CA Top Secret® r14 SP1 for z/OS
Security Target

Version 1.1
March 7, 2011

Prepared for:
CA
2400 Cabot Drive
Lisle, IL 60532

Prepared by:
Booz Allen Hamilton
Common Criteria Testing Laboratory
900 Elkridge Landing Road, Suite 100
Linthicum, MD 21090-2950

Table of Contents

List of Figures

List of Tables

1 Security Target Introduction

This chapter presents the Security Target (ST) identification information and an overview. An ST contains the Information Technology (IT) security requirements of an identified Target of Evaluation (TOE) and specifies the functional and assurance security measures offered by the TOE.

1.1 ST Reference

This section provides information needed to identify and control this ST and its Target of Evaluation. This ST targets Evaluation Assurance Level 4 (EAL4).

1.1.1 ST Identification

ST Title:	CA Top Secret for z/OS r14
ST Version:	1.1
ST Publication Date:	March 7, 2011
ST Author:	Booz Allen Hamilton

1.1.2 Document Organization

Chapter 1 of this ST provides identifying information for the CA Top Secret® for z/OS r14. It includes an ST Introduction, ST Reference, ST Identification, TOE Reference, TOE Overview, and TOE Type.

Chapter 2 describes the TOE Description, which includes the physical and logical boundaries, and describes the components and/or applications that are excluded from the evaluated configuration.

Chapter 3 describes the conformance claims made by this ST.

Chapter 4 describes the Security Problem Definition as it relates to threats, Operational Security Policies, and Assumptions met by the TOE.

Chapter 5 identifies the Security Objectives of the TOE and of the Operational Environment.

Chapter 6 describes the Extended Security Functional Requirements (SFR) and Security Assurance Requirements (SAR).

Chapter 7 describes the Security Functional Requirements.

Chapter 8 describes the Security Assurance Requirements.

Chapter 9 is the TOE Summary Specification (TSS), a description of the functions provided by CA Top Secret® r14 SP1 for z/OS to satisfy the SFRs and SARs.

Chapter 10 is the Security Problem Definition Rationale and provides a rationale or pointers to a rationale, for security objectives, assumptions, threats, requirements, dependencies, and PP claims for the chosen EAL, any deviations from CC Part 2 concerning SFR dependencies, and a mapping of threats to assumptions, objectives, and SFRs. It also identifies the items used to satisfy the Security Assurance Requirements for the evaluation.

1.1.3 Terminology

This section defines the terminology used throughout this ST. The terminology used throughout this ST is defined in Table 1-2: Terminology Definitions. This table is to be used by the reader as a quick reference guide for terminology definitions.

Terminology	Definition
Access	Access indicates an ACID's ability to use a resource.
ACID	An ACID is a unique character-string identifier by which CA Top Secret identifies a user's Security Record.
ACID Authorities	ACID Authorities specify what actions security administrators can perform on ACIDs within their scope.
ACID Type	An ACID type determines an ACID's function in the Security File structure. Types include User, Profile, Department, Division, Zone, DCA, VCA, ZCA, LSCA, SCA, and MSCA.
Administrator	An administrator is a user with privileges to manage the TOE, TOE data, and other TOE users. This includes security administrators (MSCA, SCA type ACIDs) and scoped security administrators (LSCA, DCA, VCA, ZCA type ACIDs). It can also include individual users that have been assigned admin privileges.
Administrative authority	Administrative authority indicates the different classes of authority that are assigned via the TSS ADMIN command function. This field determines the functions a security administrator can perform.
ALL Record	The ALL Record contains global access requirements that are effective for all users.
Attribute	An attribute is a specific authority, privilege, or restriction that is assigned to an ACID.
Auditor	An authorized user or administrator with the audit privilege.
Authorization	Authorization is how CA Top Secret allows access to a protected resource.
Batch	Batch is a method of processing large amounts of data at one time for jobs too large to perform immediately online.
Central Security Control ACID	An SCA is an administrator whose scope of authority includes the entire installation. An SCA can designate and authorize VCAs and DCAs.
Customer Information Control System	CICS is a teleprocessing monitor that can be used for a variety of applications. It is a transaction manager designed for rapid, high-volume online processing.
Certificate Name Filtering	CNF allows administrators the ability to associate certificates with users without having to add each certificate to the CA Top Secret security file.
Database	A database is a systemized collection of data stored for immediate access.
Data set	A data set is a group of logically related records stored together and given a unique name.
Default	Default is a value or action the computer system automatically supplies unless an administrator specifies an alternative.

Department	A department is a mandatory collection of users and profiles that a department ACID defines. A department cannot sign on and does not have a password, but it can own resources.
Department Control ACID	A Department Control ACID is used where security administration has been decentralized. The DCA's scope of authority is limited to the assigned department.
Division	A division is an optional collection of departments that a divisional ACID defines. A division cannot sign on and does not have a password, but it can own resources.
Divisional Control ACID	A VCA is an ACID used where security administration has been decentralized. The VCA's scope of authority is limited to an assigned division, including the departments attached to it.
Entity	An entity is the name of an object as referenced by the system and security.
Facility	A way of grouping options associated with a particular service that users sign on to.
Fail mode	FAIL mode indicates that CA Top Secret is in full control of all access requests. Violations result in termination of the request.
Field Description Table	The FDT contains all dynamically and pre-defined fields identified to CA Top Secret.
Global access	Global access indicates any access specified in the ALL Record.
Integrated Cryptographic Services Facility	ICSF is a component of z/OS and ships with the base product. It is the software component that provides access to the zSeries crypto hardware.
Information Management System	IBM Information Management System (IMS) is a joint hierarchical database and information management system with extensive transaction processing capabilities.
Job Control Language	The computer language that instructs the system how to run a job, assigning files to specific devices and describing each file in detail to the system
Limited Central Security Control ACID	LSCA is a control ACID can have all the authority of an SCA, but unlike the SCA, the LSCA can have a limited scope of control. Only the MSCA can determine that scope and it can encompass ZCAs, VCAs, DCAs, Profiles, Users, and other LSCAs.
Limited Command Facility	LCF is a facility that allows the security administrator to control use of commands/transactions (TSO, CICS, and so on) available to a user.
Locktime	Locktime indicates the period of time after which a terminal automatically locks if no transactions or commands are issued. The user must issue TSS UNLOCK and supply a valid password to unlock the terminal.
Master Security Control ACID	MSCA is the one Control ACID that is predefined, active, and assigned full administrative authority the first time CA Top Secret starts. This administrator's scope of authority includes the entire installation. The MSCA can designate and authorize SCAs, LSCAs, ZCAs, VCAs, and DCAs.
Multi-level security	Multi-Level Security (MLS) is 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.
Node Description Table	The NDT contains all PassTicket, LDAP and CPF information.
Object	Any resource protected by the TOE.
Ownership	Ownership indicates when an ACID has unlimited access to the resource. Ownership defines the resource to CA Top Secret. All other ACIDs must be specifically authorized to access the resource.
Passphrase	A passphrase is a password that can be longer than eight characters and can contain blanks.
PassTicket	A method of authentication the TOE utilizes which is issued for specific session and cannot be used again once that session has ended. In order to

	generate a PassTicket, a user's ACID, time of day, and session are needed.
Permissions	Permissions make an owned resource available to other users in a controlled manner.
Profile	A profile is an ACID containing a collection of access characteristics common to several users. It generally describes the access characteristics of a particular job function. A profile cannot sign on and does not have a password. Up to 254 profiles can be attached to a user's ACID. Any number of users can be associated with a single profile.
RACROUTE	The RACROUTE macro is the interface to RACF (or another external security manager) for z/OS.
Record	CA Top Secret supports several different record types. They include the main Security Record for each ACID, the Audit Record, the ALL Record, the Profile Record, the Department, Division, and Zone Records, the Resource Descriptor Table Record, and the Control ACID's Records.
Recovery File	The Recovery File contains a record of all changes made to the Security File. It is used to recreate the Security File if it becomes damaged or unusable because of hardware or software problems.
Resource	A resource is any component of the computing or operating system required by a task. For the purposes of data protection, these resources are the objects reside on the system.
Resource Access Control Facility	RACF is an IBM program product that provides system entry, resource access control, auditing, accountability, and administrative control for the z/OS operating system.
Resource Descriptor Table	The RDT contains all dynamically and pre-defined resources identified to CA Top Secret.
Scope of authority	Scope of authority indicates what logical units the user has administrative control over.
Secrec	See "Security Record."
Security administrator	A security administrator is primarily responsible for implementation and maintenance functions such as defining users, resources, and levels of access. The administrative authority determines what the security administrator can do.
Security file	A Security File is a Security Database consisting of the Security Records that contain all user and resource permissions and restrictions.
Security label	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.
Security record (secrec)	A Security Record is part of the Security File that contains a set of user and profile records copied into a user's address space, including information such as resources a particular user can access and how the user can use them. This information also contains user characteristics, authorities, and so on. Also known as "secrec."
Security validation algorithm	The Security Validation Algorithm determines whether CA Top Secret should accept or deny users' requests to use a resource such as a data set.
Source of origin	Source or origin indicates the location of an access request (a terminal or reader).
Started Task Command Record	The Started Task Command (STC) Record is a reserved or special ACID that defines a z/OS started task command to CA Top Secret.
SYSID (System Identifier)	SYSID: A maximum of four characters may be specified and the value may contain an asterisk (*) for masking. This keyword is used along with certificate name filtering (CNF).
Time Sharing Option	TSO enables two or more users to execute their programs at the same time

	by dividing the machine resources among terminal users.
User	A user is the lowest ACID level in the security structure. Generally, a user can sign on via a password, initiate jobs, and belong to a department.
User Attribute Data Set	In TSO, UADS is a partitioned data set with a member for each authorized user. Each member contains the appropriate passwords, user definitions, account numbers, LOGON procedure names, and user characteristics that define the user profile.
Violation	A violation is an unauthorized attempt to access a protected resource.
Zone	A zone is an optional collection of divisions defined by a zone ACID. It cannot sign on and does not have a password, but it can own resources.
Zone Control ACID	A ZCA is an administrative ACID whose scope of authority includes an entire zone.

Table 1-1: Customer Specific Terminology

Term	Definition
Authorized user	A user who may, in accordance with the TSP, perform an operation. This is an end user or an administrator.
External IT entity	Any IT product or system, untrusted or trusted, outside of the TOE that interacts with the TOE.
Role	A predefined set of rules establishing the allowed interactions between an end user and the TOE.
TOE Security Functions (TSF)	A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.
User	Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.

Table 1-2: CC Specific Terminology

1.1.4 Acronyms

The acronyms used throughout this ST are defined in Table 1-3: Acronym Definitions. This table is to be used by the reader as a quick reference guide for acronym definitions.

Acronym	Definition
ACID	Accessor ID
APPC	Advanced Program-to-Program Communication
ATF	Audit Tracking File
CC	Common Criteria
CCEVS	Common Criteria Evaluation and Validation Scheme
CCIMB	Common Criteria Interpretations Management Board
CICS	Customer Information Control System
CNF	Certificate Name Filtering
CPF	Command Propagation Facility
DAC	Discretionary Access Control
DCA	Departmental Control ACID
DLF	Data Lookaside Table
EAL	Evaluation Assurance Level
FDT	Field Description Table

ICSF	Integrated Cryptographic Services Facility
IMS	Information Management System
JCL	Job Control List
JES	Job Entry Subsystem
LCF	Limited Command Facility
LDAP	Lightweight Directory Access Protocol
LSCA	Limit Central Security Control ACID
MAC	Mandatory Access Control
MLS	Multi-level Security
MSCA	Master Security Control ACID
MSM	Mainframe Software Manager
MVS	Multiple Virtual Storage
NDT	Node Description Table
PPT	Program Properties Table
RACF	Resource Access Control Facility
RDT	Resource Descriptor Table
SAF	System Authorization Facility
SCA	Central Security Control ACID
SDT	Static Data Table
SMF	System Management Facility
ST	Security Target
STC	Started Task Command
SYSID	System Identifier
TMP	Terminal Monitor Program
TOE	Target of Evaluation
TSC	TOE Scope of Control
TSF	TOE Security Function
TSO	Time-sharing option
UADS	User Attribute Data Set
VCA	Divisional Control ACID
VSAM	Virtual Storage Access Method
ZCA	Zonal Control ACID

Table 1-3: Acronym Definitions

1.1.5 References

[1] Common Criteria for Information Technology Security Evaluation, CCMB-2007-09-004, Version 3.1 Revision 3, July 2009.

[2] CA Top Secret for z/OS User Guide r14

[3] CA Top Secret for z/OS Overview Guide r14

[4] CA Top Secret for z/OS Multilevel Security Planning Guide r14

[5] CA Top Secret for z/OS Auditor Guide r14

[6] CA Top Secret for z/OS Control Options Guide r14

[7] CA Top Secret for z/OS Cookbook r14

[8] z/OS Planning for Installation Version 1 Release 11

1.1.6 CC Concepts

The following are CC concepts as used in this document. A Subject is any user of the TOE (account, user, administrative user, applications or process issuing RACROUTE call). An Object (i.e., resource or entity) can be a dataset, volume, command issued by a user, etc. An Operation is any action on a resource (e.g. read, write, create, fetch, update, control, alter, or scratch). A Security Attribute is information such as username, groups, profiles, facilities, passwords, etc. that is kept in the security file for the user. An External Entity is anything outside of the TOE that affects the TOE.

1.2 TOE Reference

1.2.1 TOE Identification

CA Top Secret® r14 SP1 for z/OS

1.3 TOE Overview

CA Top Secret® r14 SP1 for z/OS, herein referred to as CA Top Secret, delivers access control software for z/OS operating systems and includes interfaces for TSO, CICS, and IMS.

The TOE allows administrators to control user access to protected resources such as datasets and volumes and their associated data. User access is controlled via Top Secret intercepts of commands issued by the users or applications acting on behalf of the users running in z/OS.

The TOE:

- Provides a platform for access control to protected data resources
- Integrates with and operates as a policy enforcement mechanism of the z/OS MVS Operating System. CA Top Secret is the called by z/OS for all security access control checks made by the operating system. No action is allowed to be performed by users and administrators of z/OS without access to CA Top Secret for an authorization decision
- Records all policy enforcement decisions made by z/OS and stores the decisions in the z/OS database
- Maintains a database of all users and their associated security attributes
- Protects critical data sets and resources so that only the appropriate people have access to them
- Reports on any unauthorized access attempts

- Provides protection against unauthorized destruction, disclosure, or modification of data and resources

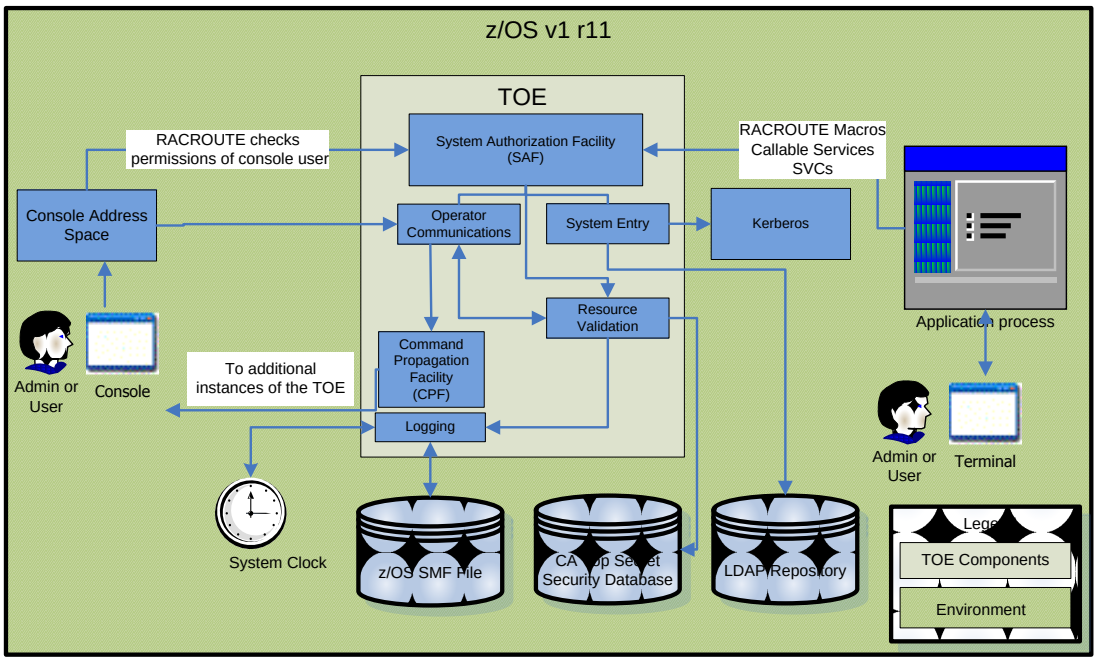

Figure 1-1: TOE Boundary

As illustrated in Figure 1-1, there are two types of users of the TOE: administrators and users. A user must be granted an administrative privilege in order to be considered an administrator. Once given an administrative privilege, an administrator can then manage the TOE based on the permissions assigned to them. Security administrators manage the entire TOE and its users, whereas Scoped Security Administrators can only perform operations defined within their scope. A non-administrator user accesses the TOE strictly to perform work. For more information on administrators of the TOE, see Section 9.1.4.1.

Users and Security Administrators can access the TOE through the Console Address Space and/or Application process. The Console Address Space is a z/OS facility that provides a callable interface that lets MVS applications communicate with other applications using Advanced Program-to-Program Communication. When attempting to communicate through the console to Console Address Space locally, RACROUTE is called to check the permissions of the user or administrator to ensure that they are allowed to access the TOE or perform the action. If the user or administrator were attempting to access the TOE remotely, they would communicate through the terminal to

the Application process, which is used for callable services, RACROUTE macros, and RAC SVCs.

The components within the TOE include System Authorization Facility (SAF), Command Propagation Facility (CPF), operator communications, and LDAP Directory Services (LDS). The SAF component is used by the TOE to get control of SAF calls *before* the z/OS router so that it can determine how to process the calls. The CPF facility routes security administration to all or selected nodes, resulting in single-point administration. Changes made to ACIDs, passwords, or access levels can be propagated to all nodes to which the user is defined. CA Top Secret operator communications calls are performed with operator communications.

The Operational Environment (OE) consists of the following components: z/OS, Console Address Space, Application Process (i.e. third party applications), Kerberos, z/OS SMF File, CA Top Secret Security File, console, terminal, and system clock. Kerberos verifies requests as a trusted third-party authentication service. CA Top Secret keeps this information in the Security File. The records are administered through the CA Top Secret TSS command. There are two types of file storage records that contain the information that can be requested through SAF: the User ACID record and the Profile ACID record.

Applications running on top of the z/OS operating system also make calls to the TOE. All calls to the TOE are used to restrict all unauthorized access to TOE objects.

CA Top Secret uses the following files:

- **Security File –** A security database consisting of the security records that contain all user and resource permissions and restrictions. When a user initiates a job or signs on to an online facility in a z/OS environment, CA Top Secret obtains the user's security record from the security file, and places it in the user's address space for the duration of the session
- **Parameter File -** Stores and defines control options at initialization and sets up the operating environment for CA Top Secret
- **Audit/Tracking File -** Records security-related events and can be shared among CPUs. These events include violations, job and session initiation, and resource access
- **Backup File -** Stores the automatic daily backup of the security file to ensure complete integrity of the security environment. The backup file is an exact copy of the security file as it existed at the time of the last backup, and it can be used if the device containing the security file becomes unavailable
- **Recovery File -** Stores recent administrative commands, depending on the size of the file allocated. The backup security file with the application of select recovery file commands can completely restore a damaged security file. This is a wraparound file

- **$$LOG$$ File -** When CA Top Secret starts under the Job Entry Subsystem (JES), it dynamically allocates a SYSOUT file with a DDNAME of $$LOG$$. This file contains:
    - All modifications to the CA Top Secret default control options originating from:
        - CA Top Secret Parameter File
        - Console operator command modification of CA Top Secret task
        - Online administrator TSS command modification
    - Status messages for the CA Top Secret task regarding:
        - Security File capacity and backup
        - Audit/Tracking File capacity and backup
        - I/O errors in CA Top Secret files during task execution
- **Command Propagation Files -** Two types of files are associated with the command propagation facility (CPF). These files must be dedicated to a single CPU.
- **CPF Recovery File -** A disk file the CPF uses to save transmitted commands until a response to those commands has been received from remote nodes.
- **CPF Journal Files -** These files provide an historical record of the command traffic to and from a particular CA Top Secret CPF node. One receives journal records commands and their response from other nodes. A send journal can exist for each connected node to record commands and responses.

The following security classes are enforced by the TOE: Security Audit, Identification & Authentication, User Data Protection, Security Management, and TOE Access. For an explanation of each of these security classes, see section 2.5 Logical Boundary.

The TOE also ensures that information critical for the security of subjects receives protection via the operational environment's cryptographic mechanisms. The TOE will call the z/OS ICSF module when any security relevant information requires encryption. The ICSF component is not part of the TOE.

1.4 TOE Type

The TOE type for CA Top Secret for z/OS is System Access Control. System Access Control is defined by CCEVS as "A technique used to define or restrict the rights of individuals or application programs to obtain data from, or place data onto, a storage device. The definition or restriction of the rights of individuals or application programs to obtain data from, or place data into, a storage device. Limiting access to information system resources only to authorized users, programs, processes, or other systems."

