CA Top Secret® for z/OS

Multilevel Security Planning Guide

r15

Second Edition

CA Technologies Product References

This documentation set references the following CA products:

- CA ACF2™ for z/OS (CA ACF2)
- CA Common Services for z/OS (CA Common Services)
- CA Distributed Security Integration Server for z/OS (CA DSI Server)
- CA LDAP Server for z/OS (CA LDAP Server)
- CA Top Secret® for z/OS (CA Top Secret)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Documentation Changes

The following documentation updates have been made since the last release of this documentation:

- User SECLABELS—Added replace to command format.

Contents

Chapter 1: Multi Level Secure System Elements

This section contains the following topics:

Features of CA Top Secret MLS

CA Top Secret supports MLS, a security policy that prevents disclosure and declassification of data based on defined levels of sensitivity of data and levels of clearance of users to that data. MLS also provides protection mechanisms based on data ownership rules and access permissions, individual accountability, file reuse protection, and audit trails. Together, these mechanisms support segregation of data by function, by system, or by row (for databases) as part of protecting disclosure and declassification of data. CA Top Secret MLS also supports making security decisions based on security labels for UNIX files and directories and their names, TCP/IP connections, servers, and DB2 resources.

CA Top Secret provides the following MLS features:

- Allows selective labeling of users and resources
- Validates accesses based on mandatory access control (MAC) and discretionary access control (DAC) protection mechanisms
- Allows separation of MLS administration from DAC administration
- Allows phased-in MLS implementation
- Allows real-time monitoring of classified users and resources
- Audits and logs accesses and violations based on security classifications and resource and access rules
- Supports labeling of UNIX resources
- Supports requiring security labels for UNIX resources
- Allows or prevents write-down of data
- Supports labeling of IPv6 addresses
- Allows or restricts READ access to UNIX file and directory names
- Supports restricting security labels to specific systems

CA Top Secret MLS does not:

- Require security label classification of all users, data and resources in a system
- Always prohibit writing data from a higher classification to data of a lower classification (“write down” or declassification)

Vulnerabilities

There are three kinds of vulnerabilities that can affect the security of an MLS system:

Unauthorized changes to the system

Unauthorized changes to the system include both software and hardware changes and can occur during code development, system distribution, and local maintenance. During the development phase, a disgruntled employee could intentionally place Trojan horses in the code or in the development tools. Source code reviews and tool reviews could reduce the likelihood of undiscovered Trojan horses. System developers should also ensure that, throughout development, strict controls monitor changes to the code. It is imperative that changes to the code not take place after the system is tested and approved.

During system distribution and local maintenance, unauthorized changes can occur if untrusted hardware and software components are substituted for the trusted computing base (TCB) hardware and software. Countermeasures to deter unauthorized modification include distribution controls, thorough tests of all maintenance, and strict controls that monitor changes to the original code. All personnel who are involved in this stage must be trusted individuals. The system manufacturer and the customer must play an active role to combat unauthorized changes during distribution and maintenance.

Assuming the identity of another user

An unauthorized user who assumes the identity of a trusted security administrator is a threat to security. The likelihood of an unauthorized user assuming the role of an authorized user is greatly reduced when the system provides individual authentication and password protection. The potential for abuse is lessened even more when users take seriously the policies regarding password use and the policies are enforced by the system and by the security administrators. Unauthorized users could also assume the role of authorized users if trusted individuals abuse their authority. For example, if an individual authorized to change passwords changes the password for a special user, and then logs on to the ID with the new password, he has all the privileges of that ID. For this reason, it is important to properly separate administrative roles. DAC and MAC for administrative users must be properly implemented.

Misuse of authority

Misuse of authority can be the result of carelessness or a deliberate attempt to misuse authority. Users can tell other users their passwords or assign privileges as a favor to a friend. An auditor can go on vacation and forget to have a substitute monitor a suspicious employee's activity. Deliberate misuse of authority may occur if individuals who have access to sensitive information share that information with unauthorized personnel. Failure to consistently perform one's job functions in a timely manner can also be misuse of authority. For example, if a systems programmer receives maintenance that solves a security problem and does not apply the fix in a timely manner, the system is vulnerable to the security problem that the maintenance fixes. Preventive measures to halt or limit misuse and abuse of authority include sufficient training and education about the use of the system, the security policy, and job responsibilities. Various audit capabilities can monitor users and detect misuse or abuse of authority.

Protection Mechanisms

CA Top Secret provides the following protection mechanisms to enable your site to create a security policy consistent with the requirements of MLS:

Mandatory Access Control (MAC)

MAC imposes a security policy based on security labels. Security labels classify users, data, and resources. Standard access rules and permissions still apply, but only after MAC label dominance checks determine that a user can access data and resources based on their security label and the security label of the data or resources the user wants to access.

The purpose of MAC is to prevent the system from allowing data with a high sensitivity security label from being disclosed to a user with a lower sensitivity security label. For example, a user with a high sensitivity security label cannot send a highly sensitive data set to a user with a lower sensitivity security label. These are known as the “simple security property” and the “confinement property” or “write-down” protection.

Discretionary Access Control (DAC)

In an MLS system, after an authorized user enters the system, data or system resources can be accessed based on whether the organization or other system users want to share data. CA Top Secret DAC security policy manages the controlled sharing of data and resources using rules. Depending on an implementation option, a security administrator or data owner can write rules to permit sharing. If a user tries to access data without permission, the system creates a violation record and denies access.

Object Reuse Protection

An MLS system ensures that no user or program can scavenge data from an object after it has been deleted. Object reuse protection ensures that when a user deletes a data set, the data set is actually erased. Without object reuse protection, the storage would be returned to the storage pool without erasure. A user who obtained storage for a new data set could read the storage and find out what the previous user had put in the data set.

Object reuse protection applies not only to data set objects but also to all objects defined in the system, including address spaces, messages, and devices. An MLS CA Top Secret system provides object reuse protection for data sets if the AUTOERASE control option is specified. Object reuse protection for other objects is provided automatically.

Accountability

An important guideline for any security policy is the need to identify system users and make them accountable for their actions. MLS systems must be able to associate all security-related events, such as system entry or data set access, with a user. The components of accountability are:

User identification and authentication

Each user in an CA Top Secret MLS system is assigned a unique acid. Acid records also contain statistics on password and access histories.

Users provide authentication by entering a secret password with their acid. No one, not even the security administrator, should know the password for another user. CA Top Secret always encrypts passwords. CA Top Secret also provides options that can force users to change their passwords at specific intervals, force users to change their passwords the first time they log on, and force users to enter their passwords in a protected field.

An CA Top Secret MLS system allows users to specify a security label at logon that they have been authorized to use. A security administrator can set, and the system will use, a default security label for users who do not specify a security label at logon. The system can determine whether a user can access objects based on the dominance relationship between the user's session security label and the labels of the objects the user wants to access.

Audit records

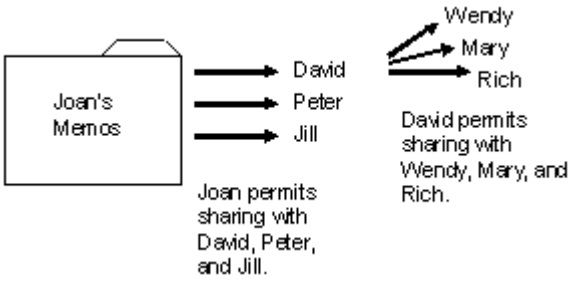
The audit mechanism in an MLS system must be able to create and maintain audit records of all security-relevant events, such as system entry, data access, and resource access. The system must also be able to protect the audit records from modification and accidental loss or disclosure. Security labels are an important part of the audit records. Audit records must display the security label of the user and the security label of the data or resource that the user attempted to access.

CA Top Secret records violations and other activity to the System Management Facility (SMF) or the CA Top Secret Audit/Tracking File. These records are secured from accidental disclosure or destruction by the standard DAC and MAC protection mechanisms. System options allow sites to determine how frequently automatic backups are taken. CA Top Secret provides report generators to produce reports on a wide range of activities. For example, the TSSUTIL report provides an audit trail of system entry events. A variety of parameters can be set to customize the reports to display the violations by a particular type or by a group of users.

MAC and DAC work together to enforce multilevel security on a system. Applying DAC, CA Top Secret access rules can effectively keep users out of data sets they are not authorized to access. However, it cannot keep those authorized to access files from copying them for others. MAC imposes a security policy based on security labels, which prevents the system from allowing data with a high sensitivity security label from being disclosed to a user with a lower sensitivity security label.

Example: Protection Mechanism Failure

In this example, Joan is manager of the Accounts Payable (AP) department and writes CA Top Secret access rules that permit David, Peter, and Jill to read her memos. David copies a memo of Joan's and forwards it to Wendy, Mary, and Rich.



Result:

- Joan's access rules prevent Wendy, Mary, and Rich from reading her memo
- David's access rules permit Wendy, Mary, and Rich to read the copy

The intent of Joan's access rules has been circumvented. On the surface, this looks like a breach of trust: David had access to the information and misused it. Breach of trust is difficult for any security system to protect against. An untrustworthy employee who wants to release information can always use the telephone, write down what is on the terminal screen, or even memorize it to get around security controls. However, there are several ways that David can innocently disclose the memo by copying it. For example, he can:

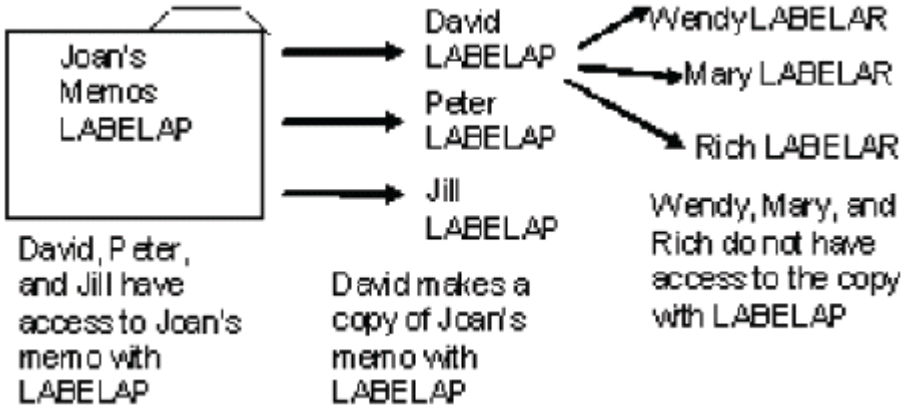
- Forward the memo without realizing that its distribution is restricted.
- Make a working copy of the memo under his logonid, forgetting that he has given Wendy, Mary, and Rich read access to all his data sets.
- Inadvertently run a Trojan horse program that, while doing something else, makes a publicly readable copy of the file. A Trojan horse program is a malicious program with a hidden agenda. It might be a handy utility or a computer game. When most users run it, it works as advertised. But, if the targeted user, in this case, someone with access to Joan's memos, runs the program, it does more. In this case, it makes a copy of Joan's memos.

With multilevel security properly configured in an CA Top Secret system, a MAC layer of security which sits on top of DAC, can prevent Joan's memos from intentionally or unintentionally being accessed by Wendy, Mary and Rich.

Example: Protection Mechanism

In this example, security labels work to protect Joan's memos from being copied to unauthorized users.

Joan has access to security label, LABELAP, and so labels her memos data set. David is also in the AP department and has access to the security label, LABELAP, and can read the memos. If David makes a copy of Joan's memo, the copy will be labeled, LABELAP.



Result:

- Wendy, Mary, and Rich are in the Accounts Receivable (AR) department and have access to the security label, LABELAR, but not to LABELAP.
- Even if David's access rules give them access to the copy of Joan's memo, they still cannot read it because they do not have access to the security label, LABELAP.

Simple Security and Confinement Properties

The simple security property restricts *read* accesses. The simple security property determines when to grant read access, that is, disclose data in an CA Top Secret MLS system. It states that the security label of the user must dominate the security label of the data or resource the user wants to access. This property is always enforced in an CA Top Secret MLS system.

The confinement property (also known as the *-property or “write-down” protection) restricts *write* accesses. The confinement property states that a user can have write access to a file if the security label of the file dominates the security label of the user. This means that if you want to write to a file, the security label of the file must dominate your security label. However, a z/OS system does not support the ability to write to a data set without the ability to first read the data set. Therefore, your security label must equal the security label of the file for you to be able to write to it.

The confinement property is inactive by default in an CA Top Secret MLS system. To enforce the confinement property, a security administrator must set the MLWRITE(NO) control option. When MLWRITE(NO) is set, users cannot write from a higher security label to a lower security label. To write to a lower label, a user must log off and log on at that label. The user would no longer have access to the data at the higher label. In addition, when MLWRITE(NO) is activated, users cannot write up. To write to a higher label, you must logoff and log on at that label. You will no longer have write access to data at the lower label. However, a security administrator or other authorized user can reclassify the data or authorize you for “controlled write-down”, which allows individual users to write down, if they have the proper authorization.

Security Labels, Levels, and Categories

In CA Top Secret, a security label is a 1- to 8-character, alphanumeric name that represents a security level and zero or more categories.

The security level on data represents the sensitivity of the data to unauthorized disclosure. Security levels on users, called *clearances*, indicate their trustworthiness and need to know information. A security label needs only one security level, because levels form a hierarchy. A site can assign the level of trust based on a variety of criteria, such as rank, position, written agreements, or personal background checks. In CA Top Secret, a security level is a numerical rank between 1 and 254 that can also have an optional corresponding name or description, such as the familiar military designations, UNCLASSIFIED, CONFIDENTIAL, SECRET, and TOP SECRET. For example, before someone is cleared for access to SECRET information, the government must perform a background investigation. Before that person is cleared for access to TOP SECRET data, a more extensive background check is done. In other words, the more dangerous unauthorized disclosure of the data would be, the more the military wants to be sure it can trust that person to follow the rules of disclosure.

Categories represent the separation of data based on some use characteristic, such as department or project. In CA Top Secret, a category is a 1- to 32-character, uppercase, alphanumeric name. Categories are independent of levels; they can exist at all levels of the system. Only the level of sensitivity of the data in the categories differs. A site can create categories based on projects, assignments, or group needs, however, categories are optional, and it is not necessary to create them, if there is no need to segregate data. For a user to access data in a particular category, the user in charge of that category must instruct the security administrator to reclassify the user to grant him the access. For example, a user is not normally granted access to personnel information unless the user is working in the personnel department.

Example Definitions

The following are examples of security levels, categories, and security labels as defined in CA Top Secret.

Security Level	Name
150	CONFIDENTIAL
100	INTERNAL USE ONLY
50	PUBLIC
25	*NONE*

Category
HUMAN RESOURCES
FINANCE
SALES

Security Label Name	Value
HIGHEST	SECLEVEL(150) CATEGORY(FINANCE HUMANRESOURCES SALES)
LABEL1	SECLEVEL(100) CATEGORY(HUMANRESOURCES)
LOWEST	SECLEVEL(50)

MAC Label Dominance

CA Top Secret uses the principle of MAC label dominance to determine how security labels compare in an MLS system, and, based on the comparison, whether MAC access is allowed or denied.

For example, if there are two security labels, X and Y:

- X dominates Y if:
  - The level of X is greater than or equal to the level of Y, and
  - X contains at least all categories contained in Y
- X is disjoint from Y if neither X nor Y includes all the categories of the other

In the first rule (X dominates Y), above, both conditions must be true for Label X to dominate Label Y. So, the “and” is important. If the Level of X is less than the Level of Y, then the dominance check has already failed and Label X will never dominate Label Y. However, if the Level of X is greater than or equal to the level of Label Y, then, the categories of Label X and Label Y must be compared to see if all the categories in Label Y are in Label X. If Label X's level is higher than Label Y's, dominance has not yet been established, until the categories are compared. In the second rule (X is disjoint from Y), above, if neither security label X nor Y includes all the categories of the other, the labels are said to be incomparable or disjoint; neither one dominates.

Note: The term “greater than” is used informally to mean dominates. Although labels cannot be compared in a numerical sense, the concept of “greater than” is a convenient way to think of label dominance.

Sample Security Labels

Based on MAC dominance checking rules, here are some security label values and examples of label comparisons and their results:

Security Label	Value	
LABELA	SECLEVEL(5)	CATEGORY(FIN)
LABELB	SECLEVEL(20)	CATEGORY(DEV)
LABELC	SECLEVEL(20)	CATEGORY(HR DEV FIN)
LABELD	SECLEVEL(10)	CATEOGRY(SUPPORT)
LABELE	SECLEVEL(5)	CATEGORY(SALES)
LABELF	SECLEVEL(5)	CATEGORY(SALES)
LABELG	SECLEVEL(5)	

Label Comparison	Comparison Result	Explanation
LABELA vs. LABELB	LABELA does **not** dominate LABELB	LABELA's level is lower than LABELB's level. The categories do not need to be checked.
LABELB vs. LABELC	LABELB does not dominate LABELC	Both labels have the same level, but all of LABELC's categories (HR, DEV and FIN) are not in LABELB's categories (DEV)
LABELD vs. LABELE	LABELD does not dominate LABELE	LABELD's level is higher than LABELE's level, but LABELE's category (SALES) is not also LABELD's category (SUPPORT).
LABELE vs. LABELF	LABELE dominates LABELF	The levels are the same and LABELE's category (SALES) is also LABELF's category (SALES). These label values are the same (for example, they are equivalent). These labels can be said to dominate each other).

Label Comparison	Comparison Result	Explanation
LABELF vs. LABELG	LABELF dominates LABELG	LABELF's level is higher than LABELG's level and LABELG does not have any categories.
LABELA vs. LABELF	LABELA is disjoint from LABELF	The levels are the same, but LABELA's category (FIN) is not LABELF's category (SALES) and LABELF's category (SALES) is not LABELA's category (FIN).

In the table above, to give users who can access data labeled LABELA and LABELE the ability to share certain data, you can create a new label, LABELX, which would have the following value:

Security Label	Value
LABELX	SECLEVEL(5) CATEGORY(SALESFIN)

You could then authorize users to LABELX, which would allow them MAC access to the shared data labeled, LABELX, if DAC checks also allowed it.

Types of MAC Label Dominance Checks

There are three types of MAC label dominance checks in an CA Top Secret MLS system:

- MAC dominance check
- Reverse MAC dominance check
- Equal MAC dominance check

The type of label dominance check performed for each requested access to a classified resource depends on what the resource's class is and whether write-down is restricted (preventing declassification of data in writing from a higher classification to a lower classification).

MAC Dominance Check

The MAC dominance check requires that because opening a data set for write access implicitly opens it for read access, to read-only or read/write to a data set, the user's label must dominate the data set's label. However, there are other resources that support true write-only access such as messages sent with the TSO SEND command and batch jobs submitted through the internal reader. To write-only when write-down is not restricted in an MLS system, the user's label and the resource's label must be comparable, for example, not disjoint. In other words, the user's label must dominate the resource's label or the resource's label must dominate the user's label.

The following table shows the MAC dominance check required for different types of access.

Access Type	When Write-Down Is Allowed	When Write-Down Is NOT Allowed
READ ONLY	Subject security label must dominate object security label	Subject security label must dominate object security label
WRITE ONLY	Subject security label must dominate object security label [or] Object security label must dominate subject security label	Object security label must dominate subject security label
READ/WRITE	Subject security label must dominate object security label	Subject security label must be equivalent to object security label

In the following example, LABELA dominates LABELB under MAC dominance checking rules in an MLS system where write-down is allowed.

Security Label	Value
LABELA	SECLEVEL(5) CATEGORY(AA BB CC)
LABELB	SECLEVEL(5) CATEGORY(AA BB)

USER07 has security label LABELA. Data set TEST.JCLLIB has security label LABELB. USER07 can read and write to TEST.JCLLIB because USER07's label, LABELA, dominates LABELB.

Reverse MAC Dominance Check

The reverse MAC dominance check is the opposite of the MAC dominance check. Reverse MAC dominance requires that the resource's security label dominates the user's security label for the requested access to be allowed. The following resource classes apply reverse MAC dominance checking rules: APPCPORT, CONSOLE, WRITER.

Access Type	When Write-Down Is Allowed	When Write-Down Is NOT Allowed
READ ONLY	Object security label must dominate subject security label	Object security label must dominate subject security label
WRITE ONLY	Subject security label must dominate object security label [or] Object security label must dominate subject security label	Subject security label must dominate object security label
READ/WRITE	Object security label must dominate subject security label	Subject security label must be equivalent to object security label

Equal MAC Dominance Check

If two labels are equal, they dominate each other. Equal MAC dominance checking, which is used for any class that requires two-way communication, requires that the user and resource security labels are the same for the requested access to be allowed. The following resource classes apply equal MAC dominance checking rules: APPL, DSNR, JESINPUT, MQCONN, SERVAUTH, SERVER, TERMINAL

Access Type	When Write-Down Is Allowed	When Write-Down Is NOT Allowed
READ ONLY	Subject security label must be equivalent to object security label	Subject security label must be equivalent to object security label
WRITE ONLY	Subject security label must be equivalent to object security label	Subject security label must be equivalent to object security label

Access Type	When Write-Down Is Allowed	When Write-Down Is NOT Allowed
READ/WRITE	Subject security label must be equivalent to object security label	Subject security label must be equivalent to object security label

The following example shows two equivalent labels:

Security Label	Value
LABELA	SECLEVEL(5) CATEGORY(AA BB)
LABELB	SECLEVEL(5) CATEGORY(AA BB)

Note: LABELA and LABELB have different label names but the same values. They are considered equal.

Separation of Administrative Functions

CA Top Secret provides separation of function as a management control to safeguard your system:

Levels of Authority

CA Top Secret provides for different levels of administrative authority over security functions. For example, one administrator might be authorized to create acid records, while another might be authorized to change data set and resource access rules. These authorities are controlled by privileges in the administrator's acid record. In an MLS system, the MSCA or an authorized SCA administrator is fully authorized to administer MLS.

Scoping of Privileges

CA Top Secret provides the ability to scope or limit privileges of users to the security functions or areas that pertain to them. For example, a scoped security administrator can perform administrative functions only for the acids defined in the group (scope) assigned to him

Centralized or Decentralized Security

CA Top Secret lets you decide how to implement control over security. You can implement a system option to centralize the security environment in the hands of one or just a few people. Or you can delegate the responsibility by placing the means to control security to a wider group of individuals in a decentralized security environment.

System Integrity

The z/OS operating system uses features of both the hardware and the software to ensure system integrity. System states are used to distinguish changes in the operating system. The two z/OS MVS system states are privileged (system) and unprivileged (user programs). The z/OS MVS operating system prevents changes in system status through the use of privileged and unprivileged states. The hardware implements and distinguishes between system and user states. You can use several different diagnostic routines to verify that your system hardware is operating correctly.

Hardware

Verify the hardware is running correctly by running microcode (firmware) and software diagnostic routines. Microcode diagnostics are resident on diskettes or internal hard disks, depending on the processor model. Microcode diagnostics are the first level of problem determination for hardware repair personnel and verify correct operation of the processors.

Error Recording

In addition, z/OS MVS maintains the SYS1.LOGREC data set for the purpose of error recording. This data set cannot be shared between systems. It provides a record of all detected hardware failures and selected software errors and system conditions. Information about each incident is written into SYS1.LOGREC by the system recording routines and can be retrieved using the environmental recording, editing, and printing service aid (IFCEREP1). The IFCEREP1 output can be used for diagnostic or measurement purposes to maintain the devices and to support the system control program.

The IFCDIP00 service aid initializes SYS1.LOGREC during system initialization. IFCDIP00 creates a header record and a time stamp record for the SYS1.LOGREC data set and allocates space for the data set that must reside on the system residence volume.

Records

A record is made on SYS1.LOGREC for every detected hardware or software failure and system condition that has an associated recording request or recording routine. The records contain different types of data that document failures and system conditions. The records are stored in chronological order on SYS1.LOGREC.

In general, each record contains:

- Relevant system information at the time of the failure
- Device hardware status at the time of the failure
- Results of any device or control unit recovery attempt
- Results of any software system recovery attempt
- Statistical data

There are various types of records, containing specific device or incident-dependent information that can be recorded on SYS1.LOGREC, that contain complete and specific information for the device, and type of failure or system condition that caused it to be written.

Machine Failures

Recording machine check records are recorded on SYS1.LOGREC whenever the following detected machine failures occur:

- Central processing unit (CPU) processor
- Storage
- Storage key
- Timer

When a machine failure occurs, the Machine Check Handler (MCH) receives control through a machine-check interrupt for a soft failure (one that was corrected by the hardware retry features) or for a hard failure (one that could not be corrected by the retry features). If the machine check interrupt is for a soft failure, MCH uses the environmental and model independent information describing the failure to build an MCH record. After the information is formatted, MCH passes control to the Recovery Termination Manager (RTM). RTM then invokes the recording request routine that queues the MCH record on the asynchronous output queue and posts the asynchronous recording task. The recording task asynchronously scans the output queue and issues an appropriate SVC to write any records on the queue to SYS1.LOGREC.

If the machine check interrupt is for a hard failure, MCH analyzes the information in the model independent logout area, isolates the error, and provides a record of the analysis to RTM. RTM then takes the same actions as it does for a soft failure.

With each initial program load (IPL), the system begins a sequential count of errors. The sequence number is unique for each detected software error or machine failure. The sequence number remains constant for subsequent software records associated with the same error (although the time stamp may change). Software records are recorded on SYS1.LOGREC for hardware detected hardware errors, hardware detected software errors, operator detected errors, and software detected software errors. For error recording purposes, error data is collected in the System Diagnostic Work Area (SDWA) to assist in identifying the System Control Program (SCP) error and then invoke the RTM.

Physical Security Assumptions

Physical security issues are an important part of your overall system security. To assure your system is secure, you must have locks on doors to secure areas. Only trusted individuals and those authorized to perform relevant job tasks should have access to computer rooms, operator consoles, and in some cases, printer rooms. Some sites may want to secure the mailroom and report distribution so that sensitive data is not left in the open where others may see it.

CA Top Secret MLS can provide output destination control for printers. Each printer can be assigned a security label that prints only those jobs whose security labels pass the dominance check against its own security label. This feature allows you to distribute printers throughout your site rather than confine them to a secure computer room. Printers must be channel-attached. Since devices are daisy-chained on the channels, if you have a printer in a room that is cleared lower than system high (SYSHIGH), you must ensure that no other device on the same channel processes data labeled higher than the clearance of the room.

Systems that allow legitimate user access to their components (for example, removable media) should be used only in environments where both administrative and ordinary users are trusted to access all data in the system and are trusted not to misuse their physical access permission.

Ensure that the level of trust associated with the physical environment containing a system's peripheral always dominates the security label associated with that peripheral.

Secure systems should include a policy that does not permit passwords to appear on JCL card decks and password encryption at terminals.

Chapter 2: Using Security Labels

This section contains the following topics:

Introduction to Security Labels

In an MLS system, most users use a security label only when they log on to the system or submit a job. The rest of the time, security labels are read, decoded, and applied by CA Top Secret and the system. Security administrators can create and assign security labels based on their organization's security policy. In addition, depending on what MLS system options have been set, CA Top Secret will assign a security label to data when it is created.

Determining MLS Access

When MLS is active in CA Top Secret, MAC security label checking is performed before DAC access rule checking, except in the case of system entry where a user must be identified to the system before label validation can be performed.

- If MAC allows an access, a request must still pass through DAC validations to ultimately allow or deny access.
- If MAC denies access, the request is denied and does not go through DAC validations.

CA Top Secret determines MAC access based on the dominance relationship between the label of the object and the label of the subject that is trying to access the object. The factors that CA Top Secret uses to determine the dominance relationship are:

- Simple security property
- Confinement property (*-Property)

Entering the System

Part of the identification and authentication that takes place at system entry is the extraction and authorization of a subject's security label information. CA Top Secret determines a subject's security label at logon only if MLS is active.

To use a different security label to logon, you *must* log off first and then logon again with that security label.

Important! You cannot change your session security label by reconnecting to a TSO session with a different security label.

Specify a Security Label at Logon

When you log on to a system, you can enter a security label. CA Top Secret verifies that you are authorized to use the label by checking your user ACID record. If you are authorized to use the security label, CA Top Secret maintains the security label in your address space and uses it to make access decisions until you log off.

You cannot alter your security label while logged onto the system. To change your security label, log off and log on using a different security label. This reduces the threat from Trojan horses and prevents inadvertent data disclosure.

To specify a label at logon, you can specify:

A user-defined security label

The security label must already be defined in the system and it must be added to your user ACID record.

A system-defined security label

There are two system-defined labels that you may be authorized to use at logon:

SYSHIGH

This label dominates all other labels in the system. The label must be assigned in your user ACID record.

SYSLOW

This label is dominated by all others in the system.

TSO/E Full-Screen Logon

If you have full-screen privileges in your ACID record, the logon panel displays after CA Top Secret validates your ACID and password. The SECLABEL field specifies the user's one- to eight-character security label. MLS must be active on the system before a value is specified in this field.

