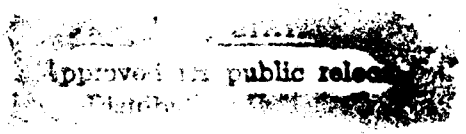
Message Handling System (X.400) Threats, Vulnerabilities, and Countermeasures

M 93B0000037
April 1993

Michelle J. Gosselin



MITRE
Bedford, Massachusetts

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 1993	

4. TITLE AND SUBTITLE

Message Handling System (X.400) Threats, Vulnerabilities, and Countermeasures

5. FUNDING NUMBERS

6. AUTHOR(S)

Michelle J. Gosselin

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

The MITRE Corporation
202 Burlington Road
Bedford, MA 01730-1420

8. PERFORMING ORGANIZATION REPORT NUMBER

M 93B0000037

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This annotated briefing discusses the security issues of integrating into the MITRE network architecture an implementation of an X.400 based message handling system (MHS). Background concerning both the MITRE Open Systems Interconnection (OSI) integration project and X.400 is given. Threats, vulnerabilities, and countermeasures are then discussed. Finally, recommendations for integrating an MHS into the MITRE networks are presented.

14. SUBJECT TERMS

security, network architecture, message handling system, open systems interconnection, X.400, threats, vulnerabilities, countermeasures

15. NUMBER OF PAGES

59

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

Message Handling System (X.400) Threats, Vulnerabilities, and Countermeasures

M 93B0000037
April 1993

Michelle J. Gosselin

Contract Sponsor N/A
Contract No. N/A
Project No. G90N
Dept. G021

Approved for public release;
distribution unlimited.

MITRE
Bedford, Massachusetts

ABSTRACT

This annotated briefing discusses the security issues of integrating into the MITRE network architecture an implementation of an X.400 based message handling system (MHS). Background concerning both the MITRE Open Systems Interconnection (OSI) integration project and X.400 is given. Threats, vulnerabilities, and countermeasures are then discussed. Finally, recommendations for integrating an MHS into the MITRE networks are presented.

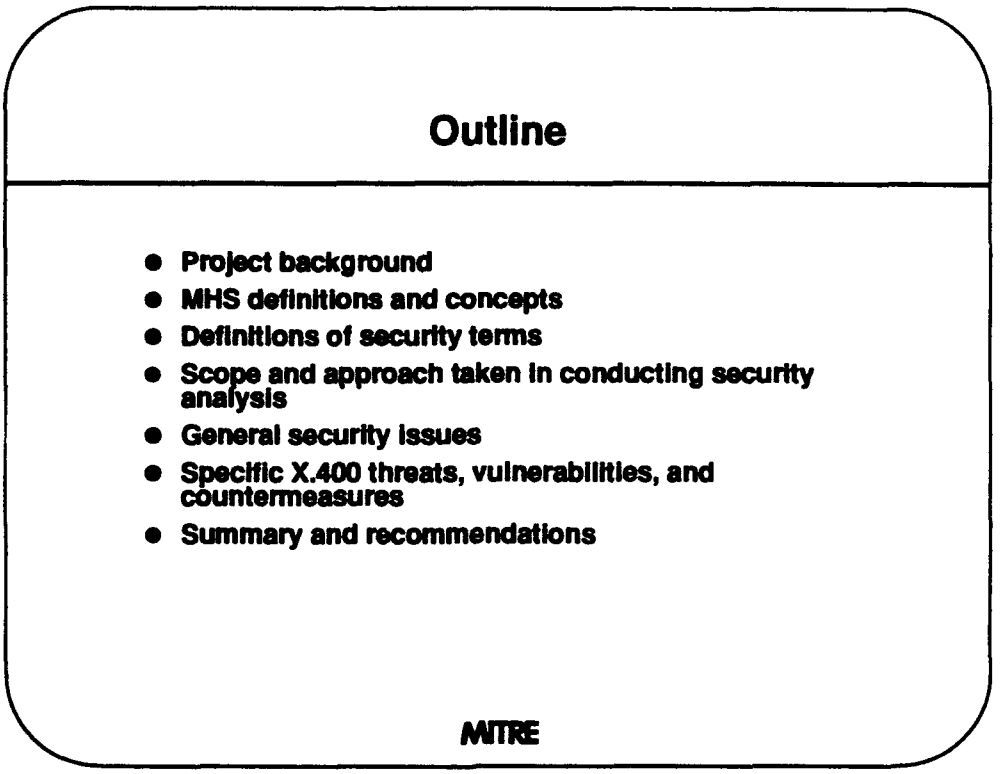
ACKNOWLEDGMENTS

The author would like to thank David Baldauf, Joel Jacobs, Ann McLaughlin, David Miller, and Marilyn Real for their contributions to this effort.

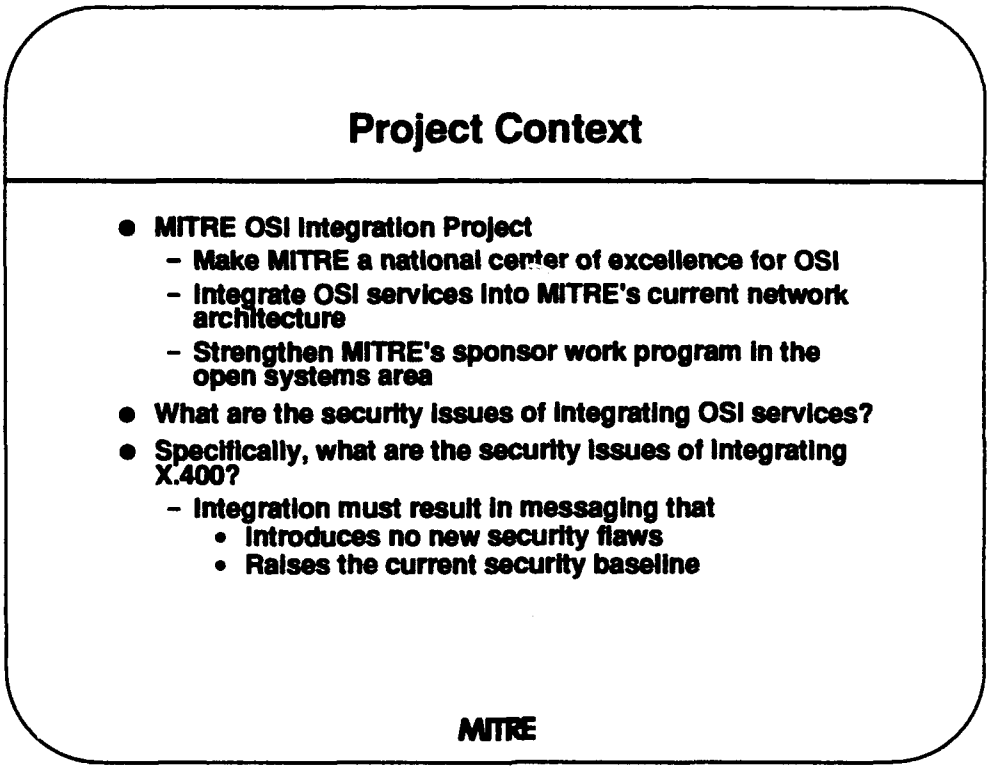
Message Handling System (X.400) Threats, Vulnerabilities, and Countermeasures

MITRE

This briefing discusses the security issues of integrating into the MITRE network architecture an implementation of a message handling system (MHS) as described in the International Telegraph and Telephone Consultative Committee (CCITT) "Data Communication Networks Message Handling Systems, Recommendations X.400-X.420." The integration of a service includes the operation of the service within MITRE networks and across the MITRE security boundary. (The security boundary is defined in later slides.) Threats, vulnerabilities, and countermeasures are discussed.



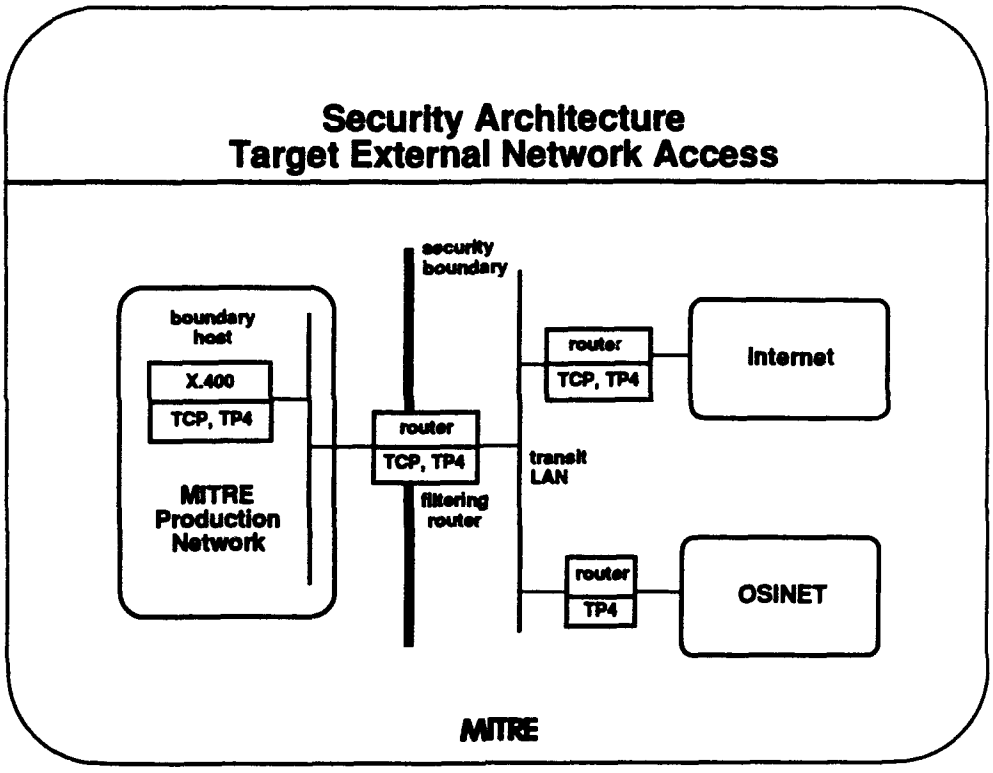
The outline for the briefing is as follows. First, background concerning the project for which this work was conducted is given. This is followed by an overview of MHS definitions and concepts. Then, definitions of the terms "threat," "vulnerability," and "countermeasure" are given. Following these definitions, the scope and approach taken to identify security issues concerning the integration of an MHS are then presented. General security issues that relate to the integration of any service across the MITRE security boundary are then discussed. Finally, these MHS-specific security issues and their countermeasures are identified followed by a summary and recommendations.



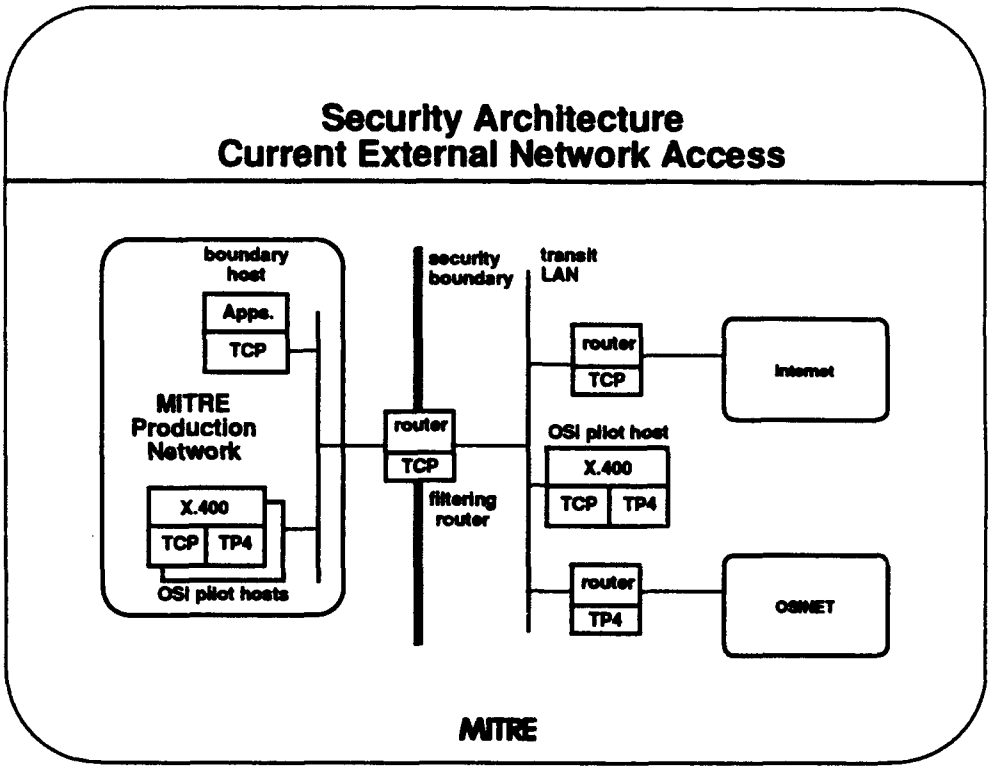
This work was conducted for the MITRE Open Systems Interconnection (OSI) Integration Project. The purpose of this project is to:

1. make MITRE a center of excellence for OSI,
2. integrate OSI services into MITRE's network to both enhance corporate network services and to facilitate the first goal, and
3. strengthen MITRE's sponsor work program in the open systems area using the experience and resources of the project.

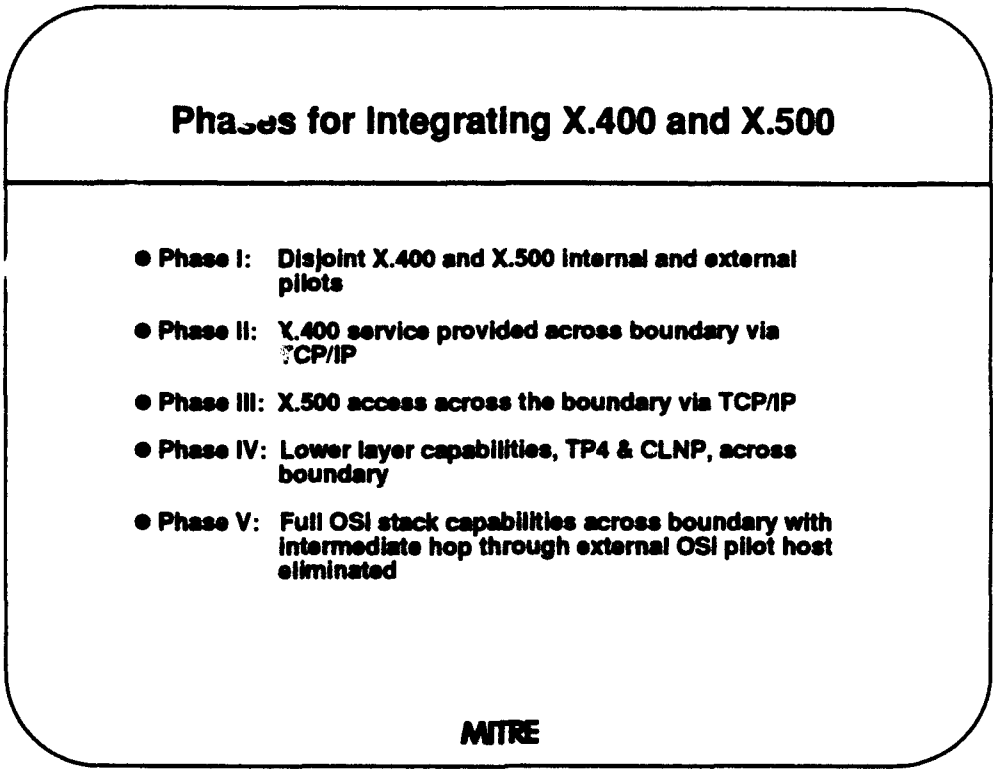
The first two services that the MITRE OSI Integration project is planning to integrate are X.400, a message handling system service, and X.500, a directory service. Before integrating these services, however, the environment that MITRE operates within must be considered. MITRE must protect its networks and electronic information from accidental release and outside attacks. Therefore, the security issues of integrating any new service must be investigated before integrating that service. In preparation for integrating the X.400 MHS into the MITRE network architecture, this briefing discusses the security issues investigated regarding the X.400 integration. During this investigation, a determination had to be made as to whether or not the integration of X.400 would introduce new security flaws. In fact, the current security baseline should be raised (some existing flaws should be eliminated).