2 TOE Description

2.1 Evaluated Components of the TOE

The following table describes the TOE components in the evaluated configuration:

Component	Definition
System Authorization Facility (SAF)	The IBM System Authorization Facility (SAF) provides a system wide interface to CA Top Secret. The CA SAF Router (A component of CA Top Secret) intercepts and passes all RACROUTE calls to CA Top Secret. CA Top Secret processes all SAF calls by default. This enables CA Top Secret to manage all of the unique processing needed to provide full security coverage for the z/OS platform.
CA Top Secret Operator Communications	CA Top Secret has three primary operator commands. They are START, STOP, and MODIFY. These commands have multiple sub-functions that are controlled by the CA Top Secret and allow for configuration setup, configuration changes and shut down options. Authorization is required to perform that function. In addition there are a number of commands that establish and configure CA SAF tracing. These are also controlled by CA Top Secret and require specific authorization. The CA SAF Sectrace commands are as follows: SECTRACE DELETE, SECTRACE DISABLE, SECTRACE DISPLAY, SECTRACE LOGERR, SECTRACE MODIFY, SECTRACE NOLOGERR, and SECTRACE SET.
Command Propagation Facility (CPF)	Routes security administration to all or selected nodes synchronously or asynchronously, resulting in single-point administration. Changes made to ACIDs, passwords, or access levels can be propagated to all nodes to which the user is defined. For example, USER01 is defined to two nodes, with NODE A as the local node and NODE B as the remote node. If the user changes the password on NODE A, CPF automatically propagates the change to NODE B. Through the use of command function keywords, a user can specify which node receives these commands and how the local node processes them. The Command Propagation Facility (CPF) component of CA Top Secret uses the CAI Common Communications Interface (*CAICCI)* as the communications layer from one system to the other. CA Top Secret puts a request packet together and sends it though CCI. CCI sits on both systems to perform 'receive' and 'send' functions.

Common Services	Common Services is a set of common components used by a number of CA's Mainframe products. It is free but requires a separate install. Top Secret uses three components in the CA Common Services product line. The CICS interface for CA Top Secret uses the *CAIENF/CICS* subcomponent of CAIENF to gain control of security calls implanted within CICS by *CAIENF/CICS.* This component then calls the TSS CICS interface to process the security call. The CA Event Notification Facility (CAIENF) is an operating system interface service that enables CA applications to obtain event data from z/OS and subsystems such as CICS and TSO. The CAI Common Communications Interface (CAICCI) is a communications facility that offers a simple and flexible approach enabling CA products to communicate with one another. This facility provides a layer that isolates application software from the specifics of the communications environment. CAICCI features include: • Single point of control • Multiple platforms support • Performance optimization • Peer-to-peer (program to program) communication • Parallel conversations • Dynamic installation configuration • Ease of customization • Error handling
LDAP Directory Services (LDS)	An LDAP directory provides a method to maintain directory information, such as email accounts, in a central location, for storage, update, retrieval, and exchange. LDAP directories can be utilized as network accessible databases for organization and indexing of network security information. As LDAP is becoming an integral part of most networks, CA Top Secret provides the LDAP Directory Services (LDS) option. LDS is the functionality that allows CA Top Secret to interface with and transmit data to the remote repository. Unlike other products and services that

	use pull technology on a scheduled basis; this is a proactive approach that pushes the change as it occurs, providing a real time update of the changed data.

Table 2-1: Evaluated Components of the TOE

2.2 Components and Applications in the Operational Environment

The following table lists components and applications in the environment that the TOE relies upon in order to function properly:

Component	Definition
Kerberos	Network Authentication and Privacy Service, known as Kerberos, uses CA Top Secret to store and administer information about principals and realms. REALM and KERBLINK records have been incorporated into CA Top Secret to store this information. The REALM record is used to store information about Network Authentication and Privacy Service principals on the local system. The KERBLINK record lets a user map principals to CA Top Secret users on a system Kerberos for z/OS verifies requests as a trusted third-party authentication service. Using conventional shared secret key cryptography, Kerberos confirms the identities of principals (users), without relying on authentication by the host operating system, without basing trust on host addresses, without necessitating physical security of all hosts on the network, and under the premise that packets traveling along the network can be read, changed, and inserted at will.
Console Address Space	Console Address Space is a z/OS facility that provides a callable interface that lets MVS applications communicate with other applications using Advanced Program-to-Program Communication (APPC). These other applications can reside in or outside of the mainframe. The MVS console can change global configuration parameters of the TOE via a modify MVS command.
z/OS v1 r11	A 64-bit operating system for mainframe computers, created by IBM, under which the TOE operates. The z/OS operating system also provides its encryption and decryption methods which the TOE employs.

Console	The local interface used by users of the TOE to access CA Top Secret through the Console Address Space for management and audit functionality
Terminal	The remote interface used by users of the TOE to access CA Top Secret directly or via the Application Process
Application Process	This includes all third party application processes which TOE users and administrators perform actions on which are monitored by the TOE for authentication, authorization, and in some instances management of objects through the TOE.

Table 2-2: Evaluated Components of the Operational Environment

2.3 Excluded from the TOE

The following optional products, components, and/or applications can be integrated with CA Top Secret but are not included in the evaluated configuration. They provide no added security related functionality. They are separated into three categories: not installed, installed but requires a separate license, and installed but not part of the TSF.

2.3.1 Not Installed

These components are not installed with CA Top Secret and are therefore not included in the TOE boundary.

- **ELM Integration -** Enterprise Log Manager is a separate product that collects and stores logs from various endpoints using agents configured with connectors.
- **CA Compliance Manager for z/OS Integration** – CA Compliance Manager for z/OS allows Administrators to collect, and report on security relevant activity, and generate alerts requiring action when possible compliance violations occur. It is an independent product that requires its own license and is not included in the evaluated configuration of the TOE.
- **CA Top Secret® Option for DB2 –** Protects several DB2 resources and replaces GRANT/REVOKE processing. PERMIT commands are written in place of GRANT commands and a conversion utility provides a transition. A catalog synchronization utility brings DB2 catalog entries up-to-date with CA Top Secret® Option for DB2 authorizations.
- **DFSMS** – IBM Subsystem. With DFSMS, the z/OS administrator can define performance goals and data availability requirements, create model data definitions for typical data sets, and automate data backup. DFSMS can

automatically assign, based on installation policy, those services and data definition attributes to data sets when they are created.

- **Event Notification Facility (ENF)** - An operating system interface component CA Top Secret uses to obtain data from z/OS. CAIENF provides the VTAM facilities to transmit and receive CA Top Secret TSS commands when using the Command Propagation Facility.
- **Standard Security Facility (SSF)** - A facility that provides an application interface for CA and non-CA products to obtain and use CA Top Secret information.

2.3.2 Installed but Requires a Separate License

There are no components that are installed with CA Top Secret that require a separate license.

2.3.3 Installed But Not Part of the TSF

These components are installed with CA Top Secret, but are not included in the TSF.

- **Group** – Group is not commonly used. The intent is for its use is for backward compatibility. It is not used for object access.
- **User Attribute Data Set (UADS)** – In TSO, UADS is a partitioned data set with a member for each authorized user. Each member contains the appropriate passwords, user definitions, account numbers, LOGON procedure names, and user characteristics that define the user profile. This is an obsolete capability.
- **SYSPLEX** – The coupling facility is a feature of MVS/ESA that allows systems in a sysplex environment to communicate and share data with each other. It allows multiple systems to share one security file. Security in a sysplex environment is based on:
    - The communication function or Cross System Coupling Facility (XCF) that provides a way for each system in the sysplex to send messages or signals to all other systems.
    - The data sharing function or Cross System Extended Services (XES) that provides the ability for systems in the sysplex to share common data that would normally be obtained from a database. This function saves system resources by reducing I/O to the database.

- **Security Modes –** The following security modes are not security enforcing and are therefore not included in the evaluated configuration:
  - **Dormant Mode** - Although CA Top Secret is installed, it is not actively validating access. Checks are made for Administrators, but not for users.

  - **Warn Mode – Warn mode is used to:**
    - Determine which users are accessing which resources
    - Test the access definitions made in DORMANT mode

Warn mode can be set by facility, profile, user, resource, or event.

Note: Some applications make RACROUTE calls with the LOG=NOFAIL parameter. In WARN mode, these types of calls are written to the audit file if the check fails, but no message displays.

- **Signon Violations** - In WARN mode, define all users to CA Top Secret or CA Top Secret generates and records signon violations. WARN mode does not prevent an undefined user from signing on or gaining access to a protected resource.
- **Password Violations** - WARN mode does not prevent a defined user from signing on with an incorrect password, but this action generates a password violation.

Note: To force a defined user to supply a correct password in WARN mode, the WARNPW sub-option of the FACILITY control option must be set. Administrators must always supply a correct password, even in DORMANT mode.

- **Resource Violations -** If default protection is given to specific resource classes by attaching the DEFPROT attribute, WARN mode generates violations for all defined resources.
- **Global Warn Mode -** Use Global WARN mode to test segments of the implementation, or to back off from FAIL mode when an implemented segment of the organization is in trouble.

- **CA Mainframe Software Manager (MSM)** – The CA MSM is a utility used by the TOE that allows for the initial acquisition of CA Top Secret. This utility is part of the operational environment and provides no security enforcing functionality for the TOE once it has been acquired.

2.4 Physical Boundary

The TOE includes the CA Top Secret components:

- Operator Communications
- System Authorization Facility (SAF)
- Command Propagation Facility (CPF)

The following table illustrates the minimum requirements needed to install CA Top Secret on a z/OS system.

Requirement	Description
Operating System	z/OS V1R9 or later OR The Customized Offerings Driver V3
A TSO/E Session	A TSO/E Session on the IPLed system must be established using a locally-attached or network-attached terminal
Proper Authority	Use the RACFDRV installation job as a sample of the security system definitions required so that a user can perform the installation tasks
Proper Security	In order to install the z/OS UNIX files, the following is required: • The ACID must be a superuser (UID=0) or have read access to the BPX.SUPERUSER resource in the RACF FACILITY class. • The ACID must have read access to FACILITY class resources BPX.FILEATTR.APF, BPX.FILEATTR.PROGCTL, and BPX.FILEATTR.SHARELIB (or BPX.FILEATTR.* if a user chooses to use a generic name for these resources). The commands to define these FACILITY class resources are in SYS1.SAMPLIB member BPXISEC1.
OMVS Address Space Active	For ServerPac only (not SystemPac), an activated OMVS address space with z/OS UNIX kernel services operating in full function mode is required.
SMS Active	The Storage Management Subsystem (SMS) must be active to allocate z/OS UNIX file systems (HFS or zFS) and PDSE data sets, whether they are SMS-managed or non-SMS-managed. In addition, the use of z/OS UNIX file systems (HFS or zFS) is supported only when SMS is active in at least a null configuration, even when the file systems are not SMS-managed. Do either of the following: • To allocate non-SMS-managed z/OS UNIX file systems (HFS or zFS) and PDSE data sets, a user must activate SMS on the driving system in at least a null configuration. A user must also activate SMS on the target system. • To allocate SMS-managed z/OS UNIX file systems (HFS or zFS) and PDSE data sets, a user must activate SMS on the driving system in at least a minimal configuration. Then a user must define a storage group, create SMS-managed volumes, and write, translate, and activate a storage class ACS routine that allows the allocation of z/OS UNIX file systems (HFS or zFS) and PDSE data sets with the names in the ALLOCDS job. A user must also activate SMS on the target system.
DFSORT	msys for Setup job XMLGNR8 requires DFSORT or an equivalent sort program on the system on which the XMLGNR8 job is run.
Language Environment Requirements	The CustomPac Installation Dialog uses the Language Environment run-time library, SCEERUN. If SCEERUN is not in the link list on the driving system, a user must edit the ServerPac installation jobs to add it to the JOBLIB or STEPLIB DD statements.

CustomPac Installation Dialog	If installing a ServerPac or dump-by-data-set SystemPac for the first time, a user will need to install the CustomPac Installation Dialog on the driving system. See *ServerPac: Using the Installation Dialog* or *SystemPac: CustomPac Dialog Reference* for instructions. For subsequent orders, a user will not need to reinstall the dialog. IBM ships dialog updates with each order. A user should check the PSP bucket for possible updates to the CustomPac Installation Dialog. For ServerPac, the upgrade is ZOSV1R11 and the subset is SERVERPAC. For SystemPac dump-by-data-set orders, the upgrade is CUSTOMPAC and the subset is SYSPAC/DBD.
Proper Level for Service	In order for a user to install service on the target system that are building, a user's driving system must minimally meet the driving system requirements for CBPDO Wave 1 and must have the current (latest) levels of the program management binder, SMP/E, and HLASM.
SMP/E ++JAR Support	If the ServerPac order contains any product that uses the ++JAR support introduced in SMP/E V3R2 (which is the SMP/E in z/OS V1R5), the driving system requires IBM SDK for z/OS, Java 2 Technology Edition, V1 (5655-I56) at SDK 1.4 or later. z/OS itself does not use the ++JAR support.
zFS Configured Properly	If using a zFS for installation, then a user must be sure that the zFS has been installed and configured, as described in *z/OS Distributed File Service zSeries File System Administration*.
Internet Delivery Requirements	If intending to receive the ServerPac or SystemPac dump-by-data-set order by way of the Internet, a user will need the following: • SMP/E PTF UO00678 (APAR IO07810) if SMP/E level is V3R4 (which is in z/OS V1R7, V1R8, and V1R9). v ICSF configured and active so that it can calculate SHA-1 hash values in order to verify the integrity of data being transmitted. If ICSF is not configured and active, SMP/E calculates the SHA-1 hash values using an SMP/E Java application class, provided that IBM SDK for z/OS, Java 2 Technology Edition, V1 (5655-I56) or later is installed. IBM recommends the ICSF method because it is likely to perform better than the SMP/E method. (To find out how to configure and activate ICSF, see *z/OS Cryptographic Services ICSF System Programmer's Guide*. For the required SMP/E setup, see *SMP/E User's Guide*.) • A download file system. The order is provided in a compressed format and is saved in a download file system. The size of this file system should be approximately twice the compressed size of the order to accommodate the order and workspace to process it. Firewall configuration. If the enterprise requires specific commands to allow the download of the order using FTP through a local firewall, a user must identify these commands for later use in the CustomPac Installation Dialog, which manages the download of the order. • Proper dialog level. If a user is using a CustomPac Installation Dialog whose Package Version is less than 17.00.00, he/she must migrate the dialog to this level or later. The user can

	determine if he/she has the correct dialog level by looking for the text "This dialog supports electronic delivery." at the bottom of panel CPPPPOLI. If the dialog is not at the minimum level, follow the migration scenarios and steps described in *ServerPac: Using the Installation Dialog*.
Additional Internet Delivery Requirements for Intermediate Download	If planning to download the ServerPac or SystemPac dump-by-data-set order to a workstation and from there to z/OS, a user will need the following in addition to the requirements listed in item 13 on page 56: • Download Director. This is a Java applet used to transfer IBM software to workstation. • The ServerPac or SystemPac dump-by-data-set order accessible to the host. To make the order (files) accessible to z/OS, can do either of the following: o Configure the workstation as an FTP server. After downloading the order to the workstation, the dialogs used to install a ServerPac or SystemPac dump-by-data-set order can point to a network location (in this case, workstation) to access the order. Consult the documentation for the workstation operating system to determine if this FTP capability is provided or if it has to install additional software. Commercial, shareware, and freeware applications are available to provide this support. However, IBM cannot directly recommend or endorse any specific application. This option requires the use of ICSF. o Use network drives that are mounted to z/OS. The mounting can be accomplished using the NFS base element, server message block (SMB) support provided by the Distributed File Service base element, or the Distributed FileManager component of the DFSMSdfp base element. The package is received from the file system defined as the SMPNTS. o CD write capability. If specified that 100% electronic delivery is required, there might be CD images associated with the order. The images are delivered in ISO9660 format and are packaged in zip files (with an extension of .zip). These files require the workstation to have CD write capability and might have to acquire software to support this capability.

Table 2-3: Minimum OS Requirements for Installation of the TOE

For more information on the hardware requirements for z/OS, see the z/OS v1 r11 Planning for Installation Guide.

2.5 Logical Boundary

The logical boundary of the TOE is described in the terms of the security functionalities that the TOE provides to the systems that utilize this product for information flow control.

The logical boundary of the TOE will be broken down into five security classes: Security Audit, Identification and Authentication, Security Management, User Data Protection, and TOE Access. Listed below are the security functions with a listing of the capabilities associated with them:

2.5.1 Security Audit

CA Top Secret uses the System Management Facility (SMF) to record all security-relevant events. These records are secured from accidental disclosure or destruction by the standard Discretionary Access Control (DAC) and Mandatory Access Control (MAC) protection mechanisms. The TOE enforces the Mandatory Access Control (MAC) policy to objects based on users, resources, and AccessLevel, Type, Object Security Label, and Subject Security label. The TOE enforces the Discretionary Access Control (DAC) policy to objects based on users, entity, security relevant attributes control option auth, resource class name, entity name, secrec, ownership, facility, time of day, day of week, sysid, Limited Command Facility (LCF), calendar, program, and library.

CA Top Secret provides report utilities to produce reports. For example, the TSSUTIL utility report provides an audit trail of security events. A variety of parameters can be set to customize the reports.

2.5.2 Identification & Authentication

CA Top Secret controls how, when, and which resources a user can access. CA Top Secret requires that each end user have a valid accessor ID (ACID) and password before entering the system. An ACID can be up to eight alphanumeric characters long, which normally corresponds with the user's system userid. The same ACID can be used for all facilities or a different ACID can be used for each facility (such as TSO, CICS, and z/VM).

By default, CA Top Secret requires that all ACIDs are password protected. A security administrator assigns the first password. The user associated with the ACID changes the password immediately (or later if they desire) when it expires. Password assignment is controlled by CA Top Secret; control option values are set and stored within CA Top Secret.

2.5.3 Security Management

The TOE maintains three roles: security administrators, scoped security administrators and users. Administrators manage the TOE and its users; whereas a user's primary function is to perform work. Any administrator with ACID (CREATE) administrative authority can establish users. While a user can be assigned most types of administrative authority, the user's scope is always limited to itself.

Security administrators can display and change fields of ACIDs based on their scope. Scoped administrators can perform administrative operations that are defined within their scope.

2.5.4 User Data Protection

CA Top Secret determines whether an individual user should be permitted access to a resource and must be able to associate a user's identity with each job or time-sharing session. No job can run on a CA Top Secret-controlled system unless it can first be identified with a valid, predefined user. Thus, CA Top Secret is also protecting the resources of the computer system itself. No one can use processing time on a system unless they are running under an ACID previously defined to CA Top Secret.

CA Top Secret performs two main methods of access control, one being mandatory access control (MAC) and the other being discretionary access control (DAC).

MAC imposes a security policy based on security labels. Security labels classify users, data, and resources. Standard 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.

DAC security policy manages the controlled sharing of data and resources using permissions. 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.

2.5.5 TOE Access

CA Top Secret is capable of denying access to TOE users who have a suspended/canceled account or have failed to enter a correct password within the threshold limit set by an administrator.

For more information on TOE Access, see Section 9.1.5.

3 Conformance Claims

3.1 CC Version

This ST is compliant with *Common Criteria for Information Technology Security Evaluation*, CCMB-2007-09-004, Version 3.1 Revision 3, July 2009.

3.2 CC Part 2 Conformant

This ST and Target of Evaluation (TOE) is Part 2 conformant for EAL4 to include all applicable NIAP and International interpretations through 1 April 2011.

3.3 CC Part 3 Conformant Plus Flaw Remediation

This ST and Target of Evaluation (TOE) is Part 3 conformant plus flaw remediation for EAL4 to include all applicable NIAP and International interpretations through 1 April 2011.

3.4 PP Claims

This ST does not claim Protection Profile (PP) conformance.

3.5 Package Claims

This TOE claims a package for EAL 4.

3.6 Package Name Conformant or Package Name Augmented

This ST and Target of Evaluation (TOE) is conformant to EAL package claims augmented with ALC_FLR.1 and ASE_TSS.2**.**

3.7 Conformance Claim Rationale

There is no Conformance Claim rationale for this ST.

4 Security Problem Definition

4.1 Threats

The TOE itself has threats and the TOE is also responsible for addressing threats to the environment in which it resides. The assumed level of expertise of the attacker for all the threats is unsophisticated. The following are threats addressed by the TOE.

T.ACCESS	Unauthorized users or administrators could gain access to objects protected by the TOE that they are not authorized to access.
T.ADMIN_ERROR	An administrator may incorrectly install or configure the TOE, or install a corrupted TOE resulting in ineffective security mechanisms.
T.AUDIT_COMPROMISE	A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded; thus masking a user's action.
T.MASK	Users whether they be malicious or non-malicious, could gain unauthorized access to the TOE by bypassing identification and authentication countermeasures.

4.2 Organizational Security Policies

There are no Organizational Security Policies that apply to the TOE.

4.3 Secure Usage Assumptions

The specific conditions listed in this section are assumed to exist in the environment in which the TOE is deployed. These assumptions are necessary as a result of practical realities in the development of the TOE security requirements and the essential environmental conditions on the use of the TOE.

4.3.1 Personnel Assumptions

A.ADMIN	One or more authorized administrators will be assigned to install, configure and manage the TOE and the security of the information it contains.
A.PATCHES	Administrators exercise due diligence to update the TOE with the latest patches and patch the Operational Environment (e.g., OS and database) so they are not susceptible to network attacks.

A.NOEVIL Administrators of the TOE are not careless, wilfully negligent, or hostile and will follow and abide by the instructions provided by the organization's guidance documentation.

4.3.2 Connectivity Assumptions

There are no connectivity assumptions that apply to the TOE.

4.3.3 Physical Assumptions

A.LOCATE The TOE will be located within controlled access facilities that will prevent unauthorized physical access.

5 Security Objectives

The following security objectives are to be satisfied by the TOE.

5.1 Security Objectives for the TOE

The following security objectives are to be satisfied by the TOE.

O.ACCESS The TOE will provide measures to authorize users and administrators to access objects protected by the TOE once they have been authenticated. User and administrator authorization is based on access rights configured by the administrators of the TOE.

