subjects. It "specifies read and write access between a subject and an object based upon the dominance relationship between the subject's label (or access class) and the object's label (or access class)". For example, if Bob has a top-secret clearance and wants to access file "foo" which is labeled secret, Bob's label dominates the label of file "foo", so Bob can access file "foo".

In the Bell and LaPadula security model, the MAC policy is expressed through two properties that must be maintained. The two properties are the *simple security property* and the *star property*. The *simple* security property states that no subject is authorized to read information above his/her sensitivity level. This means, if a person is currently logged on at secret, he/she cannot read top-secret information. Second, the *star* "*" *property* states that no subject is authorized to copy information from an object with a high sensitivity level to one of a low sensitivity level.

(2) Biba model. As with the Bell and LaPadula, the Biba model is also based on the attributes of the subjects and the objects. This model is concerned with modeling a system that enforces a mandatory integrity policy. Objects and subjects are both assigned integrity levels. It has two properties, the *simple property* and the *star "*" property*. The simple property states that no subject is authorized to read information below its integrity level. The star "*" property states that a subject cannot write information to an object with an integrity level above the integrity level of the subject.

A MAC policy has several advantages. The main advantage is that it separates broad integrity and confidentiality classes by affixing labels to subjects and objects. Therefore, "a MAC policy offers verifiable restriction on the flow of information"[6]. The second advantage of MAC is that it provides protection against malicious software such as Trojan horses.

### ***b. Discretionary Access Control (DAC)Policies***

An owner-controlled DAC policy allows users to specify who will have access to their objects within a particular computer system. Therefore, users can grant and revoke privileges such as read, write and execute on the files and other data that they have created or own.

Systems enforcing a DAC policy are often easier to implement and less costly than systems enforcing a MAC policy. Problems arise however with this policy because of the revocation and propagation of privileges and the susceptibility of DAC systems to malicious software. With the revocation of privileges, when an owner grants privileges (read, write, execute) to other users and then wants to revoke access to the information, revocation can be a problem. The other users can copy the information and pass it on to additional users. At this point, the owner no longer controls who has access to the information. Therefore, the information may be accessible to users whom the original owner never intended access. A DAC policy is susceptible to malicious software such as Trojan Horses. A Trojan Horse is "software that appears to the user to be performing one function while it hides some other, often malicious function"[9]. These Trojan Horses could leak information to unauthorized users. For example, if only a DAC mechanism was in place and a Trojan Horse was present, data classified as secret could be copied to data classified as confidential. This would be in contradiction to the intended handling for the data.

### 3. Trusted Computing Base (TCB)

As a policy implementation, the Trusted Computing Base (TCB) contains all the protection mechanisms within a computer system. It defines the security perimeter of the system. It is defined by the security policy implemented by the organization. The TCB includes the identification and authentication (I&A) mechanism, access mediation mechanism, DAC functions and audit trail. "It contains databases that represent the security policy of an organization. These internal databases are used by the TCB to create the abstraction of subjects and objects which are entities outside of the TCB "[6].

## C. ASSURANCE OBJECTIVES

Assurance is the measure of confidence that a security policy is being enforced correctly. It determines how well the computer system meets the requirements set forth in the security policy. The assurance lifecycle process controls the requirements, design documentation, implementation, configuration management, distribution and maintenance of the system. Assurance can be classified from high to low.

High assurance systems have security built into them from the beginning of the lifecycle. These systems have undergone extensive examination and have been found to be highly secure. On the other hand, low assurance systems' security features are usually added rather than built-in from the beginning; thus making these systems less secure.

### 1. Reference Monitor Concept

"In order to achieve the desired execution control of users programs, the concept of a Reference Monitor is used. The function of the reference monitor is to validate all references (to programs, data, peripherals, etc) made by programs in execution against those authorized for the subjects (users, etc). The reference monitor not only is responsible to assure that the references are authorized to shared resource objects, but also to assure that the reference is the right kind (read, or read and write, etc.)"[16]. This concept is an abstraction that describes the necessary and sufficient functions required to enforce the policy by which subjects access objects. By supplying the necessary functions, the reference monitor must mediate all access to objects by subjects. In order to be sufficient, the reference monitor <u>must</u> meditate all access to objects by subjects and no additional access validation needs to be performed. This concept is necessary to describe how a subject causes information flow between objects thereby changing the state of the system. An abstract Reference Monitor is pictured below and contains the current access authorizations, reference validation mechanism, audit trail, and the authorization database. The authorization database reflects the security policy. The current access authorizations describe what object a subject has access to and the rights or modes (read, write, execute) the subject has on that object [19]. The reference validation mechanism ensures that no unauthorized access will occur and the audit trail is used to provide a history of the activity of the subjects within the system.

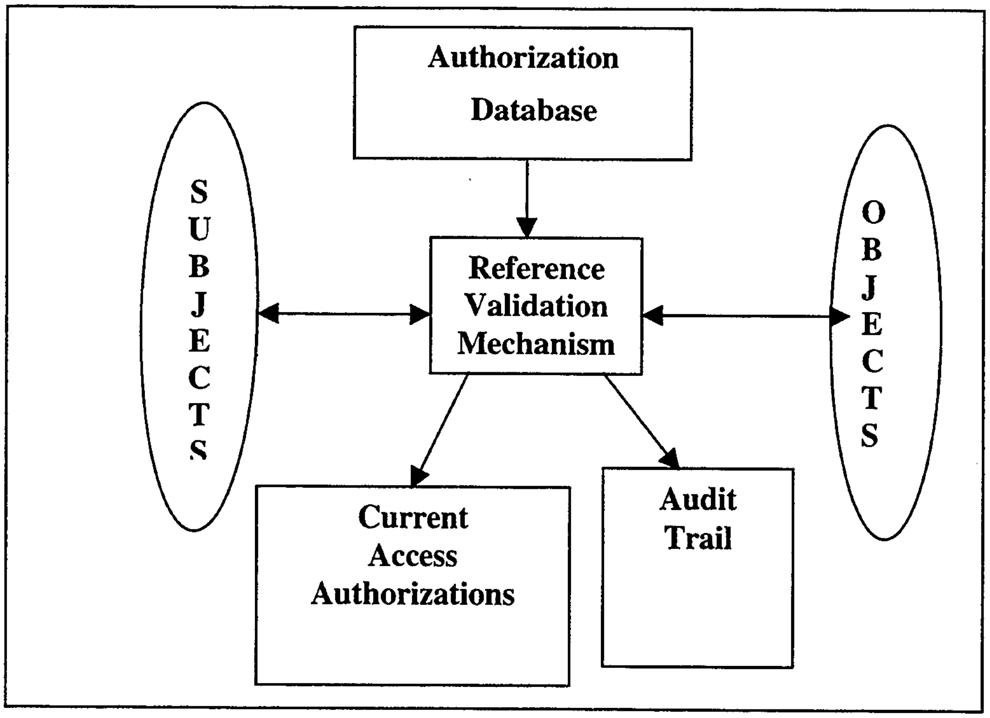

**Figure 1 Reference Validation Mechanism**

### 2. Security Kernel

A Security Kernel is a reference validation mechanism. As an implementation of the reference monitor, it attempts to achieve the following goals: it must be tamperproof, always invoked, and small enough to be analyzed.

By being tamperproof, the security kernel must separate its functions, from other operating system functions, thus aiding in the prevention of modification or tampering of its internal data. Because it is always invoked, the security kernel meditates every access to an object by a subject. For verifiability, it must be "small enough to be proven that it is correct" [6]. The security kernel includes hardware, software and firmware. A general-purpose security kernel provides the following properties and functions:

- creation of objects and subjects
- self-protecting
- enforces the security policy
- performs low-level resource management

### 3. Evaluation of Systems

Systems are evaluated to determine the amount of trustworthiness that can be placed in them. Evaluation of systems helps to determine that a system performs according to its design and specification. Criteria are used to provide comparable and consistent evaluations. Different criteria establish different levels of assurance for the correct enforcement of system security policy. The following sections describe two criteria for secure systems.