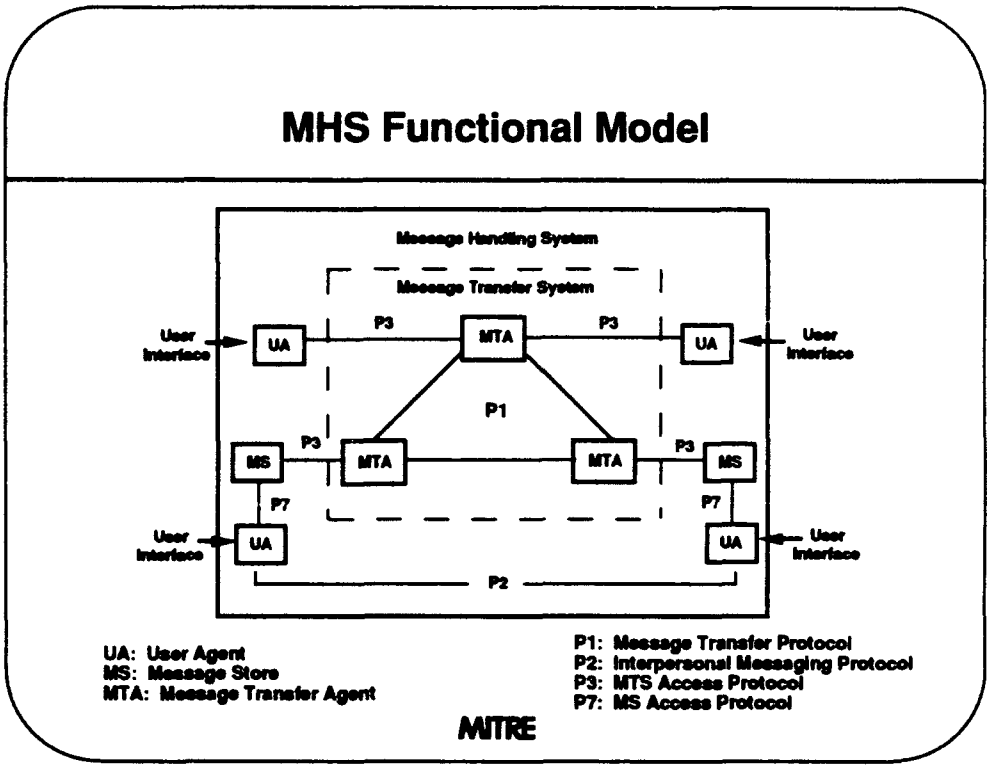
This slide depicts the target security architecture for access from and to external networks. A filtering router that allows only TCP/IP and TP4/CLNP packets through is located between the internal MITRE production network and external networks. Through filtering, this router establishes the security boundary between MITRE and the external networks. A limited set of internal boundary hosts, capable of processing both TCP and TP4 packets, are allowed to connect through the router. Other hosts within MITRE must connect to these boundary hosts before gaining access to external networks.



Currently, only TCP traffic is allowed to pass through the security boundary. TP4 (OSI) traffic is not allowed. As a result, we currently have disjoint internal and external OSI resources operating in pilot efforts of the MITRE OSI Integration Project. The term "disjoint" means that the internal pilot hosts do not communicate with the external pilot hosts. There is a concurrent ongoing effort addressing OSI packet filtering and routing to enable TP4 traffic to flow across the boundary. If this work is not complete when the project is ready to deploy MHS across the boundary, dual stacked (TCP and TP4) hosts will be used to pass X.400 messaging traffic over TCP across the boundary as an interim transition step. A phased transition plan is described on the next slide.



In transitioning to the target architecture, a phased approach is being taken. The first phase involves upper layer OSI (X.400 and X.500) pilot hosts that exist both internally and externally. The internal hosts communicate with each other, and the external host can communicate with other external OSI hosts. However, the pilot hosts in this phase do not send traffic across the boundary. The second phase is to establish X.400 traffic across the boundary using TCP/IP in the lower layers. The third phase is to add X.500 traffic. The second and third phase can occur concurrently. Phase four allows TP4 transport and CLNP network layer traffic to cross the boundary. Phase five, the target architecture, is the addition of full OSI stack capabilities at both the boundary hosts and the filtering gateway which allows the intermediate hop through the external OSI pilot host to be eliminated. At this point, we will also begin investigating additional OSI services which will require external access (e.g., File Transfer, Access, and Management (FTAM)).



Now that some project background has been given, some concepts related to X.400 will be given in preparation to discussing threats, vulnerabilities, and countermeasures specific to X.400.

This slide depicts the functional model of an MHS. The heart of an MHS is the message transfer system (MTS). The message transfer system is responsible for relaying messages within the MHS so that they can be delivered to the appropriate user. The functional entity that performs the transfers is a message transfer agent (MTA).

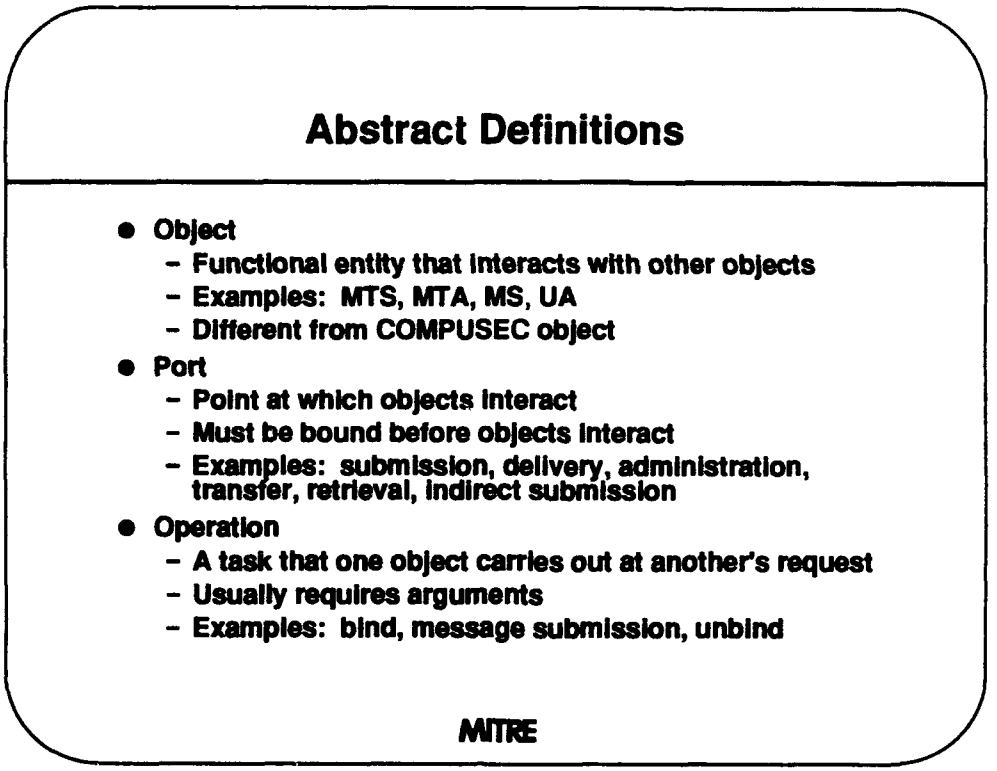
MTAs transfer messages to each other via the message transfer protocol (P1). MTAs provide access to the MTS by MHS components external to the MTS via the MTS access protocol (P3). These components include message stores (MS) and user agents (UA).

A user interfaces with a UA to gain access to the MHS to send messages to other users. A UA can either directly access the MTS via P3 or can indirectly access the MTS via an MS. Access to the MS is provided through the MS access protocol (P7). The MS stores messages that the user is submitting to the MTS and messages that are to be delivered to the user.

The higher level protocol that provides messaging between users through UAs and makes use of P1, P3, and P7 is the interpersonal messaging protocol (P2).

Not depicted in the functional model, but included in the recommendation, is an access unit (AU). An AU links another communication system (e.g., a physical delivery system or the telex network) to the MTS. Security issues relating to AUs were not addressed at this time, and AUs will not be used in the initial deployment of an X.400 implementation.

Although the functional model depicts the UA, the MS, and the MTA as being distinct entities, it is possible for them to be physically coresident.

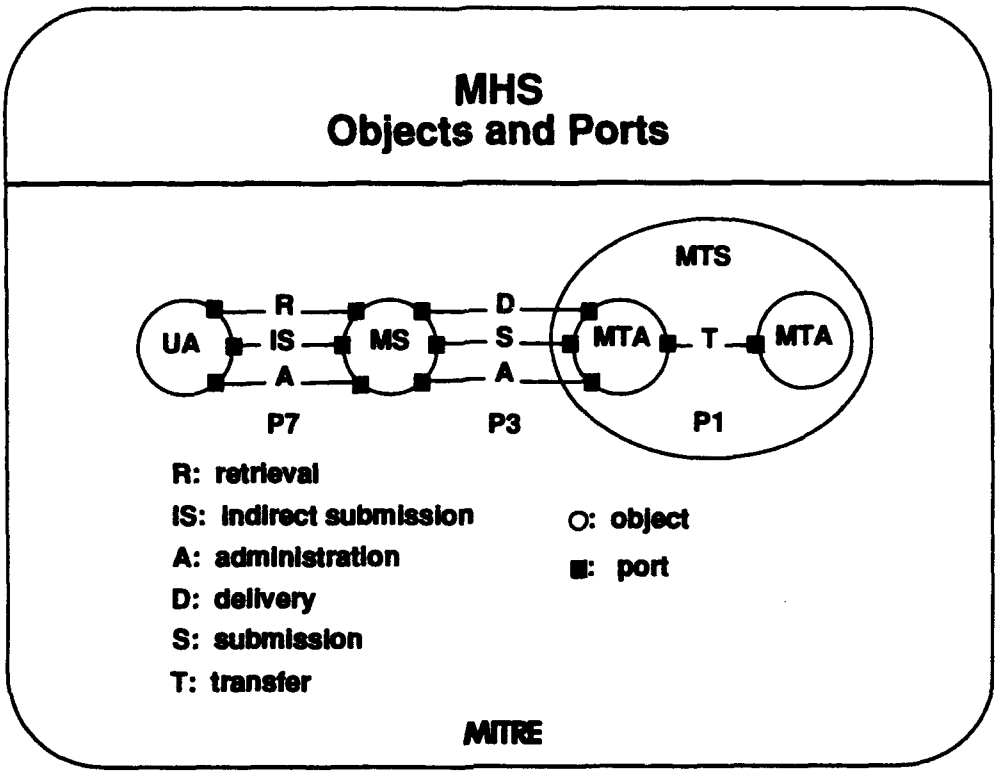


In discussing the behavior of the MHS functional model, the recommendation gives a number of abstract definitions. For example, there is a concept of an object. An object is a functional entity that interacts with other objects. Examples of objects are an MTS, an MTA, an MS, and a UA.

For purposes of clarification, the MHS definition of object is different from the computer security (COMPUSEC) definition of object. A COMPUSEC object is a container of information and is accessed by subjects. The COMPUSEC definition of subject is closer to the MHS definition of object. COMPUSEC subjects can interact with each other and act upon COMPUSEC objects.

Objects have ports that must be bound to ports of a similar type at another object in order for the objects to interact. Examples of ports include ports for submission, delivery, administration, transfer, retrieval, and indirect submission.

An operation is a task that one object carries out at the request of another object. An operation usually requires the initiator to supply arguments. Examples of operations include bind, message submission, and unbind.



This slide depicts the various objects and ports of an MHS. Individual ports are defined as follows:

A retrieval port allows a user agent to retrieve a message from an MS.

An indirect submission port allows a UA to submit a message to an MS for submission to an MTA.

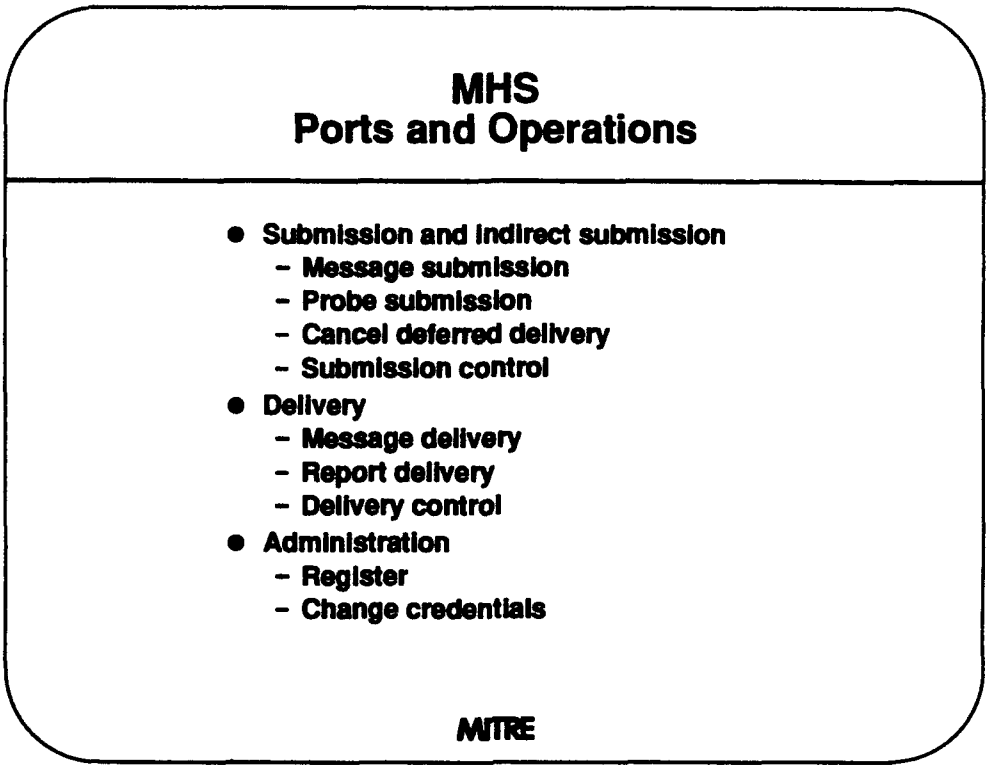
An administration port allows a UA to change administration information held at the MS and the MTA concerning the user associated with the UA.

A delivery port allows an MS to accept delivery of a message from an MTA on behalf of a UA.

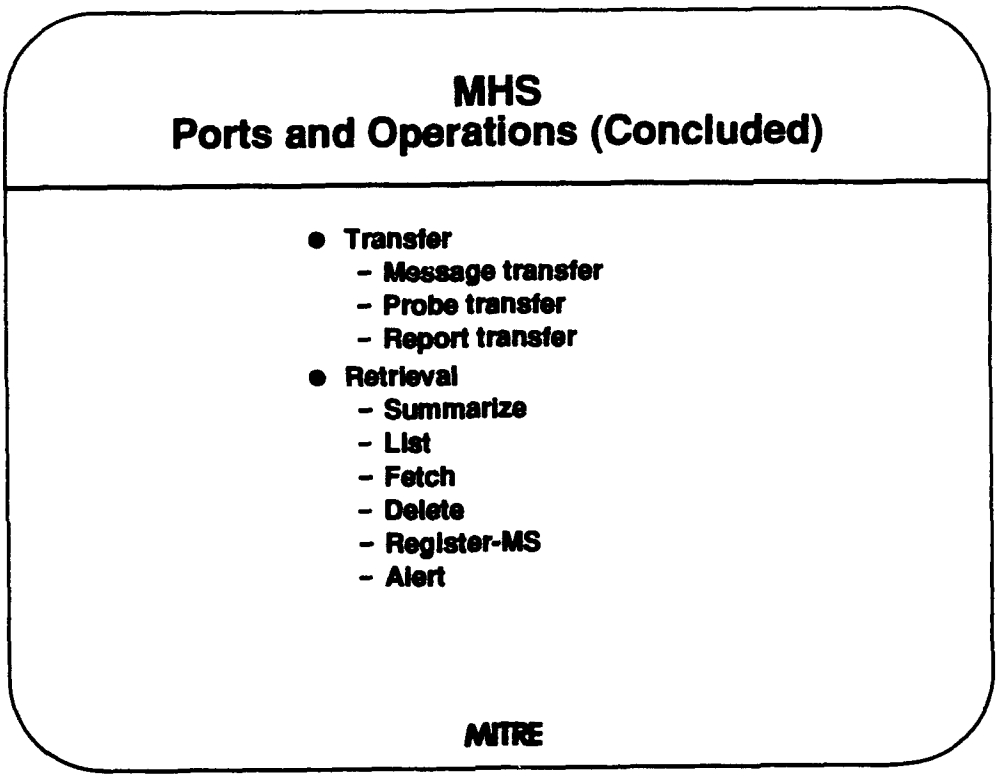
A submission port allows an MS to submit a message to an MTA on behalf of a UA.