O.AUDIT The TOE will provide measures for recording security relevant events that will assist the authorized users and administrators in detecting misuse of the TOE and/or its security features that would compromise the integrity of the TOE and violate the security objectives of the TOE.

O.AUTH The TOE will provide measures to uniquely identify all users and administrators. The TOE will authenticate the claimed identity prior to granting a user or administrator access to the objects protected by the.

O.MANAGE The TOE will provide authorized administrators with the resources to manage and monitor user accounts, objects, and security information relative to the TOE.

O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management.

O.ROBUST_TOE_ACCESS The TOE will provide mechanisms that control a user's and an administrator's logical access to the TOE and to explicitly deny access to specific users and administrators when appropriate.

5.2 Security Objectives for the operational environment of the TOE

The following security objectives for the Operational environment of the TOE must be satisfied in order for the TOE to fulfill its security objectives.

OE.ADMIN One or more authorized administrators will be assigned to configure the Operational Environment, and install,

	configure and manage the TOE and the security of the information it contains.
OE.EAVESDROPPING	The Operational Environment will encrypt TSF data when called by the TOE to prevent malicious users from gaining unauthorized access to TOE data.
OE.NOEVIL	All administrators of the TOE are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided by the organization's user guidance documentation.
OE.LOCATE	The TOE will be located within controlled access facilities that will prevent unauthorized physical access.
OE.SYSTIME	The operating environment will provide reliable system time.

6 Extended Security Functional and Assurance Requirements

6.1 Extended Security Functional Requirements for the TOE

There are no extended Security Functional Requirements for this ST.

6.2 Extended Security Assurance Requirements

There are no extended Security Assurance Requirements in this ST.

7 Security Functional Requirements

7.1 Security Functional Requirements for the TOE

The following table provides a summary of the Security Functional Requirements implemented by the TOE.

Security Function	Security Functional Components
Security Audit (FAU)	FAU_GEN.1 Audit data generation
	FAU_GEN.2 User identity association
	FAU_SAR.1 Audit review
	FAU_SAR.2 Restricted audit review
	FAU_SEL.1 Selective audit
User Data Protection (FDP)	FDP_ACC.2(1) Complete access control
	FDP_ACC.2(2) Complete access control
	FDP_ACF.1(1) Security attribute based access control
	FDP_ACF.1(2) Security attribute based access control
Identification and Authentication (FIA)	FIA_AFL.1 Authentication failure handling
	FIA_ATD.1 User attribute definition
	FIA_SOS.1(1) Verification of secrets
	FIA_SOS.1(2) Verification of secrets
	FIA_SOS.2 TSF generation of secrets
	FIA_UAU.2 User authentication before any action
	FIA_UAU.4 Single-use authentication mechanisms
	FIA_UAU.5 Multiple authentication mechanisms
	FIA_UID.2 User identification before any action
	FIA_USB.1 User-subject binding
Security Management (FMT)	FMT_MOF.1(1) Management of security functions behavior
	FMT_MOF.1(2) Management of security functions behavior
	FMT_MOF.1(3) Management of security functions behavior
	FMT_MOF.1(4) Management of security functions behavior
	FMT_MSA.1 Management of security attributes
	FMT_MSA.3 Static attribute initialization
	FMT_SMF.1 Specification of management functions
	FMT_SMR.1 Security roles

Security Function	Security Functional Components
TOE Access (FTA)	FTA_TSE.1 TOE session establishment

Table 7-1: Security Functional Requirements for the TOE

7.1.1 Class FAU: Security Audit

7.1.1.1 FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;

b) All auditable events for the [**not specified**] level of audit; and

c) [***all auditable events between subjects and resources***].

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [***CPUID, Jobname, and Source***].

Dependencies: FPT_STM.1 Reliable time stamps

Application Note: *The mode will always be FAIL in the evaluated configuration*

Application Note: *All audit records are SMF type 80 records.*

Application Note: *Auditable events are all events that a user (or an application on behalf of the user) generates, as well as events from the system that don't have the parameter log=nofail. Only applications or systems can specify this parameter.*

7.1.1.2 FAU_GEN.2 User identity association

Hierarchical to: No other components.

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

Dependencies: FAU_GEN.1 Audit data generation
FIA_ACID.1 Timing of identification

7.1.1.3 FAU_SAR.1 Audit review

Hierarchical to: No other components.

FAU_SAR.1.1 The TSF shall provide [***authorized users with the acid(report) and resource(report) privileges***] with the capability to read [***all audit information collected by FAU_GEN.1 within their scope***] from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

Dependencies: FAU_GEN.1 Audit data generation

Application Note: *All audit records are SMF type 80 records.*

7.1.1.4 FAU_SAR.2 Restricted audit review

Hierarchical to: No other components.

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

Dependencies: FAU_SAR.1 Audit review

7.1.1.5 FAU_SEL.1 Selective audit

Hierarchical to: No other components.

FAU_SEL.1.1 The TSF shall be able to select the set of events to be audited from the set of all auditable events based on the following attributes:
a) [**Object identity, User identity**]

b) *[**Permission***]

Dependencies: FAU_GEN.1 Audit data generation

Application Note: *Object identity is resource class and entity name and is captured by RACROUTE REQUEST=AUTH and FASTAUTH.*

Application Note: *Auditing of violations on the TOE is automatic and cannot be turned off.*

7.1.2 Class FDP: User Data Protection

7.1.2.1 FDP_ACC.2(1) Complete access control

Hierarchical to: FDP_ACC.1 Subset access control

FDP_ACC.2.1 (1) The TSF shall enforce the [***Discretionary Access Control Policy***] on [***users and objects***] and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2 (1) The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

Dependencies: FDP_ACF.1 Security attribute based access control

Application Note: *A user can refer to a user, an application issuing a RACROUTE call, a terminal command, or a console command. An object can be a data set, volume, command issued, etc. An example of an Access Level is an operation. For the list of all operations among subjects and objects, reference Table 7-2.*

7.1.2.2 FDP_ACC.2 (2) Complete access control

Hierarchical to: FDP_ACC.1 Subset access control

FDP_ACC.2.1 (2) The TSF shall enforce the [***Mandatory Access Control Policy***] on [***users and objects***] and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2 (2) The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

Dependencies: FDP_ACF.1 Security attribute based access control

Application Note: The following table lists all operations allowed by TOE users:

AccessLevel (DAC)	DIRAUTHLevel (MAC)
Read	Read
Create	Read
Write	Write
Control	ReadWrite
Update	ReadWrite
Scratch	ReadWrite
Fetch	ReadWrite
Alter	ReadWrite

Table 7-2: User Performed Operations on the TOE

Application Note: The following list of classes is included in the evaluated configuration:

Interface	Resource Classes	
Base CA Top Secret products	DATASET	PROGRAM
	OPERCMDS	SECLABEL
	TSOAUTH	UNIXPRIV
	FACILITY	IBMFAC
CA Top Secret CICS Interface	OTRAN	TCICSTRN
	FCT	PPT
CA Top Secret IMS Interface		TIMS
	CIMS	IIMS

Table 7-3: Resource Classes Included in the Evaluated Configuration

7.1.2.3 FDP_ACF.1 (1) Security Attribute based access control

Hierarchical to: No other components.

FDP_ACF.1.1 (1) The TSF shall enforce the [***Discretionary Access Control policy***] to objects based on the following: [***all operations between users and objects based upon the security attributes defined in Table 7-5, the objects defined by the***

***resource classes listed in Table 7-3, resource class name, and entity name*]**.

FDP_ACF.1.2 (1) The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [***the TOE will permit the requested operation to the protected object (i.e. entity), if the user has a matching secrec entry that matches the class, partial entity name, facility, time of day, day of week, calendar (i.e. day of the year), sysid, program (and library) in the program path, and has an operation in that entry that dominates the requested operation. Furthermore, each secrec that uses Limited Command Facility (LCF) will have an inclusion or exclusion list for the command issued. If that command is in the excluded list defined by LCF, the command will fail to execute. If the command was not on an inclusion list, the command will fail to execute***].

Application Note: *If the value of facility, time of day, day of week, calendar, sysid, library, program or LCF are not present, then no comparison is done for those attributes.*

FDP_ACF.1.3 (1) The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [

a) ***If the ownership matches the user, then any operation is allowed by that user.***
b) If the parameter NOVOLCHK is set, then any operation is allowed for any volume
c) If the parameter NODSNCHK is set, then any operation is allowed for any data set
d) If the parameter NORESCHK is set, then any operation is allowed for any Resource.
e) if the subject is APF authorized
f) If Unix ACID is 0 (root)].

FDP_ACF.1.4 (1) The TSF shall explicitly deny access of subjects to objects based on the [***None***].

Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialization

7.1.2.4 FDP_ACF.1 (2) Security attribute based access control

Hierarchical to: No other components.

FDP_ACF.1.1 (2) The TSF shall enforce the [***Mandatory Access Control***] to objects based on the following: [***all operations between users and objects based upon the security attributes, AccessLevel, Type, Object Security Label, and Subject Security label***].

FDP_ACF.1.2 (2) The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [***the type of access is based on the values of TYPE and ACCESSLevel, as defined below:***

ACCESSLevel	Type=MAC	Type=EQUALMAC	Type=RVRSMAC
Read	User Dominates	Equivalence	Resource Dominates
Read/Write	Equivalence	Equivalence	Equivalence
Write	Resource Dominates	Equivalence	User Dominates

Table 7-4: Mandatory Access Control

a) For Equivalence - if the security label on subject and object match, allow access
b) For User Dominates - if security label on subject dominates object, allow access
c) ***For Resource Dominates - if security label on the object dominates the user, allow access***].

FDP_ACF.1.3 (2) The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [***None***].

FDP_ACF.1.4 (2) The TSF shall explicitly deny access of subjects to objects based on the [***if the object has a security label and the subject does not***].

Application Note: *If the subject has a label and the object does not, access is allowed.*

Application Note: *Subjects are users or applications running on behalf of users, objects are entities grouped as resources. Type is a global access variable that determines what type of access control model to implement. (See table below in section 6.1.3.2) Objects are called entities in the TOE. The TOE reads the subject and object label, establishes dominance,*

and will allow access based on the AccessLevel requested and access control model type.

7.1.3 Class FIA: Identification & Authentication

7.1.3.1 FIA_AFL.1 Authentication Failure Handling

Hierarchical to: No other components

FIA_AFL.1.1 The TSF shall detect when [**an administrator configurable positive integer within** [***0-254***] unsuccessful authentication attempts occur related to [***all user login attempts***].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been [**surpassed**], the TSF shall [***suspend the user's ACID***].

Dependencies: FIA_UAU.1 Timing of authentication

Application Note: *See the CA Top Secret Control Options Guide (PTHRESH).*

Application Note: *The authentication attempt counter cannot be set to zero as the threshold. The number increments from zero to the defined number (up to 254) and suspends the user once that threshold has been exceeded. A Security Administrator must then remove the suspension manually. This sets the value back to zero.*

7.1.3.2 FIA_ATD.1 User attribute definition

Hierarchical to: No other components.

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [see Table 7-5 below]

Role	Attribute	Specific Attribute
User and Scoped Security Administrator	ACID	Unique identifier per user (ACID)
	Password	User definable password
	Passphrase	User definable passphrase
	Authentication Data	Defines what type of authentication method is used per application (Kerberos, pass ticket, passphrase, password, certificate)
	Profiles	List of secrecs for each user
	Security Label	Security labels classify users, data, and resources (MAC only).

	Proxy Records	The LDAPBIND PROXY USER profile record contains the information needed to connect to the LDAP server, including: BINDDN, BINDPW, LDAPHOST, DOMAINDN, LOCALREG, and ENABLE/DISABLE
	Source	How a user or administrator accesses the TOE (Terminal, facility)
	AppID	Application ID
	Vendor Defined Fields	Users of the TOE can further define up to 32k worth of data that can be access control decisions. This only allows further restrictions to be defined for access.
	Time/Date	Times/DaysOfWeek user is allowed to log on to TOE.
	CERTDATA	Identifies the X.509 digital certificate(s) associated with the user/administrator
	Suspend	Indicates whether the ACID is suspended, and the date this action was taken.
Security Administrator	ACID	Unique identifier per user (ACID)
	Password	User definable password
	Passphrase	User definable passphrase
	Authentication Data	Defines what type of authentication method is used per application (Kerberos, pass ticket, passphrase, password, certificate)

Table 7-5: CA Top Secret Generated User Security Attributes

Dependencies: No dependencies

Application Note: *Authorizations can be in the user record, a profile (role), or the all record.*

7.1.3.3 FIA_SOS.1 (1) Verification of Secrets

Hierarchical to: No other components

FIA_SOS.1.1 (1) The TSF shall provide a mechanism to verify that secrets meet [***based on options selected by an administrator may include the following for passwords:***

a) A password must always be set

b) A configured minimum length of characters

c) a minimum configurable number of numeric characters

d) a minimum configurable number of uppercase letters

e) a minimum configurable number of lowercase letters

f) a configured limit of repeating characters

g) Numbers cannot be the only characters used

h) restrict the password to disallow restricted password prefixes

i) Prevent a user from specifying a new password that contains his 8 byte username or first four bytes of his username

j) Expiration date of password is a configurable number of days

***k) A requirement to disallow the new password of a user/administrator to match the previous configured number of passwords*]**.

Dependencies: No dependencies

Application Note: *See CA Top Secret Control Options Guide, NEWPW control option, for quality metrics.*

7.1.3.4 FIA_SOS.1 (2) Verification of Secrets

Hierarchical to: No other components

FIA_SOS.1.1 (2) The TSF shall provide a mechanism to verify that secrets meet [

a. Minimum passphrase length is a configurable number of characters

b. Expiration date of passphrase is a configurable number of days

c. New passphrases cannot match a configured number of a user/administrator's previous passphrases].

Dependencies: No dependencies

7.1.3.5 FIA_SOS.2 TSF Generation of Secrets

Hierarchical to: No other components

FIA_SOS.2.1 The TSF shall provide a mechanism to generate secrets that meet [***see FIA_SOS.1.1 (1)].***

FIA_SOS.2.2 The TSF shall be able to enforce the use of TSF generated secrets for [***password based authentication mechanisms***].

Dependencies: No dependencies

7.1.3.6 FIA_UAU.2 User authentication before any action

Hierarchical to: FIA_UAU.1 Timing of authentication

FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Dependencies: FIA_UID.1 Timing of identification

Application Note: *Without performing a Verify function, no other actions are allowed on the TOE. (i.e., if a user does not have a security context he does not have access, security context are built with the verify command).*

7.1.3.7 FIA_UAU.4 Single-use authentication mechanisms

Hierarchical to: No other components.

FIA_UAU.4.1 The TSF shall prevent reuse of authentication data related to [***passticket authentication***].

Dependencies: No dependencies

Application Note: *Passtickets are issued for a specific session and cannot be used again once that session has ended.*

7.1.3.8 FIA_UAU.5 Multiple authentication mechanisms

Hierarchical to: No other components

FIA_UAU.5.1 The TSF shall provide [***passwords, passtickets, Kerberos, digital certificates or passphrases***] to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the [***application from which the user is requesting system entry***].

Dependencies: No dependencies

Application Note: *The authentication method required for user authentication depends on the application from which the user is accessing the TOE.*

Application Note: *The authentication method does not allow for a user to select what they wish to enter as their credential. The mechanism for authentication is selected by the TOE and can be one of the five authentication mechanisms listed. The only occurrence where more than one mechanism is accepted is for TSO where both password and passphrase can be accepted.*

7.1.3.9 FIA_UID.2 User identification before any action

Hierarchical to: FIA_UID.1 Timing of identification

FIA_UID.2.1 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Dependencies: No dependencies

7.1.3.10 FIA_USB.1 User-subject binding

Hierarchical to: No other components

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [***see Table 7-5***].

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [
1. the TOE will create the user's ACID
2. ***the TOE will create the user's security environment***].

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [***none***].

Dependencies: FIA_ATD.1 User attribute definition

7.1.4 Class FMT: Security Management

7.1.4.1 FMT_MOF.1 (1) Management of security functions behavior

Hierarchical to: No other components

FMT_MOF.1.1 (1) The TSF shall restrict the ability to [**determine the behavior of, modify the behavior of**] the functions [***see Table 7-6***] to [***the Security Administrator***].

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

7.1.4.2 FMT_MOF.1 (2) Management of security functions behavior

Hierarchical to: No other components

FMT_MOF.1.1 (2) The TSF shall restrict the ability to [**determine the behavior of, modify the behavior of**] the functions [***see Table 7-6]*** to [***the Scope Security Administrator according to their scope***].

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

7.1.4.3 FMT_MOF.1 (3) Management of security functions behavior

Hierarchical to: No other components.

FMT_MOF.1.1 (3) The TSF shall restrict the ability to [**determine the behavior of, modify the behavior of**] the functions [***self passwords and passphrases***] to [***users***].

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

7.1.4.4 FMT_MOF.1 (4) Management of security functions behavior

Hierarchical to: No other components.

FMT_MOF.1.1 (4) The TSF shall restrict the ability to [**determine the behavior of, modify the behavior of**] the functions [***see Table 7-6***] to [***users that own the object***].

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

Application Note: *Resource ownership implies that the owning ACID has an access level of ALL.*

Operation	Administrative Functions
Define	Subject security attributes
Change	Subject security attributes
Manage	User identities
Manage	Authentication data by an administrator
Control	Authentication data that users are allowed to manage
Manage	User's own authentication data
Manage	Password policies

Manage	Threshold for unsuccessful authentication attempts
Manage	Actions to be taken in the event of authentication failure
Manage	Attributes used to make explicit access or denial based decisions
Manage	Audit events
Manage	Groups with read access to the audit records

Table 7-6: Administrative Functions on the TOE

7.1.4.5 FMT_MSA.1 Management of security attributes

Hierarchical to: No other components

FMT_MSA.1.1 The TSF shall enforce the [***the Mandatory Access Control and Discretionary Access Control policies***] to restrict the ability to [**modify, delete,** [***manage, add, control, change***]] the security attributes [***defined in Table 7-6***] to [***security administrators or scoped security administrators within their scope***].

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

7.1.4.6 FMT_MSA.3 Static attribute initialization

Hierarchical to: No other components.

FMT_MSA.3.1 The TSF shall enforce the [***Mandatory Access Control and Discretionary Access Control policies***] to provide [**restrictive**] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [***security administrators or scoped security administrators within their scope***] to specify alternative initial values to override the default values when an object or information is created.

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

7.1.4.7 FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [***see Table 7-6***].

Dependencies: No dependencies

7.1.4.8 FMT_SMR.1 Security roles

Hierarchical to: No other components

FMT_SMR.1.1 The TSF shall maintain the roles [***security administrator, scoped security administrator, and user***]

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

Dependencies: FIA_UID.1 Timing of identification

Application Note: *The scoped security administrator includes the following: MSCA, SCA, ZCA, VCA, DCA, and LSCA. See section 9.1.4.1 for more information on the security administrators.*

7.1.5 Class FTA: TOE Access

7.1.5.1 FTA_TSE.1 TOE session establishment

Hierarchical to: No other components

FTA_TSE.1.1 The TSF shall be able to deny session establishment based on [

1. ***The user's status is suspended***
2. ***A policy which limits user access based on :***
    a. ***Time/Date***
    b. ***Source (terminal ID, IP address, POE)***
    c. ***AppID (the application the user is trying to authenticate by)].***

Dependencies: No dependencies

7.2 Operations Defined

The notation, formatting, and conventions used in this security target (ST) are consistent with version 3.1 of the Common Criteria for Information Technology Security Evaluation. All of the components in this ST are taken directly from Part 2 of the CC except the ones noted with "_EXT" in the component name. Font style and clarifying information conventions were developed to aid the reader.

The CC permits four functional component operations: assignment, iteration, selection, and refinement to be performed on functional requirements. These operations are defined in Common Criteria, Part 1 as:

7.2.1 Assignments Made

An assignment allows the specification of parameters and is specified by the ST author in [***italicized bold text***].

7.2.2 Iterations Made

An iteration allows a component to be used more than once with varying operations and is identified with the iteration number within parentheses after the short family name, e.g. FAU_GEN.1 (1), FAU_GEN.1 (2).

7.2.3 Selections Made

A selection allows the specification of one or more items from a list and is specified by the ST author in [**bold text**].

7.2.4 Refinements Made

A refinement allows the addition of details and is identified with "Refinement:" right after the short name. ~~The old text is shown with a strikethrough~~ and ***the new text is specified by italicized bold and underlined text***.

8 Security Assurance Requirements

This section identifies the Security Assurance Requirement components met by the TOE. These assurance components meet the requirements for EAL4 augmented with ALC_FLR.1 and ASE_TSS.2.

8.1 Security Architecture

8.1.1 Security Architecture Description (ADV_ARC.1)

ADV_ARC.1.1D: The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2D: The developer shall design and implement the TSF so that it is able to protect itself from tampering by un-trusted active entities.

ADV_ARC.1.3D: The developer shall provide a security architecture description of the TSF.

ADV_ARC.1.1C: The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2C: The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3C: The security architecture description shall describe how the TSF initialization process is secure.

ADV_ARC.1.4C: The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5C: The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

ADV_ARC.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.1.2 Functional Specification with Complete Summary (ADV_FSP.4)

ADV_FSP.4.1D The developer shall provide a functional specification.

ADV_FSP.4.2D The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.4.1C The functional specification shall completely represent the TSF.

ADV_FSP.4.2C The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.4.3C The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.4.4C The functional specification shall describe all actions associated with each TSFI.

ADV_FSP.4.5C The functional specification shall describe all direct error messages that may result from an invocation of each TSFI.

ADV_FSP.4.6C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_FSP.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.4.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

8.1.3 Implementation Representation of the TSF (ADV_IMP.1)

ADV_IMP.1.1D The developer shall make available the implementation representation for the entire TSF.