#### a. *Trusted Computer System Evaluation Criteria (TCSEC)*

The Trusted Computer System Evaluation Criteria (TCSEC) provides the basis for evaluating the effectiveness of security controls built into computer systems[18]. Different evaluation classes are used according to the amount of trust needed in the system. For instance, if everyone was cleared at the same level, very little trust is needed in the system; therefore very little assurance is needed. Organizations with users having different clearances require systems with greater trust to ensure users only access data they are authorized.

The TCSEC serves the following purposes: "1) to provide a standard to manufacturers as to what security features to build into their new and planned, commercial products in order to provide widely available systems that satisfy trust requirements (with particular emphasis on preventing the disclosure of data) for sensitive application, 2) to provide DoD components with a metric with which to evaluate the degree of trust that can be placed in computer systems for the secure processing of classified and other sensitive information, and 3) to provide a basis for specifying requirements in acquisition specifications "[9]. The criteria has four divisions (A, B, C, D) with A being the highest assurance and D being the lowest. Each division is subdivided into classes. Additional security features and assurance requirements are added as the classes advance in number and as the divisions are increased. Therefore, organizations gain more assurance of correct security policy enforcement in an A1 system relative to a C2 system.

### b. *Common Criteria for Information Technology for Security Evaluation*

The Common Criteria for Information Technology for Security Evaluation (CCITSE)[9] or Common Criteria is the replacement criteria for several earlier security evaluation criteria. It is an attempt at having one standard criteria for the United States, Canada and Europe. The CCITSE contains classes that are subdivided into families. The functional classes (components) are listed below:

**Table 1 Functional Classes of the CCITSE**

| Functional Class Name | Number of Family Members |
|---|---|
| Communication | 2 |
| Identification and Authorization | 10 |
| Privacy | 4 |
| Protection of the Trusted Security Feature | 14 |
| Resource Allocation | 3 |
| Security Audit | 10 |
| TOE Entry | 9 |
| Trusted Path | 3 |
| User Data Protection | 13 |

Security functional components are used to express a wide range of security functional requirements within protection profiles and security targets). Components are ordered sets of functional elements. These sets are grouped into families with common objectives (e.g. Security Audit Trail Protection) and classes with common intent (e.g. Audit) [1].

THIS PAGE INTENTIONALLY LEFT BLANK

# III. NPS MLS LAN OVERVIEW

## A. MLS LAN DESCRIPTION

Numerous problems are present in systems that do not enforce a mandatory security policy. These problems include redundant equipment for each sensitivity level, excessive clearances for personnel, incoherent views because data is spread across multiple systems, inconsistent content, and untimely information because data must be physically placed in each system [4]. In order to eliminate these inefficiencies, the NPS MLS LAN was built.

### 1. Overview of NPS MLS LAN and Architecture

The NPS MLS LAN is comprised of an XTS-300 which provides a Trusted Computing Base (TCB), personal computers (PCs) equipped with a Trusted Computing Base Extension (TCBE), and COTS software. Figure 2 shows the MLS LAN and all of its components. The MLS LAN components will be described in the next three subsections.

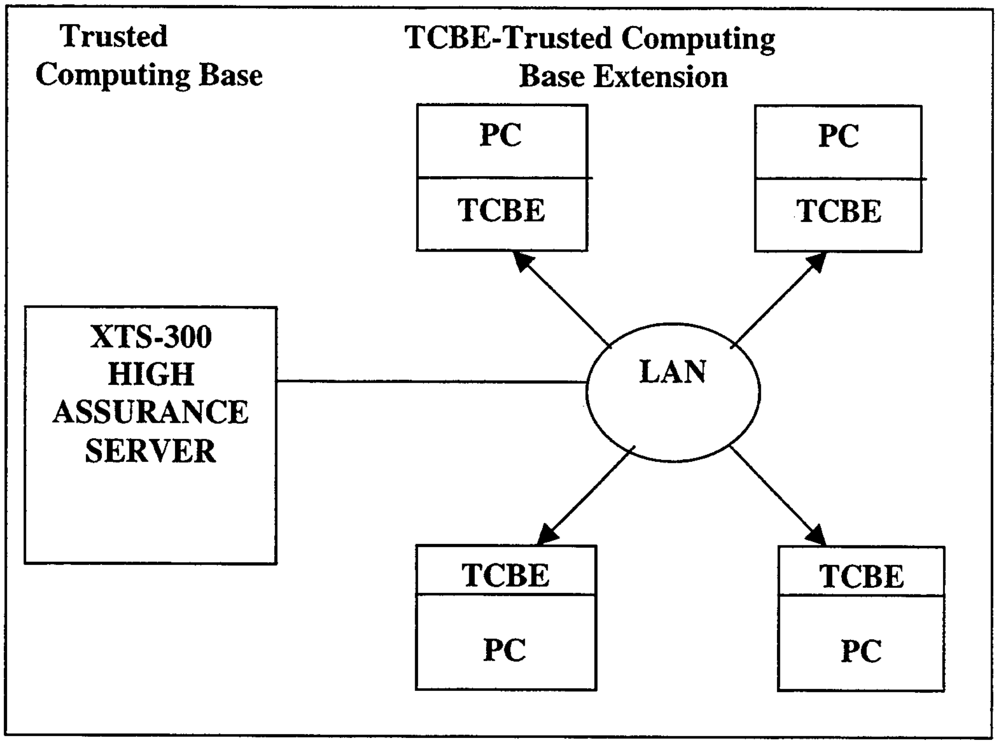

**Figure 2 MLS LAN Architecture**

### 2. High Assurance Server

The Wang Federal System XTS-300, a high assurance system, is the server platform for the MLS LAN. It is a Class B3 rated system under TCSEC. "The policy that the XTS-300 enforces is the DoD policy on multilevel security computing as formalized in the National Computer Security Center (NCSC) approved Bell and LaPadula model. It also enforces the integrity policy formulated by the Biba model" [10]. It also enforces a DAC policy.

The MAC policy is enforced by the XTS-300 by having labels on objects and subjects. The reference monitor implementation within the security kernel then mediates all access by subjects to objects.

The XTS-300 enforces both a confidentiality policy and an integrity policy as expressed by the Bell and LaPadula and Biba models respectively.

The XTS-300 STOP operating system supports a ring protection mechanism. "A ring is used to isolate portions of a process from tampering. Four rings are used to augment the security in the XTS-300"[10]. Ring 0 is the most privileged and contains the security kernel. Ring 1 provides networking, input/output (I/O), file system management, and DAC policy enforcement. Ring 2 contains the STOP trusted software, user-developed trusted code and the untrusted Commodity Application System Services (CASS). Ring 3 is reserved for untrusted applications. The XTS-300 negotiates all accesses to objects by subjects. This is how an organization's policy is enforced regardless of the application executing in Ring 3[11].

The XTS-300 provides a trusted path. The purpose of the trusted path is to provide a trusted communication link between the user and the XTS-300 and vice versa. Once a trusted path is established, the user can login, set levels, and perform other trusted functions. It also gives the user the ability to change his/her sensitivity level. To invoke a trusted path, the user presses a Secure Attention Key (SAK).

### 3. Client Workstation

The client workstations are COTS PCs equipped with a Trusted Computing Base Extension (TCBE). These PCs are networked to the XTS-300. "This architecture supports rapid upgrades of COTS software because the PCs are untrusted components of the MLS LAN. Therefore, a network administrator can simply upgrade the software on the PC and allow continued operation"[3]. A trusted component of the MLS LAN is the TCBE.