A transfer port allows one MTA to transfer a message to another MTA.

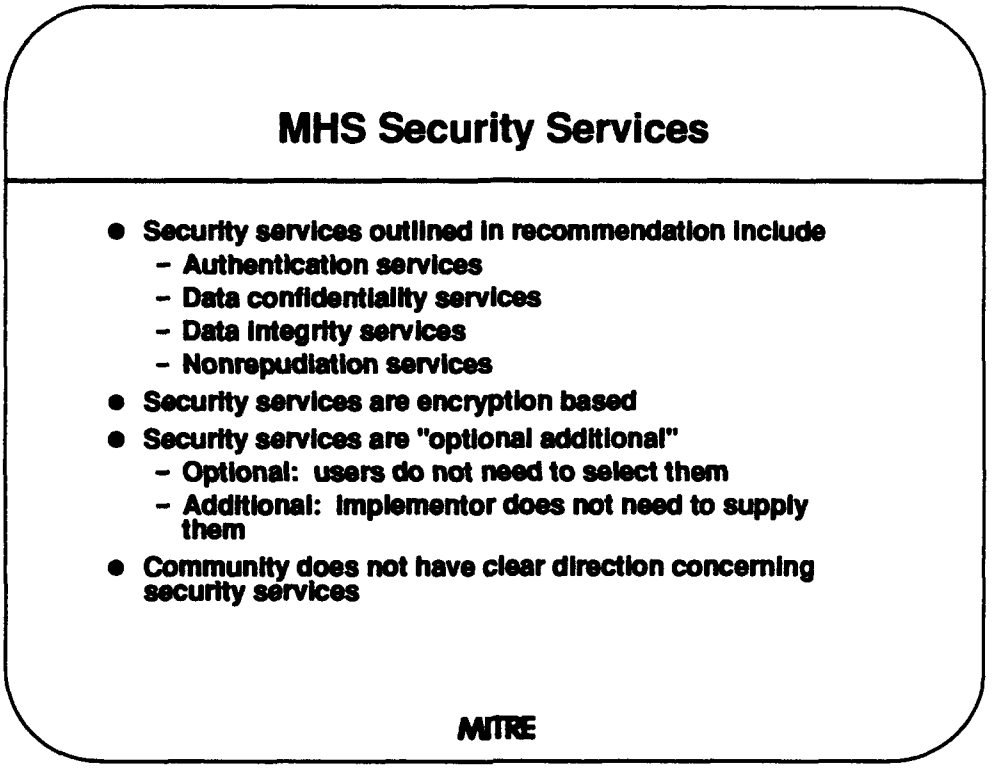
In the absence of an MS, the UA can directly access the MTA via P3 using the delivery, submission, and administration ports.



For each port within an MHS, a number of operations can take place across that port. This slide lists the operations associated with a particular port.



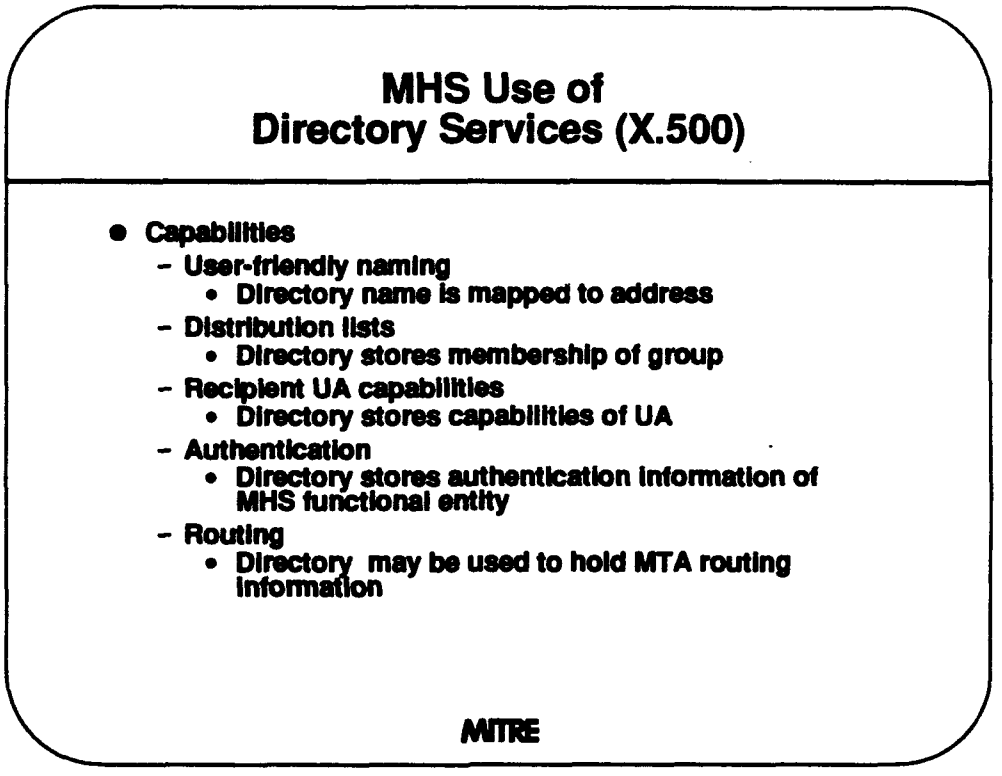
This slide continues to list the operations associated with a particular port.



To address the security threats that will be identified, the X.400 recommendation outlines a number of security services. These security services include authentication services, data confidentiality services, data integrity services, and nonrepudiation services. These security services are employed via encryption and may involve the exchange of keys.

The recommendation categorizes the security services as being "optional additional." The term "optional" indicates that the users of the MHS or MTS do not need to select these services when transferring messages. The term "additional" indicates that the services are not required to be supplied with the implementation by the vendor. Therefore, availability of these security services cannot be counted upon to provide countermeasures against identified threats and vulnerabilities.

In addition to the potential absence of these security services, a clear direction by various MHS communities towards the use of the security services has not been made. Alternatives, such as Pre-Message Security Protocol (PMSP), Message Security Protocol (MSP), and Privacy Enhanced Mail (PEM), will be considered in the future to provide enhanced security services.



The directory defined by the X.500-Series of Recommendations provides capabilities that can be used in message handling. These capabilities fall into the following four categories:

1. User-friendly naming: The originator or recipient of a message can be identified by a directory name rather than a machine-oriented address. The MHS can derive the address from the directory name by consulting the directory.

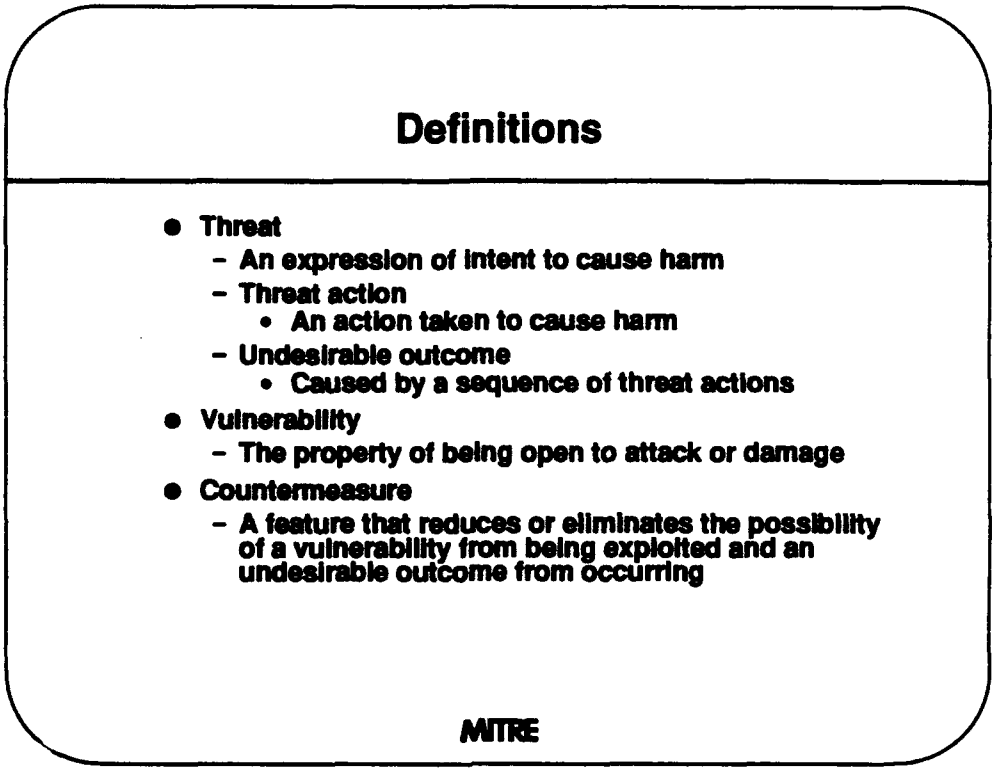
2. Distribution lists (DLs): A DL can be a group whose membership is stored in the directory. The originator simply supplies the name of the list, and the MHS can obtain the directory names of the individual recipients by consulting the directory.

3. Recipient UA capabilities: Capabilities of a UA, such as deliverable encoded information types, deliverable content types, and maximum content length, can be stored in a directory entry. The MHS can obtain these capabilities by consulting the directory.

4. Authentication: Authentication information required to establish the identity of two functional entities prior to communication can be stored in the directory.

5. Routing: There is ongoing work to standardize representations of MTA routing information in an X.500 directory. This should improve the management and scalability of MTA routing tables.

Security implications of X.500 integration have not been investigated at this time. X.500 security will be reviewed in a separate X.500 security evaluation.

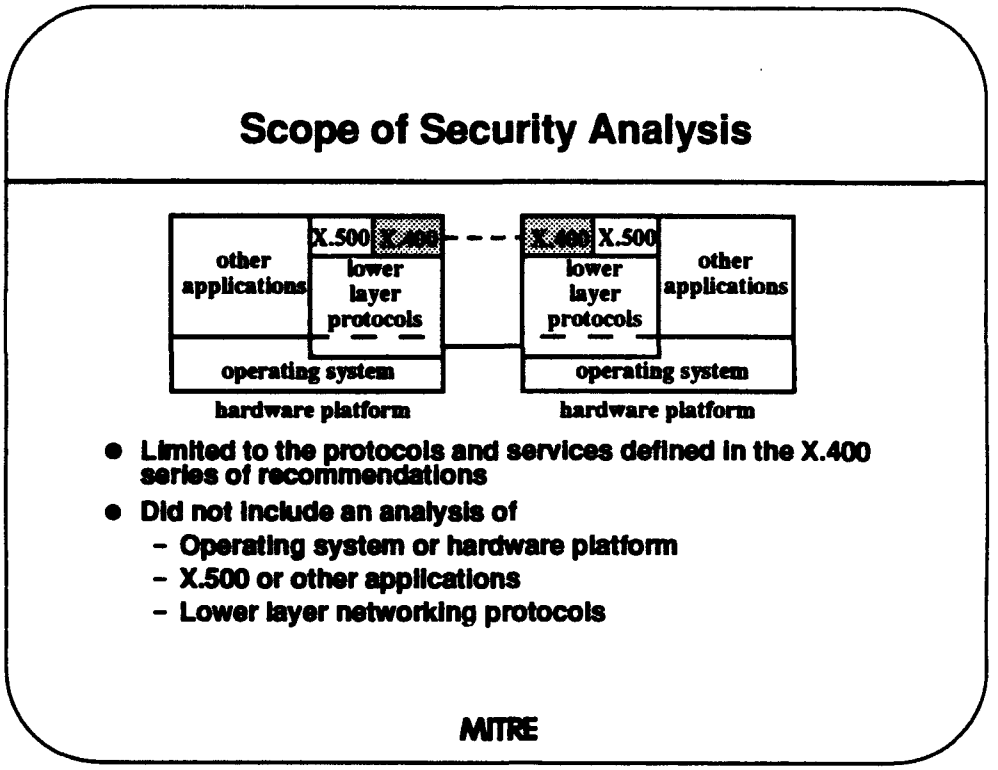


Now that project background and MHS information have been presented, some security concepts can now be discussed. First, the terms "threat," "vulnerability," and "countermeasure" are defined.

Several related concepts are bundled together into the term "threat." A threat can be an expression of intent to cause harm (e.g., burglary). It can also be an action taken to cause harm (e.g., surveillance). It can also be the undesirable outcome that results from a sequence of actions taken to cause harm (e.g., loss of assets).

A vulnerability is the property of being open to attack or damage (i.e., an undesirable outcome). For instance, an unlocked door makes a house vulnerable to the threat of burglary.

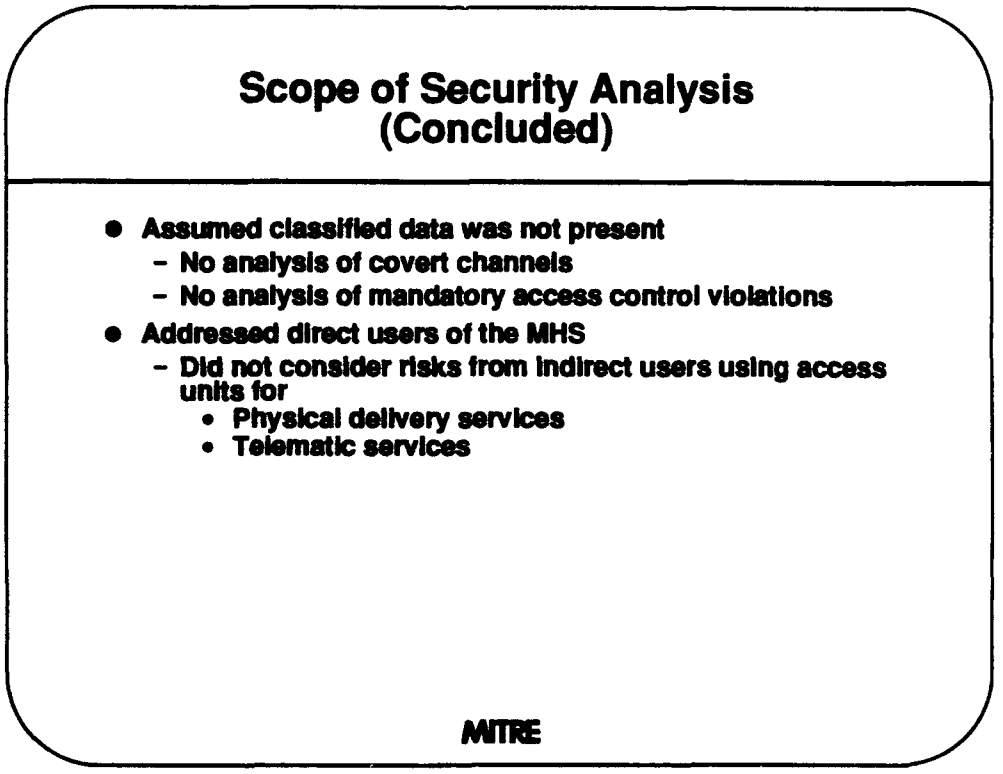
A countermeasure is a feature that reduces or eliminates the possibility of a vulnerability from being exploited and an undesirable outcome from occurring. For instance, a deadbolt lock makes a house less vulnerable to the threat of burglary.