ADV_IMP.1.2D The developer shall provide a mapping between the TOE design description and the sample of the implementation representation. Content and presentation elements:

ADV_IMP.1.1C The implementation representation shall define the TSF to a level of detail such that the TSF can be generated without further design decisions.

ADV_IMP.1.2C The implementation representation shall be in the form used by the development personnel.

ADV_IMP.1.3C The mapping between the TOE design description and the sample of the implementation representation shall demonstrate their correspondence. Evaluator action elements:

ADV_IMP.1.1E The evaluator *shall confirm* that, for the selected sample of the implementation representation, the information provided meets all requirements for content and presentation of evidence.

8.1.4 Architectural Design (ADV_TDS.3)

ADV_TDS.3.1D The developer shall provide the design of the TOE.

ADV_TDS.3.2D The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

ADV_TDS.3.1C The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.3.2C The design shall describe the TSF in terms of modules.

ADV_TDS.3.3C The design shall identify all subsystems of the TSF.

ADV_TDS.3.4C The design shall provide a description of each subsystem of the TSF.

ADV_TDS.3.5C The design shall provide a description of the interactions among all subsystems of the TSF.

ADV_TDS.3.6C The design shall provide a mapping from the subsystems of the TSF to the modules of the TSF.

ADV_TDS.3.7C The design shall describe each SFR-enforcing module in terms of its purpose and interaction with other modules.

ADV_TDS.3.8C The design shall describe each SFR-enforcing module in terms of its SFR-related interfaces, return values from those interfaces, interaction with and called interfaces to other modules.

ADV_TDS.3.9C The design shall describe each SFR-supporting or SFR-non-interfering module in terms of its purpose and interaction with other modules.

ADV_TDS.3.10C The mapping shall demonstrate that all behavior described in the TOE design is mapped to the TSFIs that invoke it.

ADV_TDS.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_TDS.3.2E The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

8.2 Guidance Documents

8.2.1 Operational User Guidance (AGD_OPE.1)

AGD_OPE.1.1D The developer shall provide operational user guidance.

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.2.2 Preparative Procedures (AGD_PRE.1)

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

AGD_PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

AGD_PRE.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1.2E The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

8.3 Lifecycle Support

8.3.1 Authorization Controls (ALC_CMC.4)

ALC_CMC.4.1D The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.4.2D The developer shall provide the CM documentation.

ALC_CMC.4.3D The developer shall use a CM system.

ALC_CMC.4.1C The TOE shall be labeled with its unique reference.

ALC_CMC.4.2C The CM documentation shall describe the method used to uniquely identify the configuration items.

ALC_CMC.4.3C The CM system shall uniquely identify all configuration items.

ALC_CMC.4.4C The CM system shall provide automated measures such that only authorized changes are made to the configuration items.

ALC_CMC.4.5C The CM system shall support the production of the TOE by automated means.

ALC_CMC.4.6C The CM documentation shall include a CM plan.

ALC_CMC.4.7C The CM plan shall describe how the CM system is used for the development of the TOE.

ALC_CMC.4.8C The CM plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

ALC_CMC.4.9C The evidence shall demonstrate that all configuration items are being maintained under the CM system.

ALC_CMC.4.10C The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.

ALC_CMC.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.3.2 CM Scope (ALC_CMS.4)

ALC_CMS.4.1D The developer shall provide a configuration list for the TOE.

ALC_CMS.4.1C The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; the implementation representation; and security flaw reports and resolution status.

ALC_CMS.4.2C The configuration list shall uniquely identify the configuration items.

ALC_CMS.4.3C For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

ALC_CMS.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.3.3 Delivery Procedures (ALC_DEL.1)

ALC_DEL.1.1D The developer shall document procedures for delivery of the TOE or parts of it to the consumer.

ALC_DEL.1.2D The developer shall use the delivery procedures.

ALC_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

ALC_DEL.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.3.4 Identification of Security Measures (ALC_DVS.1)

ALC_DVS.1.1D The developer shall produce development security documentation.

ALC_DVS.1.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC_DVS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DVS.1.2E The evaluator shall confirm that the security measures are being applied.

8.3.5 Life-cycle Definition (ALC_LCD.1)

ALC_LCD.1.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC_LCD.1.2D The developer shall provide life-cycle definition documentation.

ALC_LCD.1.1C The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

ALC_LCD.1.2C The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

ALC_LCD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.3.6 Tools and techniques (ALC_TAT.1)

ALC_TAT.1.1D The developer shall identify each development tool being used for the TOE.

ALC_TAT.1.2D The developer shall document the selected implementation-dependent options of each development tool.

ALC_TAT.1.1C Each development tool used for implementation shall be well defined.

ALC_TAT.1.2C The documentation of each development tool shall unambiguously define the meaning of all statements as well as all conventions and directives used in the implementation.

ALC_TAT.1.3C The documentation of each development tool shall unambiguously define the meaning of all implementation-dependent options.

ALC_TAT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.3.7 Flaw reporting procedures (ALC_FLR.1)

ALC_FLR.1.1D The developer shall document flaw remediation procedures addressed to TOE developers.

ALC_FLR.1.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.1.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.1.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.1.4C The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.4 Security Target Evaluation

8.4.1 Conformance Claims (ASE_CCL.1)

ASE_CCL.1.1D The developer shall provide a conformance claim.

ASE_CCL.1.2D The developer shall provide a conformance claim rationale.

ASE_CCL.1.1C The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.

ASE_CCL.1.2C The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.

ASE_CCL.1.3C The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.

ASE_CCL.1.4C The CC conformance claim shall be consistent with the extended components definition.

ASE_CCL.1.5C The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.

ASE_CCL.1.6C The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.

ASE_CCL.1.7C The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.

ASE_CCL.1.8C The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.

8.4.2 Extended Components Definition (ASE_ECD.1)

ASE_ECD.1.1D The developer shall provide a statement of security requirements.

ASE_ECD.1.2D The developer shall provide an extended components definition.

ASE_ECD.1.1C The statement of security requirements shall identify all extended security requirements.

ASE_ECD.1.2C The extended components definition shall define an extended component for each extended security requirement.

ASE_ECD.1.3C The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.

ASE_ECD.1.4C The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.

ASE_ECD.1.5C The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.

ASE_ECD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_ECD.1.2E The evaluator shall confirm that no extended component can be clearly expressed using existing components.

8.4.3 ST Introduction (ASE_INT.1)

ASE_INT.1.1D The developer shall provide an ST introduction.

ASE_INT.1.1C The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.

ASE_INT.1.2C The ST reference shall uniquely identify the ST.

ASE_INT.1.3C The TOE reference shall identify the TOE.

ASE_INT.1.4C The TOE overview shall summarize the usage and major security features of the TOE.

ASE_INT.1.5C The TOE overview shall identify the TOE type.

ASE_INT.1.6C The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.

ASE_INT.1.7C The TOE description shall describe the physical scope of the TOE.

ASE_INT.1.8C The TOE description shall describe the logical scope of the TOE.

ASE_INT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_INT.1.2E The evaluator shall confirm that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

8.4.4 Security Objectives (ASE_OBJ.2)

ASE_OBJ.2.1D The developer shall provide a statement of security objectives.

ASE_OBJ.2.2D The developer shall provide a security objectives rationale.

ASE_OBJ.2.1C The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.

ASE_OBJ.2.2C The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.

ASE_OBJ.2.3C The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.

ASE_OBJ.2.4C The security objectives rationale shall demonstrate that the security objectives counter all threats.

ASE_OBJ.2.5C The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.

ASE_OBJ.2.6C The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.

ASE_OBJ.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.4.5 Security Requirements (ASE_REQ.2)

ASE_REQ.2.1D The developer shall provide a statement of security requirements.

ASE_REQ.2.2D The developer shall provide a security requirements rationale.

ASE_REQ.2.1C The statement of security requirements shall describe the SFRs and the SARs.

ASE_REQ.2.2C All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.

ASE_REQ.2.3C The statement of security requirements shall identify all operations on the security requirements.

ASE_REQ.2.4C All operations shall be performed correctly.

ASE_REQ.2.5C Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.

ASE_REQ.2.6C The security requirements rationale shall trace each SFR back to the security objectives for the TOE.

ASE_REQ.2.7C The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.

ASE_REQ.2.8C The security requirements rationale shall explain why the SARs were chosen.

ASE_REQ.2.9C The statement of security requirements shall be internally consistent.

ASE_REQ.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.4.6 Security Problem Definition (ASE_SPD.1)

ASE_SPD.1.1D The developer shall provide a security problem definition.

ASE_SPD.1.1C The security problem definition shall describe the threats.

ASE_SPD.1.2C All threats shall be described in terms of a threat agent, an asset, and an adverse action.

ASE_SPD.1.3C The security problem definition shall describe the OSPs.

ASE_SPD.1.4C The security problem definition shall describe the assumptions about the operational environment of the TOE.

ASE_SPD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.4.7 TOE Summary Specification (ASE_TSS.2)

ASE_TSS.2.1D The developer shall provide a TOE summary specification.

ASE_TSS.2.1C The TOE summary specification shall describe how the TOE meets each SFR.

ASE_TSS.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ASE_TSS.2.2C The TOE summary specification shall describe how the TOE protects itself against interference and logical tampering.

ASE_TSS.2.2E The evaluator *shall confirm* that the TOE summary specification is consistent with the TOE overview and the TOE description.

ASE_TSS.2.3C The TOE summary specification shall describe how the TOE protects itself against bypass.

8.5 Tests

8.5.1 Analysis of Coverage (ATE_COV.2)

ATE_COV.2.1D The developer shall provide an analysis of the test coverage.

ATE_COV.2.1C The analysis of the test coverage shall demonstrate the correspondence between the tests in the test documentation and the TSFIs in the functional specification.

ATE_COV.2.2C The analysis of the test coverage shall demonstrate that all TSFIs in the functional specification have been tested.

ATE_COV.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.5.2 Basic Design (ATE_DPT.2)

ATE_DPT.2.1D The developer shall provide the analysis of the depth of testing.

ATE_DPT.2.1C The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems and SFR-enforcing modules in the TOE design.

ATE_DPT.2.2C The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.

ATE_DPT.2.3C The analysis of the depth of testing shall demonstrate that the SFR-enforcing modules in the TOE design have been tested.

ATE_DPT.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.5.3 Functional Tests (ATE_FUN.1)

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation

ATE_FUN.1.1C The test documentation shall consist of test plans, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.3C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4C The actual test results shall be consistent with the expected test results.

ATE_FUN.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.5.4 Independent Testing (ATE_IND.2)

ATE_IND.2.1D The developer shall provide the TOE for testing.

ATE_IND.2.1C The TOE shall be suitable for testing.

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

ATE_IND.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

ATE_IND.2.3E The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

8.6 Vulnerability Assessment

8.6.1 Vulnerability Analysis (AVA_VAN.3)

AVA_VAN.3.1D The developer shall provide the TOE for testing.

AVA_VAN.3.1C The TOE shall be suitable for testing.

AVA_VAN.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.3.2E The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.3.3E The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design, security architecture description and implementation representation to identify potential vulnerabilities in the TOE.

AVA_VAN.3.4E The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Enhanced-Basic attack potential.

9 TOE Summary Specification

9.1 TOE Security Functions

The following sections identify the security functions of the TOE. They include Security Audit, Identification and Authentication, User Data Protection, Security Management, and TOE Access.

9.1.1 Security Audit

CA Top Secret provides numerous report generators to create various reports for audit trail purposes. Any attempted violation CA Top Secret detects appears within a report. In addition, standard reports display each update to the CA Top Secret security file. Thus, any addition, change, or deletion of any of CA Top Secret user or permission information is visible to a person reviewing these reports. Records produced from Recovery file provide information presented in the CA Top Secret Audit Changes reports. Other reports are also available to record occurrences or produce audit trails of authorized activities. These are of various types. A user can produce each type independently of the others. They are explained in the following sections.

Through hierarchal scope an administrator can be restricted to only certain user records. CA Top Secret enforces the existing permissions for any other type of access.

9.1.1.1 Security & Audit Privileges

The TSSAUDIT utility lets authorized users with the appropriate privileges monitor changes to the Security File, sensitive facilities, and data areas. An authorized user with the audit privilege has the authority to audit the following:

- **Users** - The Administrator attaches the AUDIT attribute to the user's ACID.
- **Resources** - The Administrator updates the AUDIT record with the resource or resource prefix (up to 64 characters) and optional access levels to be audited.
- **A Specific Permission** - The Administrator includes the ACTION (AUDIT) keyword on the PERMIT command function.

9.1.1.2 Audit/Tracking File

By default, CA Top Secret logs all violations to the System Management Facility (SMF). The Audit/Tracking file is also available for logging violations and audited events. The Audit/Tracking file lets administrators:

- Eliminate SMF logging for violations and activity
- Configure so that logging to the Audit/Tracking file cannot be suppressed. This configuration eliminates a potential security exposure
- Generate reports for up-to-the-minute information
- Extract security events that are logged to produce reports using the TSSUTIL batch utility

TSSUTIL can be used to routinely archive audit-tracking information. CA Top Secret allows the online, real-time display of selected security activity on TSO and CICS terminals with the TSSTRACK utility. The TSSTRACK online tracking capability can only be used if the Audit/Tracking file is used. TSSAUDIT permits auditors to monitor changes to the security file and sensitive z/OS facilities and data areas.

Whenever an option is changed, the TOE automatically maintains an audit trail, including the ACID that changed the control option.

9.1.1.3 Violations and Logging

The LOG control option is the primary mechanism that controls the reporting of system activity and violations for all facilities. The LOG sub-option of the FACILITY control option can also be used to control individual facilities. Among the options for tracking security breaches are:

- A violation generates a descriptive message sent to the security console, the user's online terminal, or both.
- Job/session initiations and terminations, resource accesses, security violations, or any combination of these events are logged for all or selected facilities.
- A record of selected types of events is logged to SMF, the Audit/Tracking file, or both. The Audit/Tracking File can be shared across CPUs, providing a single reference source for security events.

When ACTION (AUDIT) is used with the PERMIT command function, it audits all accesses to the specified resource regardless of the mode or logging options of the user.

In FAIL mode, messages are issued regardless of the LOG control option specifications. The logging of activity is transparent to the user. The format logged is an SMF type-80 format.

9.1.1.4 Security Event Logging

CA Top Secret provides the following batch utility programs to help monitor and control system security, log system activity, and perform disaster recovery.

9.1.1.4.1 TSSAUDIT

This batch utility is run in order to monitor changes to the security file and sensitive facilities and data areas. It can be used to list:

- ACIDs that possess administrative or special privileges (such as AUDIT, CONSOLE) or any of the security bypass attributes
- The changes made to the security file. TSSAUDIT generates this information for a given date or time span by examining the recovery file. All changes made by a particular ACID can also be requested. The ACID must fall within the scope of the administrator running TSSAUDIT
- Information about modules in APF-authorized libraries
- Information about site-written (non-IBM) SVCs, the Program Properties Table (PPT), and the Terminal Monitor Program's (TMP) authorized program lists
- The last two capabilities are especially useful for pinpointing security weaknesses

9.1.1.4.2 TSSCPR

This utility is run against the CPF recovery file to produce a flat file record. This record can then be filtered through the TSSREPORT3 EARL Report option or through another report writer to depict the contents of the CPF recovery file.

9.1.1.4.3 TSSOERPT

This batch utility program is run to process security-related activity recorded in SMF data sets. To monitor user activity in an OpenEdition MVS environment, CA Top Secret logs security events under OpenEdition MVS to SMF using the standard CA Top Secret SMF record. Log records are written for any security event that denies the ACID access to an OpenEdition MVS facility. These records can assist administrators in determining the UID and GID of the ACID involved in the attempted access.

9.1.1.4.4 TSSPROT

This utility is run to determine which of these data sets have and do not have their security bit indicators turned on. The TSSPROT utility pertains to VSAM and non-VSAM data sets in an SU32 environment. Administrators can also use TSSPROT to turn security bits on or off for specific data sets or all data set located on accessible volumes.

Note: Only the MSCA or an SCA can use this utility.

9.1.1.4.5 TSSRECVR

This utility is run to aid recovery from loss or corruption of the security file. During normal system operation, the TOE maintains a record of all changes to the security file in the recovery file, which is a perpetual file. Changes are recorded to the file in a wraparound format. Therefore, this file must be large enough to accommodate all changes that occur between security file backups.

9.1.1.4.6 TSSRPTST

This batch utility program is run to process and display the output that the SAF SECTRACE command sends to SMF. To run the TSSRPTST report, administrators must have already run the SAF SECTRACE operator command and set the output destination to SMF. With few exceptions, the TOE processes all z/OS SAF security requests by default. The SAF Trace report displays the monitored RACROUTE parameter list passed by requests for SAF services. This report also displays additional environmental information, such as job name, ACID, and the program issuing the SAF call.

9.1.1.4.7 TSSTRACK

This utility is used to monitor security-related events from an online terminal in a real-time manner. This functionality lets the security administrator monitor suspicious activity "as it happens." Furthermore, TSSTRACK enables all CPUs on a single security audit file to be monitored from a single terminal. TSSTRACK can go back to a specified date to focus on a selected facility or on violations only.

The events that security administrators can monitor using TSSTRACK are limited by their administrative scope. All the information that TSSTRACK displays is obtained from the CA Top Secret Audit/Tracking file; only information logged to this file can be monitored.

9.1.1.5 Security Reports

CA Top Secret provides the following batch utilities to translate resource access attempts and security events into reports, as stated in the following descriptions.

9.1.1.5.1 TSSCHART

TSSCHART is a batch utility that provides an overview of the security file architecture by generating block charts of ACIDs/owned-resource relationships. This functionality lets administrators examine the security hierarchy "at-a-glance." Scope restrictions are honored so that if a properly authorized DCA uses the TSSCHART utility, the block chart the DCA receives only illustrates the ACIDs and resource relationships within the DCA's department. To view the installation as a whole, the MSCA or a properly authorized SCA must run the utility.

9.1.1.5.2 TSSUTIL

TSSUTIL is a flexible report generator/extract utility that provides batch reports of any security-related events logged to the Audit/Tracking File and/or SMF. Selection criteria for TSSUTIL include the following:

- ACIDs
- Jobs
- Specific resources
- Resource types

- Facilities
- Zones
- Divisions
- Departments
- Dates
- Types of access
- CPUs
- Violations
- Audited incidents

9.1.1.5.3 TSSCFILE

TSSCFILE is a batch utility that produces a fixed-format output file whose records parallel the TSS LIST command output. This file is then used with a site or vendor extract tool to write customized reports based on the configuration and content of the security file. As with the other batch reporting options, scope and administrative authority limitations are honored.

9.1.1.5.4 TSSREPORT and TSSREPORT2

Under CA Top Secret for z/OS, TSSREPORT and TSSREPORT2 use the output of TSSCFILE and TSSUTIL, respectively, to produce pre-formatted CA Earl Reports. Administrators can choose from several different report layouts, or custom layouts can be designed.

9.1.1.5.5 TSSRPTST

The SAF Trace Report lets administrators display the monitored RACROUTE parameter list passed by requests for SAF services. This report also displays additional environmental information, such as job name, ACID, and the program issuing the SAF call.

The TSSRPTST report formats and displays the output the SAF SECTRACE command sends to SMF. To run the TSSRPTST report, an administrator must have already run the SAF SECTRACE operator command and set the output destination to SMF. With few exceptions, the TOE processes all z/OS SAF security requests by default.

9.1.2 Identification and Authentication

The TOE controls how, when, and which resources a user can access. The TOE requires that each end user have a valid accessor ID (ACID) and password before entering the system. CA Top Secret security information is stored in a single shared database for all CPUs, the Security File. Within the Security File, each user is associated with a unique

Security Record (secrec) that lets CA Top Secret associate access authorizations with users.

9.1.2.1 ACIDs

By default, the TOE requires that all ACIDs are password protected. A Security Administrator assigns the first password. The user associated with the ACID changes the password immediately (or later) when it expires. For more information on ACIDs, see Section 9.1.3.5. Administrators can set limits as to how many violations users can accumulate per session (0-254) and define an automatic action through the VTHRESH control option. The TOE prevents further access of any kind by locking out the terminal. This forces the user to sign off. Note that the value of zero cannot be set as the threshold for lockout. Instead, once a user has been suspended, an administrator would set that value to zero to allow the individual the chance to authenticate to the TOE again.

9.1.2.2 Suspended User

The TOE monitors the number of failed attempts to authenticate to the TOE. Once an administrator configured threshold of failed authentication attempts is surpassed for a particular ACID, the TOE will suspend that ACID. When an ACID is suspended the TOE will not allow the user/administrator associated with the suspended ACID to authenticate to the TOE. The threshold is configured by an administrator and can be set to a value within zero to 254 failed authentication attempts. Once a user/administrator has been suspended, an administrator must reactivate the suspended ACID to allow them to attempt to authenticate to the TOE again, setting the value back to zero.

9.1.2.3 Authentication Methods

The TOE employs multiple authentication methods. The following methods are used in the evaluated configuration: Kerberos, passtickets, passwords, passphrases, and certificates. They are described in the following sections.

9.1.2.3.1 Kerberos

The TOE employs Kerberos to store and administer authentication information for TOE users and administrators. Kerberos verifies requests as a trusted third party authentication service and confirms the identities of users and administrators.

9.1.2.3.2 PassTickets

A PassTicket is a generated character string that can be used in place of a password, with the following constraints:

- A specific PassTicket may be used for authentication once
- The PassTicket must be used within 10 minutes of being generated

When a user or administrator authenticates to the TOE utilizing the PassTicket mechanism, the user/administrator must provide their PassTicket to the TOE to be checked for authorization. The TOE will then request the ICSF component to decrypt the

PassTicket, and the TOE will then evaluate the following combinations of information to determine if the provided PassTicket can be used for authentication:

1. The application name concatenated with the group name and ACID. (Note: Groups are not used in the evaluated configuration)
2. The application name concatenated with the ACID.
3. The application name concatenated with the group name. (Note: Groups are not used in the evaluated configuration)
4. The application name.

