Security Target for IBM RACF for z/OS V1R13

Version 2.04

October 4, 2012

TABLE OF CONTENTS

1. Introduction

This is version 2.04 of the Security Target for IBM® RACF for z/OS V1R13.

1.1 Security Target (ST) identification

Title: Security Target for IBM RACF for z/OS V1R13

Version: 2.04

Status: Final

Date: 2012-10-04

Sponsor: IBM Corporation

Developer: IBM Corporation

Certification ID: BSI-DSZ-CC-0816

Keywords: access control, discretionary access control, security labels, mandatory access control, security

This document is the Security Target for the Common Criteria (CC) evaluation of the IBM RACF for z/OS V1R13 access control component. It is conformant to the Common Criteria for Information Technology Security Evaluation Version 3.1 R3 [CC].

1.2 TOE Identification

The TOE is RACF for z/OS Version 1 Release 13 (provided as part of the Common Criteria Evaluated Base Package for z/OS V1R13, program number 5694-A01).

1.3 TOE overview

This Security Target (ST) documents the security characteristics of the IBM RACF for z/OS V1R13 access control component of z/OS V1R13.

RACF is the central component within z/OS responsible for user authentication, access control, management of user security attributes, and management of access rights.

RACF provides the interfaces for identification and authentication of users using different authentication mechanisms, interfaces that resource managers can use for both discretionary and mandatory access control to objects they define, interfaces for sophisticated security management functions, and the ability to generate audit records for security critical events.

1.4 TOE description

The Target of Evaluation (TOE) is the RACF component of the z/OS operating system. RACF is the component that is called within z/OS from any component that wants to perform user authentication,

access control to protected resources and the management of user security attributes and access rights.

RACF is designed as an authentication and access manager component that manages both user security attributes and access management attributes in its own database. Users are represented within RACF by user profiles and protected resources are represented by resource profiles. Users can be members of groups where each group is represented by a group profile.

Resource profiles are structured into classes, which represent the different types of resources. Within such a class a individual profile is represented by the name of the resource, which is unique within its class. Resource manager will then query RACF whenever they need to check a user's access rights to a resource. In this query they will specify the resource class, the name of the resource within the class, the type of access requested and the internal representation of the user that requests access.

RACF is also called when a component within z/OS needs to authenticate a user. In this case the z/OS component will call RACF and will pass the identity of the user, the authentication credentials presented, the name of the component requesting user authentication and several other parameters to RACF. Based on this information RACF will authenticate the user and, if successful, create a control block representing the user with the security attributes assigned. This control block is later used when a component of z/OS calls RACF for checking access rights.

RACF also provides interfaces that allow the management of user profiles, digital certificates assigned to users, group profiles, resource profiles, access rights, security labels and general RACF attributes. RACF also provides an interface that z/OS components can call to generate a security related audit record.

Note: The RACF Remote Sharing Facility (RRSF) is not considered as a part of this evaluation and therefore must not be used in an evaluated system configuration.

1.4.1 Intended method of use

RACF is designed to be used by z/OS components to perform user authentication, validate a user's access to a resource, audit security critical events, and manage RACF profiles, access rights to resources and RACF security parameter. It also provides interfaces to extract RACF status information. This interface is a programming interface implemented by the RACROUTE macro. RACF will check if the calling application has the right to use the function called.

In addition RACF exports a command interface that can be used by appropriately authorized users directly to perform management operations.

This Security Target specifies two modes of operation: a "normal" mode where labeled security features are not configured as required in this Security Target and a "Labeled Security Mode" where labeled security is configured as described in this Security Target. In "Labeled Security Mode" additional security functionality is active, which is marked with "Labeled Security Mode" in this document. Note that when functions of labeled security are configured differently than specified in this Security Target, the security functionality defined for the "normal" mode still works but additional restrictions may be imposed due to the way the functions for labeled security are configured.

1.4.2 Summary of security features

The primary security features of RACF are:

- identification and authentication of users
- discretionary access control
- mandatory access control and support for security labels (Labeled Security Mode)
- auditing
- security management

- TSF protection

These primary security features are supported by the domain separation and reference mediation properties of the other parts of the z/OS operating system, which ensure that the RACF functions are invoked when required and cannot be bypassed. RACF itself is protected by the architecture of the z/OS operating system from unauthorized tampering with the RACF functions and the RACF database.

1.4.2.1 Identification and authentication

RACF provides support for the identification and authentication of users by the means of

- 35 17 an alphanumeric RACF user ID and a system-encrypted password or password phrase.
- 35 17 an alphanumeric RACF user ID and a PassTicket, which is a cryptographically-generated password substitute encompassing the user ID, the requested application name, and the current date/time.
- 35 17 an x.509v3 digital certificate presented to a server application in the TOE environment that uses System SSL or TCP/IP Application Transparent TLS (AT-TLS) to provide TLS- or SSLv3-based client authentication, and then “mapped” (using TOE functions) by that server application or by AT-TLS to a RACF user ID.
- 35 17 a Kerberos™ v5 ticket presented to a server application in the TOE environment that supports the Kerberos mechanism, and then mapped by that application through the GSS-API programming services. The TOE also provides functions (specifically the R_ticketServ, and R_GenSec services) that enable the application server to validate the Kerberos ticket, and thus the authentication of the principal. The application server then translates (or maps) the Kerberos principal (using the TOE provided function of R_userMap) to a RACF user ID.

The TOE security functions authenticate the claimed identity of the user by verifying the password/phrase (or other mechanism, as listed above) and returning the result to the trusted program that used the RACF functions for user identification and authentication. It is up to the trusted program to determine what to do when the user identification and authentication process fails. When a user is successfully identified and authenticated RACF creates control blocks containing the user's security attributes as managed by RACF. Those control blocks are used later when a resource manager calls RACF to determine the user's right to access resources or when the user calls RACF functions that require the user to hold specific RACF managed privileges.

The required password quality can be tailored to the installation’s policies using various parameters. When creating users, administrators are required to choose an initial password and optionally a password phrase, that must usually be changed by the user during the initial logon that uses the password/phrase.

1.4.2.2 Discretionary access control

RACF implements the functions allowing resource managers within z/OS to control access to the resources they want to protect. Resources protected by RACF fall into two categories, based on the mechanisms used within RACF to describe them: Standard (e.g., MVS data sets, or general resources in classes defined by RACF or the system administrator), and UNIX (e.g., UNIX files, directories, and IPC objects instantiated by a UNIX file system). Discretionary access control (DAC) rules allow resource managers to differentiate access of users to resources based on different access types.

RACF standard DAC mechanism

Access types implemented in RACF for standard resources are hierarchical (i. e. a higher level of access implies all lower levels of access). The access types are in hierarchical order:

- ALTER
- CONTROL

- UPDATE
- READ
- EXECUTE
- NONE

RACF makes access control decisions based on the user's identity, security attributes, group authorities, and the access authority specified with respect to the resource profile. RACF leaves the interpretation of the semantics of the different access types to the resource manager. This allows for example a resource manager to manage privileges for users with RACF by defining a resource class for the privileges it wants to manage and individual resources within the class to represent individual privileges. A resource manager can then interpret a specific access type to such a resource as a specific privilege and allow a user to use functions it has bound to that privilege based on the access type the user has to the resource representing the privilege.

Access authorities to resources are stored in profiles. Discrete profiles are valid for a single, named resource and generic profiles are applicable to a group of resources, typically with similar names. For access permission checks, RACF always chooses the most specific profile for a resource. Profiles can have an access control list associated with them that contains a potentially large number of entries for different groups and users, thus allowing the modeling of complex, fine-grained access controls.

Access rights of users to resources can be set by the profile owner and by a user with the appropriate administrative privileges.

The TOE supports access decisions local applications want to enforce for the resources they control. Local applications can use the RACROUTE programming interface or the related programming interfaces from the RACF callable services to perform the access check. The request specifies the resource to be checked and the RACF user ID or group name whose access should be checked. Most RACF interfaces require either the calling program to execute with privileges (e. g. supervisor state or APF authorized) or require the user that started the program to have specific RACF managed privileges. The system or RACF privileges required are documented with each RACF interface.

RACF UNIX DAC mechanism

RACF implements POSIX-conformant access control that can be used for such named objects in the UNIX realm as UNIX file system objects and UNIX inter-process communication (IPC) objects. Access types for UNIX file system objects are read, write, and execute/search, and read and write for UNIX IPC objects. z/OS UNIX uses a dedicated interface to RACF to perform access control to file system objects which is based on the permission bits associated with a file, or based on access control lists, which are upward-compatible with the permission bits algorithm and implement the recommendations from Portable Operating System Interface for UNIX (POSIX) 1003.1e draft 17. Unlike the access rights to resources protected by RACF resource profiles, the permission bits and access control lists for those objects are not stored in the RACF database but are stored outside of RACF with the objects they protect and passed to RACF together with the request to check for access permissions. The RACF callable services contain the interfaces to perform those access checks and manage the related access permissions. The use of many of those interfaces is restricted to programs executing with system privileges and some of those interfaces are explicitly reserved for IBM's implementation of the UNIX System Services component or other z/OS components.

RACF supports resource managers in the decision when to prepare a resource for re-use.

1.4.2.3 Mandatory access control and support for security labels

In addition to DAC, RACF provides mandatory access control (MAC) functions that are required for Labeled Security Mode, which impose additional access restrictions on information flow on security classification. Users and resources can have a security label specified in their profile. Security labels contain a hierarchical classification (security level), which specify the sensitivity (for example: public, internal use, or secret), and zero or more non-hierarchical security categories (for example: PROJECTA or PROJECTB).

The access control enforced by the TOE ensures that users can only read labeled information if their security labels dominate the information's label, and that they can only write to labeled information containers if the container's label dominates the subject's, thus implementing the Bell-LaPadula model of information flow control. The system can also be configured to allow write-down for certain authorized users.

1.4.2.4 Auditing

RACF provides an auditing capability that allows generating audit records for security-critical events. RACF provides a number of logging and reporting functions that allow resource owners and auditors to identify users who attempt to access resources. Audit records are generated by RACF and submitted to another component of z/OS (System Management Facilities (SMF)), which collects them into an audit trail.

RACF always generates audit records for such events as unauthorized attempts to access the system or changes to the status of the RACF database. The security administrator, auditors, and other users with appropriate authorization can configure which additional optional security events are to be logged. In addition to writing records to the audit trail, messages can be sent to the security console to immediately alert operators of detected policy violations. RACF provides SMF records for all RACF-protected resources (either “traditional” or z/OS UNIX-based).

For reporting, auditors can unload all or selected parts of the SMF data for further analysis in a human-readable formats and can then upload the data to a query or reporting package, such as DFSORT™ if desired.

1.4.2.5 Security management

RACF provides a set of commands and options to adequately manage the TOE’s security functions. Additionally, RACF provides the capability of managing users, groups of users, general resource profiles, and RACF SETROPTS options.

RACF recognizes several authorities that are able to perform the different management tasks related to the TOE’s security:

- General security options are managed by security administrators.
- In Labeled Security Mode: management of MAC attributes is performed by security administrators.
- Management of users and their security attributes is performed by security administrators. Management of groups (and to some extent users) can be delegated to group security administrators.
- Users can change their own passwords or password phrases, their default groups, and their user names (but not their user IDs).
- In Labeled Security Mode: users can choose their security labels at login, for some login methods. (Note: this also applies in normal mode if the administrator chooses to activate security label processing.)
- Auditors manage the parameters of the audit system (a list of audited events, for example) and can analyze the audit trail.
- Security administrators can define what audit records are captured by the system.
- Discretionary access rights to protected resources are managed by the owners of the applicable profiles (or UNIX objects) or by security administrators.

1.4.2.6 TSF protection

TSF protection is based on several protection mechanisms that are provided by the underlying abstract machine and z/OS operating system:

- Privileged processor instructions are only available to programs running in the processor’s supervisor state
- Semi-privileged instructions are only available to programs running in an execution environment that is established and authorized by the TSF

- While in operation, all address spaces, as well as the data and tasks contained therein, are protected by the memory protection mechanisms of the underlying abstract machine
- z/OS protects the RACF address space and RACF functions from unauthorized access and either z/OS or RACF itself ensures that a caller of RACF services has the hardware or z/OS privileges (e. g. supervisor state, PSW key, APF authorization) required to invoke the service

z/OS address space management ensures that programs running in problem state cannot access protected memory or resources that belong to other address spaces.

Access to system services – through supervisor call (SVC) or program call (PC) instructions, for example – is controlled by z/OS, which requires that subjects who want to perform security-relevant tasks be authorized appropriately.

The hardware and firmware components that provide the abstract machine for the TOE are required to be physically protected from unauthorized access.

Tools are provided in the TOE environment to allow authorized administrators to check the correct operation of the underlying abstract machine.

In addition to the protection mechanism of the underlying abstract machine, z/OS also uses software mechanisms like the authorized program facility (APF) or specific privileges for programs in the UNIX system services environment to protect the TSF.

RACF uses the protection mechanisms provided by z/OS in the following way:

- the main functions of RACF are implemented as separate address spaces, which protects their code and data from direct interference by unprivileged programs executing in other address spaces
- RACF defines a limited set of SVCs and PCs that programs in all address spaces can invoke. Some PCs are defined such that the hardware prohibits their use unless the caller has appropriate hardware privileges. For other PCs as well as for SVCs, callable services and commands RACF performs its own check if the caller has the appropriate authorization to invoke the interface with the parameter specified. The specification of the individual interfaces also define the authorizations required to use the interface and specific parameter of the interface.
- many interfaces to RACF require the calling program to execute with system privileges like supervisor state, key 0, or APF authorization. When called without proper authorizations, those programs will fail to perform the requested action and either return with an appropriate return code, cause an exception, or cause an "abnormal end" (ABEND) with an ABEND code indicating the cause of the problem.

RACF uses the z/OS mechanisms for establishing error recovery routines which allows RACF to handle errors or exceptions detected by z/OS or the hardware and either recover from the error, perform any necessary clean-up operation and signal the error to the calling program, or (in the extreme case when RACF is not able to maintain its integrity e. g. when the RACF database is full or compromised) terminate RACF itself.

1.4.3 Configurations

1.4.3.1 Software configuration

The Target of Evaluation, RACF for z/OS V1R13, consists of:

- RACF for z/OS V1R13 (RACF) provided as part of the Common Criteria Evaluated Base Package for z/OS V1R13, program number 5694-A01
- APAR OA35973 (PTF UA62097)

  Note: This APAR will also be installed as part of the evaluated configuration for z/OS V1R13.

The z/OS Common Criteria Evaluated Base package must be installed according to the directions delivered with the media and configured according to the instructions in Chapter 7, “The evaluated configuration for the Common Criteria” in *z/OS Planning for Multilevel Security and the Common Criteria* ([PMLS]).

Although RACF allows the installation of system exits to tailor RACF processing, no such exits are allowed in the evaluated configuration with the exception of the sample ICHPWX11 exit and its associated IRRPHREX routine (which may be modified to suit the security administrator's needs). Additionally, the RACF Authorized Caller Table (ICHAUTAB) is not allowed in the evaluated configuration.

1.5 Structure

The structure of this document is as follows :

- Section 1 is the ST Introduction.
- Section 2 is the CC Conformance Claims
- Section 3 provides the Security Problem Definition
- Section 4 provides the Security Objectives
- Section 5 provides the Extended Components Definition
- Section 6 provides a statement on supporting functionality implemented by the operational environment of RACF
- Section 7 provides the statement of Security requirements
- Section 8 provides the TOE summary specification, which includes the detailed specification of the IT security functions

1.6 Terminology

This section contains a glossary of technical terms with definitions that are specific to this document. Terms defined in the [CC] are not reiterated here, unless stated otherwise. Some of these terms are used differently in other z/OS publications. This glossary includes the differences in usage where appropriate.

abstract machine

A processor design that is not intended to be implemented as hardware, but which is the notional executor of a particular intermediate language (abstract machine language) used in a compiler or interpreter. An abstract machine has an instruction set, a register set, and a model of memory. It may provide instructions that are closer to the language being compiled than any physical computer or it may be used to make the language implementation easier to port to other platforms.

access

If an authorized user is granted a request to operate on an object, the user is said to have *access* to that object. There are numerous types of access. Examples include *read access,* which allows the reading of objects, and *write access,* which allows the writing of objects.

access control policy

A set of rules used to mediate user access to TOE-protected objects. Access control policies consist of two types of rules: *access rules,* which apply to the behavior of authorized users, and *authorization rules,* which apply to the behavior of authorized administrators.

Accessor Environment Element

A RACF control block that describes the current user's security environment.

authorization

If an authorized user is granted a requested service, the user is said to have *authorization* to the requested service or object. There are numerous possible authorizations. Typical authorizations include *auditor authorization,* which allows an administrator to view audit records and execute audit tools, and *DAC override authorization,* which allows an administrator to override object access controls to administer the system.

authorized administrator

An authorized user who has been granted the authority to manage all or a defined subset of the functions of the TOE. Authorized administrators are expected to use this authority only in the manner prescribed by the guidance that is given to them.

authorized user

A user who has been properly identified and authenticated. Authorized users are considered to be legitimate users of the TOE. (Note: this is different from the z/OS concept of an "authorized program" which is a program running in supervisor state, or system key, or with APF authority.)

category

See *security category*.

classification (MLS)

A hierarchical designation for data that represents the sensitivity of the information. The equivalent IBM term is *security level.*

discretionary access control (DAC)

An access control policy that allows authorized users and authorized administrators to control access to objects based on individual user identity or membership in a group (PROJECTA, for example).

Lightweight Directory Access Protocol (LDAP)

A client/server protocol for accessing a directory service.

mandatory access control (MAC)

An access control policy that determines access based on the sensitivity (SECRET, for example) and category (PERSONNEL or MEDICAL, for example) of the information that is being accessed and the clearance of the user who is trying to gain access to that information.

mediation

When DAC and MAC policy rules are invoked, the TOE is said to be mediating access to TOE-protected objects.

password

For the purposes of this evaluation, a 6 to 8 character secret value used during some forms of user authentication, and allowing upper- and lower-case alphabetic, numeric, or national ($, #, @) characters. Passwords are initially assigned by administrators, but may be changed by the user to whom they are assigned.

password phrase

A 14 to 100 character secret value used in a manner similar to a password, except for its length and an expanded set of valid characters (upper- and lower-case alphabetic, special (including blanks), or numeric). In addition to assigning a password, administrators may assign a password phrase to a user.

Note: Phrase may be shorter (down to 9 characters) if enabled by an administrator-installed exit (ICHPWX11) that RACF supplies.

password/phrase

A shorthand term for “password or password phrase” sometimes used in this security target when statements apply equally to passwords or to password phrases.

SECLABEL

Synonym for *security label.*

SECLEVEL

Synonym for *security level (IBM).*

security category

A special designation for data at a certain level, which indicates that only people who have been properly briefed and cleared for access to data with this category can receive permission for access to the information.

security label

A name that represents the combination of a hierarchical level of classification (IBM security level) and a set of non-hierarchical categories (security category). Security labels are used as the base for mandatory access control decisions. Security labels are sometimes referred to as *SECLABELs.*

security level (IBM)

A hierarchical designation for data that represents the sensitivity of the information. Security levels are sometimes referred to as *SECLEVELs.* The equivalent MLS term is *classification.*

security level (MLS policy in the Bell-LaPadula model)

The combination of a hierarchical classification (called *security level* in z/OS) and a set of non-hierarchical categories that represents the sensitivity of information is known as the security level.

sensitivity label

A specific marking attached to subjects or objects that indicates the security level. The equivalent to this MLS term in other IBM documentation is *security label*.

user

A person who is trying to invoke a service that is offered by the TOE.

user ID

In z/OS, a string of up to eight characters defined as a RACF USER profile that uniquely identifies a user. Users who may use UNIX services will additionally have a numerical user identifier (UID) that is used by the UNIX subsystem for access decisions. The user name is an additional attribute that usually holds the user’s full name. While users can modify their user names, only administrators can change user IDs.

1.7 Abbreviations

ACEE	Accessor Environment Element
CC	Common Criteria
DAC	discretionary access control
LDAP	Lightweight Directory Access Protocol
MAC	mandatory access control
PADS	program access to data sets
PKI	Public Key Infrastructure
PP	Protection Profile
RACF	Resource Access Control Facility
SDSF	System Display and Search Facility
SFR	security functional requirement
TOE	Target of Evaluation
TSF	TOE security functions
TSP	TOE security policy

1.8 References

[ABC-V6] ABCs of z/OS System Programming Volume 6, IBM Redbook SG246986, First Edition, August 2008

[ADP] DoD Manual 5200.28-M: Techniques and procedures for Implementing, Deactivating and Evaluating Resource Sharing ADP Systems

[OSPP] Operating System Protection Profile, Version 2.0, BSI-CC-PP-0067, June 2010

[CC] Common Criteria for Information Technology Security Evaluation, Parts 1 to 3: [CCMB-2009-07-001, July 2009, Version 3.1R3], [CCMB-2009-07-002, July 2009, Version 3.1R3], and [CCMB-2009-07-003, July 2009, Version 3.1R3]

[CCINT] CCIMB Interpretations (as of October 1, 2012)

[CEM] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, [CCMB-2009-07-004, July 2009, Version 3.1R3]

[GUIDE] ISO/IEC TR 15446 Title: Information technology – Security techniques – Guide for the production of protection profiles and security targets, 2009

[PMLS] z/OS RACF Planning for Multilevel Security and the Common Criteria, Twelfth Edition, June 2011, GA22-7509-11

[ZARCH] IBM: z/Architecture: Principles of Operation, Ninth Edition, August, 2010, SA22-7832-08

[z/OS Concepts] Introduction to the New Mainframe: z/OS Basics, IBM Redbook SG246366, March 29, 2011

1.9 Trademarks

The following terms are trademarks or registered trademarks of International Business Machines

Corporation in the United States, other countries, or both:

DFSORT
@server
IBM
MVS
PR/SM
Print Services Facility
Processor Resource/Systems Manager
RACF
System z
VTAM
z/Architecture
z/OS
z/VM
zSeries
z9
z10

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

2. CC Conformance Claims

2.1 Common Criteria conformance

This Security Target is conformant to the Common Criteria for Information Technology Security Evaluation Version 3.1 R3 [CC]. It is *CC Part 2 extended* and *Part 3 conformant*, with a claimed Evaluation Assurance Level of EAL5, augmented by ALC_FLR.3.

3. Security Problem Definition

3.1 Introduction

The statement of the TOE security problem definition describes the security aspects of the environment in which the TOE is intended to be used and the manner in which it is expected to be deployed.

To this end, the statement of the TOE security environment identifies the list of assumptions made on the operational environment (including physical and procedural measures) and the intended method of use of the product, defines the threats that the product is designed to counter, and the organizational security policies with which the product is designed to comply.

3.2 Threat Environment

Threats to be countered by the TOE are characterized by the combination of an asset being subject to a threat, a threat agent and an adverse action.

3.2.1 Assets

Assets to be protected are:

1. RACF internal data used to control the operation of RACF (TSF data), including user and group profiles and profiles in other classes that RACF uses for its internal operations
2. RACF profiles managed by RACF on behalf of resource managers (user data)
3. RACF functions
4. The resources managed by the TSF that are used to store the above-mentioned objects, including the metadata needed to manage these objects.

3.2.2 Threat Agents

Threat agents are external entities that potentially may attack the TOE. They satisfy one or more of the following criteria:

- External entities not authorized to access assets may attempt to access them either by masquerading as an authorized entity or by attempting to use TSF services without proper authorization.
- External entities authorized to access certain assets may attempt to access other assets they are not authorized to either by misusing services they are allowed to use or by masquerading as a different external entity.
- Untrusted subjects may attempt to access assets they are not authorized to either by misusing services they are allowed to use or by masquerading as a different subject.

Threat agents are typically characterized by a number of factors, such as expertise, available resources, and motivation, with motivation being linked directly to the value of the assets at stake. The TOE protects against intentional and unintentional breach of TOE security by attackers possessing a moderate attack potential.

The *threat agents* can be categorized as one of the following:

- unauthorized users of the TOE (that is, individuals who have not been granted the right to access the system)
- authorized users of the TOE (that is, individuals who have been granted the right to access the system) but not given administrative authority. Those users may have programming capabilities and/or are allowed to install programs on the underlying z/OS operating system. Note: Security breaches in the operational environment that are not caused by security problems within the TOE itself are not subject of this evaluation.

The threat agents are assumed to originate from a well-managed user community in a moderately hostile working environment, and hence the product protects against threats of attempts to breach the system security by users with a moderate attack potential. The TOE is not intended to be applicable to circumstances in which protection is required against determined attempts by hostile and well-funded attackers with a high level of expertise to breach system security. The TOE relies on the z/OS operating system within its operational environment to ensure that users can not bypass the z/OS separation mechanisms and get operating system privileges (getting one of their programs to operate in supervisor state, with a storage key of 0 to 7, or as an APF authorized program). The TOE also relies on z/OS to enforce the decisions made by RACF for access to the data sets used by RACF. The TOE also relies on z/OS to enforce the protection of system memory and prohibit any untrusted program from modifying system memory used by RACF or accessing such memory other than the allowed access modes. The TOE also uses services from some z/OS components and relies upon those services to be implemented correctly.