The TCBE is the critical component for creating a trusted path within the MLS LAN. "The TCBE negotiates a trusted path between the workstation and the server. The user initiates this trusted path across the network with the XTS-300 when he/she sends the Secure Attention Sequence (SAS)"[3]. Once the sequence is sent across the network, "the user is then able to use the trusted path to initiate a secure session on the XTS-300"[3]. When the user has a secure session, he/she can begin to securely view email or send other data across the network.

In addition to providing a trusted path, the TCBE also prevents the object reuse problem and is used for hardware identification and authentication (HW I &A). It uses public key and symmetric cryptography to encrypt communications between itself and the XTS-300.

### 4. Application Software Systems

Application software systems within the MLS LAN are untrusted client and server software. The MLS LAN implements its client-server architecture using the workstation model. This workstation model allows users to have a small workstation with modest computing power. Servers are the focus for data storage and manipulation. For example, the IMAP server performs all manipulation of the email. The manipulation is initiated when the client sends a command to the server and the server responds. Appendix A illustrates an interactive session between the IMAP server and Microsoft Outlook, a client email program.

THIS PAGE INTENTIONALLY LEFT BLANK

# IV. EMAIL PROTOCOLS

## A. OVERVIEW OF EMAIL PROTOCOLS

Once the components of the MLS LAN were selected, the next step was to select the appropriate Internet-based protocol to run on the system. The two protocols considered were POP and IMAP.

IMAP and POP are client/server systems. A client initiates commands and the server responds to those commands. The results of server processing are often returned to the client. These protocols are considered to be Message Retrieval Agents (MRA). An MRA is "a service that retrieves messages from a mailbox on a remote server to a Message User Agent (MUA)"[20] or client email program.

## B. COMPARISON OF THE POP AND IMAP

### 1. Overview of POP

POP is the most common protocol in use today. It can work in any environment, but it is not intended to manipulate email on the server. POP works best when a single workstation is being utilized to process the email.

POP functions primarily as an off-line email processing tool. With this protocol, a client is configured in advance to either leave email on the server or direct the server to delete email once it is downloaded. If the email is left on the server, the user cannot manipulate (read, reply to, delete, etc.) the email in any way. If email is downloaded to the workstation, the email is either deleted from the server or a copy is left on the server [2]. Once the email is downloaded from the server to a particular workstation, it is not accessible from a different workstation via the server.

With POP, a Transport Control Protocol (TCP) connection is made between the client workstation and the server. Once the connection is made, the user is then authenticated to the server. Following the authentication, the user is permitted to manipulate his/her mailbox. At that point, the client issues POP commands and the server responds to those commands.

### 2. Overview of IMAP

IMAP is a protocol that provides more functionality than POP. It allows the user to access email from more than one client. It permits the manipulation of mailboxes (remote message folders) as though the mailboxes were local. This gives the user the ability to access email from home, the office, or even while traveling; all without the need to transfer messages or files between these various locations. The overall functions and differences between IMAP and POP are described in the following paragraphs.

IMAP functions as an online, offline, and disconnected email processing tool while POP functions only as an offline tool. "In online mode, email is delivered to the server, and then the user manipulates the email messages from his/her workstation". Offline is when mail is delivered to a (usually shared) server, and a user at a workstation "periodically invokes a email client program that connects to the server and downloads all the pending email to the user's workstation. Thereafter, all mail processing is local to the client's workstation"[2]. In disconnected mode, a copy of the email is downloaded to the client workstation. Then the client disconnects from the server. Changes may be made to the email and when the client reconnects to the server, the changes are uploaded to the server. Additionally, IMAP functions in an interactive client server mode. This means that an IMAP client can ask the server for headers, for the bodies of specified messages, or to search for messages meeting certain requirements[2]. So, when the client invokes the IMAP server, it can manipulate the mail on the server. A copy of the email may be stored locally on the client, but the client is not considered to be the permanent email repository.

Like POP, a user must be authenticated to the server prior to accessing any email. From that point, the client issues the LIST command to get a list of available mailboxes. The user then selects the desired mailbox for manipulation. At that point, the user is allowed to view or manipulate the email within the selected mailbox.

### 3. Advantages of POP

When a user accesses his/her email remotely, he/she only needs to connect to the server to download his/her email to the workstation. POP also saves expensive disk space on the server by downloading the email to the user's workstation. This feature enables the client to save his/her email for an unspecified amount of time. This is in contrast to IMAP where the email is saved on the server, therefore, giving the administrator the ability to specify how long email will be kept.

POP has only thirteen commands within its command set while IMAP has twenty-two commands. This makes POP easier to implement and a much simpler protocol to learn. Additionally, because POP is so popular, more client software is available for it.

### 4. Advantages of IMAP

IMAP provides the ability to "store and fetch" messages. It supports concurrent access and updates on shared mailboxes. Clients are informed of changes by a change in the mailbox state. These changes are pushed from the server. This is useful because it gives all users the ability to view updates as they occur. Additionally, IMAP allows users to access their email from more than one computer. This gives users more flexibility as they travel and login remotely. Last, IMAP offers support for online, offline, and disconnected modes, which is used for accessing mailboxes remotely.

### 5. Advantage of POP in an MLS Environment

In a MLS environment, the one advantage of POP is that if offers a small connection time when email messages are being downloaded. This small connection time lowers the exposure of the communication traffic on the network.

### 6. Advantages of IMAP in an MLS Environment

There are several advantages to using IMAP in an MLS environment. The first advantage is that IMAP aids in helping to solve the object reuse problem because it keeps the email on the server. "Object reuse is defined as the reassignment to some subject of a medium (e.g., page frame, disk sector, magnetic tape) that contained one or more objects. To be securely reassigned, such media must contain no residual data from the previously contained objects"[1]. To ensure that no residual data remains on the client, all data will be purged between sessions. Clearly, mail cannot be stored on client workstation. Second, because IMAP stores the email on the server; the user can gain access to multiple mailboxes vice only one with POP.

Unlike POP, IMAP gives the users the ability to access their incoming and outgoing messages from different computers and at different times. Therefore, users can still access their incoming and stored email on the MLS LAN while on travel.

## C. OVERVIEW OF IMAP OPERATIONS

Operations within IMAP enable the creation, deletion and the renaming of mailboxes, the removal of messages, checks for new messages, searches for messages, selective fetches of messages according to attributes and text, setting and clearing of flags, and parsing (RFC-822 and MIME). IMAP is also compatible with Internet message standards and does not require the client software to have any knowledge of the server's file format. It contains flags, states and commands, all of which are described below.

IMAP uses six flags and four states. The six flags that indicate the status of message and are: seen, answered, flagged, recent, draft, and deleted. The four states are: the non-authenticated state, authenticated state, selected state and the logout state. "The non-authenticated state follows after the connection to the server is made. After the client is authenticated, the authenticated state is entered" [4]. Once a mailbox is selected, the IMAP server enters the Selected State. In this state, the user manipulates his/her email. Once the user disconnects from the server, the system enters the logout state.

## D. EMAIL CLIENTS

Email clients allow users to read, receive, write, forward, save, print, export, and delete email messages. Each email program offers GUIs (Graphical User Interfaces). These GUIs range from simple to complex depending upon the client. The clients support SMTP (Simple Mail Transport Protocol), MIME (Multipurpose Internet Mail Extensions), NNTP (Network News Transport Protocol), IMAP, and POP. The key features offered with each email client differentiate it from other clients. These differences will be discussed in the following paragraphs.

### 1. Netscape Messenger

Netscape Messenger is a part of the Netscape Communicator package. Messenger provides an HTML editor which supports bullets, table paragraph aligning, font size, font color, etc.; data encryption; digital signatures, organization/prioritization of email; and access to messages from multiple locations and computers [12]. It also allows users to import email from other email programs such as Microsoft Outlook, to set up email folders, and to create email filters on the email folders. For example, if a user receives unwanted mail from a specific source, he/she can create a filter that simply deletes such messages [13].