The TOE will check each combination until one of them matches the PassTicket's character string or all combinations have been checked. If a matching combination is found the TOE then checks the PassTicket's session key values to determine if they are valid, and if they are this authorization check is successful and further authorization checks can be performed. Refer to Section 9.1.5 for more information on the TOE Access authorization checks. However, if none of the combinations match the PassTicket's character string or if the session key value(s) are not valid, user/administrator has failed authentication, and the count for number of failed authentications is increased by one (refer to Section 9.1.2.7 for suspending a user/administrator).

The TOE will use the PassTicket verification process if configured by an administrator. The TOE also requires the administrator to generate a PassTicket by calling the z/OS PassTicket Generator callable service. The administrator must provide the PassTicket Generator callable service with the ACID and the APPLID which will be used for the PassTicket's combination. The callable service will then use the current time and date stamp, and extract the necessary APPLID record from the security file to obtain the security key which the TOE will recognize as being associated with the application utilized by the user/administrator for authentication. The PassTicket Generator callable service will utilize all four pieces of information and will applying an ICSF cryptographic algorithm to the PassTicket string to generate the 8 byte PassTicket. The administrator can then provide the user/administrator with the 8 byte PassTicket.

9.1.2.3.3 Digital Certificates

The TOE provides the ability to perform authorization checks based on an X.509 Digital Certificate. Digital certificates provide a means of authentication through the use of public-key cryptography and a trusted third party, known as a Certification Authority. A digital certificate is generated by the Certification Authority and is identified uniquely by its serial number and by the associated distinguished name of the Certification Authority ("issuer's distinguished name").

A digital certificate is associated to user of the TOE through the user's ACID record. When using certificates, the following rules apply:

- More than one certificate can be added to a user's ACID record

- Each certificate is unique to a particular user; a certificate cannot be added to more than one ACID unless the certificate is attached to a key ring
- In order to define multiple digital certificates, a special file (VSAM) may be needed to hold the record

CA Top Secret provides complete functionality to generate, install, and maintain digital certificates, key rings, and digital certificate mappings, including the following:

- Request ICSF to generate a digital certificate and a public/private key pair
- Create a PKCS #12 certificate package
- Create a PKCS #10 certificate request
- Export a digital certificate or certificate package and private key from CA Top Secret to a z/OS dataset
- Display a certificate that is in a z/OS dataset and determine if it is associated with a CA Top Secret user/administrator
- Display a certificate registered with CA Top Secret
- Automatically register a digital certificate with CA Top Secret
- Associate a CA Top Secret user/administrator with a digital certificate
- Change, display, and delete information about a digital certificate for a CA Top Secret user/administrator
- Create, change, display, and delete a key ring
- Add and remove a certificate from a key ring
- Assign a CA Top Secret user/administrator to a group of certificates via User ID mapping
- Assign a CA Top Secret user/administrator to a group of certificates based on system ID, application ID, or application-defined variables
- Change, delete, and display a CA Top Secret User ID mapping

9.1.2.3.4 Key Rings

A key ring is a collection of digital certificates associated with an individual user. Once a user has had their identity verified to a system by a certificate that is unique to the user, the user can access additional resources through the certificates on their key ring. Key rings provide an installation-wide method to share digital certificates across multiple servers. A user can have more than one key ring.

9.1.2.3.5 Certificate Associations

CA Top Secret lets you check if a certificate has already been added to the CA Top Secret Security File and what ACID it is associated with. Once a certificate has been added to CA Top Secret it can be exported to a new data set.

9.1.2.3.6 Tokens

CA Top Secret employs the PKCS #11 cryptographic token standard. PKCS #11 tokens are created using the TOE's gskkyman utility. Each token has a unique token name, or label, specified by the end user or application when the token is created. This standard is used for granting access to token information on a personal identification number (PIN).

9.1.2.3.7 Password Policy

Password assignment is controlled by the TOE control option values set and stored within the TOE. The TOE requires password protection for all User and Control ACIDs by default. In addition to a password, an ACID can have an optional passphrase. Passphrases can be used instead of passwords in applications that support them.

Passwords have a maximum length of eight characters. A passphrase can be from 9 to 100 characters long and can include mixed-case letters, numbers, and special characters including blanks. The same ACID can have a password for applications that accept passwords only and a password phrase for other applications.

Security Administrators have the capability to set the TOE's password policy. The following restrictions are available as options for Security Administrators to set the password policy for the TOE:

- Minimum of one alphabetic character
- Minimum of one numeric character
- Prevent a user from specifying a new password that contains his 8 byte username or first four bytes of his username
- Minimum of one lowercase letter
- Minimum length of a password (1 to 8 characters)
- Only numbers can be used in a new password
- Limit the number of pairs of repeating characters in a new password
- Initial characters match one of the entries in the Restricted Password (RPW) prefix list
- Minimum of one character selected from the PASSCHAR list
- Must contain a special character

- Must not repeat characters
- Minimum of one uppercase letter
- Allow the administrator to create a mask to dictate the type of character accepted for each position in a password:
    - A - Any alphabetic character
    - C - Consonant
    - V - Vowel (A,E, I, O, U, and Y)
    - N - Numeric character (0 — 9)
    - X - Non-vowel (National character (@,#,$), or alphabetic including Y but excluding other vowels)
    - ? - Any character

9.1.2.3.8 Password Defaults

Security Administrators are able to set password defaults with the NEWPW control option. The defaults are:

- The password must be at least four characters long.
- The same letter cannot be repeated in succession.
- The password cannot match any of the entries in the restricted password list.
- The password cannot match the ACID or the first four characters of any word in the associated NAME field.
- CA Top Secret issues Warning messages three days before the password expires.
- The password cannot be too similar to the previous password.
- Users cannot change a password more often than once each day (except for security administrators and random password users).

Additionally, a Security Administrator can generate a random password in accordance with the defined password policy. The Security Administrator would specify the ACID in the USERID field and specify the new password as "random." Once this is done, the TOE will process the operation and generate a password at random from the requirements defined in the password policy. The new password is then provided to the user with the selected ACID who can then authenticate to the TOE.

9.1.2.3.9 Passphrase Defaults

Security Administrators can set passphrase defaults with the NEWPHRASE control option. The defaults are:

- The password phrase:

- Must be at least nine characters long
- Can be up to 100 characters long

- The TOE issues warning messages three days before the password phrase expires
- Users cannot change a phrase more often than once each day (only Security Administrators are the exception this rule)

9.1.2.4 PSTKAPPL

PSTKAPPL defines the application ID. Depending on the application, the secured sign on function uses a specific method to determine the application ID:

- For CICS, IMS, or APPC applications, the application ID is defined using the standard naming conventions used to define these applications in a VTAM APPL statement.
- For TSO, the application ID is defined by prefacing the SMF identifier of the system with the characters "TSO". For example, TSOXE05 is the application ID for TSO on machine MVXE05. The SMF system ID resides in the SMFPRMxx member of SYS1.PARMLIB.
- For z/OS batch jobs that include CA Top Secret passwords in the JCL, replace the password with a PassTicket. The application ID for batch jobs is defined by prefacing the SMF identifier of the system with the characters "z/OS". For example, MVSXE05 is the application ID for all batch jobs on machine MVXE05.

9.1.3 User Data Protection

The TOE performs access control based on Mandatory Access Control (MAC) and Discretionary Access Control (DAC) policies, which are created and modified by Security Administrators. In addition to the MAC and DAC policies, a user is granted access based on individual permissions according to his ACID, scope, and authority.

When a user initiates a job or signs on to an online facility, CA Top Secret obtains the user's Security Record (secrec) from the Security File and places it in the user's address space for the life of the session. CA Top Secret checks all access validation against the user's Security Record.

The TOE employs the following system entry restrictions:

- **Facility Authorization -** CA Top Secret protects access to system facilities—such as, CICS, TSO, and BATCH—by requiring that the user be authorized to use the facility. Only the MSCA can access any facility by default. All other users must be explicitly authorized to use a facility or any number of facilities through a TSS CREATE or ADDTO function. Administrators can also prevent an ACID from performing multiple simultaneous signons.

- **Access Authorization -** CA Top Secret can control system entry by restricting access to terminals, readers, and CPUs. Administrators can define and restrict the following device types to CA Top Secret:

    - Online terminals (TCAM, VTAM, BTAM, VM local, logical, and bi-synch)
    - Remote (RJE) and local JES readers
    - Internal readers
    - NJE nodes

The specific resource class name - such as VMRDR, TERMINAL or CPU - is the keyword to determine access authorizations and restrictions. The actual resource can be indicated by its full identifier or a Generic Prefix.

- **Batch Job Validation -** A batch job must be associated with an ACID so that CA Top Secret can determine which facilities and resources it can access and how they can be accessed. To CA Top Secret, a batch job's ACID is simply another ACID with an associated Security Record and a set of specific access authorizations. All the system entry restriction options can be specified for a User ACID, including facility, source of origin, and CPU restrictions.

- **Validation -** For jobs submitted through the following entities:

    - TSO
    - CICS
    - IMS

CA Top Secret provides an additional layer of security control beyond the basic batch job validation. The focus of this security layer is whether the submitter has the authority to submit the job. That is, CA Top Secret checks whether the ACID of the submitter is authorized to submit using the ACID associated with this job. If the user is not authorized, the job is flushed at submission time before the job is initiated.

By default, only defined users are allowed to submit jobs for execution under their own ACID. Explicit authority is required to allow a user to execute jobs using other ACIDs.

- **Terminal Locking -** The terminal locking option protects unattended terminals against unauthorized access. Terminal locking prevents use of the terminal until it

is logged off or unlocked. Terminal locking can be triggered automatically by CA Top Secret or through a user-initiated command.

- **Automatic Locking -** CA Top Secret automatically locks a terminal that has been inactive for a pre-established duration. Automatic locking thresholds can be established at both the user and facility level.
- **Console Protection for MVS -** CA Top Secret provides several protection options designed to prevent operators or other personnel from executing sensitive started tasks or changing security control options without proper identification and password authentication.

9.1.3.1 Resource Classes

The objects that CA Top Secret protects are called resources. Resources must be owned before their use can be authorized (see Section 9.1.3.1.1 for further information on resource ownership). Each resource in CA Top Secret is an instance of a Resource Class (RESCLASS). The types of resources (such as data sets, volumes, terminals, and minidisks) that CA Top Secret protects appear in the Resource Descriptor Table (RDT). The resource types are automatically defined to the RDT at installation. Some resource classes can be used in multiple facilities. Other resource classes are specifically defined for individual environments. Resources are used throughout the system to protect the objects and services provided by jobs, tasks, and sessions in multi-user environments.

Most resource classes are defined as GENERIC, which allows all resources with the same prefix to be protected by the same command. A NONGENERIC definition requires complete specification of each resource by individual commands. Resource classes defined as DEFPROT automatically protect all resources in the class, regardless of whether an explicit command has been issued to establish ownership. If the TOE is not aware of a resource and its existence, it will not protect it. The following table lists the default resource classes.

Interface	Resource Classes	
Base CA Top Secret products	DATASET	PROGRAM
	OPERCMDS	SECLABEL
	TSOAUTH	UNIXPRIV
	FACILITY	IBMFAC
CA Top Secret CICS Interface	OTRAN	TCICSTRN
	FCT	PPT
CA Top Secret IMS Interface		TIMS
	CIMS	IIMS

Table 9-1: Resource Classes Included in the Evaluated Configuration

Note: OTRAN and TIMS are used for transactions. CIMS is used for commands. IIMS is used for PSBs.

Permission can be restricted under the control of the RDT definition:

- **ACTION -** Indicates unusual actions or modes associated with the permission.
- **ACCESS -** Restricts the access levels that are permitted.
- **LIB -** Restricts the program from which the permission is initially executed to a specific data set.
- **PRIVPGM -** Restricts the program from which the permission is initially executed to a specific program.

Each resource in the class has security protection even if it is not defined to the TOE. A security violation occurs when a request is made to access the resource.

A User's or an administrator's access to resources can be restricted by day, time, facility, program, or access level. Resource protection can also be extended on a default or global basis. Most resources, however, do not require full default protection. Resources to the RDT can be defined dynamically in order to assign default protection.

Securing resources is a two-step process. Once the resource type or class is defined in the RDT, then each resource must be:

- Assigned ownership by an individual or department ACID
- Permitted access to additional ACIDs (if necessary)

9.1.3.1.1 Resource Ownership

As aforementioned, resources must first be owned before usage of that resource can be authorized. Ownership of a resource automatically implies full access to that resource. Zone, division, department, profile, and user ACIDs can own resources. For other ACIDs to have access to that resource, they must first be authorized or permitted to do so. Furthermore, only users and profiles can be granted access to resources.

After the resource class is defined to the RDT, the ACIDs that have ownership of the individual resources within that class must be determined. Assigning ownership allows for the individualization of access restrictions for particular resources within the resource class. Resource ownership implies that the owning ACID has an access level of ALL. As it may not be desirable to grant unlimited access to individual users or profiles, resource ownership should be relegated to a Department or Division ACID using the ADDTO command function. At that point, full or restricted resource access can be authorized for other, non-owning ACIDs using the PERMIT command function.

9.1.3.2 Security Validation Algorithm

Once all resources have been defined to CA Top Secret and their access levels have been specified, any future request to access those resources is processed through the TOE's validation algorithm. The Security Validation Algorithm determines whether an ACID has the appropriate authorizations and permissions to access a particular resource.

The security validation algorithm considers factors such as the security mode, resource ownership, the order of permissions, and the use of the installation exit in order to determine which resources (if any) the user is allowed to access. The algorithm determines whether the requested access is granted by searching these records in the following order:

1. The user record
2. Each attached profile in the order they were attached
3. The ALL Record, if included in the search sequence

Whether the algorithm grants access depends on:

- The value selected for the AUTH control option. This value controls how the records are searched, which records are included in the search, and when to stop the search.
- Whether a user has specified the ATTR keyword for RESCLASS.
- Whether the request is for a volume, a data set, or another resource.

If the algorithm locates multiple PERMITs for the resource, the algorithm selects the PERMIT that most closely matches the request.

The security validation algorithm consists of:

- The AUTH control option to determine how the algorithm works.
- (Optional) The ATTR keyword of a resource class to fine-tune the search by specifying how the search is performed for specific resources.

9.1.3.2.1 Volume Request Processing

When validating access to a volume, the Volume Call is invoked first. This call grants or denies access to the volume or data set or allows normal processing to continue. Volume access overrides data set access as shown in the following chart. Before processing continues, CA Top Secret inquires about the records being searched, asking the following questions:

- Does the ACID have the NOVOLCHK attribute?
- Does the ACID own the volume?
- Was an access PERMIT found?
- Were multiple PERMITs found for the requested resource?

A PERMIT is not checked if it has an access level of:

- BLP when the access request is for a DASD volume of a data set
- CREATE, SCRATCH, or CONTROL if the access request is for a tape volume

9.1.3.2.2 Criteria for Volumes

The criteria to determine best fit for volumes are:

- A valid PERMIT with full access to the volume is considered a match.
- A volume PERMIT with an access level of CREATE is considered a match when the request was for data set CREATE (data set checking follows).
- The first violation due to access level mismatch denies access.
- The first violation for any other reason (for example, DAY, TIME, FACILITY, or PRIVPGM mismatches) denies access.

9.1.3.2.3 Data Set Requests

CA Top Secret performs tape data set checking when a DSNAME has been opened for TAPEVOL calls when the TAPE (DSNAME) control option is in effect. When an ACID requests access to a particular DASD data set, the pertinent volume and data set access authorizations must be evaluated depending on what bypass options are associated with that ACID or PERMIT. The bypass options include whether the ACID has NODSN access for that specific request, and whether that ACID has been given the NODSNCHK or NOVOLCHK attributes.

When CA Top Secret is checking both volume and data set level access, CA Top Secret always performs volume level first. In some cases, a request to access a data set is granted or failed strictly based on the ACID's volume access authorizations, without checking whether the user has specific authorization to access that particular data set.

The following table shows how volume access authorizations affect an ACID's request to access a data set on a volume:

Authorized Volume Access	Data Set Read	Data Set Update	Data Set Create	Data Set Scratch
READ	OKAY	DSNAME	FAIL	DSNAME
UPDATE	OKAY	OKAY	FAIL	DSNAME
CREATE	DSNAME	DSNAME	DSN	DSNAME
NOCREATE	DSNAME	DSNAME	FAIL	DSNAME
ALL	OKAY	OKAY	OKAY	OKAY
NONE	FAIL	FAIL	FAIL	FAIL
SCRATCH	DSNAME	DSNAME	FAIL	OKAY
UNDEFINED	DSNAME	DSNAME	FAIL	DSNAME
BLP	BLP	BLP	N/A	N/A
CONTROL	N/A	N/A	CONTROL	N/A
INQUIRE	N/A	N/A	N/A	N/A
SET	N/A	N/A	N/A	N/A

Figure 9-1: Data Set Access Requests on a Volume

The table can only be used for requests where both volume and data set authorizations are checked. For cataloging a tape data set, a CREATE access check occurs only at the data set level. This is independent of the setting of the TAPE control option. No volume check is made, so even having ALL access at the volume level will not allow a tape data set to be catalogued. In addition, when allocating an SMS-managed data set, no volume is identified on the data set call and no volume checking occurs.

If a VOLUME check that was qualified by PRIVPGM, FACILITY, DATE, or TIME denies access on a data set request, this rule of data set authorization relationships are honored, rather than continuing to the DATASET name checking.

Note the following:

- If the user has both CREATE and CONTROL access, data set creation is performed regardless of the data set name.
- A VSAM control (delete and define) would require data set scratch and data set create.

9.1.3.2.4 Criteria for Data Sets

The criteria to determine best fit for data sets are:

- PERMITs using quotes (for example, 'A.B.C.') are considered matches.
- PERMITs without quotes but not starting with a masking character are considered matches.
- PERMITs starting with a mask are considered matches.
- A valid PERMIT with an access level of ALL is considered a match.
- The first violation against a FETCH-protected data set denies access.
- The first access level mismatch denies access.
- The first violation for any other reason (for example, DAY, TIME, FACILITY, or PRIVPGM mismatches) denies access.

9.1.3.2.5 General Resource Requests

When a user requests access to a particular resource, the security validation algorithm determines CA Top Secret access authorizations. When multiple PERMITs are found, the algorithm determines access by best fit, rather than by the most restrictive match. The TOE makes the following inquiries about the records being searched:

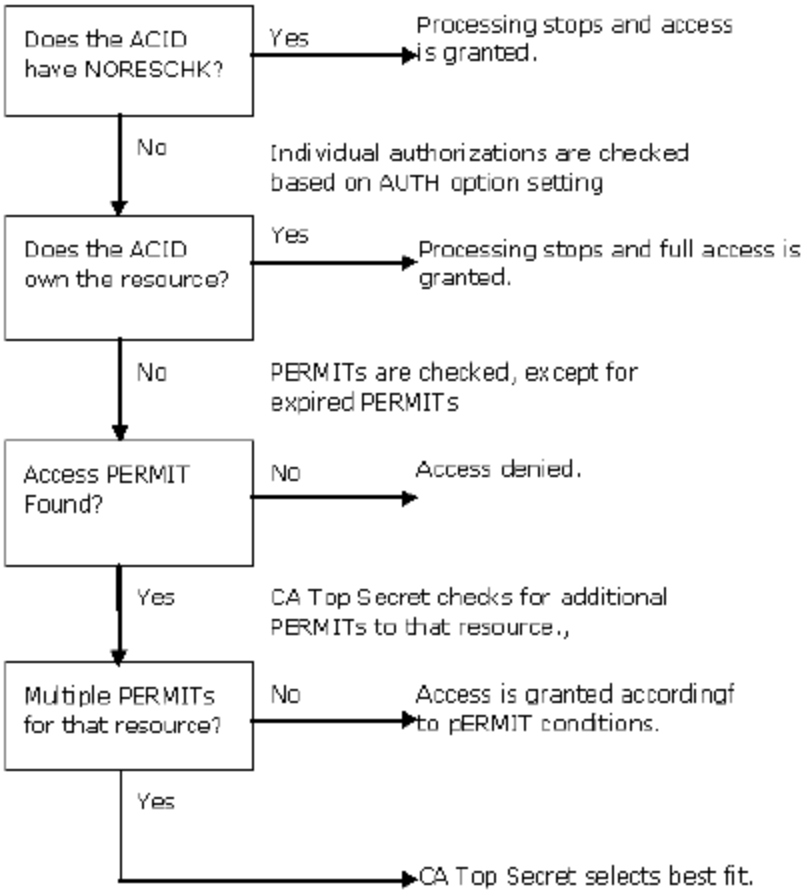

Figure 9-2: General Resource Requests

When the algorithm locates multiple PERMITs during its search, the algorithm selects the PERMIT that most closely matches the request according to a set of criteria.

The following criteria are always the same for all resource types (volumes, data sets, and general resources):

- The first criterion is the longest resource name specified. When the algorithm locates multiple PERMITs, the one containing the longest resource name is considered the best match, unless the PERMIT begins with a masking character.
- If the PERMIT is valid but the access level for the resource is NONE, CA Top Secret denies access.

To determine the longest resource name, each character in the resource name counts as one character whether it is a normal character or a masking character. If the floating mask "-" is used in a PERMIT, only the characters prior to the mask are counted. If the best fit cannot be determined by resource name length or an access level of NONE, then the criteria to determine the best fit differs for volumes, data sets, and general resources.

9.1.3.2.6 General Resources Criteria

The criteria to determine best fit for other resources are:

- The first violation due to access level mismatch denies access.
- The first violation for any other reason (for example, DAY, TIME, FACILITY, or PRIVPGM mismatches) denies access.