3.2.3 Threats countered by the TOE

T.ACCESS.TSFDATA

A threat agent may read or modify TSF data using functions of the TOE without the necessary authorization

T.ACCESS.USERDATA

A threat agent may gain access to user data stored, processed or transmitted by the TOE without being appropriately authorized according to the TOE security policy by using functions provided by the TOE.

T.ACCESS.TSFFUNC

A threat agent may use or manage functionality of the TSF bypassing protection mechanisms of the TSF.

T.IA.MASQUERADE

A threat agent might masquerade as an authorized entity including the TOE itself or a part of the TOE in order to gain unauthorized access to user data, TSF data, or TOE resources.

T.IA.USER

A threat agent might gain access to user data, TSF data or TOE resources with the exception of public objects without being identified and authenticated by the TSF.

T.SENSITIVITY (Labeled Security Mode only)

The TOE may not adequately separate data on the basis of its sensitivity label, thereby allowing access to RACFinternal data, RACF user data or resources protected by external resource managers in violation of the label-based access control policy.

3.3 Assumptions

This section describes the security aspects of the environment in which the TOE is intended to be used. It includes information about the physical, personnel, procedural, and connectivity aspects of the environment.

The TOE is assured to provide effective security measures in its intended environment only if it is installed, managed, and used correctly. The operational environment must be managed in accordance with user/administrator guidance documentation. The following specific conditions are assumed to exist in an environment where the TOE is employed.

3.3.1 Environment of use of the TOE

Physical

The TOE is intended for application in user areas that have physical control and monitoring. It is assumed that the following physical conditions will exist:

A.PHYSICAL

It is assumed that the IT environment provides the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE.

Personnel

A.MANAGE

The TOE security functionality is managed by one or more competent individuals. The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the guidance documentation.

A.AUTHUSER

Authorized users possess the necessary authorization to access at least some of the information managed by the TOE and are expected to act in a cooperating manner in a benign environment.

A.TRAINEDUSER

Users are sufficiently trained and trusted to accomplish some task or group of tasks within a secure IT environment by exercising complete control over their user data.

Procedural

A.DETECT

Any modification or corruption of security-enforcing or security-relevant files of the TOE, user or the underlying platform caused either intentionally or accidentally will be detected by an administrative user.

Operating system

A.OPERATING_SYSTEM

The z/OS operating system the TOE is integrated in protects the TOE from modification and access of the TOE's TSF data and user data other than through the interfaces provided by the TOE. The operating system also ensures that no unauthorized user program can escalate its privileges such that it can bypass the separation and memory protection functions of the operating system. RACF also relies on the functional support of the following components of z/OS:

- SMF for the collection, protection and analysis of the audit records.
- DFSMS for access to data sets RACF uses for its TSF data and user data.

- UNIX System Services for storage of and access to TSF data RACF stores with UNIX file system and IPC objects.
- PKI for the provision of PKI services backing the RACF services provided by the R_pkiserv function.
- IBM Tivoli directory server for the provision of LDAP services used by the R_proxyserv callable service
- ICSF to manage PKCS#11 key tokens used by RACF and provide cryptographic services used by RACDCERT
- SystemSSL for certificate validation
- BCP for address space separation, assignment of hardware privileges to address spaces and programs, management of APF authorizations, and general protection of data in address spaces that need to be protected from read and/or write access by untrusted subjects

Note: the RACF callable services R_dceinfo and R_dceruid have no use in the evaluated configuration of RACF as z/OS (the environment for this evaluation) no longer provides DCE functionality, and always return with an error, since the administrators will have no reason to activate the DCEUUIDS class of RACF nor to define DCE segments in USER profiles.

A.TRUSTED_PROGRAMS

The operational environment prohibits the installation of untrusted programs that execute with hardware or operating system privileges which would allow this program to use RACF interfaces reserved for trusted programs. Trusted Programs use the RACF interfaces in accordance with their specification.

3.4 Organizational security policies

P.ACCOUNTABILITY

The users of the TOE shall be held accountable for their security-relevant actions within the TOE.

P.USER

Authority shall only be given to users who are trusted to perform the actions correctly.

P.RESOURCE_LABELS (Labeled Security Mode only)

All resources accessible by subjects and all subjects must have associated labels identifying their sensitivity levels.

P.USER_CLEARANCE (Labeled Security Mode only)

All users must have a clearance level identifying the maximum sensitivity levels of data they may access.

4. Security objectives

This section defines the security objectives of the TSF and its supporting environment. Security objectives for the TOE, reflect the stated intent to counter identified threats, comply with any organizational security policies identified, or both. In addition, assumptions related to the operational environment of the TOE are upheld by respective objectives for the TOE environment. All of the identified threats, organizational policies, and assumptions are addressed under one of the following categories.

4.1 Security objectives for the TOE

O.AUDITING

The TSF must be able to generate audit records for defined security-relevant events (including security-critical actions of users of the TOE). The TSF must submit those records to the operating system auditing function, which adds the time and date and stores the audit records in a protected area.

O.DISCRETIONARY.ACCESS

The TSF must control access of subjects and/or users to named resources based on identity of the object. The TSF must allow authorized users to specify for each access mode which users/subjects are allowed to access a specific named object in that access mode.

O.I&A

The TOE must ensure that users have been successfully authenticated before allowing any action the TOE has defined to provide to authenticated users only.

O.MANAGE

The TSF must provide all the functions and facilities necessary to support the authorized users that are responsible for the management of TOE security mechanisms and must ensure that only authorized users are able to access such functionality.

O.I&A.MULTIPLE

The TOE shall allow the concurrent use of multiple identification and authentication mechanisms implementing the identification and authentication policy.

O.PROGRAM_INTEGRITY_SUPPORT

The TOE shall provide a functionality that allows its operational environment the implementation of a function that verifies the integrity and authenticity of programs before they are loaded and executed.

O.LS.CONFIDENTIALITY (Labeled Security Mode only)

The TOE shall support resource managers to control information flow between entities and resources based upon the sensitivity labels of users and resources.

O.LS.LABEL (Labeled Security Mode only)

The TOE shall provide the capability to label all users, and all resources, to allow resource managers to restrict information flow based on the sensitivity labels.

4.2 Security objectives for the operational environment

OE.ADMIN

Those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of the information it contains.

OE.INFO_PROTECT

Those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner. In particular:

1) All applications trusted by the operating system must be approved for the handling of the most sensitive data held by the system. Such applications as well as the operating system itself are assumed to be adequately protected against threats to the confidentiality and integrity of the data handled.
2) DAC protections on security-relevant resources shall always be set up correctly.
3) Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data.
4) Resource managers that use the TOE to protect their resources invoke the TOE on all attempts to access resources they manage, provide correct information about the subject that attempts to access the resource, the resource itself and the attempted type of access. Resource managers will honor the access decision made by the TOE.

OE.INSTALL

Those responsible for the TOE must establish and implement procedures to ensure that the components that comprise the TOE are distributed, installed and configured in a secure manner supporting the security mechanisms provided by the TOE.

OE.MAINTENANCE

Authorized users of the TOE must ensure that the comprehensive diagnostics facilities provided by the product are invoked at every scheduled preventative maintenance period.

OE.PHYSICAL

Those responsible for the TOE must ensure that those parts of the TOE critical to enforcement of the security policy are protected from physical attack that might compromise IT security objectives. The protection must be commensurate with the value of the IT assets protected by the TOE.

OE.RECOVER

Those responsible for the TOE must ensure that procedures and/or mechanisms are provided to assure that after system failure or other discontinuity, recovery without a protection (security) compromise is achieved.

OE.OS_SEP

The z/OS operating system provides the mechanisms to separate the address spaces of RACF from any untrusted address spaces and provides the mechanisms to protect RACF programs and data within an address space from any uncontrolled access by untrusted entities. The operating system assigns hardware and software privileges only to programs that are defined as trusted and prohibits any untrusted subject to escalate its privileges to supervisor state, a privileged storage key or to APF authorization. Such privilege escalation will only happen when the defined z/OS interfaces that imply such privilege escalations are used and the program that is executed after this privilege escalation has been defined as a trusted program.

OE.TRUSTED_PROGRAMS

Those responsible for the operating system the TOE is integrated in must ensure that only programs that are fully trusted are installed such that they execute with hardware privileges (system storage key or supervisor state) or with operating system privileges (APF authorization).

4.3 Security Objectives Rationale

4.3.1 Security Objectives Coverage

The following table provides a mapping of TOE objectives to threats and policies, showing that each objective counters or enforces at least one threat or policy, respectively.

Objective	Threats / OSPs
O.AUDITING	P.ACCOUNTABILITY
O.DISCRETIONARY.ACCESS	T.ACCESS.TSFDATA T.ACCESS.USERDATA
O.I&A	T.IA.MASQUERADE T.IA.USER
O.MANAGE	T.ACCESS.TSFFUNC P.ACCOUNTABILITY P.USER
O.I&A.MULTIPLE	T.IA.MASQUERADE

Objective	Threats / OSPs
	T.IA.USER
O.PROGRAM_INTEGRITY_SUPPORT	T.ACCESS.TSFDATA T.ACCESS.TSFFUNC
O.LS.CONFIDENTIALITY	T.SENSITIVITY P.USER_CLEARANCE
O.LS.LABEL	P.RESOURCE_LABELS P.USER_CLEARANCE T.SENSITIVITY

Table 1: Mapping of security objectives to threats and policies

The following table provides a mapping of the objectives for the Operational Environment to assumptions, threats and policies, showing that each objective holds, counters or enforces at least one assumption, threat or policy, respectively.

Objective	Assumptions / Threats / OSPs
OE.ADMIN	A.MANAGE A.AUTHUSER A.TRAINEDUSER
OE.INFO_PROTECT	A.MANAGE A.AUTHUSER A.TRAINEDUSER P.USER
OE.INSTALL	A.MANAGE A.DETECT
OE.MAINTENANCE	A.DETECT
OE.PHYSICAL	A.PHYSICAL
OE.RECOVER	A.MANAGE

Objective	Assumptions / Threats / OSPs
OE.OS_SEP	A.OPERATING_SYSTEM T.ACCESS.USERDATA T.ACCESS.TSFDATA T.ACCESS.TSFFUNC
OE.TRUSTED_PROGRAMS	A.TRUSTED_PROGRAMS T.ACCESS.USERDATA T.ACCESS.TSFDATA T.ACCESS.TSFFUNC

Table 2: Mapping of security objectives for the Operational Environment to assumptions, threats and policies

4.3.2 Security Objectives Sufficiency

The following rationale provides justification that the security objectives are suitable to counter each individual threat and that each security objective tracing back to a threat, when achieved, actually contributes to the removal, diminishing or mitigation of that threat:

Threat	Rationale for security objectives
T.ACCESS.TSFDATA	The threat of accessing TSF data without proper authorization is removed by: • O.DISCRETIONARY.ACCESS requiring that data, including TSF data stored with the TOE, have discretionary access control protection (Note: some TSF data is not intended to be directly accessible by external entities and therefore no discretionary access can be assigned to such TSF data), • O.PROGRAM_INTEGRITY_SUPPORT requiring the TOE to implement a supporting function for program integrity verification • OE.OS_SEP requiring that the operating system protects RACF from access to its TSF data and user data other than through the RACF provided interfaces and also prohibits that untrusted user programs escalate their privileges such that it can bypass the hardware and OS provided protection mechanisms, • OE.TRUSTED_PROGRAMS requiring that those responsible for the operating system don't install untrusted programs with privileges that would allow those programs to bypass operating system protection

Threat	Rationale for security objectives
T.ACCESS.USERDATA	The threat of accessing user data without proper authorization is removed by: • O.DISCRETIONARY.ACCESS requiring that data including user data stored with the TOE, have discretionary access control protection, • OE.OS_SEP requiring that the operating system protects RACF from access to its TSF data and user data other than through the RACF provided interfaces and also prohibits that untrusted user programs escalate their privileges such that it can bypass the hardware and OS provided protection mechanisms, • OE.TRUSTED_PROGRAMS requiring that those responsible for the operating system don't install untrusted programs with privileges that would allow those programs to bypass operating system protection
T.ACCESS.TSFFUNC	The threat of accessing TSF functions without proper authorization is removed by: • O.MANAGE requiring that only authorized users utilize management TSF functions, • O.PROGRAM_INTEGRITY_SUPPORT requiring the TOE to implement a supporting function for program integrity verification • OE.OS_SEP requiring that the operating system protects RACF from access to its TSF data and user data other than through the RACF provided interfaces and also prohibits that untrusted user programs escalate their privileges such that it can bypass the hardware and OS provided protection mechanisms, • OE.TRUSTED_PROGRAMS requiring that those responsible for the operating system don't install untrusted programs with privileges that would allow those programs to bypass operating system protection
T.IA.MASQUERADE	The threat of masquerading as an authorized entity in order to gain unauthorized access to user data, TSF data or TOE resources is removed by: • O.I&A requiring that each entity interacting with the TOE is properly identified and authenticated before allowing any action the TOE is defined to provide to authenticated users only.

Threat	Rationale for security objectives
	• O.I&A.MULTIPLE requiring that the TOE supports multiple identification and authentication mechanisms, allowing to use the most appropriate one.
T.IA.USER	The threat of accessing user data, TSF data or TOE resources without being identified and authenticated is removed by: • O.I&A requiring that each entity interacting with the TOE is properly identified and authenticated before allowing any action the TOE has defined to provide to authenticated users only. • O.I&A.MULTIPLE requiring that the TOE supports multiple identification and authentication mechanisms, allowing to use the most appropriate one.
T.SENSITIVITY	The threat of not adequately separating data on the basis of its sensitivity label, thereby allowing information to flow illicitly from or to users is removed by: • O.LS.CONFIDENTIALITY requiring the TOE to provide assistance to resource managers allowing them to control information flow between entities and resources based upon the sensitivity labels of users and resources. RACF uses its own functionality to protect its resources, • O.LS.LABEL requiring the TOE to provide labels used to enforce a label-based mandatory access control policy. Note: while RACF mainly provides this functionality to support external resource managers, it also uses this functionality for its own resources.

Table 3: Sufficiency of objectives countering threats

The following rationale provides justification that the security objectives for the environment are suitable to cover each individual assumption, that each security objective for the environment that traces back to an assumption about the environment of use of the TOE, when achieved, actually contributes to the environment achieving consistency with the assumption, and that if all security objectives for the environment that trace back to an assumption are achieved, the intended usage is supported:

Assumption	Rationale for security objectives
A.PHYSICAL	The assumption on the IT environment to provide the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE is covered by: • OE.PHYSICAL requiring physical protection.

Assumption	Rationale for security objectives
A.MANAGE	The assumptions on the TOE security functionality being managed by one or more competent trustworthy individuals is covered by: • OE.ADMIN requiring trustworthy personnel managing the TOE, • OE.INFO_PROTECT requiring personnel to ensure that information is protected in an appropriate manner, • OE.INSTALL requiring personnel to ensure that components that comprise the system are distributed, installed and configured in a secure manner supporting the security mechanisms provided by the TOE, • OE.RECOVER requiring personnel to assure that after system failure or other discontinuity, recovery without a protection (security) compromise is achieved.
A.AUTHUSER	The assumption on authorized users to possess the necessary authorization to access at least some of the information managed by the TOE and to act in a cooperating manner in a benign environment is covered by: • OE.ADMIN ensuring that those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of the information it contains, • OE.INFO_PROTECT requiring that DAC protections on security-relevant files (such as audit trails and authentication databases) shall always be set up correctly and that users are authorized to access parts of the data maintained by the TOE.
A.TRAINEDUSER	The assumptions on users to be sufficiently trained and trusted to accomplish some task or group of tasks within a secure IT environment by exercising complete control over their user data is covered by: • OE.ADMIN requiring competent personnel managing the TOE, • OE.INFO_PROTECT requiring that those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner and that users are trained to exercise control over their own data.

Assumption	Rationale for security objectives
A.DETECT	The assumption that modification or corruption of security-enforcing or security-relevant files will be detected by an administrative user is covered by: • OE.INSTALL requiring an administrative user to ensure that the TOE is distributed, installed and configured in a secure manner supporting the security mechanisms provided by the TOE, • OE.MAINTENANCE requiring an administrative user to ensure that the diagnostics facilities are invoked at every scheduled preventative maintenance period, verifying the correct operation of the TOE,
A.OPERATING_SYSTEM	The assumptions on the separation functions provided by the z/OS operating system is covered by: • OE.OS_SEP requiring the z/OS operating system to protect the address spaces of RACF and the RACF programs and data within an address space from uncontrolled access by subject not executing with hardware privileges or APF authorization.
A.TRUSTED_PROGRAMS	The assumptions on the trusted programs installed on the z/OS operating system is covered by: • OE.TRUSTED_PROGRAMS requiring the persons responsible for installing trusted programs on z/OS to not install any program executing with critical privileges unless they trust the program to not misuse those privileges and not use the RACF interfaces for privileged programs in a way not allowed by the specification of the interface.

Table 4: Sufficiency of objectives holding assumptions

The following rationale provides justification that the security objectives are suitable to cover each individual organizational security policy, that each security objective that traces back to an OSP, when achieved, actually contributes to the implementation of the OSP, and that if all security objectives that trace back to an OSP are achieved, the OSP is implemented:

OSP	Rationale for security objectives
P.ACCOUNTABILITY	The policy to hold users accountable for their security-relevant actions within the TOE is implemented by: • O.AUDITING providing the TOE with audit functionality, • O.MANAGE allowing the management of this function.
P.USER	The policy to match the trust given to a user and the actions the user is given authority to perform is implemented by:

OSP	Rationale for security objectives
	• O.MANAGE allowing appropriately-authorized users to manage the TSF, • OE.INFO_PROTECT, which requires that users are trusted to use the protection mechanisms of the TOE to protect their data.
P.RESOURCE_LABELS	The policy that resources accessible by subjects and all subjects must have associated labels identifying their sensitivity levels is implemented by: • O.LS.LABEL providing the capability to label all subjects and all objects accessible by subjects to restrict the information flow based on the sensitivity labels.
P.USER_CLEARANCE	The policy that all users must have a clearance level identifying the maximum sensitivity levels of data they may access is implemented by: • O.LS.CONFIDENTIALITY requiring the TOE to control information flow between entities and resources based upon the sensitivity labels of users and resources. • O.LS.LABEL ensures that objects and subjects are accurately labeled for the TOE to enforce the label policy.

Table 5: Sufficiency of objectives enforcing Organizational Security Policies

5. Extended Components Definition

This Security Target uses security functional requirements taken from part 2 of the Common Criteria plus two extended security functional requirements (FIA_USB.2 and FAU_GEN_SUB.1).

5.1 FIA_USB.2

FIA_USB.2 has been taken from the "Operating System Protection Profile" [OSPP]. This extended SFR is defined there as:

FIA_USB.2 Enhanced user-subject binding

FIA_USB.2 is analog to FIA_USB.1 except that it adds the possibility to specify rules whereby subject security attributes are also derived from TSF data other than user security attributes.

Component leveling

FIA_USB.2 is hierarchical to FIA_USB.1.

Management

See management description specified for FIA_USB.1 in [CC].

Audit

See audit requirement specified for FIA_USB.1 in [CC].

FIA_USB.2 Enhanced user-subject binding

Hierarchical to: FIA_USB.1 User-subject binding

Dependencies: FIA_ATD.1 User attribute definition

FIA_USB.2.1	The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [assignment: list of user security attributes].
FIA_USB.2.2	The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [assignment: rules for the initial association of attributes].
FIA_USB.2.3	The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [assignment: rules for the changing of attributes].
FIA_USB.2.4	The TSF shall enforce the following rules for the assignment of subject security attributes not derived from user security attributes when a subject is created: [assignment: rules for the initial association of the subject security attributes not derived from user security attributes].

Rationale

An operating system may derive subject security attributes from other TSF data that are not directly user security attributes. An example is the point-of-entry the user has used to establish the connection. An access control policy may also use this subject security attribute within its access control policy, allowing access to critical objects only when the user has connected through specific ports-of-entry.

5.2 FAU_GEN_SUB.1

This Security Target includes an extended component for audit data generation. This extended component defines a subset of the component FAU_GEN.1 as defined in part 2 of the CC.

This extended component needed to be defined since RACF uses the audit trail interfaces provided by the SMF component of z/OS for trusted components that want to store their audit records in the common audit trail provided by z/OS. While RACF collects all the information for its own audit records and formats the audit records, it does not generate an audit record for the start and the stop of the audit system (this is done by SMF, which controls the audit trail). RACF also does not store the time and date into the audit record. This is also a function that is performed by SMF when a trusted component submits an audit record for inclusion into the SMF audit trail.

This way of handling audit records is common for a large number of products that on the one hand want to produce audit records and on the other hand do not want to implement the functions to manage the audit trail, protect the audit trail, or process audit records for evaluation. Many applications use audit trails provided either by the underlying operating system or by a dedicated audit component in the IT environment (e. g. a log host). In order to have a single time source most of those dedicated audit systems that provide an interface to other product to submit audit records for storing them in the audit trail will put the time stamp with the time and date into the audit records themselves rather than relying on the other systems to synchronize their time sources.

The extended component FAU_GEN_SUB.1 has been derived as a subset from the component FAU_GEN.1 as listed in part 2 of the CC. The requirement for auditing the start and stop of the audit system has been dropped compared to FAU_GEN.1.1 and in FAU_GEN.1.2 the requirement to include the time and date has been dropped. As a consequence of dropping the inclusion of the time and date, the extended component no longer has a dependency on FPT_STM.1

Component leveling

FAU_GEN_SUB.1 is not hierarchical to any other component.

Management

There are no management activities foreseen..

Audit

There are no auditable events foreseen.

FAU_GEN_SUB.1 Subset audit data generation

Hierarchical to: No other components.

Dependencies: none

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

a) All auditable events for the [selection, choose one of: minimum, basic, detailed, not specified] level of audit; and

b) [assignment: other specifically defined auditable events].

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

a) type of event, subject identity (if applicable), 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 ST, [assignment: other audit relevant information].

6. Supporting Functionality provided by the Operational Environment

In the case of RACF the security functionality defined in chapter 8 of this Security Target depends on the supporting functionality of the z/OS operating system defined in this section. Note that also z/OS itself relies on security functional requirements provided by its operational environment to support the provision of its security functionality. Since RACF relies on the general protection mechanisms provided by z/OS, this implies by transitivity that RACF also relies on the security functional requirements provided by the operational environment of z/OS. To avoid duplication, the security functional requirements for the z/OS operational environment are not listed here.

There are several components of z/OS that are used by the TOE to implement its security functional requirements. Those are:

- SMF to record and store audit records generated by RACF, prevent records from getting lost and other z/OS components to provide utilities for reading and searching audit records.
- PKI Services to provide the PKI related services used by the R_PKIServ RACF callable service. This includes the generation of key pairs, the generation of digital certificates.
- DFSMS which provides the functions for managing and accessing data sets. RACF uses z/OS data sets for persistent storage of its TSF data.
- UNIX System Services for storage and access to TSF data RACF stores with UNIX file system and IPC objects.
- IBM Tivoli directory server for the provision of directory services used by the R_proxyserv callable service.
- ICSF which is used to manage PKCS#11 tokens used by RACF and provide cryptographic services used by RACDCERT.
- SystemSSL for certificate validation.
- BCP for address space and memory management, assignment of hardware privileges to address spaces and programs, management of system calls, interrupts and exceptions, management of APF authorizations, general protection of data in address spaces that need to be protected from read and/or write access by untrusted subjects, scheduling of processes, provision of serialization services, and provision of notification services.

7. Security Functional Requirements

In this section assignment or selection operations on SFRs are marked **bold** while refinements are marked ***bold and italic***.

Iteration operations are identified by a suffix in parentheses added to the SFR identifier.

7.1 Security audit (FAU)

Subset audit data generation (FAU_GEN_SUB.1)

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

a) All auditable events for the **not specified** level of audit; and

b) all modifications to the set of events being audited;

c) all user authentication attempts;

d) all denied accesses to resources;

e) explicit modifications of access rights to resources; and

f) the events listed in table 6, "Auditable Events".

Component	Event	Details
FAU_GEN_SUB.1	Startup of RACF .	SMF type 81 record (RACF initialization).
FAU_GEN.2	None.	
FAU_SAR.1	Reading of information from the audit records.	SMF type 80 record for the raw and saved SMF data sets.
FAU_SEL.1	All modifications to the audit configuration that occur while the audit collection functions are operating.	SMF records generated by the RACF commands that modify the audit configuration (SMF type 90 record, subtypes 5 and 9. IFASMFDP and IDCAMS (which are part of the TOE environment) can be used to report on these events).