If the user on the logon command does not specify a label, one may default from the user's previous TSO session or ACID record. The user can override this label by specifying a different authorized one. The label specified must be defined and valid in the system. When the MLMODE control option is:

FAIL

The user is prompted for a valid label until he specifies one or blanks out the value in the SECLABEL field.

WARN or DORM

The user is assigned SYSLOW if an invalid or unauthorized seclabel is specified in the SECLABEL field.

TSO/E Line-Mode Logon

If MLS is active on the system, a user may specify a security label on the LOGON command, such as:

LOGON *logonid*/*password* SECLABEL(*seclabel*)

Logon Without a Security Label

Supplying a security label is not required. CA Top Secret will attempt to default a security label for a user who does not supply a label. If a user does not specify a security label at logon, if there was a previous TSO/E session for the user, the security label from that session will be used (in full-screen mode, only). If there was no security label for the previous TSO/E session, CA Top Secret uses the security label from the TERMINAL or SERVAUTH class MLS resource record, if there is one. If the terminal or server does not have a security label, CA Top Secret uses the default security label from the SECLABEL field in the user acid record, if one exists for the user. If CA Top Secret cannot default a security label from any of these places, the user will be logged on with the system-defined security labels, SYSLOW, the lowest label in the system.

Session Security Label Display

The TSS WHOAMI command is used to display your current active security label in an MLS environment once you are successfully logged onto the system. This security label is the one with which you entered the system and which endures for the duration of your session. It cannot be used to display any user's security label other than your own.

Specify a Security Label in JCL

To specify a security label in JCL, specify a SECLABEL logon parameter with JCL JOB statement parameters:

SECLABEL=*seclabel*

Specify a Security Label for Started Tasks

Certain started tasks represent important system components that are required for z/OS to run properly. Security labels cannot be specified when a START or MOUNT command is issued at the console (started task) or when the TSO LISTBC command is issued or when system address spaces are started, such as CATALOG, SMF, DUMPSRV, CONSOLE, SMS, JES2, JES3. In these cases, the operating system generates a RACROUTE REQUEST=VERIFY call and assigns the TRUSTED parm a value of YES, indicating a “trusted” user is entering the system. When MLS is active, although you can assign a default security label to a started task, such as OMVS, by adding a SECLABEL to the acid record for it, if it is considered “trusted”, MAC security label checking will be bypassed, but logged (if the MLS mode is WARN, or FAIL), at system entry and for all requested access to data sets and other resources by the started task. In addition, MAC label checking is bypassed, and logged for BYPASS acids.

Console Logon

To specify a security label on the console at logon, enter:

SECLABEL=*seclabel*

Verifying User Access to An Object

After a user has successfully logged onto the system, their security label becomes attached to their address space. When the user tries to access an object, CA Top Secret performs two checks: a MAC check and a DAC check. The MAC check compares the user's label with the object's label to establish the label dominance relationship. The DAC check determines if a rule exists that permits the user to access the object and what type of access is permitted. If MAC permits the access, DAC is performed. If the MAC check fails, no DAC check is performed.

Access is granted according to the following criteria:

- If the user requesting the access is trusted or has bypass attributes in their acid, MAC checking is bypassed but logged by CA Top Secret, if the MLS mode is WARN, or FAIL.
- If the MLS mode is DORM, MAC checking is bypassed without logging; only DAC checking is performed.

The following applies if write-down is not restricted:

- If a user requests READ, EXECUTE, or CREATE access, CA Top Secret checks to see if the security label of the user dominates the security label of the object. If MAC permits the access, DAC checking is performed to ultimately allow or deny the access.
- If the user requests WRITE access, CA Top Secret checks to see if the security label of the user dominates the security label of the object or the security label of the object dominates the security label of the user, for example, the labels must not be disjoint. If MAC permits the access, DAC checking is performed to ultimately allow or deny the access.
- If a user requests ALL, UPDATE or SCRATCH access, CA Top Secret checks to see if the security label of the user dominates the security label of the object. If MAC permits the access, DAC checking is performed to ultimately allow or deny the access.

The following applies if write-down is restricted:

- If a user requests READ, EXECUTE or CREATE access, CA Top Secret checks to see if the security label of the user dominates the security label of the object. If MAC permits the access, DAC checking is performed to ultimately allow or deny the access.
- If a user requests UPDATE or SCRATCH access, CA Top Secret checks to see if the security label of the user is equivalent to the security label of the object. If MAC permits the access, DAC checking is performed to ultimately allow or deny the access.
- If the user requests WRITE access, CA Top Secret checks to see if the security label of the object dominates the security label of the user. If MAC permits the access, DAC checking is performed to ultimately allow or deny the access.

- If a user requests ALL access, CA Top Secret checks to see if the security label of the user is equivalent to the security label of the object. If MAC permits the access, DAC checking is performed to ultimately allow or deny the access.

Access Classified Data Sets

MAC access to classified data sets is determined by label dominance checking rules, which depend on what MLS options have been set in the system and what kind of access has been requested. In addition, if MAC checking allows the access, DAC checking is then performed to ultimately allow or deny the requested access.

The following security labels are used in the examples below:

Security Label	Value
TSAABBDD	SECLEVEL(50) CATEGORY(AA BB DD)
LABELB	SECLEVEL(50) CATEGORY(AA BB)
LABELA	SECLEVEL(50) CATEGORY(AA)
LABELD	SECLEVEL(50) CATEGORY(KK)
TSRR	SECLEVEL(50) CATEGORY(RR)
SSAABBRR	SECLEVEL(25) CATEGORY(AA BB RR)
LABELC	SECLEVEL(25) CATEGORY(AA)
LABELE	SECLEVEL(25) CATEGORY(KK)

Sending Messages, Mail, and Data Sets

Sending messages, mail, and data sets all work similarly with regard to labeling in an MLS environment.

If write-down is allowed, messages, mail, and data sets that are created do not inherit the label of the user who created them. Instead, they will be unclassified and users can access the data, if access rules allow it.

However, when write-down is prohibited, messages, mail, and data sets inherit the label of the user who creates them. For example, if you write a message while logged on at SECRET, the message is labeled SECRET. In a write-down protected MLS environment, to enable sites to send BRODCAST messages across all labels of the system, the security administrator must set the facilities that are used to send BRODCAST messages to the lowest label on the system (SYSLOW). To send a BRODCAST message, one must log on at the lowest label of the system (SYSLOW). Messages can be sent up but not down. These protections ensure that a user cannot place sensitive data in a message that can be received and stored at a level where unauthorized users can access it.

Examples: sending messages

Bill logs on with the SYSHIGH label. He can send messages, mail, or data sets to Mary, who is logged on with the SYSHIGH label, and to John, who is logged on with the SYSHIGH label.

Bill can send a message to Jane, who is logged on at the SECRET label, but she cannot receive the message while logged on at the SECRET label. When she logs on with a SYSHIGH label she can receive Bill's message. Until she does, however, she cannot view the message.

If Bill sent Mary a SYSHIGH data set, it would not appear in her reader until she logged on with her SYSHIGH label.

Jennifer is logged on with the SYSLOW label (the system low), can send a BRODCAST message about a company meeting to users at all labels of the system. Any user can read the message without having to log off his current label.

Delete Data Sets

An MLS system does not permit a classified user to delete a classified data set that is at a higher label than his current label. A user must have a label that dominates that of the object he wants to delete.

Create Data Sets

When MLS is active and write-down is allowed, data sets that are created do not inherit the label of the user who created them. Instead, they will be unclassified until an authorized security administrator labels them. Therefore, users can access the data as long as it remains unclassified. However, once a security administrator classifies the data, MAC access to it is determined by label dominance checking rules.

When MLS is active and write-down is prohibited, the data set inherits the session security label of the user who created it. When a user logs on with a SECRET label and creates a data set, that data set is labeled SECRET.

Renaming Data Sets

When MLS is active and write-down is not protected, a new data set will not be labeled by the system at the time it is created. Also, if it was not assigned a security label by the security administrator before it was created, the data set will be unlabeled (“unclassified”). Thus, if the data set is renamed, the new data set will not have a security label unless it is assigned one by the security administrator.

When MLS is active and write-down is protected, when an existing data set (“old data set”) is renamed, if the newly named data set (“new data set”) does not have a security label, then the security label of the old data set will be used as the security label of the new data set.

When a data set is renamed, read/write access is requested, which requires the user's security label to be equal to the old data set's security label (if it has a label), to rename it. If the data set does not have a label, the user can rename it, as long as access rules permit it. In most cases, the new data set label will be equal to the old data set label, except when the old data set label is SYSNONE or SYSMULTI and the user label isn't.

In the case where the old data set name and the new data set name both already have security labels, but they are different from each other, the following requirements must be met in order for the data set to be successfully renamed, as long as access rules also permit it:

- Global write-down is protected (MLWRITE(NO) control option is set)
- One of the following is true:
  - The old data set has a label of SYSNONE (which is equal to any other label)
  - All of the following must be true:
    - The user's security label dominates the old data set's security label
    - The user's security label equals the new data set's security label
    - The new data set's security label dominates the old data set's security label

Copy Data Sets

When MLS is active and write-down is allowed, users can copy classified data as long as their security labels are not disjoint, based on MAC label dominance checking rules. The user's security label must dominate the security label of the data.

When MLS is active and write-down is protected, the data's security label must be equivalent to the user's security label. The following examples explain what happens when a user attempts to copy a data set when write-down is prohibited.

Example: Copy a dataset

In this example, Bill logs on at LABELA.

Bill's security label grants him access to the WORK.DISCOVER data set, which is also labeled with LABELA.

Bill copies the WORK.DISCOVER data set (LABELA) to the WORK.ATLANTIS data set (LABELA).

Bill can copy the data set because his label is equal to the label of the new data set.

Bill logs on at LABELB.

Bill copies WORK.DISCOVER to the new WORK.ATLANTIS that is then labeled LABELB.

Example: Copy a dataset

Jim logs on with LABELD.

Jim's manager tells him that the members of the data set WORK.BUDGET (LABELD) should be moved to the WORK.ACTUAL data set (LABELE).

Jim cannot copy the WORK.BUDGET data set (LABELD) to the WORK.ACTUAL data set (LABELE) because the label of the data set (LABELE) does not equal his label (LABELD) and he cannot write down.

If Jim logs on with LABELE, he cannot get read access to the data set WORK.BUDGET (LABELD). Jim must reclassify the data set. To reclassify this data set, Jim must enlist the help of the security administrator who is authorized to reclassify data. The security administrator is trusted to follow the reclassification procedures. To give Jim the ability to move members of the data set WORK.BUDGET (LABELD) to the WORK.ACTUAL data set, the security administrator could reclassify WORK.ACTUAL from LABELE to LABELD. After the security administrator reclassifies the data sets, Jim can copy the data sets.

Only authorized security administrators can reclassify data. CA Top Secret creates a logging record each time a subject or object is reclassified. In this way, the site can monitor when data sets get reclassified, how the classification changes, and who executes the change.

Note: A security administrator can also give a user “controlled write-down” authorization, to allow a user to declassify data without having to reclassify it. This is not recommended, except in special cases.

Accessing Classified z/UNIX Files and Directories

When MLS is active, CA Top Secret assigns a security label to a file or directory in an HFS or zFS file system at the time it is created based on the security label of the parent directory or user. In addition, a security administrator can issue the UNIX **chlabel** command to assign security labels to files and directories in a zFS file system that do not have security labels because they were created before MLS was activated on the system.

To access a classified file or directory, the user must be signed on with a security label that will allow the access according to MAC label dominance checking rules and other USS permissions.

Chapter 3: Implementing and Administering an Multilevel Secure System

This section contains the following topics:

Implementation Checklist

Use the following checklist to track completion of each step of the implementation process:

Task	
Determine who will administer MLS	□
Delegate MLS administrative authority (optional)	□
Select what to classify with a security label	□
Define security levels	□
Define categories (optional)	□

Task	
Define security labels	□
Activate security levels, categories, and security labels	□
Assign security labels to objects	□
Assign security labels to data sets	□
Assign security labels to resources	□
Assign security labels to DB2 resources	□
Assign security labels to IP addresses	□
Assign security labels to UNIX files and directories	□
Assign security labels to UNIX IPC objects	□
Assign security labels to users	□
Establish the MLS environment	□
Define the MLS Control Options	□
Require security labels (optional)	□
UNIX files and directories (optional)	□
UNIX IPC objects (optional)	□
Prohibit write-down (optional)	□
Activate “controlled write-down” (optional)	□
Activate name hiding (optional)	□
Activate system-specific security labels (optional)	□
Change the MODE setting	□
Activate MLS in DORM mode	□
Test MLS in DORM mode	□
Activate MLS in WARN mode	□
Test MLS in WARN mode	□
Fine-tune MLS in WARN mode	□
Migrate MLS to FAIL mode	□
Deactivate MLS	□
Monitor MLS	□
HELP MLS command	□
TSS WHOAMI command	□

Task	
MLWRITE command	□
MODIFY(STATUS(MLS)) command	□
LIST(MLS) command	□
Audit MLS	□
Check authorization	□
TSSUTIL Report Generator	□
TSS sectrace	□
Trace SAF requests	□
Trace OMVS	□
Use ISPF panels to administer MLS	□
Use TSS commands to administer MLS	□

Determine Who Administers MLS

The MLS administrator performs all security-related functions for the MLS environment in CA Top Secret. The MLS administrator:

- Delegates MLS administrative authority
- Selects what to classify with a security label based on implementation of the security policy as defined for the organization and the plans agreed on by the implementation team
- Defines security levels
- Defines categories
- Defines security labels
- Activates security classifications
- Assigns security labels to users
- Assigns security labels to objects
- Establishes the MLS environment options
- Configures software to run in an MLS environment
- Activates MLS
- Reclassifies users and objects
- Monitors MLS
- Audits MLS
- Traces SAF requests for MLS-related events
- Uses ISPF panels to administer MLS
- Uses TSS commands to administer MLS

The MSCA or an SCA acid with the MLSADMIN authority in their acid has the authority to perform all of the above MLS functions. Because the powers are unlimited, the MLS administrator must be a highly trusted individual.

Determine What to Classify

The MLS implementation team must decide how to label the subjects and objects at a site. For example, objects can be any of the following:

- Data sets
- User-defined resources and devices
- Graphics terminals
- VTAM nodes
- Communication lines
- Communications Controllers
- Channel-to-channel adapters
- DB2 resources
- UNIX files and directories
- UNIX IPC objects
- Servers

Subjects can be any of the following:

- System users
- Programs and processes
- Devices

Planning Questions

Before a security administrator can begin to create security labels and assign them to subjects and objects in the system, the MLS implementation team must first determine what users, data, and resources need to be classified and the implications of doing this. The following is a list of general questions to consider in planning for MLS:

- What levels of sensitivity of data exist?
- What areas of your organization have similar security requirements?
- What areas at the site need to segregate data from other areas while sharing resources?
- Which users need access to data at defined levels of sensitivity?
- What authorization should users have to access data at defined levels of sensitivity?
- How stringent does security need to be at your site? For example, does every resource and user need to be classified, or just certain users and resources?
- Does your site have the resources to establish and maintain the level of protection deemed necessary according to your security policy to protect your sensitive data and resources with security labels in an MLS environment?
- What software is currently running on your systems and what modifications, if any, would need to be made to configure this software in an MLS environment?
- What authorized programs are running on systems that might compromise MLS or be impacted by how MLS is established in the system?
- How would system performance be impacted by establishing or phasing in MLS and activating certain MLS system options?

Define Security Levels

In an MLS environment, after determining what degrees of sensitivity and trust are necessary to the organization or parts of the organization, an authorized security administrator can create levels, which are the hierarchical elements of security labels.

MLS SECLEVEL Records

An CA Top Secret MLS SECLEVEL Record segment defines a security level available in the system. You must define a separate record for each level you want to use in the system. You must define levels before you can define and assign security labels to users, data sets and resources.

Important! If you change or delete an existing security label, (for example, MLS Seclabel data record) that has been assigned to users or resources, you may get unexpected results during MLS validation. Before changing or removing a security label from the system, check whether it has been assigned to any users or resources. If it has, confirm that the change or deletion is intended. If it is, make any necessary changes to user acids and MLS resource records that are using the security label. If you delete a security level or category used in any existing security label, before removing the level or category from the system, confirm that the deletion is intended. If it is, make any necessary changes to existing security labels, and any user acids and MLS resource records using the security labels.

The format for this command is:

`[Add | Remove | List] (MLS) SECLEVEL(`*`level`*`) lvlname(`*`seclevel-name`*`)`

level

Specifies a record ID, which is a number between 1 and 254 without leading zeros or internal spaces. The number specified is the numeric rank of a security level. This field is required. The value supplied places the level in the hierarchy of all levels. The higher the number, the higher the level. A security label with a particular level dominates labels, whose levels have lower values, except as further restricted by categories. You cannot assign the same value for more than one level for a system. To change the value of a level, remove the SECLEVEL record and add a new one.

Range: 1 to 3 characters

Valid Record IDs: 1, 5, 10, 254

Invalid Record Ids: 01, 005, 010, 255

Lvlname(seclevel-name)

Specifies the unique, alphanumeric name of a security level. The name is always uppercased. Internal spaces are allowed, however, any leading or trailing blanks are trimmed off of the specified name. The name may never begin with the letters 'SYS', since this may cause confusion with any existing or future system-defined security labels. This field is optional.

Range: 1 to 255 characters

MLS SECLEVEL Record Creation

To create a SECLEVEL data record, enter:

```
TSS ADD(mls) SECLEVEL(200) LVLNAME('top secret')

TSS ADD(mls) SECLEVEL(100) LVLNAME(secret)

TSS ADD(mls) SECLEVEL(75) LVLNAME(classified)

TSS ADD(mls) SECLEVEL(25) LVLNAME(unclassified)
```

View an MLS SECLEVEL Record

To view a SECLEVEL data record, enter:

```
TSS LIST(mls) SECLEVEL(200)

 MLS SECLEVEL RECORDS
 SECLEVEL = 200   LVLNAME = TOP SECRET

TSS LIST(mls) SECLEVEL(all)

 MLS SECLEVEL RECORDS
 SECLEVEL = 025   LVLNAME = UNCLASSIFIED
 SECLEVEL = 075   LVLNAME = CLASSIFIED
 SECLEVEL = 100   LVLNAME = SECRET
 SECLEVEL = 200   LVLNAME = TOP SECRET
```

MLS SECLEVEL Record Deletion

To delete a SECLEVEL data record, enter:

```
TSS REM(mls) SECLEVEL(100)
```

Defining Categories

In an MLS environment, after determining if it is necessary to isolate users, data, and resources within the organization, an authorized security administrator can create categories, which are the optional, non-hierarchical elements of security labels. If security labels in your system will contain categories, you must define these records before you can define and assign security labels to users, data sets and resources.

MLS CATEGORY Record

An CA Top Secret CATEGORY Data Record defines a category available in the system. You must define a separate record for each category you want to use in the system.

Important! If you change or delete an existing security label, (for example, Seclabel data record) that has been assigned to users or resources, you may get unexpected results during MLS validation. Before changing or removing a security label from the system, check whether it has been assigned to any users or resources. If it has, confirm that the change or deletion is intended. If it is, make any necessary changes to user acids and MLS resource records that are using the security label. Likewise, if you delete a security level or category that is used in any existing security label, before removing the level or category from the system, confirm that the deletion is intended. If it is, make any necessary changes to existing security labels, and any user acids and MLS resource records that are using the security labels.

The format of this command is:

```
(Add|List|Remove) Category(category-name)
```

category-name

Specifies the unique, uppercase, alphanumeric name of a category in the system. The category name cannot contain internal spaces. Duplicate categories are not allowed. In addition, the category name may never begin with the letters 'SYS', since this may cause confusion with any existing or future system-defined security labels. This field is required. The maximum number of categories that can be defined is limited only by the size of the database. To change a category, delete the CATEGORY record and add a new one.

Range: 1 to 32 characters

MLS CATEGORY Record Creation

To create a CATEGORY Data Record, enter:

```
TSS ADD(mls) CATEGORY(humanresources)

TSS ADD(mls) CATEGORY(finance)

TSS ADD(mls) CATEGORY(sales)

TSS ADD(mls) CATEGORY(development)
```

MLS CATEGORY Record View

To view a CATEGORY Data Record, enter:

```
TSS LIST(mls) CATEGORY(all)

 MLS CATEGORY RECORDS
 CATEGORY = DEVELOPMENT
 CATEGORY = FINANCE
 CATEGORY = HUMANRESOURCES
 CATEGORY = SALES
```

MLS CATEGORY Record Deletion

To delete a CATEGORY Data Record, enter:

```
TSS REM(mls) CATEOGRY(sales)
```

Defining Security Labels

In an MLS environment, after defining and creating security levels and categories in the system, an authorized security administrator can define the security labels that will be assigned to users, data sets and resources.

SECLABEL Data Record

The SECLABEL Data Record defines the value of a security label. You must define this record before you can assign the security label to users, data sets and resources.

Important! If you change or delete an existing security label, (for example, MLS Seclabel data record) that has been assigned to users or resources, you may get unexpected results during MLS validation. Before changing or removing a security label from the system, check whether it has been assigned to any users or resources. If it has, confirm that the change or deletion is intended. If it is, make any necessary changes to user acids and MLS resource records that are using the security label. Likewise, if you delete a security level or category that is used in any existing security label, before removing the level or category from the system, confirm that the deletion is intended. If it is, make any necessary changes to existing security labels, and any user acids and MLS resource records that are using the security labels.

The format of this command is:

```
TSS ADD|REMOVE(MLS) SECLABEL(seclabel)
                    SECLEVEL(seclevel)
                    [CATEGORY(category1,…category50)]
                    [SYSID(sysid1,…sysidn)]
```

seclabel

(Required) Specifies the alphanumeric-national character name of a security label. The security label cannot start with the letters 'SYS'. Security labels that begin with 'SYS' are reserved for existing or future system-defined security labels.

Range: 1 to 8 bytes

Note: To assign a security label to a resource, the security label record ID must be specified in the SECLABEL field of an MLS resource record. To assign the security label to a user, the security label record ID must be specified in the SECLABEL or DFLTSLBL field of a User acid record.

Seclevel(seclevel)

(Required) Specifies the security level which is the record ID of an existing MLS SECLEVEL Record. The security level must be a number between 1 and 254 without leading zeros.

Range: 1 to 3 characters

Important! Any seclevel specified must be a valid MLS SECLEVEL Record defined in the system. Otherwise, this security label will be ignored by the system at the time the security classification tables are built and any users or resources that have been assigned this security label and try to use it will be not be able to.

Category(category1,…category50)

Specifies the alphanumeric names of 1 to 50 categories that are the record IDs of existing MLS CATEGORY records. This field is optional. A comma or blank is the only valid delimiter between specified categories.

Range: 1 to 32 characters

Notes:

- Any category specified must be a valid CATEGORY Data Record defined in the system. Otherwise, this security label will be ignored by the system at the time the security classification tables are built and any users or resources that have been assigned this security label and try to use it will be not be able to.
- To implement roles-based classification (for example, classification using categories only), assign the same SECLEVEL in each security label when it is created. To properly implement security classification with categories only, the security level (rank) of all active security labels must be the same. Otherwise, this could cause problems with MLS validations on your system.

Sysid(sysid1,…sysidn)

Specifies one or more alphanumeric system IDs on which this security label can be used, if the MLS option for use of system-specific security labels is active (MLSECBYS). However, if the option is inactive, this field is ignored during MLS validations, and this security label can be used on any system. This field is optional and can be masked by using asterisk(*) or dash(-) masking characters. If no system IDs are specified, by default, the security label will apply to all systems. A comma or blank is the only valid delimiter between specified system IDs.

Note: You must specify the TSS control option MLSECBYS(YES) to limit the use of security labels to certain systems.

Range: 1 to 4 characters

System-Defined Security Labels

CA Top Secret provides the following system-defined security labels which are internal to CA Top Secret and can never be directly created or modified by a user but can only be assigned to users, data sets, and resources:

SYSHIGH

The highest security label in any system. It is comprised of the highest level in the system and all categories. Therefore, it dominates all security labels.

SYSLOW

The lowest security label in any system. It is comprised of the lowest level in the system and no categories. Therefore, all security labels dominate it.

SYSNONE

A security label used in a system when write-down is not allowed. It should be assigned only to non-sensitive data and resources to which everyone, regardless of their security label, must have access, such as catalogs. It compares equivalent to any other security label. This security label cannot be assigned to users.

SYSMULTI

A security label that is equivalent to all other defined security labels. It should be assigned only to servers that can properly isolate users and data based on security labels or UNIX directories that contain subdirectories and files at different security levels. This security label is usually not assigned to users, but there are some exceptions.

SECLABEL Data Record Creation

To create a SECLABEL Record, enter:

```
TSS ADD(mls) SECLABEL(labelaaa)
             SECLEVEL(150)
             CATEGORY(humanresources,finance,sales)
```

SECLABEL Data Record View

To create a SECLABEL Record, enter:

```
TSS ADD(mls) SECLABEL(labelaaa)
             SECLEVEL(150)
             CATEGORY(humanresources,finance,sales)
```

Change a SECLABEL Data Record

After a SECLABEL record is defined, additional categories and/or sysid's may be added. To change the record, enter:

```
TSS ADD(mls) SECLABEL(labelaaa)
             SECLEVEL(50)
             SYSID(sysa)
```

Delete an MLS SECLABEL Record

To delete a SECLABEL Record, enter:

```
TSS REMOVE(mls) SECLABEL(labelaaa)
```

Activating Security Levels, Categories, and Security Labels

Once security levels, categories, and security labels have been defined in the system, they can be used.

Assigning Security Labels to Objects

Security labels for data sets and non-DB2 resources are defined in the MLS record. The security administrator should create these records for any data and non-DB2 resources that are deemed sensitive in an organization, that if left unprotected, might result in data disclosure or declassification.

Note: when the option to protect write-down is set, CA Top Secret will label data at the time it is created and store the security label for the data set in an MLS resource record.

Assigning Security Labels to Data Sets

In an MLS environment, an authorized security administrator can assign defined security labels to data sets.

The format of this command is:

```
TSS ADD|REMOVE(MLS) DSN(dsname)
                    SECLABEL(seclabel)
                    MODE(mode)
```

DSN(dsname)

Specifies the name of the dataset to be protected with a security label. The dsname can be the full dataset name or a prefix name and can contain any of the masking characters that are supported in CA Top Secret.

Seclabel(seclabel)

Specifies the alphanumeric-national character name of a security label. The security label must be predefined in the MLS record or it may be one of the system defined security label names.

Range: 1 to 8 bytes

Mode(mode)

Specifies the security mode under which security validation will be performed

To assign a security label to an existing data set, create an MLS record for it.

This example assigns security label, SYSLOW, to data set, SYS1.BRODCAST, and sets the mode to FAIL. When MLS is activated on the system, users who have entered the system with a security label of at least SYSLOW, would be able to access dataset SYS1.BRODCAST.

```
TSS ADD(mls) DSN(sys1.broadcast)
             SECLABEL(syslow)
             MODE(fail)
```

Labeling Catalogs and Critical Data Sets

While MLS is still inactive on a system, and write-down protection is not yet active, you should label all catalogs SYSNONE. In addition, you should label all critical data sets if you plan on protecting write-down on the system.

The following table lists the security labels that are recommended for system data sets:

Data Set Name	Security Label	Data Set Content
Data sets specified in LNKLSTxx and LPALSTxx members of SYS1.PARMLIB	SYSLOW	Publicly readable data
JES spool data sets	SYSHIGH	
Page and swap data sets and SYS1.STGINDEX	SYSHIGH	System page and swap data sets
SYS1.BRODCAST	SYSLOW	Notices for all system users
SYS1.DAE	SYSHIGH	Dump analysis and elimination data sets
SYS1.DUMPxx	SYSHIGH	System dumps
SYS1.HASPACE	SYSHIGH	JES2 spool spaces
SYS1.HASPCKPT	SYSHIGH	JES2 checkpoint data sets
SYS1.HELP	SYSLOW	Online command documentation
SYS1.IMAGELIB	SYSLOW	FCB images
SYS1.LINKLIB	SYSLOW	
SYS1.LOGREC	SYSHIGH	Hardware and software error loggings
SYS1.MANx	SYSHIGH	SMF records
SYS1.PARMLIB	SYSLOW	
SYS1.PROCLIB	SYSLOW	
SYS1.VTAMLIST	SYSLOW	
Trace data sets created by GTF	SYSHIGH	Data about tasks
User mail logs	SYSHIGH	Mail with various security labels depending on the label of the user that sent the mail

Assign Security Labels to Non-data set Resources

While MLS is still inactive on a system, all non-dataset resources that require MAC protection should be labeled.

To assign a security label to a non-dataset resource, you need the name of the resource that you want to secure. In CA Top Secret these names are referred to as resource names

To assign a security label to a resource, create an MLS data record for it.

This command has the format:

```
TSS ADD|REMOVE(MLS) RESCLASS(resname)
                    SECLABEL(seclabel)
                    MODE(mode)
```

Resclass(resname)

Specifies the name of the resource class and the resource name to be protected with a security label. The resclass must be a resource class defined in the RDT record. The resname can be the full resource name or a prefix name and can contain any of the masking characters that are supported in CA Top Secret.