### 2. Pine

Pine was developed at the University of Washington and runs on both UNIX and PCs. "The guiding principles for Pine's user interface were: careful limitation of features, one-character mnemonic commands, always-present command menus, immediate user feedback, and high tolerance for user mistakes. It is intended to be learned by exploration rather than reading manuals"[11]. Pico, a message editor that comes with Pine, can also act as a stand-alone editor. It offers very limited formatting such as justification and a spell checker.

The key features of Pine are online help, a message index "that includes the status, sender, size, data and subject of [each] message"[11], a message composer, an address book, support for message attachments, Internet news and aggregate operations. Saving a selected set of messages at once is an aggregate operation. The primary advantage of Pine is that the source code is freely available [11].

### 3. Lotus Notes

Lotus Notes offers a welcome page, bookmark bar for quick links to web pages, Notes application and Internet sites, window tabs, search, and email setup wizards. It supports Java, JavaScript and X.509 certificates. Lotus Notes is supported on Windows 95, Windows 98, Windows NT 4.0 workstation, Mac PowerPC 7,6 and 8.5 [15].

### 4. Microsoft Outlook

Microsoft Outlook allows users to publish calendars as web pages, schedule group meetings quickly, communicate and collaborate with team members by publishing schedules on web sites, and manage contact information efficiently. Of the five email clients, Microsoft Outlook offers the most features.

The key features of Microsoft Outlook are inbox rules (email filters), storing of messages in multiple server folders, automatic dial-up of email accounts, offline storage of email messages (this feature depends on whether the email is stored on the server), digital signatures, and encryption. The availability of these features is dependent on the email server the client accesses [14].

### 5. Postal

Postal, like Pine, is freely available over the Internet. It only allows a user to read his/her email not manipulate it. Postal provides users with access to their email anywhere on the Internet. It is derived from JavaMail, a Java email package that is "useful for accessing a variety of message-based systems, particularly IMAP" [12]. Because it is written in Java, Postal is based on the notion of classes (moveMessageMenu, Folder Menu, and the DynamicMenu) and objects (session, folder, message, and server, etc.). "The JavaMail package can produce a number of events to inform a program of various changes in the state of the email database or the connection to the database. In Postal, the MessageCount events are used to indicate a change in the number of messages in the mail-folder. To listen for events, the MailWatcher object implements the MessageCountListener interface, which provides the messageAdded() and messageRemoved() callback methods. The main() method adds a MessageWatcher object as a listener for MessageCountEvents so that it will be notified whenever new messages are added to the INBOX folder. Unfortunately, the user will not receive a message count event just because he or she is listening, at least not with IMAP. He or she will have to interact with the email server in order to know that new email has arrived. To perform this interaction, the MailWatcher objects provide the watch() method that periodically queries the server for new messages. The frequency of the check can be tuned using the interval option; otherwise, the client will check every sixty (60) seconds "[12].

THIS PAGE INTENTIONALLY LEFT BLANK

# V. TESTING OF EMAIL CLIENTS

## A. PRELIMINARY REQUIREMENTS PRIOR TO TESTING

In order to utilize any of the email clients, the user must first invoke the trusted path between the server and the workstation, authenticate himself/herself with the server, and select both the integrity and secrecy levels for the session. Once the user has taken those steps, he/she can initiate one of the email programs (Postal, Messenger, Outlook, Pine, and Notes).

When the user starts an email program from the client, IMAP sends a response back to the client that shows all its capabilities. Examples of some the responses are included in Appendix A. Once the user selects to read his/her email, the IMAP server then responds to the client with a list of all accessible mailboxes. After getting the list from IMAP, the user is free to select only one mailbox at a time. If the client is allowed to modify the mailbox at the current level, then IMAP responds to the client with a "read/write" indicator. If the client is not allowed to modify the email at a particular level, then IMAP responds with a "read only" indicator. For example, if a user were to logon at secret and issue the "select" command for his secret mailbox, then IMAP would respond with "read/write".

If the user is still logged on at secret and issues the "select" command for the confidential mailbox, IMAP would respond with "read only". Since the MAC policy is being enforced by the XTS-300, the user is only allowed to manipulate data at his/her current sensitivity level.

IMAP responses displayed to the users are dependent upon the particular email client that is currently being utilized. The six flags associated with each email message response and the twenty-two commands that manipulate the mailboxes and messages are described below[21].

**FLAGS**

/Seen - the message has been read

/Answered - the message has been answered
(This may be a permanent flag)

/Flagged - the message is "flagged" for special attention

/Deleted - the message has been deleted and will be removed later by "Expunge" (This flag may be permanent)

/Draft - the message is still being composed

/Recent - this is the first session in which the message has been presented

# COMMANDS

- Select - allows the user to select a specific mailbox

- Examine - the same as the Select command, but this command is used for a read only mailbox

- Create - creates a mailbox with a given name that follows the command

- Delete - deletes a mailbox with a given name that follows the command

- Rename - renames a selected mailbox

- Subscribe - adds a specified mailbox to the server's list of active mailboxes

- Unsubscribe - removes a mailbox from the server's list of active mailboxes

- Lsub - returns a list of mailboxes that the user has declared as active using the Subscribe command

- Status - requests the status of the indicated mailbox without affecting the selected mailbox

- Append - places the literal argument as a new message at the end of the specified mailbox

- Check - performs housekeeping on a specific mailbox (e.g. resolving the server's in-memory state of the mailbox with the state on its disk)

- Close - permanently removes all messages with the \deleted flag set and returns to the authenticated state (no mailbox selected)

- Expunge - permanently removes all messages with the \Deleted flag set, but remains in the selected state

- Search - searches the mailbox for messages that match the specified search criteria

- Fetch - retrieves the requested data elements associated with the specified message(s)

- Noop - always succeeds

- Logout - closes the connection between the client and server

- Capability - returns a listing of the capabilities that the server supports such as the version of IMAP

- Store - used to update or change the flags associated with the specified messages

- Copy - copies the selected message to the destination mailbox

- UID - used to perform Copy, Fetch, and Search by the unique identifier instead of a sequence number. Therefore, UID takes one of those commands as an argument and performs the specified command according to the UID

- List - returns a list of all the mailboxes that the user can access

## B. TESTING CRITERIA FOR EMAIL CLIENTS

The testing criteria are based on the feedback users receive regarding disallowed operations. These operations range from attempting to delete a message to attempting to write a message from a high sensitivity level to a low sensitivity level. Regardless of the operation, the user should receive some type of "user-friendly" message on the screen. In other words, the user should be informed whether specific operations are legal or illegal. Additionally, the users should also be informed whether or not an error has occurred in the system. Therefore, the following criteria were used to check the user-friendliness of the clients.

- Ability to display the actual IMAP responses. When a client program interacts with the IMAP server, IMAP responds to the client program with a message. If the client program displays the actual IMAP response, then it passes this criterion.

- Ability to display error messages to the user when an illegal operation has been performed

- Ability to display a "user-friendly" message to the user

- Ability to handle group email. Group email is email shared by multiple users on the server. IMAP does not allow the email client programs to create group readable email. So, the owner of the group email must specify who can read the email. This is the DAC policy enforced. In order for that person to gain access, his or her sensitivity level must dominate that of the email. This is the MAC policy being enforced.

- Ability to display "read only" on the mailboxes. For single-level systems all email is at one level. In these systems, users can perform all operations on their email. In multilevel systems, a user's access class may strictly dominate the access class of the mail. In this case, users cannot perform all operations on the email; therefore a user should be informed that he or she can only read an email; not manipulate it. So, if a system can display this message, it greatly increases the friendliness of the system. For example, if the user wants to delete "read only" email, he or she would know that he or she must logon at the level of mail to perform the delete command.
- Ability to display "read/write" on the mailboxes. This type of message informs the user that he/she can perform all operations on his/her email. Therefore, the user can delete, read, and reply to all email messages that are read/write.