Before discussing the analysis and results, the scope of the analysis and the approach taken in performing the analysis must be presented.

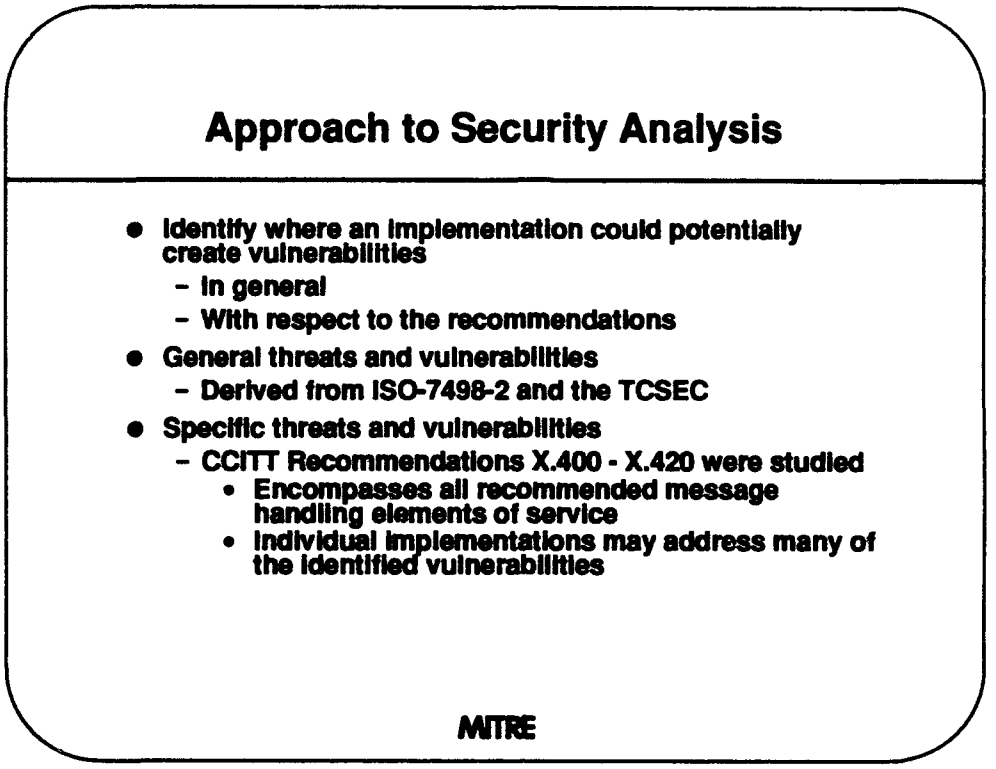
The scope of the effort was limited to investigating those security issues that pertain to the protocols and services defined in the X.400 series of recommendations. The areas considered are the greyed boxes and the dashed line between them (representing the P1, P2, P3, and P7 protocols).

The analysis did not consider risks relating to the operating system (except to a minimal extent), the hardware platform, X.500, other applications, and the lower layer networking protocols. Networking vulnerabilities not addressed include, for example, eavesdropping. However, there are no new networking related vulnerabilities introduced with the use of X.400 that do not exist with current messaging.



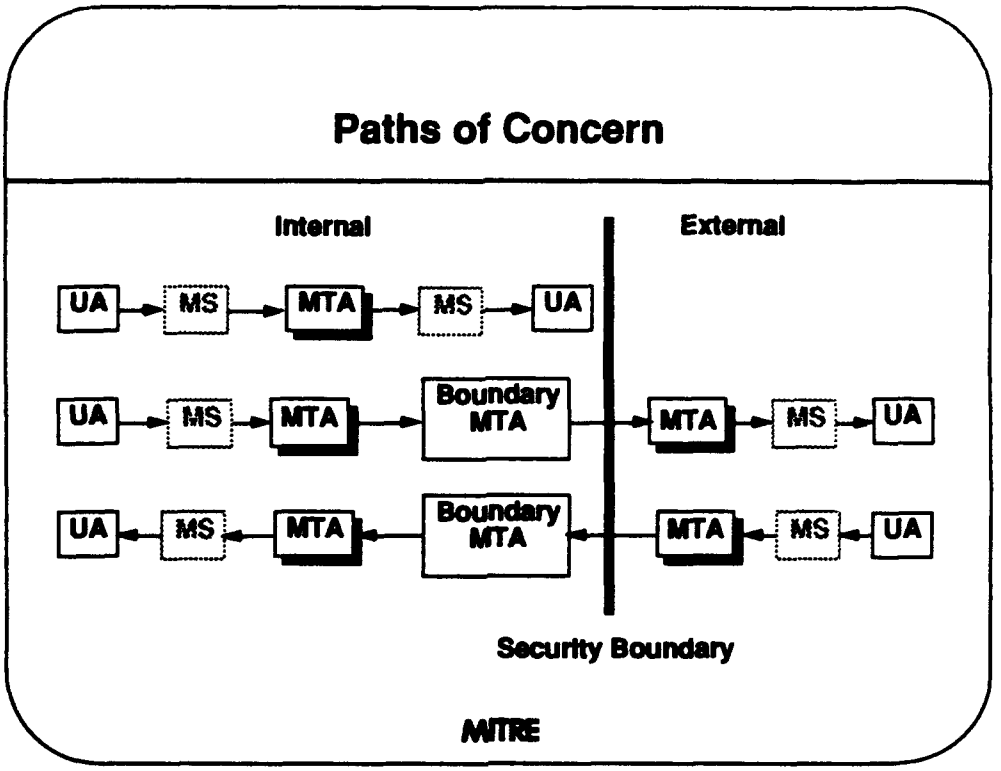
The analysis also assumed that classified data was not present on MITRE networks or systems connected to those networks since this is against MITRE security policy. Therefore, there was no analysis of possible covert channels or mandatory access control violations.

Other vulnerabilities not considered were those related to indirect users since, initially, X.400 will not be configured to have these users when integrated into MITRE networks. Indirect users include those users who require physical, rather than electronic, delivery of their messages, or users that employ telematic services to receive, for example, voice mail.



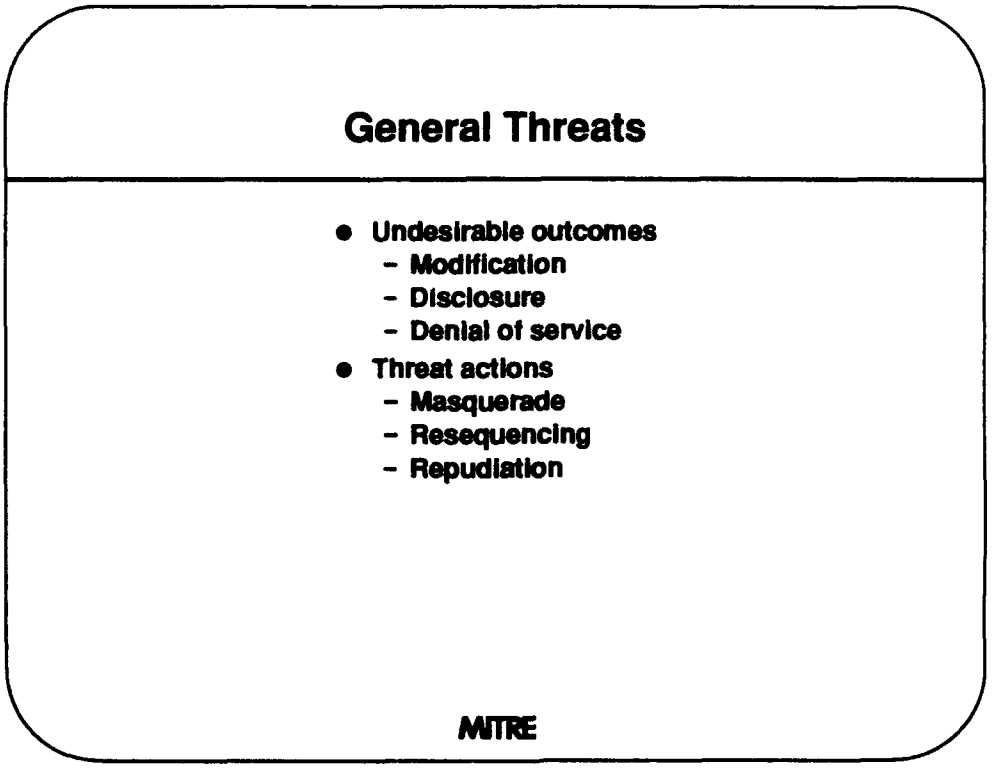
The approach taken in performing the security analysis was to identify where an implementation could potentially create vulnerabilities once integrated into the MITRE networks. The analysis was conducted at two levels. First, threats and vulnerabilities that could arise with the introduction of any new service were analyzed. These threats and vulnerabilities were derived from *Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 2: System Architecture* (ISO-7498-2) and the *Trusted Computer System Evaluation Criteria* (TCSEC).

The second level of the approach was to identify threats and vulnerabilities specific to MHS by thoroughly searching CCITT recommendations X.400 through X.420 for possible security issues. These recommendations encompass all aspects of message handling elements of service. Individual implementations have not yet been reviewed; however, we expect that specific implementations will address many of the identified vulnerabilities.



In investigating the specific vulnerabilities, the different paths that an MHS operation (e.g., message submission) and accompanying data (e.g., the message) could take within the MHS were also considered. One path is a strictly internal one. A request for an operation can be initiated within MITRE and responded to within MITRE. Other requests could be initiated within MITRE but be responded to externally. Finally, requests could be initiated externally and be responded to internally. Although the first two paths are important, the external to internal path poses the highest risk to MITRE and must be considered carefully. (The consideration of a strictly external path was not within the scope of this task.)

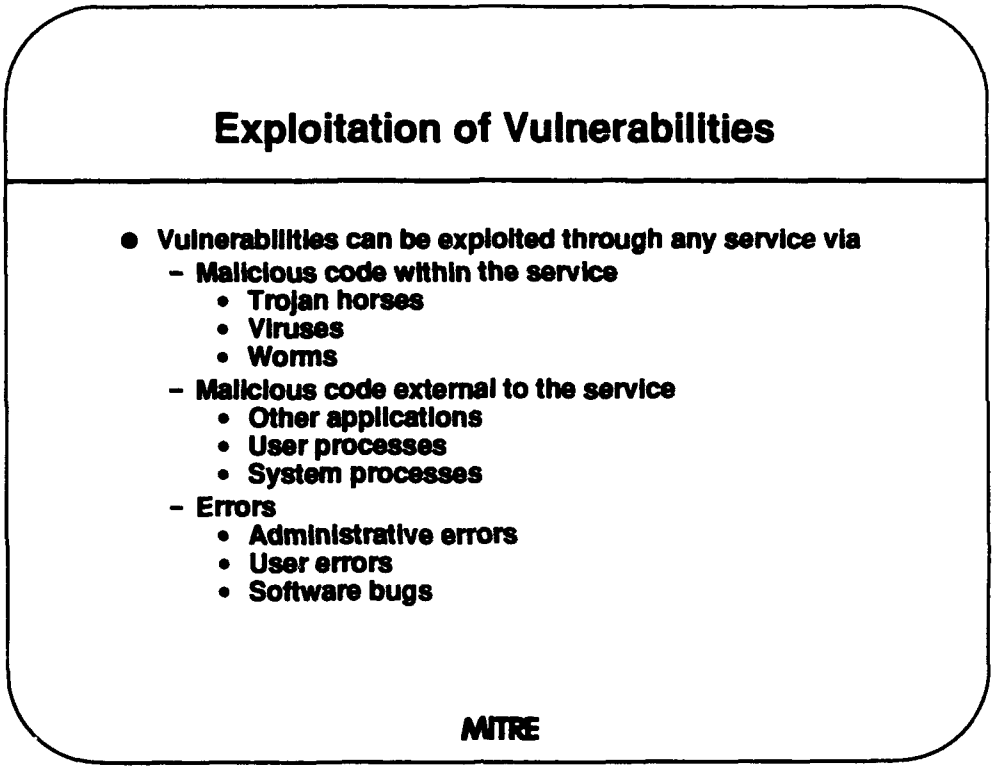
The boxes in grey represent elements that are optional. The shadowed boxes represent elements of which there could be one or several.



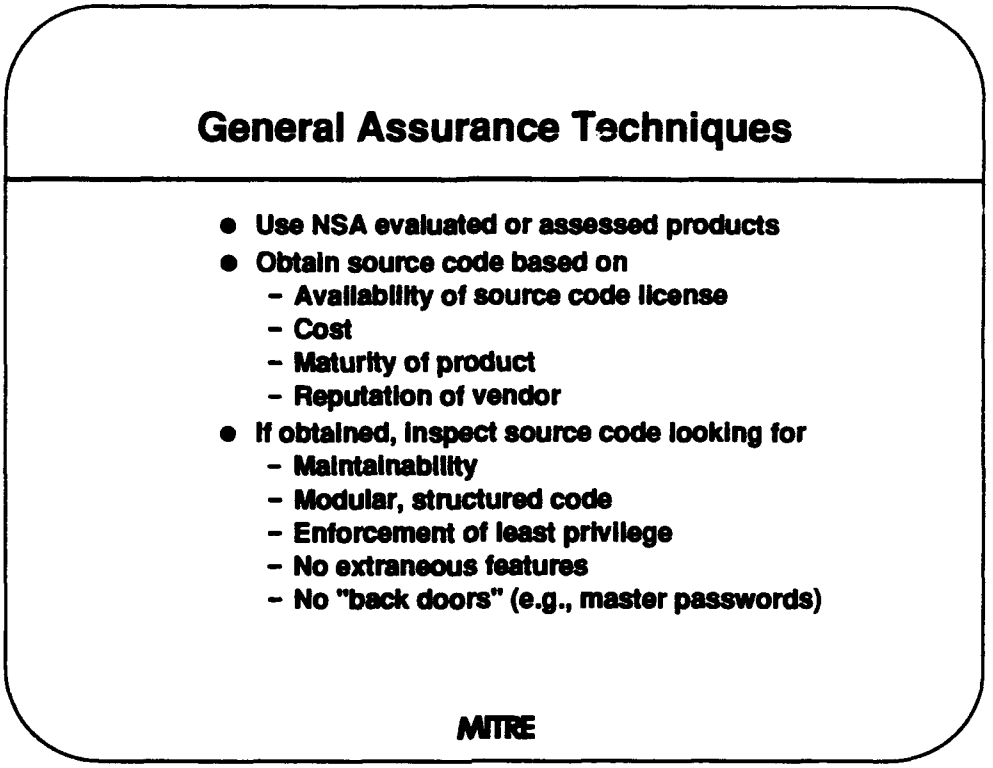
There are a number of general threats that exist for any service that is to be integrated (where "service" includes OSI services, other networking software, and other applications). The three major types of undesirable outcomes an attacker might seek to achieve include modification, disclosure, and denial of service. Modification is the unauthorized altering of data, which includes changing, adding, or deleting data. Disclosure occurs when a subject reads data without proper authorization. Denial of service is when a feature or system is rendered unavailable.

For each of these outcomes, a variety of threat actions are applicable. These threat actions include masquerade, resequencing, and repudiation. Masquerade is when a subject (e.g., host, user) pretends to be some other subject. Resequencing happens when the ordering of the data is changed. Repudiation is when a subject falsely denies having either sent or received data.

(This list of threats is derived from ISO 7498-2, *Informtion Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 2: System Architecture.*)



The general threats described previously can occur through the exploitation of vulnerabilities specific to a service. For any service, the following general sources could exploit service vulnerabilities. Within the service source code, malicious code could exist that implemented trojan horses, viruses, and worms. External to the code, malicious software within other applications, user processes, or system processes could take advantage of supplied features or a poor design within the service to exploit these vulnerabilities. Finally, errors such as those committed by system administrators, users, and the vendor of the service could result in the unintentional exploitation of a vulnerability.