Component	Event	Details
FCS_COP.1	Success or failure of signature verification for program signing. Failure to validate the certificate chain.	SMF type 80 record event code 86 (56 hex)
FDP_ACF.1(GRD)	All requests to perform an operation on an object covered by the Security Function Policy (SFP).	SMF type 80 record, event code 2 for access to MVS resources.
FDP_ACF.1(UFS) FDP_ACF.1(IPC)	All requests to perform an operation on an object covered by the Security Function Policy (SFP).	SMF type 80 record, event codes 28-30 for access to UNIX resources.
FDP_IFC.2	None.	
FDP_IFF.2	All decisions on requests for information flow.	SMF type 80 record, event code 2, with reason indicating SECLABEL AUDIT.
FIA_AFL.1	Authentication failure notification and account locking.	SMF type 80 record, event code 1, all qualifiers except 0, 12 and 13. Qualifier 7 especially reports account locking.
FIA_ATD.1(HU) FIA_ATD.1(LS)	None.	
FIA_SOS.1	Rejection or acceptance by the TSF of any tested secret.	SMF type 80 record, event code 1, qualifier 1 (password/phrase not valid). Also SMF type 80, event code 68, qualifier 0 (success) or 1 (failure) to generate a Kerberos TGT (generated by Network authentication service, not the TOE) Also SMF type 80, event code 70, qualifier 2 for R_PKIServ Export function with incorrect passphrase.
FIA_UAU.1	All use of the authentication mechanism.	SMF type 80 record, event code 1, various qualifiers

Component	Event	Details
		and SMF record type 30 subtypes 1 and 5). Also SMF type 80, event code 68, qualifier 0 (success) or 1 (failure) to generate a Kerberos TGT (generated by Network authentication service, not the TOE) Also SMF type 83, subtype 3, event codes 2,4,6,11 for LDAP bind operations.
FIA_UAU.5	None specific. All authentication functions produce the audit records mentioned for FIA_UAU.1 and FIA_UID.1	
FIA_UAU.7	None.	
FIA_UID.1	All use of the user identification mechanism, including the identity provided during successful attempts.	SMF type 80 record, event code 1, various qualifiers. (Note: the authorized caller of the RACF authentication function can control whether RACF audits the result or not, and in cases where the caller instructs RACF not to audit the result the caller is responsible for providing appropriate alternative auditing.)
FIA_USB.1 FIA_USB.2	Success and failure of binding user security attributes to a subject (e.g. success and failure to create a subject).	SMF type 80 record, event code 1, various qualifiers. (Note: the authorized caller of the RACF authentication function can control whether RACF audits the result or not, and in cases where the caller instructs RACF not to audit the result the caller is responsible for providing appropriate alternative auditing.)
FMT_MSA.1(GRD) FMT_MSA.1(UFS) FMT_MSA.1(IPC)	All modifications of the values of security attributes.	SMF type 80 record (generated by the RACF commands).

Component	Event	Details
FMT_MSA.1(FLA) FMT_MSA.1(LS)		
FMT_MSA.3(LS) FMT_MSA.3(GRD) FMT_MSA.3(UFS) FMT_MSA.3(FLA)	Modifications of the default setting of permissive or restrictive rules. All modifications of the initial value of security attributes.	SMF type 80 record (generated by the RACF commands).
FMT_MSA.3(IPC)	none	
FMT_MTD.1(SO) FMT_MTD.1(AE) FMT_MTD.1(UA) FMT_MTD.1(RA) FMT_MTD.1(TH) FMT_MTD.1(AD) FMT_MTD.1(RC) FMT_MTD.1(DC)	All modifications to the values of TSF data.	SMF type 80 record (generated by the RACF commands).
FMT_REV.1(USR)	All attempts to revoke security attributes.	SMF type 80 record (generated by the RACF commands).
FMT_REV.1(OSA)	All attempts to revoke security attributes.	SMF type 80 record (generated by the RACF commands).
FMT_SMF.1	None specifically associated with this SFR, but auditing is covered under the FMT_MSA.1, FMT_MSA.3, FMT_MTD.1, FMT_REV.1, FAU_SAR.1, FAU_SEL.1, and FMT_SMR.1 requirements which are implied by FMT_SMF.1 as discussed in chapter 7.	
FMT_SMR.1	Modifications to the group of users that are part of a role.	SMF type 80 record (generated by the RACF commands). See the event codes related to the individual commands
FMT_SMR.1	Every use of the rights	SMF type 80 record.

Component	Event	Details
	of a role. (Additional / Detailed)	
FPT_TDC.1(RA) FPT_TDC.1(LS)	None	

Table 6: Audit event details

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

a) Type of event, subject identity (if applicable), and 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 ST,

a) Date and time of the event (obtained from the underlying operating system)

b) User identity (if applicable); and

c) (in Labeled Security Mode) The sensitivity labels of subjects, objects, or information involved; and

d) The additional information specified in the "Details" column of table 6 Auditable Events.

Application note: Each SMF record has a standard header that includes the ID of the job that caused the event. The ID of the job is related to the user ID under which the job has been started by SMF. Also, in cases of a client authenticating using a digital certificate, RACF certificate mapping rules are used to assign an administrator-specified ID rather than a unique ID. The audit records will contain the administrator-specified ID and the X500-based distinguished name from the client's digital certificate for accountability purposes. In the case where a user has been authenticated by a remote trusted system and the ID of the user on the remote system has been mapped to a RACF userID using the identity mapping function, audit records generated for that user will contain both the RACF userID and the remote identity that mapped to the RACF userID.

Application note: RACF requires SMF to be active and configured to be able to record the RACF related audit records. In this case the RACF initialization marks the audit record for the start of the (RACF-related) audit system and the record for stopping RACF marks the shutdown of the audit system

User identity association (FAU_GEN.2)

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.

Application Note: As mentioned above audit records generated for users that have been authenticated by a remote trusted IT system and where the remote userID has been mapped to a RACF userID will contain both the RACF userID as well as the remote userID, allowing to trace the real user even in cases where different remote userIDs have been mapped to the same RACF userID.

Audit review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide **users with access rights to the data set containing the audit records** with the capability to read **all audit information** 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.

Selective audit (FAU_SEL.1)

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) Type of audit event;

b) user identity;

c) Outcome (success or failure) of the audit event;

d) Named object identity;

e) subject sensitivity label; (Labeled Security Mode only)

f) object sensitivity label; (Labeled Security Mode only)

7.2 Cryptographic support (FCS)

Cryptographic operation (FCS_COP.1)

FCS_COP.1.1 The TSF shall perform **digital signature generation and digital signature verification** in accordance with ***the following*** cryptographic algorithm **SHA256withRSAEncryption** and cryptographic key sizes **1024, 2048, and 4096 bit** that meet the following: **X.509 v3 certificates and signatures according to PKCS#1 V2.1 (June 14, 2002)..**

Application Note: RACF can digitally sign a program and verify the digital signature of a signed program. To do this RACF uses an implementation of the RSA algorithm that is identical to the one used by the software cryptographic service provider in ICSF. Since ICSF itself is a signed program, RACF requires to have the implementation of RSA as part of RACF in order to verify the signature of ICSF when this component is loaded.

7.3 User Data Protection (FDP)

7.3.1 General resource class and data set access control policy

Subset access control: (FDP_ACC.1(GRD))

FDP_ACC.1.1(GRD) The TSF shall enforce the **RACF general resource class and data set access control policy** on **subjects represented by an ACEE, the defined resources in the classes listed in tables 7 to 15 below as objects and the RACF-defined access modes of ALTER, CONTROL, UPDATE, READ, EXECUTE, and NONE as operations among subjects and objects covered by the SFP.**

Application Note: RACF uses profiles in the resource classes to identify a profile that matches a resource name specified.

Application Note: Except where noted in the table, RACF does not imply any semantics to the classes and the profiles in the classes. The semantics of the profiles (i. e., what resource they protect and what the semantics of the different access modes is) is completely left to the resource managers that use those classes to protect access to their resources. Profile classes that are used by RACF itself are marked in red. Profile classes listed in black are those defined by IBM and used outside of RACF. Since an installation may define its own profile classes and use them in their applications, the set of profile classes is an open set and profiles not used by RACF itself are user data (as far as RACF is concerned).

Application Note: RACF uses RACF profiles to control access to its own resources or control the use of RACF privileges. Classes listed in red contain profiles used by RACF, but in some classes not all profiles in the class may be used by RACF. For example RACF uses only some profiles in the FACILITY class, while other profiles in this class are used by other parts of z/OS.

Application Note: The following table only defines the general resource classes. There are also profiles in the USER, GROUP, and DATASET classes, which have their own profile format. The access control policy covers the general resource classes listed here and resources in the DATASET class.

Class Name	Purpose
ALCSAUTH	Supports the Airline Control System/MVS (ALCS/MVS) product.
APPCLU	Verifying the identity of partner logical units during VTAM session establishment.
APPCPORT	Controlling which user IDs can access the system from a given LU (APPC port of entry). Also, conditional access to resources for users entering the system from a given LU.
APPCSERV	Controlling whether a program being run by a user can act as a server for a specific APPC transaction program (TP).
APPCSI	Controlling access to APPC side information files.
APPCTP	Controlling the use of APPC transaction programs.
APPL	Controlling access to applications.
CACHECLS	Contains profiles used for saving and restoring cache contents from the RACF database.
CBIND	Controlling the client's ability to bind to the server.

Class Name	Purpose
CDT	Contains profiles for installation-defined classes for the dynamic CDT.
CFIELD	Contains profiles that define the installation's custom fields.
CONSOLE	Controlling access to MCS consoles. Also, conditional access to other resources for commands originating from an MCS console.
DASDVOL	DASD volumes.
DBNFORM	Reserved for future IBM use.
DEVICES	Used by MVS allocation to control who can allocate devices such as: 35/17 Unit record devices (printers and punches) (allocated only by PSF, JES2, or JES3) 35/17 Graphics devices (allocated only by VTAM) Teleprocessing (TP) or communications devices (allocated only by VTAM)
DIGTCERT	Contains digital certificates and information related to them. See chapter 19 of [RACF.SAG] and the description of the RACDCERT command.
DIGTCRIT	Specifies additional criteria for certificate name filters. See chapter 19 of [RACF.SAG] and the description of the RACDCERT command.
DIGTNMAP	Mapping class for certificate name filters. See chapter 19 of [RACF.SAG] and the description of the RACDCERT command.
DIGTRING	Contains a profile for each key ring and provides information about the digital certificates that are part of each key ring. See chapter 19 of [RACF.SAG] and the description of the RACDCERT command.
DIRAUTH	Setting logging options for RACROUTE REQUEST=DIRAUTH requests. Also, if the DIRAUTH class is active, security label authorization checking is done when a user receives a message sent through the TPUT macro or the TSO SEND, or LISTBC commands.
DLFCLASS	The data lookaside facility.
FACILITY	Miscellaneous uses. Profiles are defined in this class so resource managers (typically elements of z/OS or z/VM) can check a user's access to the profiles when the user takes some action. Examples are the profiles used to control execution of RACDCERT command functions and the profiles used to control privileges in the z/OS UNIX environment. RACF does not document all of the resources used in the FACILITY class by other products. For information on the FACILITY class resources used by a specific product (other than RACF itself), see that product's documentation.
FIELD	Fields in RACF profiles (field-level access checking).
FSACCESS	Allows control of access to the root of a zFS file system through RACF resource profiles rather than UNIX permission bits or ACLs.
GDASDVOL	Resource group class for DASDVOL class.
GLOBAL	Global access checking table entry.
GMBR	Member class for the GLOBAL class.
GSDSF	Resource group class for SDSF class.
GTERMINL	Resource group class for TERMINAL class.
GXFACILI	Grouping class for XFACILIT resources.

Class Name	Purpose
IBMOPC	Controlling access to OPC/ESA subsystems.
IDIDMAP	Contains distributed identity filters created with the RACMAP command. See chapter 25 of [RACF.SAG] and the description of the RACMAP command.
JESINPUT	Conditional access support for commands or jobs entered into the system through a JES input device.
JESJOBS	Controlling the submission and cancellation of jobs by job name.
JESSPOOL	Controlling access to job data sets on the JES spool (that is, SYSIN and SYSOUT data sets).
KEYSMSTR	Contains profiles that hold keys to encrypt data stored in the RACF database, such as LDAP BIND passwords and DCE passwords.
LDAPBIND	Contains the LDAP server URL, bind distinguished name, and bind password.
LOGSTRM	Controls system logger resources, such as log streams and the coupling facility structures associated with log streams.
NODES	Controlling the following on MVS systems: 35 17 Whether jobs are allowed to enter the system from other nodes 35 17 Whether jobs that enter the system from other nodes have to pass user identification and password verification checks
NODMBR	Member class for the NODES class.
OPERCMDS	Controlling who can issue operator commands (for example, JES and MVS, and operator commands).
PMBR	Member class for the PROGRAM class.
PROGRAM	Protects executable programs.
PROPCNTL	Controlling if user ID propagation can occur, and if so, for which user IDs (such as the CICS or IMS main task user ID), user ID propagation is not to occur.
PSFMPL	Used by PSF to perform security functions for printing, such as separator page labeling, data page labeling, and enforcement of the user printable area.
PTKTDATA	PassTicket key class enables the security administrator to associate a RACF secured sign-on secret key with a particular mainframe application that uses RACF for user authentication. Examples of such applications are IMS, CICS, TSO, z/VM, APPC, and MVS batch.
RACFEVNT	Contains profiles that control the following events: 35 17 LDAP change log notification for changes to certain RACF profiles New password and password phrase enveloping for a given user.
RACFHC	Used by IBM Health Checker for z/OS. Contains profiles that list the resources to check for each installation-defined health check.
RACFVARS	RACF variables. In this class, profile names, which start with & (ampersand), act as RACF variables that can be specified in profile names in other RACF general resource classes. See [RACF.SAG], chapter 7.
RACGLIST	Class of profiles that hold the results of RACROUTE

Class Name	Purpose
	REQUEST=LIST,GLOBAL=YES or a SETROPTS RACLIST operation.
RACHCMBR	Used by IBM Health Checker for z/OS. Member class for the RACHCMBR class.
RDATALIB	Used to control use of the R_datalib callable service (IRRSDL00 or IRRSDL64).
RRSFDATA	Used to control RACF remote sharing facility (RRSF) functions.
RVARSMBR	Member class for the RACFVARS class.
SCDMBR	Member class for the SECDATA class.
SDSF	Controls the use of authorized commands in the System Display and Search Facility (SDSF). See also GSDSF class.
SECDATA	Security classification of users and data (security levels and security categories).
SECLABEL	If security labels are used, and, if so, their definitions.
SECLMBR	Member class for the SECLABEL class.
SERVAUTH	Contains profiles used by servers to check a client's authorization to use the server or to use resources managed by the server. Also, can be used to provide conditional access to resources for users entering the system from a given server.
SERVER	Controlling the server's ability to register with the daemon.
SMESSAGE	Controlling to which users a user can send messages (TSO only).
SOMDOBJS	Controlling the client's ability to invoke the method in the class.
STARTED	Used in preference to the started procedures table to assign an identity during the processing of an MVS START command.
SURROGAT	If surrogate submission is allowed, and if allowed, which user IDs can act as surrogates.
SYSMVIEW	Controlling access by the SystemView for MVS Launch Window to SystemView for MVS applications.
TAPEVOL	Tape volumes.
TEMPDSN	Controlling who can access residual temporary data sets.
TERMINAL	Terminals (TSO or z/VM). See also GTERMINL class.
VTAMAPPL	Controlling who can open ACBs from non-APF authorized programs.
WRITER	Controlling the use of JES writers.
XFACILIT	Miscellaneous uses. Profile names in this class can be longer than 39 characters in length. Profiles are defined in this class so that resource managers (typically elements of z/OS) can check a user's access to the resources when the users take some action.

Table 7: General Resource Classes

DB2 Classes

DSNADM	DB2 administrative authority class.
DSNR	Controls access to DB2 subsystems.

GDSNBP	Grouping class for DB2 buffer pool privileges.
GDSNCL	Grouping class for DB2 collection privileges.
GDSNDB	Grouping class for DB2 database privileges.
GDSNJR	Grouping class for Java archive files (JARs).
GDSNPK	Grouping class for DB2 package privileges.
GDSNPN	Grouping class for DB2 plan privileges.
GDSNSC	Grouping class for DB2 schemas privileges.
GDSNSG	Grouping class for DB2 storage group privileges.
GDSNSM	Grouping class for DB2 system privileges.
GDSNSP	Grouping class for DB2 stored procedure privileges.
GDSNSQ	Grouping class for DB2 sequences.
GDSNTB	Grouping class for DB2 table, index, or view privileges.
GDSNTS	Grouping class for DB2 tablespace privileges.
GDSNUF	Grouping class for DB2 user-defined function privileges.
GDSNUT	Grouping class for DB2 user-defined distinct type privileges.
MDSNBP	Member class for DB2 buffer pool privileges.
MDSNCL	Member class for DB2 collection privileges.
MDSNDB	Member class for DB2 database privileges.
MDSNJR	Member class for Java archive files (JARs).
MDSNPK	Member class for DB2 package privileges.
MDSNPN	Member class for DB2 plan privileges.
MDSNSC	Member class for DB2 schema privileges.
MDSNSG	Member class for DB2 storage group privileges.
MDSNSM	Member class for DB2 system privileges.
MDSNSP	Member class for DB2 stored procedure privileges.
MDSNSQ	Member class for DB2 sequences.
MDSNTB	Member class for DB2 table, index, or view privileges.
MDSNTS	Member class for DB2 tablespace privileges.
MDSNUF	Member class for DB2 user-defined function privileges.
MDSNUT	Member class for DB2 user-defined distinct type privileges.

Table 8: General Resource Classes for DB2

Enterprise Identity Mapping (EIM) Classes

RAUDITX	Controls auditing for Enterprise Identity Mapping (EIM).

Table 9: General Resource Classes for EIM

ICSF Classes

CRYPTOZ	Controls access to PKCS #11 tokens.

CSFKEYS	Controls access to ICSF cryptographic keys.
CSFSERV	Controls access to ICSF cryptographic services.
GCSFKEYS	Resource group class for the CSFKEYS class.
GXCSFKEY	Resource group class for the XCSFKEY class.
XCSFKEY	Controls the exportation of ICSF cryptographic keys.

Table 10: General Resource Classes for ICSF

Infoprint Classes

PRINTSRV	Controls access to printer definitions for Infoprint Server.

Table 11: General Resource Classes for Infoprint

Network authentication (Kerberos) Classes

KERBLINK	Mapping class for user identities of local and foreign principals.
REALM	Used to define the local and foreign realms.

Table 12: General Resource Classes for Kerberos

DFSMS Classes

MGMTCLAS	SMS management classes.
STORCLAS	SMS storage classes.
SUBSYSNM	Authorizes a subsystem (such as a particular instance of CICS) to open a VSAM ACB and use VSAM record level sharing (RLS) functions.

Table 13: General Resource Classes for DFSMS

TSO Classes

ACCTNUM	TSO account numbers.
PERFGRP	TSO performance groups.
TSOAUTH	TSO user authorities such as OPER and MOUNT.
TSOPROC	TSO logon procedures.

Table 14: General Resource Classes for TSO

z/OS UNIX Classes

DIRACC	Controls auditing (using SETROPTS LOGOPTIONS) for access checks for read/write access to z/OS UNIX directories. This class need not be active to control auditing.
DIRSRCH	Controls auditing (using SETROPTS LOGOPTIONS) of z/OS UNIX directory searches. This class need not be active to control auditing.
FSOBJ	Controls auditing (using SETROPTS LOGOPTIONS) for all access checks for z/OS UNIX file system objects except directory searches. Controls auditing (using SETROPTS AUDIT) of creation and deletion of z/OS UNIX file system objects. This class need not be active to control auditing.
FSSEC	Controls auditing (using SETROPTS LOGOPTIONS) for changes to the security data (FSP) for z/OS UNIX file system objects. This class need not be active to control auditing. When this class is active, it also controls whether

	ACLs are used during authorization checks to z/OS UNIX files and directories.
IPCOBJ	Controls auditing (using SETROPTS LOGOPTIONS) of access checks for inter-process communication (IPC) objects and changes to security information of IPC objects. Controls auditing (using SETROPTS AUDIT) of the creation and deletion of IPC objects. This class need not be active to control auditing.
PROCACT	Controls auditing (using SETROPTS LOGOPTIONS) of functions that look at data from, or affect the processing of, z/OS UNIX processes. This class need not be active to control auditing.
PROCESS	Controls auditing (using SETROPTS LOGOPTIONS) of changes to UIDs and GIDs of z/OS UNIX processes. Controls auditing (using SETROPTS AUDIT) of dubbing and undubbing of z/OS UNIX processes. This class need not be active to control auditing.
UNIXMAP	Contains profiles that are used to map z/OS UNIX UIDs to RACF user IDs and z/OS UNIX GIDs to RACF group names.
UNIXPRIV	Contains profiles that are used to grant z/OS UNIX privileges.

Table 15: General Resource Classes for z/OS UNIX

Security attribute based access control (FDP_ACF.1(GRD))

FDP_ACF.1.1(GRD) The TSF shall enforce the **RACF general resource class and data set access control policy** to objects based on the following: **users represented by an ACEE as subjects, profiles in one of the resource classes as representatives of objects, using the user's name, the RACF attributes stored in the ACEE, the user's group memberships, the ACL associated with the resource profile, the value of UACC stored in the resource profile, and the entry for the resource containing the object in the global access checking table as security attributes.**

FDP_ACF.1.2(GRD) The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

a subject's requested type of access to a protected resource is granted or denied by the following algorithm, if the resource class is known and active, and if a matching resource profile in the class was found,

a) if access is allowed by global access checking (Note: does not apply for user with the RESTRICTED attribute; does not apply to checks performed by RACROUTE REQUEST=FASTAUTH)

or, if a) is not true,

b) (in Labeled Security Mode) if the access is not denied by the mandatory access control

if a) did not grant access, and b) did not deny access,

c) if the resource is a tape or DASD data set and the high-level qualifier of the data set name is identical to the user ID

if c) did not grant access,

d) if the requested type of access is allowed by an access control list (ACL) entry for this particular user (Note: does not apply to checks performed by RACROUTE REQUEST=FASTAUTH with the AUTHCHKS=CRITONLY option)

if d) neither granted nor denied access then continue with e) Otherwise, if d) denied access, continue with h),

e) if the requested type of access is allowed by an ACL entry for the group the user belongs to. If list-of-groups processing is not in effect, this rule is evaluated only for the current connect group. Otherwise, this rule is evaluated for all groups to which the user is connected. (Note: does not apply to checks performed by RACROUTE REQUEST=FASTAUTH with the AUTHCHKS=CRITONLY option)

if no entries in e) granted access, and no entries in e) denied access, then continue with f). Otherwise, if at least one entry in e) denied access, then continue with h),

f) if the user does not have the RESTRICTED attribute and the requested type of access is granted by the universal access authority (UACC) in the profile protecting the resource or granted by an ACL with ID(*)(Note: does not apply to checks performed by RACROUTE REQUEST=FASTAUTH with the AUTHCHKS=CRITONLY option)

if f) did not grant access,

g) if the user has the OPERATIONS role or the group-OPERATIONS role (for a group to which the user is connected and the resource is within the group's scope) and OPERATIONS access is allowed for the class

if g) did not grant access,

h) if the user has an entry in the conditional access list for the profile that allows the requested type of access and the user meets the condition defined in this conditional access list entry (Note: for checks performed by RACROUTE REQUEST=FASTAUTH with the AUTHCHKS=CRITONLY option, only conditional access list entries specifying WHEN(CRITERIA(SQLROLE...)) will apply)

or, if h) did not grant access,

i) if the user is a member of a group that has an entry in the conditional access list for the profile that allows the requested type of access and the user meets the condition defined in this conditional access list entry. If list-of-groups processing is not in effect, this rule is evaluated only for the current connect group. Otherwise, this rule is evaluated for all groups to which the user is connected. (Note: for checks performed by RACROUTE REQUEST=FASTAUTH with the AUTHCHKS=CRITONLY option, only conditional access list entries specifying WHEN(CRITERIA(SQLROLE...)) will apply)

or, if i) did not grant access,

j) if a conditional access list entry for ID(*) exists with requested type of access, the user does not have the RESTRICTED attribute set and the user satisfies the condition of the conditional access list entry. (Note: for checks performed by RACROUTE REQUEST=FASTAUTH with the AUTHCHKS=CRITONLY option, only conditional access list entries specifying WHEN(CRITERIA(SQLROLE...)) will apply).

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

a) **the subject is a trusted subject and has specified a nested ACEE in its call to RACF with a second user ID. In this case access is allowed if either the primary user ID specified in the first ACEE or the secondary user ID specified in the nested ACEE has the requested access right to the object and the object has been designated as eligible for nested ACEE processing and the authorization check is made using RACROUTE REQUEST=FASTAUTH.**

b) **when "program control" is activated (using the WHEN(PROGRAM) option in the SETROPTS command) and the program is protected by a profile in the PROGRAM class and the user has at least EXECUTE access to this profile, the user can execute the program in a clean z/OS environment not "contaminated" by any untrusted program. If the user has at least READ access then untrusted programs may also be used by the user.**

c) **when "program control" is activated and "PADCHK" has been defined in the profile for a program, a user may access a data set via PADS if the program that attempts the access or a higher program in the execution hierarchy is allowed to access the file in the intended mode by the conditional access list for the data set and all other active programs not from the link pack area that have been defined using the WHEN PROGRAM operand with "PADCHK" are included in the conditional access list of the data set. While a data set is open using PADS, for any new program defined with PADCHK and started in this situation in the same environment, the TOE checks that the new program is also in the conditional access list of that data set.**

Application note: "trusted" in this sense means "defined to RACF via profiles in the PROGRAM class, or resident in the system link pack area.