Additionally, if the email clients failed to meet any of the criteria, a determination as to the likelihood that the email client can be configured in some form or fashion to meet the criteria was determined.

### 1. Netscape Messenger

Netscape Messenger passed three of the six tests. It was not able to display "user friendly" messages to the user, or either display the "read only" or "read/write" indicators on the mailboxes. However, it was able to display the actual IMAP responses, error messages for illegal operations, and is capable of handling group email. Netscape Messenger searches for mailboxes designated for groups. If it finds one to which the user has access to, it presents that particular mailbox to the user.

### 2. Pine

Since both Pine and IMAP were developed at the University of Washington, the assumption was that it would have no problem interpreting the IMAP responses. Even though Pine does not have all the features of Microsoft Outlook and Netscape Messenger, it provided the most capabilities for the MLS LAN. The capabilities Pine provided are listed in the following paragraphs.

Pine was able to satisfy four of the six testing criteria. It presented the user with the "read only" indicator, displayed the actual IMAP responses, and was able to handle group email. Pine also displayed useful messages to the user whenever he or she tried to perform both illegal and legal operations. However, Pine failed to properly display the "read/write" indicator and to display a user-friendly message to the user. In the case of the "read/write" indicator, Pine displayed a mailbox that was not followed by a [READ/WRITE]. Apparently, there is a default in this email client program that specifies that if the mailbox is not "read only", display nothing. Pine, as with all the email clients, displays the actual IMAP responses. However, most users of the email clients probably would not understand those responses. Pine, like Netscape Messenger searches for mailboxes designated for groups. If it finds one that the user has access to, it presents that particular mailbox to the user.

### 3. Lotus Notes

Lotus Notes passed two of the six tests. It was not able to provide "user friendly" messages to the user, display either the "read only" or "read/write" indicators on the mailboxes, or handle group mail. However, Lotus Notes displayed the actual IMAP responses, and provided error messages to the user concerning illegal/legal operations.

### 4. Microsoft Outlook

Because Microsoft Outlook is so robust and full of features, it was considered the likely candidate to pass a majority of the tests. In fact, Outlook passed only two of the six criteria. The points of failure for this email client were its inability to display "user friendly" messages, display the "read only" and "read/write" indicators on the mailboxes, and handle group mail. This client was able to display the actual IMAP responses and display error messages to the user when legal and illegal operations were performed. It was also able to provide the user with messages when email was deleted or moved.

### 5. Postal

Postal lacks any ability to be evaluated under the set criteria because it only allows users to read their mail not manipulate it. However, when a user selects a mailbox, Postal sends the select command to the IMAP server. Postal was modified so if the IMAP server responded with the "read only" indicator, Postal would close the mailbox and issue the examine command to the IMAP server. By issuing this command, Postal is informing the IMAP server to open the mailbox in read only mode. Therefore, Postal has the capability to recognize the "read only" response from the IMAP server. Additionally, if the mailbox is open in "read only" mode, Postal won't issue any commands to modify the mailbox, such as changing the flags. This is because Postal caches the fact that the mailbox was open in "read only" mode. Unfortunately, all this interaction between Postal and the IMAP server is occurring without the knowledge of the user. However, Postal does provide error messages when the user tries to perform an invalid operation.

**Table 2 Results of Tests**

| | NETSCAPE MESSENGER | LOTES NOTES | PINE | POSTAL | MICROSOFT OUTLOOK |
|---|---|---|---|---|---|
| DISPLAYS IMAP COMMANDS | YES | YES | YES | NO | YES |
| DISPLAYS ERROR MESSAGES | YES | YES | YES | YES | YES |
| DISPLAYS USER FRIENDLY MESSAGES | NO | NO | NO | NO | NO |
| HANDLES GROUP MAIL | YES | NO | YES | NO | NO |
| DISPLAYS THE "READ/ WRITE" INDICATOR | NO | NO | NO | NO | NO |
| DISPLAYS THE "READ ONLY" INDICATOR | NO | NO | YES | NO | NO |

# VI. CONCLUSIONS AND FUTURE WORK

## A. FUTURE WORK

Additional work on the email client programs is required in order to improve the user interfaces. The ability of these email clients to present the user with more descriptive information increases the usability of the MLS system. Therefore, more features such as group mail or hierarchical classes could be added to the client programs such as Microsoft Outlook and Lotus Notes. In the cases where the email clients do not display the "read only" and "read/write" indicators, the code of the clients could be modified to display the response from the server.

## B. CONCLUSIONS

System development is usually a long and tedious process and may involve many iterations. The Human Computer Interface (HCI) should be included in the specification of any system. Companies should not waste countless man-hours and money building systems without first considering the users who will interact with these systems. If the users are considered from the beginning of the development of software and hardware, more systems might succeed.

An MLS environment helps to eliminate equipment duplication, excessive clearances, incoherent content and untimely information. However, eliminating these problems does not guarantee users will embrace the system. Therefore, the MLS environment should also provide a meaningful user interface. One of the ways to accomplish this task is to make email more "user friendly" in a multilevel environment.

In this thesis, five email clients were evaluated to see how well they would perform in a multilevel environment. The results vary according to the capabilities of each email program. Unfortunately, none of the email clients could be configured to display more "user-friendly" messages. In order for these email client programs to provide a better user interface, the developer would need to change the client code or make the source code freely so that others could make these modifications.