9.1.3.3 Discretionary Access Control

The TOE enforces the Discretionary Access Control (DAC) policy to objects based on users, entity, security relevant attributes control option auth, resource class name, entity name, secrec, ownership, facility, time of day, day of week, sysid, Limited Command Facility (LCF), calendar, program, and library.

Note: users can be applications issuing a RACROUTE call. An entity can be a data set, program, or an issued command.

9.1.3.3.1 Secrecs

A secrec is a list of permission entries that contain a resource class, a resource entity, and the highest-level operation allowed by that user. The entity name can be a partial match (e.g. sys.*). The "control options auth" attribute will determine when to stop searching the various secrecs of the user. The Security user has many secrecs:

- **User** – A user secrec is unique to the user. Each user has one user secrec.
- **Profile** – A user has a secrec for each profile assigned to that user. Each user can have one or several profile secrecs.
- **All** – The all secrec is the permissions shared by all users. Each user has one all secrec.

The secrecs are searched according to the chosen mode in order to determine a user's access. The access modes included in the evaluated configuration are:

- **Override allover** – A value of "override allover" for the control options auth attribute will look through each secrec one record at a time until it finds an entry that matches the entity and will compare the permission in that entry to the request permission to determine if access is allowed. Once a matching entity record is found, the search will stop at the end of that secrec and return the best fit (longest length entity name that matches). If there is more than one entry of the same length, it will return the most permissive operation value unless it is defined as "None" or "ActionDenied."
- **Merge allover** – A value of "merge allover" will search every secrec for the best fit and only process the all record (i.e. all secrec) if the search yielded no results.
- **Merge allmerge** – A value of "merge allmerge" will search every secrec including the all record and return the best fit among all secrecs searched.

Table 9-1 shows the stopping point for checking each secrec.

Access Mode			**Secrec**
Override allover	Merge allover	Merge allmerge	
x			User
x			Profile 1
x	x		Profile n
x	x	x	All

Table 9-2: Access Modes

The TOE will permit the requested operation to the protected object (i.e. entity), if the user has a matching secrec entry that matches the class, partial entity name, facility, time of day, day of week, calendar (i.e. day of the year), sysid, program (and library) in the program path, and has an operation in that entry that dominates the requested operation. Furthermore, each secrec that uses LCF will have an inclusion or exclusion list for the command issued. If that command is in the excluded list defined by LCF, the command will fail to execute. If the command was not on an inclusion list, the command will fail

to execute. If the value of facility, time of day, day of week, calendar, sysid, library, program or LCF is not present, then no comparison is done for those attributes.

9.1.3.3.2 LDAP Directory Services

An LDAP directory provides a method to maintain directory information, such as email accounts, in a central location, for storage, update, retrieval, and exchange. LDAP directories can be utilized as network accessible databases for organization and indexing of network security information.

LDS uses the LDAP protocol and native TCP/IP to communicate the changes from the CA Top Secret Security Databases to the remote LDAP repository. Servers enabled with Secure Sockets Layer (SSL) technology protect unauthorized parties from viewing sensitive information during a secure session. Using the XREF mapping record, you configure which LID fields are to be sent to the remote repository and what the remote attribute name is.

When CA Top Secret uses LDS to connect to the remote LDAP directory, it is the client application to the remote LDAP Server. Using the standard LDAP protocol, CA Top Secret formats the add, modify, or delete request and sends it to the remote LDAP Server

9.1.3.4 Mandatory Access Control

The TOE enforces the Mandatory Access Control (MAC) policy to objects based on users, resources, Access Level, Type, Object Security Label, and Subject Security label.

Note: Users can be applications issuing a RACROUTE call. Resources can be data sets, programs, or an issued command.

The type of access granted to a user is based on the values of TYPE and ACCESSLevel, as defined below:

ACCESSLevel	Type=MAC	Type=EQUALMAC	Type=RVRSMAC
Read	User Dominates	Equivalence	Resource Dominates
Read/Write	Equivalence	Equivalence	Equivalence
Write	Resource Dominates	Equivalence	User Dominates

Table 9-3: Mandatory Access Control

For example, if the TYPE = MAC, the following rules apply:

- Read access will be allowed if the security label on the subject dominates the object.
- Read/write access will be allowed if the security label on the subject and object match.
- Write access will be allowed if the security label on the object dominates the user**.**

Note: For all access decisions, the TOE first checks the MAC policy, followed by the DAC policy.

The enforcement of MAC prevents the system from allowing data with higher sensitivity security labels from being disclosed to a user with a lower security label. For instance, a user with a high label cannot send a highly sensitive data set to a user with a lower security label. This is known as "simple security property" and "confinement property." This is also covered within "write-down" protection that is provided by the TOE.

9.1.3.4.1 MAC Label Dominance

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 the TOE 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, the TOE will assign a security label to data when it is created.

When users log on to a system, they can enter a security label. The TOE verifies that they are authorized to use the label by checking their user ACID record. If a user is authorized to use the security label, the TOE maintains the security label in the user's address space and uses it to make access decisions until the user logs off.

Users cannot alter their security label while logged onto the system.

When MLS is active in The TOE, 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.

The TOE 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 The TOE uses to determine the dominance relationship are:

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

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.

9.1.3.4.2 Types of MAC Label Dominance Checks

There are three types of MAC label dominance checks in the 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).

9.1.3.4.3 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.

9.1.3.4.4 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.

9.1.3.4.5 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.

9.1.3.5 ACIDs

By default, the TOE requires that all ACIDs are password protected. A Security Administrator assigns the first password. The user associated with the ACID changes the password immediately (or later) when it expires. An ACID represents many different structures within the TOE, from a USER or a PROFILE to a structure like a Department.

CA Top Secret validates ACIDs in different ways to protect against unauthorized use. First, CA Top Secret checks the security file to determine whether a designated ACID is defined. It does this by seeing if a security record exists for it. Second, if the ACID is undefined, CA Top Secret responds based on the initial control option settings for the security mode and various system options.

An ACID's authorization to access a resource is determined by the PERMITs in:

- The user's record
- The profile records attached to the user's ACID
- The ALL record, which lists globally accessible resources

An ACIDs access can be restricted to:

- A particular terminal or CPU
- Access only on particular days of the week or during certain hours
- Access through a particular facility, such as TSO or CICS

An ACID can be up to eight alphanumeric characters long, which normally corresponds with the user’s system userid. The same ACID can be used for all facilities or a different ACID can be used for each facility (such as TSO, CICS, and z/VM).

CA Top Secret recognizes several different types of ACIDs, ranging from a user to an entire zone. These types comprise the basic hierarchical structure of the CA Top Secret database. Each ACID type is then associated with a set of resource access authorizations.

9.1.3.5.1 Functional ACIDs

Functional ACIDs are associated with organizational units that are defined by organizational ACIDs. This association determines the authorizations that these

functional ACIDs have to interact with objects protected by the TSF which reside in the Operational Environment. The functional ACIDs available are:

- **User ACIDs** - The user is a person. Individuals are associated with user ACIDs or control ACIDs. A user ACID designates a specific employee in a department but can refer to any ACID type (functional or organizational). Every user ACID must be associated with a single department ACID.
- **Profile ACIDs** - When a group of users needs to use a set of identical resources in the same way (the users perform similar or related job functions), define this set of access authorizations once and then associate the entire set with each of the users in the group. In CA Top Secret, this set of common resource access characteristics is termed a profile. Every profile is assigned a unique profile ACID. Once a profile is defined, it can be associated with any number of users (at the same or different levels in the hierarchy), thereby eliminating the need to define each resource access authorization separately for every user. Every profile ACID must be associated with, and defined to, a single department ACID.

9.1.3.5.2 Organizational ACIDs

Organizational ACIDs report to other organizational ACIDs. Organizational ACIDs never report to functional ACIDs. The organizational ACIDs available are:

- **Department ACIDs -** Users typically work for a particular department. CA Top Secret recognizes this logical separation by requiring each user ACID to be associated with one department ACID. Every department is assigned a unique Department ACID. In the evaluated configuration, resources are assigned to a department.

  Note: A Department ACID cannot be directly attached to a Zone ACID. This ACID must be attached to a Division ACID that is attached to a Zone ACID.

- **Division ACIDs –** Multiple divisions can be defined within the security structure. Each division is composed of one or more departments. Every division is assigned a unique division ACID.

  Note: A division can have one or more VCA administrative ACIDs assigned to administer various authorities for other ACIDs assigned to the division.

- **Zone ACIDs - A** zone is used to group two or more divisions. Every zone is assigned a unique zone ACID. Resources can be assigned to a zone. A Department ACID cannot be directly attached to a Zone, but it can be attached to a Division ACID, which is attached to a Zone ACID.

  Note: A zone can have one or more ZCA administrative ACIDs assigned to administer various authorities for other ACIDs assigned to the zone.

- **Control ACIDs – A** Control ACID is an ACID that can be logged on to the system as an administrator. This defines security administrators that are

associated with various structural levels within the CA Top Secret security file. A control ACID can be a regular user of system facilities. A control ACID can issue subsystem commands and perform other functions-such as access data sets and submit jobs. Initially, CA Top Secret knows of only one control ACID—the MSCA ACID—for the security administrator. This ACID is defined during installation. Each type of control ACID performs administrative tasks for the structural level it is associated with. To enable the control ACID to perform these tasks, each one is assigned a scope of authority and administrative authorities within that scope. The TOE, by design, is a hierarchy. Each ACID within a department, division, or zone has its own type of control ACID (MSCA, DCA, VCA ZCA and special types like LSCA and SCA). The C in each of those names means Control.

9.1.3.5.3 Model ACIDs

Model (also referred to as template) ACIDs are used to when creating a large number of new ACIDs. Basic information is copied from the model ACID, where the security administrator has the ability to change, add, or omit any basic information from the model. Basic information can include, but is not limited to, the name associated with the ACID, password, ACID type, facilities permitted access, and associated profiles. An existing ACID can be used as the model or template. In addition, all ACID types can be created.

Model ACIDs support all rules of scope and administrative authority. If the model ACID is outside the scope of the security administrator and/or the model ACID has an information field that requires ADMIN authority the security administrator does not have, the model ACID cannot be used to create the new ACID.

9.1.3.5.4 Pre-Defined ACIDs

CA Top Secret has reserved or special ACIDs and tables that are pre-defined and maintain resource and attribute information. These include:

- **ALL Record -** Identifies resources that are globally accessible to all signed on users.
- **APPCLU Record -** Stores the names and security requirements of the logical units (LUs) involved in APPC conversations.
- **Audit Record -** Stores the resource names that are to be audited.
- **Data Lookaside Table (DLF) -** Controls the loading of selected data sets into ESA hyperspace by selected jobs. With the proper authority and keywords, CA Top Secret can identify and control those data sets and jobs valid for DLF.
- **Delegate Record -** Contains delegate resource definitions. Each definition specifies a resource class and entity name used in nested ACEE processing.

- **Field Descriptor Table (FDT) -** Defines fields (classes) that can be attached to ACIDs within the Security File. Each field description contains a field name, field code, and field attributes.
- **MLS -** Contains SECLABEL, CATEGORY, and SECLEVEL records, which are the hierarchical elements of multi-level security.
- **Node Descriptor Table (NDT) -** Contains all CPF, LDAP, LINUX, and PassTicket application and session key-related node information. The NDT is a global record similar to the Resource Descriptor and Field Descriptor Tables.
- **Resource Descriptor Table (RDT) -** Contains pre-defined resource classes. Each resource class is identified by a unique keyword and has certain attributes associated with it.
- **Started Task Table (STC) -** Stores all started task procedure names and the ACIDs associated with them. CA Top Secret offers security protection for all required STC definitions or only for STCs that reference sensitive data or affect system integrity.
- **Static Data Table (SDT) -** Contains record elements that Administrators can use to control which users have access to certain resources.

9.1.3.6 Scope

The Security Administrator is responsible for the scope of authority. The TOE provides several different levels of control ACID scope. Each level corresponds to a level in the corporate structure. For example, a division control ACID (VCA) is responsible for administering security for all the ACIDs within a particular division (including ACIDs assigned to departments associated with that division). The following table shows an example of how security administrators can be defined, and the scope that result:

Title	Scope	Example
MSCA	Entire installation	The master SCA (MSCA) can create all CA Top Secret administrators, including SCAs, LSCAs, ZCAs, DCAs, VCAs, and auditors.
SCA	Entire installation	An SCA's scope of authority depends on the administrative authorities that they were granted. An SCA can create ZCAs, DCAs, VCAs, Profile, and User ACIDs, but not other SCAs.
LSCA	A zone and/or another LSCA	An LSCA can have all the authority of an SCA, but unlike the SCA, the LSCA must have a scope of authority assigned to it. This scope of authority can be

		one or more LSCAs and/or zones.
ZCA	A zone	Permit access to resources owned by his zone, all connected divisions, departments and users within that zone.
		Define profiles and perform maintenance for ACIDs that are within his scope.
		Create ACIDs in his zone.
		Permit ACIDs in other zones to access his zone's resources, but cannot perform maintenance for ACIDs in other zones.
VCA	A division	Permit access to resources owned by his division, all departments and users within that division, and can define profiles and perform maintenance for ACIDs that are within his scope.
		Create ACIDs in his division.
		Permit ACIDs in other divisions to access his division's resources, but cannot perform maintenance for ACIDs in other divisions.
DCA	A department	Department administrators have the same scope over a department that a VCA has over a division. DCAs can also create ACIDs in their department.

Table 9-4: Security Administrators and Associated Scope of Authority

For more information on the types of Security Administrators, see Section 9.1.4.1.

9.1.3.7 Authority

In addition to scope of authority, each Security Administrator must also be assigned particular types of administrative authorities. These authorities define the security functions the control ACIDs can perform for ACIDs within their scope. Upper level security administrators can grant administrative authorities to lower level administrators within their scope, provided the higher-level administrators already possess the appropriate authorities.

9.1.3.7.1 Types of Administrator Authorities

An ACID's authority determines what can be done with the administration of ACIDs, resources, facilities, and the display of security file information. The administrative authorities are:

- ACID
- DATA

- RESOURCE
- FACILITY
- MISC1
- MISC2
- MISC3
- MISC4
- MISC5
- MISC7
- MISC8
- MISC9
- SCOPE

Each type of authority approximately corresponds to a different set of security environment control and maintenance functions (for example, ACID maintenance or resource maintenance).

A group of operands is associated with each type of authority. Each operand designates a specific functional authority. For example, ACID(CREATE) authority lets the control ACID create and delete ACIDs within their scope, while RESOURCE(INFO) lets the control ACID perform certain inquiries for any resource within their scope.

Note: Administrative authorities cannot be assigned to a zone, division, department, group or profile ACID.

9.1.3.7.2 Global Authorities

To give every ACID the ability to perform specified administrative functions, the administrator can assign the administrative authority to the ALL record. For example, assigning MISC1(LTIME) to the ALL record gives all ACIDs the authority to set their own terminal lock time interval. The ALL record can also contain resources access levels.

Only administrators with MISC9(GLOBAL) authority can assign administrative authorities to the ALL Record.

9.1.3.8 Facilities

A facility is a way of grouping options associated with a particular service that users sign on to. To sign on to a service, a user must have access to the facility. Only the MSCA can access any facility by default. All other TOE users must be authorized to access one or more facilities.

In the evaluated configuration, the TOE provides security for the following facilities:

- CICS – Associated with the appropriate facility in the following ways:
    - Explicitly - By assigning a MASTFAC parameter to the ACID executing the batch job or STC that created the service.
    - Implicitly - CA Top Secret looks at the program in control and matches it with the program name specified in the facility. If there is not a match for the INITPGM or no MASTFAC was assigned, a facility of STC is assigned if the particular service was started as a started task. If the service was started as a batch job, the facility of BATCH is assigned.
    - Multiple user address space - each user that signs on to a CICS region is given part of the region's block of space. When a user requests access, standard CICS only identifies to z/OS the Security Record for the region involved, not the specific user's Security Record. This procedure is typical of multiple user address space systems.
- IMS - Associated with the appropriate facility in the following ways:
    - Explicitly - By assigning a MASTFAC parameter to the ACID executing the batch job or STC that created the service.
    - Implicitly - CA Top Secret looks at the program in control and matches it with the program name specified in the facility. If there is not a match for the INITPGM or no MASTFAC was assigned, a facility of STC is assigned if the particular service was started as a started task. If the service was started as a batch job, the facility of BATCH is assigned.
    - Multiple user address space
- TSO - Single user address space. Each signed on user gets his own address space.

Note that storage keys protect the security data area in the user address space. User programs are normally allowed to only modify storage key 8 while security data is kept under either key 0 or key 3. These two storage keys require programs to be in that key to modify them. These programs must be APF authorized and running in a supervisor state. User programs are not allowed to receive this authorization.

Considerations for implementing CA Top Secret for CICS and IMS are similar because the security interfaces behave in a similar fashion. The transactions and resources available within CICS and IMS are similar in that the resources available in TSO, but have different names. This table shows the relationship among the resources appropriate to each facility.

TSO	CICS	IMS
Commands	Transactions	Transactions
Programs	PPTs	PSBs

Data Sets	FCTs, DCTs, JCTs, TSTs	DBDs

Table 9-5: Facility Resources

9.1.3.9 Access Restrictions

9.1.3.9.1 Time of Use Monitoring

Administrators can control when an ACID can enter the system through a facility or device by specifying the following time-related parameters:

- **DAYS -** Specifies the day of the week.
- **TIMES** - Specifies the time of day.
- **FOR/UNTIL -** Specifies a particular length of time.
- **CALENDAR -** Specifies an inclusive or exclusive list of dates, which gives users the ability to establish calendars to reflect site-specific events (such as company holidays). The CALENDAR specified does not need to exist to do the permission; however, the PERMIT does not execute correctly until the named record exists in the Static Data Table (SDT).
- **TIMEREC** - Specifies multiple time intervals within a 24-hour day. Administrators can specify multiple 15-minute intervals within a 24-hour day when access to a resource is permitted. The TIMEREC specified does not need to exist to do the permission; however, the PERMIT does not execute correctly until the named record exists in the SDT

Note: The DAYS and CALENDAR keywords are mutually exclusive. Any existing permissions that use the DAYS keyword can be replaced with a permission that uses the CALENDAR keyword. Similarly, the TIMES and TIMEREC keywords are mutually exclusive. Permissions that use the TIMES keyword can be replaced with a permission that uses the TIMEREC keyword.

9.1.3.9.2 Restrictions by Device

CA Top Secret can control system entry by restricting access to:

- CPUs
- Online terminals (TCAM, VTAM, BTAM, VM local, logical, and bi-synch)
- Remote (RJE) and local JES readers
- Internal readers
- NJE nodes

The specific resource class name (CPU, TERMINAL, VMRDR, and NODES) is used as the keyword to determine access authorizations and restrictions for that resource.

9.1.3.9.3 Multiple Node Users

Only users defined to the target node can initiate jobs on that node. To define users to multiple nodes, administrators must adhere to the following requirements:

- If CA Top Secret is running on all nodes and each node has its own security file, the Command Propagation Facility (CPF) is used to propagate security information for a user across all nodes while only entering it once.
- If CA Top Secret is running on all nodes that share the same security file, then each user is implicitly defined to every other node in the system.

9.1.3.10 Security Modes

CA Top Secret for z/OS r14 uses several security modes to assist in access control. However, only Fail mode is included in the evaluated configuration. During Fail mode, the TOE is in full control of access requests. All users must be defined and resources protected by being owned or by DEFPROT protection. All unauthorized access requests fail by default. For more information on the excluded security modes, see Section 2.3.

9.1.3.11 Object Reuse Protection

A z/OS 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 an object such as data set, that object is physically 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. A 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.

The AUTOERASE and AUTOEDSN control options govern whether automatic erase is active in a CA Top Secret environment and, if so – the specific data set names/data set name prefixes that are eligible for automatic erase processing. In the evaluated configuration, both these control options are enabled.

CA Top Secret itself does not perform the actual physical data erasure. When a user makes a request to delete a data set, it results in a request to the access method in the Operational Environment to perform the data set delete. Deletion of a data set entails removal of the data set entry in the volume table of contents (VTOC) on one or more volumes (if multi-volume) and optionally the related catalog. The actual data contents themselves remain largely intact.

When a user issues a delete for a data set, the platform will issue an access method service to perform the actual delete. The access method routines will then issue a SAF security call to ascertain the user's authority to delete the data set. If the user is permitted

to perform the delete, the access method will go about the task of data set deletion. If the user is denied, the data set will not be deleted.

The access method passes an indicator, STATUS=ERASE, on the RACROUTE REQUEST=AUTH request which interrogates the external security manager (ESM) for erase characteristics of that to-be-deleted data set. The ESM responds to the access method through a reason code when the access is allowed.

A returned reason code value of zero means that the data set should not be erased; a reason code value of four means that the data set should be erased. The TOE will set reason code zero or four based on the access determination and the settings of the AUTOERASE and AUTOEDSN control options in relation to the data set object being deleted. The access method itself is responsible for performing the actual data erasure when reason code=four is returned.

9.1.4 Security Management

The TOE maintains three roles – security administrators, scoped security administrators, and users. Administrators manage the TOE and its users; whereas a user's primary function is to perform work. Any administrator with ACID(CREATE) administrative authority can establish users. The table below describes what functions a user can perform.