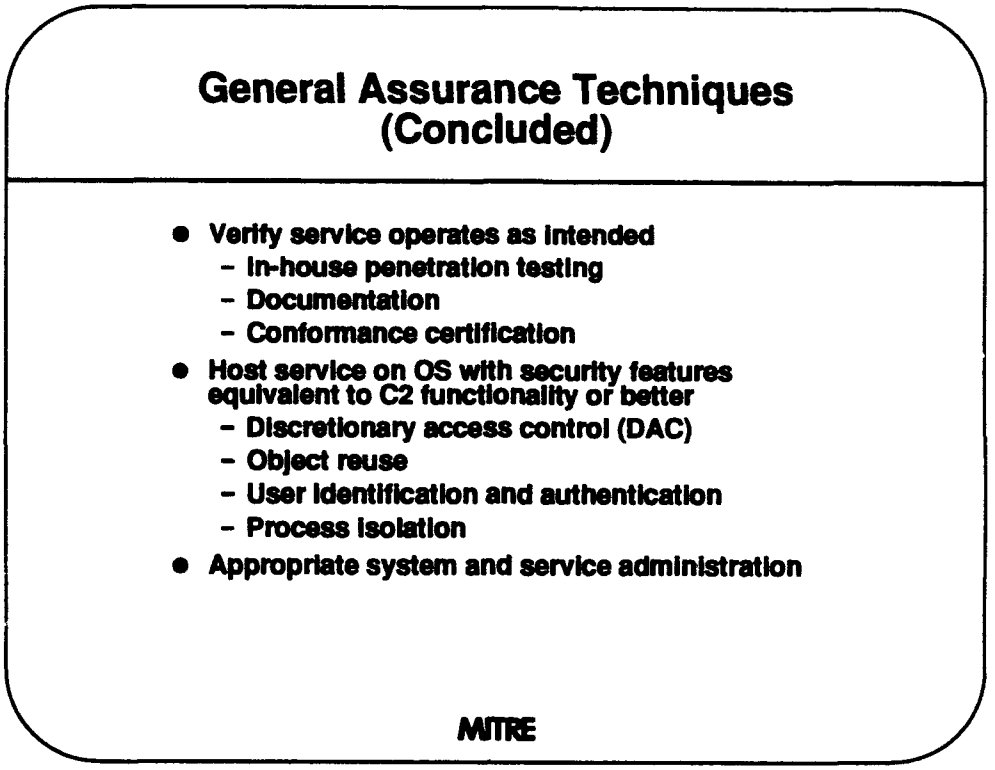


For vulnerabilities specific to a service, countermeasures can often be implemented to prevent or limit the chance of those vulnerabilities being exploited. For any service, a number of assurance techniques could be instituted that would ensure that countermeasures have been correctly implemented and cannot be easily circumvented and that no malicious code has been incorporated into the system.

One assurance technique is to use products evaluated by the National Security Agency or assessed by some other organization. If possible, source code should be obtained. To obtain source code, a license must be available for purchase from the vendor. The decision to obtain source code also depends on the cost of the code, whether or not the code is expected to undergo many revisions, and the reputation of the vendor. If little is known about the quality of the work produced by the vendor, inspection of the source code could give an indication of the quality.

In addition to quality, the source code (if obtained) should be inspected for several characteristics. The source code should be easily maintainable since properly maintained software enhances the secure operation of that software. For purposes of understanding the code, the code should be modular and well structured.

The need for the code to have special privileges granted by the operating system should be kept to a minimum since programs executing in a privileged state can effectively be used to exploit vulnerabilities. The service should only provide features that are related to that service to keep the code concise and understandable thereby reducing the risk of introducing security flaws. There should be no "back doors" in the code, such as master passwords and undocumented commands used for debugging purposes, since these back doors result in improper monitoring and access control of the user.



Another important assurance technique, and one that is critical in the absence of a formal evaluation or source code, is to verify that the service operates as intended. In-house, informal penetration testing can be done to perform this verification. It is important to note, however, that, without source code, penetration testing may not find all vulnerabilities.

Adequate documentation describing the operation of the service should be supplied be the vendor to aid in the verification process. This documentation is essential when source code is not available so that penetration testing can be effective. Formal conformance certification would also show that the service operates as intended.

Hosting the service on an operating system (OS) with security features equivalent to C2 functionality or above is another useful assurance technique. The C2 criteria and others are described in the *Trusted Computer System Evaluation Criteria* (TCSEC) published by NSA. The functionality within the C2 criteria include discretionary access control (DAC), object reuse, user identification and authentication, and process isolation. These are defined briefly as follows:

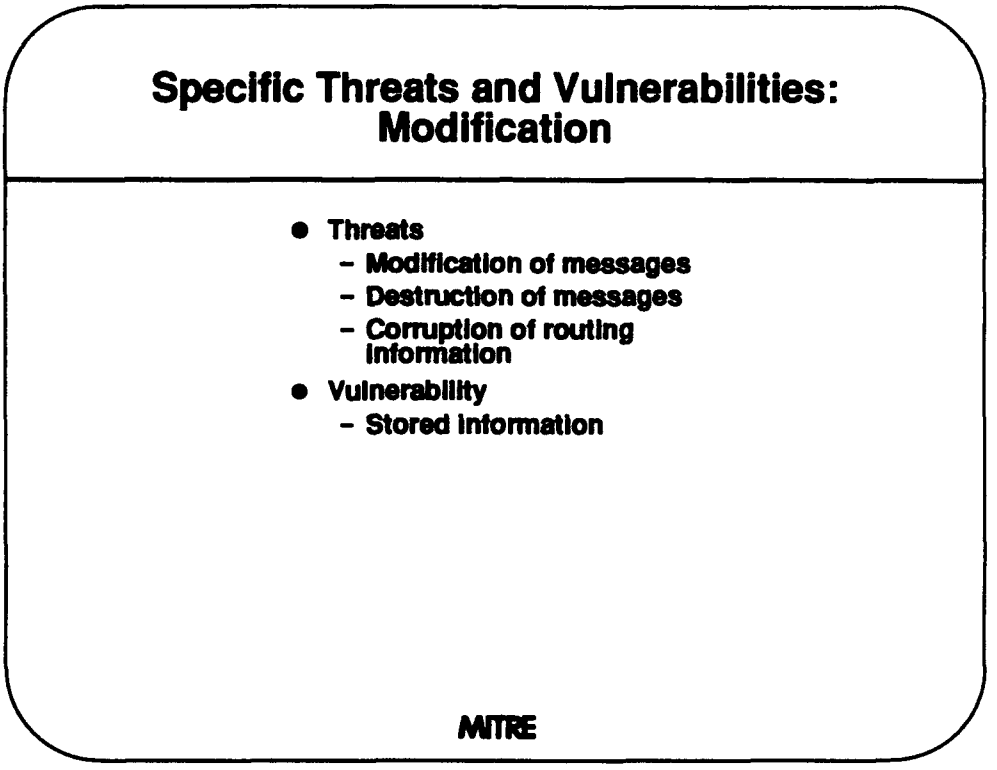
1. DAC involves read, write, and execute permissions granted to an individual, a group, and the public.

2. Object reuse guarantees that no information produced by a prior user is to be made available to another user that obtains access to an object that was released back to the system. For instance, when a user deletes a file, another user should not see information from the deleted file when the disk space is reallocated.

3. User identification and authentication requires that users must identify themselves to the OS, and that the OS must authenticate the users before any actions are performed on behalf of the users.

4. Process isolation requires that resources within the system be isolatable so that access control mechanisms can be used to protect them.

Once installed, the continued correct operation of the service should be ensured through careful administration of both the service and the system hosting the service. Included in this administration is the establishing and monitoring of configuration information relating to the host and to the service. This countermeasure is critical for boundary host systems since they are part of MITRE's security boundary.



Threats, vulnerabilities, and countermeasures specific to X.400 will now be identified.

One threat within the MHS environment is the threat of unauthorized modification. As messages travel through the MHS, there is the potential that they could be modified or destroyed.

In addition to messages being modified, routing information could be corrupted. Routing information is the information required to get the message from the originator to its destination. X.500 is responsible for maintaining this information. However, if X.500 is not in place, the information can be stored statically for use by X.400.

The method in which information is stored is one vulnerability that, if exploited, could lead to unauthorized modification. This vulnerability is discussed on the following slide.

Specific Vulnerabilities and Countermeasures: Stored Information

- **Description**
  - **Saved messages**
  - **Messages deferred for delivery**
  - **Messages held for delivery**
  - **Queued messages**
  - **Routing information**
- **Vulnerability**
  - **Stored information may not be adequately protected**
  - **Users may gain unauthorized access**
- **Countermeasure**
  - **Operating system capable of DAC**
  - **Stored information is an OS object owned by appropriate user**

MITRE

There are various types of stored messages within an MHS. These stored messages include the following:

1. Messages that users save in a message store,

2. Messages that the MTS is deferring delivery of to remote users until a certain time specified by the originator,

3. Messages that users have requested the MTS to hold for delivery to them until they are available to process the messages, and

4. Messages that are queued for delivery within the MTS.

If these stored messages are not protected adequately, users could modify them without proper authorization.

In addition to stored messages, routing and management information may also be stored. Again, if not adequately protected, this information could be modified without authorization.

To protect internally stored information, the operating system where the information is located must be capable of performing discretionary access control. Also, the information must be recognized as an object by the OS so that the OS can use the DAC mechanisms to protect the information. Each message must be associated with the appropriate owner. Saved messages should be owned by the recipient MTS-user. Deferred messages should be owned by the originating MTS-user. Held messages and queued messages should be owned by the MHS administrator.

A group of messages can be contained in one OS object if all messages are owned by one user. However, multiple messages owned by several users cannot be contained in one OS object since operating systems usually perform DAC at the object level.

MITRE cannot provide for the protection of externally stored information. Information received from external sources cannot be guaranteed to have not been modified, and the prevention of the modification of information sent externally cannot be ensured.

Specific Threats and Vulnerabilities: Disclosure

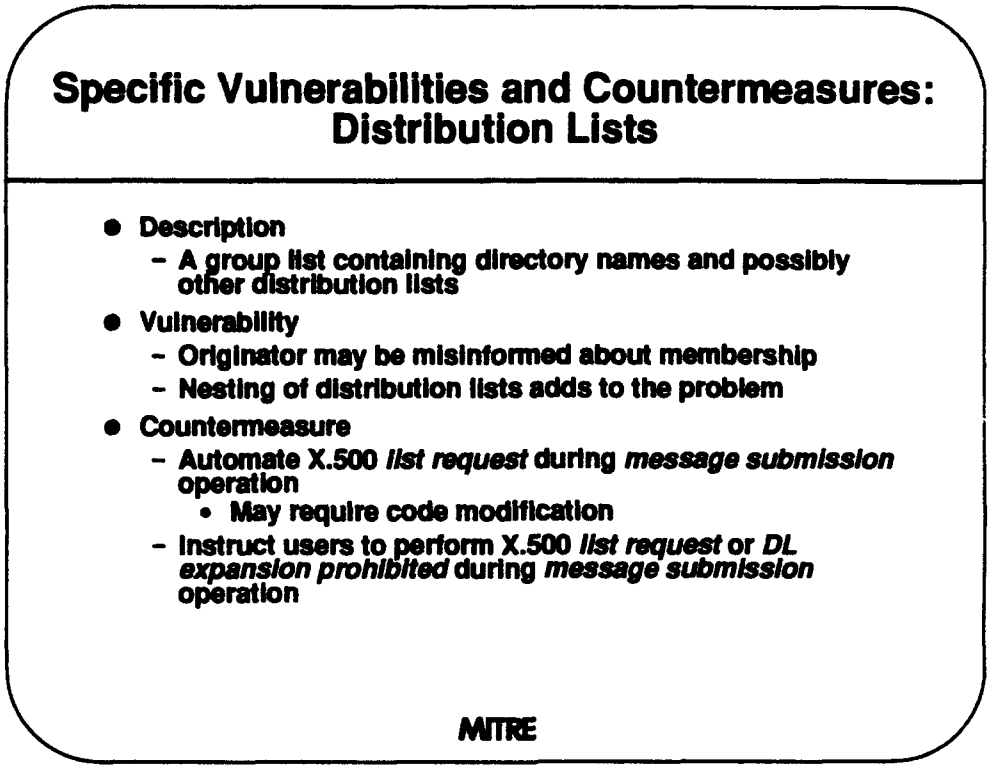
- Threats
  - Loss of confidentiality
  - Loss of privacy
  - Misappropriation of messages
  - Traffic analysis
- Vulnerabilities
  - Message storage
    - As described previously
  - Distribution lists
  - Alternate recipient allowed argument
  - Recipient reassignment allowed argument
  - Replies to messages with blind copy recipients

MITRE

As with unauthorized modification, there are also various forms of unauthorized disclosure. Loss of confidentiality occurs when the content of a message is captured and read by users for whom the message was not intended. By reading message header information that may not be protected, an MTS-user can detect who authored a particular message which would result in loss of privacy concerning authorship. Misappropriation occurs when messages are delivered to the wrong MTS-user, either through misuse or errors. Also, message traffic can be analyzed to ascertain information. (Pizza shops in the Pentagon area know important events are transpiring when the number of requests for delivery take a sharp increase.)

There are a number of vulnerabilities relating to unauthorized disclosure. As described previously, unauthorized access can be gained to stored messages to either read or modify the messages. Distribution lists can also result in unauthorized disclosure. Two arguments supplied during a message submission operation, *alternate recipient allowed* and *recipient reassignment allowed*, can result in a message being sent to a recipient without the knowledge of the originator. Finally, the manner in which replies are sent to messages that have blind copy recipients can result in unauthorized disclosure.

More detail on each of these vulnerabilities is given in the following slides.



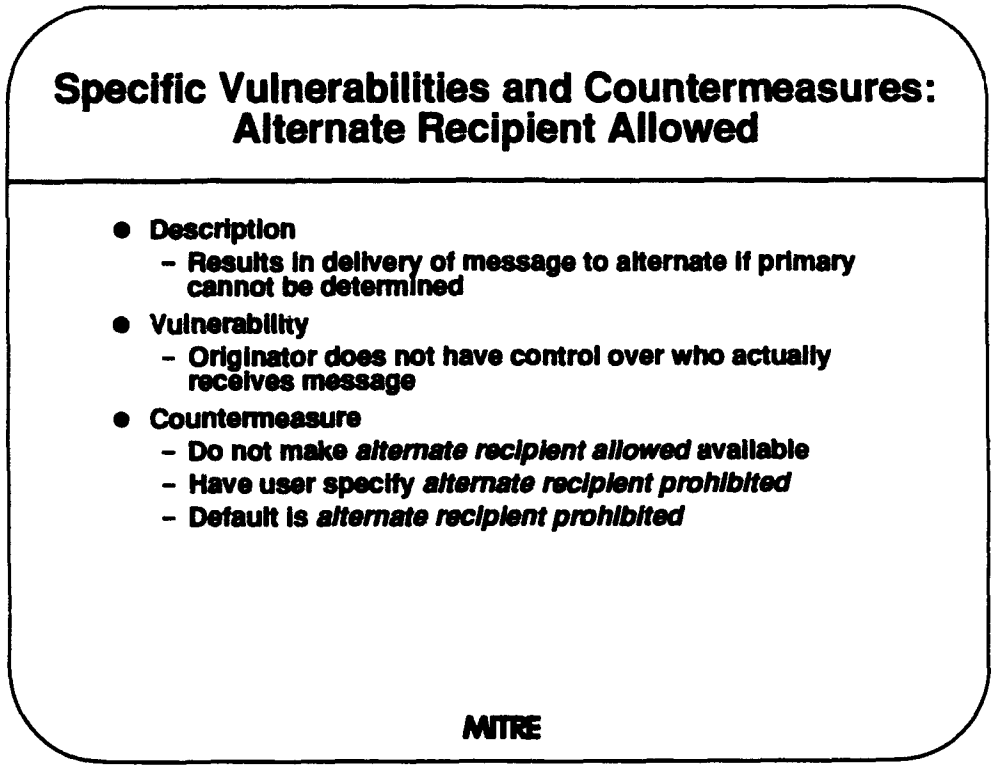
Distribution lists (DL), provided through X.500, are used to identify a group of people that have common interests so that messages can be easily sent to all individuals interested in a particular topic. A distribution list contains directory names and possibly other distribution lists.

Since distribution lists may undergo many changes and may be lengthy, the originator of a message using a distribution list may be misinformed as to the actual list membership. Nesting of distribution lists (a list within a list) adds to the confusion. If the originator is misinformed about the list membership, a message could be sent to unintended recipients.