## APPENDIX A. IMAP SESSION

This is an IMAP session. "Send to Client" is the response from the IMAP server and "Send to Server" is the command from Microsoft Outlook.

```
Send to CLIENT  [* PREAUTH
holmes.astro.cs.nps.navy.mil IMAP4rev1 v12.256 server
ready]

Send to SERVER  [0000 CAPABILITY]

Send to CLIENT  [* CAPABILITY IMAP4 IMAP4REV1
NAMESPACE IDLE SCAN SORT MAILBOX-REFERRALS LOGIN-
REFERRALS AUTH=LOGIN THREAD=ORDEREDSUBJECT
0000 OK CAPABILITY completed]

Send to SERVER  [0001 IDLE]

Send to CLIENT  [+ Ready for argument]

Send to SERVER  [DONE]

Send to CLIENT  [0001 OK IDLE completed]

Send to SERVER  [0002 LIST "" "*"]

Send to CLIENT  [* LIST (\NoSelect) "/" unclass
* LIST (\NoInferiors \UnMarked) "/" unclass/xbox
* LIST (\NoSelect) "/" conf
* LIST (\NoInferiors \UnMarked) "/" conf/xbox
* LIST (\NoSelect) "/" secret
* LIST (\NoInferiors \UnMarked) "/" secret/xbox
* LIST (\NoSelect) "/" topsecret
* LIST (\NoInferiors \UnMarked) "/" topsecret/xbox
* LIST (\NoInferiors) NIL INBOX
0002 OK LIST completed]

Send to SERVER  [0003 LSUB "" "*"]

Send to CLIENT  [* LSUB () "/" unclass/xbox
* LSUB (\NoInferiors) NIL INBOX
* LSUB () "/" conf/xbox
* LSUB () "/" secret/xbox
* LSUB () "/" topsecret/xbox
* LSUB (\NoInferiors) NIL INBOX
0003 OK LSUB completed]

Send to SERVER  [0004 UNSUBSCRIBE "INBOX"]

Send to SERVER  [0005 LIST "" "INBOX*"]

Send to CLIENT  [0004 OK UNSUBSCRIBE completed
* LIST (\NoInferiors) NIL INBOX
0005 OK LIST completed]
Send to SERVER  [0006 LSUB "" "INBOX*"]

Send to CLIENT  [* LSUB (\NoInferiors) NIL INBOX
0006 OK LSUB completed]

Send to SERVER  [0007 IDLE]

Send to CLIENT  [+ Ready for argument]

LOAD: Child Process ID - 4290

Send to CLIENT  [* PREAUTH
holmes.astro.cs.nps.navy.mil IMAP4rev1 v12.256 server
ready]

Send to SERVER  [0008 CAPABILITY]

Send to CLIENT  [* CAPABILITY IMAP4 IMAP4REV1
NAMESPACE IDLE SCAN SORT MAILBOX-REFERRALS LOGIN-
REFERRALS AUTH=LOGIN THREAD=ORDEREDSUBJECT
0008 OK CAPABILITY completed]

Send to SERVER  [0009 IDLE]

Send to CLIENT  [+ Ready for argument]

Send to SERVER  [DONE]

Send to CLIENT  [0009 OK IDLE completed]

Send to SERVER  [000A SELECT "topsecret/xbox"]

Send to CLIENT  [* 3 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 968263650] UID validity status
* OK [UIDNEXT 4] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (\* \Answered \Flagged \Deleted
\Draft \Seen)] Permanent flags
* OK [UNSEEN 1] first unseen message in
/usr2/mail/shifflet/topsecret/xbox
000A OK [READ-WRITE] SELECT completed]

Send to SERVER  [000B IDLE]

Send to CLIENT  [+ Ready for argument]

Send to SERVER  [DONE]

Send to CLIENT  [000B OK IDLE completed]

Send to SERVER  [000C UID FETCH 1:*
(BODY.PEEK[HEADER.FIELDS (References X-Ref X-Priority
X-MSMail-Priority Newsgroups)] ENVELOPE RFC822.SIZE
UID FLAGS INTERNALDATE)]

Send to CLIENT  [* 1 FETCH (UID 1 BODY[HEADER.FIELDS
("REFERENCES" "X-REF" "X-PRIORITY" "X-MSMAIL-PRIORITY"
"NEWSGROUPS")] {44}
X-Priority: 3
X-MSMail-Priority: Normal
ENVELOPE ("Fri, 8 Sep 2000 10:39:38 -0700" "Test 4
from Outlook" (("dave" NIL "dave" "here")) (("dave"
NIL "dave" "here")) (("dave" NIL "dave" "here")) ((NIL
NIL "shifflet" "holmes")("Emma J Brown" NIL "ejbrown"
"holmes.astro.cs.nps.navy.mil")(NIL NIL "everette"
"holmes")) NIL NIL NIL
"<000701c019bb$c8180510$540a7883@astro.cs.nps.navy.mil
>") RFC822.SIZE 464 FLAGS () INTERNALDATE " 8-Sep-2000
17:42:00 +0000")
* 2 FETCH (UID 2 BODY[HEADER.FIELDS ("REFERENCES" "X-
REF" "X-PRIORITY" "X-MSMAIL-PRIORITY" "NEWSGROUPS")]
{2}
ENVELOPE ("Fri, 08 Sep 2000 10:40:50 -0700" "Test 4
from Netscape" (("kip" NIL "kip"
"henry.astro.cs.nps.navy.mil")) (("kip" NIL "kip"
"henry.astro.cs.nps.navy.mil")) (("kip" NIL "kip"
"henry.astro.cs.nps.navy.mil")) ((NIL NIL "shifflet"
"holmes")(NIL NIL "ejbrown" "holmes")(NIL NIL
"everette" "holmes")) NIL NIL NIL
"<39B924A2.D5608B1A@henry.astro.cs.nps.navy.mil>")
RFC822.SIZE 434 FLAGS () INTERNALDATE " 8-Sep-2000
17:43:00 +0000")

* 3 FETCH (UID 3 BODY[HEADER.FIELDS ("REFERENCES" "X-
REF" "X-PRIORITY" "X-MSMAIL-PRIORITY" "NEWSGROUPS")]
{2}

ENVELOPE ("Fri, 8 Sep 2000 10:43:44 -0700 (Pacific
Daylight Time)" "Test 4 from Pine" (("David Shifflett"
NIL "dave" "astro.cs.nps.navy.mil")) (("David
Shifflett" NIL "dave" "astro.cs.nps.navy.mil"))
(("David Shifflett" NIL "dave"
"astro.cs.nps.navy.mil")) ((NIL NIL "shifflet"
"holmes")(NIL NIL "ejbrown" "holmes")(NIL NIL
"everette" "holmes")) NIL NIL NIL
"<Pine.WNT.4.20.0009081043090.258-
100000@henry.astro.cs.nps.navy.mil>") RFC822.SIZE 417
FLAGS () INTERNALDATE " 8-Sep-2000 17:46:00 +0000")
000C OK UID FETCH completed]

Send to SERVER  [000D UID FETCH 1:3 (UID FLAGS)]

Send to CLIENT  [* 1 FETCH (UID 1 FLAGS ())
* 2 FETCH (UID 2 FLAGS ())
* 3 FETCH (UID 3 FLAGS ())
000D OK UID FETCH completed]

Send to SERVER  [000E IDLE]

Send to CLIENT  [+ Ready for argument]

Send to SERVER  [DONE]

Send to CLIENT  [000E OK IDLE completed]

Send to SERVER  [000F CLOSE]

Send to CLIENT  [000F OK CLOSE completed]

Send to SERVER  [000G IDLE]

Send to CLIENT  [+ Ready for argument]

Send to SERVER  [DONE]

Send to CLIENT  [000G OK IDLE completed]
Send to SERVER  [000H SELECT "secret/xbox"]

Send to CLIENT  [* 3 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 968263650] UID validity status
* OK [UIDNEXT 4] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS ()] Permanent flags
* OK [UNSEEN 1] first unseen message in
/usr2/mail/shifflet/secret/xbox
000H OK [READ-ONLY] SELECT completed]

Send to SERVER  [000I IDLE]

Send to CLIENT  [+ Ready for argument]

Send to SERVER  [DONE]

Send to CLIENT  [000I OK IDLE completed]

Send to SERVER  [000J UID FETCH 1:*
(BODY.PEEK[HEADER.FIELDS (References X-Ref X-Priority
X-MSMail-Priority Newsgroups)] ENVELOPE RFC822.SIZE
UID FLAGS INTERNALDATE)]

Send to CLIENT  [* 1 FETCH (UID 1 BODY[HEADER.FIELDS
("REFERENCES" "X-REF" "X-PRIORITY" "X-MSMAIL-PRIORITY"
"NEWSGROUPS")] {44}
X-Priority: 3
X-MSMail-Priority: Normal

ENVELOPE ("Fri, 8 Sep 2000 10:34:29 -0700" "Test 3
from Outlook" (("dave" NIL "dave" "here")) (("dave"
NIL "dave" "here")) (("dave" NIL "dave" "here")) ((NIL
NIL "shifflet" "holmes")("Emma J Brown" NIL "ejbrown"
"holmes.astro.cs.nps.navy.mil")(NIL NIL "everette"
"holmes")) NIL NIL NIL
"<000701c019bb$10ca3540$540a7883@astro.cs.nps.navy.mil
>") RFC822.SIZE 463 FLAGS () INTERNALDATE " 8-Sep-2000
17:37:00 +0000")
* 2 FETCH (UID 2 BODY[HEADER.FIELDS ("REFERENCES" "X-
REF" "X-PRIORITY" "X-MSMAIL-PRIORITY" "NEWSGROUPS")]
{2}
ENVELOPE ("Fri, 08 Sep 2000 10:35:48 -0700" "Test 3
from Netscape" (("kip" NIL "kip"
"henry.astro.cs.nps.navy.mil")) (("kip" NIL "kip"
"henry.astro.cs.nps.navy.mil")) (("kip" NIL "kip"
"henry.astro.cs.nps.navy.mil")) ((NIL NIL "shifflet"
"holmes")(NIL NIL "ejbrown" "holmes")(NIL NIL
"everette" "holmes")) NIL NIL NIL
"<39B92374.4C6E8359@henry.astro.cs.nps.navy.mil>")
RFC822.SIZE 429 FLAGS () INTERNALDATE " 8-Sep-2000