FDP_ACF.1.4(GRD) The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **data sets that are not protected by a discrete or generic profile can not be accessed except by users with the SPECIAL role.**

7.3.2 UNIX file system object access control policy

Subset access control: (FDP_ACC.1(UFS))

FDP_ACC.1.1(UFS) The TSF shall enforce the **RACF UNIX file system access control policy** on **subjects represented by a UNIX System Services Credentials Structure (CRED), the UNIX file system resources represented by a File Security Packet (FSP) as objects and the UNIX access modes of read, write, execute (for ordinary files), and search (for directories) as operations among subjects and objects covered by the SFP.**

Application Note: The File Security Packet needs to be created by the caller. It contains the UID and GID of the owner of the file system object, the permission bits, and other file system object security attributes associated with the file system object. RACF provides the makeFSP callable service to create a FSP.

Application Note: The UNIX System Services Credentials Structure (CRED) needs to be created by the caller. It contains information about the user that initiated the request but also information about the object being accessed (including the ACL for the object)

Security attribute based access control (FDP_ACF.1(UFS))

FDP_ACF.1.1(UFS) The TSF shall enforce the **RACF UNIX file system access control policy** to ***UNIX file system*** objects based on the following: **users represented by a UNIX System Services Credentials Structure (CRED) as subjects, UNIX file system objects represented by**

a File Security Packet (FSP) as objects and the security attributes used in the algorithm defined in FDP_ACF.1.2(UFS).

FDP_ACF.1.2(UFS) The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

The mandatory access control (Labeled Security Mode only) must allow access and the following algorithm for the discretionary access control must also result in granting access.

If the FSACCESS class is active and SETROPTS RACLISTed, and if the subject is accessing the root directory in a mounted zFS file system (but not the system root directory), then if the MVS data set name of the file system container is protected by a profile in the FSACCESS class the subject must have UPDATE access to that FSACCESS profile or the request will fail.

A subject must have search permission for every element of the path name and the requested access for the object. A subject has a specific type of access to an object if:

a) the user has the AUDITOR attribute, the requested type of access is READ or SEARCH, and the object is a directory.

b) the effective user ID is 0 and the requested type of access is not EXECUTE. If this is the case, access is granted. If the effective user ID is 0, and the requested type of access is EXECUTE access is granted only if there is either a permission bit, or an ACL that provides EXECUTE access to any user.

c) the effective user ID is the one of the file owner and has been granted access according to the owner permission bits, access is granted.

d) the FSSEC class is active in RACF and an ACL exists within the set of ACLs for the file that grants the required type of access to the requesting user, access is granted.

e) the effective user ID is the one of the owner of the file, the algorithm continues with step j.

f) the effective group ID (GID) or any of the user's supplemental GIDs matches the group of the file and has the requested type of access defined in the group permission bits, access is granted.

g) the effective GID or any of the user's supplemental GIDs has an ACL defined for the file that allows the requested type of access, access is granted.

h) the requested type of access is defined in the "other" permission bits and the user does not have the RESTRICTED attribute defined in his profile, access is granted.

i) the user has the RESTRICTED attribute defined and has the requested type of access defined in the RESTRICTED.FILESYS.ACCESS resource profile and the ACLs associated with this profile, access is granted.

j) the user has the RESTRICTED attribute defined, the RESTRICTED.FILESYS.ACCESS profile is not defined in RACF, and the requested type of access is allowed according to the "other" permission bits, access is granted.

k) the UNIXPRIV class is active and RACLISTed, and if the SUPERUSER.FILESYS.ACLOVERRIDE resource is protected by a profile in the UNIXPRIV

class, then the user must have the correct access level as documented for the ck_access (IRRSKA00) callable service in z/OS Security Server: RACF Callable Services. If the profile exists, it determines whether file access is granted or denied.

l) If this step of the algorithm is reached and no decision for granting or denying access has been made, access is denied.

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

a) the object is a z/OS UNIX file system object, the UNIXPRIV class is active in RACF, the access was denied by an ACL entry and the user has the requested type of access to the file defined as access to the SUPERUSER.-FILESYS.ACLOVERRIDE profile

or

b) the object is a z/OS UNIX file system object, the UNIXPRIV class is active in RACF, the access was denied by the permission bits, the SUPERUSER.-FILESYS.ACLOVERRIDE profile is not defined in the UNIXPRIV class and the user has the requested type of access to the SUPERUSER.FILESYS profile, that is, if the user wants to read the file, the user must have read access to the profile, if the user wants to read and write the file, the user must have write access to the profile, if the user wants to update any directory, the user must have control access.

FDP_ACF.1.4(UFS) The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **none.**

7.3.3 UNIX IPC access control policy

Subset access control: (FDP_ACC.1(IPC))

FDP_ACC.1.1(IPC) The TSF shall enforce the **RACF UNIX IPC access control policy** on **subjects represented by a UNIX System Services Credentials Structure for IPC (CREI), the UNIX IPC resources represented by an IPC Security Packet (ISP) as objects and the UNIX access modes of read and write as operations among subjects and objects covered by the SFP.**

Application Note: The ISP needs to be created by the caller. It contains the UID and GID of the owner and creator of the IPC object, the permission bits, and the security label associated with the IPC object. RACF provides the makeISP callable service that creates an ISP.

Application Note: The UNIX System Services Credentials Structure for IPC (CREI) needs to be created by the caller. It contains information about the user that initiated the request but also information about the object being accessed (including the ACL for the object).

Security attribute based access control (FDP_ACF.1(IPC))

FDP_ACF.1.1(IPC) The TSF shall enforce the **RACF UNIX IPC access control policy** to ***UNIX IPC*** objects based on the following:

a) The z/OS UNIX user identity and group membership(s) associated with a subject; and

b) The following access control attributes associated with an object: permission bits.

Access rights for z/OS UNIX IPC objects are:

a) read

b) write

Access is defined by permission bits only.

FDP_ACF.1.2(IPC) The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

The mandatory access control (Labeled Security Mode) must allow access and the following algorithm for the discretionary access control must also result in granting access.

Access permissions are defined by permission bits of the IPC object only. IPC objects don't have ACLs associated with them. The process creating the object defines the creator, owner, and group based on the user ID of the current process. Access of a process to an IPC object is allowed if:

a) access is allowed by the mandatory access control (Labeled Security Mode) and the following algorithm:

a. the effective UID of the current process is equal to zero or the UID of the IPC object creator or owner and the "owner" permission bit for the requested type of access is set or,

b. the effective UID of the current process is not equal to the UID of the IPC object creator or owner and the effective GID of the current process or any supplementary z/OS UNIX GIDs the user is a member of is equal to the GID of the IPC object and the "group" permission bit for the requested type of access is set or,

c. the "other" permission bit for the requested type of access is set for users who do not satisfy one of the first two conditions.

FDP_ACF.1.3(IPC) The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **none**.

FDP_ACF.1.4(IPC) The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **none**.

7.3.4 RACF Field-level access control policy

Subset access control: (FDP_ACC.1(FLA))

FDP_ACC.1.1(FLA) The TSF shall enforce the **RACF Profile Field-Level Access Control Policy** on

a) Subjects: RACF commands executed on behalf of a user

b) Objects: fields in segments other than the base segment in RACF profiles

c) Operations: reading or modifying dedicated fields in segments other than the base segment in RACF profiles using RACF commands or the R_admin callable service.

Application Note: Field-level access control allows an installation to define access to fields within segments other than the base segment of RACF profiles. Access to those fields is via the RACF commands used to manage/list RACF profiles or via the R_admin callable service.

Security attribute based access control (FDP_ACF.1(FLA))

FDP_ACF.1.1(FLA) The TSF shall enforce the **RACF Profile Field-Level Access Control Policy** to objects based on the following:

a) The RACF user identity and group membership(s) associated with the user executing the RACF command; and

b) The access rights defined in a RACF profile in the FIELD class of the type *profile-type.segment-name.field-name* where *profile-type* is either the class name for a general resource profile, DATASET, GROUP, or USER; *segment-name* is the name of a valid segment for the profile type; and *field-name* is the name of a defined segment field in the segment type that corresponds to the command operand controlling that field.

Application Note: The complete list of field names for the segments of the individual profile types is provided in Table 18 of [RACF.SAG]. Access to specific fields in segments within RACF profiles requires of course that the related segment is defined for the profile type. Table 18 of [RACF.SAG] also identifies which segments exist for which profile type and maps the field names of those segments to the RACF command parameter used to update those fields.

FDP_ACF.1.2(FLA) The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

a subject has the requested type of access to a field in a segment of a RACF profile, if access is allowed by RACF based on the information in the related profile in the FIELD class using the RACF algorithm for determining access to general resources (see FDP_ACF.1(GRD)) using the access control information from the profile.

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

a) users with the SPECIAL attribute always have READ and UPDATE access to all fields in all non-base segments that are defined for the different RACF profile types.

b) users with the AUDITOR attribute always have READ access to all fields in all non-base segments that are defined for the different profile types.

c) when the FIELDS class is active and SETROPTS RACLIST processing is active for the FIELDS class, users are allowed to read or modify fields in their own user profile when ID(&RACUID) is specified in the PERMIT command defining access.

FDP_ACF.1.4(FLA) The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **users without the SPECIAL attribute or AUDITOR attribute have no access to fields in RACF segments until the FIELDS class is active and SETROPTS RACLIST processing is activated for the FIELDS class.**

7.3.5 Mandatory access control policy

Complete information flow control: labeled security (FDP_IFC.2) (Labeled Security Mode only)

FDP_IFC.2.1 The TSF shall enforce the **mandatory access control policy** on

a) Subjects represented by an ACEE, CRED or CREI;

b) Objects represented by a RACF profile, a File Security Packet (FSP), or an IPC Security Packet (ISP)

and all operations that cause that information to flow from subjects covered by the SFP.

FDP_IFC.2.2 The TSF shall ensure that all operations that cause any information in the TOE to flow ***among untrusted subjects and named objects*** in the TOE are covered by ***the mandatory access control policy***.

Hierarchical security attributes (FDP_IFF.2) (Labeled Security Mode only)

FDP_IFF.2.1 The TSF shall enforce the **mandatory access control policy** based on the following types of subject and ***object*** security attributes:

a) Subject security attributes:

Sensitivity label of the subject consisting of at least 8 site definable hierarchical levels and a set of 60 site definable non-hierarchical categories;

b) Object security attributes:

the sensitivity label of the object consisting of at least 8 site definable hierarchical levels and a set of 60 site definable non-hierarchical categories;

FDP_IFF.2.2 The TSF shall permit an information flow between a controlled subject and ***a controlled object*** via a controlled operation if the following rules, based on the ordering relationships between security attributes hold:

a) If the sensitivity label of the subject is greater than or equal to the sensitivity label of the object, then the flow of information from the object to the subject is permitted (a read operation);

b) If the sensitivity label of the object is greater than or equal to the sensitivity label of the subject; then the flow of information from the subject to the object is permitted (a write operation);

c) If the information flow is between objects, the sensitivity label of the destination object must be greater than or equal to the sensitivity label of the source object.

FDP_IFF.2.3 The TSF shall enforce **no additional information flow rules.**

FDP_IFF.2.4 The TSF shall explicitly authorize an information flow based on the following rules: **a user is permitted to bypass the information flow policy, if the profile IRR.WRITEDOWN.BYUSER in the FACILITY class exists and is active and the user has at least read access to it.**

FDP_IFF.2.5 The TSF shall explicitly deny an information flow based on the following rules: **objects that are supposed to have a security label but do not have a security label can not be accessed.**

FDP_IFF.2.6 The TSF shall enforce the following relationships for any two valid information flow control security attributes:

a) There exists an ordering function that, given two valid security attributes, determines if the security attributes are equal, if one security attribute is greater than the other, or if the security attributes are incomparable ***with the following properties:***

   a. Sensitivity labels are equal if the hierarchical level of both labels are equal and the non-hierarchical category sets are identical;

   b. Sensitivity label A is greater than sensitivity label B if the hierarchical level of A is greater than or equal to the hierarchical level of B, and the non- hierarchical category set of A is equal to or a superset of the non- hierarchical category set of B;

   c. Sensitivity labels are incomparable if they are not equal and neither label is greater than the other as defined in 1 and 2 above;

b) There exists a “least upper bound” in the set of security attributes, such that, given any two valid security attributes, there is a valid security attribute that is greater than or equal to the two valid security attributes; and

c) There exists a “greatest lower bound” in the set of security attributes, such that, given any two valid security attributes, there is a valid security attribute that is not greater than the two valid security attributes.

7.4 Identification and authentication (FIA)

Authentication failure handling (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when **an administrator configurable positive integer within the range 1 to 255** ***number of consecutive*** unsuccessful authentication attempts ***for the authentication methods passwords, password phrases and RACF PassTickets*** occur related to **all authentication events using these authentication methods**.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been **surpassed**, the TSF shall **set the user status to REVOKE**.

User attribute definition: human users (FIA_ATD.1(HU))

FIA_ATD.1.1(HU) The TSF shall maintain the following list of security attributes belonging to individual ***human*** users:

a) User identifier; c

b) Group memberships;

c) User password and optionally a password phrase; c

d) Security roles;

e) default access rights for objects created by the user (UACC); c

f) classes in which the user can define profiles (CLAUTH); c

g) indicator that global access checking, the ID(*) entry on the access list, and the UACC will not be used to allow this user access to a protected resource (RESTRICTED);

h) z/OS UNIX UID (for users also defined to UNIX System Services); c

i)

j) Kerberos principal name (for users defined to the z/OS Network Authentication Service and for foreign Kerberos principals that are defined to a Kerberos realm that has a cross realm trust relationship with the z/OS Network Authentication Service); c

k) Kerberos ticket maximum lifespan for users defined to the z/OS Network Authentication Service; c

l) indicator of the encryption algorithm used by the z/OS Network Authentication Service; c

m) X.509v3 certificate(s).

Application note: Attributes such as SPECIAL, group-SPECIAL, AUDITOR, group-AUDITOR, OPERATIONS and group-OPERATIONS designate roles in the model of this Security Target and are therefore further explained in the role model in FMT_SMR.1

User attribute definition: labeled security (FIA_ATD.1(LS)) (Labeled Security Mode only)

FIA_ATD.1.1(LS) The TSF shall maintain the following list of security attributes belonging to individual users:

a) Sensitivity label,

b) user clearances.

Verification of secrets (FIA_SOS.1)

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet **the following quality metric:**

a) the probability that a secret can be obtained by an attacker during the lifetime of the secret is less than 2^{-20}

Application note: Some authentication functions depend on cryptographic functions, such as certificate-based client authentication. No strength of function analysis is provided in this ST for these, nor for any cryptographic key generation functions that may be a part of the identification and authentication mechanisms.

Timing of authentication (FIA_UAU.1)

FIA_UAU.1.1 The TSF shall allow **all functions allowed to be performed by the individual pseudo-user assigned by the authorized administrator for started procedures (started tasks)** on behalf of the user to be performed before the user is authenticated.

Application Note: Trusted applications can request the creation of an ACEE. It is up to the trusted application to ensure that the user has been successfully authenticated via RACF.

In z/OS, predefined jobs known as started procedures (or started tasks) may be started automatically, or by an operator who has the required privileges. Those started tasks operate under a pseudo-user-ID assigned to them by the system administrator when the started task job was created and stored in a protected data set. RACF allows the definition of protected user IDs for this purpose. Protected user IDs don't have a password or password phrase associated with them and cannot be authenticated using RACF. They need to be defined in RACF and they are bound by the same RACF access control rules as a normal user. Activities performed by such a started task are accounted to the pseudo-user-ID assigned to them and not with the ID of the operator that started those tasks (because, in most cases, the operator would not know what those started tasks are doing and the operator would not be allowed to access the resources that the started tasks needs access to). No "user authentication" is performed for started tasks. Instead, they can only be started from predefined libraries.

FIA_UAU.1.2 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Multiple authentication mechanisms (FIA_UAU.5)

FIA_UAU.5.1 The TSF shall provide **the following authentication mechanisms** to support user authentication:

a) Authentication based on username and password and password phrases;

b) Authentication based on software token verification data (digital certificates);

c) RACF PassTickets

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the **following rules:**

a) Authentication based on username and password, password phrase or RACF PassTicket is performed for authentication requests using the RACROUTE REQUEST=VERIFY, RACROUTE REQUEST=VERIFYX and initACEE functions;

b) Authentication based on software token verification data is performed for authentication requests using the initACEE function;

Protected authentication feedback (FIA_UAU.7)

FIA_UAU.7.1 The TSF shall provide **no feedback** to the user while the authentication is in progress.

Application Note: The "user" in this case is the application that calls RACF to perform user authentication. RACF does not provide any feedback to a caller until the authentication process has finished.

Timing of identification (FIA_UID.1)

FIA_UID.1.1 The TSF shall allow **all functions allowed to be performed by the individual pseudo-user assigned by the authorized administrator for started procedures (started tasks)** on behalf of the user to be performed before the user is identified.

FIA_UID.1.2 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

User-subject binding (FIA_USB.1) (Labeled Security Mode only))

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user:

a) User sensitivity level that is used to enforce the mandatory access control policy which consists of the following:

a. A hierarchical level; and

b. A set of non-hierarchical categories

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:

a) The sensitivity label associated with a subject shall be within the clearance range of the user;

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**

Enhanced user-subject binding (FIA_USB.2)

FIA_USB.2.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user:

a) The RACF user identity that is associated with auditable events;

b) The RACF user security attributes that are used to enforce the RACF general resource class and data set access control policy, the UNIX file system object security policy and the UNIX IPC object security policy;

c) The software token that can be used for subsequent identification and authentication with the TSF or other remote IT systems;

d) Active roles;

e) Active groups;

f) The RACF attributes/roles SPECIAL, group-SPECIAL, AUDITOR, group-AUDITOR, CLAUTH, OPERATIONS, and group-OPERATIONS;

g) The port of entry (POE)

FIA_USB.2.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users:

a) A started task executes with the user ID defined in the started class or started procedures table defining the started task.

b) in the case the user was authenticated by a remote trusted IT system and then mapped to a RACF userID using a profile in the IDIDMAP class: the identity of the user on the remote trusted IT system is also assigned as a security attribute to that user (and later used in audit records generated for that user).

FIA_USB.2.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users:

a) An administrator may define specific z/OS Applications to execute with an administrator defined user ID.

b) An administrator may use the SURROGAT authority mechanism to allow a user to switch his identity to another defined user without specifying the password/phrase for this user.

c) An application executing in supervisor state may change the group IDs of a subject representing a user with the R_setegid, R_setgid, or R_exec callable services according to the following rules:

- **a. R_setegid: If the high-order bit of the input GID is on, the real, effective, and saved GIDs are changed for the current process.**
- **b. R_setegid: If the high-order bit of the input GID is off and if the user is the superuser or if the input GID is equal to the real or saved GID of the calling process, the effective GID of the process is changed to the input GID. The real and saved GIDs are not changed.**
- **c. R_setgid: If the calling process is a superuser, the real, saved, and effective GIDs are changed. If the calling process is not a superuser but the input GID is equal to the real or saved GID, the effective GID of the process is changed. If neither condition is met, the GIDs of the process are not changed.**
- **d. R_exec: sets the effective and saved UNIX group identifier to the specified values (if the call requested a change of the GID)**
- **e. The new GID must be known to RACF.**

d) An application executing in supervisor state may change the user IDs of a subject representing a user with the R_seteuid, R_setuid, or R_exec callable services according to the following rules.

- **a. R_seteuid: If the high-order bit of the input UID is on, the real, effective, and saved UIDs are changed for the current process.**
- **b. R_seteuid: If the high-order bit of the input z/OS UNIX user identifier (UID) is off and if the user is the superuser or if the input UID is equal to the real or saved UID of the calling process, the effective UID of the process is changed to the input UID. The real and saved UIDs are not changed.**
- **c. R_setuid: If the calling process is a superuser, the real, saved, and effective z/OS UNIX user identifiers (UIDs) are changed. If the calling process is not a superuser, but the input UID is equal to the real or saved UID, the effective UID of the process is changed. If neither condition is met, the UIDs of the process are not changed.**
- **d. R_exec: sets the effective and saved UNIX user identifier to the specified values (if the call requested a change of the UID).**

e. The new UID must be known to RACF.

FIA_USB.2.4 The TSF shall enforce the following rules for the assignment of subject security attributes not derived from user security attributes when a subject is created:

a) The Port of Entry (POE) is set to the value specified by the caller of initACEE in the SERVAUTH_name parameter or to the value of the POE or POENET parameter for callers of RACROUTE REQUEST=VERIFY or RACROUTE REQUEST=VERIFYX

7.5 Security management (FMT)

Management of security attributes (FMT_MSA.1(GRD))

FMT_MSA.1.1(GRD) The TSF shall enforce the **RACF general resource class and data set access control policy** to restrict the ability to **modify** the security attributes **(ACLs using the PERMIT command and security attributes that can be modified using the ALTDSD or RALTER commands) of the objects covered by the SFP** to **users that are allowed to use those commands according to the conditions defined in table 38 in the TOE Summary Specification.**

Management of security attributes (FMT_MSA.1(UFS))

FMT_MSA.1.1(UFS) The TSF shall enforce the **UNIX file system object access control policy** to restrict the ability to **modify** the security attributes **(ACLs) of the objects covered by the SFP** to **users that satisfy one of the following conditions:**

a) The user is a superuser (has an UID of 0 or has at least READ access to the resource named SUPERUSER.FILESYS.CHANGEPERMS in the UNIXPRIV class)

b) The user is the owner of the file system object

(Labeled Security Mode only): In addition to one of the conditions above, the current security label of the subject must be greater than or equal to the security label of the file system object or the security label of the file system object must be greater than or equal to the current security label of the process (the labels are not disjoint). Security label checking is bypassed if the ACEE indicates trusted or privileged authority or if the service has passed a system CRED.

Management of security attributes (FMT_MSA.1(IPC))

FMT_MSA.1.1(IPC) The TSF shall enforce the **UNIX IPC object access control policy** to restrict the ability to **modify** the security attributes **(permission bits) of the objects covered by the SFP** to **users that satisfy one of the following conditions:**

a) The user is a superuser (has an UID of 0)

b) The user is the owner or the creator of the IPC object

(Labeled Security Mode only): In addition to one of the conditions above, the current security label of the subject must be greater than or equal to the security label of the IPC object or the security label of the IPC object must be greater than or equal to the current security label of the process (the labels are not disjoint). Se-

curity label checking is bypassed if the ACEE indicates trusted or privileged authority.

Management of security attributes (FMT_MSA.1(FLA))

FMT_MSA.1.1(FLA) The TSF shall enforce the **RACF profile field-level access control policy** to restrict the ability to **modify** the security attributes **defined by profiles in the FIELD class** to **users that are allowed manage profiles in this class and define ACLs to such profiles using the PERMIT command.**

Management of security attributes (FMT_MSA.1(LS)) (Labeled Security Mode only)

FMT_MSA.1.1(LS) The TSF shall enforce the **mandatory access control policy** to restrict the ability to **modify** the security attributes **(security labels) of the objects covered by the SFP** to **users that satisfy one of the following conditions:**

a) **For UNIX file system objects: the caller of the R_setfsecl callable service must be in supervisor state and pass a system CRED, or the caller must have the SPECIAL attribute and no security label is currently assigned to the file system object.**

b) **For UNIX IPC objects: security labels can not be modified.**

c) **For general resource and data set profiles: the user must have the SPECIAL attribute or at least READ access to the security label profile.**

Static attribute initialization (FMT_MSA.3(LS)) (Labeled Security Mode only)

FMT_MSA.3.1(LS) The TSF shall enforce the **mandatory access control policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2(LS) The TSF shall allow **the users with the SPECIAL attribute and the owner of the profile protecting the object** to specify alternative initial values to override the default values when an object or information is created.

Static attribute initialization (FMT_MSA.3(GRD))

FMT_MSA.3.1(GRD) The TSF shall enforce the **RACF general resource class and data set access control policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2(GRD) The TSF shall allow the **users with the SPECIAL attribute and the owner of the profile protecting the object** to specify alternative initial values to override the default values when an object or information is created.

Static attribute initialization (FMT_MSA.3(UFS))

FMT_MSA.3.1(UFS) The TSF shall enforce the **UNIX file system object access control policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2(UFS) The TSF shall allow the **z/OS components allowed to use the R_umask callable service** to specify alternative initial values to override the default values when an object or information is created.

Static attribute initialization (FMT_MSA.3(IPC))

FMT_MSA.3.1(IPC) The TSF shall enforce the **RACF UNIX IPC object access control policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2(IPC) The TSF shall allow the **no user** to specify alternative initial values to override the default values when an object or information is created.

Static attribute initialization (FMT_MSA.3(FLA))

FMT_MSA.3.1(FLA) The TSF shall enforce the **RACF profile field-level access control policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2(FLA) The TSF shall allow the **users that can manage access control lists for the related profiles in the FIELD class (and therefore set the value for UACC for those profiles)** to specify alternative initial values to override the default values when an object or information is created.