One countermeasure to this problem would be to have DL membership automatically reported to the originator before the message is sent through the *list request* operation. However, in addition to requiring modification of OSI code, expanding the distribution list could be cumbersome for the user. As stated previously, distribution lists can be lengthy, and, when the message content is not sensitive in the originator's opinion, the originator may not want to see every name on the distribution list.

Another countermeasure is to simply instruct the originator to expand the list through a *list request* or prevent the delivery of messages that unknowingly contain a distribution list recipient with the *distribution list expansion prohibited* argument.

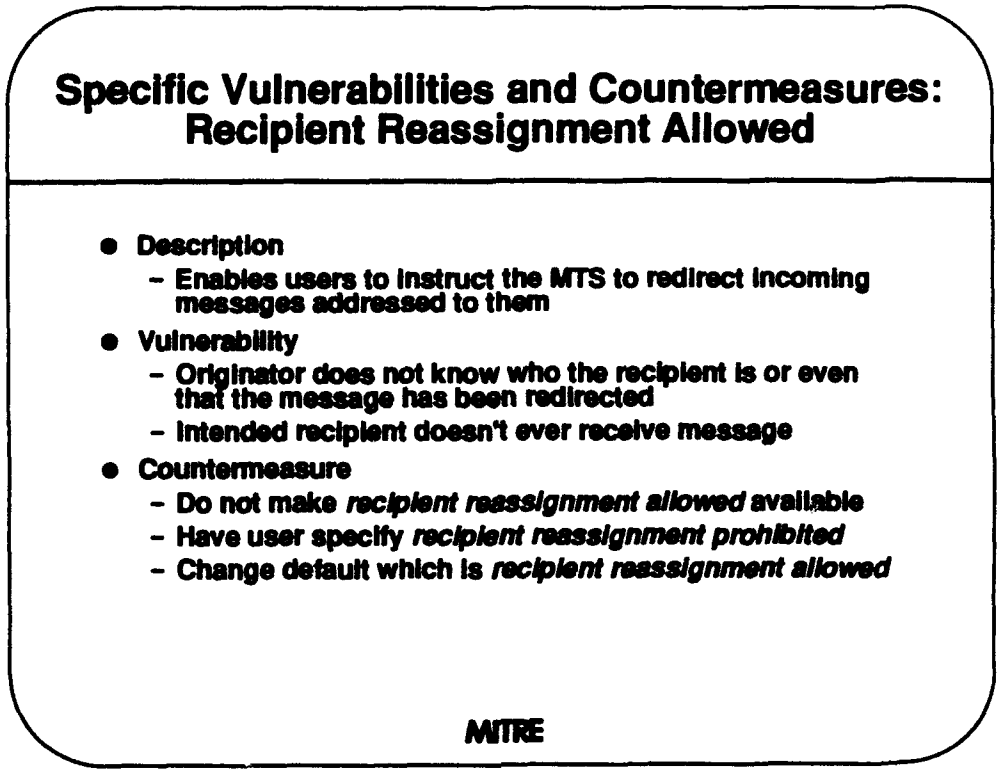
MITRE cannot instruct external originators to perform this countermeasure. Therefore, inbound mail received from an external source could be delivered to recipients within MITRE without the knowledge of that external source. However, this does not pose a security threat to MITRE data.



The alternate recipient feature allows a destination MTA to deliver a message to an alternate recipient designated by that MTA if the primary recipient cannot be determined from the information provided by the originator. For this feature to work, the *alternate recipient allowed* argument would have to be specified by the originator during message submission, and the destination MTA would have to have an alternate recipient designated. The problem with this feature is that the originator does not know who the alternate recipient is, and the originator may not want the alternate recipient to receive the information.

One possible countermeasure is to not allow the originator to supply the *alternate recipient allowed* argument by having the originating MTA automatically override this argument with the *alternate recipient prohibited* argument. If automation is not possible or desired (since at times it is beneficial to have an alternate recipient for critical messages), users could be asked to set the *alternate recipient prohibited* argument during message submission. The default specified in the recommendation is to prohibit an alternate recipient.

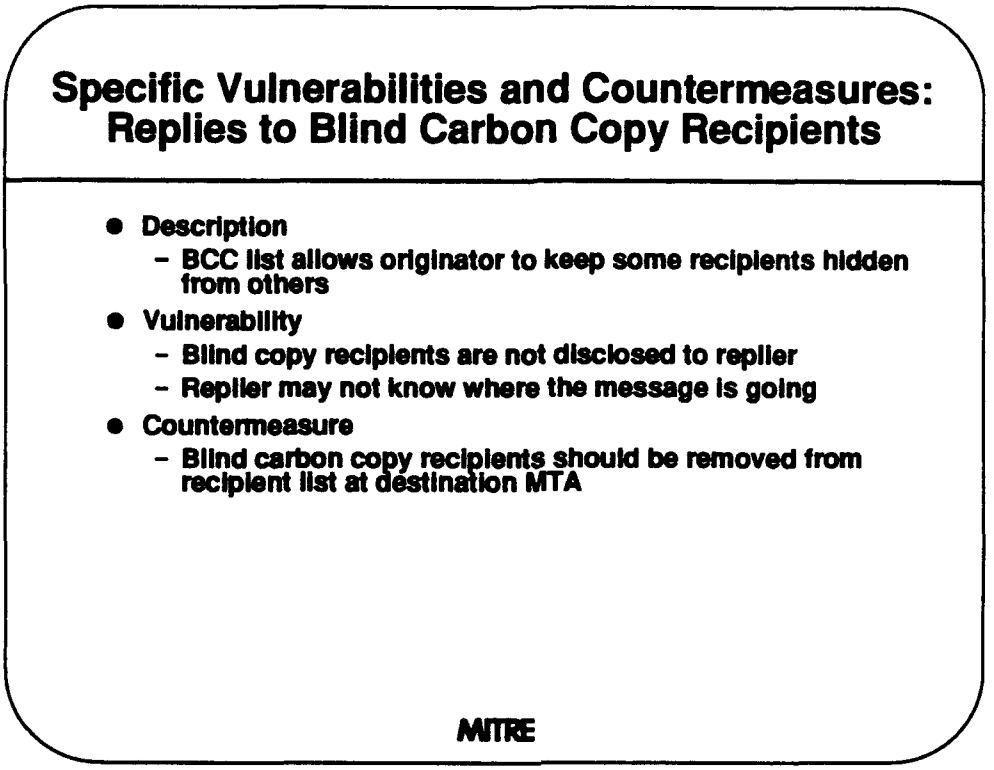
MITRE cannot instruct external originators to perform this countermeasure. Therefore, inbound mail received from an external source could be delivered to an alternate recipient within MITRE without the knowledge of that external source. However, this does not pose a security threat to MITRE data.



The recipient reassignment feature allows users to instruct the MTS to redirect incoming messages addressed to them. The intended recipient specifies to whom the messages are to be redirected, without the knowledge or approval of the originator. With this feature, the intended recipient never receives the message. For this feature to work, the *recipient reassignment allowed* argument would have to be specified by the originator during message submission, and the intended recipient would have to have an alternate recipient designated.

As with the alternate recipient feature, one possible countermeasure is to not allow the originator to supply the *recipient reassignment allowed* argument by having the originating MTA automatically override this argument with the *recipient reassignment prohibited* argument. If automation is not possible or desired, users could be asked to set the *recipient reassignment prohibited* argument during message submission. The default specified in the recommendation is to allow recipient reassignment. This default should be changed so that this feature is not invoked without the originator specifically allowing it.

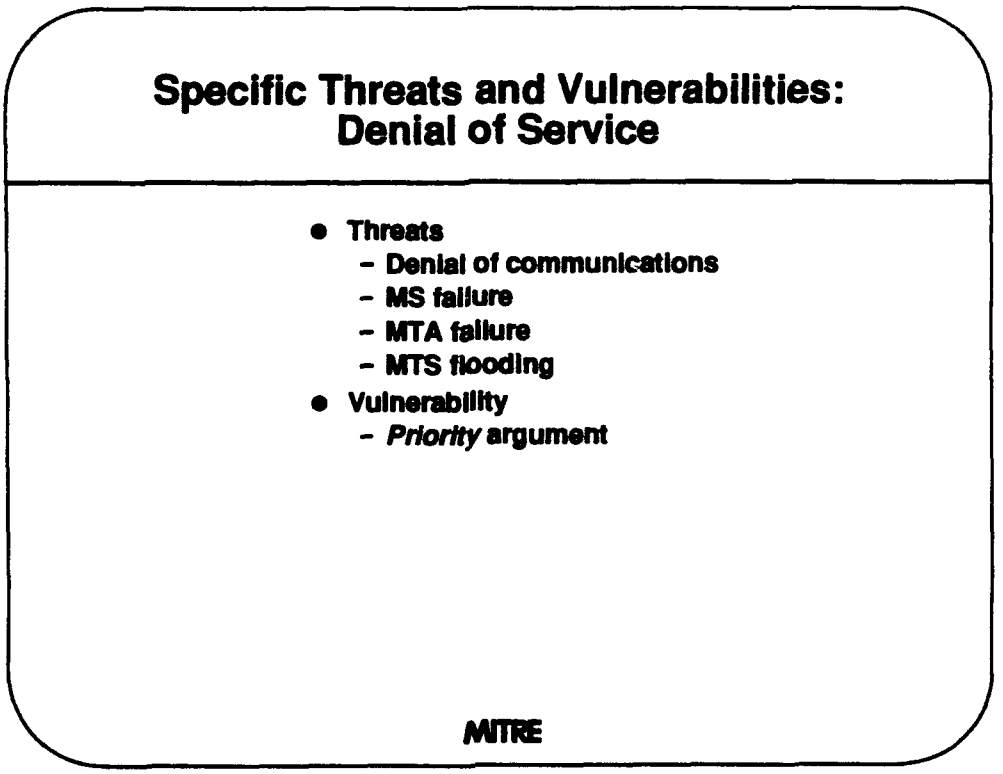
MITRE cannot instruct external originators to perform this countermeasure. Therefore, inbound mail received from an external source could be redirected within MITRE without the knowledge of that external source. However, this does not pose a security threat to MITRE data.



The blind carbon copy (BCC) feature allows a user to specify recipients of a message that direct recipients and carbon copy recipients of the message do not see. The concern with this feature involves replies to messages that have BCC recipients. A user can globally reply to a message and have the reply automatically sent to the originator and all recipients of the message. The recommendation does not state that BCC recipients should not receive any replies to a message. Therefore, depending on the implementation, a reply could be sent to someone without the knowledge of the replier.

The countermeasure to this problem is to verify that the implementation removes the BCC list at each destination MTA.

MITRE cannot instruct external MTAs to perform this countermeasure. Therefore, blind carbon copy recipients within MITRE may receive replies to messages without the knowledge of the external replier. However, this does not pose a security threat to MITRE data.



Denial of service could be achieved through a breakdown in the network, the failure of an MS or MTA, or the flooding of the MTS with messages. Any of these problems results in the inability to deliver messages.

One vulnerability that can be exploited to cause a denial of service concerns the user's ability to specify the priority of a message. This is discussed on the following slide.

Specific Vulnerabilities and Countermeasures: *Priority* Argument

- **Description**
  - **User requests urgent, normal, or nonurgent**
  - **Different from importance indication**
- **Vulnerability**
  - **Modifies time periods**
- **Countermeasure**
  - **Do not allow users this privilege**
  - **Establish threshold**

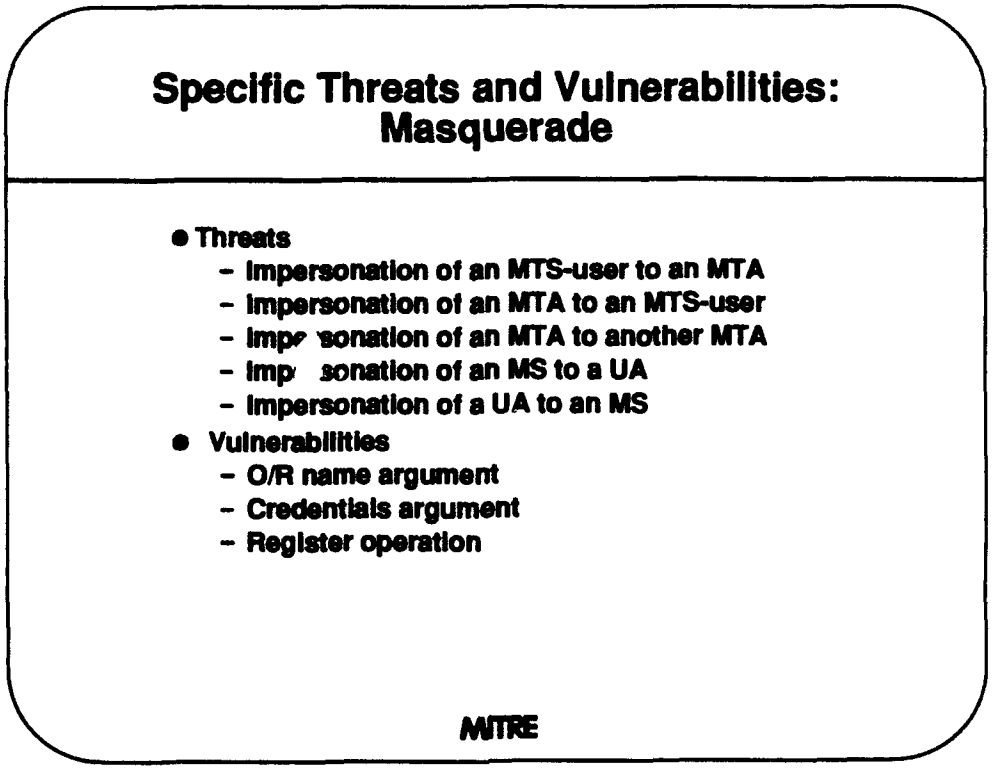
MITRE

Denial of service can be achieved through users flooding the MTS with messages. The ability of users to specify the priority of their messages increases the rate at which flooding could occur.

The priority of a message can be either urgent, normal, or nonurgent. This is different from an importance indication which informs the recipient whether or not the originator considers the message an important one that should be read as soon as possible. In terms of the quality of service that the MTS must provide, an urgent message has a shorter period of time in which it must be processed by the MTS than a normal or nonurgent message. Therefore, an urgent message may be processed more quickly than a normal or nonurgent message.

Since the MTS may process urgent messages more quickly than other messages, a user could flood the MTS with urgent messages and delay the processing of normal or nonurgent messages.

As a countermeasure to this problem, the privilege to set the priority on a message should only be granted to a system administrator or MHS administrator. A standard MHS-user should not be granted this privilege. To prevent external MHS-users or MTAs from flooding MITRE's internal MHS with urgent messages, a threshold on the number of urgent messages processed could be established. Once this threshold was reached, the boundary MTA could reset the priority of the messages to normal.



There are five forms of masquerade that could take place: impersonation of an MTS-user to an MTA, impersonation of an MTA to an MTS-user, impersonation of an MTA to another MTA, impersonation of an MS to a UA, and impersonation of a UA to an MS.

There are three specific vulnerabilities that pose a threat of masquerade. These vulnerabilities relate to the originator/recipient (O/R) name supplied with many operations, the credentials given during a bind, and the register operation. These are discussed in more detail on the following slides.

Specific Vulnerabilities and Countermeasures: O/R Name

- Description
  - Contains O/R address and/or directory name
  - If both are present
    - O/R address is used
    - Directory name is ignored but passed on to receiver
- Vulnerability
  - Sender is able to supply false directory name
  - Receiver is more likely to consult directory name
- Countermeasure
  - Have originating and destination MTA resolve directory name

MITRE

An O/R name comprises a directory name, an O/R address, or both. A directory name is intended to be a user-friendly name that can be easily associated with a particular user. The directory name can be used to determine an O/R address by performing a look-up in the X.500 directory. An O/R address contains information that enables the MHS to uniquely identify users and to route messages or return notifications to them.