Seclabel(seclabel)

Specifies the alphanumeric-national character name of a security label. The security label must be predefined in the MLS record or it may be one of the system defined security label names.

Range: 1 to 8 bytes

Mode(mode)

Specifies the security mode under which security validation will be performed.

Example: assigning a security label

```
TSS ADD(mls) JESJOBS(submit.mynode.*.user01)
             SECLABEL(usrlbl1)

TSS ADD(mls) OTRAN(payr)
             SECLABEL(labelap)
```

Assigning Security Labels to DB2 Resources

While MLS is still inactive on a system, all DB2 resources that require MAC protection should be labeled.

To assign a security label to a resource, add a seclabel to the resource name in the MLS record.

Example: assigning a security label

```
TSS ADD(MLS) DB2(TEST.QEWRQER.*.ASDF)
             SECLABEL(LABELA)
```

Assigning Security Labels to IPv6 Addresses

In z/OS V1R5, IBM created a new SESSION type, IP, and a new port-of-entry class, SERVAUTH, on the RACROUTE REQUEST=VERIFY/X macro. The SERVAUTH keyword specifies the address of the identifier of the server through which a user is trying to gain access to the system. The address points to a 1-byte length field followed by a 64-byte data area, which contains the name of the resource in the SERVAUTH class. This resource name is the network access security zone name that contains the IPv6 address of the user. Security zone mappings are defined in the NETACCESS parameter block in a TCP/IP profile.

The network access zone name to which IPv6 addresses are mapped is in the following format:

EZB.NETACCESS.*sysname.stackname.zone*

While MLS is inactive on a system, a security administrator should label all SERVAUTH resources that require MAC protection, including IPv6 addresses through which users will attempt to gain access to the system.

To protect system entry from an IPv6 address:

- Assign an IPv6 address to a network security zone by creating a TCP/IP Profile definition. For example:

  ```
  TCP/IP Profile:
  NETACCESS
          9.24.104.0/24        ZONE1
          9.24.104.119/32      ZONE2
  ENDNETACCESS
  ```

- Create an appropriate CA Top Secret Seclabel record, which classifies the network access zone name and the IPv6 address mapped to it.
- The following CA Top Secret MLS resource record would allow USERA to access the system from IPv6 address 9.24.104.119/32 which is in ZONE2 only if the security label with which USERA enters the system is equivalent to LABEL2 (the security label of the network security ZONE2) and resource rule validation also allows the access.

  ```
  TSS ADD(MLS) SERVAUTH(ezb.netaccess.-.zone2)
               seclabel(label2)
  ```

Once this is done, and MLS is active on the system, if a security label *is not* specified by a user or application at signon, the seclabel is defaulted from the SERVAUTH resource (if there is one and it is not SYSMULTI), only if the user is authorized to it in his User SECLABEL acid record. If a security label *is* specified by a user or application at signon, system entry is allowed if the user is authorized to the security label specified, it is equivalent to the security label that is protecting the IPv6 address in the SERVAUTH profile (if there is one), and rule validation allows the access.

Security label checking is performed at system entry to ensure that the user's security label is equivalent to the security label of the SERVAUTH resource. If it is not, the user will be denied access to the system through the server. If the security label check succeeds, rule validation is then performed to ultimately allow or deny the access request.

Note: To allow a user to enter the system from an IPv6 address, do not assign a security label to the network security zone. In addition, create a resource rule for the network security zone in which the user's IPv6 address is mapped. Otherwise, access will be denied.

The following CA Top Secret resource rule would allow USERA access the system from IPv6 address 9.24.104.119/32, which is in ZONE2.

```
TSS PER(USERA) servauth(ezb.netaccess.-.zone2)
               access(read)
```

Important! To support IPv6 addresses, which are much longer than IPv4 addresses, the TERMID is no longer used as the source ID for IP-based ports of entry trying to gain access to the system and resources. Instead, the network access security zone name in the SERVAUTH class contains the IPv6 address of the user trying to gain access to the system and resources. This functionality replaces conversion of IPv4 addresses to hexadecimal terminal names.

Assign Security Labels to Objects

In an MLS environment, after defining and creating security levels, categories and security labels and activating them, an authorized security administrator can assign defined security labels to objects in the system, such as data sets, UNIX files, directories, symbolic links, IPC objects, and other kinds of resources.

In an MLS environment, there are two ways that security labels can be assigned to data, depending on whether or not write-down protection is enabled on a system and whether or not the data is being newly created or previously existed before write-down protection was enabled:

- If the MLWRITE(NO) control option is set and write-down is not allowed, when data is created, CA Top Secret will assign to it the session security label of the user who created the data. The security label assigned is stored in an MLS data record, which can never be modified, only viewed or deleted. Once data has been labeled in this way, to reclassify it by assigning a different security label, a security administrator must create an MLS record for the data set with the changed security label. A security administrator can issue REMOVE or LIST commands to delete or view the security label that CA Top Secret-assigned to a new data set in the MLS record
- If the MLWRITE(YES) control option is set and write-down is allowed, when data is created, CA Top Secret will NOT assign to it a security label. Instead, if the data should be classified, a security administrator must create an MLS data record for the data set.

Assigning Security Labels to UNIX Files and Directories

When MLS is active on an CA Top Secret system, the user security label is assigned by CA Top Secret to a UNIX file or directory at the time it is created/allocated.

However, if MLS is not active on the system at the time the UNIX files and directories are created, but is later turned on, these objects will not have security labels. As a result, the security administrator should provide security labels for any existing UNIX files and directories that do not have them by issuing the UNIX **chlabel** shell command; otherwise, if the MLS option to require security labels for UNIX files and directories is active (MLFSOBJ(YES)), all accesses to these files and directories by users will be denied by CA Top Secret.

Impotant! The UNIX **chlabel** command many only be issued in a zFS file system. It will not work in an HFS file system. Once a file or directory has been assigned a security label, it cannot be deleted or changed with the **chlabel** command.

The following table lists the security labels that are recommended for UNIX files, directories, and symbolic links that have not been assigned security labels by the system.

Directory/File	Security Label
/bin and contents	SYSLOW
/lib and contents	SYSLOW
root	SYSMULTI
root, symbolic links in: /tmp, /dev, /etc, /var	SYSLOW
/samples	SYSLOW
/SYSTEM	SYSMULTI
/SYSTEM/tmp mountpoint	SYSMULTI
/SYSTEM/dev mountpoint	SYSMULTI
/SYSTEM/etc mountpoint	SYSMULTI
/SYSTEM/var mountpoint	SYSMULTI
/SYSTEM, symbolic links in: /SYSYTEM /tmp, /SYSTEM /dev, /SYSTEM /etc, /SYSTEM /var	SYSLOW
/u	SYSMULTI

Directory/File	Security Label
/u, symbolic link for security label substitution	SYSLOW
/u/*seclabel* mountpoint directories	*Seclabel*
/usr and contents	SYSLOW
/usr/lpp and contents	SYSLOW
/usr/man and contents	SYSLOW

Assigning Security Labels to UNIX IPC Objects

When MLS is active, CA Top Secret assigns the security label of the process, if one exists, to the IPC security packet (ISP) at the time the ISP is created. Then, processes can only communicate with each other if their seclabels are equivalent.

Important! Once a security label has been assigned to an IPC object, it can never be changed.

If the MLS option to require security labels for UNIX IPC objects has been activated (MLIPCOBJ(YES)), all UNIX IPC objects must have security labels; otherwise, all accesses to these objects will be denied by CA Top Secret.

Assigning Security Labels to Users

In an MLS system, after defining and creating security levels, categories, and security labels, activating them, and assigning them to data and resources (classification), the security administrator must ensure that all users and work entering the system, including started tasks, batch jobs, processes, UNIX daemons, etc., are identified. Each user must be assigned a unique acid. In addition, all users entering the system are required to have security labels. A security administrator can assign security labels to users to give them the clearance they need to perform their work in the system.

The SECLABEL field of the USER acid record is used to assign security labels to users in an MLS environment. Depending on how you implement MLS at your site, you do not have to define a Seclabel for each system user. If MLS will impact only certain departments, or certain types of data, then most users can logon to CA Top Secret without specifying a security label and perform their jobs without any knowledge of MLS. The system will default a security label for the user.

- If you do not want users to specify a security label at logon, then use the following guideline:
- Add a default seclabel to users and specify the label the user will need to perform his tasks in the DFLTSLBL field. Make sure to specify the same value in the SECLABEL field. For example, if USER01 needs to read and write to data labeled, LABELA, add DFLTSLBL(LABELA) to the user acid record for USER01.

  ```
  TSS add(user01) seclabel(labela)
                  dfltslbl(labela)
  ```
- USER01 can access the labeled data sets and resources by entering his acid and password. No label is required.
- If you want each user to specify a security label at logon, then use the following guideline:
- Add Seclabel's to each user and, in the SECLABEL field, specify the labels the user will need to perform their tasks. Set the default label to the lowest label in the system, SYSLOW. For example, USER01 needs to read and write to data labeled, LABELA, LABELB, and LABELAAA. Then enter:

  ```
  TSS ADD(user01) SECLABEL(labela,labelb,labelaaa,syslow)
                  DFLTSLBL(syslow)
  ```
- USER01 can access the labeled data sets and resources by entering his acid and password and either security label, LABELA, LABELB, LABELAAA, or SYSLOW.

Note: If a security label is not specified by a user at signon and cannot be defaulted by the system from an existing acid record, SYSLOW will be assigned for the user.

The following table lists the default security labels that are recommended for special users in an MLS system.

User	Default Security Label	Reason for Security Label

User	Default Security Label	Reason for Security Label
CA Top Secret Started Task	SYSMULTI	
Console	SYSHIGH	
JES2 or JES3 Started Task	SYSMULTI	
OMVS Started Task	SYSMULTI	
Security Administrator	SYSHIGH	At least one SCA acid with the MLSADMIN attribute should be assigned the SYSHIGH security label.

User SECLABELs

In an MLS environment, security labels are added to user acid records. One or more seclabels may be specified. You must define SECLEVEL, CATEGORY, and SECLABEL data records in the MLS record before you can add a seclabel to a user.

Important! If you change or delete an existing security label (for example Seclabel data record) that has been assigned to users or resources, you may get unexpected results during MLS validation. Before changing or removing a security label from the system, check whether it has been assigned to any users or resources. If it has, confirm that the change or deletion is intended. If it is, make any necessary changes to user and resource Seclabel records that are using the security label. Likewise, if you delete a security level or category that is used in any existing security label, before removing the level or category from the system, confirm that the deletion is intended. If it is, make any necessary changes to existing security labels, and any user and resource Seclabel records that are using the security labels.

This command has the format:

```
TSS {add|remove|replace}{acid}
    SECLABEL(seclabel1,…seclabeln)
    DFLLSLBL(seclabel)
```

Seclabel(seclabel1,...seclabeln)

Specifies the security labels which a user is authorized to use when entering a system and that will be used during validation to determine whether acces to classified MLS data sets and resources will be allowed or denied. The *seclabel* value is the 1- to 8-character uppercased name of an existing MLS SECLABEL record segment that contains the security label data. You may assign more than one security label to a user, but only one label may be active at a time and used to validate MLS access to data sets and resources. If multiple security labels are assigned, any of these are available to the user to signon to a system. This field is required and cannot be masked. A comma or blank is the only valid delimiter between specified security label values. The system-defined security label SYSNONE is not valid for a user.

Dfltslbl(seclabel|SYSLOW)

Specifies the name of a security label that will be active and used to validate MLS access if a security label is not specified at system entry when MLS is active. The *seclabel* value is the 1- to 8-character name of an existing MLS SECLABEL Record segment that contains the security label data. This field is required. The default value is the system-defined label, SYSLOW, which is always the lowest security label defined by the system and will be dominated by all other security labels.

System-Defined Labels

CA Top Secret provides three system-defined, internal security labels that can never be directly created or modified by a user but can be assigned to users: SYSHIGH, SYSLOW, and SYSMULTI.

Add a SECLABEL to a User

To add a SECLABEL to a user, enter:

```
TSS ADD(usera) SECLABEL(label2)
               DFLTSLBL(syslow)
```

Note: Any security label specified in the record must be valid (defined in the system) for the record to be successfully added.

Remove a SECLABEL from a User

To delete a SECLABEL from a user, enter:

```
TSS REM(usera) SECLABEL(label2)
```

Establishing the MLS Environment

The MLS implementation team must revise the security policy to include MAC, which requires that all selected resources (objects) and users (subjects) be labeled. These security labels are used to determine access. In general, MLS (fully implemented, with all options activated) states that subjects can only read data at their security label or below and they can only write to data with a security label that equals theirs.

The CA Top Secret MLS control options determine the MLS environment. The MLS administrator is responsible for maintaining the security options that control the MLS environment and security labeling. After these options are set, the system enforces them and creates loggings when an MLS administrator attempts to change them. The following sections describe the options that can be set to establish and activate an MLS environment.

Physical Environment for Multilevel Security Preparation

Multilevel Security can only be activated if MLSBLOCKS have been allocated within the Security File and Backup Security File. If you have not allocated MLSBLOCKS, allocate a new security file, formatted with TSSMAINS. Copy your existing security file using TSSXTEND. The Backup Security File must be formatted with TSSMAINB.

MLS Options Definition

The MLS environment is defined by specifying the MLS control options or by accepting the default values. The following control options are supported: MLACTIVE, MLMODE, MLFSOBJ, MLIPCOBJ, MLWRITE, MLNAME.

MLS Related Control Options

MLACTIVE(YES|NO)

Specifies whether MLS is active on the system.

Default: NO

MLFSOBJ(YES|NO)

Specifies whether security labels are required for UNIX directories and files.

Default: NO

MLIPCOBJ(YES|NO)

Specifies whether security labels are required for UNIX IPC objects.

Default: NO

MLSECBYS(YES|NO)

Specifies whether security labels can be restricted for use on specific systems as specified in the SYSID field of theMLS SECLABEL data record.

Default: NO

Note: The system-defined security labels, SYSLOW, SYSHIGH, SYSMULTI, and SYSNONE, are always available on all systems regardless of whether this option is on or off.

MLWRITE(YES|NO)

Specifies whether writing data from a higher security label to a lower security label is allowed or prohibited in the system. When MLWRITE(NO) is set, write-down is prevented and new data is labeled automatically and internally by the system at the time it is created with the session security label of the user who first created the data. In addition, when MLWRITE(NO) is set, there may be some situations where a user may need to write-down. In these cases, a security administrator can authorize a user for “controlled write-down”. Controlled write-down lets a user enter the system with the ability to write-down by default (UPDATE access to the IRR.WRITEDOWN.BYUSER resource in the FACILITY class) or issue the TSS MLWRITE subcommand to set, reset or query write-down privilege (READ access to the IRR.WRITEDOWN.BYUSER resource in the FACILITY class). Controlled write-down is also available in a UNIX System Services environment through the UNIX **writedown** command.

Default: YES

Before enabling MLWRITE(NO), CA recommends that you successfully test the system in MLS WARN mode before implementing FAIL mode. There may be some system performance degradation when NOMLWRITE is set.

MLMODE(DORM|WARN|FAIL)

Defines a mode for MLS security, independent of the MODE for standard permissions. MLMODE determines how security reacts to an MLS validation rule violation. Permissions with ACTION(WARN|FAIL) have no effect on the interpretation or reaction of security to MLS validation rules. Valid modes are:

DORM

(Default) Validates security labels only at system entry. Logging is done for violations. No messages are returned to the console or to the user.

WARN

Permits MLS accesses to classified data sets and resources that normally would violate MLS validation rules and sends a warning message to the user (or the system log). Violations are logged.

FAIL

Prevents MLS accesses to classified data sets and resources based on MLS validation rules and sends an error message to the user (or the system log). Violations are logged.

Viewing MLS Control Options

To view the MLS Control Options, enter:

```
TSS MODIFY(STATUS(MLS))
```

Changing an MLS Control Option

To change the MLS control options use the TSS modify command from the console or the TSS TSO MODIFY command.

```
TSS MODIFY(MLACTIVE(YES)
```

Require Security Labels for UNIX Files and Directories

The MLFSOBJ option specifies whether security labels are required for UNIX directories and files in an MLS system. The default is MLFSOBJ(NO), security labels are not required for UNIX directories and files. This option should be turned on if you want to use MAC protection and security labels for UNIX files and directories.

Require Security Labels for UNIX IPC Objects

The MLIPCOBJ option specifies whether security labels are required for UNIX IPC objects, such as semaphores. The default is MLIPCOBJ(NO), security labels are not required for UNIX IPC objects. This option should be turned on if you want to use MAC protection and security labels for UNIX IPC objects.

Prohibiting Write-Down

The MLWRITE(NO) option in the control options is used to prevent declassification of data in an CA Top Secret MLS system. This is known as the “confinement property” or “*-property”. When this option is specified, writing data from a higher to a lower security label is not allowed. In addition, CA Top Secret will assign to data, at the time it is created, the session security label of the user who created the data. The security label assigned is stored in the MLS record. Once data has been labeled in this way, to reclassify it by assigning it a different security label, a security administrator must create an MLS resource record for the data set with the changed security label.

Examples: MLWRITE subcommand

The following examples illustrate use of the MLWRITE subcommand:

To display your controlled write-down setting, enter:

```
TSS MLWRITE(status)
```

TSS1409I WRITE-DOWN FOR USER M1USER IS: DISABLED

To activate controlled write-down for yourself, enter:

```
TSS MLWRITE(enable)
```

TSS1409I WRITE-DOWN FOR USER M1USER IS: ENABLED

To deactivate controlled write-down for yourself, enter:

```
TSS MLWRITE(disable)
```

TSS1409I WRITE-DOWN FOR USER M1USER IS: DISABLED

If MLS is not active on the system and the user issues the MLWRITE command, an error message will be displayed.

```
TSS MLWRITE(status)
```

TSS1408E MLWRITE COMMAND NOT AVAILABLE

If MLS is active on the system, but the control option is MLWRITE(YES) , (global write-down is allowed), and the user issues the MLWRITE command, an error message will be displayed.

```
TSS MLWRITE(status)
```

TSS1408E MLWRITE COMMAND NOT AVAILABLE

Activate Name-Hiding

Because catalog entries can be read by a job with any security label, users must be careful how they name their data sets. People tend to create data set names that help them to remember the contents of their data sets. This can also reveal more information than was intended. For example, a data set named PAYROLL.LAYOFF.PLANS listed in a catalog could fuel rumors and hurt employee morale, even if its contents could not be read. Users should be cautioned to choose their data set names carefully. If sensitive data set or file names must be protected, a security administrator can hide the names of data sets and files.

Activate “Controlled Write-Down”

When MLS is active on a system, the MLWRITE subcommand can be issued by an authorized user to override global write-down protection by setting, resetting or querying the setting of the write-down mode for the user's address space. When MLWRITE(NO) has been set as a control option to globally prevent writing data from a higher to a lower security label, and a user has READ access to the IRR.WRITEDOWN.BYUSER resource in the FACILITY class (using a resource rule), the user is authorized to issue the MLWRITE subcommand to:

- Set the mode to allow write-down during their current session
- Set the mode to prevent the ability to write-down for their current session,
- Display the write-down mode for their current session
- Reset the write-down mode for the current session to their default setting with which he entered the system.

However, if a user enters the system and has UPDATE access to the IRR.WRITEDOWN.BYUSER resource, CA Top Secret will, by default, allow the user to write-down without issuing the MLWRITE subcommand during their session, although the user can issue the command if they also have READ access to the IRR.WRITEDOWN.BYUSER resource. When the RESET operand on the MLWRITE subcommand is specified, CA Top Secret will reset the user write-down mode to the default value that was set at the time the user entered the system.

Note: A user who has the SECURITY privilege in their acid has READ and UPDATE access to the IRR.WRITEDOWN.BYUSER resource in the FACILITY class and is, therefore, authorized to write-down after successful system entry as well as issue the MLWRITE subcommand. However, it is recommended that security administrators disable “controlled write-down” for themselves after system entry to protect against unintentional declassification of data, unless the administrator needs to exercise this privilege.

The MLWRITE command has the following syntax:

```
MLWRITE [STATUS | ENABLE | DISABLE ]
```

STATUS

Displays the status of the write-down mode during the user's current session. This is the default when no parameters are specified with the MLWRITE command.

ENABLE

Specifies that a user's write-down mode be turned on and displays the status of the user's ability to write-down during the current session as 'ENABLED'.

DISABLE

Specifies that a user's write-down mode be turned off and displays the status of the user's ability to write-down during the current session as 'DISABLED'.

To allow a user to control write-down for himself by issuing the MLWRITE subcommand, a security administrator should do the following:

- Activate MLS (MLACTIVE(YES))
- Specify the MLWRITE(NO) control optoin
- Determine which users may need to write data from a higher to lower security classification when MLS is active on the system and write-down is protected globally with the MLWRITE(NO) option.
- Create resource rules for the IRR.WRITEDOWN.BYUSER resource in the FACILITY class, giving READ or UPDATE access to those users who need to write down. For example:

```
TSS PER(mluser1) IBMFAC(irr.writedown.byuser access(read)
TSS PER(mluser2) IBMFAC(irr.writedown.byuser access(update)
```

Restrict Security Labels to Specific Systems

When MLS is active, it is possible to separate work in a SYSPLEX based on security labels. This may be useful if you want to run all work with one security label (LABELA) on one system while running all work with another security label (LABELB) on another system in a sysplex while sharing the CA Top Secret database. The MLSECBYS(YES|NO) control option determines whether seclabels are restricted for use on specific systems as specified in the SYSID field of MLS seclabel records.

To restrict the use of a security label to one or more systems, you must do the following:

- Define a security label in the MLS record and specify the system IDs to which the security label should be restricted in the SYSID field. This is known as a "system-specific" security label. If no system IDs are specified, the security label is available for use on all systems. For more detailed information on how to create an MLS SECLABEL Record, see the section on Defining Security Labels.
- The following CA Top Secret record restricts the use of LABELA to system SYSA in a sysplex:

```
TSS ADD(mls) SECLABEL(labela)
             SECLEVEL(50)
             CATEGORY(development)
             SYSID(sysa)
```

  If a user tries to signon to an MLS system other than SYSA with security label, LABELB, the access will be denied. Likewise, if a user is signed onto a system other than SYSA and tries to access a resource classified with LABELB, the access will be denied, if MLMODE(FAIL) is set as the global MLS mode. Finally, JES2 will ensure that any jobs with LABELA are submitted and executed on the correct system, SYSA.

Note: The system-defined security labels, SYSLOW, SYSHIGH, SYSMULTI, and SYSNONE, are always available on all systems regardless of whether the MLSECBYS option is turned on or off.

Important! JES3 does not support system-specific security labels. In addition, JES2 also does not support system-specific security labels for systems that do NJE and OFFLOAD processing.

Change the MODE Setting

The MLMODE control option lets you change the MLS component mode without IPLing the system.

Activating MLS in DORM Mode

After setting up your MLS environment, including defining options, security levels, categories (if used) and security labels, and assigning security labels to those users and resources that require classifications, you are ready to activate MLS on your system in DORM mode to begin phasing in and testing MLS before fully activating it in MLS mode.

To activate MLS in DORM mode, issue the following commands:

```
F TSS,MLMODE(dorm)
```

```
F TSS,MLACTIVE(yes)
```

If the MLS environment is set in QUIET mode at the time MLS is activated, security labels will only be checked at system entry to allow or deny access to the system. If access to the system is denied, logging occurs and DAC validation is not performed. If MLS allows access to the system, DAC validation is performed and will either allow or deny the access.

Starting your implementation in DORM mode provides you time to set up your MLS environment and train users on how to use the security labels that you create and assign to them.

Testing MLS in DORM Mode

You can test User Seclabels and system entry validation by having users log off the system and logon again using the security labels they are authorized to use. CA Top Secret permits or denies system access based on the security label provided or defaulted at logon.

Activating MLS in WARN Mode

After you are satisfied that users can access the system using labels, you must test whether users can access the data sets that they must to perform their jobs. After you have tested system access in DORM mode, you can migrate to WARN mode. To migrate to WARN mode, use the following procedure.

To activate MLS in WARN mode, enter:

```
TSS MODIFY(mlmode(warn))
```

If the MLS environment is set in WARN mode at the time MLS is activated, security labels will be checked at system entry as well as when access to any classified resource is requested. The system will log and permit MLS accesses to classified resources that would normally violate MLS validation rules and issue a warning message to the user, and DAC validation will then be performed to allow or deny the access.

Testing MLS in WARN Mode

After your MLS system is running in WARN mode, you should test data set and resource accesses by doing the following:

- Ask users to log on with their labels and try to access labeled data sets and resources at their label or below.
- Ask select users to log on with a low label and try to access higher labeled data that they are authorized to access when logged on with the appropriate label.
- Ask users to send messages to other users at their same label and to lower labels.
- Monitor user feedback and CA Top Secret reports until you are satisfied that your site security policies are being enforced.

Fine-tuning MLS in WARN Mode

After your MLS system has been running in WARN mode, you may need to make adjustments to the settings in the various MLS records. For example, you can decide to test your security policy by implementing MLS for a select group of data sets. You might decide to relabel some data sets or resources that are incorrectly labeled.

Migrating MLS to FAIL Mode

Follow the same procedure you used to migrate to WARN mode.

If the MLS environment is set in FAIL mode at the time MLS is activated, security labels will be checked at system entry as well as when access to any classified resource is requested. The system will log and prevent MLS accesses to classified resources that violate MLS validation rules and DAC validation will not performed if MLS access was denied.

Deactivating MLS

After establishing an MLS environment, if you need to deactivate MLS for any reason, you must be sure that doing so will not compromise any protected data and resources in the system.

MLS may be deactivated from the console or from TSO with the following commands

```
F TSS,MLACTIVE(no)
```

```
TSS MODIFY(mlactive(no))
```

Monitoring MLS

The following TSS commands can be used to display MLS-related information online in CA Top Secret systems:

- WHOAMI
- MLWRITE
- STATUS(MLS)
- LIST(MLS)

WHOAMI Command

The WHOAMI command is used to display your current active security label in an MLS environment. This security label is the one with which you entered the system and endures for the duration of your session. It cannot be used to display any user's security label other than your own.

MLWRITE Command

When MLS is active on a system, and write-down is globally protected, the MLWRITE subcommand can be issued by an authorized user not only to set or reset the user's controlled write-down privilege, but also to display it. However, it cannot be used to display any other user's controlled write-down privilege.

To display your controlled write-down privilege, enter:

```
TSS MLWRITE
TSS1409I WRITE-DOWN FOR USER USERA IS: DISABLED
```

F TSS,STATUS(MLS)

The settings of the MLS control options can be displayed using the TSS modify command from an operator console or from a TSO session.

LIST MLS Command

The LIST MLS subcommand displays the MLS security classifications defined on the system.

```
LIST(MLS) SECLEVEL(level|ALL)
CATEGORY(categoryname|ALL)
SECLABEL(labelname|ALL)  DATA(ALPHA|hi-lo|lo-hi|active)
RESOURCE(ALL) | resclass(entity) | resclass(ALL)
```

SECLEVEL

Displays the defined security levels in numerical order. To display security levels, you must have the MLSADMIN authority in your acid.

CATEGORY

Displays the defined categories in alphabetical order. To display categories, you must have the MLSADMIN privilege in your acid.

SECLABEL

Displays the defined security labels in alphabetical order by default. Security labels may also be displayed in high-to low (or low-to-high) order by security level. The DATA(ACTIVE) option will display only the security labels that are active on the system from which the command was issued. To display security labels, you must have the MLSADMIN privilege in your acid.

Auditing MLS

The following section discusses auditing and the TSSUTIL report program.

Checking Authorization

If the SYS1.MANx SMF data sets are MLS-protected in your system, you must sign on with an MLS security label that dominates that of the SYS1.MANx SMF data sets to read the data from them. Since the SYS1.MANx data sets are generally classified with the highest label in the system, your signon security label will generally be SYSHIGH.

If MLS validation allows access to the SYS1.MANx data sets, CA Top Secret will perform subsequent DAC checks of whether the user submitting the utility is authorized to view or manipulate the input SMF data.

TSSUTIL Report Generator

The TSSUTIL report generator processes the SMF or Audit/Tracking File records issued by CA Top Secret to provide an updated activity report for system entry requests and requests to access resources. When MLS is active, the following new fields are captured whenever an unauthorized attempt is made to access a classified data set:

USER SECLABEL

The 8-byte user session seclabel

RSRC SECLABEL

The 8-byte resource seclabel

Tracing SAF Requests

The SECTRACE command lets you trace any security request made to the System Authorization Facility (SAF). SAF is a z/OS component that permits resource managers of the operating system to request security services. SAF's primary function is to route security information between the application and the security product that resides with the z/OS software. Any program using SAF automatically interfaces with CA Top Secret because CA Top Secret translates SAF security requests into CA Top Secret requests.

Tracing UNIX System Services (OMVS)

The SECTRACE facility can be used to trace SAF requests made by OMVS.