Management of TSF data (FMT_MTD.1(SO))

FMT_MTD.1.1(SO) The TSF shall restrict the ability to **initialize or change** the **additional TOE configuration parameters (as set by the SETROPTS command)** to **authorized administrators that satisfy the criteria for using the SETROPTS RACF command as defined in table 38 in the TOE Summary Specification.**

Management of TSF data (FMT_MTD.1(AE))

FMT_MTD.1.1(AE) The TSF shall restrict the ability to **query or modify** the **set of audited events** to

a) users in the AUDITOR role

b) for events related to a profile: the profile owner

Management of TSF data (FMT_MTD.1(UA))

FMT_MTD.1.1(UA) The TSF shall restrict the ability to **initialize, modify, delete** the **user security attributes used for the identification and authentication policy** to **the authorized administrators that satisfy the rules defined for the ADDUSER, ALTUSER, and DELUSER RACF commands as defined in table 38 in the TOE Summary Specification.**

Management of TSF data (FMT_MTD.1(RA))

FMT_MTD.1.1(RA) The TSF shall restrict the ability to **re-enable** the **authentication to the account subject to authentication failure** to **users that satisfy the rules defined for the use of the RESUME operand in the ALTUSER RACF command in table 38 in the TOE Summary Specification**.

Management of TSF data (FMT_MTD.1(TH))

FMT_MTD.1.1(TH) The TSF shall restrict the ability to **modify** the **threshold for unsuccessful authentication attempts** to **users that satisfy one of the following rules:**

a) user has SPECIAL

b) user has group-SPECIAL in the group that owns the user, or group-SPECIAL in a higher group in the group tree if group ownership is setup appropriately.

Management of TSF data (FMT_MTD.1(AD))

FMT_MTD.1.1(AD) The TSF shall restrict the ability to **initialize, modify, delete** the **user security attributes authentication data** to **users that satisfy the following rules:**

a) users authorized to modify their own authentication data

b) Users with the SPECIAL or appropriate group-SPECIAL attribute can modify a user's password/phrase;

c) Users with access to FACILITY resource IRR.PASSWORD.RESET are allowed to reset passwords/phrases for any user that does not have the PROTECTED, SPECIAL, AUDITOR, or OPERATIONS attributes;

d) **Users with access to FACILITY resource IRR.PWRESET.OWNER.owner-value are allowed to reset passwords/phrases for users owned by "owner-value" if those users do not have the PROTECTED SPECIAL, AUDITOR, or OPERATIONS attributes and are not exempted from reset by the IRR.PWRESET.EXCLUDE.userID resource in the FACILITY class;**

e) Users with access to FACILITY resource IRR.PWRESET.TREE.owner-value are allowed to reset passwords/phrases for users in the scope of the group specified by "owner-value" if those users do not have the PROTECTED SPECIAl, AUDITOR, or OPERATIONS attributes and are not exempted from reset by the IRR.PWRESET.EXCLUDE.userID resource in the FACILITY class. (Note: this "tree" function applies to the same target users that group-SPECIAL would affect.);

f) Users may be allowed to renew or revoke their own digital certificates via the z/OS PKI Services component.

Management of TSF data (FMT_MTD.1(RC))

FMT_MTD.1.1(RC) The TSF shall restrict the ability to **manage** the **TSF data operated upon by other RACF commands** to **users with the authority to those commands as defined in the table 38 in the section on using RACF management commands in the TOE summary specification.**

Management of TSF data (FMT_MTD.1(DC))

FMT_MTD.1.1(DC) The TSF shall restrict the ability to **perform management functions for** the **digital certificates** to **users with the SPECIAL attribute and users assigned the authority to specific management functions as defined in the tables in the section on managing digital certificates in the TOE summary specification**.

Application note: To perform a specific management function for digital certificates, a user that does not have the SPECIAL attribute must have RACF authority to a profile of the type IRR.DIGTCERT.function in the FACILITY class where function is the name of the management function. The list of management functions and the semantics of READ, UPDATE and CONTROL authority for each function is defined in the tables in Authority checking for RACDCERT Processing, Authority Checking for R_datalib Processing and the section "Authority Checking for PKCS#11 Cryptographic Tokens in the ICSF TKDS" in the z/OS Security Target. That chapter in the z/OS Security Target also discusses use of

resources in the CRYPTOZ resource class to control access to PKCS#11 tokens. To determine the authority a user has to those profiles, RACF uses the algorithm defined in FDP_ACF.1(GRD).

Revocation: object security attributes (FMT_REV.1(OSA))

FMT_REV.1.1(OSA) The TSF shall restrict the ability to revoke **object security attributes defined by SFPs** associated with the **corresponding object** under the control of the TSF to **users that satisfy the following rules: users authorized to modify the security attributes by the RACF general profile access control policy, the UNIX file system object access control policy, the UNIX IPC objects access control policy or (in Labeled Security Mode) the mandatory access control policy.**

FMT_REV.1.2(OSA) The TSF shall enforce the ***following*** rules:

a) The access rights associated with an object shall be enforced when an access check is made;

b) Labeled Security Mode only: the rules of the mandatory access control policy are enforced on all future operations.

Revocation: user security attributes (FMT_REV.1(USR))

FMT_REV.1.1(USR) The TSF shall restrict the ability to revoke **user security attributes defined by the SFP** associated with the **corresponding user** under the control of the TSF to **the authorized identified roles allowed to modify user security attributes.**

FMT_REV.1.2(USR) The TSF shall enforce the ***following*** rules:

a) The enforcement of the revocation of user security attributes stored in the user profile with the next user-subject binding process during the next authentication of the user;

b) the immediate revocation of security-relevant access authorization (active with the next access check being made).

Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions:

a) Management of auditing parameters and configuration;

b) Management of RACF General Resource and Data set Profile Access Control Policy;

c) Management of UNIX File System Object Access Control Policy;

d) Management of UNIX IPC Object Access Control policy;

e) Management of Field Level Access Control;

f) Management of identification and authentication policy;

g) Management of user security attributes;

h) Management of RACF profiles

i) Management of UNIX file system object security attributes

j) Management of UNIX IPC objects security attributes

k) Management of RACF configuration data

Security Roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles:

a) User role with the following rights:

- **Users are authorized to modify their own user password;**
- **Users are authorized to modify the access control permissions for the named objects they own;**

b) users authorized by the RACF general resource class and data set access control policy to modify object security attributes;

c) in Labeled Security Mode: users authorized by the mandatory access control policy to modify the object label as an object security attribute;

d) users authorized to perform administrative actions within a defined group (group-SPECIAL attribute)

e) users authorized to perform administrative actions for user or group security attributes via ownership

f) RACF auditors (users who have the RACF AUDITOR attribute in their profiles)

g) RACF group auditors (users who have the RACF group-AUDITOR attribute in their profiles)

h) Operations roles (users with the OPERATIONS attribute)

i) group-Operations roles (users with the RACF OPERATIONS attribute within a group or set of groups)

j) z/OS pseudo-user (protected user IDs)

k) z/OS UNIX superusers

l) Users authorized to perform management operations for digital certificates based on access rights to RACF profiles protecting the individual management operations

m) Users authorized to perform other management functions based on access rights to RACF profiles protecting the individual management operations

n) authorized administrator (user with the SPECIAL attribute).

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

7.6 Protection of the TSF (FPT)

Inter-TSF basic TSF data consistency (FPT_TDC.1(RA))

FPT_TDC.1.1(RA) The TSF shall provide the capability to consistently interpret **information in the RACF database and security attributes of UNIX file system objects** when shared between the TSF and another trusted IT product.

FPT_TDC.1.2(RA) The TSF shall use **the rules to interpret RACF profiles and authorizations and the rules to interpret extended attributes of UNIX file system objects** when interpreting the TSF data from another trusted IT product.

Application note: Inter-TSF data consistency shall ensure that access control information is consistently interpreted when this information is shared between different instantiations of the TOE or when UNIX file system objects with their extended attributes are exported from one system and imported into another system. The discretionary access control information either has to be identical (which requires that the same users, groups and user membership of groups are defined in the involved systems) or this information has to be updated accordingly by a system administrator before the UNIX file system object is made available to other user on the system importing the object.

Inter-TSF basic TSF data consistency (FPT_TDC.1(LS)) (Labeled Security Mode only)

FPT_TDC.1.1(LS) The TSF shall provide the capability to consistently interpret **label-related security attributes** when shared between the TSF and another trusted IT product.

FPT_TDC.1.2(LS) The TSF shall use **the list of security labels to be applied by the TSF** when interpreting the TSF data from another trusted IT product.

Application note: Inter-TSF data consistency shall ensure that access control information including security labels are consistently interpreted when this information is shared between different instantiations of the TOE. In order to do this, at least the definition of the security labels between the systems involved have to be identical.

7.7 Security Functional Requirements Rationale

7.7.1 Security Requirements Coverage

The following table provides a mapping of SFR to the security objectives, showing that each security functional requirement addresses at least one security objective.

Security Functional Requirements	Objectives
FAU_GEN_SUB.1	O.AUDITING
FAU_GEN.2	O.AUDITING
FAU_SAR.1	O.AUDITING
FAU_SEL.1	O.AUDITING

Security Functional Requirements	Objectives
FCS_COP.1	O.PROGRAM_INTEGRITY_SUPPORT
FDP_ACC.1(GRD)	O.DISCRETIONARY.ACCESS
FDP_ACC.1(UFS)	O.DISCRETIONARY.ACCESS
FDP_ACC.1(IPC)	O.DISCRETIONARY.ACCESS
FDP_ACC.1(FLA)	O.DISCRETIONARY.ACCESS
FDP_ACF.1(GRD)	O.DISCRETIONARY.ACCESS
FDP_ACF.1(UFS)	O.DISCRETIONARY.ACCESS
FDP_ACF.1 (IPC)	O.DISCRETIONARY.ACCESS
FDP_ACF.1 (FLA)	O.DISCRETIONARY.ACCESS
FDP_IFC.2 (Labeled Security Mode only)	O.LS.CONFIDENTIALITY
FDP_IFF.2 (Labeled Security Mode only)	O.LS.CONFIDENTIALITY
FIA_AFL.1	O.I&A
FIA_ATD.1(HU)	O.I&A
FIA_ATD.1(LS) (Labeled Security Mode only)	O.LS.LABEL
FIA_SOS.1	O.I&A
FIA_UAU.1	O.I&A
FIA_UAU.5	O.I&A, O.I&A.MULTIPLE
FIA_UAU.7	O.I&A
FIA_UID.1	O.I&A
FIA_USB.1 (Labeled Security Mode only)	O.LS.LABEL
FIA_USB.2	O.I&A

Security Functional Requirements	Objectives
FMT_MSA.1(GRD)	O.MANAGE
FMT_MSA.1(UFS)	O.MANAGE
FMT_MSA.1(IPC)	O.MANAGE
FMT_MSA.1(FLA)	O.MANAGE
FMT_MSA.1(LS) (Labeled Security Mode only)	O.MANAGE
FMT_MSA.3(GRD)	O.MANAGE
FMT_MSA.3(UFS)	O.MANAGE
FMT_MSA.3(IPC)	O.MANAGE
FMT_MSA.3(FLA)	O.MANAGE
FMT_MSA.3(LS) (Labeled Security Mode only)	O.MANAGE
FMT_MTD.1(AE)	O.MANAGE
FMT_MTD.1(SO)	O.MANAGE
FMT_MTD.1(UA)	O.MANAGE
FMT_MTD.1(RA)	O.MANAGE
FMT_MTD.1(TH)	O.MANAGE
FMT_MTD.1(AD)	O.MANAGE
FMT_MTD.1(RC)	O.MANAGE
FMT_MTD.1(DC)	O.MANAGE
FMT_REV.1(OSA)	O.MANAGE
FMT_REV.1(USR)	O.MANAGE
FMT_SMF.1	O.MANAGE
FMT_SMR.1	O.MANAGE

Security Functional Requirements	Objectives
FPT_TDC.1(RA)	O.DISCRETIONARY.ACCESS
FPT_TDC.1(LS) (Labeled Security Mode only)	O.LS.CONFIDENTIALITY, O.LS.LABEL

Table 16: Mapping of security functional requirements to security objectives

7.7.2 Security Requirements Sufficiency

The following rationale provides justification for each security objective for the TOE, showing that the security functional requirements are suitable to meet and achieve the security objectives:

Security objectives	Rationale
O.AUDITING	The events to be audited are defined in FAU_GEN_SUB.1 and are associated with the identity of the user that caused the event (FAU_GEN.2). Authorized users are provided the capability to read the audit records (FAU_SAR.1). The authorized user must have the capability to specify which audit records are generated (FAU_SEL.1). The audit trail is stored and managed by the SMF component of the z/OS operating system, which ensures the protection of the audit trail and protects audit data from being lost.
O.DISCRETIONARY.ACCESS	The TSF must control access to resources based on the identity of users that are allowed to specify which resources they want to access for storing their data. There are 4 different discretionary access control policies supported by the TOE: 1. The "general resource access control policy" (GRD), which the TOE enforces for "general resource profiles. Some of those profiles are used by RACF itself to control access to its own resources. External resource managers that use general resource profiles may call RACF to decide if a user is allowed to access a resource. This policy is defined by the SFRs FDP_ACC.1(GRD) and FDP_ACF.1(GRD). 2. The "UNIX file system access control policy" (UFS) used by RACF to control access to UNIX file system objects. For this policy RACF uses information it stores with z/OS UNIX file system objects. This policy is defined by the SFRs FDP_ACC.1(UFS) and FDP_ACF.1(UFS). 3. The "inter-process communication access control policy" used by RACF to control access to UNIX IPC objects. For this policy RACF uses information it stores with z/OS UNIX IPC objects. This policy is defined by

Security objectives	Rationale
	the SFRs FDP_ACC.1(IPC) and FDP_ACF.1(IPC). 4. The "field level access control policy" (FLA) used by RACF to control access to fields in RACF profiles via RACF commands. This policy is defined by the SFRs FDP_ACC.1(FLA) and FDP_ACF.1(FLA). In addition the objective is also supported by FPT_TDC.1(RA) which ensures the consistent interpretation of the TSF data used for controlling access.
O.I&A	The TSF must ensure that only authorized users gain access to the TOE functions and resources. Users authorized to access the TOE must use an identification and authentication process (FIA_UID.1, FIA_UAU.1). This process is initiated by a trusted function within the TOE environment, which calls the TOE to perform the actual user authentication. Multiple I&A mechanisms are allowed as specified in FIA_UAU.5. To ensure authorized access to the TOE, authentication data is protected (FIA_ATD.1(HU), FIA_UAU.7). Proper authorization for subjects acting on behalf of users is also ensured (FIA_USB.2). The appropriate strength of the authentication mechanism is ensured (FIA_SOS.1). To support the strength of authentication methods, the TOE is capable of identifying and reacting to unsuccessful authentication attempts (FIA_AFL.1).
O.I&A.MULTIPLE	The TOE supports multiple I&A mechanisms, which is specified with FIA_UAU.5.
O.MANAGE	The TOE provides management interfaces globally defined in FMT_SMF.1 for: • the access control policies (FMT_MSA.1(GRD), FMT_MSA.1(UFS), FMT_MSA.1(IPC), FMT_MSA.1(FLA), FMT_MSA.3(GRD), FMT_MSA.3(UFS)), FMT_MSA.3(IPC), FMT_MSA.3(FLA); • Note: since the default values for FMT_MSA.3(IPC) can not be changed, there is of course no management interface for this SFR. • the information flow control policy (FMT_MSA.1(LS) and FMT_MSA.3(LS)); • the user security attributes (other than authentication data) (FMT_MTD.1(UA)); • the auditing aspects (FMT_MTD.1(AE)); • the identification and authentication aspects (FMT_MTD.1(RA), FMT_MTD.1(AD), FMT_MTD.1(TH)); • the use of RACF commands (FMT_MTD.1(RC)); • the general management of RACF (FMT_MTD.1(SO));

Security objectives	Rationale
	• the management of digital certificates (used for user authentication and for program signing and verification) (FMT_MTD.1(DC)); The rights management for the different management aspects is defined with FMT_SMR.1. The management interfaces for the revocation of user and object attributes is provided with (FMT_REV.1(OSA), FMT_REV.1(USR)).
O.PROGRAM_INTEGRITY_SUPPORT	The support for program integrity verification is defined by FCS_COP.1, which describes the functional requirement for program signature generation and verification support.
O.LS.CONFIDENTIALITY	The information flow control policy is defined by specifying the subjects, objects, security attributes and rules in FDP_IFC.2 and FDP_IFF.2. This objective is also supported by FPT_TDC.1(LS), which ensures the consistent interpretation of the TSF data used for the label based access control policy.
O.LS.LABEL	The assignment of labels to users is performed during user-subject binding (FIA_USB.1(LS)) with security attributes maintained by the TOE (FIA_ATD.1(LS)). FPT_TDC.1(LS) ensures the consistent interpretation of the labels.

Table 17: Security objectives for the TOE rationale

7.7.3 Security Requirements Dependency Analysis

The following table demonstrates the dependencies of SFRs modeled in CC Part 2 and how the SFRs for the TOE resolve those dependencies:

Security Functional Requirement	Dependencies	Resolution
FAU_GEN_SUB.1	no dependencies	
FAU_GEN.2	FAU_GEN.1	FAU_GEN_SUB.1
	FIA_UID.1	FIA_UID.1
FAU_SAR.1	FAU_GEN.1	FAU_GEN_SUB.1
FAU_SEL.1	FAU_GEN.1	FAU_GEN_SUB.1
	FMT_MTD.1	FMT_MTD.1(AE)
FCS_COP.1	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FMT_MTD.1(DC) (see discussion below)

Security Functional Requirement	Dependencies	Resolution
	FCS_CKM.4	no (see discussion below)
FDP_ACC.1(GRD)	FDP_ACF.1	FDP_ACF.1(GRD)
FDP_ACC.1(UFS)	FDP_ACF.1	FDP_ACF.1(UFS)
FDP_ACC.1(IPC)	FDP_ACF.1	FDP_ACF.1(IPC)
FDP_ACC.1(FLA)	FDP_ACF.1	FDP_ACF.1(FLA)
FDP_ACF.1(GRD)	FDP_ACC.1	FDP_ACC.1(GRD)
	FMT_MSA.3	FMT_MSA.3(GRD)
FDP_ACF.1(UFS)	FDP_ACC.1	FDP_ACC.1(UFS)
	FMT_MSA.3	FMT_MSA.3(UFS)
FDP_ACF.1(IPC)	FDP_ACC.1	FDP_ACC.1(IPC)
	FMT_MSA.3	FMT_MSA.3(IPC)
FDP_ACF.1(FLA)	FDP_ACC.1	FDP_ACC.1(FLA)
	FMT_MSA.3	FMT_MSA.3(FLA)
FDP_IFC.2 (Labeled Security Mode only)	FDP_IFF.1	FDP_IFF.2(LS) (Labeled Security Mode only)
FDP_IFF.2 (Labeled Security Mode only)	FDP_IFC.1	FDP_IFC.2(LS) (Labeled Security Mode only)
	FMT_MSA.3	FMT_MSA.3(LS) (Labeled Security Mode only)
FIA_AFL.1	FIA_UAU.1	FIA_UAU.1
FIA_ATD.1(HU)	No dependencies.	
FIA_ATD.1(LS) (Labeled Security Mode only)	No dependencies.	
FIA_SOS.1	No dependencies.	
FIA_UAU.1	FIA_UID.1	FIA_UID.1
FIA_UAU.5	No dependencies.	

Security Functional Requirement	Dependencies	Resolution
FIA_UAU.7	FIA_UAU.1	FIA_UAU.1
FIA_UID.1	No dependencies.	
FIA_USB.1(LS) (Labeled Security Mode only)	FIA_ATD.1	FIA_ATD.1(LS) (Labeled Security Mode only)
FIA_USB.2	FIA_ATD.1	FIA_ATD.1(HU)
FMT_MSA.1(GRD)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(GRD)
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.1(UFS)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(UFS)
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.1(IPC)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(IPC)
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.1(FLA)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(FLA)
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.1(LS)	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.2
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.3(GRD)	FDP_MSA.1	FMT_MSA.1(GRD)

Security Functional Requirement	Dependencies	Resolution
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.3(UFS)	FMT_MSA.1	FMT_MSA.1(UFS)
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.3(IPC)	FMT_MSA.1	FMT_MSA.1(IPC)
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.3(FLA)	FMT_MSA.1	FMT_MSA.1(FLA)
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.3(LS)	FMT_MSA.1	FMT_MSA.1(LS)
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(SO)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(AE)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(UA)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(RA)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(TH)	FMT_SMR.1	FMT_SMR.1

Security Functional Requirement	Dependencies	Resolution
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(AD)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(RC)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(DC)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_REV.1(OSA)	FMT_SMR.1	FMT_SMR.1
FMT_REV.1(USR)	FMT_SMR.1	FMT_SMR.1
FMT_SMF.1	No dependencies.	
FMT_SMR.1	FIA_UID.1	FIA_UID.1
FPT_TDC.1(RA)	No dependencies.	
FPT_TDC.1(LS) (Labeled Security Mode only)	No dependencies.	

Table 18: TOE SFR dependency analysis

7.7.4 Discussion of dependencies not satisfied

The dependencies mentioned in CC part 2 for FCS_COP.1 are not satisfied. CC part 2 lists as dependencies either the import of user data or key generation. In addition in either case a dependency to key destruction is listed.

The RACF security target does not satisfy those dependencies for the following reason:

- RACF does not generate cryptographic keys, but imports them. For this import FDP_ITC.1 or FDP.ITC.2 are not suitable requirements, since they relate to "user data" while the certificates imported are actually TSF data. Still their import is protected, since importing certificates to be used program signing and verification requires the user to have the right to use the RACDCERT command with the required parameter and also have the appropriate access to the RACF key ring used to store the imported certificate. This is reflected in the SFR FMT_MTD.1(DC), which requires appropriate administrative privileges to manage the digital certificates. Therefore FMT_MTD.1(DC) resolves the dependency to control the rights to import digital certificates.
- RACF does not destruct cryptographic keys but leaves this to the environment (ICSF), which RACF uses to store the RACF key rings. Therefore the dependency to FCS_CKM.4 is

addressed in the TOE environment.

7.7.5 TSF Rationale

The following table maps the SFRs to the TOE summary description and provides pointers into the TSS where the implementation of the SFR is described.

Security Functional Requirements	Security Functions
FAU_GEN_SUB.1	Section 8.5.1 explains how audit records are generated. This section also explains the structure of the audit records.
FAU_GEN.2	Section 8.5.1 explains the information contained in the audit records. Tools to export audit records in human-readable format are mentioned in this section, too.
FAU_SAR.1	Section 8.4.1.2.1, subsection "AUDITOR and group-AUDITOR" explains the auditor role. Section 8.4.3.1 explains the options of the SETROPT command a user in the AUDITOR role may use. Section 8.5.4 describes the purpose of the audit dump programs that read audit records from the audit trail and store them in a data set where they can be assessed.
FAU_SEL.1	Sections 8.4.3.2 table 38 explain how the auditor role can configure the events that are audited using the RACF commands and the operands reserved for users in the AUDITOR role. This section also explain that the owner of a profile can define which events related to the profile are audited.
FCS_COP.1(SGN)	Support for program signing and signature verification is explained in section 8.5.5
FDP_ACC.1(GRD)	The general resource access control policy is described in sections 8.3.4.1 and 8.3.4.2.
FDP_ACC.1(UFS)	The UNIX file system access control policy is explained in section 8.3.4.4.
FDP_ACC.1(IPC)	The IPC object access control policy is described in section 8.3.4.5.
FDP_ACC.1(FLA)	The RACF field level access control policy is explained in sections 8.4.2 and 8.4.3.2 in table 38.
FDP_ACF.1(GRD)	The general resource access control policy is described in sections 8.3.4.1 and 8.3.4.2.
FDP_ACF.1(UFS)	The UNIX file system access control policy is explained in section 8.3.4.4.

Security Functional Requirements	Security Functions
FDP_ACF.1 (IPC)	The IPC object access control policy is described in section 8.3.4.5.
FDP_ACF.1 (FLA)	The RACF field level access control policy is explained in sections 8.4.2 and 8.4.3.2 in table 38.
FDP_IFC.2(LS) (Labeled Security Mode only)	The labeled security access control policy is explained in section 8.3.3.
FDP_IFF.2(LS) (Labeled Security Mode only)	The labeled security access control policy is explained in section 8.3.3.
FIA_AFL.1	The system-wide attribute REVOKE for the number of failed consecutive authentication attempts is explained in sections 8.4.1, table 28, and the section titled "User Revocation" and the section titled "User profiles" where the REVOKE attribute in a user profile is explained. The effect of a user ID being revoked is described in section 8.2.1.
FIA_ATD.1(HU)	User attributes for human users are defined in the user profile, which is described in section 8.4.1 in the subsection titled "User profiles".
FIA_ATD.1(LS) (Labeled Security Mode only)	Labels as users attributes are also described in section 8.4.1 in the subsection titled "User profiles".
FIA_SOS.1	The password and password phrase specifics are defined in section 8.2.2 where the options for the password and passphrase policy are defined in the subsections titled "Password Quality" and "Password Phrase Quality".
FIA_UAU.1	User authentication is explained in section 8.2.1.
FIA_UAU.5	Authentication using passwords and password phrases is explained in section 8.2.2. Authentication using RACF Pass Tickets is explained in section 8.2.3. Authentication using digital certificates is explained in section 8.2.4. Authentication using Kerberos tickets is explained in section 8.2.5.
FIA_UAU.7	Section 8.2.1 describes the RACF interfaces that can be invoked for user authentication. The functions do provide any feedback to the caller while they are in progress.
FIA_UID.1	User identification is explained in 8.2.1.
FIA_USB.1(LS) (Labeled Security Mode only)	The user sensitivity level bound to subjects while the TOE operates in Labeled Security Mode is explained in 8.2.1 (under "some additional considerations").