When both an O/R address and a directory name are given as part of an O/R name, the MHS will use the O/R address but will carry the directory name and present both to the recipient. This presents the opportunity for the sender to supply a false directory name with the intention of deceiving the receiver as to who actually sent the message. When the message is delivered to the receiver, the receiver is more likely to consult the user-friendly directory name than the O/R address. The receiver could then respond to the message thinking that the response is going to the user associated with the directory name rather than the user associated with the O/R address.

The countermeasure to this vulnerability is to have the destination MTA and, for added security, the originating MTA resolve the directory name and compare the result with the O/R address. If the O/R address did not match the directory name, the message should be discarded or the directory name should be changed to match the O/R address.

This countermeasure is not currently possible, however, when messages are generated externally. MITRE does not currently have a method to verify O/R addresses and directory names that are external to MITRE. Eventually, distributed directory services will be more mature, and MITRE will be able to perform this countermeasure for externally generated messages. Until that time, users within MITRE should take care with any information that is received from outside of the company.

Specific Vulnerabilities and Countermeasures: Credentials

- Description
  - Contain password or token
  - Presence is required, but authentication is not
- Vulnerability
  - Sender can supply any O/R name
  - Receiver must believe supplied O/R name
- Countermeasure
  - Perform local authentication

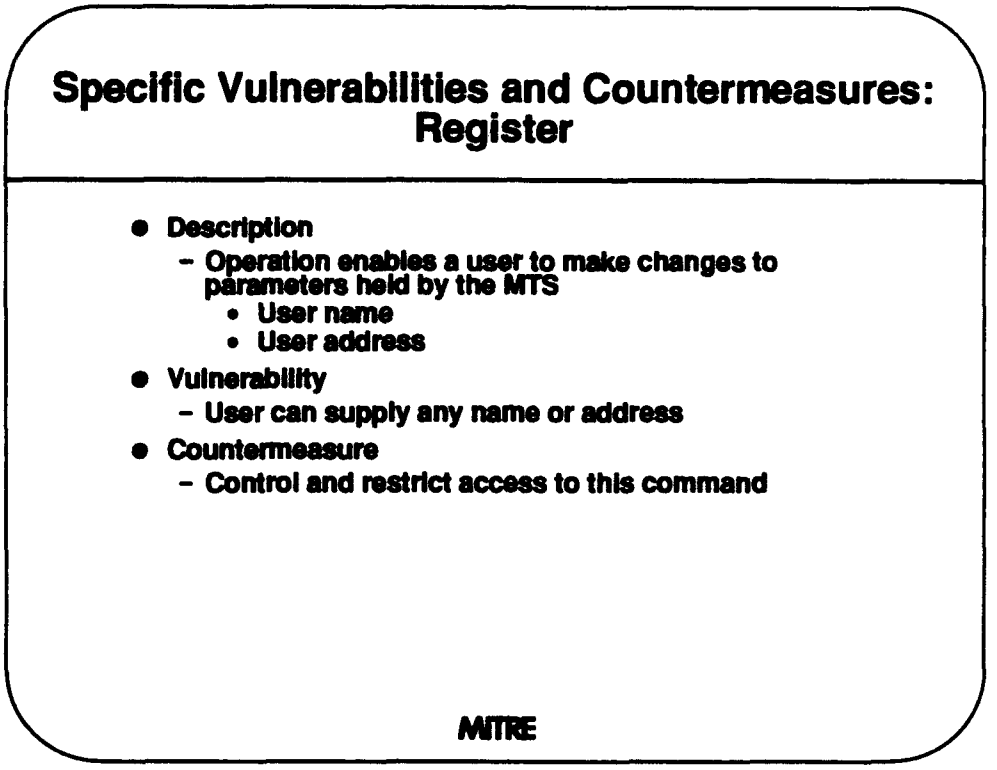
MITRE

When binding to the MTS, credentials must be supplied by the initiator. If simple authentication is used, the credentials contain a password. If strong authentication is used, the credentials contain a token and, optionally, a certificate.

Although the credentials must be supplied by the initiator, the responder is not required to perform any authentication using these credentials. If no authentication is performed, the initiator can supply any credentials.

A countermeasure to false credentials is to have a valid authentication scheme resident on all MTAs within MITRE.

MITRE cannot instruct external MTAs to perform this countermeasure. Therefore, a MITRE user may be able to gain access to an external MHS by providing false credentials. However, this does not pose a security threat to MITRE data.



The register operation allows an MTS-user to make long-term changes to various parameters of the MTS-user held by the MTS concerned with delivery of messages to the MTS-user. Two of these parameters are the user name and the user address. The user name is the O/R name, and the user address is either the X.121 address, the transport service access point (TSAP) identifier, or the presentation service access point (PSAP) address. The recommendation does not specify any restrictions concerning the use of this operation, therefore, an MTS-user can supply any name and any address. Depending on how the MTS uses this information, other MTS-users or the MTS itself could be deceived as to who the actual user is. Access to this command needs to be restricted to system or mail administrators. Alternatively, users can be allowed restricted write access to their own entry. Under this restricted access, users can modify only those attributes that have been designated as being changeable by standard users.

MITRE cannot instruct external MTAs to perform this countermeasure. Therefore, MITRE users may be able to change their register information held by an external MTS. However, this does not pose a security threat to MITRE data.

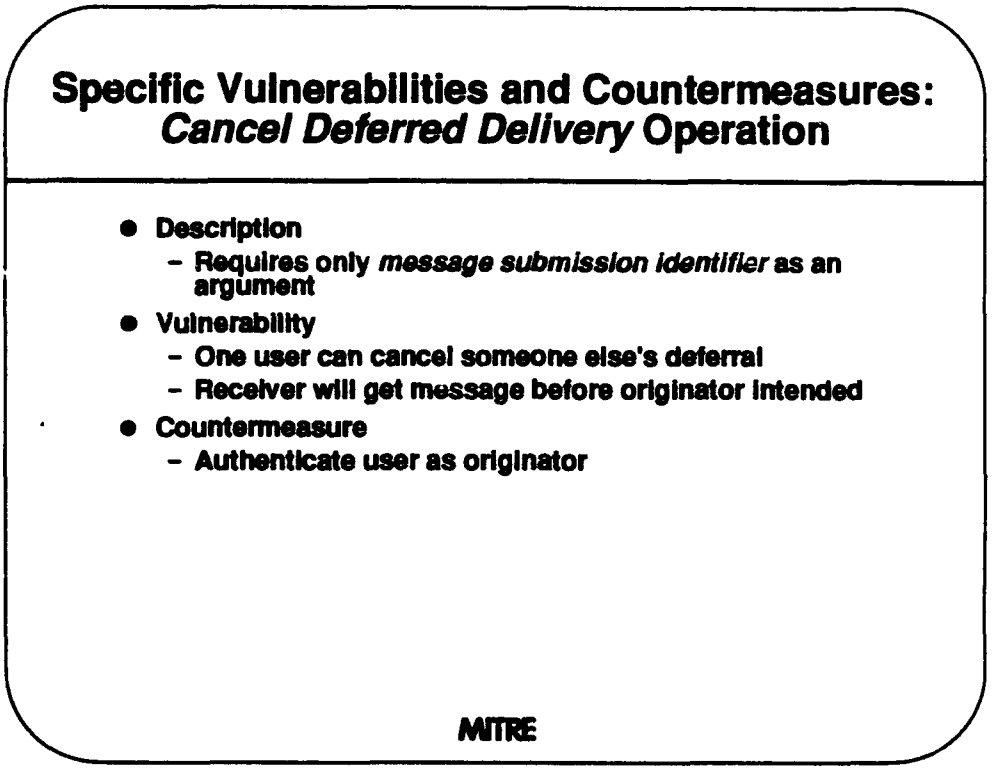
Specific Threats and Vulnerabilities: Resequencing

- **Threats**
  - **Replay of messages**
  - **Reordering of messages**
  - **Preplay of messages**
  - **Delay of messages**
- **Vulnerability**
  - ***Cancel Deferred Delivery* operation**

MITRE

In terms of resequencing, messages can be replayed, reordered, preplayed, or delayed. Any of these resequencings could cause confusion or result in information arriving too late or too early.

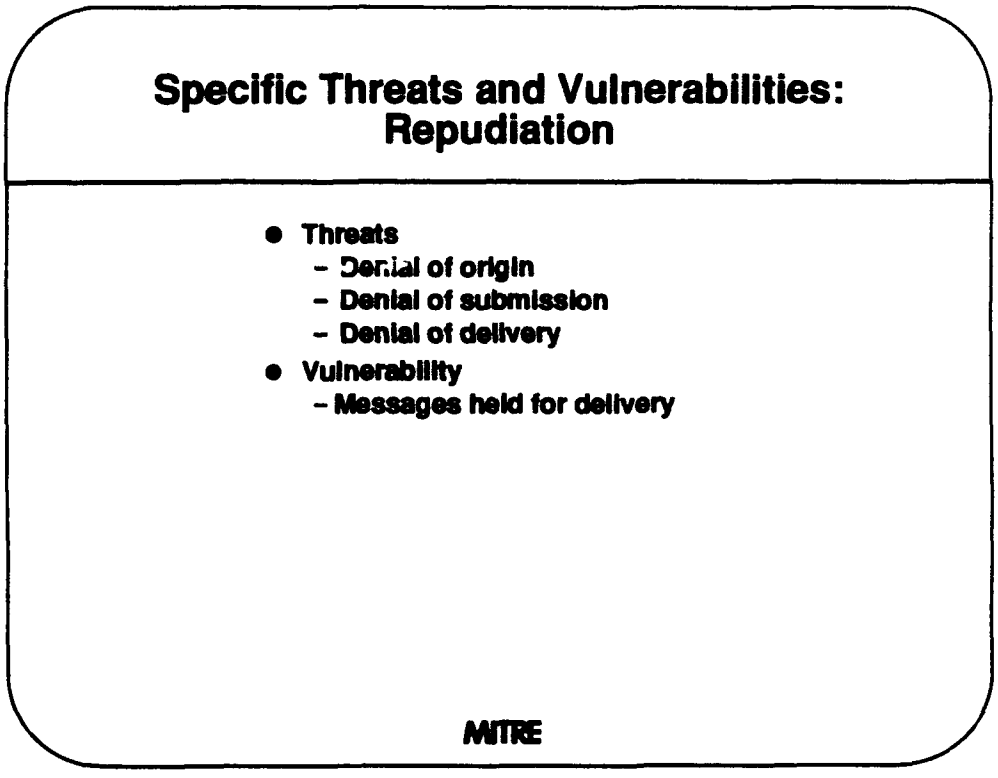
As described on the next slide, cancelling a deferred delivery could cause preplay of messages.



The deferred delivery feature allows an originator to submit a message to an MTS but request that the MTS not deliver the message to the intended recipient until a specific time. As a complement to this feature, the *cancel deferred delivery* operation allows a user to cancel the delay time associated with the delivery of a deferred message and have the message delivered immediately. However, the only argument that a user must supply to perform a cancellation is a message submission identifier. Therefore, one user could supply any message submission identifier and cancel another user's deferral resulting in the delivery of a message earlier than intended by the originator.

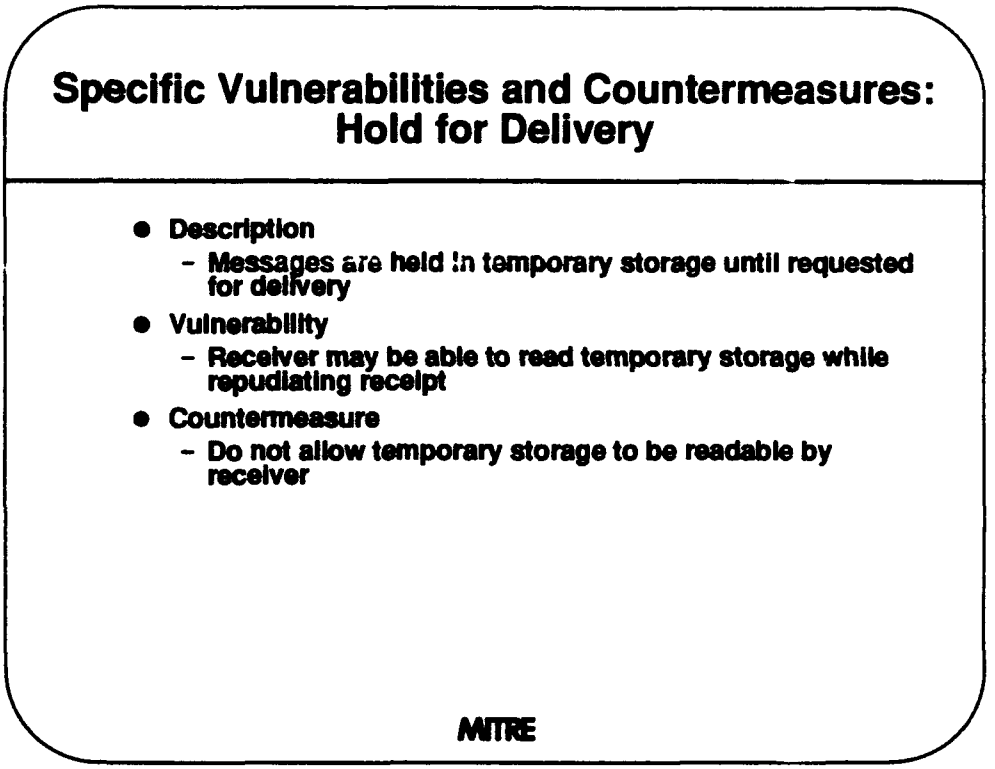
As a countermeasure to this vulnerability, the MTS should authenticate that the user performing the cancellation of the deferred delivery time is the originator of the deferred message.

MITRE cannot instruct external MTAs to perform this countermeasure. Therefore, a MITRE user could gain access to an external MHS and cancel the deferred delivery of an external user. However, this does not pose a security threat to MITRE data.



Repudiation could take three forms within an MHS. The author could deny having originated the message (denial of origin), the MTS could deny having received the message from the originator (denial of submission), and the recipient could deny having received the message from the MTS (denial of delivery).

The method in which messages that are held for delivery are protected could result in a user being able to deny having been delivered a message. This vulnerability is described on the following slide.



Along with unauthorized disclosure and modification, messages that the MTS is holding for delivery can result in false repudiation of receipt. If the temporary storage where the messages are located is not adequately protected, users can gain access to the storage, read the messages that are being held for them, and then repudiate having received them since the messages were never actually delivered to the users.

The countermeasure to this repudiation problem is to provide proper discretionary access control on the temporary storage, as should be done with the disclosure and modification threat. These held messages should be owned by the system or mail administrator and should be readable by only these administrators.

Assurance Techniques: Essential (E) or Desired (D)

- The following assurance techniques are desirable, but not essential, since penetration testing can be used to verify implementation
  - Evaluated or assessed product: D
  - Acquisition of source code: D
  - Source code that is modular and concise: D
  - Source code that is easily maintainable: D
  - Enforcement of least privilege: D
  - Absence of extraneous features: D
- Absence of "back doors" E
  - If source code is available, a determination should be made that no back doors exist

MITRE

That concludes the discussion of identified threats, vulnerabilities, and countermeasures specific to X.400. Those threats, vulnerabilities, and countermeasures identified are intended to be inclusive given the scope of the analysis. However, additional issues may be identified as penetration testing and evaluations of specific implementations proceed.

The general assurance techniques and specific countermeasures will now be examined in terms of whether they are essential to any implementation integrated into MITRE networks or whether they are desired, but not essential. The criticality of each will be discussed in the order that they were previously discussed. These next several slides are summarized towards the end of the briefing.

Many of the general assurance techniques work in concert to ensure that the service operates as intended. The primary assurance technique that ensures this is penetration testing. Therefore, general assurance techniques, such as the use of evaluated or assessed products, the acquisition of source code, modular and concise source code, maintainable source code, the enforcement of least privilege, and the absence of extraneous features, are all desirable assurance techniques, but are not essential.

The absence of back doors into the service is essential to ensure correct operation of the service. If source code is available, the code must be inspected for these back doors, and the back doors must be removed, if present.