AccessLevel (DAC)	DIRAUTHLevel (MAC)
Read	Read
Create	Read
Write	Write
Control	ReadWrite
Update	ReadWrite
Scratch	ReadWrite
Fetch	ReadWrite
Alter	ReadWrite

Table 9-6: User Performed Operations on the TOE

Administrators can perform the following functions:

- Define default subject security attributes (nothing is set by default, therefore default is NONE)
- Change subject security attributes
- Manage user identities
- Manage the authentication data by an administrator
- Control the authentication data that users are allowed to manage
- Manage the authentication data (i.e., passwords) by the associated user
- Manage the metric used to verify and generate the secrets

- Define additional security attributes for users
- Manage the threshold for unsuccessful authentication attempts
- Manage the actions to be taken in the event of an authentication failure
- Manage the attributes used to make explicit access or denial based decisions
- Manage the rights to modify the audit events
- Delete the group with read access to the audit records
- Modify the group with read access to the audit records
- Add the group with read access to the audit records

The types of Administrators are described in the following sections.

9.1.4.1 Security Administrator

The security administrator:

- Is responsible for implementing and maintaining system security
- Defines users, resources, access levels, and facilities
- Controls the security environment by using control options and TSS commands
- Monitors resource access violations with the auditing, tracking, and reporting options

The security administrator's role is determined by:

- The scope of authority (the entire installation or a single department within that installation)
- The designated administrative authorities (create ACIDs, run reports, change control options)

Note: Each security administrator must first possess the appropriate administrative authority.

9.1.4.2 Types of Security Administrators

The CA Top Secret administrative hierarchy has seven levels. The first six levels represent ACIDs whose primary function is to control security administration.

- **Master Security Control ACID (MSCA) -** Referred to as the Master Central Security Administrator. There can be only a single MSCA. The MSCA's ACID is pre-defined; it exists as soon as the Security File is created through TSSMAINT.

The MSCA's ACID cannot be deleted, although it can be renamed. An MSCA has unlimited scope. Only an MSCA has implicit unlimited administrative authority. Only the MSCA can create SCAs. The MSCA can log on or initiate with only password checking in force; no expiration, facility, source, or terminal checking is performed by CA Top Secret.

- **Central Security Control ACID (SCA) -** Referred to as the Central Security Administrator. An SCA is not associated with any specific zone division or department but has unlimited scope. An SCA, with this authority, can do almost everything except define another SCA.
- **Limit Central Security Control ACID (LSCA) -** An LSCA is not associated with any specific zone, division, or department. It has the same capabilities as an SCA but rules of scope checking apply.
- **Zonal Control ACID (ZCA) -** Only a central security administrator can establish zonal administrators. Each ZCA is associated with a particular zone. A ZCA can perform the following activities:
    - administrative tasks for the **divisions** linked to this zone
    - administrative tasks for the **departments** linked to this zone
    - administrative tasks for the **users** linked to this zone
    - administrative tasks for the **profiles** linked to this zone

A zone may have several ZCAs, or under a centralized administrative system, no ZCAs. In the latter case, a central security administrator must perform the administrative requirements for this zone.

- **Divisional Control ACID (VCA) -** A central security administrator can establish divisional administrators. Each VCA is associated with a particular division. A VCA can perform the following activities:
    - administrative tasks for the **departments** linked to this division
    - administrative tasks for the **users** linked to this division
    - administrative tasks for the **profiles** linked to this division

A division may have several VCAs, or under a centralized administrative system, no VCAs. In the latter case, central security administrator or ZCA must perform the administrative requirements for this division.

- **Departmental Control ACID (DCA) -** Departmental administrators can be established by a central security administrator or a VCA for a department that is linked to that VCA's division. The responsibilities that can be performed by a DCA include the following activities:
    - administrative tasks for the **users** that belong to this department
    - administrative tasks for the **profiles** that belong to this department

A department may have several DCAs or no DCAs. In the latter case the administrative requirements for this department have to be performed by either a central security administrator or the appropriate ZCA or VCA.

9.1.4.3 Auditor

As with any other type of administrator, an auditor's scope is a function of the TYPE that was designated at ACID creation, for example, TYPE(VCA). A central auditor is defined as an SCA; a divisional auditor is defined as a VCA.

9.1.4.4 Command Functions

Command functions are the primary tool of the security administrator. A command function lets an administrator define ACIDs, assign attributes, and determine resource access. Commands have the following syntax:

TSS FUNCTION(*acid*|STC|AUDIT|RDT|FDT|DLF|ALL|NDT|SDT)
KEYWORD(OPERAND)

The following definitions apply for command functions:

- TSS – CA Top Secret commands must always begin with "TSS."
- FUNCTION – Specifies the name of the function CA Top Secret will perform. The following rules apply:
    - The function must immediately follow "TSS."
    - Only one function can be entered per TSS command.
    - One or more spaces must be entered between TSS and the function.
- (*acid*|STC|AUDIT|RDT|FDT|DLF|ALL|NDT|SDT) – Specifies the ACID or record the function will affect.

- KEYWORD - Specifies the resource type or security attribute the function is processing. Keywords can be entered:
    - In any order
    - Online: Keywords can be entered from line to line without special action.
    - In batch: The last keyword must follow on a continuing line with a blank and a dash. The next keyword can be entered on the next input line.
- (OPERAND) - Specifies the prefix, resource name, or the required value for a security attribute. Operands must be provided, and parentheses must be provided to indicate no value. If an operand is missing, any following keyword is ignored.

CA Top Secret functions can be entered free form onto the command screen of an online terminal or into any of the CA Top Secret full-screen administration panels.

CA Top Secret provides online status information on the security environment and the ACIDs and resources within that environment. The informational TSS commands are:

- **TSS LIST -** Displays information on individual or reserved ACIDs (such as the RDT).
- **TSS WHOHAS -** Displays information on a resource, field name (FDT), facility, attribute or administrative authority, depending on which is specified.
- **TSS WHOOWNS -** Displays information on whether a resource is defined to CA Top Secret and who is the owner.
- **TSS WHOAMI -** Displays information on the security environment for the ACID currently signed on to the terminal.
- **TSS MODIFY STATUS -** Displays information on the global security environment and control options currently in effect.

9.1.4.5 Command Propagation Facility

In addition to propagating changes, the Command Propagation Facility (CPF) allows administrators to view the contents of the CA Top Secret Security Database from other nodes. The viewing is completely secure since the administrator's scope is verified at both locations, allowing the administrator to review the security information for which he or she is responsible for at all nodes in the CPF domain.

CPF propagated administration executes on the remote system using the authority of the administrator as defined in the remote system, and not using the authority and scope of the administrator from the originating system. For example, if a security administrator on one system propagates a CA Top Secret command to a system where that administrator is not defined as a security administrator, then the command is limited to the non-security administrator authority. It should also be noted in regards to data propagation that user-initiated password changes at system entry that are propagated using CPF cause the user's password to change at each node where the change is sent.

9.1.5 TOE Access

CA Top Secret has the ability to allow or deny session establishment based upon certain configured settings. Access can be allowed or denied depending on the time or date, source, or APPLID. Additionally, an attempt to establish a session can be denied if the user account has been suspended.

Suspension of an account can occur for several reasons. The first being inactivity of the account. Users with the appropriate privilege can configure the inactivity period length through TSS MODIFY INACTIVE (1-255). This determines the number of days an account must be inactive before being suspended. A user with the appropriate privileges can suspend ACIDs until manually removed or for a limited time. An ACID will also get suspended if the password violation threshold is reached.

For the time or date setting, session establishment can be denied for instances such as holidays, weekends, or after hours. For instance, with proper configuration, the TOE can deny session establishment for Friday and Saturday or after the workday has ended.

The source and APPLID can be denied as well to ensure that the TOE only allows communication and sessions from those locations it trusts. This can be by a specific terminal ID, IP address, Point of Entry, or from the application by which a user is trying to authenticate.

9.2 Self Protection (ADV_ARC.1)

The TOE forces users/administrators to authenticate to it prior to allowing them to take any actions on objects which the TOE protects within the operational environment. This includes the TOE's configuration files and processes. The TOE also forces the users/administrators to authenticate in a manner consistent with administrative defined policies. These policies include the authentication mechanisms that need to be utilized and any access restrictions that need to be placed against their ability to authenticate. The TOE will also suspend the ability of users/administrators to successfully authenticate to the TOE due to a consecutive number of failed authentication attempts based on ACID. This assists in protecting the TOE from a potential unauthorized user. In addition, the TOE enforces strong secrets according to administrative defined policies, which further prevents unauthorized users from successfully authenticating to the TOE.

Once users/administrators are authenticated, the TOE maintains individual sessions associated with them through the creation of a security environment, which is associated with their ACID. Each security environment defines the authorizations that the associated user/administrator has to perform operations on objects. The TOE protects these security environments such that a process of another user/administrator cannot affect the contents of each individual security environment.

The TOE monitors all requests from subjects within its operational environment, including requests from users/administrators, OS processes, and system processes. The TOE determines which requests will be authorized to be performed. Therefore, the TOE has the ability to monitor actions by subjects external to itself on its own configuration files and processes. The TOE's access control policies only allow those actions which are defined to be actions from a trusted subject.

The TOE also ensures that information critical for the security of subjects is protected via the operational environment's cryptographic mechanisms. The TOE will call the z/OS ICSF module when the TOE determines that security relevant information should be encrypted before being stored in the operational environment or sent outside the TOE's security perimeter.

The TOE records log data in two potential locations. SMF, which is a system level file, stores the information and receives security protection preventing deletion access. The second file, Audit and Tracking file (or ATF), is a file that is attached to the TOE's address space. The file is in constant use and is also protected from deletion access. Additionally, the data collected cannot be removed without being overwritten. This requires system knowledge and the significant level of access.

9.3 TOE Summary Specification Rationale

This section identifies the security functions provided by the TOE mapped to the security functional requirement components contained in this ST. This mapping is provided in the following table.

Security Function	Security Functional Components
Security Audit (FAU)	FAU_GEN.1 Audit data generation
	FAU_GEN.2 User identity association
	FAU_SAR.1 Audit review
	FAU_SAR.2 Restricted audit review
	FAU_SEL.1 Selective audit
User Data Protection (FDP)	FDP_ACC.2(1) Complete access control
	FDP_ACC.2(2) Complete access control
	FDP_ACF.1(1) Security attribute based access control
	FDP_ACF.1(2) Security attribute based access control
Identification and Authentication (FIA)	FIA_AFL.1 Authentication failure handling
	FIA_ATD.1 User attribute definition
	FIA_SOS.1(1) Verification of secrets
	FIA_SOS.1(2) Verification of secrets
	FIA_SOS.2 TSF generation of secrets
	FIA_UAU.2 User authentication before any action
	FIA_UAU.4 Single-use authentication mechanisms
	FIA_UAU.5 Multiple authentication mechanisms
	FIA_UID.2 User identification before any action
	FIA_USB.1 User-subject binding
Security Management (FMT)	FMT_MOF.1(1) Management of security functions behavior
	FMT_MOF.1(2) Management of security functions behavior
	FMT_MOF.1(3) Management of security functions behavior
	FMT_MOF.1(4) Management of security functions behavior
	FMT_MSA.1 Management of security attributes
	FMT_MSA.3 Static attribute initialization
	FMT_SMF.1 Specification of management functions
	FMT_SMR.1 Security roles

Security Function	Security Functional Components
TOE Access (FTA)	FTA_TSE.1 TOE session establishment

Table 9-7: Security Functional Components

9.3.1 Security Audit

The audit function of the TOE enforces the FAU_GEN.1, FAU_GEN.2, FAU_SAR.1, FAU_SAR.2, FAU_SAR.3, and FAU_SEL.1 requirements.

Section 9.1.1 details how the TOE collects security and system audit information. Administrators with proper privileges are able to monitor, alert, and report information about user activity across platforms.

The examination of the TSS showed that each of these requirements were successfully mapped to the SFRs listed above the information provided in the ST introduction.

The generation of audit records (FAU_GEN.1.1) is provided in Section 2.5.1 as well as in the TSS, Section 9.1.1. In addition to the generation of audits, the AUDIT privilege is discussed in section 9.1.1.1 to explain the use of the privilege in order to view the audited information. FAU_GEN.1.2 is then fulfilled in Section 9.1.1.2 with the mapping of information audited in relation to the event that is occurring. Section 2.5.1 of the introduction covers this information as well but in less detail.

FAU_SAR.1 and FAU_SAR.2 are covered in the TSS, Sections 9.1.1.1 and 9.1.1.2. These sections discuss the types of reports/logs that are provided in the TOE as well as the AUDIT privilege. These sections demonstrate the use of scoping (the AUDIT privilege) to apply restrictions on auditing. Finally, FAU_SEL.1 is covered in Section 9.1.1 with the discussion of the customization of reports in relation to the preferences of the users on the system.

9.3.2 User Data Protection

The User Data Protection function of the TOE enforces the FDP_ACC.2(1), FDP_ACC.2(2), FDP_ACF.1(1), and FDP_ACF.1(2) requirements.

MLS is a security policy in CA Top Secret that provides discretionary access control (DAC) protection mechanisms and includes mandatory access control (MAC). MLS is an optional layer of protection on top of DAC, which forces security classifications, called security labels, for virtually all users, data and resources in a system. Additionally, it validates all access based on these labels, regardless of permissions and ownership. MLS offers selective protection of data and resources based on a user's organization's individual needs. CA Top Secret lets a user activate MLS and implement security labels for the users, resources, and data that require a higher level of security.

MAC imposes a security policy based on security labels. Security labels classify users, data, and resources. Standard 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.

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.

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.

9.3.3 Identification and Authentication

The identification and authentication function of the TOE enforces the FIA_AFL.1, FIA_ATD.1, FIA_SOS.1(1), FIA_SOS.1(2), FIA_SOS.2, FIA_UAU.2, FIA_UAU.4, FIA_UAU.5, FIA_UID.2, and FIA_USB.1 requirements.

CA Top Secret uses the ACID record to verify a user's system access and privileges. Section 2.5.2 of the introduction discusses the basic overview of the Identification and Authentication requirements and is covered in more detail in the sections of the TSS discussed below.

Section 9.1.2 discusses the primary of attributes for users with the use of the security file. The security file is responsible for storing the security attributes which are relevant to all users of the TOE. This information supports the FIA_ATD.1 requirement.

FIA_SOS.1(1) is fulfilled in Sections 9.1.2.5 and 9.1.2.5.1 with the discussion of the password policy and requirements for password defaults. FIA_SOS.1(2) is then fulfilled in Section 9.1.2.5.2 with the policies relating to a user's passphrase.

FIA_AFL.1.1 and FIA_AFL.1.2 is fulfilled by Section 9.1.2.5 with the information regarding the suspension of user accounts when too many attempts are made to connect to the TOE with invalid information.

FIA_UID.2, FIA_USB.1, and FIA_UAU.2 are fulfilled in Section 9.1.2 with the discussion of authentication before any action and authentication methods provided by

the TOE which are chosen by the application. Additionally, the TSS speaks at length about a user being associated with the attributes attached to their roles.

Section 9.1.2.3 fulfills the FIA_UAU.4 requirement where it speaks to passticket authentication. FIA_UAU.5 is fulfilled with Sections 9.1.2.1/2/3/4/5/7/8/9. These sections cover the five methods of authentication that can be used when accessing the TOE. These methods include passwords, passphrases, certificates, passtickets, and Kerberos.

9.3.4 Security Management

The security management function of the TOE enforces the FMT_MOF.1(1), FMT_MOF.1(2), FMT_MOF.1(3), FMT_MOF.1(4), FMT_MSA.3, FMT_SMF.1, and FMT_SMR.1 requirements.

Security management is required to manage the users, groups and the privileges of users. This is supporting identification and authentication as well as access control. Different aspects of security management support each other. For example, user and group management supports the management of access control, because the definition of access rights can be simplified by defining access on a group level and assign users that require access to the appropriate groups. Security management also supports auditing because it allows to define the events to be audited based on individual users, individual protected objects, privileges of the users, type of event, and (in LSPP mode) security label.

In addition, the security management of the audit data (especially dumping the SMF data sets when they get full) also supports audit. Security management also includes the management of access rights including (in LSPP mode) the definition of the security labels and the definition how they get printed on a printer that supports multiple labels. The management of discretionary access rights can be performed by users with the required privileges and the management of those privileges is part of the user and group management. This structure allows delegation of some management functions to users with privileges limited to the scope of a group. Security management also supports communication security by providing the ability to configure the different protection mechanisms SSL/TLS, IPSec, SSH, Kerberos, and ATTLS.

Section 2.5.3 of the introduction discusses a general overview of the Security Management requirements as shown above and is expounded upon in the TSS.

The security management requirements outlined by the TOE are covered by the TSS in Section 9.1.1.4. The main paragraph of this section identifies the division of roles into Security Administrators Scoped Security Administrators, and Users. These roles can be scoped based on privileges and allow for variations of off the default roles. This applies to FMT_MOF.1(1), FMT_MOF.1(2), FMT_MOF.1(3), and FMT_MOF.1(4). These SFRs apply to the security administrator, user, and scoped security administrator. Section 9.1.4.3, Scope, also addresses the FMT_MOF.1 requirements by detailing how scope is used to enforce different levels of privileges for users of the TOE.

Following FMT_MOF.1, the requirements for FMT_MSA.1 are fulfilled in Section 9.1.4.1 of the TSS with discussion of the privileges for the previously discussed roles. FMT_MSA.3 is then fulfilled by Section 9.1.3.2 with the discussion of the default values and how they can be changed on the TOE.

FMT_SMF.1 is fulfilled again by Section 9.1.4.1 as well as with Section 9.1.4.4. 9.1.4.4 discusses the types of roles provided to users and what privileges apply to those roles. These can then be mapped to Section 9.1.4.1 to further detail what these roles can perform on the TOE. Finally, FMT_SMR.1 is covered by the same two sections.

9.3.5 TOE Access

The TOE access function of the TOE enforces the FTA_TSE.1 requirement.

This requirement is fulfilled by Section 2.5.5 of the introduction by discussing the denial of a session based on a user's status or failure to authenticate correctly. Section 9.1.5 and 9.1.5.1 further detail this information with a more in-depth look at the suspension of a user account when on vacation or when the authentication attempt limit is surpassed.

10 Security Problem Definition Rationale

10.1 Security Objectives Rationale

The following table provides a mapping with rationale to identify the security objectives that address the stated assumptions and threats.

Assumption	Objective	Rationale
A.ADMIN One or more authorized administrators will be assigned to install, configure and manage the TOE and the security of the information it contains.	OE.ADMIN One or more authorized administrators will be assigned to install, configure and manage the TOE and the security of the information it contains.	OE.ADMIN maps to A.ADMIN in order to ensure that authorized administrators install, manage and operate the TOE in a manner that maintains its security objectives.
A.PATCHES Administrators exercise due diligence to update the TOE with the latest patches and patch the Operational Environment (e.g., OS and database) so they are not susceptible to network attacks.	OE.ADMIN One or more authorized administrators will be assigned to configure the Operational Environment, and install, configure, and manage the TOE and the security of the information it contains.	OE.ADMIN maps to A.PATCHES in order to ensure that the authorized administrators properly patches the Operational Environment and the TOE in a manner that maintains the security objectives of the TOE.
A.NOEVIL Administrators of the TOE are not careless, wilfully negligent, or hostile and will follow and abide by the instructions provided by the organization's user guidance documentation.	OE.NOEVIL All administrators of the TOE are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided by the organization's user guidance documentation.	OE.NOEVIL maps to A.NOEVIL in order to ensure that there are no careless, willfully negligent, or hostile administrators of the TOE.
A.LOCATE The TOE will be located within controlled access facilities that will prevent unauthorized physical access.	OE.LOCATE The TOE will be located within controlled access facilities that will prevent unauthorized physical access.	OE.LOCATE maps to A.LOCATE in order to ensure that physical security is provided in the environment where the TOE operates.

Table 10-1: Assumption to Objective Mapping

Threat	Objective	Rationale
T.ACCESS Unauthorized users could gain access to objects protected by the TOE that they are not authorized to access.	O.ACCESS The TOE will provide measures to authorize users to access objects protected by the TOE once the user has been authenticated. User authorization is based on access rights configured by the authorized users of the TOE.	O.ACCESS (FDP_ACC.2(1), FDP_ACC.2(2), FDP_ACF.1(1), FDP_ACF.1(2)) addresses T.ACCESS by providing the authorized users with the capability to specify access restrictions on the objects protected by the TOE to authorized users.
	OE. EAVESDROPPING The Operational Environment will encrypt TSF data when called by the TOE to prevent malicious users from gaining unauthorized access to TOE data.	OE.EAVESDROPPING addresses T.ACCESS by ensuring that the underlying Operating System provides the capability to encrypt TSF data used by the TOE.
T.ADMIN_ERROR An administrator may incorrectly install or configure the TOE, or install a corrupted TOE resulting in ineffective security mechanisms.	O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management.	O.ROBUST_ADMIN_GUIDANCE (ALC_DEL.1, AGD_PRE.1, AGD_OPE.1) helps to mitigate T.ADMIN_ERROR by ensuring the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner and to provide the administrator with instructions to ensure the TOE was not corrupted during the delivery process. Having this guidance helps to reduce the mistakes that an administrator might make that could cause the TOE to be configured in a way that is unsecure.
	O.MANAGE The TOE will provide authorized administrators with the resources to manage and monitor user accounts, objects, and security information relative to the TOE.	O.MANAGE (FMT_MOF.1(1), FMT_MOF.1(2), FMT_MOF.1(3), FMT_MOF.1(4), FMT_MSA.1, FMT_MSA.3, FMT_SMR.1, FMT_SMF.1) addresses T.ADMIN_ERROR by ensuring only authorized administrators can use the provided resources for managing and monitoring user accounts, objects, and security information relative to the TOE.