Security Functional Requirements	Security Functions
FIA_USB.2	User subject binding for z/OS is explained in section 8.2.7 with respect to group processing and 8.2.8, which describes how RACF creates an ACEE using the information from the user's profile.
FMT_MSA.1(GRD)	Management of access control for the general resource access control policy is explained in table 38 where the RACF commands and the restrictions on their usage is described. Access control to general resource profiles is managed by the PERMIT command.
FMT_MSA.1(UFS)	Management of access control for the UNIX file system access control policy is described in section 8.4.3 in the section titled "Management of z/OS UNIX file system objects and IPC objects" where the RACF interfaces for the management of the UNIX file system access policy and IPC access policy are described.
FMT_MSA.1(IPC)	Management of access control for the IPC access control policy is described in section 8.4.3 in the section titled "Management of z/OS UNIX file system objects and IPC objects" where the RACF interfaces for the management of the UNIX file system access policy and IPC access policy are described.
FMT_MSA.1(FLA)	Management of the RACF profile field level access control policy is performed via the management of profiles in the FIELD class and defining access to those profiles using the PERMIT command. The management of access control lists for general resource class profiles therefore applies here.
FMT_MSA.1(LS) (Labeled Security Mode only)	Management of security labels being restricted to users with the SPECIAL attribute is described in section 8.3.3 as well as in table 38 which states that for the various commands that may set or alter a security label the user has to have the SPECIAL attribute to use this functionality.
FMT_MSA.3(GRD)	Default values for the general resource access control policy are described in section 8.3.4.1. Default access of a user is defined by the UACC value in the profile protecting the resource, which has a value of "none" as the default.
FMT_MSA.3(UFS)	Default values for the UNIX file system access control policy are described in section 8.3.4.3 which explains that access is denied unless it is given either by the ACLs or the permission bits.
FMT_MSA.3(IPC)	The default values for this access control policy can not be managed.
FMT_MSA.3(FLA)	The default values for this access control policy are defined by the UACC value for the profiles in the FIELD class which are managed by the functions of the RACF general resource class access control policy.

Security Functional Requirements	Security Functions
FMT_MSA.3(LS) (Labeled Security Mode only)	Default values for the security label are defined in the SECLABEL attribute in the resource profiles as explained in section 8.3.3 (and subsections) in the description of the resource profiles and in section 8.3.2.4 for z/OS UNIX objects
FMT_MTD.1(AE)	Audit trail management is performed using the command options reserved for users in the AUDITOR or group-AUDITOR role as defined in table 38. Those are: • SETROPTS options reserved for users with the AUDITOR privilege • GLOBALAUDIT keyword for ALTDSD and RALTER commands • UAUDIT/NOUAUDIT keyword for the ALTUSER command
FMT_MTD.1(SO)	The SETROPTS command related management is described in section 8.4.3.1 and in table 38.
FMT_MTD.1(UA)	User security attribute management is explained in section 8.4.1.
FMT_MTD.1(RA)	Management for re-enabling authentication is described in section 8.4.1 where the authority to reset a user's password is described in detail.
FMT_MTD.1(TH)	Management of the threshold for unsuccessful authentication events is described in section 8.4.3.1 it is explained that this threshold can be set using the SETROPTS command.
FMT_MTD.1(AD)	Management of authentication data is explained in sections 8.2.1 and 8.4.1.
FMT_MTD.1(RC)	Management of RACF commands is explained in table 38.
FMT_MTD.1(DC)	Management of digital certificates is explained in sections 8.2.4 and 8.4.1 where the RACDCERT command and the authorities required for the use of its parameter are described in detail.
FMT_REV.1(OSA)	Revocation of object security attributes is explained in section 8.4.2 for the management of general resource profiles (and data set profiles) as well as for the field-level access control are defined. Section 8.4.3, subsection titled "Management of z/OS UNIX file system objects and IPC objects" describes the interfaces used for the revocation of object security attributes for z/OS UNIX file system objects and IPC objects.
FMT_REV.1(USR)	Revocation of user security attributes is explained in section 8.4.1.

Security Functional Requirements	Security Functions
FMT_SMF.1	See SFRs FMT_MTD.1(x)
FMT_SMR.1	The roles are explained in section 8.4.1 in the subsection titled "RACF Roles".
FPT_TDC.1(RA)	The inter-TSF data consistency is given by the general structure of the profiles in the RACF database, which ensures that a RACF database is consistently interpreted when used by a remote system.
FPT_TDC.1(LS) (Labeled Security Mode only)	see above. This applies also for security labels, which are part of the RACF profiles or part of the file security attributes of z/OS UNIX file system objects.

7.7.6 Mutual support of the security functions

This section demonstrates that the TOE security functions are mutually supportive by showing how the individual functions are interrelated.

Identification and authentication is a prerequisite for discretionary and (in Labeled Security Mode) mandatory access control as well as the security management functions that require the user to have the required privileges to perform the management activities. It also is a prerequisite to auditing by provision of a unique and reliable reference to a user causing an audit event. Identification and authentication is supported by access control that protects the user and group profiles (including the authentication information) against unauthorized access and modification. In addition identification and authentication is supported by security management that defines user with their credentials and assigns initial authentication information to them.

Discretionary access control supports audit by protecting the audit data sets against unauthorized access, supports security management by protecting security management information stored in data sets or files and by ensuring that the user performing management functions have the required privileges.

Labeled Security Mode: Mandatory access control is implemented in the TOE in addition to discretionary access control. Mandatory access control is supported by identification and authentication as well as security management with respect to the definition of security labels, the assignment of labels to objects and the assignment of security classification to users. The TOE implements a label-based mandatory access control mainly for use by external resource managers, but also applies for profiles it uses to protect its own resources

Security management is required to manage the users, groups, access rights, authentication data, digital certificates, the privileges of users, and other TSF data. This is supporting identification and authentication, audit, as well as the different access control policies. 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 Labeled Security Mode) security label.

Security management also includes the management of access rights including (in Labeled Security Mode) the definition of the security labels. 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.

Auditing is a secondary security function that does not provide direct support for other security functions. Auditing provides indirect support to other security functions, because it allows identification of security problems and allows definition of appropriate measures (in the TOE configuration or the TOE environment) to prevent those events in the future.

TOE self-protection supports all other security functions to ensure that they can not be tampered with or bypassed.

7.7.7 Security assurance requirements rationale

The evaluation assurance level has been chosen commensurate with the threat environment that is experienced by typical consumers of the TOE. In addition, the evaluation assurance level has been augmented with ALC_FLR.3 commensurate with the augmented flaw remediation capabilities offered by the developer beyond those required by the evaluation assurance level.

The assurance measures and how they are satisfied are explained in the table in section "TOE Assurance Measures". The authors of this Security Target view this table as sufficient justification for the individual assurance measures.

8. TOE summary specification

This chapter provides a summary of the security functions of RACF that are subject to the evaluation.

The chapter also provides some overview material required for a basic understanding how the security functions work. Those details of the security functions that are the focus of the evaluation are marked in brackets using an identifier for the security function and a number.

8.1 Overview of the RACF architecture

RACF is implemented as a dedicated component that can be used by operating system components and trusted applications to

- authenticate users
- control access to resources
- manage access rights
- manage users and groups with their security attributes
- log and report attempts to authenticate or access protected resources

RACF manages the information it relies on in its own database. This database includes user and group profiles, which store information about individual users and groups with the security attributes assigned to the users and groups. It also includes resource profiles, where those profiles represent the resources for which a resource manager can call RACF in order to check for a user's authority to access a specific resource. In addition RACF manages access rights associated with resource profiles, which define the type of access users or groups have to the resource.

RACF has the capability to generate audit records for specific events. RACF also provides an interface to resource managers that they can use to cause RACF to generate a specific audit record. Audit records generated by RACF are not stored in the RACF database, but passed to the components of z/OS, which are centrally used to store and manage all types of audit records.

RACF has three main sets of interfaces:

- The RACROUTE macro interface, which sufficiently authorized programs executing within z/OS can use to request services from RACF
- The RACF callable services, which also sufficiently authorized programs executing within z/OS can use to request services from RACF. Those services include specific services related UNIX file systems and UNIX IPC objects.
- The RACF command interface which sufficiently authorized users can use to perform RACF management functions.

The authorizations required are defined with the individual functions and may even be dependent on the parameters used.

8.2 Identification and authentication support by RACF

8.2.1 Authentication function

An application can use RACF to support the identification and authentication of users. RACF provides the following interfaces to perform user authentication:

- the RACROUTE REQUEST=VERIFY macro
- the RACROUTE REQUEST=VERIFYX macro
- the initACEE RACF callable service

When authenticating a user the TOE allows applications to accept and provide to RACF:

- A user ID defined to RACF {IA1::IA.1.4-R8-RACF-1} and the RACF password {IA.1::IA.1.4-R8-RACF-2} or password phrase {IA.1::IA.1.4-R10-RACF-4} or a PassTicket {IA.1::IA.1.4-R8-RACF-3}.
- A valid x.509v3 digital certificate that the application has validated using TLSv1.1-, TLSv1-, or SSLv3-based client authentication and presented to RACF via initACEE (or indirectly via __certificate()) or mapped to a RACF user ID via R_usermap(), for applications supporting TLSv1.1-, TLSv1- or SSLv3-based client authentication (see Authentication via Client Digital Certificates) . {IA.1::IA.1.4-R13-RACFEAL5-1} (Note: the mapping of a certificate to an ID is the responsibility of the application, and as it is not directly a function of RACF will not be tested during this evaluation.)
- A valid Kerberos service ticket for the client Kerberos principal, which the R_usermap() service will convert to a RACF user ID, for applications supporting Kerberos (see Authentication via Kerberos) {IA.1::IA.1.4-R13-RACFEAL5-2}.

Some additional considerations:

- If security label (SECLABEL) processing is active, the user may also specify the security label he wants to have for the session or job unless the security label is already restricted by the port of entry. This user-supplied label must be within the set of labels the user is allowed to use. With this processing active, if the user does not supply a security label, a defined default security label is chosen depending on the user's label and the label of the port of entry {IA.1::IA.1.5}
- For access to UNIX functions, the user must have a valid UID and his default group must have a valid GID {IA.1::IA.1.6}. For users without a UID or GID, the FACILITY class profile BPX.DEFAULT.USER may be used to derive a default UID and GID which will be used for UNIX access checking {IA.1::IA.1.6-R8-USS-1}. For accountability, any audit records created by UNIX functions for such a default user will indicate that the default ID was assigned, and will show both the UID and the RACF user ID {IA.1::IA.1.6-R8-USS-2}.

  If the user is in additional groups they may have GIDs, too, and if so UNIX access checking will make use of those additional GIDs {IA.1::IA.1.6-R8-USS-3}.

- If the user ID is in REVOKE status, RACF prevents user from entering the system at all or entering the system with certain groups {IA.1::IA.1.7}.
- For a user defined as a system administrator (that is, one who has the system SPECIAL attribute) a message is displayed on the console asking the operator if the user shall be revoked if he exceeds the number of failed login attempts due to incorrect passwords {IA.1::IA.1.7-R8-RACF-1} or if he exceeds the system inactivity interval {IA.1::IA.1.7.R8-RACF-

2}.

- For users in the TSO environment the administrator can impose restrictions on whether the user can use the system on this day and at this time of the day. This is checked only when using a terminal from a defined set. {IA.1::IA.1.8}.
- RACF also checks if the user is authorized to access the terminal (which can also include day and time restrictions for accessing that terminal) or other port of entry {IA.1::IA.1.9}.
- RACF also checks if the user is authorized to use the application (if specified) {IA.1::IA.1.10}.
- A user may have SURROGAT authority for another user. This allows him to submit a job under the user ID of this other user without specifying the password or to use the z/OS UNIX su command to switch to this user's ID without specifying the password {IA.1::IA.1.11}. It also allows appropriately-authorized servers to switch a session to run under a pre-specified ID {IA.1::IA.1.11-R8-MULTI-1}. In Labeled Security Mode, the surrogate user who submits the job must have read access to the security label under which the job runs {IA.1::IA.1.12}. The audit record for surrogate job submission identifies both the surrogate user and the jobcard user ID {IA.1::IA.1.13}.

The application requesting RACF to authenticate a user will usually also request RACF to create a specific control block, named Accessor Control Environment Element (ACEE). The content of the ACEE is built from information taken from the user's profile in the RACF database and from information provided by the application requesting authentication. Examples of information supplied by the requesting application are the "Port of entry" or the name of the application. The ACEE is later used in access checks performed by RACF when the user related information like the user's RACF attributes, group memberships, port of entry, etc. are evaluated as part of the access control algorithm.

8.2.2 RACF Passwords and Password Phrases

In RACF, the user selects his own password/phrase and only the user knows the value chosen. If the user has forgotten his password/phrase and it needs to be reset, the security administrator will reset the password/phrase {IA.2::IA.2.1-R10}. When the system administrator follows the rules for the evaluated configuration, this new password/phrase should be in an expired state, thus forcing the user to enter a new password/phrase on the next logon {IA.2::IA.2.2-R10}. When creating a new user ID for a pseudo-user that is not a protected user ID, the initial password/phrase may be marked as nonexpired, allowing it to be used without being changed first. {IA.2::IA.2.3-R10}.

8.2.2.1 Password Quality

A system administrator can set a variety of system-global rules for forming valid passwords using the SETROPTS command (for system-wide settings) or (to a lesser extent) using the password command to affect only one user. He can change such parameters as the number of days a password is valid for, how long to maintain password history to prevent the user from reusing the same password again, the minimum number of days between password changes, and syntax rules for password content.

When a user changes a password, RACF treats the new, user-supplied password as an encryption key to transform the RACF user ID into an encoded form using the DES algorithm that it stores on the database. The password is not stored in clear text {IA.2::IA.2.4}.

The following system-wide options can be set to enforce a minimum strength of passwords using the PASSWORD option in the SETROPTS command:

- Minimum and maximum length of passwords (LENGTH(m1:m2) as part of a RULE suboption) {IA.2::IA.2.5}
- Maximum password lifetime (INTERVAL suboption) {IA.2::IA.2.6} and minimum password change time (MINCHANGE option) {IA.2::IA.2.V1R7-1}

- Number of passwords from the user's password history that are not allowed for a new password (HISTORY suboption) {IA.2::IA.2.7}
- Maximum number of consecutive failed authentication attempts until the REVOKE attribute is set in the user's profile (REVOKE suboption) {IA.2::IA.2.8}
- Differentiate between upper- and lowercase characters with the PASSWORD(MIXEDCASE) option {IA.2::IA.2.V1R7-2}
- Type of character for each character position of a password. Possible types are {IA.2::IA.2.9}:
  - ALPHA
  - ALPHANUM (which includes also the special characters $, # and @)
  - VOWEL
  - NOVOWEL
  - CONSONANT
  - NUMERIC
  - MIXEDCONSONANT
  - MIXEDVOWEL
  - MIXEDNUM
  - NATIONAL

If the value ALPHANUM is defined for more than one position in the password, at least one alphabetical value and one numeric value are required by RACF.

Note that the TSF can not ensure that passwords entered into programs executing with the user's privilege are fully protected from being spoofed. The user has to take care about his password in those cases as explained in the guidance.

Note that,as previously mentioned, for a local Kerberos user, when using RACF as the KDC's registry, the user's RACF password/phrase and Kerberos password are the same..

8.2.2.2 Password Phrase Quality

Many of the system rules for passwords set by SETROPTS apply to password phrases, too. However, RACF does not provide support for content syntax rules when using password phrases.

When a password phrase is established for a user, RACF treats the new phrase as a sequence of encryption keys to transform the RACF user ID into an encoded form using the DES algorithm with chaining, that it then stores on the database. The password phrase is not stored in clear text {IA.2::IA.2-R10-RACF-1}.

The following system-wide options that can be set to enforce a minimum strength of passwords using the PASSWORD option in the SETROPTS command also apply to password phrases:

- Maximum password phrase lifetime (INTERVAL suboption) {IA.2::IA.2-R10-RACF-2} and minimum password phrase change time (MINCHANGE option) {IA.2::IA.2-R10-RACF-3}
- Number of password phrases from the user's password phrase history that are not allowed for a new password phrase (HISTORY suboption) {IA.2::IA.2-R10-RACF-4}
- Maximum number of consecutive failed authentication attempts using a password or password phrase until the REVOKE attribute is set in the user's profile (REVOKE suboption) {IA.2::IA.2-R10-RACF-5}

Rather than having an administrator specify syntax rules to specify valid password phrase content,

RACF enforces the following set of predefined rules:

- maximum length: 100 characters in the absence of exit ICHPWX11 {IA.2::IA.2-R10-RACF-6}

  Note: The evaluated configuration of the TOE generally does not allow customers to implement exits to change the system processing. However, RACF supplies a sample ICHPWX11 exit and a sample REXX exec IRRPHREX that the sample ICHPWX11 will invoke. The administrator may install the sample ICHPWX11 unmodified, and may specify tailoring options in IRRPHREX to apply some additional syntax/content rules.

- minimum length:
  - 14 characters in the absence of exit ICHPWX11 {IA.2::IA.2-R10-RACF-7}
  - 9 characters if exit ICHPWX11 is present and allows the phrase {IA.2::IA.2-R10-RACF-8}

- The phrase may not contain the user ID, in either sequential uppercase or sequential lowercase characters {IA.2::IA.2-R10-RACF-9}
- The phrase must contain at least two alphabetic characters (A-Z, a-z) {IA.2::IA.2-R10-RACF-10}
- The phrase must contain at least two non-alphabetic characters (numeric, punctuation, special (including blanks)) {IA.2::IA.2-R10-RACF-11}
- The phrase may not contain more than two consecutive identical characters {IA.2::IA.2-R10-RACF-12}

If the administrator chooses to install the supplied sample exit ICHPWX11, the sample REXX exec IRRPHREX may then apply the following additional checks, if selected by the administrator, and may then accept a shorter phrase or reject a phrase that RACF would have accepted:

- The administrator can set the minimum allowable phrase length to a value between 9 and 100 inclusive by setting variable Phr_minlen {IA.2::IA.2-R10-RACF-26}
- The administrator can set the maximum allowable phrase length to a value between 9 and 100 inclusive by setting variable Phr_maxlen {IA.2::IA.2-R10-RACF-13}
- The administrator can set a more restrictive set of characters for password phrases by setting the variables numbers, letters, special, and Phr_allowed_chars {IA.2::IA.2-R10-RACF-14}
- The administrator can prevent leading or trailing blanks in password phrases by setting the variables Phr_leading_blanks or Phr_trailing_blanks to “no” IA.2::IA.2-R10-RACF-15}
- The administrator can prevent use of password phrases that contain a case-insensitive character string from the user's name by setting the variable Phr_name_allowed to “no” and setting the variable Phr_name_minlen to the longest substring allowed {IA.2::IA.2-R10-RACF-16}

  Example: if the user's name is John Smith the administrator could prevent the user from specifying a phrase containing John or john or jOhn or Smit by appropriate settings of the variables.

- The administrator can enable a triviality check by setting the variable Phr_triviality to “yes”. This will prevent use of a new password phrase that differs from the old one only insertion/deletion of spaces or changing character case. It also will reject a new phrase when the shorter of the old and new phrases is simply a substring of the other. {IA.2::IA.2-R10-RACF-17}
- The administrator can prevent use of new phrases that do not differ in a significant number of characters from the old phrase by setting the variable Phr_min_unique to the number of positions that must differ. In addition, if the variable Phr_min_unique_norm has the value “yes” the exec will first normalize the old and new phrases to be checked by converting them

to uppercase and removing spaces. {IA.2::IA.2-R10-RACF-18}

- The administrator can prevent the user of a new phrase which simply reorders the words of the old phrase by setting the variables Phr_unique_words (number of words that must be unique), Phr_word_minlen (minimum length of the unique words), and Phr_word_unique_upper (if “yes” then the exec will convert the old and new phrases to uppercase for this check {IA.2::IA.2-R10-RACF-19}
- The administrator can provide a list of disallowed words by setting the variables Phr_dict.0 to the number of words in a supplied list, and supplying the list in variables Phr_dict.1, Phr_dict.2, etc. {IA.2::IA.2-R10-RACF-20}

8.2.3 RACF PassTickets

PassTickets provide a one-time {IA.2::IA.2.14-R8-RACF-1} (by default, though administrators can change that for selected applications {IA.2::IA.2.14-R8-RACF-2}), cryptographically-computed, password substitute that may be used to authenticate a user {IA.2::IA.2.14-R8-RACF-3}. The computed value comprises information about the user ID, the application to which the user is authenticating, and the date and time-of-day {IA.2::IA.2.14-R8-RACF-4}. A given PassTicket is usable only within a time interval of plus-or-minus 10 minutes from the time of generation {IA.2::IA.2.14-R8-RACF-5}.

The cryptographic computation of a PassTicket requires usage of a secret key assigned by the administrator, and (for computations on z/OS) maintained within a profile in the PTKTDATA class. PassTicket evaluation also uses PTKTDATA profiles to determine the appropriate secret key to use.

For PassTicket generation, RACF locates a PTKTDATA profile whose name matches the application name, and extracts the secret key from it. The generation of the PassTicket then proceeds, using the user ID, application name, time/date, and selected key as inputs to the generation algorithm.

For PassTicket evaluation, RACF receives a user ID, application name, and optionally a group name, and locates a PTKTDATA profile to determine the secret key using a series of profile lookups, until a matching profile is found:

1. application-name.group-name.user-ID {IA.2::IA.2.14-R8-RACF-6}
2. application-name.user-ID {IA.2::IA.2.14-R8-RACF-7}
3. application-name.group-name {IA.2::IA.2.14-R8-RACF-8}
4. application-name {IA.2::IA.2.14-R8-RACF-9}

RACF provides two services for generation of PassTickets:

1. An internal service usable only by key 0 callers and located via the RCVT (RCVTPTGN); {IA.2::IA.2.14-R8-RACF-10}
2. An external service usable by appropriately authorized users or servers, and invoked by R_ticketserv() or R_gensec() {IA.2::IA.2.14-R8-RACF-11}. To use one of these services for PassTicket generation the caller needs UPDATE authority to resource IRRPTAUTH.application-name.target-user-ID in the PTKTDATA class. {IA.2::IA.2.14-R8-RACF-12}

RACF also allows applications to evaluate PassTickets by using the R_ticketserv() or R_gensec() services {IA.2::IA.2.14-R8-RACF-13}. Use of these services for PassTicket evaluation requires READ authority to IRRPTAUTH.application-name.target-user-id in the PTKTDATA class {IA.2::IA.2.14-R8-RACF-13a}.

8.2.4 Authentication via Client Digital Certificates

z/OS applications can accept client certificates and map them to RACF user IDs as part of the client authentication process. Such applications must be configured to use RACF to store the keyrings that contain the application private key and the allowed Certificate Authority (CA) certificates that may be used to provide the client certificates that the application will support. The security administrator will use RACDCERT to establish those keyrings, which may reside in RACF profiles in the DIGTRING class or in PKCS#11 tokens maintained in ICSF, and thus to approve of any CAs that will be used. Any CA used in the evaluated configuration must support Certificate Revocation Lists (CRLs) maintained in an LDAP registry, and the security administrator must configure the application to use the CRLs. This configuration may be application-specific, or may be done by establishing LE environment variables that System SSL will use in the absence of specific application-provided CRL configuration information.

The first step in the client authentication process is for the application to acquire the client certificate, usually via the standard SSLv3 or TLS data flows. As part of that processing, another z/OS component, System SSL will validate the client certificate using the gsk_validate_certificate_mode() function, passing the validation mode to be applied to the validation processing. This validation is outside the scope of RACF.

After System SSL has validated the client certificate, the application (or AT-TLS) can map it to a RACF user ID via the R_usermap() callable service {IA.2::IA.2.16-R8-RACF-1}. Or the application can directly create a security environment for the user by using the InitACEE() service {IA.2::IA.2.16-R8-RACF-3}. In this case, RACF will:

1. Examine the RACF database and determine whether the certificate is installed and registered to a specific user. If so, return that user ID {IA.2::IA.2.17-R8-RACF-1}
2. Otherwise, RACF will try to find the best-matching mapping profile (DIGTNMAP class), and return the user ID specified in the profile's APPLDATA field:
    a. Check for a filter of subject's-full-name.issuer's-full-name {IA.2::IA.2.17-R8-RACF-2}
    b. Iteratively remove nodes from the subject's name and check for a filter of the form: subject's-partial-name.issuer's-full-name {IA.2::IA.2.17-R8-RACF-3}
    c. Check for a filter of the form: subject's-full-name {IA.2::IA.2.17-R8-RACF-4}
    d. Iteratively remove nodes from the subject's name and check for a filter of the form: subject's-partial-name {IA.2::IA.2.17-R8-RACF-5}
    e. Check for a filter of the form: issuer's-full-name {IA.2::IA.2.17-R8-RACF-6}
    f. Iteratively remove nodes from the issuer's name and check for a filter of the form: issuer's-partial-name {IA.2::IA.2.17-R8-RACF-7}
3. Otherwise, RACF will try to find the best-matching mapping profile (DIGTNMAP, DIGTCRIT class) that matches the mapping criteria specified by the application that presented the certificate to RACF, and if found return the user ID specified in the DIGTNMAP profile's APPLDATA field {IA.2::IA.2.17-R8-RACF-8}.
4. Otherwise, if the certificate contains at least one hostIDMappings extension with a host-name and user ID {IA.2::IA.2.17-R8-RACF-9} and the certificate was issued by a CA defined to RACF as having the HIGHTRUST status {IA.2::IA.2.17-R8-RACF-10}, then RACF will examine each of the hostIDMappings extensions, in order {IA.2::IA.2.17-R8-RACF-11}. RACF will determine whether the application has READ access to IRR.HOST.host-name in the SERVAUTH class, and if so RACF will return the user ID associated with that host-name {IA.2::IA.2.17-R8-RACF-12}.