When MLS is active on a system, the following MLS-related data appears in the OMVS SECTRACE output:

FSP SECLABEL=security label

The 8-byte file or directory security label or, *NONE*, if no security label exists

USER SECLABEL=security label

The 8-byte session security label or, *NONE*, if no security label exists

Chapter 4: Configuring a Multilevel Secure System

This section contains the following topics:

Introduction to Configuration

When MLS is active on an CA Top Secret system, MLS will behave differently depending on what options have been set to establish the MLS environment. In its most basic implementation, a security administrator can assign a security label to only those users and resources that need enhanced security protection, rather than to all users and resources. In its most complex implementation, a security administrator can assign security labels to all users and most resources in the system while protecting declassification of data. This is very stringent MLS. To be effective, MLS should be implemented based on the specific security requirements of your organization.

Hardware Configuration

The IBM z/OS operating system has features that protect the CPU storage area from unauthorized access. The following hardware configurations are permitted:

- Uniprocessors
- Multiple system complexes

Software Configuration

Many components make up a z/OS system. Because some components depend on macros from other components, the installation sequence is important. The following list details one sequence that we recommend for CA Top Secret MLS system software installation, although other installation sequences work equally well:

- z/OS V1R5
- JES2
- JES3
- DFSMS
- TSO/E
- VTAM
- PSF
- ISPF
- TCP/IP
- z/OS UNIX System Services (UNIX)
- CA Top Secret
- CA Examine r3.5

Restrictions

If your site shares DASD, the following restrictions apply:

- All z/OS systems must be operating at z/OS V1R5 or higher
- z/OS MLS systems should not share DASD with systems that are not MLS.
- All z/OS MLS systems must share the same CA Top Secret databases
- All z/OS MLS systems in the global resource serialization complex must be the same set of z/OS systems that are sharing the CA Top Secret databases.
- The JES2 complex must be the same set of z/OS systems that are sharing the CA Top Secret databases or subset of these systems.

DFSMSdfp

DFSMS*dfp* controls storage on DASD and tape volumes for the system. DFSMS*dfp* communicates information between the processor and the storage devices to provide data, device, program, and storage management activities.

Support for MLS

The following is supported when MLS is active on an CA Top Secret system:

- Control access to data on DASD
- Control access to data on tape
- Control access to temporary data sets
- Protect catalogs
- Protect the *DFSMSdfp* subsystem

Restrictions

The following restrictions apply when MLS is active on an CA Top Secret system:

DASDVOL class

Do not activate the DASDVOL class. Users with DASDVOL authority to a volume can access its data sets without being restricted by DAC rules.

DASDVOL authority is necessary when using the AMASPZAP service aid to modify a Volume Table of Contents (VTOC) on a disk pack. This operation takes the system out of an MLS configuration, and should be done only under controlled conditions, and with only trusted users on the system.

CVOLs and VSAM catalogs

Do not use CVOLs or VSAM catalogs. Only Integrated Catalog Facility (ICF) catalogs should be used in an MLS system. The following steps prevent the use of CVOLs and VSAM catalogs:

- Write DAC access rules to allow only authorized users write access to the master catalog. This prevents unauthorized users from using the IMPORT CONNECT command to connect VSAM catalogs to the master catalog, or from using the DEFINE ALIAS command to connect CVOLs to the master catalog.
- When defining new user catalogs with the DEFINE USERCATALOG command, be sure to specify the ICFCATALOG keyword.
- Do not use the DEFINE ALIAS command to connect CVOLs to the master catalog. When connecting user catalogs to the master catalog, using the IMPORT CONNECT command, make sure the catalogs are ICF catalogs. (The LISTCAT command will tell you.)

Configuration Checklist

This checklist describes the software configuration requirements when MLS is active on an CA Top Secret system.

Requirement	Complete
Control access to data on DASD	□
Control access to data on tape	□
Control access to temporary data sets	□
Protect ICF catalogs	□
Assign security label to catalogs	□
Write access rules to control access	□
Activate name-hiding (optional)	□
Protect the DFSMS subsystem	□

Controlling Access to Data on DASD

In an MLS system, data stored on DASD devices is secured by protection provided by MLS resource records.

Controlling Access to Data on Tape

In an MLS system, data stored on tape is secured by protection provided by MLS resource records.

Controlling Access to Temporary Data Sets

In an MLS system, access restrictions apply to temporary data sets. A temporary data set is a special data set created and deleted in the same job. Unlike an ordinary (non-temporary) data set, it is not cataloged and has a system-generated name. Only the job that creates a temporary data set can access it for read, write or scratch purposes. In an MLS system, temporary data sets must be protected from unauthorized access and disclosure. The security administrator must do the following:

- Define procedures for processing temporary data sets
- If necessary, write access rules to control access

A job can always access its own temporary data sets, and in general, other jobs cannot. When a job ends, its temporary data sets are automatically deleted by the system. However, there are some cases where data sets may not be deleted:

- System failure
- Initiator failure or initiator termination by the FORCE command
- Automatic restart

If access to temporary data sets were restricted to just the creating job, these leftover data sets would never be deleted, and would stay around forever, taking up valuable space. To prevent this, it is necessary to allow selected authorized users access to these data sets, so they can be deleted. For this reason, users with the NODSNCHK attribute in their acids can access temporary data sets that they did not create. A logging record is created for each access.

Protecting Integrated Catalog Facility Catalogs

In an MLS system, a site should protect its ICF catalogs using MAC and DAC mechanisms.

Assigning Security Labels to Catalogs

When write-down is protected on an MLS system, a security administrator should assign security label, SYSNONE, to all ICF catalogs. This enables a user logged on with any security label to access the catalog based on the DAC access rules.

Access Rules for Catalogs

A security administrator must write access rules to control access to the catalogs. The security administrator must write access rules for the master catalog and the user catalogs. All system users should be given read access to the master catalog and only a limited number of users should be allowed to write to the master catalog. Below is a sample command:

```
TSS PER(ALL) DSN(CATALOG.MASTER)
             ACCESS(READ)

TSS PER(SYSADM) DSN(CATALOG.MASTER)
                ACCESS(UPDATE)
```

CATALOG.MASTER

The first and second-level qualifiers of the data set name of the master catalog

SYSADM

A user authorized to update the master catalog

This rule permits all users to update user catalogs named CATALOG.-.

Activating Name-Hiding

The MLNAME control option activates name-hiding on an CA Top Secret system.

MLNAME(YES|NO)

Specifies whether the names of data sets, files, and directories are protected from disclosure to users who do not have at least READ access to the data. Rule validation, and, if MLS is active, security label checking, will be performed to allow or prevent the user from viewing the names of data sets in a catalog or on a VTOC, and files and directories in listing the contents of a UNIX directory. However, if a user requests to view the name of a specific data set, file, or directory, the names will appear but the user may not be able to access the data. Name hiding can be used on a system where MLS is not active. However, MLS must be active to support name hiding for UNIX files and directories. The default is MLAME(NO), do not hide the names of data sets, files and directories from users.

When name-hiding is active, users are prevented from seeing the names of data sets to which they do not have at least READ access in a catalog, unless the exact name of the data set is specified.

Note: When a user issues an =3.4 in ISPF to access data sets, and does not specify a volume serial number, catalog processing is performed.

When name-hiding is active, users are prevented from seeing the names of data sets to which they do not have at least READ access on a VTOC when directly reading the VTOC or VTOC index or using CCHHR for CVAF reading of a VTOC.

Name hiding for Data Sets:

- If MLS is inactive, only access and resource rule validation will be performed to allow or prevent the user from viewing the names of data sets in a catalog or on a VTOC.
- If MLS is active, a MAC label dominance check will be done for each data set name in a catalog or on a VTOC to determine whether a user can see it. If the MAC check allows the access, access and resource rule validation is performed to ultimately allow or prevent the user from viewing the names of data sets in a catalog or on a VTOC.

Name hiding for UNIX Files and Directories:

- If MLS is active, a MAC label dominance check will be performed for each file and directory name to determine if it can be viewed by the user who issues the command to list the contents of a directory.
- If MLS is inactive, name hiding is not supported for UNIX files and directories in CA Top Secret.

Important! Name hiding is not available in an HFS file system; only in a zFS file system.

Note: Name hiding degrades the performance of a system. When name hiding is active in an MLS environment, system performance is further degraded. Do not activate name hiding if any system sharing the CA Top Secret database does not meet the minimum software requirements for MLS support. Use of the name hiding option should not cause problems on these systems, but it does not provide full protection on these systems. You must be operating at z/OS R1V5 or above to activate name hiding in an CA Top Secret system.

Protecting DFSMSdfp Subsystem

DFSMSdfp is the storage management system for z/OS MVS. It enables a site to centralize the management of external storage. The storage administrator uses the Interactive Storage Management Facility (ISMF) to implement a site's storage management policy. Through a combination of automatic class selection (ACS) routines and resource rules to protect the STORCLAS, MGMTCLAS, and PROGRAM classes, *DFSMSdfp* and CA Top Secret provide protection for *DFSMSdfp* functions.

CA Top Secret

The following is supported when MLS is active on an CA Top Secret system:

- MAC protection
- DAC protection
- Object reuse protection
- Identification and authentication
- Auditing

Restrictions

If you are running any of the following software, it must meet the following minimum requirements to run concurrently with CA Top Secret:

- CA Top Secret Security Option for DB2 r1.2
- CICS/ESA Release 4.1 (includes CICS Transaction Server for z/OS and OS/390 Version 1 Release 1)
- IMS/ESA Version 6 Release 1
- CA Common Services for z/OS r1, Genlevel 9901

If your site shares DASD, the following restrictions apply:

- All z/OS systems must be operating at z/OS 1.5 or higher
- z/OS MLS systems should not share DASD with systems that are not MLS
- All z/OS MLS systems must share the same CA Top Secret databases
- All z/OS MLS systems in the global resource serialization complex must be the same set of z/OS systems that are sharing the CA Top Secret databases
- The JES2 complex must be the same set of z/OS systems that are sharing the CA Top Secret databases or a subset of these systems

Configuration Checklist

This checklist describes the software configuration requirements when MLS is active on an CA Top Secret system.

Requirement	Complete
Provide DAC controls	□
Control options	□
DAC control mechanisms	□
Resource rules	□
Provide accountability controls	□
Do not use UADS	□
Identify all system users to CA Top Secret	□
Do not reuse acids	□
Define required acids	□
Identify users with special privileges	□

Requirement	Complete
Specify PSWD control options	□
Provide MLS controls	□
Define the MLS control options	□
Define MLS SECLEVEL Records	□
Define MLS CATEGORY Records	□
Define MLS SECLABEL Records	□
Assign Security Labels to Users	□
Assign Security Labels to Objects	□
Assign Security Labels to DB2 Objects	□
Protect object reuse	□

DAC Control Mechanisms

CA Top Secret provides several mechanisms to achieve DAC controls. Global control options establish site options for your CA Top Secret system. The security administrator must ensure that the system provides controlled access to data sets using ownership and permissions.

Providing Accountability Controls

This section describes the following identification and authentication controls that you should configure in an MLS system:

- Do not use the user attribute data set (UADS)
- Identify all system users to CA Top Secret
- Do not reuse acids
- Define required acids
- Identify users with special privileges
- Specify password control options

Do Not Use UADS

If your site is currently using UADS, CA Top Secret provides a conversion utility.

Identifying All System Users

A security administrator must create a unique acid record for each system user. The account manager assigns various privileges to each user based on the tasks the user must perform.

Do Not Reuse Acids

Acids should never be reissued to different people. When this is done, it makes it very difficult to determine from security logs who the person responsible for an action was and it then becomes necessary to keep a log of who owned each acid at various times. This is error-prone, and can be avoided by simply not reissuing acids.

Defining Required Acids

You should define the following CA Top Secret acids that are used by system-started tasks: INIT, JES2 or JES3, LLA, VLF and other system address spaces.

In addition, you should assign security label, SYSHIGH, to each of these acids by adding the seclabel to the acid record.

Providing MLS Controls

This section describes the required settings of the following MLS records:

- Define the MLS control options
- Define MLS SECLEVEL Records
- Define MLS CATEGORY Records
- Define MLS SECLABEL Records
- Assign Security Labels to Users
- Assign Security Labels to Objects (MLS resource records)
- Assign Security Labels to DB2 Objects (MLS resource records)

Defining the MLS Control Options

This section describes the ML control option field settings that the security administrator should select to fully implement an MLS system.

- MLACTIVE
- MLFSPOBJ
- MLIPCOBJ
- MLSECBYS
- MLWRITE
- MLMODE

MLMODE(FAIL) causes CA Top Secret to deny any accesses that would violate the MAC simple security and confinement properties.

Defining MLS SECLEVEL Records

The security administrator should create MLS SECLEVEL records for all security levels the site wants to use. The security administrator must specify a unique level number for each SECLEVEL record.

To define a security level with a value of 50 and the descriptive name SECRET, enter:

```
TSS ADD(mls) SECLEVEL(50)
             LVLNAME(secret)
```

Defining MLS CATEGORY Records

The security administrator should create MLS CATEGORY records for all categories the site wants to use, if any. (Categories are optional in an MLS system.) The security administrator must specify a unique name for each CATEGORY record.

Examples: MLS CATEGORY definition

To define a category with a value of HUMANRESOURCES, enter:

```
TSS ADD(mls) category(humanresources)
```

Defining MLS SECLABEL Records

The security administrator should create SECLABEL records for all the security labels the site wants to use. Security labels, SYSHIGH, SYSLOW, SYSMULTI, and SYSNONE, are already provided with the system and do not need to be defined. The security administrator must specify a unique name for each SECLABEL record.

Example: MLS SECLABEL resource

To define a security label with a security level of 50 and the category HUMANRESOURCES, enter:

```
TSS ADD(mls) SECLABEL(labela)
             SECLEVEL(50)
            CATEGORY(humanresources)
```

Assigning Security Labels to Users

One or more security labels and a default security label may be added to users. The security administrator should create add these fields to users who need access to data or resources that are classified with security labels. The security administrator should also delegate the privilege to reclassify data sets to any users who need this capability.

Example: assign security label to DB2 object

To assign the security label LABELA to the DB2 table resource TEST.QEWRQER.ASDF.TESTCOLUMN3, issue the following commands:

```
TSS ADD(mls) DB2TABLE(test.qewrqer.asdf.testcolumn3)
             SECLABEL(labela)
```

Assigning Security Labels to DB2 Objects

DB2 resources may be protected with security labels The security administrator should create MLS resource records for any DB2 resources that are deemed sensitive in an organization, that if left unprotected, might result in data disclosure or declassification.

Examples: assign security label to an object

To assign the security label LABELA to the CICS transaction CEMT, enter:

```
TSS ADD(mls) OTRAN(cemt)
             SECLABEL(labela)
```

To assign the security label SYSHIGH to the SYS1.MANx data sets, enter:

```
TSS ADD(mls) DSN(sys1.man)
             SECLABEL(syshigh)
```

CA Examine

CA Examine is an optional component of an MLS configuration.

CA Examine can help you determine if your z/OS system is properly configured along with identifying possible integrity exposures on your system.

Do not make CA Examine APF authorized. CA-Examine is not intended to be marked APF authorized in an MLS configuration. It does not install authorized, and must not be made authorized after the fact.

Configuration Checklist

This checklist describes the software configuration requirements when MLS is active on an CA Top Secret system.

Requirement	Complete
Install ISPF/PDF	□
Protect CA Examine libraries	□
Use CA Examine to verify proper configuration	□

Installing ISPF/PDF

CA Examine requires the services of ISPF/PDF: therefore, if CA Examine is installed, you must install ISPF/PDF.

Protecting CA Examine Libraries

Users typically concatenate their own ISPF/PDF CLIST, panel, skeleton, and message libraries to tailor the way ISPF/PDF works to meet their needs. In most cases, this is perfectly acceptable in an MLS configuration. There is, however, one case when it is not acceptable. This is when a security administrator is using CA Examine to verify the proper configuration of the system. In this case, the ISPF/PDF libraries, CA Examine libraries, and CA Top Secret ISPF libraries must be concatenated in front of any user libraries.

Example

The following example shows how the JCL for this concatenation would look in a TSO LOGON procedure:

```
//ISPPLIB       DD DSN=ISP.V3R5M0.ISPPENU,DISP=SHR       ISPF panels
//              DD DSN=ISP.V3R5M0.ISRPENU,DISP=SHR       PDF panels
//              DD DSN=CAI.CAISPP,DISP=SHR               CA Top Secret panels
//              DD ...                                   User panels
//*
//ISPMLIB       DD DSN=ISP.V3R5M0.ISPMENU,DISP=SHR       ISPF messages
//              DD DSN=ISR.V3R5M0.ISRMENU.DISP=SHR       PDF messages
//              DD DSN=CAI.CAISPM,DISP=SHR               CA Top Secret messages
//              DD DSN=CAI.EXAMINE.MESSAGES,DISP=SHR    CA Examine messages
//              DD ...                                   User messages
//*
//ISPSLIB       DD DSN=ISP.V3R5M0.ISPSENU,DISP=SHR       ISPF skeletons
//              DD DSN=ISR.V3R5M0.ISRSENU,DISP=SHR       PDF skeletons
//              DD DSN=CAI.CAISPS,DISP=SHR               CA Top Secret skeletons
//              DD ...                                   User skeletons
//*
//ISTPLIB       DD DSN=ISP.V3R5M0.ISPTENU,DISP=SHR       ISPF tables
//              DD D5N=ISP.V3R5M0.ISRTENU,DISP=5HR       PDF tables
//              DD DSN=CAI.EXAMINE.TABLES,DISP=SHR      CA Examine tables
//              DD ...                                   User tables
//*
//SYSPROC       DD DSN=ISR.V3R5M0.ISRCLIB,DISP=SHR       PDF CLISTs
//              DD DSN=CAI.CAICLIB,D1SP=SHR              CA Top Secret CLISTs
//              DD DSN=CAI.EXAMINE.CLIST,DISP=SHR       CA Examine CLISTs
//              DD ...                                   User CLISTs
//*
```

These libraries must be protected with CA Top Secret access rules so only system maintenance personnel can update them. In an MLS system, if the option to require security labels is activated, they should be labeled "SYSLOW" so they are accessible to all users.

Using CA Examine to Verify Proper Configuration

CA Examine can help you determine if your z/OS system is properly configured. It shows you various facets of z/OS by way of interactive, easy-to-read screens. Its batch facility makes it possible to save scripts of examinations, and run them periodically as batch jobs to ensure that the configuration has not changed. The following list details examples of what CA Examine can display:

- The SVC Analysis option can show if you have any user SVCs, which are forbidden in an MLS system configuration.
- The Appendages option shows if you have any user I/O appendages, which are forbidden in a mulilevel-secure system configuration.
- The PARMLIB display takes the work out of determining your system options, by displaying for each member a list of other members that it points to. Members can be selected from the list for browsing.
- The JES2 Options and SMF Options displays show you the options that are in effect for these subsystems.
- The APF library display shows you all your APF-authorized libraries. This makes it easy to ensure that they are properly protected by access rules.
- The operator console display shows how your consoles are configured.
- The Freezer option saves a checksum of selected libraries. It can be rerun periodically to ensure that system libraries have not been changed.

All these functions and many more are described in the CA Examine documentation.

Interactive System Productivity Facility (ISPF)

ISPF/Program Development Facility (PDF) is an optional component in an MLS system configuration. In particular, CA Examine uses ISPF services to display and control its dialogs. If you are using CA Examine in an MLS system, you must also install ISPF/PDF.

Executed under a TSO/E subsystem session as an unauthorized program, ISPF is a dialog manager. A dialog is a “conversation” between a person using an interactive display terminal and a computer executing a program for a particular application.

Support for MLS ISPF

The following is supported when MLS is active on an CA Top Secret system:

- ISPF/PDF provides services for dialogs to support full screen panels, message display, table storage, and skeleton JCL use.
- ISPF dialogs can be written in many programming languages and scripting languages (CLIST and REXX). ISPF/PDF provides a subroutine call for these languages to use the ISPF dialog services.
- ISPF/PDF provides a set of utility dialogs to: browse, edit, allocate, rename, delete, copy, move, print, compare, and list files.

Restricting Jobs to Specific Systems

A security administrator can restrict security labels to specific systems in a sysplex by defining security labels that can only be used on those systems to which the security label has been defined. Specifying one or more system IDs in the SYSID field of the SECLABEL Record and activating the MLSECBYS control option does this.

Note: If the SYSID field is excluded from the record, then the security label can be used on all systems.

When a security label is restricted to one or more systems, JES2 will ensure that a job that is using the security label is executed only on a system on which that security label is defined and active. This allows sharing of the CA Top Secret databases in a sysplex while keeping work segregated to different systems. If the security label of a job is not defined and active on any system, the job will remain in the conversion phase.

For more information about how to define and use “system-specific” security labels, see the “Implementing and Administering a Multilevel-Secure System” chapter. See also the manual, IBM *JES2 Introduction,* for more information about the conversion phase in JES2 processing.

Restrictions

If ISPF/PDF is used, the following restrictions apply when MLS is active on an CA Top Secret system:

- Do not make ISPF APF authorized
- Protect ISPF administration libraries
- Do not install ISPF session manager exits

Configuration Checklist ISPF

This checklist describes the software configuration requirements when MLS is active on an CA Top Secret system.

Requirement	Complete
Do not make ISPF APF authorized	□
Protect ISPF administration libraries	□
Do not install ISPF session manager exits	□

Protecting ISPF Administration Libraries

It is typical for users to concatenate their own ISPF CLIST, panel, skeleton, and message libraries, in order to tailor the way ISPF works to meet their needs. In most cases, this is perfectly acceptable in an MLS configuration. There is, however, one case when it is not acceptable. This is when a security administrator is using the CA Top Secret ISPF panels to administer security. In this case, the ISPF libraries and the CA Top Secret ISPF libraries must be concatenated in front of any user libraries.

Example: ISPF libraries

This example shows the JCL for how the ISPF libraries and CA Top Secret ISPF libraries concatenation would look in a TSO LOGON procedure:

```
//ISPPLIB       DD DSN=ISP.V3R5M0.ISPPENU,DISP=SHR      ISPF panels
//              DD DSN=ISP.V3R5M0.ISRPENU,DISP=SHR      PDF panels
//              DD DSN=CAI.CAISPP,DISP=SHR              CA Top Secret panels
//              DD ...                                  User panels
//*
//ISPMLIB       DD DSN=ISP.V3R5M0.ISPMENU,DISP=SHR      ISPF messages
//              DD DSN=ISR.V3R5M0.ISRMENU.DISP=SHR      PDF messages
//              DD DSN=CAI.CAISPM,DISP=SHR              CA Top Secret messages
//              DD ...                                  User messages
//*
//ISPSLIB       DD DSN=ISP.V3R5M0.ISPSENU,DISP=SHR      ISPF skeletons
//              DD DSN=ISR.V3R5M0.ISRSENU,DISP=SHR      PDF skeletons
//              DD DSN=CAI.CAISPS,DISP=SHR               CA Top Secret skeletons
//              DD ...                                  User skeletons
//*
//ISTPLIB       DD DSN=ISP.V3R5M0.ISPTENU,DISP=SHR      ISPF tables
//              DD D5N=ISP.V3R5M0.ISRTENU,DISP=5HR      PDF tables
//              DD ...                                  User tables
//*
//SYSPROC       DD DSN=ISR.V3R5M0.ISRCLIB,DISP=SHR      PDF CLISTs
//              DD DSN=CAI.CAICLIB,D1SP=SHR             CA Top Secret CLISTs
//              DD ...                                  User CLISTs
//*
```

These libraries must be protected with CA Top Secret access rules so only system maintenance personnel can update them. In an MLS system, if the option to require security labels is activated, they should be labeled, SYSLOW, so they are accessible to all users.

Note: If CA Examine is included in the configuration, the CA Examine libraries must also be concatenated before any user libraries. See the Protect CA-Examine Libraries section for an example of the library concatenations with CA-Examine.

Note: Do Not Install ISPF Session Manager Exits. ISPF includes exit routines for SVC 93 (TGET/TPUT/TPG) and SVC 94 (STCC) to allow the session manager to be invoked under ISPF, instead of the more usual case of invoking ISPF under the session manager. These exits should not be installed in an MLS environment. For more information about these exits, see the *IBM z/OS TSO/E Customization* manual.

Configuring Network Job Entry (NJE) and Remote Job Processing (RJP)

If you want to successfully use NJE and RJP in an CA Top Secret MLS system, configure them as follows:

- Assign a security label and write resource rules for each NJE and RJP input device in the JESINPUT resource class. NJE or RJP input devices are not multi-label devices-they can only handle data with the same security label. All work coming to a JES2 input device is assumed to have the same security label as the JES2 input device.
- Assign a security label to each NJE and RJP printer in the WRITER resource class. The security label of the printer must dominate the security label of the work for JES2 to transmit the work to it.

JES2

JES2 uses the system authorization facility (SAF) to pass security information about jobs and resources to CA Top Secret. CA Top Secret makes access decisions based on information in its databases and passes its decision back to JES2.

Support for MLS

The following is supported when MLS is active on an CA Top Secret system:

- Control the use of JES2 operator commands
- Control access to JES2 spool data sets
- Control access to JES2 system data sets
- Audit the use of all JES2 operator commands and access to JES2 data sets
- Control submission of jobs through JES2 input devices
- Restrict jobs to specific systems based on security labels

In addition, CA Top Secret provides additional support beyond MLS requirements. You can control what data is output to a particular device and restrict certain users to specific output devices.

Restrictions

Certain JES2 functions should not be permitted in an MLS system when certain MLS options have been activated. The following restrictions apply when MLS is active on an CA Top Secret system:

- The network job entry (NJE) and remote job entry (RJE) functions can be used, but they must be configured properly.
- The only permissible output devices should be page printers, controlled by the Print Services Facility (PSF), and operated in deferred-printing mode, and line printers, operated as single-label devices, and labeled through procedural means. See the Print Services Facility (PSF) section for more information about printer restrictions.
- No site-written routines should be permitted in JES2 libraries, nor should modifications to JES2 routines be permitted.
- Entry of system commands through the input job stream is controlled by acid, just as when commands are entered from an operator console.
- Do not use the JES2 spool offload facility

Configuration Checklist

This checklist describes the software configuration requirements when MLS is active on an CA Top Secret system.

Requirement	Complete
Control the use of JES2 operator commands	□
Protect JES2 Spool Data Sets	□
Define acid for JES2 started task	□
Assign security label SYSMULTI to the JES2 started task ID	□
Define access rules for JES2 started task	□
Control job input	□
Configure Network Job Entry (NJE) and Remote Job Processing (RJP)	□
Restrict jobs to specific systems	□

Control the Use of JES2 Commands

The security administrator must be able to audit all JES2 commands in an MLS system. It is also necessary to control who can issue commands, since it is possible to issue commands not only from an operator console, but also from batch JCL. In either case, access is validated based on the acid associated with the job.

To control JES2 commands and provide an audit trail for all JES2 commands:

- Enable the protection of operator commands
- Write resource rules to protect JES2 commands

JES2 Command Resource Names

JES2 commands have resource names that follow the example below:

jesname.command[.qualifier]

jesname

The name of the JES2 system requesting the command validation

command

The name of the JES2 command

qualifier

The type of object the command specifies, such as JOB or SYS.

See the IBM z/OS *JES2 Initialization and Tuning Guide* to determine the resource name of the JES2 command. It provides a list of JES2 commands, their resource names, and the SAF access level required to issue the command.

Example: JES2

The $C'*jobname*' command has the following resource name:

jesx.CANCEL.JOB.

A user requires UPDATE access to issue the command.

Protecting JES2 Spool Data Sets

JES2 maintains data sets in the JES2 spool. Some of these data sets are JES2 system and user data sets. Others contain SYSIN and SYSOUT data for jobs in the system. This section describes how to protect the following types of JES2 spool data sets:

- SYSIN and SYSOUT data sets
- JESNEWS data set
- SYSLOG data set
- System data sets (trace and checkpoint data sets)

In order for any users to read or update classified JES2 spool data sets in an MLS system, their security labels must dominate the security labels of the spool data sets they are trying to access.

Protection for SYSIN and SYSOUT Data Sets

MLS protection mechanisms for JES2 SYSIN (for the job's input) and SYSOUT (for the job's output) data sets allow access to them only by the user who created the data sets. The user can also allow other users access.

When the MLS option to protect write-down is active, the system assigns the SYSIN and SYSOUT data sets the same label as the job. The subject that submits the job can access these data sets if their security label dominates the job's security label.