Assurance Techniques: Essential (E) or Desired (D) (Concluded)

- Penetration testing: E
  - Without source code, only method of determining if implementation operates as intended
- Documentation: E
  - Aids in penetration testing
- Conformance certification: D
  - Conformance can be indicated through penetration testing
- OS security: E
  - To protect the operation of the service, the OS must provide C2 functionality
- System and service administration: E

MITRE

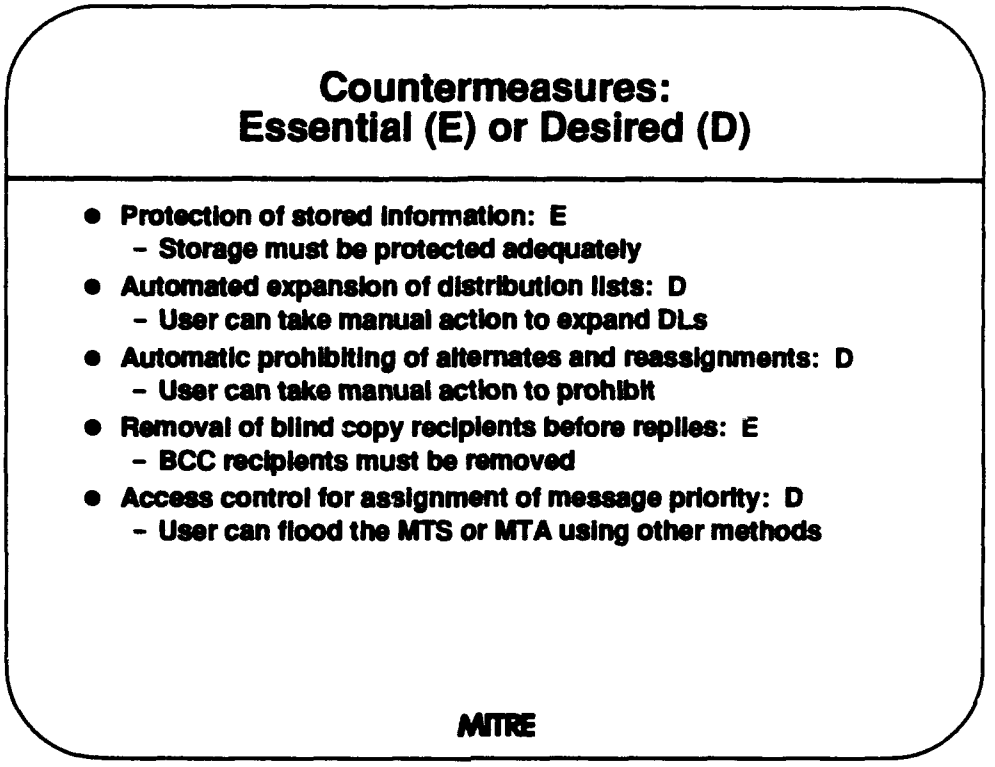
Penetration testing, as implied earlier, is essential to verifying that the service operates as intended. Without source code, penetration testing is the only method of making this determination.

As an aid to effective penetration testing, documentation is also essential.

Conformance certification, though desirable, may not be practical to require. Penetration testing can provide an indication as to how well the implementation conforms to the recommendation.

To protect the proper operation of the service, OS security is essential. At a minimum, the functionality specified within the C2 requirements of the TCSEC must be provided by the OS.

To guarantee the continued correct operation of the service, both the service and the system hosting the service must be carefully administered. This includes correctly establishing and periodically inspecting the configuration of the host and the service.



The protection of stored information is an essential countermeasure to prevent unauthorized modification and unauthorized disclosure.

Some countermeasures need not be automated since the user can take a manual action to provide the countermeasure. For instance, the user can manually expand distribution lists, prohibit alternate recipients, and prohibit recipient reassignments.

To prevent unauthorized disclosure, blind carbon copy recipients should be hidden from other recipients in every respect. Therefore, the removal of the BCC list as soon as possible is an essential countermeasure.

Since the risk of flooding the MTS or MTA with high priority messages is not much greater than the risk of flooding the MTS or MTA with normal priority messages, establishing access controls on the use of the priority argument is desirable, but not essential.

Countermeasures: Essential (E) or Desired (D) (Concluded)

- Resolution of directory name with O/R address: E
  - Resolution must be performed automatically
- Authentication of credentials: E
  - Users must be authenticated
- Access control for register operation: E
  - Access to register information must be controlled
- Authentication of deferred delivery cancellation: E
  - Users should only be able to cancel their own deferred delivery times
- Protection of held messages: E
  - To prevent repudiation, held messages must be protected from intended recipient

MITRE

Other essential countermeasures include the resolution of the directory name with the O/R address supplied in the O/R name, the authentication of credentials, and access control to the register operation. These countermeasures are critical to the prevention of masquerading performed to achieve unauthorized disclosure and unauthorized modification.

To prevent resequencing problems, the cancellation of a deferred delivery must be authenticated.

To prevent repudiation, held messages must be protected from the intended recipient.

Location of X.400 Specific Countermeasures

Countermeasure	Internal MTA	Boundary MTA	Internal UA	Internal MS
Stored information protection	E	E	E	E
Distribution list expansion			D	
No alternate recipient			D	
No recipient reassignment			D	
Removal of BCC list	E	E		
Priority access control	D	D		
Resolution of O/R name	E	E		
Authentication	*	*	*	*
Register access control	E	E		
Authenticate cancellation	E	E		

* See next slide

MITRE

This table shows the MHS component on which each countermeasure is implemented. MTAs are responsible for the protection of stored information, removal of BCC list, resolution of O/R names, authentication of credentials, access control to the register operation, access control in the setting of the priority argument, and authentication of deferred delivery cancellation.

User agents are responsible for protection of locally stored information, distribution list expansion, prohibition of alternate recipients, and prohibition of recipient reassignment.

The message store is responsible for protection of locally stored information.

The responsibility for authentication is distributed among the components, and the type of authentication required depends upon the connection being made. This is discussed in the next slide.

Authentication Requirements

Responder / Initiator	Internal UA	Internal MS	Internal MTA	Boundary MTA	External MTA
Internal UA	-	password	password	password	-
Internal MS	static password	-	static password	static password	-
Internal MTA	static password	static password	static password	static password	-
Boundary MTA	static password	static password	static password	static password	static password
External MTA	-	-	-	static password	*
External UA	-	*strong*	-	*strong*	*

MITRE

This table shows the type of authentication that is required when one MHS component connects to another. The dashes represent connections that cannot be made. The asterisks represent connections that are outside the scope of this briefing.

When a user agent connects to an MS or an MTA, password authentication is required.

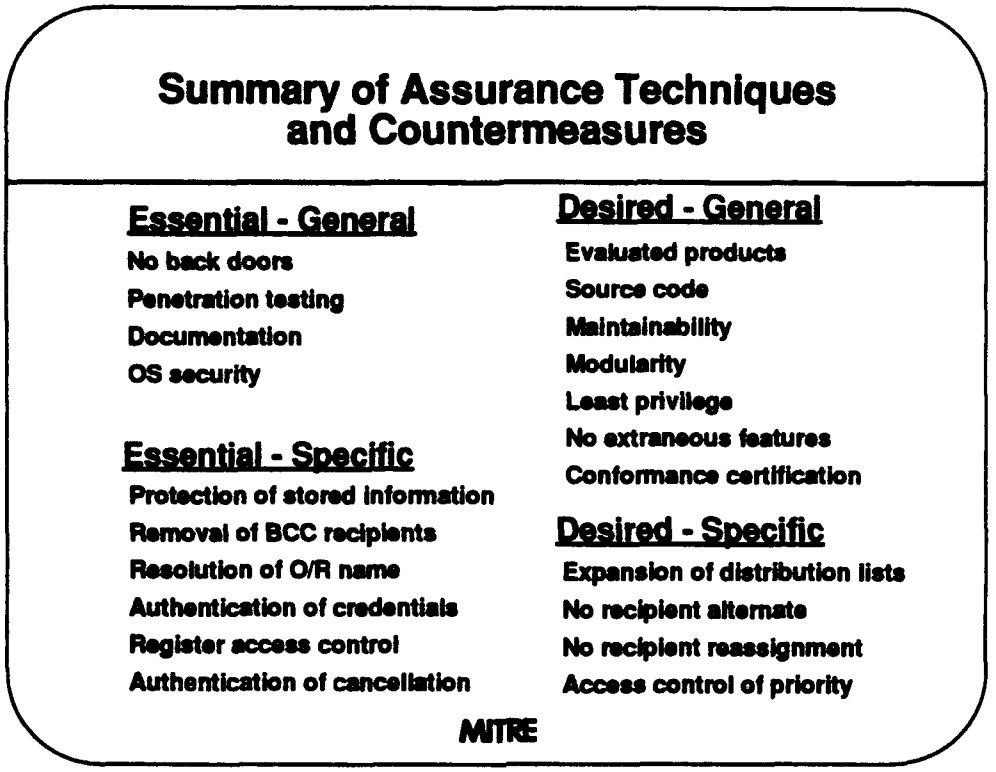
When an MS connects to a UA, MS, or MTA, credentials that are stored on the components are exchanged. This is referred to as a static password exchange since the credentials are stored and not dynamically entered by a user.

When an MTA, internal or boundary, connects to a UA, an MS, or another MTA, static password authentication is required.

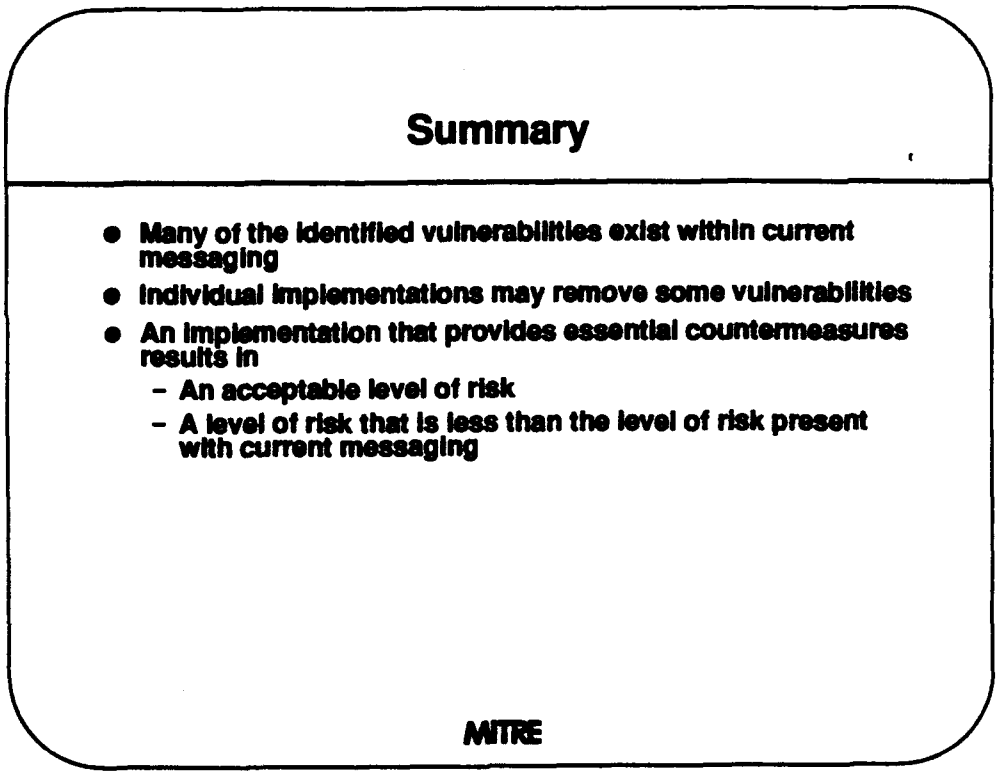
When an external MTA connects to a boundary MTA, static password authentication is required.

Currently, an external user agent cannot directly connect to an internal MS or boundary MTA. There is a requirement to be able to access mail remotely, however, and we may want to enable these connections in the future. The chart depicts that if these connections were to be allowed they would require strong authentication.

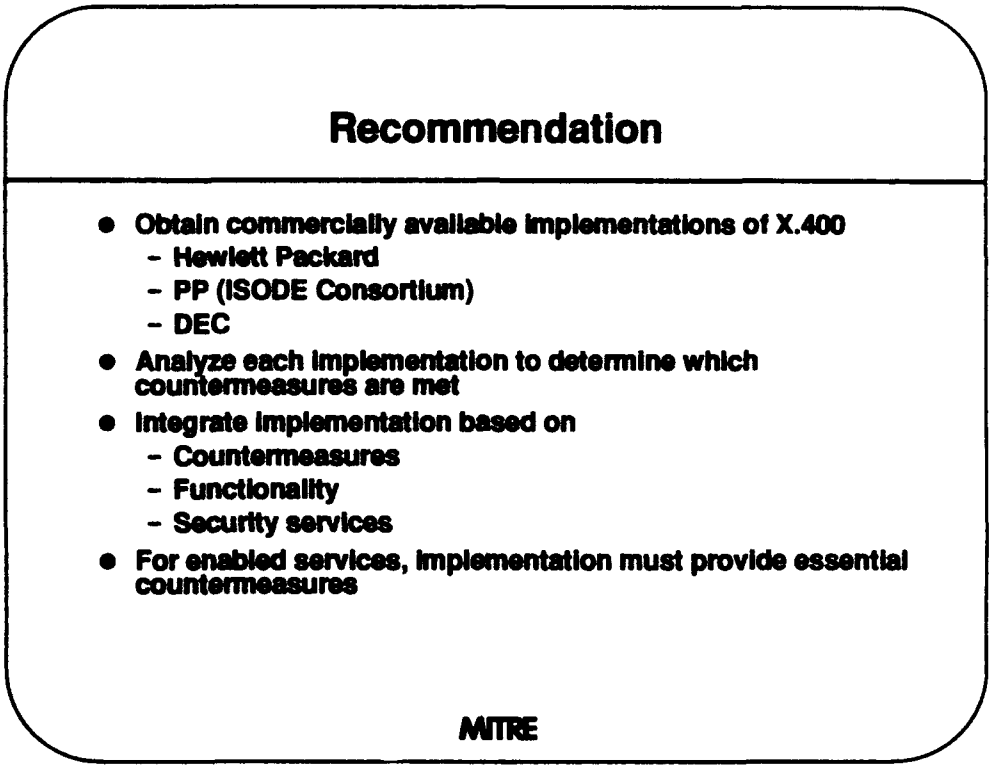
This chart also depicts connections that may not be permitted at MITRE. In particular, boundary MTAs will probably only serve as relay MTAs with no end-users being served directly by the boundary MTA. Therefore, direct connections between a boundary MTA and an internal UA or internal MS may not be allowed.



This chart simply presents a breakdown of the assurance techniques and countermeasures according to whether they are essential or desired and general or specific.



Now that these vulnerabilities have been described, it is important to note that many of these vulnerabilities exist within our current method of messaging. Also, individual implementations may address and remove many of these vulnerabilities. Therefore, an implementation that provides all essential countermeasures not only results in an acceptable level of risk when integrating it into the MITRE network, but also results in a level of risk that is less than the level of risk present with current messaging.



For the MITRE OSI Integration Project, the recommendations are the following: First, commercially available implementations of X.400 (based on the 1988 recommendation), such as implementations from Hewlett Packard, ISODE Consortium, and DEC, should be obtained and analyzed. Each implementation should be analyzed to determine which countermeasures are met. An implementation should be selected for integration into the MITRE network based on the countermeasures and functionality provided by that implementation. If two or more implementations are very close in terms of countermeasures and functionality, a third criteria for selection could be security services provided since these may prove useful and may indicate how concerned the vendor was with security.

If an essential countermeasure is not present in the selected implementation, then the functionality or service resulting in a threat that requires the countermeasure must be disabled and made unavailable. For all enabled services, the implementation that is integrated into the MITRE network must provide the essential countermeasures specified in this briefing.

LIST OF REFERENCES

DOD 5200.28-STD, December 1985, *Department of Defense Trusted Computer System Evaluation Criteria.*

The International Telegraph and Telephone Consultative Committee, November 1988, *Data Communication Networks Message Handling Systems, Recommendations X.400 - X.420.*

ISO-7498-2, 15 February 1989, *Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 2: System Architecture.*