17:38:00 +0000")
* 3 FETCH (UID 3 BODY[HEADER.FIELDS ("REFERENCES" "X-
REF" "X-PRIORITY" "X-MSMAIL-PRIORITY" "NEWSGROUPS")]
{2}

ENVELOPE ("Fri, 8 Sep 2000 10:37:45 -0700 (Pacific
Daylight Time)" "Test 3 from Pine" (("David Shifflett"
NIL "dave" "astro.cs.nps.navy.mil")) (("David
Shifflett" NIL "dave" "astro.cs.nps.navy.mil"))
(("David Shifflett" NIL "dave"
"astro.cs.nps.navy.mil")) ((NIL NIL "shifflet"
"holmes")(NIL NIL "ejbrown" "holmes")(NIL NIL
"everette" "holmes")) NIL NIL NIL
"<Pine.WNT.4.20.0009081037100.307-
100000@henry.astro.cs.nps.navy.mil>") RFC822.SIZE 414
FLAGS () INTERNALDATE " 8-Sep-2000 17:40:00 +0000")
000J OK UID FETCH completed]

Send to SERVER  [000K UID FETCH 1:3 (UID FLAGS)]

Send to CLIENT  [* 1 FETCH (UID 1 FLAGS ())
* 2 FETCH (UID 2 FLAGS ())
* 3 FETCH (UID 3 FLAGS ())
000K OK UID FETCH completed]

Send to SERVER  [000L IDLE]

Send to CLIENT  [+ Ready for argument]

Send to SERVER  [DONE]

Send to CLIENT  [000L OK IDLE completed]

Send to SERVER  [000M CLOSE]

Send to CLIENT  [000M OK CLOSE completed]
Send to SERVER  [000N IDLE]

Send to CLIENT  [+ Ready for argument]

Send to SERVER  [DONE]

Send to CLIENT  [000N OK IDLE completed]

Send to SERVER  [000O SELECT "conf/xbox"]

Send to CLIENT  [* 4 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 968361985] UID validity status
* OK [UIDNEXT 5] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS ()] Permanent flags
* OK [UNSEEN 1] first unseen message in
/usr2/mail/shifflet/conf/xbox
000O OK [READ-ONLY] SELECT completed]

Send to SERVER  [000P IDLE]

Send to CLIENT  [+ Ready for argument]
Send to SERVER  [DONE]

Send to CLIENT  [000P OK IDLE completed]

Send to SERVER  [000Q UID FETCH 1:*
(BODY.PEEK[HEADER.FIELDS (References X-Ref X-Priority
X-MSMail-Priority Newsgroups)] ENVELOPE RFC822.SIZE
UID FLAGS INTERNALDATE)]

Send to CLIENT  [* 1 FETCH (UID 1 BODY[HEADER.FIELDS
("REFERENCES" "X-REF" "X-PRIORITY" "X-MSMAIL-PRIORITY"
"NEWSGROUPS")] {2}

ENVELOPE ("Wed, 06 Sep 2000 10:47:54 -0700" "Test of
multi-addressees via netscape" (("kip" NIL "kip"
"henry.astro.cs.nps.navy.mil")) (("kip" NIL "kip"
"henry.astro.cs.nps.navy.mil")) (("kip" NIL "kip"
"henry.astro.cs.nps.navy.mil")) ((NIL NIL "ejbrown"
"holmes.astro.cs.nps.navy.mil")(NIL NIL "shifflet"
"holmes.astro.cs.nps.navy.mil")) NIL NIL NIL
"<39B6834A.B3E9F28F@henry.astro.cs.nps.navy.mil>")
RFC822.SIZE 507 FLAGS () INTERNALDATE " 6-Sep-2000
17:49:00 +0000")
* 2 FETCH (UID 2 BODY[HEADER.FIELDS ("REFERENCES" "X-
REF" "X-PRIORITY" "X-MSMAIL-PRIORITY" "NEWSGROUPS")]
{44}
X-Priority: 3
X-MSMail-Priority: Normal

 ENVELOPE ("Fri, 8 Sep 2000 10:28:56 -0700" "Test 2
from Outlook" (("dave" NIL "dave" "here")) (("dave"
NIL "dave" "here")) (("dave" NIL "dave" "here")) ((NIL
NIL "shifflet" "holmes")("Emma J Brown" NIL "ejbrown"
"holmes.astro.cs.nps.navy.mil")(NIL NIL "everette"
"holmes")) NIL NIL NIL
"<000701c019ba$4a4bb1a0$540a7883@astro.cs.nps.navy.mil
>") RFC822.SIZE 457 FLAGS () INTERNALDATE " 8-Sep-2000
17:31:00 +0000")
* 3 FETCH (UID 3 BODY[HEADER.FIELDS ("REFERENCES" "X-
REF" "X-PRIORITY" "X-MSMAIL-PRIORITY" "NEWSGROUPS")]
{2}

 ENVELOPE ("Fri, 08 Sep 2000 10:30:31 -0700" "Test 2
from Netscape" (("kip" NIL "kip"
"henry.astro.cs.nps.navy.mil")) (("kip" NIL "kip"
"henry.astro.cs.nps.navy.mil")) (("kip" NIL "kip"
"henry.astro.cs.nps.navy.mil")) ((NIL NIL "shifflet"
"holmes")(NIL NIL "ejbrown" "holmes")(NIL NIL
"everette" "holmes")) NIL NIL NIL
"<39B92237.D511F6B6@henry.astro.cs.nps.navy.mil>")
RFC822.SIZE 429 FLAGS () INTERNALDATE " 8-Sep-2000
17:33:00 +0000")
* 4 FETCH (UID 4 BODY[HEADER.FIELDS ("REFERENCES" "X-
REF" "X-PRIORITY" "X-MSMAIL-PRIORITY" "NEWSGROUPS")]
{2}

 ENVELOPE ("Fri, 8 Sep 2000 10:32:35 -0700 (Pacific
Daylight Time)" "Test 2 from Pine" (("David Shifflett"
NIL "dave" "astro.cs.nps.navy.mil")) (("David
Shifflett" NIL "dave" "astro.cs.nps.navy.mil"))
(("David Shifflett" NIL "dave"
"astro.cs.nps.navy.mil")) ((NIL NIL "shifflet"
"holmes")(NIL NIL "ejbrown" "holmes")(NIL NIL
"everette" "holmes")) NIL NIL NIL
"<Pine.WNT.4.20.0009081031530.188-
100000@henry.astro.cs.nps.navy.mil>") RFC822.SIZE 414
FLAGS () INTERNALDATE " 8-Sep-2000 17:35:00 +0000")
000Q OK UID FETCH completed]
Send to SERVER  [000R UID FETCH 1:4 (UID FLAGS)]
Send to CLIENT  [* 1 FETCH (UID 1 FLAGS ())
* 2 FETCH (UID 2 FLAGS ())
* 3 FETCH (UID 3 FLAGS ())
* 4 FETCH (UID 4 FLAGS ())
000R OK UID FETCH completed]
Send to SERVER  [000S IDLE]

Send to CLIENT  [+ Ready for argument]

Send to SERVER  [DONE]

Send to CLIENT  [000S OK IDLE completed]

Send to SERVER  [000T CLOSE]

Send to CLIENT  [000T OK CLOSE completed]

Send to SERVER  [000U IDLE]

Send to CLIENT  [+ Ready for argument]

Send to SERVER  [DONE]

Send to CLIENT  [000U OK IDLE completed]

Send to SERVER  [000V SELECT "unclass/xbox"]

Send to CLIENT  [* 2 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 968361947] UID validity status
* OK [UIDNEXT 3] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS ()] Permanent flags
* OK [UNSEEN 1] first unseen message in
/usr2/mail/shifflet/unclass/xbox
000V OK [READ-ONLY] SELECT completed]

Send to SERVER  [000W IDLE]

Send to CLIENT  [+ Ready for argument]

Send to SERVER  [DONE]

Send to CLIENT  [000W OK IDLE completed]
Send to SERVER [000X UID FETCH 1:*
(BODY.PEEK[HEADER.FIELDS (References X-Ref X-Priority
X-MSMail-Priority Newsgroups)] ENVELOPE RFC822.SIZE
UID FLAGS INTERNALDATE)]

Send to CLIENT [* 1 FETCH (UID 1 BODY[HEADER.FIELDS
("REFERENCES" "X-REF" "X-PRIORITY" "X-MSMAIL-PRIORITY"
"NEWSGROUPS")] {2}

ENVELOPE ("Fri, 08 Sep 2000 10:22:59 -0700" "Test 1
from Netscape" (("kip" NIL "kip"
"henry.astro.cs.nps.navy.mil")) (("kip" NIL "kip"
"henry.astro.cs.nps.navy.mil")) (("kip" NIL "kip"
"henry.astro.cs.nps.navy.mil")) ((NIL NIL "shifflet"
"holmes")(NIL NIL "ejbrown" "holmes")(NIL NIL
"everette" "holmes")) NIL NIL NIL
"<39B92073.342DC4A6@henry.astro.cs.nps.navy.mil>")
RFC822.SIZE 432 FLAGS () INTERNALDATE " 8-Sep-2000
17:25:00 +0000")
* 2 FETCH (UID 2 BODY[HEADER.FIELDS ("REFERENCES" "X-
REF" "X-PRIORITY" "X-MSMAIL-PRIORITY" "NEWSGROUPS")]
{2}

ENVELOPE ("Fri, 8 Sep 2000 10:26:15 -0700 (Pacific
Daylight Time)" "Test 1 from Pine" (("David Shifflett"
NIL "dave" "astro.cs.nps.navy.mil")) (("David
Shifflett" NIL "dave" "astro.cs.nps.navy.mil"))
(("David Shifflett" NIL "dave"
"astro.cs.nps.navy.mil")) ((NIL NIL "shifflet"
"holmes")(NIL NIL "ejbrown" "holmes")(NIL NIL
"everette" "holmes")) NIL NIL NIL
"<Pine.WNT.4.20.0009081025290.303-
100000@henry.astro.cs.nps.navy.mil>") RFC822.SIZE 420
FLAGS () INTERNALDATE " 8-Sep-2000 17:29:00 +0000")
000X OK UID FETCH completed]

Send to SERVER [000Y UID FETCH 1:2 (UID FLAGS)]

Send to CLIENT [* 1 FETCH (UID 1 FLAGS ())
* 2 FETCH (UID 2 FLAGS ())
000Y OK UID FETCH completed]

Send to SERVER [000Z IDLE]

Send to CLIENT [+ Ready for argument]
Send to SERVER  [DONE] ZZZZ LOGOUT]

Send to SERVER  [DONE]ZZZZ LOGOUT]
```