While a job executes, JES2 creates SYSIN and SYSOUT data sets using the following naming conventions:

nodeid.userid.jobname.jobid.dsnumber.name

nodeid

The name of the node where the data sets reside. In an MLS system, this is always the local node. The ID of the local node appears in the job log of each job. **Note:** The variable, nodeid, is not part of the data set name. Rather, it is added to the front of the data set name for the SAF call.

userid

The ID of the user associated with the job. This is the acid specified in the USER= keyword on the JCL JOB statement or the acid of the user who submits the job.

jobname

The name of the job as it appears in the NAME field of the JOB statement.

jobid

The job number JES2 assigns to the job. JES2 displays the job ID in messages sent to the submitter and in the job log of every job.

dsnumber

The unique data set number assigned by JES2 to the spool data set. D is the first character of the number.

name

The name of the data set as it is specified in the DSN= parameter of the DD statement in the job. The name cannot be JESYSMSG, JESJCLIN, JESJCL, or JESMSGLG. If the DSN= parameter is not specified in the DD statement that creates the spool data set in the JCL, JES2 uses a question mark (?) for the name.

Example: JES2

USER01 submits a job named JOB01 to run on the local node HOME. JES2 assigns a job ID of JOB000l and the value of the DSN= parameter for a SYSOUT data set is OUTPUT. The name of the spool data set for this job is HOME.USER01.JOB01.JOB000l.DOQ000003.OUTPUT.

To allow other users access to this data set, enter:

- Define ownership of the JESPOOL resource using the TSS ADD command
- Write a resource rule to allow access.

JESNEWS Data Set

The JESNEWS data set contains information for all JES2 users. All users should be able to read this data set. JESNEWS information prints after the header separator page of a job. The name of the JESNEWS data set takes the following form:

nodeid.*jesid*.$JESNEWS.STC*taskid*.D*newslvl*.JESNEWS

nodeid

The ID of the node where the JESNEWS data set was created. In an MLS system, this is always the ID of the local JES2 node.

Note: The variable, *nodeid*, is not part of the data set name. Rather, it is added to the front of the data set name for the SAF call.

jesid

The user ID associated with the JES2 system at your site.

taskid

The name of the task that created the JESNEWS data set.

newslvl

The level of this copy of JESNEWS. The value can vary from 0000101 to 0065535.

Example: JESNEWS

The resource name for JESNEWS on HOME that is created by STC05998 is:

```
HOME.JES2.$JESNEWS.STC05998.D0000101.JESNEWS
```

The job that updates the JESNEWS data set should be assigned security label, SYSLOW, and the security administrator must create a resource rule in the OPERCMDS resource class to permit that job to update JESNEWS. The resource name for the update job is '*jesname*.UPDATE.JESNEWS'.

SYSLOG Data Set

The SYSLOG data set contains a record of a system's daily activities. To prevent unauthorized access to SYSLOG in an MLS system, the SYSLOG data set should be assigned the security label, SYSHIGH. This means that only trusted programs and processes that are part of or defined to the system can access the SYSLOG.

To allow an operator DAC access to the SYSLOG data set, a security administrator must create a resource rule for the JESSPOOL resource class and a resource name such as:

home.+MASTER+.-.-.-.SYSLOG.

Note: The variable, *home*, is not part of the data set name. Rather, it is added to the front of the data set name for the SAF call.

Control Access to JES2 System Data Sets

The JES2 spool space data sets hold JES2 sysin and sysout files for jobs that are waiting for execution or printing. The JES2 checkpoint data sets provide an index into the spool space, and allow communication between JES2 address spaces running on different members of a multi-access spool complex. JES2 issues SAF calls to protect individual sysin and sysout data sets. To prevent other jobs from circumventing JES2 security, only JES2 may be given access to the spool space and checkpoint data sets. Since the JES2 acid does not have any special privileges, JES2 must be explicitly granted access to these data sets. The following is an example of an access rule to give JES2 access to its spool and checkpoint data sets.

To prevent unauthorized access to JES2 checkpoint data sets in an MLS system, these data sets should also be assigned the security label, SYSHIGH, since they may contain data labeled with any label up to SYSHIGH.

Other JES2 data sets, such as the data set containing JES2 initialization parameters, and the SYSl.HASPSRC and SYS1.AOSH3 libraries, which contain JES2 source and object modules, must be protected from modification by unauthorized users. Access rules should allow access only by system programmers, and these data sets should be labeled, SYSLOW. (Since ordinary users use security labels that dominate SYSLOW, MAC dominance checks prevent them from writing to these data sets.) JES2 must be given explicit read access to the data set containing its initialization parameters.

Important! Do not use the JES2 Spool Offload Facility. Use of the JES2 spool offload facility should not be permitted in an MLS system. No OFFLOADx statements should be included in the JES2 initialization deck.

Defining Acid for JES2 Started Task

You must define an acid for the JES2 started task. This acid must have the STC attribute. No other attributes need be specified.

Assigning Security Label SYSMULTI to the JES2 Started Task ID

You should also assign a default security label of SYSMULTI to the JES2 started task. This will allow ACEEs with different security labels to be anchored in TCBs in the JES2 address space.

Controlling Job Input

A security administrator can create resource rules to control which users can submit or cancel specific jobs and which input devices users can submit jobs from.

Controlling Job Submission and Cancellation

A site can optionally control which users can submit or cancel specific jobs, such as those that offload spool data sets. To do this the security administrator must activate the JESJOBS resource class and write resource rules for the SUBMIT and CANCEL resources.

Example: JESNEWS input

The following rule allows OPER1 to submit the NEWSJOB job that updates the JESNEWS data set:

```
$KEY(SUBMIT) TYPE(JES)
TSS PER(oper1) JESJOBS(submit.home.newsjob) ACCESS(read)
```

home

The name of the local JES2 node

newsjob

The name of the job that updates the JESNEWS data sets.

ACCESS(READ) is sufficient to submit a job.

JES3

JES3 uses the system authorization facility (SAF) to pass security information about jobs and resources to CA Top Secret. CA Top Secret makes access decisions based on information in its databases and passes its decision back to JES3.

Support for MLS JES3

The following is supported when MLS is active on an CA Top Secret system:

- Control the use of JES3 operator commands
- Control access to JES3 spool data sets
- Control access to JES3 system data sets
- Audit the use of all JES3 operator commands and access to JES3 data sets
- Control submission of jobs through JES3 input devices

In addition, CA Top Secret provides additional support beyond MLS requirements. You can control what data is output to a particular device and restrict certain users to specific output devices.

Restrictions

Certain JES3 functions should not be permitted in an MLS system when certain MLS options have been activated. The following restrictions apply when MLS is active on an CA Top Secret system:

- The network job entry (NJE) and remote job entry (RJE) functions can be used, but they must be configured properly.
- The only permissible output devices should be page printers, controlled by the PSF, and operated in deferred-printing mode, and line printers, operated as single-label devices, and labeled through procedural means. See the Print Services Facility (PSF) section for more information about printer restrictions.
- No site-written routines should be permitted in JES3 libraries, nor should modifications to JES3 routines be permitted.
- Entry of system commands through the input job stream is controlled by acid, just as when commands are entered from an operator console.
- JES3 does not support isolating work to specific systems based on a security label

Configuration Checklist JES3

This checklist describes the software configuration requirements when MLS is active on an CA Top Secret system.

Requirement	Complete
Control the use of JES3 operator commands	□
Protect JES3 Spool Data Sets	□
Define acid for JES3 started task	□
Assign security label SYSMULTI to the JES3 started task ID	□
Define access rules for JES3 started task	□
Control Job Input	□
Configure Network Job Entry (NJE) and Remote Job Processing (RJP)	□

Controlling the Use of JES3 Commands

The security administrator must be able to audit all JES3 commands in an MLS system. It is also necessary to control who can issue commands, since it is possible to issue commands not only from an operator console, but also from batch JCL. In either case, access is validated based on the acid associated with the job. To control JES3 commands and provide an audit trail for all JES3 commands, do the following:

- Enable the protection of operator commands
- Write resource rules to protect JES3 commands

JES3 Command Resource Names

JES3 commands have resource names that follow the example below:

jesname.command[.qualifier]

jesname

The name of the JES3 system requesting the command validation

command

The name of the JES3 command

qualifier

The type of object the command specifies, such as JOB or SYS.

See the IBM z/OS *JES3 Initialization and Tuning Guide* to determine the resource name of the JES3 command. It provides a list of JES3 commands, their resource names, and the SAF access level required to issue the command.

Examples: JES3

The $C'*jobname*' command has the following resource name:

jesx.CANCEL.JOB.

A user requires UPDATE access to issue the command. Here are some sample rules to protect JES3 commands:

The following permit allows system operator OPER1 to issue any JES3 commands, but CA Top Secret creates a log record for each JES3 command issued.

```
TSS PER(oper1) OPERCMDS(*all*)
               ACTION(audit)
```

You could create more specific entries in the permit to establish a finer control over the operators issuing JES3 commands.

The following rule lets OPER1 cancel jobs

```
TSS PER(oper1) OPERCMDS(JES.CANCEL)
```

Protecting JES3 Spool Data Sets

JES3 maintains data sets in the JES3 spool. Some of these data sets are JES3 system and user data sets. Others contain SYSIN and SYSOUT data for jobs in the system. This section describes how to protect the following types of JES3 spool data sets:

- SYSIN and SYSOUT data sets
- JESNEWS data set
- SYSLOG data set
- System data sets (trace and checkpoint data sets)

In order for any users to read or update classified JES3 spool data sets in an MLS system, their security labels must dominate the security labels of the spool data sets they are trying to access.

Protection for SYSIN and SYSOUT Data Sets

MLS protection mechanisms for JES3 SYSIN (for the job's input) and SYSOUT (for the job's output) data sets allow access to them only by the user who created the data sets. The user can also allow other users access.

When the MLS option to protect write-down is active, the system assigns the SYSIN and SYSOUT data sets the same label as the job. The subject that submits the job can access these data sets if their security label dominates the job's security label.

While a job executes, JES3 creates SYSIN and SYSOUT data sets using the following naming conventions:

nodeid.userid.jobname.jobid.dsnumber.name

nodeid

The name of the node where the data sets reside. In an MLS system, this is always the local node. The ID of the local node appears in the job log of each job.

Note: The variable, nodeid, is not part of the data set name. Rather, it is added to the front of the data set name for the SAF call.

userid

The ID of the user associated with the job. This is the acid specified in the USER= keyword on the JCL JOB statement or the acid of the user who submits the job.

jobname

The name of the job as it appears in the NAME field of the JOB statement.

jobid

The job number JES3 assigns to the job. JES3 displays the job ID in messages sent to the submitter and in the job log of every job.

dsnumber

The unique data set number assigned by JES3 to the spool data set. D is the first character of the number.

name

The name of the data set as it is specified in the DSN= parameter of the DD statement in the job. The name cannot be JESYSMSG, JESJCLIN, JESJCL, or JESMSGLG. If the DSN= parameter is not specified in the DD statement that creates the spool data set in the JCL, JES3 uses a question mark (?) for the name.

SYSLOG Data Set

The SYSLOG data set contains a record of a systems' daily activities. To prevent unauthorized access to SYSLOG in an MLS system, the SYSLOG data set should be assigned the security label, SYSHIGH. This means that only trusted programs and processes that are part of or defined to the system can access the SYSLOG.

To allow an operator DAC access to the SYSLOG data set, a security administrator must create a resource rule for the JESSPOOL resource class and a resource name such as:

home.+MASTER+.-.-.-.SYSLOG.

Note: The variable, *home*, is not part of the data set name. Rather, it is added to the front of the data set name for the SAF call.

Controlling Access to JES3 System Data Sets

The JES3 spool space data sets hold JES3 sysin and sysout files for jobs that are waiting for execution or printing. The JES3 checkpoint data sets provide an index into the spool space, and allow communication between JES3 address spaces running on different members of a multi-access spool complex. JES3 issues SAF calls to protect individual sysin and sysout data sets. To prevent other jobs from circumventing JES3 security, only JES3 may be given access to the spool space and checkpoint data sets. Since the JES3 acid does not have any special attributes, JES3 must be explicitly granted access to these data sets.

Assigning Security Label SYSMULTI to the JES3 Started Task ID

You should also assign a default security label of SYSMULTI to the JES3 started task. This will allow ACEEs with different security labels to be anchored in TCBs in the JES3 address space.

Controlling Job Input

A security administrator can create resource rules to control which users can submit or cancel specific jobs and which input devices users can submit jobs from.

Controlling Job Submission and Cancellation

A site can optionally control which users can submit or cancel specific jobs, such as those that offload spool data sets. To do this the security administrator must activate the JESJOBS resource class and write resource rules for the SUBMIT and CANCEL resources.

Configuring NJE and RJP

If you want to successfully use NJE and RJP in an CA Top Secret MLS system, configure them as follows:

- Assign a security label and write resource rules for each NJE and RJP input device in the JESINPUT resource class. NJE or RJP input devices are not multi-label devices-they can only handle data with the same security label. All work coming to a JES3 input device is assumed to have the same security label as the JES3 input device.
- Assign a security label to each NJE and RJP printer in the WRITER resource class. The security label of the printer must dominate the security label of the work for JES3 to transmit the work to it.

Print Services Facility (PSF)

In an MLS system, operators, rather than end-users, are responsible for separating and distributing printed output. There are procedures that operators in an MLS system should follow to manage deferred-printing mode page printers. Deferred-printing mode printers are those that select output from the JES2 output queues, rather than being under direct control of a particular job. This should be the only mode allowed for page printers.

The security separator pages can be used to identify the user who submitted the print job.

Although PSF will enforce “print labeling”, the practice of putting security labels on all printed output, CA Top Secret does not support print labeling in an MLS environment.

Configuration Checklist

This checklist describes the software configuration requirements when MLS is active on an CA Top Secret system.

Requirement	Complete
Use security separator pages	□

Using Security Separator Pages

Printing on all page printers is done through PSF. The PSF subsystem controls all paged, hardcopy printing in CA Top Secret. To reduce the chance of users tampering with separator pages, PSF ensures that all printing is identified with the user who submitted the print job. It does this by putting the user's UID and an unforgeable, randomly assigned number on the beginning and ending separator pages for each job. Operators must check the numbers on the beginning and ending separator pages to ensure that they match and are authentic. If they do not, the output stack should be searched further for a matching ending page. Since the numbers are determined when the job is printed, rather than when it is run, the job cannot find out what the number is going to be to simulate authentic separator pages. To use security separator pages, you must replace the PSF-supplied default job header and trailer routines with PSF exit routines APSUX01S and APSUX02S from SYS1.SAMPLIB.

TCP/IP

In an MLS system, in addition to writing DAC resource rules, which control authorization to TCP/IP resources in the SERVAUTH resource class, security labels can be used to protect TCP/IP resources. Using SAF calls, TCP/IP provides CA Top Secret with the information it needs to do MAC and DAC checking in the system.

Support for MLS TCP/IP

The following is supported when MLS is active on an CA Top Secret system:

- Security labeling of TCP/IP resources in the SERVAUTH class, such as stacks (including network security zones and IP addresses), sockets, and socket commands
- MAC checking of access to resources in the SERVAUTH class
- DAC checking of access to resources in the SERVAUTH class

Restrictions

When MLS is active on an CA Top Secret system, audit all programs used.

Note: Not all client-server applications and user commands are authorized for use in an MLS system.

Configuration Checklist

This checklist describes the software configuration requirements when MLS is active on an CA Top Secret system.

Requirement	Complete
Configure TCP/IP	□
Assign security labels to resources in the SERVAUTH class	□
Protect TCP/IP stack access	□
Protect TCP and UDP port access	□
Protect access to the IP network or hosts on the IP network	□
Assign security labels to acids for users/tasks that must access TCP/IP resources	□

Configuring TCP/IP

Applications in a network use sockets to communicate with each other. In an MLS system, if you want to protect network resources with security labels to ensure that no sensitive data is disclosed or declassified, the user sessions under which the applications run and communicate with each other must have equivalent security labels.

Assigning Security Labels to Resources in the SERVAUTH Class

CA Top Secret uses the SAF to control access to network resources and allows a security administrator to assign security labels to these resources. To do this, you need the name of the TCP/IP resource that you want to secure. In SAF, these names are referred to as entity names. In CA Top Secret, these names are referred to as resource names.

- TCP/IP stack resources are named:
- EZB.STACKACCESS.*sysname.tcpname*
- TCP port resources are named:
- EZB.PORTACCESS.*sysname.tcpname*.SAF*keyword*
- TCP/IP security zone resources are named:
- EZB.NETACCESS.*sysname.tcpname.zonename*

Protect TCP/IP Stack Access

TCP/IP uses stacks to control the creation of sockets, the use of socket commands and the use of gethostid() and gethostname() commands.

To provide MAC protection for access to TCP/IP stacks in an CA Top Secret MLS environment, assign security labels to the EZB.STACKACCESS.sysname.tcpname resources in the SERVAUTH resource class by creating MLS resource records.

Example: TCP/IP stack protection

To assign security label, LABEL2, to the TCP/IP stack, enter: create a SECLABEL Compiled Record for it and include the $RTYPE control statement. You must have the SECURITY privilege in your logonid to create the record.

```
TSS ADD(mls) SERVAUTH(ezb.STACKACCESS.SYSNAME.TCPNAME)
             SECLABEL(LABEL2)
```

Protect Access to and Hosts on the IP Network

TCP/IP also uses stacks to control access to IP networks. IP addresses are mapped into network security zones. Resource names are created for each network security zone on a stack.

To provide MAC protection for access to a system from an IP address in an CA Top Secret MLS environment:

- Assign an IP address to a network security zone by creating a TCP/IP Profile definition
- Assign a security label to the EZB.NETACCESS.*sysname.tcpname.zonename* network zone name to which the IP address is mapped in the SERVAUTH resource class by creating an MLS resource record for it

Example: host protection

To assign security label, LABEL2, to an IPv6 address mapped into network security zone, ZONEB, create an MLSresource record for it.

```
TSS ADD(mls) SERVAUTH(ezb.NETACCESS.SYSNAME.STACKNAME.ZONEB)
             SECLABEL(LABEL2)
```

When MLS is activated on the system, and a security label is not specified by a user or application at signon, the seclabel is defaulted from the SERVAUTH resource (if there is one and it is not SYSMULTI). If a seclabel is specified by a user or application at signon, system entry is allowed if the user is authorized to the seclabel specified and it is equivalent to the seclabel that is protecting the IP address in the MLS SERVAUTH resource record.

Important! To support IPv6 addresses, which are much longer than IPv4 addresses, the TERMID is no longer used as the source ID for IP-based ports of entry trying to gain access to the system and resources. Instead, the network access security zone name in the SERVAUTH class contains the IP address of a user trying to gain access to the system and resources. This functionality replaces conversion of IPv4 addresses to hexadecimal terminal names.

Protecting TCP and UDP port access

To provide MAC protection for access to TCP and UDP ports in an CA Top Secret MLS environment, do the following:

- Assign security labels to the EZB.PORTACCESS.sysname.tcpname.SAFkeyword resources in the SERVAUTH class

Example: port protection

To assign security label, LABEL2, to a TCP or UDP resource, create an MLS resource record.

```
TSS ADD(mls) SERVAUTH(ezb.PORTACCESS.SYSNAME.TCPNAME.SAFkeyword)
             SECLABEL(LABEL2)
```

Assigning Security Labels to Acids for Access to TCP/IP Resources

In an MLS environment, the userids associated with tasks trying to access classified TCP/IP resources can be assigned security labels.

Time Sharing Option (TSO/E)

In an MLS CA Top Secret system, all users must log on to the system and undergo identification and authentication checks. CA Top Secret creates a security environment for each TSO/E user. The security label that is used to logon to TSO/E is maintained in a user's security environment and is used to make access decisions until the user logs off. CA Top Secret ensures that a user cannot alter his security label in any way while logged onto the system. To change his security label, a user must log off and log on again to TSO/E with a new label.

Support for MLS TSO/E

The following is supported when MLS is active on an CA Top Secret system:

- All users can be be identified and authenticated in the system with a acid and security label
- All user logon attempts can be audited
- All user messages can be protected
- Sending and receiving data sets with the TRANSMIT and RECEIVE commands can be controlled based on security labels
- Users can be restricted to using the CANCEL and OUTPUT commands only on jobs whose job names begin with their acids

Restrictions

The following restrictions apply when MLS is active on an CA Top Secret system:

- Remove all user-written exits
- Do not give any TSO/E user the OPERATOR privilege
- Do not activate the Information Center Facility

Configuration Checklist

This checklist describes the software configuration requirements when MLS is active on an CA Top Secret system.

Requirement	Complete
Define an acid for the TSO started task	□
Define access rules for the TSO started task	□

Requirement	Complete
Provide identification and authentication checks	□
Define a acid for each TSO/E user	□
Assign security labels to TSO/E users	□
Audit all logon attempts	□
Protect user messages	□
TSO/E SEND and LISTBC commands	□
Follow requirements for protecting message transmission	□
Modify IKJTSOxx member of SYS1.PARMLIB	□
Create resource rules for each user mail log	□
Label user mail logs SYSHIGH	□
Create an acid record for *LISTBC ID	□
Assign security label SYSLOW to SYSI1.BROADCAST	
Control use of RECEIVE and TRANSMIT commands	□
Assign security labels to LOG.MISC data sets	□
Assign security labels to NAMES.TEXT data sets	□
Replace default IKJEFF53 exit	□
Provide an audit trail for SEND and LISTBC commands	□

Defining an Acid for the TSO Started Task

You must define a acid for the TSO started task. This acid must have the STC attribute. No other attributes need be specified.

Defining Access Rules for the TSO Started Task

When the TSO started task starts up, it reads initialization parameters from SYS1.PARMLIB. Because of this, it must be granted read access to the data set. Here is an example of a rule granting TSO read access to SYS1.PARMLIB:

```
$KEY(SYS1)
 PARMLIB UID(TSO) R(A)
```

Providing Identification and Authentication Checks

In an MLS system, all TSO/E users must undergo identification and authentication checks by CA Top Secret.

Defining a Logonid for Each TSO/E User

The security administrator must create a unique acid record for each TSO/E user. CA Top Secret stores the information in security file and retrieves it when a user attempts to log on to the system.

Assigning Security Labels to TSO/E Users

The security administrator should assign security labels to TSO/E users so that users can access the system and classified data and resources that they need to perform their work on the system.

In an MLS system, when a user logs on to a system, he may specify a security label. However, CA Top Secret will always default a security label for a user who does not supply a label. If a user does not specify a security label at logon on the full-screen panel, if there was a previous TSO/E session for the user, the security label from that session will be used. If there was no security label for the previous TSO/E session, CA Top Secret uses the security label from the terminal (an MLS resource record in the TERMINAL class), if there is one. If the terminal does not have a security label, CA Top Secret uses the default security label from the User's acid record, if one exists. If CA Top Secret cannot default a security label from any of these places, the user will be logged on with the SYSLOW security label.

Once CA Top Secret verifies that the user is authorized to use a security label, the security label is maintained in the user's address space and is used to make access decisions until the user logs off. CA Top Secret ensures that a user cannot alter his security label in any way while logged onto the system.

Important! If a user wants to change his security label, he must log off and log on using a different security label. This reduces the threat from Trojan horses and prevents inadvertent data disclosure.

The TSO/E LOGON command lets users provide a security label at logon by specifying the SECLABEL parameter. In addition, the TSO/E full-screen panel lets users provide a security label at logon. A new field, SECLABEL, has been added to the TSO/E Full-Screen. When MLS is active, a valid security label may be specified in this field. If an invalid security label is specified in this field, the user will repeatedly be prompted until a valid security label is specified or the field is left blank.

Important! When MLS is inactive on a system, the location of the new SECLABEL field may affect some “screen-scrapers”.

Audit Logon Attempts

In an MLS system, in addition to checking to see that a user provides a valid logonid and password, CA Top Secret checks to see that a user provides a security label that he is authorized to use. CA Top Secret creates a record for each unsuccessful logon attempt. These records are stored in the ATF/SMF data sets and can be viewed using the TSSUTIL report program.

Protecting User Messages

In an MLS system, when write-down is protected, all data created by a user receives the security label of the user. This means that a user logged on with a security label LABELA whose value is SECLEVEL(50) CATEGORY(AA BB) creates data labeled LABELA. This includes messages. To prevent inappropriate disclosure, a user cannot receive messages from another user whose security label dominates his. For example, a user whose security label has a value of SECLEVEL(25) CATEGORY(AA QQ) cannot receive messages from a user whose security label has a value of SECLEVEL(50) CATEGORY(AA BB). This places some restriction on where messages that cannot be immediately sent (those sent with the SAVE or LOGON options of the TSO/E SEND command) are stored. Traditionally, the SYS1.BRODCAST data set was where these messages were stored, along with broadcast messages intended for all users. In an MLS system, SYS1.BRODCAST is considered a public object, readable by any user. This makes it unsuitable for storing messages with security labels higher than SYSLOW.

Instead, an MLS system requires a separate data set, called a mail log, for each user. This log is labeled SYSHIGH, and contains all the saved messages for the user, regardless of his security label. The user cannot access the mail log as an ordinary data set, but only through the LISTBC command, a trusted command that will only show the user the messages that are dominated by his current session security label. The LISTBC command also displays any saved broadcast messages. These messages, which are intended for all users and are always labeled SYSLOW, are saved in the SYS1.BRODCAST data set.

The goal in separating messages sent to the SYS1.BRODCAST data set and those sent to individual user mailboxes is to prevent the disclosure of sensitive information. A user logged on with a high label should not be able to send messages to a user logged on at a lower label. This could allow sensitive information to be disclosed. Similarly, if a user who can log on with a high label is logged on with a low label, he should not be able to view messages sent to him at the high label while he is logged on at the low label.

Using TSO/E SEND and LISTBC Commands

TSO/E provides the capability for TSO/E users to communicate with each other, while logged on, through messages. By default, when a message is sent, the operating system attempts to display the message to the user or console, if active. However, the sender can specify that the message is to be saved, if it cannot be delivered immediately. In an MLS system, immediate delivery might not occur for one of several reasons:

- The intended recipient is not logged on
- The intended recipient is logged on, but is not receiving messages
- The recipient's security label does not permit the message's display

When a message is saved for an individual TSO/E user, it is called “mail”, and is placed in a user's exclusive mail log. When an operator sends a message to all users, it becomes a “notice”, if it is saved in the SYS1.BRODCAST data set for later display to users who could not receive it interactively. If an operator-sent message is directed to an individual user, if it is to be retained, it is treated as mail, not as a notice, and it is stored in the user's mail log until the user logs on or asks to review his mail.

Messages can originate from sources other than user TSO/E address spaces and system consoles. JCL supports a NOTIFY parameter that lets a batch job notify the user identified by this parameter of a job's completion. JES2 implements this feature by issuing the OPERATOR SEND command, so that it's processing is identical to that of a console operator sending an individual TSO/E user message.

The TSO SEND command verifies the sender's authority to send a message to the designated recipient through a RACROUTE AUTH request for the SMESSAGE class that includes the user's acid. If the SEND program, running in the sender's TSO/E address space, must deliver the message, even if the recipient is not logged on or the intended recipient is not receiving messages, the message is written to the user's mail log or, in the case of a broadcast message, to SYS1.BRODCAST.

The following table summarizes some of the options that a sender has when sending a message using the TSO/E SEND and OPERATOR SEND commands:

Command	Options	Original Destination	Alternative Destination
SEND	NOW(default)	Recipient's Terminal	None
SEND	LOGON	Recipient's Terminal	Recipient's Mail Log
SEND	SAVE	Recipient's Mail Log	None
OPERATOR SEND	ALL or SAVE	Recipient's Terminal	SYS1.BROADCAST
OPERATOR SEND	USER and LOGON	Recipient's Terminal	Recipient's Mail Log

The retrieval of mail and notices is performed by LISTBC. The LISTBC command displays broadcast messages and messages saved in the TSO/E user's mail log. All broadcast messages are stored in the SYS1.BRODCAST data set, which should be labeled, SYSLOW, while LISTBC creates for each user a mail log labeled, SYSHIGH, the first time the command is invoked. To do this, the LISTBC command automatically issues a RACROUTE VERIFY request with an acid of *LISTBC, and TRUSTED='YES', when it is first called. CA Top Secret treats this call as a started task and saves the security label as, SYSHIGH. If the mail log does not exist, LISTBC proceeds to create it with its current security label of SYSHIGH.

Because LISTBC is trusted and, thus, labeled SYSHIGH, it is able to access a user's mail log, no matter what the security label is for the user's TSO/E session. Thus, on subsequent invocations, LISTBC is able to open the mail log and for each message record it finds, it issues a RACROUTE DIRAUTH request with the security label for the message record (returned by the RACROUTE SECLABEL parameter). CA Top Secret compares that security label to the security label of the user assigned to the current TSO/E address space. If the security label of the TSO/E address space dominates that of the message, then the message is allowed to be displayed. Otherwise, LISTBC deletes the message record from the mail log.

In an MLS system, when a user issues a TSO SEND command, CA Top Secret performs a MAC and DAC check. The MAC check processing works differently depending on whether the sending user specifies the NOW, LOGON, or SAVE parameters.