8.2.5 Authentication via Kerberos

In the evaluated configuration Kerberos-aware applications can accept Kerberos service tickets from Kerberos clients (principals), map them to RACF user IDs, and allow them to access the system using their RACF identities. In addition, users running on z/OS may have Kerberos identities, and act as clients (Kerberos principals) to Kerberos-aware servers. For more information on authentication using the z/OS Network Authentication Service and how it uses RACF see the z/OS Security Target. The main support RACF provides for this function is the provision and management of the KERB segment of a user profile, How the z/OS Network Authentication Services use RACF is beyond the scope of this Security Target.

8.2.6 Started procedures

With the concept of a started procedure, z/OS provides a mechanism where a defined task can be started by an operator, but then operates under a defined user ID that is specifically assigned to the started procedure itself.

A started procedure consists of a set of job control language statements that are frequently used together to achieve a certain result. Started procedures usually reside in the system procedure library, SYS1.PROCLIB, which is a partitioned data set. A started procedure is usually started by an operator, but can be associated with a functional subsystem. For example, SMS is treated as a started procedure even though it does not need to be specifically started with a START command.

Only RACF-defined users and groups can be specifically authorized to access RACF-protected resources {IA.3::IA.3.2}. Other users can access those resources with the authority allowed in the UACC entry of the RACF profile controlling access to the resource. However, started procedures have system-generated JOB statements that do not contain the USER, GROUP, or PASSWORD parameter.

To enable started procedures to access RACF-protected resources with other authorities than those defined in the UACC entry of the profile protecting the resource, started procedures must have RACF user IDs and group names. By assigning them RACF identities, an installation can give started procedures specific authorization to access RACF-protected resources. For example, one can allow JES to access spool data sets.

To associate the names of started procedures with specific RACF group names and user IDs, an administrator can do one of the following:

- Set up the STARTED class (the recommended method)
- Create a started procedures table (ICHRIN03)

8.2.6.1 Assigning RACF user IDs to started procedures

As with any other user ID and group name, the user ID and group name that is assigned to a started procedure must be defined to RACF using the ADDUSER and ADDGROUP commands, and the user must be connected to the group. The administrator also needs to use the PERMIT command to authorize the users or groups to get access to the required resources.

8.2.6.2 Protected user IDs

The user IDs that an administrator assigns to started procedures should have the PROTECTED attribute unless the started procedure is required to have a user ID with a password defined. Protected user IDs are user IDs that have both the NOPASSWORD and NOOIDCARD attributes {IA.3::IA.3.5}. They are defined or modified using the ADDUSER and ALTUSER commands. Protected user IDs can not be authenticated via a password, password phrase, or RACF PassTicket, and are protected from being revoked through incorrect password attempts {IA.3::IA.3.6-R12-RACFEAL5}.

8.2.7 Handling of Groups During Authentication

During authentication, RACF constructs security information that represents the user (subject) for subsequent use during access checking.

- During RACF authentication, RACF determines whether list-of-groups processing is in effect or not. If list-of-groups is not in effect, RACF puts the user's default group into the subject's ACEE, or the group specified by the caller of the RACF interfaces for user authentication. If list-of-groups is in effect, RACF gathers a list of all the groups to which the user is connected, and makes a copy of that list in the subject's ACEE. During access checking (DAC) for MVS resources, RACF can then base its decisions on both the user ID and on the group membership of the user {IA.3::IA.1.14-R12-RACFEAL5-1}.
- When an application attempts to use the RACF interfaces for UNIX functions, RACF selects from the group(s) in the subject's ACEE up to the first 300 (alphabetically) which have OMVS segments with GIDs defined. During access checking (DAC) for UNIX resources, RACF can then base its decisions on the user's UID and the selected groups' GIDs {IA.3::IA.1.14-R10-RACF-2}.

8.2.8 Assertion of User Identity

{IA.5::IA.5-R12-IDPROP-RACF-1} RACF supports specification on initACEE and RACROUTE REQUEST=VERIFY of a distributed identity via a structure called an IDID (containing a user's distinguished name (DN) and a domain/realm name (DC)):

If an IDID is specified on initACEE but a RACF user ID is not specified, then initACEE will perform a mapping operation using the IDIDMAP class to determine the associated RACF user ID to use during RACROUTE REQUEST=VERIFY processing and will also include the IDID information.

If both an IDID and a RACF user ID are specified on initACEE, then initACEE will create an ACEE for that user ID as it usually would and not perform mapping. Again, it will include the IDID information on the RACROUTE REQUEST=VERIFY call.

When an IDID is specified on RACROUTE REQUEST=VERIFY, RACF uses the other parameters to create the ACEE as it normally does, but will anchor the IDID information in the ACEE for later use during auditing.

{IA.5::IA.5-R12-IDPROP-RACF-2} RACF provides a ‘RACMAP’ command to allow the security administrator to define ‘mapping filter rules’ to RACF that will support the mapping of distributed user identities, as specified within the IDID data area, into RACF userIDs as required by the customer. This new RACF command is similar to the existing RACDCERT command, which allows the specification of mapping filter rules that RACF uses to map distributed user identities based on the ‘subject’ and ‘issuer’ information within Digital Certificates. But instead of being limited to only user identities within Digital Certificates, the command supports the definition of mapping filter rules within the IDIDMAP class based on an x.500 representation of the user identity and the ‘Name-Space’ that the user is defined within.

{IA.5::IA.5-R12-IDPROP-RACF-3} The RACF R_cacheserv callable service provides a function (function code 7) that will extract a copy of the ACEE for the currently active user in the form of a RACF environment object (aka RACO), save that RACO in a data space, and return a context reference (ICRX) that will uniquely identify that saved RACO. Subsequently an invoker of RACROUTE REQUEST=VERIFY can provide that ICRX and RACF will recreate the security environment (ACEE) of the original user from the RACO or from the IDID information in the ICRX if necessary. R_cacheserv will also allow deletion of a cached security environment.

{IA.5::IA.5-R12-IDPROP-RACF-5} The RACF R_cacheserv service can also return a pseudo-userID and pseudo-password that RACF authentication functions (initACEE, RACROUTE REQUEST=VERIFY) will subsequently accept and use to create an ACEE for the previously specified RACF user ID with an ICTX data area cached on the earlier R_cacheserv invocation. The pseudo-userID and pseudo-password may be used at most once on a subsequent authentication request.

{IA.5::IA.5-R12-IDPROP-RACF-4} RACF will provide an ENF signal when an administrator has issued an ALTUSER REVOKE or a CONNECT or REMOVE command that changes a user's group connections, allowing applications that have cached ACEEs locally or via R_cacheserv to remove their cache entries and recreate the ACEEs if needed.

{IA.5::IA.5-R12-IDPROP-USS-1} The UNIX System Services __passwd (BPX1PWD) and pthread_security_np() (BPX1TLS) function allows appropriately authorized servers to assert a user identity and create a security environment by specification of the pseudo-userID and pseudo-password obtained via a prior authentication and use of R_cacheserv.

8.3 Access control

8.3.1 Access control principles

The Resource Access Control Facility (RACF) is the component that performs access control between subjects acting on behalf of a user and resources protected by the discretionary and (if active) mandatory access control policies. RACF uses user and resource profiles it stores in the RACF database to decide if a subject has access to a non-UNIX resource. For UNIX resources, the access permissions are carried with the resource itself (permission bits).

While external resource manager can use RACF to manage and control access of users to the resources they control, RACF acts also as a resource manager to some of its own resources. For external resource manager RACF neither knows what the actual resource protected by a specific RACF profile is nor knows the semantics the resource manager places on the individual access modes. Therefore this summary specification only describes the purpose of profiles and the semantics of access modes that protect resources owned by RACF itself.

All z/OS components that have to make access decisions will call RACF through a z/OS interface. The following figure shows the flow of requests and replies within z/OS when a request to access a protected resource is made.

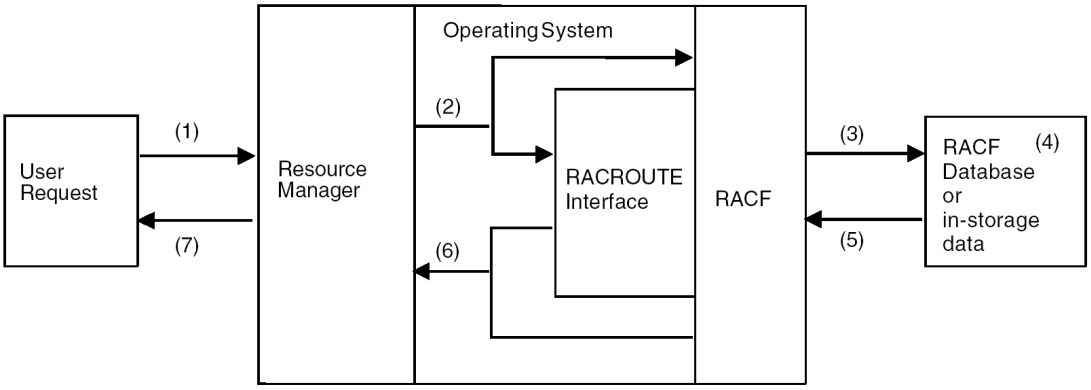

Figure 1: RACF and its relationship to the operating system

A program that wants to access a resource uses a function that is part of the external interface provided by the z/OS operating system to one of the z/OS components (1). An example is a program that wants to open a data set.

The z/OS component responsible for managing the resource calls the RACF component using the internal interface to RACF (mainly the RACROUTE interface) to check the access rights of the user that initiated the user request and passes the name and type of the resource and the requested type of access to RACF {AC.1::AC.1.1}. The caller may also pass the ID of the user or an explicit user security context (ACEE), or RACF obtains those values from the security context of the user that has been established during user authentication (2) {AC.1::AC.1.2}.

RACF extracts the user information from the security context of the user or (in a few cases) from the user profile, extracts the resource profile from its external database or the internal cache (3), and checks to see if the user with his current security attributes is allowed to access the resource in the requested access mode (4 and 5).

If the resource is known to RACF, RACF returns either a “yes” or a “no” decision for the access request {AC.1::AC.1.3}. If the resource is not known to RACF, RACF may return a “don’t know” return code unless there are specific options set that allow RACF to take a yes or no decision (6) {AC.1::AC.1.4}. In the case of a “don’t know” result, the resource manager needs to make its own decision whether to allow access or not. Depending on the decision, the resource manager will either perform or reject the access request of the user program (7) {AC.1::AC.1.5}.

The protection philosophy of RACF is based on “profiles” that represent protected resources but also users and groups. Profiles are organized in profile classes, where each class represents a type of resource (such as data sets or terminals) or other entity (such as users or groups). A profile stores attributes of the subject or object it represents.

For profiles that represent a protected resource, an access list can be assigned {AC.1::AC.1.6}. This access list specifies the type of access subjects may have to the resource represented by the profile.

Access control to UNIX file system objects and IPC objects are also handled by RACF, but in the case of these objects, the access rights are stored with the object itself. RACF still performs the access check. For details, see the description of access control for UNIX objects.

8.3.2 Protected resources

Resource profiles of RACF are structured into an open set of “resource classes”. IBM provides a set of resources classes used by z/OS (stored in the “static class descriptor table”), but RACF also allows for the definition and activation of additional resource classes using the RDEFINE or RALTER commands addressing the CDT general resource class (those are stored in the “dynamic class descriptor table”). The dynamically defined classes need to be “activated” using the command SETROPTS RACLIST(CDT) REFRESH. Resource classes represent “types” of objects that are access protected by RACF. IBM supplies a default static class descriptor table, which is structured into resources used by different components of z/OS as well as resources used by specific other IBM products like DB2 or CICS.

8.3.2.1 General z/OS Resource Classes

IBM supplies a class descriptor table that defines classes used by z/OS or other IBM products that use the services of RACF for controlling access to the resources they manage. The following tables list the classes of general resources used by z/OS and DB2. Resource classes marked in red include resources that are used by RACF internally for managing access to RACF resources or use of RACF controlled privileges.

Class Name	Purpose
ALCSAUTH	Supports the Airline Control System/MVS (ALCS/MVS) product.
APPCLU	Verifying the identity of partner logical units during VTAM session establishment.
APPCPORT	Controlling which user IDs can access the system from a given LU (APPC port of entry). Also, conditional access to resources for users entering the system from a given LU.
APPCSERV	Controlling whether a program being run by a user can act as a server for a specific APPC transaction program (TP).
APPCSI	Controlling access to APPC side information files.

Class Name	Purpose
APPCTP	Controlling the use of APPC transaction programs.
APPL	Controlling access to applications.
CACHECLS	Contains profiles used for saving and restoring cache contents from the RACF database. See the description of the R_cacheserv RACF callable service.
CBIND	Controlling the client's ability to bind to the server.
CDT	Contains profiles for installation-defined classes for the dynamic CDT.
CFIELD	Contains profiles that define the installation's custom fields.
CONSOLE	Controlling access to MCS consoles. Also, conditional access to other resources for commands originating from an MCS console.
DASDVOL	DASD volumes.
DBNFORM	Reserved for future IBM use.
DEVICES	Used by MVS allocation to control who can allocate devices such as: $\frac{35}{17}$ Unit record devices (printers and punches) (allocated only by PSF, JES2, or JES3) $\frac{35}{17}$ Graphics devices (allocated only by VTAM) Teleprocessing (TP) or communications devices (allocated only by VTAM)
DIGTCERT	Contains digital certificates and information related to them. See chapter 20 of [RACF.SAG] and the description of the RACDCERT command.
DIGTCRIT	Specifies additional criteria for certificate name filters. See chapter 20 of [RACF.SAG] and the description of the RACDCERT command.
DIGTNMAP	Mapping class for certificate name filters. See chapter 20 of [RACF.SAG] and the description of the RACDCERT command.
DIGTRING	Contains a profile for each key ring and provides information about the digital certificates that are part of each key ring. See chapter 20 of [RACF.SAG] and the description of the RACDCERT command.
DIRAUTH	Setting logging options for RACROUTE REQUEST=DIRAUTH requests. Also, if the DIRAUTH class is active, security label authorization checking is done when a user receives a message sent through the TPUT macro or the TSO SEND, or LISTBC commands.
DLFCLASS	The data lookaside facility.
FACILITY	Miscellaneous uses. Profiles are defined in this class so resource managers (typically elements of z/OS or z/VM) can check a user's access to the profiles when the user takes some action. Examples are the profiles used to control execution of RACDCERT command functions and the profiles used to control privileges in the z/OS UNIX environment. RACF does not document all of the resources used in the FACILITY class by other products. For information on the FACILITY class resources used by a specific product (other than RACF itself), see that product's documentation.
FIELD	Fields in RACF profiles (field-level access checking).
FSACCESS	Allows control of access to the root of a zFS file system through RACF resource profiles rather than UNIX permission bits or ACLs.
GDASDVOL	Resource group class for DASDVOL class.

Class Name	Purpose
GLOBAL	Global access checking table entry.
GMBR	Member class for the GLOBAL class.
GSDSF	Resource group class for SDSF class.
GTERMINL	Resource group class for TERMINAL class.
GXFACILI	Grouping class for XFACILIT resources.
IBMOPC	Controlling access to OPC/ESA subsystems.
IDIDMAP	Contains distributed identity filters created with the RACMAP command.
JESINPUT	Conditional access support for commands or jobs entered into the system through a JES input device.
JESJOBS	Controlling the submission and cancellation of jobs by job name.
JESSPOOL	Controlling access to job data sets on the JES spool (that is, SYSIN and SYSOUT data sets).
KEYSMSTR	Contains profiles that hold keys to encrypt data stored in the RACF database, such as LDAP BIND passwords and DCE passwords.
LDAPBIND	Contains the LDAP server URL, bind distinguished name, and bind password.
LOGSTRM	Controls system logger resources, such as log streams and the coupling facility structures associated with log streams.
NODES	Controlling the following on MVS systems: 35 17 Whether jobs are allowed to enter the system from other nodes 35 17 Whether jobs that enter the system from other nodes have to pass user identification and password verification checks
NODMBR	Member class for the NODES class.
OPERCMDS	Controlling who can issue operator commands (for example, JES and MVS, and operator commands).
PMBR	Member class for the PROGRAM class.
PROGRAM	Protects executable programs.
PROPCNTL	Controlling if user ID propagation can occur, and if so, for which user IDs (such as the CICS or IMS main task user ID), user ID propagation is not to occur.
PSFMPL	Used by PSF to perform security functions for printing, such as separator page labeling, data page labeling, and enforcement of the user printable area.
PTKTDATA	PassTicket key class enables the security administrator to associate a RACF secured sign-on secret key with a particular mainframe application that uses RACF for user authentication. Examples of such applications are IMS, CICS, TSO, z/VM, APPC, and MVS batch.
RACFEVNT	Contains profiles that control the following events: 35 17 LDAP change log notification for changes to certain RACF profiles New password and password phrase enveloping for a given user.
RACFHC	Used by IBM Health Checker for z/OS. Contains profiles that list the re-

Class Name	Purpose
	sources to check for each installation-defined health check.
RACFVARS	RACF variables. In this class, profile names, which start with & (ampersand), act as RACF variables that can be specified in profile names in other RACF general resource classes. See [RACF.SAG], chapter 7.
RACGLIST	Class of profiles that hold the results of RACROUTE REQUEST=LIST,GLOBAL=YES or a SETROPTS RACLIST operation.
RACHCMBR	Used by IBM Health Checker for z/OS. Member class for the RACHCMBR class.
RDATALIB	Used to control use of the R_datalib callable service (IRRSDL00 or IRRSDL64).
RRSFDATA	Used to control RACF remote sharing facility (RRSF) functions.
RVARSMBR	Member class for the RACFVARS class.
SCDMBR	Member class for the SECDATA class.
SDSF	Controls the use of authorized commands in the System Display and Search Facility (SDSF). See also GSDSF class.
SECDATA	Security classification of users and data (security levels and security categories).
SECLABEL	If security labels are used, and, if so, their definitions.
SECLMBR	Member class for the SECLABEL class.
SERVAUTH	Contains profiles used by servers to check a client's authorization to use the server or to use resources managed by the server. Also, can be used to provide conditional access to resources for users entering the system from a given server.
SERVER	Controlling the server's ability to register with the daemon.
SMESSAGE	Controlling to which users a user can send messages (TSO only).
SOMDOBJS	Controlling the client's ability to invoke the method in the class.
STARTED	Used in preference to the started procedures table to assign an identity during the processing of an MVS START command.
SURROGAT	If surrogate submission is allowed, and if allowed, which user IDs can act as surrogates.
SYSMVIEW	Controlling access by the SystemView for MVS Launch Window to SystemView for MVS applications.
TAPEVOL	Tape volumes.
TEMPDSN	Controlling who can access residual temporary data sets.
TERMINAL	Terminals (TSO or z/VM). See also GTERMINL class.
VTAMAPPL	Controlling who can open ACBs from non-APF authorized programs.
WRITER	Controlling the use of JES writers.
XFACILIT	Miscellaneous uses. Profile names in this class can be longer than 39 characters in length. Profiles are defined in this class so that resource managers (typically elements of z/OS) can check a user's access to the resources when the users take some action.

Table 19: General Resource Classes

DB2 Resource Classes

DB2 Resource classes are mentioned here to support the evaluation of DB2. Within this Security Target they are not used for any security function. The IBM supplied class descriptor table just contains those classes (as well as other classes related to applications like CICS, IMS, or Websphere), which are not used unless those applications are installed and configured to use RACF.

DSNADM	DB2 administrative authority class.
DSNR	Controls access to DB2 subsystems.
GDSNBP	Grouping class for DB2 buffer pool privileges.
GDSNCL	Grouping class for DB2 collection privileges.
GDSNDB	Grouping class for DB2 database privileges.
GDSNJR	Grouping class for Java archive files (JARs).
GDSNPK	Grouping class for DB2 package privileges.
GDSNPN	Grouping class for DB2 plan privileges.
GDSNSC	Grouping class for DB2 schemas privileges.
GDSNSG	Grouping class for DB2 storage group privileges.
GDSNSM	Grouping class for DB2 system privileges.
GDSNSP	Grouping class for DB2 stored procedure privileges.
GDSNSQ	Grouping class for DB2 sequences.
GDSNTB	Grouping class for DB2 table, index, or view privileges.
GDSNTS	Grouping class for DB2 tablespace privileges.
GDSNUF	Grouping class for DB2 user-defined function privileges.
GDSNUT	Grouping class for DB2 user-defined distinct type privileges.
MDSNBP	Member class for DB2 buffer pool privileges.
MDSNCL	Member class for DB2 collection privileges.
MDSNDB	Member class for DB2 database privileges.
MDSNJR	Member class for Java archive files (JARs).
MDSNPK	Member class for DB2 package privileges.
MDSNPN	Member class for DB2 plan privileges.
MDSNSC	Member class for DB2 schema privileges.
MDSNSG	Member class for DB2 storage group privileges.
MDSNSM	Member class for DB2 system privileges.
MDSNSP	Member class for DB2 stored procedure privileges.
MDSNSQ	Member class for DB2 sequences.
MDSNTB	Member class for DB2 table, index, or view privileges.
MDSNTS	Member class for DB2 tablespace privileges.

MDSNUF	Member class for DB2 user-defined function privileges.
MDSNUT	Member class for DB2 user-defined distinct type privileges.

Table 20: General Resource Classes for DB2

Enterprise Identity Mapping (EIM) Classes

RAUDITX	Controls auditing for Enterprise Identity Mapping (EIM).

Table 21: General Resource Classes for EIM

ICSF Resource Classes

CRYPTOZ	Controls access to PKCS #11 tokens.
CSFKEYS	Controls access to ICSF cryptographic keys.
CSFSERV	Controls access to ICSF cryptographic services.
GCSFKEYS	Resource group class for the CSFKEYS class.
GXCSFKEY	Resource group class for the XCSFKEY class.
XCSFKEY	Controls the exportation of ICSF cryptographic keys.

Table 22: General Resource Classes for ICSF

PSF Resource Classes

PRINTSRV	Controls access to printer definitions for Infoprint Server.

Table 23: General Resource Classes for PSF

Kerberos Resource Classes

KERBLINK	Mapping class for user identities of local and foreign principals.
REALM	Used to define the local and foreign realms.

Table 24: General Resource Classes for Kerberos

DFSMS Resource Classes

MGMTCLAS	SMS management classes.
STORCLAS	SMS storage classes.
SUBSYSNM	Authorizes a subsystem (such as a particular instance of CICS) to open a VSAM ACB and use VSAM record level sharing (RLS) functions.

Table 25: General Resource Classes for DFSMS

TSO Resource Classes

ACCTNUM	TSO account numbers.
PERFGRP	TSO performance groups.
TSOAUTH	TSO user authorities such as OPER and MOUNT.
TSOPROC	TSO logon procedures.

Table 26: General Resource Classes for TSO

UNIX System Services Resource Classes

DIRACC	Controls auditing (using SETROPTS LOGOPTIONS) for access checks for read/write access to z/OS UNIX directories. This class need not be active to control auditing.
DIRSRCH	Controls auditing (using SETROPTS LOGOPTIONS) of z/OS UNIX directory searches. This class need not be active to control auditing.
FSOBJ	Controls auditing (using SETROPTS LOGOPTIONS) for all access checks for z/OS UNIX file system objects except directory searches. Controls auditing (using SETROPTS AUDIT) of creation and deletion of z/OS UNIX file system objects. This class need not be active to control auditing.
FSSEC	Controls auditing (using SETROPTS LOGOPTIONS) for changes to the security data (FSP) for z/OS UNIX file system objects. This class need not be active to control auditing. When this class is active, it also controls whether ACLs are used during authorization checks to z/OS UNIX files and directories.
IPCOBJ	Controls auditing (using SETROPTS LOGOPTIONS) of access checks for inter-process communication (IPC) objects and changes to security information of IPC objects. Controls auditing (using SETROPTS AUDIT) of the creation and deletion of IPC objects. This class need not be active to control auditing.
PROCACT	Controls auditing (using SETROPTS LOGOPTIONS) of functions that look at data from, or affect the processing of, z/OS UNIX processes. This class need not be active to control auditing.
PROCESS	Controls auditing (using SETROPTS LOGOPTIONS) of changes to UIDs and GIDs of z/OS UNIX processes. Controls auditing (using SETROPTS AUDIT) of dubbing and undubbing of z/OS UNIX processes. This class need not be active to control auditing.
UNIXMAP	Contains profiles that are used to map z/OS UNIX UIDs to RACF user IDs and z/OS UNIX GIDs to RACF group names.
UNIXPRIV	Contains profiles that are used to grant z/OS UNIX privileges.