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF REFERENCES

1. National Security Agency Evaluated Products List, [http://www.radium.ncsc.mil/tpep/epl]

2. The IMAP Connection - "*What is IMAP?*" [http://www.imap.Org/about/whatisIMAP.html

3. Bryer-Joyner, S., Heller, Scott D., "*Secure Local Area Network Services for a High Assurance Multilevel Network*", March 1999

4. Eads, Bradley R., "*Developing a High Assurance Multilevel Mail Server*, March 1999

5. USIA, U.S. Foreign Policy Agenda [http://usinfo.state.gov/journal/1198/ijpe/pj48min.htm]

6. Brinkley, Donald L, Schell Roger R., "*Concepts and Terminology for Computer Security*", March 1993

7. Clark, David D., Wilson, David R., "*A Comparison of Commercial and Military Computer Security Policy*", 1987

8. Sterne, Daniel F., "*On the Buzzword "Security Policy"*

9. White, Gregory B., Fisch, Eric A., Pooch Udo W., *Computer Systems and Network Security*

10. Wang Government Services, XTS-300 Users Manual, March 1998, Stop 4.42 Version

11. University of Washington, Pine Information Center [http://www.washington.edu/pine/overview/index/html]

12. Server-Workstation, issues December 1999 through July 2000

13. Netscape Messenger [http://home.netscape.com/communicator/messenger/v4.0/index.html]

14. Microsoft Outlook [http://www.microsfot.com/office/outlook/olfeatur.htm]

15. Lotus Notes Release R5 [http://www.lotus.com/products/rdweb.nsf/webpi/Notes?opendocument]

16. Anderson, James P., *"Computer Security Technology Planning Study"*, October 1972

17. Hix, Deborah, Hartson, Rex, H., *Developing User Interfaces Ensuring Usability Through Product and Process*

18. Schell, Roger R., Brinkley Donald L., *"Evaluation Criteria for Trusted Systems"*, March 1993

19. Center for Information Assurance and INFOSEC Studies and Research, Naval Postgraduate School, Monterey, CA 93940, CS4600 Secure Systems

20. Wood, David, *Programming Internet Mail*, 1999

21. Network Working Group, RFCs 2060, University of Washington, December 1996

# INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center .............. 2
   8725 John J. Kingman Road, Ste 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library .............................. 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Chairman, Code CS ................................ 1
   Naval Postgraduate School
   Monterey, CA 93943-5101

4. Dr. Cynthia E. Irvine ............................ 3
   Computer Science Department, Code CS/Ic
   Naval Postgraduate School
   Monterey, CA 93943-5000

5. Mr. James P. Anderson ............................ 1
   James P. Anderson Company
   Box 42
   Fort Washington, PA

6. David Shifflett .................................. 1
   Computer Science Department, Code CS
   Naval Postgraduate School
   Monterey, CA 93943-5000

7. Theresa M. Everette ............................. 2
5565 St. Charles Drive
Woodbridge, VA 22193

8. Mr. Paul Pitelli ................................. 1
National Security Agency
Research and Development Building
R2, Technical Director
9800 Savage Road
Fort Meade, MD 20755-6000

9. Mr. Richard Hale ................................. 1
Defense Information Systems Agency
5600 Columbia Pike, Suite 400
Falls Church, VA 22041-3230

10. Ms. Barbara Flemming ............................ 1
Defense Information Systems Agency
5600 Columbia Pike, Suite 400
Falls Church, VA 22041-3230

11. Carl Siel ....................................... 1
Space and Warfare Systems Command
PMW 161
Building OT-1, Room 1024
4301 Pacific Highway
San Diego, CA 92110-3127

12. Commander, Naval Security Group Command .......... 1

Naval Security Group Headquarters
9800 Savage Road
Suite 6585
Fort Meade, MD 20755-6585

13. Ms Deborah Cooper ................................. 1

Deborah M. Cooper Company
P. O. Box 17753
Arlington, VA 22216

14. Ms. Louis Davidson ................................ 1

N643
Presidential Tower 1
2511 South Jefferson Davis Highway
Arlington, VA 22202

15. Mr. William Dawson ................................ 1

Community DIO Office
Washington, DC 20505

16. Capt James Newman ................................. 1

N64
Presidential Tower 1
2511 South Jefferson Davis Highway
Arlington, VA 22202

17. Mr. James Knoke .................................. 1
Wang Government Services, Inc
7900 Westport Dr.
McLean, VA 22102-4299

18. Mr. Mike Focke .................................... 1
Wang Government Services, Inc
7900 Westport Dr.
McLean, VA 22102-4299