- If the sending user specifies the NOW parameter, or if the sending user specifies the LOGON parameter and the receiving user is currently signed on, CA Top Secret checks to see if the security label of the target user dominates the security label of the sending user. If it does, the DAC check searches for the SMESSAGE resource rule to see if the user can send a message to the target user. If not, the user receives an error message. If so, the command executes and the message appears immediately on the target user's screen. However, if the security label of the sending user dominates the security label of the target user, the command fails and the sending user is notified.
- If the sending user specifies the SAVE parameter, or if the sending user specifies the LOGON parameter and the receiving user is not currently logged on, the system stores the message in the target user's user log until he logs on or issues the LISTBC command. MAC label checking is done at that time.
- In an MLS system, when a user issues the LISTBC command, the system sends any SYS1.BRODCAST messages and then any other messages whose security labels are dominated by the user's security label. The user can only view those messages in his user log if his security label dominates the security label of the user who sent the message. If there are other messages in the user's log, the system does the following:
- If the security label of the message dominates the security label of the user issuing the LISTBC command, but the user can log on with a security label that dominates the security label of the message, the system displays the following message:

```
IKJ5G9621 YOUR USER LOG CONTAINS MESSAGES THAT CANNOT BE VIEWED AT YOUR CURRENT
SECURITY LABEL
```

- The user can then decide whether he wants to log off and log on with a higher security label to view the message.

Requirements for Protecting Message Transmission

The following are requirements for protecting message transmission:

- Modify the IKJTSOxx member of SYS1.PARMLIB
- Create resource rules for each user mail log
- Label user mail logs SYSHIGH
- Create acid record for *LISTBC
- Assign a security label to the *LISTBC ID
- Create an access rule for SYS1.BRODCAST
- Assign security label SYSLOW to SYS1.BRODCAST

Modifying IKJTSOxx Member of SYS1.PARMLIB

The security administrator can configure the SEND parameter of IKJTSOxx member of SYS1.PARMLIB properly to protect individual user mail logs and restrict access to messages based on labels. The proper settings are as follows:

- Set the CHKBROD operand to off. This prevents the LISTBC command from looking in SYS1.BRODCAST for user messages. Instead, it looks in the individual mail logs only.
- Set the MSGPROTECT operand to ON. This setting turns on label validation for user mail logs. When a user issues the LISTBC command to view messages, CA Top Secret performs security label validation to ensure that the user's security label dominates the security label of any messages in the box. The user cannot view messages whose security labels dominate his security label.
- Set the LOGNAME operand to specify the high-level qualifiers of the data set names to be used in user mail logs. The value specified will be suffixed with each user's acid. For example, if LOGNAME were set to MAIL.BOX, the user mail log for the acid USER01 would be named MAIL.BOX.USER01.
- Set the USEBROD operand to OFF. This prevents messages for users who do not have individual user mail logs from being stored in SYS1.BRODCAST. Instead, the SEND command is rejected.

The security administrator should make these changes before IPLing the system. For more information about these changes, see the following:

- IBM *z/OS MVS Initialization and Tuning Reference* for details on the IKJTSOxx member of SYS1.PARMLIB
- IBM *z/OS Customization* for details on the use of user mail logs and the SEND PARMLIB parameter

The OPERSEND, USERSEND, SAVE, and OPERSEWAIT operands do not affect security and can be set however your site pleases.

Creating Resource Rules for Each User Mail Log

The security administrator must create a resource rule to protect user mail logs.

Labeling User Mail Logs SYSHIGH

Because messages with various labels can be sent to a user, the security administrator should label each user log SYSHIGH by creating an MLS resource record. This is necessary only for mail logs created before write-down is prohibited (NOMLWRITE is set in the MLSOPTS record). After that, mail logs are labeled automatically when they are created. For example, the following record labels all user logs with SYSHIGH:

Creating Acid Record for *LISTBC ID

The security administrator should create an acid record for the *LISTBC acid. This acid is used by the *LISTBC command when it allocates a new mail log for a user. The *LISTBC acid should have the RESTRICT attribute. It must not have the JOB or TSO attributes, as these attributes might allow it to be used as an ordinary acid without a password. The following is an example of defining the *LISTBC acid to CA Top Secret:

Assigning a Security Label to the *LISTBC ID

The security administrator can assign security label SYSHIGH to the *LISTBC acid. However, since the *LISTBC ID enters the system as trusted, the system will assign it the SYSHIGH security label, by default. The *LISTBC ID is used by the LISTBC command processing to update individual user mail logs. It runs as a trusted process.

Creating an Access Rule for SYSl.BRODCAST

The security administrator should create an access rule for the SYSl.BRODCAST data set, to allow all users to read it. For example:

```
$KEY(SYSl)
TSS PER(all) DSN(sys1.broadcast)
             ACCESS(read)
```

This rule permits all users to read SYSl.BRODCAST, but prevents them from sending individual users messages there.

Assigning Security Label SYSLOW to SYSl.BRODCAST

Because all users can view messages in SYSl.BRODCAST, the security administrator should label it SYSLOW. For example:

```
TSS ADD(mls) DSN(sys1.broadcast)
             SECLABEL(SYSLOW)
```

Controlling Use of TRANSMIT and RECEIVE Commands

When a user issues the TRANSMIT command to send a data set to another user, the security label of the data set is the security label of the user issuing the TRANSMIT command. When the target user tries to issue the RECEIVE command, his security label must dominate the security label of the data set. If the security label of the target user does not dominate the security label of the data set, he will not receive any notice that a data set was sent to him. If the target user cannot specify a label that dominates the label of the data set sent to him, the system creates an SMF record when he tries to receive it.

Assigning Security Labels to LOG.MISC Data Sets

The LOG.MISC data set for a user contains information on TRANSMIT and RECEIVE command processing. In an MLS system, users who are authorized to use more than one security label should have a LOG.MISC data set for each security label.

In practice, security administrators should assign to a user's LOG.MISC data set the security label he uses most often. If a user wants to receive or transmit messages at a different security label, he must use the LOGDATASET or LOGDSNAME parameters of the RECEIVE and TRANSMIT commands.

When write-down is protected, a user can also label his own LOG.MISC data set by logging on at the security label he wants the data set to have, allocating the data set, then performing a TRANSMIT or RECEIVE.

Assigning Security Labels to NAMES.TEXT Data Sets

The NAMES.TEXT data set contains data that the RECEIVE and TRANSMIT commands can view or update. The security administrator should assign to the NAMES.TEXT data set for a user the lowest security label that the user can specify at logon. This lets the RECEIVE and TRANSMIT commands have access to the NAMES.TEXT data set and the user can update the data set when logged on at that lowest label.

When write-down is protected, the user can also label his own NAMES.TEXT data set by logging on at the security label he wants the data set to have, allocating the data set, then editing it.

Replacing Default IKJEFF53 Exit

To restrict users to using the CANCEL and OUTPUT commands only on jobs whose job names begin with their acids, a security administrator must replace the default IKJEFF53 exit with the IKJEFF53 exit in SYS1.SAMPLIB. The version of IKJEFF53 in SYS1.SAMPLIB suppresses the restriction of the CANCEL command if the JESJOBS class is active, and of the OUTPUT command, if the JESSPOOL class is active. This allows the command restriction to be done based on resource rules, rather than job names.

Example

When IKJEFF53 exit is replaced with the IKJEFF53 exit in SYS1.SAMPLIB, a user with an acid of USER01 could CANCEL jobs named USER01A or USER01TST, but could not cancel a job named USER02X.

VTAM

VTAM passes security information to CA Top Secret at system entry. Using SAF calls, VTAM provides CA Top Secret with the identification and authentication information it needs to return an access decision to VTAM.

Support for MLS VTAM

The following is supported when MLS is active on an CA Top Secret system:

- Identification and authentication information is passed to CA Top Secret
- The transmission and reception of cross-address-space TSO/E messages is controlled
- The VTAM LOGON command has not changed as a result of MLS.

Restrictions

The following restrictions apply when MLS is active on an CA Top Secret system:

- The VTAMAPPL class should be active

Configuration Checklist

This checklist describes the software configuration requirements when MLS is active on an CA Top Secret system.

Requirement	Complete
Define an acid for the NET started task	□
Define access rules for the NET started task	□
Define resource rules for VTAM devices	□
Control access of applications	□
Train users in trusted path logon sequences	□

Defining an Acid for NET Started Task

A security administrator should define an acid for the NET started task. This acid must have the STC attribute. No other attributes need be specified.

Defining Access Rules for NET Started Task

When the NET started task starts up, and when VTAM resources are activated, VTAM reads initialization parameters from the data sets allocated to the VTAMLST ddname in the NET procedure (generally SYS1.VTAMLST). VTAM also fetches modules from the libraries allocated to the VTAMLIB ddname (generally SYS1.VTAMLIB). VTAM must be granted read access to these data sets.

The following access rule grants NET read access to SYS1.VTAMLIB and SYS1.VTAMLST:

```
TSS PER(net) DSN(sys1.vtamlib) ACCESS(read)
TSS PER(net) DSN(sys1.vtamlst) ACCESS(read)
```

Controlling Access of Applications

Application programs request permission to open VTAM access method control blocks (ACBs) in order to access VTAM resources and facilities. VTAM makes a SAF call to CA Top Secret to verify that the application can open an ACB. The security administrator usually identifies those applications to CA Top Secret in a resource rule for the VTAMAPPL resource class. However, in an MLS system, the VTAMAPPL class should be active, but non-APF-authorized programs cannot access VTAM resources.

The security administrator does not have to write a resource rule because CA Top Secret prevents access by default.

Training Users in Trusted Path Logon Sequences

CA Top Secret provides trusted path support for users logging on to VTAM terminals. Trusted path is an important defense against password grabbers, programs that simulate logon prompts while collecting the passwords of unsuspecting users, and it is optional, but recommended for use in an MLS system configuration. For trusted path protection to be effective, people must invoke the trusted path every time they sign on. New users must be trained in use of the trusted path logon sequences.

z/OS MVS

z/OS MVS uses the IBM System Authorization Facility (SAF) as an interface to all external security products such as CA Top Secret. SAF uses the MVS router to process the security calls. CA Top Secret uses the information stored in its database to make a recommendation to the program making the security call. The program must act on the recommendation of CA Top Secret.

Support for MLS z/OS

The following is supported when MLS is active on an CA Top Secret system:

- CA Top Secret controls access to data sets and resources in a z/OS MVS system based on access and resource rules and security labels assigned to those data sets and resources.
- Only users identified in CA Top Secret can access a z/OS MVS system

Restrictions

The following restrictions apply when MLS is active on an CA Top Secret system:

- User-written exits or modifications should be removed from the system. **Note:** The system may still function if these exits and modifications have not been removed, however, depending on the options that have been set to establish the MLS environment, results may compromise an MLS system.
- Do not allow operation of the system console in problem determination mode.
- Do not allow remote APPC/MVS transactions or remote logical units, APPC/MVS servers, or multi-trans APPC/MVS transaction programs (TPs).
- Some modules shipped with z/OS MVS and resident in link list or LPA libraries, implement features that should not be allowed in an MLS system. These modules should be moved to other, non-APF authorized libraries or deleted in order to disable these features:
  - CIPOPRT-This 3800 offline utility should be moved out of SYS1.LINKLIB
  - MVSSERV-This client-server platform should be moved out of SYS1.LPALIB

Configuration Checklist z/OS

This checklist describes the software configuration requirements when MLS is active on an CA Top Secret for z/OS MVS system.

Requirement	Complete
Force console operators to log on before issuing commands	□
Modify the CONSOLxx member of SYS1.PARMLIB	□
Create acid records for all operators	□
Assign security labels to console operators	□
Define console source controls (optional)	□
Write resource rules to control console access	□
Write resource rules for operator commands	□
Configure SCHEDxx for data set protection	□
Specify SMF controls	□
Provide an audit trail for accesses to protected objects using operator commands	□
Protect critical data sets	□
Write access rules	□

Requirement	Complete
Assign security labels to critical data sets	□
Protect UNIX files and directories	□
Assign security labels to UNIX files and directories	□
Protect resources	□
Identify and classify users on the system	□
Create acids for users	□
Assign security labels to users	□
Establish JCL standards	□
Define acids for MVS started tasks	□
Define resource rules for LLA started task	□
Define access rules for BLSJPRMI started task	□
Make sure SMS is active in IEFSSNxx	□
Move forbidden modules out of system libraries	□

Forcing Log On

Console operators are considered trusted users. It is assumed that anyone with physical access to operator consoles is cleared to the label of all data on the system. However, the actions of console operators must be audited so that a specific action can be traced to a specific console operator. In order to ensure accountability, operators must sign on to operator consoles and undergo identification and authentication. Before signing on, they are able to view message traffic on the consoles, but they can issue no commands.

To force operators to sign on before issuing commands, you must do the following:

- Modify the CONSOLxx member of SYS1.PARMLIB
- Create acid records for console operators
- Assign security labels to console operators
- Define console source controls (optional)

Modifying the CONSOLxx Member of SYS1.PARMLIB

You must specify LOGON(REQUIRED) on the DEFAULT statement in the CONSOLxx member of SYS1.PARMLIB. This forces users to undergo identification and authentication checks before entering commands through an operator console. The following exceptions apply:

- Operators can issue commands from the master console before CA Top Secret is active.
- Operators can issue the VARY MSTCONS command from any console before CA Top Secret is active. This command allows an operator to define the master console before CA Top Secret is active.

After CA Top Secret is active, operators must log on to all system consoles.

Creating Acid Records for all Operators

Each console operator must have a unique acid record. In addition to the usual fields that must be set, you may want to specify an entry source for each operator to define the consoles that they can access the system.

Assigning Security Labels to Console Operators

In an MLS system, you should assign the system-defined security label SYSHIGH to console operators. To authorize an operator to use the SYSHIGH security label at console logon, an MLS administrator must add the SYSHIGH seclabel to the operator's acid record. Operators need this high-level clearance. Their unrestricted authority to view console message traffic prevents the situation where a critical system error message is issued, and no operator is signed on with a high enough label to see it.

Defining Console Source Controls

It is possible to restrict the use of particular consoles to specific users. For example, printer operators might be allowed only to use consoles in the printer room, while master operators could use any console, and ordinary users could not use any consoles.

Writing Resource Rules to Control Operator Commands

To meet the requirements of an MLS system, the system must be able to audit all operator commands. These commands include all MVS operator commands and all operator commands for other products or subsystems, such as JES2 or CA Top Secret. The information in the audit records lets a security administrator or auditor determine the following:

- The command that was entered
- The user who issued the command
- The terminal the command was issued from
- When the command was issued
- Whether the user had the authority to issue the command
- The action the system took

It is also necessary to control the use of operator commands issued through the JCL of batch jobs. These commands are controlled the same way as commands issued through an operator console; a user who is authorized to issue commands from a console can also issue commands in batch JCL.

Protecting UNIX Files and Directories

In an MLS system, CA Top Secret assigns security labels to UNIX files and directories at the time they are created. However, if MLS is not active on the system at the time UNIX files and directories are created, but is later turned on, these objects will not have security labels. As a result, the security administrator must provide security labels for any UNIX files and directories that do no have security labels by issuing the UNIX **chlabel** shell command; otherwise, all accesses to these files and directories may be denied by CA Top Secret.

See the Assigning Security Labels to UNIX Files and Directories section in the "Implementing and Administering a Multilevel-Secure System" chapter for a list of security labels that are recommended for UNIX files and directories to which the system did not assign security labels. In addition, see the z/OS Unix System Services (UNIX) section for a complete discussion of how to configure UNIX in an CA Top Secret MLS system.

Configuring SCHEDxx for Data Set Protection

The PPT statement in the SCHEDxx member of SYS1.PARMLIB specifies attributes for programs. The PASS|NOPASS option, which originally specified whether data set protection by passwords should be honored, also determines whether RACROUTE calls are issued at data set open. NOPASS suppresses the RACROUTE calls. PASS, the default, enables the RACROUTE calls, and must be specified for all programs in the SCHEDxx member.

The SCHED00 member of SYS1.SAMPLIB contains a sample SCHEDxx member.

Protecting Critical Data Sets

In an MLS system, you should protect the data sets managed by library lookaside (LLA), as well as all system data sets. LLA is an MVS service that caches information about production libraries in order to provide faster access to them. The list of production libraries includes all link list data sets (all data sets specified in the LNKLSTxx member of SYS1.PARMLIB) and all data sets specified in the CSVLLAxx members of SYS1.PARMLIB.

To protect these data sets, do the following:

- Write access rules
- Assign security labels to the data sets

Writing Access Rules

Use the TSS command to create access rules for the high-level qualifiers of the data sets specified in the CSVLLAxx and LNKLSTxx members of SYS1.PARMLIB. You must allow console operators write access to these data sets. This is necessary because when a console operator issues a MODIFY LLA, REFRESH command (to make it recognize updated members in it libraries), MVS validated this as if the operator were actually writing to each of the libraries.

Assigning Security Labels to Critical Data Sets

In an MLS system, when the option to protect write-down of data has been activated, a security administrator should provide security labels for any system data sets that are not labeled, in addition to writing DAC access rules. See the Labeling Catalogs and Critical Data Sets section in the “Implementing and Administering a Multilevel-Secure System” chapter for a list of security labels that are recommended for critical data sets.

Protecting Resources

The security administrator must ensure that all resources that are considered “classified”, have been assigned security labels. When MLS is active in MLS mode, CA Top Secret will fail any attempt to access a classified resource that violates MAC label dominance checking rules.

Identifying and Classifying Users

All users must be identified and authenticated in an CA Top Secret environment. In addition, all users entering the system should have a security label.

Creating Acids

In an MLS system, each user entering the system must be assigned a unique acid.

Assigning Security Labels to Users

In an MLS system, security labels should be assigned to all users who require them before MLS or any other MLS options have been activated on the system.

Establishing JCL Standards

In an MLS system, all jobs must be submitted by users identified to CA Top Secret or be associated with an CA Top Secret user. If a subject submits a job from TSO (using the SUBMIT command, for example), the job ordinarily inherits the subject's acid. However, if the subject wants a job to run under a different acid, the JCL must identify that acid and its password.

An acid and password can also be specified through the USER= or PASSWORD= keywords on the JCL JOB statement. The password is not printed in the JCL listing.

Note: Good security practice dictates that users should not submit jobs under other people's acids. Using another's acid would require knowing that person's password, which would cause loss of accountability. However, it is acceptable for a person who is assigned more than one acid to sign on with one of those acids and submit jobs under it.

In an MLS system, a user can also specify a security label for a job in the SECLABEL= keyword on the JCL JOB statement. If a security label is specified, the user must be authorized to specify the seclabel. The job will then be assigned that security label.

Example: JCL standards

```
//USERAX JOB (40100000),'TEST',MSGCLASS=A,MSGLEVEL=(1,1),
//   NOTIFY=USERA,SECLABEL=SYSHIGH
```

Defining Acids for Started Tasks

A security administrator must define acids for several z/OS MVS started tasks before IPLing the system. The following table lists which started tasks require acids. All of these acids have an STC attribute. No other attribute need be specified.

Started Task	Description
IOSAS	I/O system address space
XCFAS	Cross-coupling facility address space
LLA	Linklist Lookaside started task
BLSJPRMI	IPCS initialization task
VLF	Virtual Lookaside Facility address space
INIT	z/OS MVS initiator
DEALLOC	Deallocation started task
IEEVMPCR	MOUNT command

Defining Resource Rules for LLA Started Task

When the LLA started task starts up, and whenever an F LLA,REFRESH operator command is issued, LLA validates its access to the linklist data sets. It does this by means of RACROUTE REQUEST=AUTH calls with a class of FACILITY and an entity name of CSVLLA.*dsname* where *dsname* is the data set name.

A security administrator must write resource rules of the FAC type (the FACILITY class is mapped into FAC resource rules) for all data sets listed in the LNKLSTxx or CSVLLAxx members of SYS1.PARMLIB.

Defining Access Rules for BLSJPRMI Started Task

The BLSJPRMI started task performs initialization of IPCS using parameters read from SYS1.PARMLIB. Because of this, a security administrator must write an access rule giving BLSJPRMI read access to SYS1.PARMLIB.

Ensuring SMS Is Active in IEFSSNxx

SMS should be active on an MLS system. This does not mean that all, or any, volumes, on the system must be SMS-managed, merely that SMS must be active. This is necessary because having SMS active changes the code path for many allocation functions.

To activate SMS, the following line must be present in the IEFSSNxx member of SYS1.PARMLIB:

```
SMS,IGDSSIIN,'ID=00,PROMPT=DISPLAY'
```

The *xx* in IEFSSNxx represents the suffix specified in the keyword of the IEASYS00 member of SYS1.PARMLIB. If no line beginning with “SMS” is present in the IEFSSNxx member, this line should be added after the last line. If an SMS entry is present, but does not specify IGDSSIIN, it should be changed to do so. The ID=00 and PROMPT=DISPLAY keywords can be changed as needed. See SMS documentation for details on these parameters.

Moving Forbidden Modules Out of System Libraries

Some modules shipped with z/OS MVS and resident in link list or LPA libraries, implement features that should not be allowed in an MLS system. These modules should be moved to other, non-APF authorized libraries or deleted in order to disable these features:

- CIPOPRT-This 3800 offline utility should be moved out of SYS1.LINKLIB
- MVSSERV-This client-server platform should be moved out of SYS1.LPALIB

z/OS UNIX SYSTEM SERVICES

In environments where users move across multiple hardware platforms and operating systems to access numerous applications, security is a major concern. Sites need the same control over data and resources accessed in an open system as they have in their mainframe environment. CA Top Secret offers security for such open environments by supporting z/OS UNIX System Services (UNIX) and the standards developed for a Portable Operating System Interface (POSIX). In addition, CA Top Secret supports security labels and security-label checking in a UNIX environment. This chapter explains how to implement and use security labels for UNIX functions in an CA Top Secret MLS environment.

Support for MLS UNIX

In addition to the basic UNIX functions that are supported by CA Top Secret, the following are also supported when MLS is active:

- Security labeling in a zSeries file system (zFS) and Hierarchical File System (HFS)
- Requiring security labels for all files and directories
- Requiring security labels for all IPC objects
- Assigning security labels to files and directories at the time they are created based on the security label of the parent directory or user (the label is stored in the File Security Packet (FSP))
- Assigning security labels to IPC objects at the time they are created (the label is stored in the IPC Security Packet (ISP))
- Assigning a security label to a UNIX user
- Assigning a security label to a UNIX session created by the **su** command
- Allowing authorized users to classify files and directories that were not labeled at the time they were created with the UNIX **chlabel** command (zFS only)
- Using symbolic links to define a different home and program for each security label that a user is authorized to use
- Controlling access to files and directories based on security labels
- Allowing communication between IPC objects only if their security labels are equivalent
- Validating user authorization to the following socket functions based on security labels: givesocket, takesocket, sendmsg, and recvmsg
- Allowing authorized users to display the security labels of files, directories and IPC objects using UNIX commands
- Hiding the names of files and subdirectories in a directory
- Defaulting a security label for a user who is entering the system from a remote IP address (using **rlogin**)
- Validating that the security label of a subject who is accessing the system from a remote IP address is equivalent to the security label of the IP address
- Bypassing security label checking for trusted users. If a user is identified in the system with an acid that has bypass attributes (NODSNCHK, NORESCHK, NOSUBCHK, NOVOLCHK, NOLCFCHK) he is considered trusted and MLS validation is bypassed.
- Displaying, activating, and deactivating individual user write-down protection status with the UNIX **writedown** command

Restrictions

The following restrictions apply when MLS is active on an CA Top Secret system:

- The HFS physical file system (PSF) does not support name-hiding
- The HFS PSF does not support the UNIX **chlabel** command
- The HFS PSF must be mounted in read-only mode. HFS can only properly be used in read-only mode (in this mode, security labels are not assigned to UNIX files and directories nor is security label checking performed); if used in read-write mode, MLS validations based on security labels will be inconsistent and inaccurate
- Mount an HFS file system only on a directory that has the same security label as the file system's root to prevent conflicts between directory search access and name-hiding
- Assign the security label SYSMULTI to mount point directories

Configuration Checklist

This checklist describes the software configuration requirements when MLS is active on an CA Top Secret system.

Requirement	Complete
Using security labels	□
Entering the system	□
Changing the user ID of a session	□
Accessing files and directories	□
Accessing IPC objects	□
Using signal services	□
Using the ptrace service	□
Displaying security labels	□
Identify and classify users	□
Define users	□
Assign security labels to users	□
Assign security labels to the OMVS started task	□
Assign security labels to the zFS started task (optional)	□
Assign a home directory and program for each user's security label (optional)	□

Requirement	Complete
Configure an HFS file system	□
Assign a security label to an HFS system data set	□
Assign a security label to a root directory	□
Default a security label for an HFS file system	□
Assign a security label to a subdirectory	□
Assign security labels to files and directories	□
Assign a security label to an IPC object	□
Migrate an HFS file system to a zFS file system (optional)	□
Configure a zFS file System	□
Protect the cron daemon	□
Activate name-hiding (optional)	□
Use the **chlabel** command to label existing files and directories	□
Use DFSMSdss instead of **pax** or **tar** commands for file backup and restoration	□
Establish MLS system options in a UNIX environment	□
Require security labels for files and directories (optional)	□
Require security labels for IPC objects (optional)	□
Authorize users for controlled write-down (optional)	□

Using Security Labels

Most users use a security label when they log on to the an MLS system. The rest of the time, security labels are read assigned an MLS system. Depending on what MLS options have been set by the security administrator, CA Top Secret will assign security labels to UNIX objects at the time they are created. A security administrator can also assign security labels to existing UNIX files and directories that do not have them. The following topics are discussed in detail:

- Entering the system
- Changing the user ID of a session
- Accessing files and directories
- Accessing IPC objects
- Using signal services
- Using the ptrace service
- Displaying security labels

Entering the System

When a user attempts to enter UNIX, CA Top Secret validates the user before initializing the shell, including validating the user associated with a program attempting to access resources and verifying that the user has specified a valid security label and is authorized to use it.

If the user is entering the system from a remote IP address and no security label has been specified at login, CA Top Secret will attempt to default a security label for the user from the user's port of entry (SERVAUTH class), if one exists. The user must have authorization to use the label that is defaulted by the system.

Changing the User ID of a Session

The **su** command can be issued to change the user ID associated with a session. It creates a new shell and allows the user who issued the command to operate in the shell with the privileges of a superuser or another specified user. In an CA Top Secret MLS system, the shell that is started by a user who issues the **su** command, inherits the security label of the user who issued the command, if the user has a security label. The security administrator must authorize the new user to the inherited security label, if the user has not already been authorized to it. For example:

User	Security Label
USERA	SYSHIGH

User	Security Label
USERB	SYSLOW

USERA, whose session security label is SYSHIGH, issues the **su** command and specifies userid USERB on the command. The child shell that is started will then inherit SYSHIGH as its session security label. However, USERB has only been authorized to security label SYSLOW. Therefore, the security administrator must also give USERB authorization to security label SYSHIGH, as follows:

```
TSS ADD(userb) SECLABEL(syslow,syshigh)
               DFLTSLBL(syslow)
```

Accessing Files and Directories

UNIX controls access to files and directories based on their security labels as well as POSIX permissions and access control lists (ACLs). In addition, a user cannot access a classified file or directory if his security label does not permit the access according to MAC label dominance checking rules. Security labels are assigned to files and directories by the system at the time they are created and stored in the File Security Packet (FSP). Once a security label is assigned to a file or directory, it can never be changed. However, files and directories which were created in a zFS file system before MLS was activated, and therefore are not labeled, can be assigned security labels by a security administrator with the UNIX **chlabel** command.

If the MLS option to require security labels for UNIX files and directories is active (MLFSOBJ), all files and directories must have security labels; otherwise, all accesses to these objects will be denied by CA Top Secret.

Accessing IPC Objects

UNIX allows connection between interprocess communication (IPC) objects only if their security labels are equivalent, according to MAC label dominance checking rules. Security labels are assigned to IPC objects by the system at the time they are created and stored in the ISP.

Note: The security label assigned to an IPC object is the security label of the process that created the connection.

Once a security label is assigned to an IPC object, it can never be changed.

If the MLS option to require security labels for UNIX IPC objects has been activated (MLIPCOBJ), all UNIX IPC objects must have security labels; otherwise, all requests to connect to an unlabeled process will be denied by CA Top Secret.

Identifying and Classifying Users

In an MLS system, all users and work entering the system (including started tasks, batch jobs, processes, UNIX daemons, etc.) must be identified. Each user must be assigned a unique acid. In addition, all users are assigned security labels by the security administrator or by CA Top Secret.

Using Signal Services

When the signal service **sigqueue( )** is used, the security label of the signaling process must be equivalent to the security label of the target process, according to MAC label dominance checking rules.

Using the ptrace Service

When the ptrace service is used, the security label of the process that initiated ptrace must be equivalent to the security label of the target process, according to MAC label dominance checking rules.

Displaying Security Labels

To display the security label for his current address space, a user can issue the following UNIX command:

```
id -M
```

To display the security label of a UNIX file or directory, an authorized user can issue the following UNIX command:

```
ls -M filename
```

To display the security label of an IPC object, an authorized user can issue the following UNIX command:

```
ipcs -M
```

Assigning Security Labels to Users

Once a UNIX user has been defined to CA Top Secret, the security administrator should assign security labels to each user, which will allow users access to the system and any classified resources they may need.