Threat	Objective	Rationale
T.AUDIT_COMPROMISE A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.	O.ACCESS The TOE will provide measures to authorize users to access objects protected by the TOE once the user has been authenticated. User authorization is based on access rights configured by the authorized users of the TOE.	O.ACCESS (FDP_ACC.2(1), FDP_ACC.2(2), FDP_ACF.1(1), FDP_ACF.1(2)) addresses T.AUDIT_COMPROMISE by providing the authorized users with the capability to specify access restrictions on the objects protected by the TOE.
T.MASK Users whether they be malicious or non-malicious, could gain unauthorized access to the TOE by bypassing identification and authentication countermeasures.	O.AUDIT The TOE will provide measures for recording security relevant events that will assist the authorized users in detecting misuse of the TOE and/or its security features that would compromise the integrity of the TOE and violate the security objectives of the TOE.	O.AUDIT (FAU_GEN.1, FAU_GEN.2, FAU_SAR.1, FAU_SAR.2, FAU_SEL.1) addresses T.MASK by providing the authorized users with tools necessary to monitor user activity to ensure that misuse of the TOE does not occur.
	OE.SYSTIME The operating environment will provide reliable system time.	OE.SYSTIME helps to mitigate T.MASK by ensuring the accuracy of the tools necessary to monitor user activity as provided via O.AUDIT.
	O.ROBUST_TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate.	O.ROBUST_TOE_ACCESS (FIA_AFL.1, FIA_UAU.2, FIA_UAU.4, FIA_UAU.5, FIA_ACID.2, FTA_TSE.1), addresses T.MASK by controlling the logical access to the TOE and the objects the TOE protects. By constraining how and when authorized users can access the TOE, and by mandating the type and strength of the authentication scheme, this objective helps mitigate the possibility of a user attempting to login and masquerade as an authorized user. In addition, this objective provides the administrator the means to control the number of failed login attempts a user can generate before an account is suspended, further reducing the possibility of a user gaining unauthorized access to the TOE.
	O.AUTH The TOE will provide measures to uniquely identify all users and administrators. The TOE will authenticate the claimed	O.AUTH (FIA_ATD.1, FIA_SOS.1(1), FIA_SOS.1(2), FIA_SOS.2, FIA_UAU.2, FIA_UAU.4, FIA_UAU.5,

Threat	Objective	Rationale
	identity prior to granting a user or administrator access the objects protected by the TOE.	FIA_UID.2, FIA_USB.1) addresses T.MASK by providing measures to uniquely identify and authenticate users to the TOE through multiple authentication methods. In addition, this objective ensures that the strength of user's password or passphrase meets a scheme which ensure that unauthorized users cannot easily impersonate an authorized user by guessing their password.

Table 10-2: Threat to Objective Mapping

10.2 Security Functional Requirements Rationale

The following table provides a mapping with rationale to identify the security functional requirement components that address the stated TOE and environment objectives.

Objective	Security Functional Components	Rationale
O.ACCESS The TOE will provide measures to authorize users to access objects protected by the TOE once the user has been authenticated. User authorization is based on access rights configured by the authorized users of the TOE.	FDP_ACC.2(1) Complete access control	FDP_ACC.2(1) states the TSF shall enforce the Discretionary Access Control Policy on users requesting access to protected objects within the Operational Environment.
	FDP_ACC.2(2) Complete access control	FDP_ACC.2(2) states the TSF shall enforce the Mandatory Access Control Policy on users requesting access to protected objects within the Operational Environment .
	FDP_ACF.1(1) Security attribute based access control	FDP_ACF.1(1) states the TSF shall enforce the Discretionary Access Control Policy on users requesting access to objects based on the security attributes defined in Table 7-5, as well as, resource class name and entity name
	FDP_ACF.1(2) Security attribute based access control	FDP_ACF.1(2) states the TSF shall enforce the Mandatory Access Control Policy on users requesting access to objects based on AccessLevel, Type, Object Security Label, and Subject Security label.

Objective	Security Functional Components	Rationale
O.AUDIT The TOE will provide measures for recording security relevant events that will assist the authorized users in detecting misuse of the TOE and/or its security features that would compromise the integrity of the TOE and violate the security objectives of the TOE.	FAU_GEN.1 Audit data generation	FAU_GEN.1 states that the TSF shall be able to generate an audit record for the start-up and shutdown of the audit functions and all auditable events between subjects and resources. For each record, the TSF shall record the date/time/type/outcome of the event, the subject identity of the user which caused the event, the system ID, terminal ID, audit reason indicator, authority, mode, resource name, and if applicable the application or program that the user request was initiated from.
	FAU_GEN.2 ACIDentity association	FAU_GEN.2 states the TSF shall be able to associate each auditable event with the identity of the user that caused the event.
	FAU_SAR.1 Audit Review	FAU_SAR.1 states the TSF shall provide the authorized user with the audit privilege with the capability to read all events collected in FAU_GEN.1 from the audit records within their scope.
	FAU_SAR.2 Restricted audit review	FAU_SAR.2 states the TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.
	FAU_SEL.1 Selective audit	FAU_SEL.1 states that the TSF shall be able to select the events to be audited from the set of all auditable events based upon a Security Administrator's selection of the event's object identity, user identity, or permission.
O.AUTH The TOE will provide measures to uniquely identify all users and administrators. The TOE will authenticate the claimed identity prior to granting a user or administrator access the	FIA_ATD.1 User attribute definition	FIA_ATD.1 specifies the security attributes that should be maintained at the level of the user. This means that the security attributes listed are assigned to and are changed at the level of the user. In other words, changing a security attribute (see

Objective	Security Functional Components	Rationale
objects protected by the TOE.		Table 7-5) associated with a user should have no impact on the security attributes of any other user.
	FIA_SOS.1(1) Verification of Secret	FIA_SOS.1 states that the TSF shall enforce a password scheme that is sufficient to ensure that unauthorized users cannot easily impersonate an authorized user by guessing their password.
	FIA_SOS.1(2) Verification of Secret	FIA_SOS.1 states that the TSF shall enforce a passphrase scheme that is sufficient to ensure that unauthorized users cannot easily impersonate an authorized user by guessing their passphrase.
	FIA_SOS.2 TSF Generation of Secrets	FIA_SOS.2 states that the TSF shall enforce the password mechanism defined in FIA_SOS.1(1) to a Security Administrator defined NEWPW mask phrase which dictates the type of character accepted for each position in a password.
	FIA_UAU.2 User authentication before any action	FIA_UAU.2 requires a user be authenticated before any access to the TOE and objects protected by the TOE is allowed.
	FIA_UAU.4 Single-use authentication mechanisms	FIA_UAU.4 states that the TSF will prevent the reuse of a passticket for an authenticated session, which prevents an unauthorized user performing a replay attack with this authentication data.
	FIA_UAU.5 Multiple authentication mechanisms	FIA_UAU.5 states that the TSF shall provide password, passphrase, passtickets, Kerberos, or digital certificates for authentication, and the TSF shall authenticate any user's claimed identity according to the application from which the user is requesting system entry.
	FIA_UID.2 User identification before any action	FIA_UID.2 requires a user be identified before any access to the TOE and objects protected by the TOE is allowed.

Objective	Security Functional Components	Rationale
	FIA_USB.1 User-Subject Binding	FIA_USB.1 requires the TOE to perform a binding of security attributes associated with users that are authenticated with the subjects that represent them in the TOE. The TOE will create a user's ACID and security environment during this process
O.MANAGE The TOE will provide authorized administrators with the resources to manage and monitor user accounts, objects, and security information relative to the TOE.	FMT_MOF.1(1) Management of security functions behavior	FMT_MOF.1(1) states the TSF shall restrict the ability to perform operations specified in Table 7-5 to the Security Administrator.
	FMT_MOF.1(2) Management of security functions behavior	FMT_MOF.1(2) states the TSF shall restrict the ability to perform operations specified in Table 7-6 to the Scope Security Administrator based on their scope.
	FMT_MOF.1(3) Management of security functions behavior	FMT_MOF.1(3) states the TSF shall restrict the ability to perform self password and passphrase changes to users.
	FMT_MOF.1(4) Management of security functions behavior	FMT_MOF.1(4) states that users who own objects can perform all management functions on the object that are listed in Table 7-6.
	FMT_MSA.1 Management of security attributes	FMT_MSA.1 states the TSF shall enforce the Mandatory Access Control and Discretionary Access Control policies to modify,delete, manage, add, control, or change the user attributes as listed in Table 7-5 to Security Administrators or Scoped Security Administrators within their scope.

Objective	Security Functional Components	Rationale
	FMT_MSA.3 Static attribute initialization	FMT_MSA.3 states the TSF shall enforce the Mandatory Access Control and Discretionary Access Control policies to provide restrictive default values for security attributes that are used to enforce the SFP. It allows the Security Administrators or Scoped Security Administrators within their scope to override the default values set for security attributes when creating user accounts.
	FMT_SMF.1 Specification of management functions	FMT_SMF.1 requires that the TOE provide the ability to manage its security functions as defined it Table 7-6.
	FMT_SMR.1 Security Roles	FMT_SMR.1 requires the TOE to provide the ability to maintain the roles security administrator, scoped security administrator, and user. In addition, it requires that users be associated with these roles.
O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management.	ALC_DEL.1 Delivery Procedures	ALC_DEL.1 describes product delivery and a description of all procedures used to ensure objectives are not compromised in the delivery process.
	AGD_PRE.1 Preparative Procedures	AGD_PRE.1 documents the procedures necessary and describes the steps required for the secure installation, generation, and start-up of the TOE.
	AGD_OPE.1 Operational user guidance	AGD_OPE.1 describes the proper use of the TOE from a user standpoint.

Objective	Security Functional Components	Rationale
O.ROBUST_TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate.	FIA_AFL.1 Authentication Failure Handling	FIA_AFL.1 provides a detection mechanism for unsuccessful authentication attempts by users and administrators. The requirement enables a Security Administrator to set a threshold that prevents unauthorized users from gaining access to an authorized user's account by guessing authentication data. Once the threshold is surpassed the TOE will suspend the targeted user's ACID until the Security Administrator takes some action (e.g., un-suspends the account). Thus, limiting an unauthorized user's ability to gain unauthorized access to the TOE.
	FIA_UAU.2 User authentication before any action	FIA_UAU.2 requires a user be authenticated before any access to the TOE and objects protected by the TOE is allowed.
	FIA_UAU.4 Single-use authentication mechanisms	FIA_UAU.4 states that the TSF will prevent the reuse of a passticket for an authenticated session, which prevents an unauthorized user performing a replay attack with this authentication data.
	FIA_UAU.5 Multiple authentication mechanisms	FIA_UAU.5 states that the TSF shall provide password, passphrase, passtickets, Kerberos, or digital certificates for authentication, and the TSF shall authenticate any user's claimed identity according to the application from which the user is requesting system entry.
	FIA_UID.2 User identification before any action	FIA_UID.2 requires a user be identified before any access to the TOE and objects protected by the TOE is allowed.
	FTA_TSE.1 TOE session establishment	FTA_TSE.1 states that the TOE will deny session establishment if the user is suspended, or if there is a policy that defines the date/time, source, and AppID where access is allowed and the user's request does not adhere to the policy.

Table 10-3: Security Functional Requirements Rationale

10.3 EAL 4 Justification

The threats that were chosen are consistent with attacker of medium attack potential, therefore EAL4 was chosen for this ST.

10.4 Requirement Dependency Rationale

All Security Functional Requirement component dependencies have been met by the TOE as defined by the CC Part 2 with the exception of FPT_STM.1 and FMT_MTD.1.

FPT_STM.1, Reliable Time Stamps is a dependency of FAU_GEN.1. This dependency is met by the Operational Environment. The underlying Operating System (z/OS) will be available to the TOE for use in determining the timestamp for the audit trail.

FMT_MTD.1, Management of TSF Data is a dependency of FAU_SEL.1. The FMT_MTD.1 requirement has not been included in the ST because the TOE does not perform the management of data that is specific to the TSF. The data which the TOE allows users and administrators to perform management activities on is data that belongs to z/OS or the computing system, which has already been covered through the inclusion of the four iterations of FMT_MOF.1. The data which is managed through the TOE is considered z/OS or computing system data because the TOE provides the ability to manage the security aspects of the subjects (i.e. users, administrators, applications) and objects (e.g. component of the computing or operating system, dataset, volume) which belong to the computing or operating system. Therefore, the inclusion of the FMT_MTD.1 requirement would not satisfy the intent of the requirement or the intent of the dependency.

10.5 Assurance Measures

This section identifies the assurance measures provided by the developer in order to meet the security assurance requirement components for EAL4 augmented with ASE_TSS.2 and ALC_FLR.1. A description of each of the TOE assurance measures follows in Table 10-4.

Component	Document(s)	Rationale
ADV_ARC.1 Security Architecture Design	TOE Design Specification v1.0 IBM z/Architecture Principles of Operation CA Top Secret® for z/OS Control Options Guide r14 CA Top Secret® for z/OS Design Guide r14 z/OS Initial Program Load	These documents describe the security architecture of the TOE and its underlying operating system.
ADV_FSP.4 Functional Specification with complete summary	Functional Specification v1.0 CA Top Secret® for z/OS Auditor Guide r14	These documents describe the external interfaces to the TOE.

Component	Document(s)	Rationale
	CA Top Secret® for z/OS Compliance Information Analysis Guide r14 CA Top Secret® for z/OS Command Functions Guide r14 CA Top Secret® for z/OS Cookbook r14 CA Top Secret® for z/OS Design Guide r14 CA Top Secret® for z/OS Implementation: Other Interfaces Guide r14 CA Top Secret® for z/OS Messages and Codes Guide r14 CA Top Secret® for z/OS Report and Tracking Guide r14 CA Top Secret® for z/OS User Guide r14 IBM z/Architecture Principles of Operation IBM MVS System Commands IBM RACF Callable Services IBM RACROUTE Macro Reference	
ADV_IMP.1 Implementation Representation of the TSF	Low Level Design Specifications (annotated)	These documents demonstrate the correspondence between source code and design documentation.
ADV_TDS.3 Architectural Design	TOE Design Specification v1.0 Low Level Design Specifications (folder)	These documents describe the internal design of the TOE.
AGD_OPE.1 Operational User Guidance	CA Top Secret® for z/OS Auditor Guide r14 CA Top Secret® for z/OS Best Practices Guide r14 CA Top Secret® for z/OS Compliance Information Analysis Guide r14 CA Top Secret® for z/OS Implementation: CICS Guide r14 CA Top Secret® for z/OS Command Functions Guide r14 CA Top Secret® for z/OS Control Options Guide r14 CA Top Secret® for z/OS Cookbook r14 CA Top Secret® for z/OS Design Guide r14 CA Top Secret® for z/OS Installation Guide r14 CA Top Secret® for z/OS Implementation: Other Interfaces	These documents provide guidance on how to use CA Top Secret.

Component	Document(s)	Rationale
	Guide r14 CA Top Secret® for z/OS Messages and Codes Guide r14 CA Top Secret® for z/OS Multilevel Security Planning Guide r14 CA Top Secret® for z/OS Overview Guide r14 CA Top Secret® for z/OS Quick Reference Guide r14 CA Top Secret® for z/OS Release Notes r14 CA Top Secret® for z/OS Report and Tracking Guide r14 CA Top Secret® for z/OS Troubleshooting Guide r14 CA Top Secret® for z/OS User Guide r14	
AGD_PRE.1 Preparative Procedures	CA Top Secret® for z/OS Installation Guide r14 CA Top Secret® for z/OS Implementation: Best Practice Guide r14 CA Top Secret® for z/OS Implementation: CICS Guide r14 CA Top Secret® for z/OS Implementation: IMS Guide r14	These documents describe the setup procedures for CA Top Secret.
ALC_CMC.4 Authorizations Controls	MF CM Plan Development MF CM Plan Documentation 2010 STAR_Data_Sets STAR_Transfer	These documents describe and demonstrate the implementation and documentation configuration management.
ALC_CMS.4 CM Scope	CI list bookshelf TSS CI List Clarity TSS Top Secret EAL ALC_CMC file list 22 november	These documents demonstrate the CM scope of the TOE.
ALC_DEL.1 Delivery Procedures	ALC_DEL_MF Security_Electronic Delivery and Installation	This document describes product delivery for CA Top Secret.
ALC_DVS.1 Identification of Security Measures	MF CM Plan Development MF CM Plan Documentation Site inspection (folder with pictures) 2010.10.12 Onsite Assessment Report 11-Backup_Procedure-GIS-2008Jun09 1619-GRC-Global_Security-Pre-employment_Screening-2008Apr05 1621 – GSAP 3649-Access_Procedure-2007Jun29 5725-GRC-BP-C-RIM-Records_Security_and_Confidentiality_Policy-2008May23	These documents define the vendor's organizational security measures and provide verification that these measures are followed.

Component	Document(s)	Rationale
	5727-GRC-BP-C-RIM-Records_Disposal_Procedure-2008May15 5804-Privileged_Access-2008Jun24 7417-Enterprise_Procedure-Privacy_and_Data_Protection-2007Mar06 7705-Inactive_User_Account_procedure-2007Jun29 77260Server_Security_Procedure-2008Jun24 7978-US_Employee_Handbook-NorthAmerica-USA-2008Jul14 ALC_DVS CA Lisle Office_Building Security ALC_DVS Lisle Information Brochure	
ALC_FLR.1 Basic Flaw Remediation	Mainframe Security Flaw Remediation STAR ticket documentation STAR ticket Techsupp_policy	These documents provide the policies for issuing new releases of the TOE as corrective actions.
ALC_LCD.1 Life-Cycle Definition	5153-Project_360_Reference_Guide-2008Jul25	This document provides the life-cycle definition of the TOE.
ALC_TAT.1 Tools and Techniques	ALC_TAT[1].1 Overview updated 23 november Assembler Reference Summary	These documents describe the tools and techniques used in the development of the TOE.
ASE_CCL.1 Conformance Claims	CA Top Secret for z/OS r14 Security Target v1.1	This document describes the CC conformance claims made by the TOE.
ASE_ECD.1 Extended Components Definition	CA Top Secret for z/OS r14 Security Target v1.1	This document provides a definition for all extended components in the TOE.
ASE_INT.1 Security Target Introduction	CA Top Secret for z/OS r14 Security Target v1.1	This document describes the Introduction of the Security Target.
ASE_OBJ.2 Security Objectives	CA Top Secret for z/OS r14 Security Target v1.1	This document describes all of the security objectives for the TOE.
ASE_REQ.2 Security Requirements	CA Top Secret for z/OS r14 Security Target v1.1	This document describes all of the security requirements for the TOE.
ASE_SPD.1 Security Problem Definition	CA Top Secret for z/OS r14 Security Target v1.1	This document describes the security problem definition of the Security Target.
ASE_TSS.2 TOE Summary Specification	CA Top Secret for z/OS r14 Security Target v1.1	This document describes the TSS section of the Security Target.

Component	Document(s)	Rationale
ATE_COV.2 Analysis of Coverage	TSSAUDIT TSSCFILE_Testplan TSSCHART_TESTPLAN TSSCPR TSSOERPT_TESTPLAN TSSPROT_TESTPLAN TSSRCVR_backup test TSSREPORTS_TESTPLAN TSSrptst_TESTPLAN TSSTrack test plans TSSUTIL Test Plans Autoerase CICS_TESTPLAN MLSTESTPLAN SYSTEMENTRYTESTPLAN TSS_IMS Test Plan TSSCPF TEST PLAN Kerberos Test Plan OutBound_LDS-LDAP TestPlan PassPhrase PassTicket Testing Base Certificate Testing TSS_PASSWORD_TESTPLAN	These documents demonstrate functional test coverage for the TOE.
ATE_DPT.2 Testing: Security enforcing modules	CA Top Secret Functional Specification CA Top Secret TOE Design Specification CA Top Secret Low Level Design Specifications TSSAUDIT TSSCFILE_Testplan TSSCHART_TESTPLAN TSSCPR TSSOERPT_TESTPLAN TSSPROT_TESTPLAN TSSRCVR_backup test TSSREPORTS_TESTPLAN TSSrptst_TESTPLAN TSSTrack test plans TSSUTIL Test Plans Autoerase CICS_TESTPLAN MLSTESTPLAN SYSTEMENTRYTESTPLAN TSS_IMS Test Plan TSSCPF TEST PLAN Kerberos Test Plan OutBound_LDS-LDAP TestPlan PassPhrase PassTicket Testing Base Certificate Testing TSS_PASSWORD_TESTPLAN Administrator SECURITY	These documents demonstrate depth of functional testing.

Component	Document(s)	Rationale
	MANAGEMENT	
ATE_FUN.1 Functional Testing	System Configuration (folder) TSSAUDIT TSSCFILE_Testplan TSSCHART_TESTPLAN TSSCPR TSSOERPT_TESTPLAN TSSPROT_TESTPLAN TSSRCVR_backup test TSSREPORTS_TESTPLAN TSSrptst_TESTPLAN TSSTrack test plans TSSUTIL Test Plans Autoerase CICS_TESTPLAN MLSTESTPLAN SYSTEMENTRYTESTPLAN TSS_IMS Test Plan TSSCPF TEST PLAN Kerberos Test Plan OutBound_LDS-LDAP TestPlan PassPhrase PassTicket Testing Base Certificate Testing TSS_PASSWORD_TESTPLAN Administrator SECURITY MANAGEMENT tss r14 beta1 checklist tss r14 beta2 checklist	These documents demonstrate the format of functional testing and provide evidence of its completeness.
ATE_IND.2 Independent Testing – sample	Booz Allen Hamilton Independent Functional Test Plan TSS System Configuration Test Output (folder) Test Recordings (folder)	These documents outline the tests the evaluators executed to verify the correctness of the functional testing as well as the state of the system prior to the testing. They also provide evidence of the data reviewed during independent functional testing to demonstrate that the tests completed appropriately.
AVA_VAN.3 Focused Vulnerability Testing	Booz Allen Hamilton Vulnerability Test Plan Test Batch Jobs Wireshark Trace Test Output (folder)	These documents outline the vulnerability research performed and the tests which were used in order to attempt to tamper with or bypass the operational TOE. It also contains a record of the specific batch jobs run to complete the testing as well as the system outputs of these test activities.

Table 10-4: Assurance Requirements Evidence