Assigning Security Labels to the OMVS Started Task

The security administrator should assign the OMVS started task the security label, SYSMULTI.

Assigning Security Labels to the zFS Started Task

If a zFS file system will be used in the UNIX environment, the security administrator should assign the zFS started task the security label, SYSMULTI.

Assigning Security Labels to User Home Directories and Programs

Users can logon to an MLS system with any one of multiple security labels that they have been authorized to use. As a result, a security administrator can allow a user to have a different home directory and shell program for each of their assigned security labels at logon by using the special symbolic links, $SYSSECA/ (absolute directory name) and $SYSSECR/ (relative directory name). Since a “symbolic link” is a file that contains the pathname for another file, whenever the symbolic link $SYSSECA/ or $SYSSECR/ is found in a pathname, the user's session security label is substituted by the system in place of that symbolic link in the pathname, either as an absolute directory name or as a relative directory name, depending on the symbolic link used ($SYSSECA/ or $SYSSECR/).

- **$SYSSECA/ or $SYSSECA/pathname** specifies that pathname resolution continues at the root with a directory name of the user's session security label (substituted for $SYSSECA/).
- **$SYSSECR/ or $SYSSECR/pathname** specifies that pathname resolution continues in the directory in which the symbolic link is found with a directory name of the user's session security label (substituted for $SYSSECA/).

The following examples illustrate how security label substitution for special symbolic links works.

USERA has the following OMVS segment defined:

```
TSS ADD(userb) HOME(/u/symlnka/usera) OMVSPGM(/bin/sh) UID(0)
```

USERA's home directory (**/u/users/secdev/usera)** contains the symbolic link, $SYSSECA/, which was created with the following UNIX command:

```
Lin -s $SYSSECA /u/symlnka
```

Therefore, when USERA logs on with security label, LABELA, that label will be substituted into the following resolved home directory pathname as an *absolute* directory name:

/LABELA/usera

USERB has the following OMVS segment defined:

```
TSS ADD(userb) HOME(/u/symlnkb/userb) omvspgm(/bin/sh) UID(0)
```

USERB's home directory (**/u/symlnkb/userb**) contains the symbolic link, $SYSSECR/, which was created with the following UNIX command:

```
Ln -s $SYSSECR /u/symlnkb
```

Therefore, when USERB logs on with security label, LABELB, that label will be substituted into the following home directory pathname as a relative directory name:

/u/LABELB/userb

If a user logs on without a security label, a dot (.) will be substituted in place of the symbolic link ($SYSSECA/ or $SYSSECR/), and the pathname will be resolved either from the root or the current pathname, depending on the symbolic link used ($SYSSECA/ or $SYSSECR/).

USERC has the following OMVS segment defined and logs on without a security label:

```
TSS ADD(userc) HOME(/u/symlnka/userc) OMVSPGM(/bin/sh) UID(0)
```

USERC's home directory (**/u/symlnkr/userc**) contains the symbolic link, $SYSSECA/. Therefore, the following will be the resolved home directory pathname:

/./usera

If USERC's home directory (**/u/users/secdev/userc**), instead, contains the symbolic link, $SYSSECR/, the following will be the resolved home directory pathname:

/u/./usera

When MLS is not active, security label substitution is not performed and the symbolic links ($SYSSECA/ or $SYSSECR/) are left in the pathname as relative directory names.

Note: Security label substitution can also be used with automount.

WARNING! HFS only supports security label substitution when mounted in read-only mode; however, a zFS file system supports security label substitution in any mode.

Configuring an HFS File System

Security labels are supported in both zFS file systems and HFS file systems (mounted in read-only mode). When MLS is active and no other MLS options have been set, MLS security label checking is performed only on files and directories and objects that are labeled. If an object is not labeled, it is considered unclassified, and access to it is allowed, as long as permissions and ACLs allow the access.

Assigning a Security Label to an HFS File System Data Set

A security administrator can assign a security label to the HFS file system data set on a z/OS V1R5 or later system by creating a CA Top Secret MLS resource record for it. The following illustrates how to assign a security label to an HFS file system data set named HLQ.FILESYS.NAME.

Example: HFS data set security label

```
TSS ADD(mls) DSN(hlq.filesys.name)
             SECLABEL(sysmulti)
```

Assigning a Security Label to a Root Directory in an HFS File System

When an HFS file system data set is created in an MLS system:

- If the option to require security labels for files and directories is activated (MLFSOBJ(YES)), CA Top Secret assigns the user's session security label to the root of the file system.
- If the option to require security labels for files and directories is not activated (MLFSOBJ(NO)), CA Top Secret assigns the security label from an MLS resource record for the file system data set, if one exists, to the root of the file system.
- If an MLS resource record for the file system data set does not exist and the option to require security labels for files and directories is not activated, (MLFSOBJ(NO)), CA Top Secret does not assign a security label to the root of the file system, unless the MLS option to protect write-down (MLWRITE(NO)) has been set, in which case, CA Top Secret assigns the user's session security label to the root of the file system.

Note: If MLS is inactive on an CA Top Secret system, system labeling of files and directories is not supported.

Defaulting a Security Label for an HFS File System

UNIX defaults a security label for an HFS file system at the time it is mounted by using the same security label for it that is in the MLS resource record that protects the file system data set. Because defaulted security labels can change at mount based on the value in the MLS resource record for the aggregate, they are not the same as security labels that are assigned to file systems (and are stored in FSPs), which, once assigned, can never be changed.

USS defaults a security label for an HFS file system at the time it is mounted only if all of the following requirements are met:

- A security label has been assigned to the HFS file system data set. The system will assign this security label for the HFS file system.
- The MLS option to require security labels for files and directories has been activated before the file system is mounted
- The HFS file system is mounted in read-only mode
- The root directory of the file system does not already have a security label assigned to it

Once the file system is unmounted, it no longer will have a security label. If the file system is reclassified with a new security label by changing the MLS resource record for the file system data set and then remounted, the file system will be assigned the new security label.

Assigning a Security Label to a Subdirectory

When the UNIX mkdir command is issued in an CA Top Secret MLS system:

- If the security label of the owning directory is not SYSMULTI, CA Top Secret assigns the owning directory's security label to the FSP (subdirectory)
- If the security label from the owning directory is SYSMULTI, and a security label is not passed in the system CRED, CA Top Secret assigns the security label of the requesting user's address space to the FSP (subdirectory)
- If the security label from the owning directory is SYSMULTI, and a security label is passed in the system CRED, CA Top Secret assigns the security label from the CRED to the FSP (subdirectory)

Assigning Labels to Files and Directories in an HFS or zFS File System

When a file or directory is created in a zFS or HFS file system, CA Top Secret assigns a security label at the time the file, directory or symbolic link is created as follows:

- If the parent has no security label, no security label is assigned to the new file or directory
- If the parent has a security label which is not SYSMULTI, the parent's security label is assigned to the new file or directory
- If the parent's security label is SYSMULTI, the user's session security label is assigned to the new file or directory.
- If the parent's security label is SYSMULTI, and the user session does not have a security label, no security label is assigned to the new file or directory. However, if a security label was not assigned to a file or directory at the time it was created, the security administrator can assign a security label to it with the **chlabel** command.

Important! The UNIX chlabel command may only be issued in a zFS file system. It will not work in an HFS file system. Once a file or directory has been assigned a security label, it cannot be deleted or changed with the chlabel command.

The following lists the security labels that are recommended for files, directories, and symbolic links, which have not already been assigned security labels by the system.

Directory/File	Security Label
automount-managed file system	*seclabel*
/bin and contents	SYSLOW
/lib and contents	SYSLOW
root	SYSMULTI

Directory/File	Security Label
root, symbolic links in: /tmp /dev /etc /var	SYSLOW
/samples	SYSLOW
/SYSTEM	SYSMULTI
/SYSTEM/tmp mountpoint	SYSMULTI
/SYSTEM/dev mountpoint	SYSMULTI
/SYSTEM/etc mountpoint	SYSMULTI
/SYSTEM/var mountpoint	SYSMULTI
/SYSTEM, symbolic links in: /SYSTEM/tmp /SYSTEM/dev /SYSTEM/etc /SYSTEM/var	SYSLOW
/u	SYSMULTI
/u, automount-managed directory	SYSMULTI
/u, symbolic link for security label substitution	SYSLOW
/u/seclabel mountpoint directories	Seclabel
/usr and contents	SYSLOW
/usr/lpp and contents	SYSLOW
/usr/man and contents	SYSLOW

Assigning a Security Label to a UNIX IPC Object

When an ISP is created in an MLS system:

- If the option to require security labels for UNIX IPC objects is activated (MLIPCOBJ), CA Top Secret assigns the security label of the creating process to the ISP.
- If the option to require security labels for UNIX IPC objects is activated (MLIPCOBJ(YES)), but the creating process does not have a security label, the system will fail the attempt to create the ISP (authorization failure).

Note: If MLS is inactive on an CA Top Secret system, system labeling of UNIX IPC objects is not supported.

Migrating an HFS File System to a zFS File System

When MLS is active, do not mount an HFS file system in read-write mode. Instead, copy or move it to a zFS file system; otherwise, MLS will not be fully supported. The following MLS features can only be used in a zFS file system and cannot be used in an HFS file system:

- Protecting the cron daemon
- Name-hiding
- UNIX **chlabel** command

Since the **chlabel** command cannot be used in HFS file system to label files and directories that were created before MLS was activated, if the option to require security labels for files and directories is set (MLFSOBJ), all attempts to access these unlabeled files and directories will fail (if the MLS mode option is set to MLS), and may prevent users from doing their work. Therefore, it is recommended that for full MLS support in a USS environment, you migrate all HFS file systems that require protection from data disclosure and declassification to zFS file systems.

For more information on how to migrate an HFS file system to a zFS file system, see the section, Migrating your HFS version root to a zFS version root with security labels, in the IBM *z/OS V1R5 Planning for Multilevel Security* manual.

Configuring a zFS File System

A zFS file system follows most of the same configuration requirements as an HFS file system, as previously described. However, the following MLS features can only be used in a zFS file system and cannot be used in an HFS file system:

- Protecting the cron daemon
- Using name-hiding
- Using the UNIX **chlabel** command

Protecting the cron Daemon

The **cron** daemon, which runs commands on a schedule, does not support MAC security label checking. Therefore, it is recommended that you do not allow general users to run cron jobs, including crontab jobs and batch shell commands in an CA Top Secret MLS system.

To protect the **cron** daemon, do the following:

- Define and assign a unique security label (for example, LBLCRON) to **cron** that contains the highest security level defined in the system and one unique category that no other security label has. This will prevent any user, other than a user logged on with the security label, SYSHIGH, from using **cron**.
- For information on how to define and assign security levels, categories, and security labels, see the “Implementing and Administering a Multilevel-Secure System” chapter.
- Using the **chlabel** command, assign the newly defined security label or SYSHIGH to the following directories: /usr/spool/cron/, /usr/spool/cron/crontabs, and /usr/spool/cron/atjobs, by issuing the following commands:

```
chlabel lblcron /usr/spool/cron/
chlabel lblcron /usr/spool/cron/crontabs
chlabel lblcron /usr/spool/cron/atjobs
```

Using Name-Hiding

Name-hiding is an CA Top Secret global system option which, when activated by a security administrator, prevents users from displaying the names of files and directories which their security label does not authorize them to see. It prevents users from knowing about the existence of files and directories to which they do not have read access. However, if a user requests to view the name of a specific file or directory, he will be able to see the name, although he may not be able to access the data.

To activate name-hiding for UNIX files, set the control option (MLNAME(YES), which will do the following:

- If MLS is active, a MAC label dominance check will be performed for each file or directory name to determine whether the user can view it.
- When name-hiding is active, the user's security label must dominate the file or directory's security label to see the name of the file or directory.
- When name-hiding is active, the user's security label must dominate the symbolic link's security label to read the link.

To deactivate the name-hiding option, turn off the name-hiding option (MLNAME(NO)).

Important! MLS must be active in a zFS system to support name-hiding of UNIX files and directories. Name-hiding is not supported in HFS systems.

Note: Name-hiding degrades the performance of a system. Do not activate name hiding if any system sharing the CA Top Secret databases does not meet the minimum software requirements for MLS support. Use of the name-hiding option should not cause problems on these systems, but it does not provide full protection on these systems. You must be operating at z/OS R1V5 or later to activate name hiding-in an CA Top Secret system.

Using the UNIX chlabel Command

The **chlabel** command can be issued by an authorized user to assign security labels to files and directories that were created before MLS was activated, that therefore, do not have security labels.

Use DFSMSdss for File Backup and Restoration

The UNIX **pax** and **tar** commands do not support backup or restoration of security labels assigned to files. Therefore, it is recommended that DFMSMdss be used instead of **pax** and **tar** commands for file backup and restoration.

Establishing MLS System Options in a Environment

The security administrator is responsible for maintaining the following security options that control MLS in a environment:

- Require security labels for files and directories (MLFSOBJ)
- Require security labels for IPC objects (MLIPCOBJ)
- Authorize users for controlled write-down

Requiring Security Labels for IPC Objects

The MLIPCOBJ option specifies whether security labels are required for UNIX IPC objects, such as semaphores, in an MLS system. This option should be set if you want to use MAC protection and security labels for UNIX IPC objects.

Important! Before activating this option, let the system run with MLS active, so that the system is able to label IPC objects as they are created, without failing the access requests.

Requiring Security Labels for Files and Directories

The MLFSOBJ(YES) option specifies that security labels are required for UNIX directories and files in an MLS system. This option should be set if you want to use MAC protection and security labels for UNIX files and directories.

Important! Before activating this option, make sure you have properly set up HFS and zFS file systems with MLS active, including migrating HFS files systems to zFS file systems, as necessary; otherwise, users may be erroneously prevented from accessing files that they need to do their work.

Authorizing Users for Controlled Write-Down

The UNIX **writedown** command can be issued by an authorized user to override global write-down protection on a system by setting, resetting or querying the setting of the write-down mode for his address space.

When MLS is active on the system and the control option MLWRITE(NO) has been set, and a user has been given READ access to the IRR.WRITEDOWN.BYUSER resource in the IBMFAC class, the user is authorized to issue the UNIX **writedown** command to either:

- Set the mode to allow himself to write-down during his current session
- Set the mode to prevent his ability to write-down for his current session
- Display the write-down mode for his current session
- Reset the write-down mode for his current session to the default setting with which he entered the system

When a user enters the system and has UPDATE access to the IRR.WRITEDOWN.BYUSER resource, CA Top Secret will, by default, allow the user to write-down without issuing the UNIX **writedown** command during his session, although the user can, if he wishes, issue the command if he has both UPDATE and READ access to the IRR.WRITEDOWN.BYUSER resource.

To allow an CA Top Secret user to control write-down for himself by issuing the UNIX **writedown** command, do the following:

- Activate the MLS control option (MLACTIVE(YES))
- Disable write-down (MLWRITE(NO))
- Determine which users may need to write data from a higher to lower security classification when MLS is active on the system and write-down is protected globally with the NOMLWRITE option.
- Create a resource rule(s) for the IRR.WRITEDOWN.BYUSER resource in the IBMFAC class giving READ access to those users who need to issue the **writedown** command. For example:

```
TSS PER(user01) IBMFAC(irr.writedown.byuser)
                ACCESS(read)
```

The UNIX user can now issue the **writedown** command

Examples: write-down

To deactivate and display your current write-down mode, enter:

```
> writedown -ip
inactive
```

To activate and display your current write-down mode, enter:

```
> writedown -ap
active
```

To reset and display your write-down mode, enter:

```
> writedown -dp
active
```

Chapter 5: Auditing a Multilevel Secure System

This section contains the following topics:

Security Events

An MLS system must create, maintain, and protect the audit records for all accesses to protected objects. Determine which security events to audit.

The following events must always be audited in an MLS system:

- Use of identification and authentication mechanisms
- Introduction of objects into a user's address space
- Deletion of objects from a user's address space
- System access
- Use of privileges

CA Top Secret creates records in the Audit/Tracking File or SMF data sets when:

- A user attempts to sign on or access the JES system and CA Top Secret rejects or allows the access for any reason
- A user with the AUDIT attribute set in his acid record accesses the system
- A user with the TRACE attribute set in his acid record accesses a data set or resource
- A user attempts to access a data set or resource and it is denied or allowed by CA Top Secret
- A user accesses a data set or resource and CA Top Secret is instructed to log the access due to a system option or permit entry
- An account manager adds, modifies, or deletes an acid record
- An operator issues a MODIFY TSS command, and each time he stops or starts CA Top Secret
- Security labels are added, updated or removed
- A subject attempts to access an object
- An administrator creates, updates, or deletes MLS-related records
- A job, STC, or TSO session tries to enter the system
- A security label violation occurs during processing of a RACROUTE REQUEST=DIRAUTH call
- A “trusted” user enters the system or is allowed access to a resource during MAC label dominance checking
- A user accesses a resource, both the user and the resource have a seclabel, and seclabel auditing is set 'on' on either seclabel

Audit Access to Resources

Any resource, specific resources, or all those matching a generic prefix, can be audited. All access attempts are recorded in the Audit/Tracking File and/or the SMF datasets.

To audit accesses, enter:

```
TSS ADD(AUDIT) resource(resource-name)
```

Audit by Seclabel

You can audit individual seclabels. (Except for SYSHIGH, SYSLOW, SYSNONE, and SYSMULTI). To specify the auditing, use the keyword MLAUDIT and specify an access type. To audit a seclabel with no specific access type, enter ALL.

```
TSS ADD(MLS) SECLABEL(LABEL1)
             MLAUDIT(READ,UPDATE)
             SECLEVEL(10)
```

The following access types are allowed:

READ, CREATE, WRITE, CONTROL, UPDATE, SCRATCH, FETCH, ALTER, and ALL.

Any other access type entered defaults to READ.

To activate the auditing feature on the seclabels, set the control option MLSECAUD to YES.

```
TSS MODIFY(MLSECAUD(YES))
```

Important! Seclabel auditing for all security labels in the system severely degrades performance and therefore auditing every security label in the system is not recommended.

To see the SMF records cut from the seclabel auditing, run TSSUTIL and specify the long report or run TSSTRACK. The seclabels involved in the event are displayed and the record is marked with +A (audited event). The audited seclabel(s) are marked with an "*".

Report Generation

The CA Top Secret reports and utilities audit the activity on your system. They let you format the Audit/Tracking File or SMF records used to obtain user responses and reactions to controls enforced by CA Top Secret.

The available reports are:

TSSUTIL

Batch report of any security related events that have been logged to the Audit/Tracking File and /or SMF. Multiple and varied reports can be produced and events can be archived to tape/DASD.

TSSTRACK

This utility can be used to monitor security related events from an online terminal in a real-time manner. It also can go back to a specified date and time to focus on selected events.

TSSAUDIT

This batch utility monitors changes made to the Security File and sensitive z/Os facilities and data areas.

TSSCHART

This utility lets you generate the ACIDs and owned resource relationships within the CA Top Secret database in the form of an organization chart.

TSSSIM

Enable the simulation of access attempts to resources to test and verify resource permissions. It can aid an auditor in deciding whether or not users should have access to particular resources.

TSSCFILE

This utility produces a fixed-format output file whose records closely parallel the output of a TSS LIST command. The output can then be used to generate custom reports.

TSSOERPT

The z/OS UNIX System Services (UNIX) report identifies user activity in a USS environment. CA Top Secret logs security events under USS to SMF using the standard CA Top Secret SMF record. Log records are written for any security event that denies the user access to a USS facility. This report includes the UID, GID, and security label of the user involved in the attempted access as well as the security label of the resource in the attempted access.

CA Earl®

CA Earl allows you to run the CA Top Secret reports. This gives you the capability of generating customized reports to accommodate local installation requirements.

Reports for Auditing

In all cases, the records in a given CA Top Secret report can be affected by:

- The report generator JCL, which has parameter fields that enable you to specify various options and selection criteria
- The actual Audit/Tracking File and/or SMF data sets used for input
- The authorities of the user who ran the report

When you review the reports:

- Include all proper inputs
- Make sure that the selection parameters do not inappropriately exclude important records, such as records from a certain time period or for certain data set names or acids
- Remember that various system options and the use of exits can affect the data that is or is not included in the report

Part of the CA Top Secret audit should be directed to review the normal processing of the CA Top Secret reports. Verify that the reports are produced regularly and that they include all appropriate records. The timely and proper use of the CA Top Secret reports is an important aspect of internal controls and should be carefully reviewed. The CA Top Secret report generators can also be executed at z/OS MVS/TSO sites by means of the CA Top Secret ISPF panels.

Report Execution

In general, you can execute the CA Top Secret reports with:

- JCL supplied with CA Top Secret. The CA Top Secret distribution tape provides a prototype JCL procedure that you can use to generate CA Top Secret reports. The SAMPJCL file contains the JCL..
- ISPF panels. With these panels, you can create a report generator online and have the results displayed on your terminal screen.
- CA Top Secret utilities through the TSO CALL command

Vulnerabilities of Misused Audit Privileges

The potential for misuse or abuse by an auditor includes:

- The auditor may lose audit log consistency due to failure to audit required events.
- The auditor's actions may cause loss of privilege to audit files or loss of privacy due to misuse of audit file privileges (shared access with unauthorized users).
- The auditor may deny service to administrative and other users. For example, the auditor may turn on the audit of a user action while processes of that user are already in execution. This may cause a large number of inconsistent or unusable events to be written, filling up the audit logs. Auditor inaction in emptying audit logs may cause the system to stop.

Chapter 6: Operating a Multilevel Secure System

This section contains the following topics:

Introduction to MLS Operation

In an CA Top Secret MLS system, the console operator is considered trusted. Because operators perform tasks that alter the state of the system, their actions must be auditable. Console operators are required to log onto the system and undergo identification and authentication before performing any tasks. Because of the functions they perform, operators in an MLS system should log on with a SYSHIGH security label to ensure that they can issue commands to meet user requests.

Physical security should also be used to prevent other users from logging on to the system consoles. Even when not logged on, consoles display all message traffic.

Operator Consoles and Commands

Operator consoles can be forced to sign on before issuing commands. This is controlled by the LOGON keyword on the DEFAULT statement in the CONSOLxx member of SYS1.PARMLIB. The modes of operation are:

LOGON(OPTIONAL)

In this mode, operators can log onto consoles if they choose to. The commands are validated and logged using the user ID with which they log on. Consoles that are not logged on use a special user ID of *BYPASS*.

LOGON(REQUIRED)

In this mode, operators must log onto consoles before they can enter commands. An exception is made for the master console, which will accept commands before the security system is up and able to process logons. (The exception is necessary to make it possible to enter system startup parameters.) Before security is initialized, only the master console can enter commands. After security is initialized, the entry area of all consoles (including the master) displays a formatted area where the operator can enter a user ID, password, group, and security label (SECLABEL). The operator must log on before any commands are accepted. When the operator logs off, the formatted logon prompt is once again displayed. When a console is varied offline, it is automatically logged of.

LOGON(AUTO)

In this mode, consoles are automatically logged on using a user ID that is equal to their console ID. This can provide logging and access control when physical security makes it possible to correlate a specific console and the operator that uses it.

In an MLS system, LOGON(REQUIRED), should be specified.

When an operator logs on to the console, a RACROUTE REQUEST=VERIFY call is issued to validate the identification and authentication (I&A) information entered by the operator. A SECTRACE for the logon looks like this:

```
TRACE ID:   TRACE001 JOBNAME: CONSOLE ASID: 0008 USERID: N/A
PROGRAM:    IEECB902 RB CURR: IEECB902 APF: YES SFR/RFR: 0/0:0
RACROUTE    REQUEST=VERIFY,REQSTOR='IEECB902',SUBSYS='CONSOLE ',
   DECOUPL=YES,MSGSP=0,WORKA=,ACEE=,ENCRYPT=YES,ENVIR=CREATE,
   LOG=ASIS,PASSCHK=YES,PASSWRD=*SUPRESSED'MACRO',POE='LRGCON,
   RELEASE=7708,SECLABL='SYSHIGH',SESSION=CONSOPR,SMC=YES,

   STAT=ASIS,SUBPOOL=239,USERID='USERA'
```

When an operator enters a command, a RACROUTE REQUEST=AUTH call is issued, with a class of OPERCMDS.

System Initialization and Shutdown

The START and STOP commands can be protected using resource rules for the OPERCMDS class. Your site may want to generate a logging record when CA Top Secret is stopped or restarted.

When CA Top Secret is stopped, the console operator can vary the master console using the VARY MSTCONS command. Any activities performed by the operator while CA Top Secret is down are not logged. However, an SMF logging record is generated whenever CA Top Secret is started or stopped. This is displayed on the TSSUTIL report. The report should be reviewed for unscheduled CA Top Secret shutdowns.

System Clocks

The accuracy of the system clock can affect routine operations (such as scheduling tape erasures and archiving) and the accuracy of audit records. Ensure that authorized operators perform the setting of these clocks and that the operator commands used to set the clocks are protected.

Messages

In an CA Top Secret MLS system, the addressee, regardless of their security label, can receive a message sent by the console operator. The operator message is considered to be labeled SYSLOW. Since any user can send messages to the operator, operators must be careful not to echo information received in user messages.

System operators must not only be cleared to the system at a high level, they must also be cautioned to think twice before they send messages to users, to make sure they are not revealing classified information to those not cleared to see it. In MLS systems, if a resource rule was created for the SMESSAGE class, the operator may be prevented by DAC checking as well.

To send messages to users, the operator should send a public broadcast notice directed to all users. Because any user can view the information in the SYS1.BROADCAST data set, operators should use care to not disclose sensitive information about the activities they are performing in broadcast notices.

Example: messages

In this example, USERA, who is logged on with security label, SECRET, sends the following message to the console operator: “Will the system be up Saturday? We're attacking the Duchy of Grand Fenwick then, and we need the system.”

The operator is busy, and he sees the message just as it rolls off the screen. He misses the acid, so he sends out a broadcast message: “Who was asking me about attacking Grand Fenwick?” This message goes out to all users, even those logged on below SECRET. The operator has just revealed the existence of the attack, which is classified.

Printed Matter

In an MLS system, operators, rather than end-users, should be responsible for separating and distributing printed classified output. There are procedures that operators in an MLS system should follow to manage deferred-printing mode page printers. Deferred-printing mode printers are those that select output from the JES2 output queues, rather than being under direct control of a particular job. This should be the only mode allowed for page printers.

Printing on all page printers is done through PSF. The PSF subsystem controls all paged, hardcopy printing in CA Top Secret. To reduce the chance of users tampering with separator pages, PSF ensures that all printing is identified with the user who submitted the print job. It does this by putting the user's acid and an unforgeable, randomly assigned number on the beginning and ending separator pages for each job. Operators must check the numbers on the beginning and ending separator pages to ensure that they match and are authentic. If they do not, the output stack should be searched further for a matching ending page. Since the numbers are determined when the job is printed, rather than when it is run, the job cannot find out what the number is going to be to simulate authentic separator pages. To use security separator pages, replace the PSF-supplied default job header and trailer routines with PSF exit routines, APSUX01S and APSUX02S, from SYS1.SAMPLIB.

Important! Although PSF enforces “print labeling”, the practice of putting security labels on all hard copy printed output; CA Top Secret does not support print labeling in an MLS environment.

Dumps and Traces

In an MLS system, the security administrator should assign security labels to all dumps and traces. The following rules should be followed for labeling dumps and traces:

- If a dump data set contains system data, it should be labeled SYSHIGH
- If the Generalized Trace Facility (GTF) creates the data set, or if the trace data set contains data from many address spaces, it should be labeled SYSHIGH.
- If users, such as SYSDUMP, SYSABEND, and SYSMDUMP data sets, generate dumps these data sets should be labeled with the label of the job that generated the dump.
- If the system fails, SMF records may be waiting to be written to DASD. To recover these records from a system dump, a user with a SYSHIGH label must do the following:
  - Allocate a VSAM data set to receive the SMF records.
  - Use the IPCS SMFDATA subcommand to recover the SMF records that remain in buffers.
  - WARNING! CA Top Secret does not assign a security label to all dumps and traces based on the original security label of the data. The security administrator must assign labels, as necessary.

The security administrator should assign security label SYSHIGH to any users who must work with system dump data sets so they are authorized to log on with the SYSHIGH label. Access rules for the dump and trace data sets should specify LOG to keep a record of the activity by these users.

Testing Devices and the System

Part of the operations of the system includes testing online devices and the integrity of the system. By running these tests periodically and ensuring that they perform satisfactorily, you ensure the hardware performs correctly and that the system, including the security-relevant code, works as coded. Several IBM tools are available for these purposes. The processor complex exerciser (PCX), channel subsystem exerciser (CSX), online test executive program (OLTEP), and online test standalone executive program (OLTSEP) are some of the tools that enable testing of various pieces of hardware such as the CPU and I/O devices.

Disk Pack Processing

In an MLS system, protection of disk data sets is provided by data set name.

Note: An object does not retain its security label when transferred from virtual storage to a device attached to the system such as a DASD device.

Tape Processing

In an MLS system, access to tape data sets is validated based on whatever tape management system is use at your installation.

When a data set on tape is deleted, it is necessary for the operator to manually degauss (erase) the tape before it can be reissued for use by another data set. This is necessary to prevent data scavenging, where a user creates a tiny data set at the beginning of the tape, then reads the data that follows the data set to see the data from the previous user.

Note: An object does not retain its security label when transferred from virtual storage to a device attached to the system, such as a tape volume.

Temporary Data Set Protection

In an MLS, during system failure, initiator failure, or system restarts, temporary data sets may be left on a DASD volume. These data sets are protected from unauthorized disclosure. A user with the NODESCHK attribute and a security label of SYSHIGH, should periodically scratch any residual temporary data sets on disk volumes. The SECURITY privilege and SYSHIGH label permit the user to access residual temporary data sets. However, these accesses are logged, and are reported in the TSSUTIL report.

Security Label Change Prevention

In an MLS system, even security-irrelevant requests may require that the operator be able to change the security label of a device for the request to be honored. CA Top Secret preserves the integrity of a site's security-labeling scheme by allowing only SCA's with the MLSADMIN administrative authority acid record to change the security label of a data set.

Backing Up the CA Top Secret Database

All CA Top Secret records are stored in the CA Top Secret security file database. The records stored in the database include:

- ACID records
- Access permissions
- MLS-related records

Ensure the integrity of this database and ensure that it is backed up regularly. Be prepared with accurate, tested backup procedures to ensure the proper operation of the system.

Chapter 7: Modifying a Multilevel Secure System

This section contains the following topics:

Introduction to MLS Modification

The trusted computing base (TCB) components of an CA Top Secret MLS system include hardware and software. Changes to the TCB must be authorized to ensure the TCB remains trusted and is protected from unauthorized access. Any authorized programs or site-developed authorized code added to the TCB must adhere to the same or equivalent controls and checking as the TCB performs to maintain integrity. Even though integrity is maintained, the addition of any authorized software outside of the TCB may compromise MLS.

System Integrity

System integrity prevents an unauthorized program from:

- Bypassing storage or fetch protection
- Bypassing OS password or VSAM password
- Bypassing security checking
- Obtaining control in an authorized state

z/OS accomplishes this by using hardware and software features.

Software features ensure that only authorized programs can access functions that might compromise integrity. To be authorized, a program must:

- Execute in supervisor state
- Execute with a program status word (PSW) key of 0-7
- Be authorized by the authorized program facility (APF)

If a program satisfies one of these requirements, it can access a restricted supervisor call (SVC), certain exit and I/O appendages, or another system function that could compromise the security and integrity of the system.

Possible Integrity Exposures

In general, a software program does not harm system integrity if it:

- Uses only unauthorized and unrestricted MVS interfaces
- Runs only as a problem program
- Does not modify z/OS MVS

System integrity of a secure system might be compromised if a program:

- Runs authorized or with special privileges
- Uses an SVC, program call, exit, or I/O appendage
- Modifies MVS
- Uses APF
- Places its name in the program properties table (PPT)
- Runs in supervisor state
- Runs with a PSW of 0-7
- Operates with a acid that has special CA Top Secret attributes

An authorized program could introduce integrity exposures in the following areas:

- Supplying and verifying addresses for user storage areas
- Supplying and verifying control blocks and addresses
- Identifying and validating resources
- Having SVC routines call other SVC routines
- Accessing control program and user data
- Serializing resources

IBM provides information about guidelines that enable an authorized program to use system and user resources. These guidelines include:

Protection

Ensures the protection of sensitive data owned by authorized programs, the protection of user data from unauthorized users, and the protection of sensitive functions, such as SVCs.

Identification

Ensures that system and user resources are not counterfeited by separating these resources and that authorized programs can identify which program has responsibility for validating user data.

Validation

Ensures the validity of requests to use main storage and system resources by unauthorized programs and the validity of data passed by authorized programs.

Serialization

Ensures that access to system resources is serialized and that a validation process does not alter variables before the operation being validated is complete.

Acceptable Modifications

Any product that runs authorized and is not part of the TCB is not considered part of an MLS TCB system.

Important! This does not mean that software that is not part of the TCB will not run on the system.

CA Top Secret Features Not Part of a TCB Configuration

The following CA Top Secret features are not part of a TCB configuration:

- CA Top Secret CICS
- CA Top Secret IMS
- Distributed Database (DDB)
- Command Propagation Facility (CPF)
- Cache facility
- Record-level protection (RLP)
- CA Top Secret Exits
- VTAM Common Sign-on Managers
- PSF print labeling

Appendix A: Bibliography

The following publications may be required to install and configure an CA Top Secret MLS system:

- IBM z/OS Version 1 Release 5 *Assembler Guides*
- IBM z/OS Version 1 Release 5 *CICS Guides*
- IBM z/OS Version 1 Release 5 *Communications Server Guides*
- IBM z/OS Version 1 Release 5 *DB2 Version 8 Guides*
- IBM z/OS Version 1 Release 5 *DFP Guides*
- IBM z/OS Version 1 Release 5 *DFSMS Guides*
- IBM z/OS Version 1 Release 5 *DFSORT Guides*
- IBM z/OS Version 1 Release 5 *Distributed File Service Guides*
- IBM z/OS Version 1 Release 5 *IMS Guides*
- IBM z/OS *Hardware Guides*
- IBM z/OS Version 1 Release 5 *ISPF Guides*
- IBM z/OS Version 1 Release 5 *JES2 Guides*
- IBM z/OS Version 1 Release 5 *JES3 Guides*
- IBM z/OS Version 1 Release 5 *MQSeries Guides*
- IBM z/OS Version 1 Release 5 *MVS Guides*
- IBM z/OS Version 1 Release 5 *Printer Services Facility (PSF) Guides*
- IBM z/OS Version 1 Release 5 *RMF Guides*
- IBM z/OS Version 1 Release 5 *SDSF Guides*
- IBM z/OS Version 1 Release 5 *Security Server RACROUTE Macro Reference*
- IBM z/OS Version 1 Release 5 *SMP/E Guides*
- IBM z/OS Version 1 Release 5 *TSO/E Guides*
- IBM z/OS Version 1 Release 5 *UNIX System Services (USS) Guides*
- IBM z/OS Version 1 Release 5 *VTAM Guides*

The following guides are not required to install and configure an CA Top Secret MLS system, but provide important information on the National Computer Security Center (NCSC) evaluation criteria and trusted systems:

- Computer Security Requirements: Guidance for Applying the Department of Defense Trusted Computer Systems Evaluation Criteria in Specific Environments (CSC-STD-003-85)
- Department of Defense Trusted Computer System Evaluation Criteria (DOD 5200.28-STD)
- Department of Defense Password Management Guideline (CSC-STD-002-85)
- Introduction to Certification and Accreditation (NCSC-TG-029)
- Ratings Maintenance Phase Program Document (NCSC-TG-013)
- Technical Rationale Behind CSC-STD-003-85: Computer Security Requirements: Guidance for Applying the Department of Defense Trusted Computer System Evaluation Criteria in Specific Environments (CSC-STD-004-85)
- Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria (NCSC-TG-005)
- Turning Multiple Evaluated Products into Trusted Systems (NCSC Tech. Report-003)
- Use of the Trusted Computer Systems Evaluation Criteria (TCSEC) for Complex, Evolving, Multipolicy Systems (NCSC Tech. Report-002)

These publications can be ordered from the U.S. Government Printing Office at:

Superintendent of Documents
U.S. Government Printing Office
Washington, DC 20402
(202) 783-3238

Appendix B: Case Study

This section contains the following topics:

Corporate Security Policy

(First Tennessee Bank, written by Bob Wicksel)

All computer-based data and programs are corporate assets and, as such, must be protected against unauthorized access, disclosure, and/or manipulation.

The Transaction and Information Group under the direction of the Priority Committee, as part of its custodial responsibility for the main computer data and programs, will assure that certain base security controls are defined, implemented, and administered for these corporate assets. The Transaction and Information Group also has responsibility for advising the user/owner of those additional controls that can be added to provide further security beyond the base controls. The user/owner of the respective application data and programs will assume the responsibility for determining which of these additional control features will be employed within their respective functional area and for assuring that the proper ongoing administrative procedures are observed.

Base Security Controls

Base security controls for the First Tennessee host computer environment are established by the security task force. These controls are monitored and administered via the CA Top Secret security software package.

The implementation of base security controls, with the framework of CA Top Secret, will be based upon the principle of “least possible privilege”. Under this principle, initial communication with the host computer, access to data/programs and the use of computing functions will be summarily denied unless specific authorization has been granted and is resident within the CA Top Secret Security File.

Base controls 1 through 3 deal with the initiation of communication between the user and the host computer; three control levels, access, identification, and authentication must be satisfied to establish this base linkage.

Base controls 4 and 5 determine the information resources (data/programs) and computing functions the user will have access to.

Base control 6 deals with the mandatory changing of user passwords at a regular, specified interval.

Base control 7 addresses the logging and reporting of security violations.

Device Access Control

All computer terminals and card readers must have a unique, fixed hardware identification code known to the security system to communicate with the host computer. The Transaction and Information Group will assign this code to all existing devices, and, for any such devices added to the system. Attempts to gain access by any unknown device will be denied.

User Identification Control

All authorized users must have a unique personal identification code. This code must be supplied immediately after the initial host communication link is established, or further access will be denied and the communication link will be terminated. The Transaction and Information Group will assign this code based upon appropriate authorization supplied by a recognized user/owner.

User Authentication Control

All authorized users must have a unique personal password that is associated with their personal identification code. This password must be supplied immediately after their identification is successfully validated, or further access will be denied and the communication link will be terminated. Each user is responsible for the selection and protection of their personal password.

Information Access Control

All corporate information resources are owned and are identified by owner within the security system. Information resource owners must authorize access rights for each user requiring access to those resources. Access control will be automatically enforced by the security system. Attempted unauthorized access to owned resources will be denied.

Computing Function Control

All host computing functions (for example, the insertion, changing, or deletion of data; the execution of computer programs; the creation, copying, or deletion of data files/programs) will be protected by the security system. Resource owners must authorize computing function capability for each user with these requirements. Attempted unauthorized performance of computing functions will be denied.

Mandatory Changing of Passwords Control

Each authorized user must change his/her personal password every thirty days. The security system provides this capability directly to the user; therefore the responsibility for password security rests with each user. Passwords may be changed more often as necessary, but non-observance of the 30-day requirement will result in the automatic suspension of access rights.

Violation Logging and Reporting Control

All security violations, whether intentional or unintentional, will be logged when they occur. Security violation reports will be prepared and distributed to appropriate individuals, such as the security office, EDP Audit, and owner department manager.

Repeated intentional security violations by individuals may result in suspension of computer access rights, disciplinary action, and/or termination.

Unattended Terminal Locking Control

This control provides each user with the ability to lock their terminal, preventing unauthorized access, in the event the terminal is left unattended for a period of time. Attempts to gain access from a locked terminal will be denied.

Note: This control is recommended in place of an automatic time-out feature that could cause loss of data or dysfunction within a particular application.

Recommended Additional Owner Controls

In addition to the base security controls, it is strongly recommended that all information resources (data, programs) owners carefully evaluate the additional controls listed in this section and, based upon the potential risk and/or exposure, select those appropriate for their application.

The combination of base controls and the following additional controls provides the user/owner with a comprehensive set from which to build an effective, yet tailored, security program.

User Device Restriction Control

This control ties the unique personal identification code to specifically identified devices (terminals). Attempts to gain access from an unauthorized device would be denied.

User Facility Restriction Control

This control is tied to personal identification codes, and may be used to limit user access to only specified computer facilities (hardware/software access mechanisms), such as TSO, CICS, and batch. Attempts to gain access to unauthorized facilities will be denied.

Day(s) of Week Restriction Control

This control provides for the limiting of individual access privileges for specific users to specify days of the week, (for example, Monday through Friday, Wednesday only). Attempts to gain access on restricted days will be denied.

Time of Day Restriction Control

This control provides for the limiting of individual access privileges for specific users to specify hours of the day (for example, 8:00-5:00, 4:00-12:00). Attempts to gain access during restricted hours will be denied.

Central Security Administrator - Responsibilities

This section describes the functional responsibilities of the central security administrator (CSA) at First Tennessee Bank. The mission of the CSA is to administer a corporate-wide data security program designed to protect against unauthorized access, the intentional or unintentional disclosure, manipulation, and/or destruction of computer-based corporate information assets.

The objective of the First Tennessee data security program is to minimize potential exposure of the corporation. The approach to meeting this objective is based upon the following actions:

- Document existing security controls for all computer-based applications.
- Evaluate existing security controls, in terms of strengths and weaknesses, to estimate current risk/exposure levels.
- Install standardized base controls, to be applied to all computer terminals in the corporation. For example:
  - Assign each computer terminal a unique identification code to be automatically verified upon each computer access attempt.
  - Assign each authorized person a unique identification code to be automatically verified upon each computer access attempt.
  - Assign each authorized person a unique, secret password to be automatically verified upon each computer access attempt.
  - Restrict access, disclosure, manipulation, and erasure capabilities to only authorized individuals for each computer-based application.
  - Restrict the ability to initiate computer programs, copy computer data files, and perform other computing functions to only authorized individuals for each computer-based application.
  - Install automated mechanism to log and report all data security violations.

- Develop recommended additional security controls to further enhance the security level within a particular functional area:
  - Automatically enforce mandatory changing of secret individual passwords at specified intervals.
  - Automatically enforce restriction of individual users to only specified computer terminals.
  - Automatically disconnect unattended, inactive terminals after specified time limit expiration.
  - Automatically enforce restriction of individual access to specified days of the week only (for example, Monday-Friday).
  - Automatically enforce restriction of individual access to specified time of day only (for example, 8-5).
  - Automatically control each individual's ability to display, manipulate, and/or erase only authorized data files, programs, and so on.
  - Automatically control each individual's ability to initiate computer programs, copy computer data files, and perform other computing functions based upon granted authority.
- Specific CSA administrative functions:
  - Develop and install a comprehensive data security violation monitoring capability.
  - Perform regular reviews of all security violation reports and initiate appropriate corrective actions.
  - Regularly distribute security violation reports to business unit and, as required, department management for follow-up action.
  - Develop a corporate security awareness program to inform and educate all FTB employees of their security responsibilities.
  - Assist department security coordinators in the communication and resolution of highly technical security issues to other departments (T & IS).
  - Install automatic enforcement mechanisms to enforce and maintain base controls.
  - Develop and recommend additional data security controls to department security coordinators.
  - Maintain comprehensive documentation of the corporate security environment.

The CSA function will reside in the Transaction and Systems Information business unit, but will be accountable to all appropriate corporate management charged with data security responsibility. Further, the activities of the CSA will be closely monitored at all times by the EDP Audit group.

Departmental Security Coordinator - Responsibilities

Each business unit within the First Tennessee corporation is responsible for the naming of one or more department security coordinator(s) to provide first-level security administration for the major functional departments within that business unit.

This section describes the functional responsibilities of the Department Security Coordinator (DSC) at First Tennessee Bank. The DSC role is performed at the department level; all the functions and responsibilities defined here relate to that level.

The mission of the DSC is to assist the Central Security Administrator in the implementation and ongoing maintenance of the corporate data security program.

The DSC will be the focal point for all security-related communications from a department to the Central Security Administrator.

Specific DSC Administrative Function:

- To document the existing computer application requirements for his/her department, as follows.
  - Prepare a complete list of all computer terminals used in your department, including their location.
  - Prepare a complete list of all current user IDs, including the user's name, phone number, location (mail code), computer applications used, and primary functions performed.
  - Prepare a complete list of all computer files used by your department.
  - Prepare a complete list of all computer programs used by your department.
  - Prepare a complete list of any current password protected computer files used by your department.
  - Prepare a cross reference matrix that relates each individual user to the previously mentioned items, including required read, write, update, scratch, and create authority of computer files for each.
- To validate the implementation of the base controls.
- To select, install, and perform administrative functions for all recommended additional data security controls.
- To regularly receive and review security violation reports and take appropriate actions.
- To report security violations to department management for follow-up action.

Introduction to CA Top Secret

Data security is a key issue with the First Tennessee National Corporation. Computer information resources, whether in the form of programs or data, are viewed as corporate assets and therefore must be protected from either intentional, or more commonly unintentional, destruction and/or misuse.

CA Top Secret was selected by First Tennessee over IBM's RACF and CA-ACF2 because of ease of installation and low overhead requirements. CA Top Secret places no hooks into the z/OS operating system and is therefore independent of normal z/OS maintenance. It does, however, utilize the standard IBM RACF interface for inter-system communications.

Although the implementation of this package will necessitate many changes in our current environment, and the related procedures, every effort will be made to minimize disruption and loss of productivity.

Impact Areas

The implementation of CA Top Secret security will require both short and long term changes to our current operating environment.

In the short run, the immediate changes affecting current day-to-day activities are:

- Limited, or restricted, access to previously available data and libraries.
- Production problem resolution must now be coordinated with, and authorized by, production (3rd floor C/T) management.
- Previously unenforceable standards will now be enforced.

The longer-term changes will include:

- Major changes to existing library control function.
- Formalized procedures for data access authority.
- Enhancement to existing standards and addition of comprehensive standards.
- Data security reviews of new or modified applications.
- Data security reviews of new or modified hardware.

These changes will take time, but the potential benefits are substantial in terms of both asset protection and greater productivity due to a more standardized environment.

Potential Problem Areas

The following paragraphs describe potential problems that you may encounter and recommended solutions to those problems.

TSO Logon

CA Top Secret does not allow the combined entry of user ID and password when logging on to TSO. Each entry must be made separately at the appropriate prompt.

If your TSO PROFILE currently is set with a parameter of NOPROMPT, you may experience difficulty when logging on. This parameter (within z/OS TSO) requires you to enter both your ID and password in one entry (TS9999/PASSWORD).

You may check, and correct, this as follows:

- After logon, at either READY or option 6:
  ```
  ENTER => PROFILE
  ```
  This will display your current TSO PROFILE parameters.
- If you see the value NOPROMPT displayed
  ```
  ENTER => PROFILE PROMPT
  ```
  This will change the parameter value to the correct value of PROMPT.

As stated earlier, CA Top Secret does not allow the combined entry of ID/PASSWORD, but it does not prevent you from this form of entry. If you enter the combined entry, CA Top Secret will display the message DO NOT SUPPLY PASSWORD(S) WITH LOGON COMMAND, time-out the terminal for 30 seconds, and then ask you to reenter the password.

Mandatory Changing of Password

Under CA Top Secret, you are totally in control of, and responsible for, your secret password. Your TSO (z/OS) password is no longer operative and has been replaced by your CA Top Secret password.

In terms of responsibility, you have already read how CA Top Secret discourages you from entering your password at logon (except in non-display mode). In keeping with this philosophy, you will be required to change your secret password at least every 30 days. You may change it more often, as necessary.

Three days prior to your password's expiration you will begin receiving an CA Top Secret message informing you that your password will expire within three days. For your convenience, it is recommended that you change your password as soon as you get this message, at the next logon.

CA Top Secret will not automatically suspend your ID unless you totally ignore this message. That is, if you do not use your ID for 45 days, it will not be suspended, but change your password the first time you logon.

- To change your password, at logon (password prompt)
  ```
  => ENTER: OLDPASSWORD/NEWPASSWORD
  ```
- You will receive the message PASSWORD CHANGED.

Keep in mind that:

- Your password must be at least four characters long.
- You cannot reuse any of your last three previous passwords.

Every time you log on you will get a last used message displayed on your terminal. This message informs you of the date, time, facility used, system used, and a numeric count of the number of times your ID has been used. If you suspect, or are sure, that your ID is being used by another individual you should change your password immediately.

Remember:

- Do not share your password with other individuals.
- Do not write your password down and leave it where it can be obtained by others.
- Change your password regularly.

Use of Production High Level Indexes

The access rules are primarily determined based upon the current high level indexes in use. That is, production files (data sets) are generally protected from all access except for production batch processing and authorized terminal inquiry/update functions. Systems development users have been authorized read access, by group, to the applications they are responsible for.

CA Top Secret enforces the DPS standards that have been defined to it. Therefore, make every effort to conform to the current published standards. Whenever possible use the TEST prefix to eliminate conflict with production indexes.

Due to the previous inability to enforce standards, many users have created test mechanisms that either bypassed or ignored the published standards. Without exception CA Top Secret will intercept each and every one of these and prevent them from accomplishing the desired result.

All test jobs and/or libraries that you currently use should be reviewed to ensure compliance with this rule. A thorough review of your existing procedures, rather than job-by-job experimentation, will save you lost time and headaches.

Exceptions to this rule must be justified and will be addressed on a case-by-case basis.

Access Change Rules

There are many situations that will require changes to the currently defined access rules such as:

- Employee new hire, transfer, or termination
- Major system conversion/parallel
- Special project requirements
- Vendor imposed exception conditions
- Production problem resolution

Requests for access rule changes will be subject to base control standards established for all T & I users:

- Unauthorized access to production data is prohibited.
- Unauthorized access to production libraries is prohibited.

All requests for access rule changes are to be made in the following manner:

- Document the requirements and current restrictions.
- Obtain approval from your designated Departmental Security Coordinator.
- Forward approved request to the Central Security Administrator, in advance of needed date.

The above documentation may be submitted by memo, and countersigned by the DSC (Departmental Security Coordinator).

Additional information may be required- such as the access duration and specific user IDs affected. Please try to be prepared to answer these questions.

Production Problem Resolution

The function of production problem resolution will be strictly monitored to ensure adherence to the security standards. In the past, many types of problems could be resolved informally by knowledgeable personnel. This will no longer be the case. The access restrictions placed upon T & I personnel will require a formalized procedure to be initiated to obtain the needed access to resolve most production problems.

The procedure to be followed for the resolution of production problems is:

- A properly documented I/R form is required for each problem.
- The I/R must be presented to the acting central production manager (or designee, if not present) for his/her review.
- The central production manager, or designee, is authorized to grant the use of special TSO IDs that have appropriate access capabilities needed to resolve production problems.

  Note: All use of these IDs will be monitored and must be accounted for with supporting problem documentation.
- The central production manager, or designee, will record the assigned special ID number on the I/R and ensure that the problem resolver's name is also recorded on the document. The password for the ID will be given to the problem resolver.
- The problem resolver will use the ID provided to fix the problem. Upon satisfactory resolution, he/she will return I/R, with completed resolution data, to the central production manager or designee.
- The central production manager will deactivate the ID by changing the password and recording the password in a secure location.
- The central production manager will route a copy of the completed I/R to the Central Security Administrator for reconciliation to the audit trail report.

Human Resource Security Policy

Subject: Human Resource Security Policy

Effective: For All Divisions on July 1, 2000

Objective: To ensure that human resource information is protected from accidental or intentional unauthorized modification, destruction, or disclosure.

Issuing Officer(s): Vice President - Personnel

Contact(s): Vice President - Personnel/Operations

Vice President - Administrative Planning

Vice President - Internal Audit

Vice President & Treasurer - Financial Control

Cross Reference(s): None

Purpose

Our purpose in establishing a data security policy is to ensure that human resource information is protected from accidental or intentional unauthorized modification, destruction, or disclosure. Further, due to the sensitive and confidential nature of this information, it is critical that access to it be highly restricted.

Policy

Our Human Resources Security Policy defines the information to which the policy applies, who has proprietary rights to the information, individual accountability, responsibility for procedures, and outlines specific responsibilities within the organization.

Scope

This policy applies to all human resource information created or maintained within the corporation and its subsidiaries. Information includes data recorded on physical documents and on automated devices. The policy also applies to automated procedures and facilities (source code, job control, load modules), because these are the means through which the data can be accessed, altered, or destroyed.

Proprietary Rights

Human resource information is the property of the Profit Center responsible for the data.

The corporate personnel/payroll function is the custodian of the data and will centrally process all maintenance to human resource data.

For all Profit Centers except Central Office. The authority to grant access to the data resides in the personnel function within the appropriate Profit Center. Requests for access to the data must be channeled through the corporation personnel function only with the approval of the appropriate Profit Center personnel representative.

For Central Office. Central Office is the repository of the data and is thus ultimately responsible for its protection. The corporate personnel/payroll function has complete access to data for all Profit Centers without the approval of the Profit Center personnel function, because they are responsible for corporate-wide processing of the data. Only the corporate personnel/payroll function may fully access production information. Each Profit Center may access its own production information.

None of the foregoing shall preclude Internal Audit from having access to the data needed to fulfill their responsibilities as detailed next.

Accountability

Any individual who is involved in unauthorized disclosure of human resource information, procedures, or facilities used to extract information is subject to punitive action or dismissal.

Procedure

Each functional unit named within this policy will maintain comprehensive procedures to support the Human Resource Security Policy.

Responsibilities

The corporation, in its role as an employer of people, has a legal responsibility as well as a moral obligation to strictly limit access to human resource information. Specific responsibilities with regard to human resource security within the corporate organizations are detailed below.

- Human Resource Security Committee
  - To approve any amendments to the Human Resource Security Policy.
  - To review all human resource procedures developed to support the Human Resource Security Policy. It is understood that the scope of this committee relates only to human resource security matters and not to other areas which are the responsibility of the other involved departments.
  - To meet at regular intervals to review all aspects of the Human Resource Security Policy and its associated procedures.
- Personnel
  - To validate and process approved modifications to employee personnel information in a secure manner.
  - To process and distribute reports and other personnel information in a secured manner to appropriate field personnel or other approved recipients.
  - To recommend security policies governing the nature and format of employee records of the Profit Centers.
  - To monitor and audit the performance of the Profit Centers in the administration of approved security policies, plans and practices.
  - To monitor and coordinate the Profit Centers' compliance with employee-related legal requirements and to act as liaison with the corporation's Legal Department.
  - To secure the Personnel area to maintain the confidentiality of all employee information under their control.
  - To approve requested modifications to human resource procedures and facilities which are under their control and to ensure that these modifications comply with human resource security provisions.
- Payroll
  - To process the payroll for all approved corporate organizations in a secure manner.
  - To validate and process approved modifications to employee payroll information in a secure manner.
  - To distribute checks, reports, and other payroll information in a secured manner to appropriate field personnel or other approved recipients.
  - To secure the Payroll area to maintain the confidentiality of all employee information under their control.

- To approve requested modifications to human resource procedures and facilities which are under their control and to ensure that these modifications comply with human resource security provisions.

- Benefit Plans Accounting
  - To process the employee savings plan system for all approved corporate organizations in a secure manner.
  - To validate and process approved modifications to employee savings plan information in a secure manner.
  - To distribute reports and other savings plan information in a secured manner to appropriate field personnel or other approved recipients.
  - To secure the Benefit Plans Accounting area to maintain the confidentiality of all employee information under their control.
- Profit Center Personnel Function
  - To ensure that any request for extraction of human resource information is granted on a “need to know” basis. Access is only granted to data which an individual requires to perform an authorized function. It is understood that no Profit Center may have access to the human resource information of any other Profit Center, unless a reporting relationship exists.
  - To maintain a security policy for the protection of human resource information that is consistent with the Human Resource Security Policy.
- Financial Systems
  - To ensure that any request made to Financial Systems for extraction of human resource information has been made through approved channels.
  - To secure any Financial Systems area allowing access to human resource information or documentation.
  - To approve requested modifications to human resource procedures and facilities which are under their control and to ensure that these modifications comply with human resource security provisions.
- Internal Audit

  Internal Audit has complete access to human resource information consistent with overall audit responsibilities. These responsibilities as they relate to human resource security include:
  - to serve in a review and advisory capacity with respect to human resource security measures to ensure compliance with responsibilities as defined by the policy.
  - To review individual Profit Center security policies for adequacy and adherence.
  - To review requested accesses to human resource information on a periodic basis for adherence to this policy.

  - To perform any audit involving human resource information in a responsible and secure manner. Internal Audit will be accountable for any information gained during the course of an audit.
  - To secure any Internal Audit area allowing access to human resource information or documentation.

- Human Resource Systems
  - To maintain the automated procedures and facilities capable of accessing human resource information which comprise the human resource application in a secure manner.
  - To ensure that access to automated facilities capable of accessing automated human resource information is restricted to members of Data Center Human Resource Systems, approved user personnel, and approved Data Center Operations personnel.
  - To implement only approved modifications to human resource procedures and facilities.
  - To secure the Data Center Human Resource Systems area to restrict access to automated procedures and facilities.
- Data Center-Operations
  - To execute all human resource automated processing in a secure manner by authorized Data Center-Operations personnel only as requested by authorized user personnel.
  - To ensure that the distribution of human resource systems output is made only to authorized personnel.
  - To secure specified areas of Data Center-Operations to maintain confidentiality of human resource information while it is under their control.
- Data Center-Technical Services
  - To ensure that any access to human resource information, procedures or facilities as required by the nature of their responsibilities be done in a secure and responsible manner.
  - To ensure that the security system software is maintained in a secure manner since this software is the basis for protection of automated human resource information, procedures and facilities.

Index

A

B

C

D

E

F

H

I

J

L

M

O

P

R

S

T

U

V

W

Z