Table 27: General Resource Classes for z/OS Unix

8.3.2.1 Installation Defined Resource Classes

As a general-access control system, RACF is capable of protecting a number of other resources including installation defined resource classes, but those are not included in this evaluation. The reader should note that some RACF classes are included in this evaluation that the resource

managers do not use to represent “resources” but represent privileges or restrictions, where assigning “access” to a resource in such a class to a user or a group just determines that the user or group has the privilege or restriction associated with the profile. As stated before it is up to the resource manager to make the association between the real "resource" and the profiles within RACF that represent the resource. It is also up to the resource manager to determine the semantics of specific access types to the resources it manages. Classes with resources that are used by RACF are marked in red.

8.3.2.2 Data sets

8.3.2.2.1 Standard data set naming conventions

By default, RACF expects a data set name (and the data set profile name) to consist of at least two qualifiers. RACF also expects the high-level qualifier of the data set profile name to be either a RACF-defined user ID or a RACF-defined group name.

If an installation has chosen to define data set profiles under the standard RACF naming conventions, they can create a group for each high-level qualifier that is not a user ID, and permit users to protect any data set that has that high-level qualifier by giving them CREATE authority in that group {AC.2::AC.2.1}.

8.3.2.2.2 Table-driven data set naming conventions

An installation can use the naming convention table to set up and enforce a data set naming convention other than that used by RACF (AC.2.2). The table can:

- Supply a qualifier to be used as the high-level qualifier for authorization checking {AC.2::AC.2.3}
- Convert data set names to RACF naming convention form for RACF use {AC.2::AC.2.4}
- Convert names in RACF form to the installation’s format for external display {AC.2::AC.2.5}
- Enforce a naming convention by not allowing the definition of data sets that do not conform to an installation’s rules {AC.2::AC.2.6}
- Reduce RACF overhead by determining whether a data set is a user or group data set

  An installation can create a naming convention table (module ICHNCV00), which RACF uses to check and modify (internally to RACF) the data set name in all commands and macros that process data set names {AC.2::AC.2.7}. An installation can use the table to selectively rearrange data set names to “fit” the RACF convention without actually changing those names.

8.3.2.2.3 Protecting data sets that have single-qualifier data set names

If some of the data sets in an installation have names that consist of a single qualifier, one can still RACF-protect those data sets {AC.2::AC.2.8}. To get RACF protection for single-qualifier names, the SETROPTS command with the PREFIX operand must be issued.

This command defines a high-level qualifier to be used as a prefix for single-qualifier names and activates the facility {AC.2::AC.2.9}. Then, when RACF processes requests for the data set, RACF internally modifies single-qualifier names by adding the prefix, making the data set names acceptable to RACF routines {AC.2::AC.2.10}. All SMF log records and all messages from RACF contain the RACF-modified version of the data set name {AC.2::AC.2.11} unless the SETROPTS REALDSN option is in effect {AC.2::AC.2-R10-RACF-1}.

8.3.2.2.4 Protecting user data sets

A user data set is a data set whose high-level qualifier is a RACF user ID. The following rules apply to user data sets:

- In general, all RACF-defined users can protect their own data sets {AC.2::AC.2.12}
- A user can RACF-protect a data set for another user under any of the following conditions:

– The user who is protecting the data set has the SPECIAL attribute. A discrete or generic profile can be created {AC.2::AC.2.13}.

– The user who is protecting the data set has the group-SPECIAL attribute, and the high-level-qualifier of the data set name is a user within the group-SPECIAL user's scope of authority. A discrete or generic profile can be created {AC.2::AC.2.14}.

– The user who is protecting a data set has the OPERATIONS attribute (or the group-OPERATIONS attribute if the data set is within his scope of authority) and is simultaneously creating the data set {AC.2::AC.2.15}.

In this case, the user can create a discrete profile:

- Through ADSP {AC.2::AC.2.16}
- By specifying the PROTECT operand on the TSO ALLOCATE command that creates the data set {AC.2::AC.2.17}
- By specifying the PROTECT=YES OR SECMODEL= profile-name operands on the JCL DD statement that creates the data set {AC.2::AC.2.18}

8.3.2.2.5 Protecting group data sets

A group data set is a data set whose high-level qualifier is a RACF group name. A RACF-defined user can RACF-protect a group data set under any of the following conditions:

- The user has JOIN, CONNECT, or CREATE authority in the group {AC.2::AC.2.19};
- The user has the SPECIAL attribute (or the group-SPECIAL attribute for that group) and the request is made using the ADDSD command {AC.2::AC.2.20};
- The user has the OPERATIONS attribute and is not connected to the group {AC.2::AC.2.21}.

8.3.2.2.6 Controlling the creation of new data sets

Using data set profiles, an administrator can control whether users can create (allocate) new data sets.

For cataloged data sets, creating, deleting, or renaming the data set involves access not only to the data set profile protecting the data set, but also to the catalog in which the data set is cataloged {AC.2::AC.2.22}. In general, users need the following:

- To add entries to the catalog, users need authority to create the data set as specified below and (except for SMS-managed data sets) UPDATE authority to the catalog {AC.2::AC.2.23}.
- To delete entries from the catalog, users need ALTER authority to the protecting profile or to the catalog {AC.2::AC.2.24}.

The following cases describe how RACF can be used to control the creation of new user and group data sets.

A user can create a new user data set in the following situations:

- The data set is covered by an existing generic profile and the user does not have ADSP {AC.2::AC.2.25}. The creation is allowed if (1) the user has ALTER authority to the data set

through a generic profile or global access checking, or (2) the data set is the user's own data set {AC.2::AC.2.26}.

- The data set name is not covered by an existing generic profile and the user does not have ADSP and the data set is covered by the Global Access check table granting ALTER. {AC.2::AC.2.27}
- The user has ADSP and the data set is the user's own data set. The creation is allowed and RACF creates a discrete profile for the data set {AC.2::AC.2.28}.
- The user has the OPERATIONS attribute. If the user has the group-OPERATIONS attribute (that is, the user is connected to a group with the OPERATIONS attribute), the high-level qualifier of the new data set must be the ID of a user who is within the scope of that group {AC.2::AC.2.29}.

A user can create a new group data set in the following situations:

- The data set name is protected by an existing generic profile and the user does not have ADSP.

  The creation is allowed if at least one of the following is true:

  - The user has ALTER authority to the data set through the generic profile or global access checking {AC.2::AC.2.30}
  - The user has CREATE authority in the group {AC.2::AC.2.31}
- The data set name is not covered by an existing generic profile and the user does not have ADSP {AC.2::AC.2.32}
- The user has ADSP and the data set belongs to a group of which the user is a member. The creation is allowed only if the user has CREATE authority in the group. If the creation is allowed, RACF creates a discrete profile for the data set {AC.2::AC.2.33}
- {AC.2::AC.2.36-R12-RACF}The user has the OPERATIONS attribute , or the group-OPERATIONS attribute for the group in question (directly or via a superior group), except when both of the following are true: The user is connected to the group with less than CREATE authority {AC.2::AC.2.34-R12-RACF}, and the user has less than ALTER access to the data set if it protected by a generic profile {AC.2::AC.2.35-R12-RACF}

8.3.2.2.7 Data set profile ownership

Each data set profile defined to RACF requires a RACF-defined user or group as the owner of the profile. The owner (if a user) has full control over the profile, including the access list {AC.2::AC.2.37}.

If the owner of the data set profile is a group, users with group-SPECIAL in that group have full control over the profile {AC.2::AC.2.38}.

Ownership of data set profiles is assigned when the profiles are defined to RACF but may be changed later. Note that ownership of a data set profile does not mean that the owner can automatically access that data set. To access a data set, the owner must still be authorized by the DAC and (in Labeled Security Mode) MAC policy rules {AC.2::AC.2.39}.

8.3.2.3 Programs

The ability of users to execute programs can be restricted by the RACF program control function. This feature is useful for programs operating with privileges like authorized programs. Program control can for example be used to restrict the ability of a user to start an authorized program from an authorized library in a way such that it executes with APF authorization {AC.2::AC.2-V1R7-1}. Users may still

have read access to the library and may therefore copy the program into another library and execute it from this library. Although this is possible, the program will then not execute with the privileges it has when executed from the original library {AC.2::AC.2-V1R7.2}.

Program control (as described in this section) applies to programs residing in z/OS partitioned data sets or libraries, not to programs stored as part of z/OS UNIX file system. Mechanisms for program control for the z/OS UNIX subsystem are explained in another section of this Security Target.

z/OS allows for three modes for program control: BASIC, ENHANCED and ENHANCED-WARNING. The mode is defined by the strings 'BASIC', 'ENHANCED' or 'ENHANCED-WARNING' in the APPLDATA field of the IRR.PGMSECURITY profile in the FACILITY class {AC.2::AC.2.V1R7.3}. An empty value or any other value than 'BASIC' or 'ENHANCED' will result in the ENHANCED-WARNING mode {AC.2::AC.2.V1R7.4}. If the IRR.PGMSECURITY profile is not defined, BASIC mode is used {AC.2::AC.2.V1R7.5}. In ENHANCED-WARNING mode the access decisions made by the TOE are the same as in BASIC mode but a warning message is issued whenever the access would have been denied in ENHANCED mode {AC.2::AC.2.V1R7.6}.

The checks that RACF makes when a user makes a request to load (execute) a program are:

1. If program control has been activated with SETROPTS WHEN(PROGRAM) {AC.2::AC.2-V1R7.7}
2. If program control is active, RACF checks to see whether the program is protected by a profile in the PROGRAM class {AC.2::AC.2-V1R7.8}
3. If the program is not protected, RACF determines whether there are any data sets currently open using PADS or whether there are any execute-controlled programs in storage in the address space:
    - If there are no such data sets or programs, RACF marks the environment dirty (uncontrolled) and allows the user to execute the program {AC.2::AC.2-V1R7.9}.
    - If there are data sets currently opened using PADS, or programs to which the user has only EXECUTE authority, RACF fails the request and the system abends the task. RACF issues message ICH423I to document the execute-controlled programs, or message ICH424I to document the PADS data sets that caused the operation to fail. In this way, RACF prevents uncontrolled programs from gaining access to protected data or programs inappropriately {AC.2::AC.2-V1R7.10}.
4. If the program is protected by a profile but the user does not have at least EXECUTE authority to the program, RACF causes the system to abend the task because the user is not authorized to execute the program {AC.2::AC.2-V1R7.11}.
5. If the program is protected by a profile and the user has only EXECUTE authority to the PROGRAM profile or to the library that contains the program (when the program is loaded from a JOBLIB, STEPLIB, or tasklib), and if the job step or TSO session is running in ENHANCED program security mode, RACF checks whether an appropriate program established the program environment. RACF determines if the first program executed in the job step had the ’MAIN’ attribute, or (if necessary) if the program invoked by TSOEXEC or IKJEFTSR had the ’MAIN’ attribute. If the program does not have MAIN, RACF next determines if the first program run in the current task (TCB) or the first program executed in some parent task had the ’BASIC’ attribute. If so, RACF allows the Program control request. Otherwise, RACF fails the request and issues message ICH429I to describe the problem and tell you what program established the environment {AC.2::AC.2-V1R7.12}.
6. If the user is still authorized to execute the program and the program was defined with the PADCHK attribute, RACF checks whether any program-accessed data sets are open.
    - If no program-accessed data sets are open, RACF allows the user to execute the program {AC.2::AC.2-V1R7.13}.
    - If program-accessed data sets are open, RACF checks the user or program combination to verify that the combination has at least the same authority to each data set in the list that was

required when each data set was opened.

- If the user or program combination has sufficient authority to all of the opened data sets, RACF allows the user to execute the program {AC.2::AC.2-V1R7.14}
- If the user or program combination does not have sufficient authority to all of the opened data sets, RACF causes the system to end the task (with abend code 306 or 806) {AC.2::AC.2-V1R7.15}.

With program control enabled, z/OS provides the ability to allow users to access data sets which they are not allowed to access directly by using program controlled programs {AC.2::AC.2.V1R7.16}.

The following algorithm is used to determine if a user has access to a data set via a controlled program:

Whenever the user has the requested access to the data set as determined by normal RACF access checking, access is granted {AC.2::AC.2.V1R7.17}.

If the user is not granted access to the data set with normal authorization checking, RACF checks the data set's conditional access list if program control is active and the program currently executing is executing as a RACF-controlled program in a clean environment. RACF authorizes the user to open the program-accessed data set with the currently executing program if all of the following conditions are met:

1. The conditional access list contains the name of the currently running program, the name of the first program currently running in the current task (TCB), or the name of the first program currently running in a parent task, with the requested level of access or higher {AC.2::AC.2.V1R7.18}.
2. The user's group or user ID is associated with the program name in the conditional access list {AC.2::AC.2.V1R7.19}.
3. The current program environment (job step, or task established under TSO/E using TSOEXEC or IKJEFTSR) is controlled. In other words, it has not loaded an uncontrolled program. If either of these conditions are not met, the environment is considered uncontrolled. The user's attempt to open the program-accessed data set fails and the task ends with abend code 913. RACF issues message ICH417I, specifying what caused the environment to become uncontrolled {AC.2::AC.2.V1R7.20}.
4. If the job step or TSO session is running in ENHANCED program security mode, one of the following is true:
   - The current environment (job step or task created by TSOEXEC or IKJEFTSR) first ran a program defined with the 'MAIN' attribute.
   - The current program running in the current task, or the first program run in the current task or a parent task, has the BASIC attribute. If neither of these conditions is met, the user's attempt to open the program-accessed data set fails and the task ends with abend code 913. RACF issues message ICH426I, specifying the non-MAIN program that established the current environment {AC.2::AC.2.V1R7.21}.
5. If there is more than one controlled program running in the current environment (job step or task created by TSOEXEC or IKJEFTSR), all of those programs defined with the PADCHK attribute have conditional access list entries allowing them to access the data set. If one or more programs in the environment are not authorized, the attempt fails and the task terminates with abend code 913. RACF issues message ICH418I specifying one or more programs that were missing from the conditional access list {AC.2::AC.2.V1R7.22}.
6. If all the conditions for program access to data set are met and the requested type of access is granted to the program by the profile protecting the data set, access is granted {AC.2::AC.2.V1R7.23}.

8.3.2.4 RACF protection of UNIX file system objects

UNIX file system objects in the HFS or zFS file system have their access control defined by:

- UNIX permission bits
- Access control list entries
- In Labeled Security Mode: security labels (zFS file system)

All of those access-control-related attributes of file system objects are stored with the object. Access control lists and (in Labeled Security Mode) security labels are stored and managed as extended attributes of the file system object and are not stored in the RACF database {AC.2::AC.2.65}. RACF is still involved when an access decision is made to a UNIX file system object {AC.2::AC.2.66}. The UNIX System Services subsystem of the TOE extracts the permission bits, access control list entries and (in Labeled Security Mode) the security label from the file system object as well as the effective user ID and (in Labeled Security Mode) the security label of the user that performed the request and passes this information to RACF. RACF then evaluates this information, extracts other information relevant for the access decision from the RACF database, performs the auditing in accordance with the audit policy defined by the system administrator and returns the access decision to the calling UNIX System Services subsystem {AC.2::AC.2.67}.

Besides the access control lists and (in Labeled Security Mode) the security label, additional privileges and restrictions may be defined to allow a finer granularity. Those privileges and restrictions are defined as profiles in the UNIXPRIV class and users can be granted those privileges or restrictions by giving them authority to those profiles. The ones that are considered in this Security Target are:

- SUPERUSER.FILESYS.ACL.ACLOVERRIDE

When this profile is defined and active in RACF, a user who has been given authority to this profile is able to override the access control defined by the access control lists for z/OS UNIX file system objects.

In z/OS, a UNIX superuser can access all z/OS UNIX files, but is still bound by his rights defined in RACF with respect to z/OS data sets and other resources {AC.2::AC.2.68}. In Labeled Security Mode, a z/OS UNIX superuser is also bound by the mandatory access control rules when accessing z/OS UNIX files {AC.2::AC.2.69}.

8.3.2.5 z/OS UNIX IPC objects

z/OS UNIX IPC objects are subject to discretionary access control. The permission bits associated with the IPC object define the discretionary access to those objects. The permission bits are determined by the creator of the IPC object and are saved in-memory by the UNIX Kernel. For security claims see DAC for UNIX objects.

8.3.3 Mandatory access control (Labeled Security Mode only)

Label based mandatory access control is supported by z/OS. User profiles may contain one or two SECLABEL names, representing defaults for that user (one for TSO/E, and one for other applications) which are the name of profiles in the SECLABEL class. Each profile in the SECLABEL class contains a security classification consisting of a hierarchical security level and a set of non-hierarchical categories. The values for the levels and the categories are defined by the system administrator {AC.3::AC.3.1}. z/OS supports more than 8 levels and more than 60 categories {AC.3::AC.3-R12-RACF-1}. The system administrator can then also define resources in the SECLABEL resource class as a combination of one security level and zero or more categories. Such a resource is called a “security label”.

The system defines a set of predefined security labels:

- SYSHIGH
  This label consists of the highest security level and all categories defined for the system
- SYSLOW
  This label consists of the lowest security level defined for the system and no categories
- SYSNONE
  This is used for resources that need to be read and written by users with different security labels. It needs to be reserved for resources that can only be accessed in a controlled way using trusted programs to avoid a breach of the information flow policy
- SYSMULTI
  This is used for resources that support a range of security labels. It needs to be reserved for resources controlled by trusted programs. Administrators can also be allowed to operate as SYSMULTI. An organization should apply great care when assigning and using this option

z/OS enforces the rules of the Bell-LaPadula model for mandatory access control:

- a subject has read access to an object when:
  - the security level of the subject is higher or equal to the security level of the object
  - the set of categories of the subject includes the set of the categories of the object
  - read access is allowed by the discretionary access control rules {AC.3::AC.3.2}
- a subject has write (update or control) access to an object when
  - the security level of the subject is lower or equal to the security level of the object
  - the set of categories of the object includes the set of categories of the subject
  - write (update or control) access is allowed by the discretionary access control rules {AC.3::AC.3.3}
- a subject has alter access to an object when:
  - the security label of the subject and the security label of the object are identical
  - the user has ALTER access according the discretionary access control rules {AC.3::AC.3.4}

z/OS prohibits the modification of a security label of a resource unless the system is in a state that allows to the activity to be performed in a secure way. This prohibits unauthorized flow of information due to users operating on a resource while the security label of the resource is changed. A change of security labels is restricted to users with the SPECIAL attribute {AC.3::AC.3.V1R7.3}.

The following types of resources are subject to mandatory access control:

- Data sets {AC.3::AC.3.5}
- Volumes (DASD and tape) {AC.3::AC.3.6}
- Devices {AC.3::AC.3.7}
- Terminals {AC.3::AC.3.8}
- TCP/IP connections {AC.3::AC.3.9}
- UNIX file system objects (for zFS file systems and read-only HFS file systems) {AC.3::AC.3.11}
- UNIX IPC objects {AC.3::AC.3.12}

A system administrator can allow a user to bypass the mandatory access control rules. To do this, the administrator needs to define the profile IRR.WRITEDOWN.BYUSER in the FACILITY class and give the user at least READ authority to this profile. A user with this privilege can then activate the ability to

downgrade using the RACPRIV command {AC.3::AC.3.23}.

8.3.4 Discretionary Access Control

Discretionary access control (DAC) applies to all system resources, but the implementation differs depending on the type of resource. This evaluation considers MVS (non-UNIX) resources (RACF general resource classes and data sets), and UNIX resources. RACF provides the discretionary access controls for MVS and UNIX resources. See the sections above on the different profiles for details on what is stored in those profiles.

8.3.4.1 DAC for RACF general resources and data sets

RACF controls the types of access to all MVS (non-UNIX) resources. The access types are ordered hierarchically, an access type listed higher in the list implies all the access types lower in this list (except for NONE access). The full semantics of each access type are defined by the resource manager. This Security Target therefore only defines semantics that RACF generally associates with the access types. The access types are:

- **ALTER**

  When specified in a discrete profile, ALTER allows users to read, alter, and delete the profile itself including the access list {AC.4::AC.4.1}.

  ALTER does not allow users to change the owner of the profile using the ALTDSD command {AC.4::AC.4.2}. However, if a user with ALTER access authority to a discrete data set profile renames the data set, changing the high-level qualifier to his or her own user ID, both the data set and the profile are renamed, and the OWNER of the profile is changed to the new user ID {AC.4::AC.4.3}.

  When specified in a generic profile, ALTER gives users no authority over the profile itself {AC.4::AC.4.4}.

- **CONTROL**

  RACF requires users to have at least CONTROL authority for some of the privileges associated with RACF and managed by profiles that RACF itself uses. The semantics are described with the individual profiles that RACF uses itself. See the sections where the semantics of profiles used by RACF with the semantics of the different access modes is described (e. g. in the processing of privileges used by the RACDCERT command).

- **UPDATE**

  RACF requires users to have at least UPDATE authority for some of the privileges associated with RACF and managed by profiles that RACF itself uses. The semantics are described with the individual profiles that RACF uses itself. See the sections where the semantics of profiles used by RACF with the semantics of the different access modes is described (e. g. in the processing of privileges used by the RACDCERT command).

- **READ**

  RACF requires users to have at least READ authority for some of the privileges associated with RACF and managed by profiles that RACF itself uses. The semantics are described with the individual profiles that RACF uses itself. See the sections where the semantics of profiles used by RACF with the semantics of the different access modes is described (e. g. in the processing of privileges used by the RACDCERT command).

- **EXECUTE**

RACF requires users to have at least EXECUTE authority for some of the privileges associated with RACF and managed by profiles that RACF itself uses. The semantics are described with the individual profiles that RACF uses itself. See the sections where the semantics of profiles used by RACF with the semantics of the different access modes is described. EXECUTE authority is used by RACF itself in the context of program control (see the related section in this ST).

- **NONE**

  The specified user or group is not permitted to access the resource or list the profile {AC.4::AC.4.11}.

These access types can be defined per user, group or for all users not addressed specifically by a user or group access entry (“universal access”) {AC.4::AC.4.12}. It is also possible to specify ID(*) in an ACL, which then applies to all RACF defined users, while the value for UACC applies to users not defined in RACF {AC.4::AC.4.13}. To modify those entries (as well as other parts of the resource profile) a user must be the owner of the profile, have ALTER access to the discrete profile of the resource or must have the SPECIAL attribute in his user profile {AC.4::AC.4.14}.

The access lists defined in a profile can be either a standard access lists, allowing access in general or a conditional access lists allowing access under defined conditions. Possible conditions are:

- the user must be logged on using a defined terminal that the user has been granted access to {AC.4::AC.4.15}
- the user must be logged on to a defined console {AC.4::AC.4.16}
- the batch job requesting access must have been submitted from a defined JES input device {AC.4::AC.4.17}
- the user must have entered the system from a defined network port {AC.4::AC.4.18}
- the resource manager has asserted a criteria, such as the name of an SQL role (SQLROLE), which applies to this check, on the authorization request (note: this applies only to a FASTAUTH type of authorization check) {AC.4::AC.4-R8-RACF-1}.

Access to resources can be controlled by discrete resource profiles or generic profiles for a set of resources of the same type. Discrete profiles protect one single resource (e. g. one data set) while generic profiles can be used to define a whole set of resources and protect them using a single profile based on patterns in the resource name. Whenever a discrete profile exists for a resource it has precedence over a generic profile that also would apply for the resource {AC.4::AC.4.19}. If more than one generic profiles would apply, z/OS always chooses the most specific profile applicable based on a matching algorithm {AC.4::AC.4.20}.

The access types above also apply to MVS resources other than data sets (called general resources). However while the usages remain hierarchical in definition (ALTER includes UPDATE, UPDATE includes READ, etc.) the interpretation and usage of the access types is the responsibility of each resource manager. For most resource managers and resources, the meaningful access types are NONE (the user/group has no access) or READ (ther user/group does have access). For most cases access levels higher than READ convey no added authority (except that ALTER allows administration of a discrete profile). Iin specific cases the resource manager may treat UPDATE, CONTROL, and ALTER as granting additional authority. This security target and evaluation will not address all of those cases.

8.3.4.2 Algorithm to check for DAC access to general resources and data sets

RACF performs the following checks to identify, if a subject has the requested type of access to a resource protected by RACF. This algorithm is performed after RACF has checked that the resource is protected by RACF and (in Labeled Security Mode) after the checks for the mandatory access control have been performed:

1. If users attempt to access their own resources, RACF grants the request {AC.4::AC.4.43}.

2. If the resource manager has performed the authorization check using RACROUTE REQUEST=FASTAUTH (rather than RACROUTE REQUEST=AUTH) and in addition has specified AUTHCHKS=CRITONLY for this check, and has specified a criteria value using the CRITERIA keyword, RACF uses only the criteria-related conditional access list entries to make the determination, and skips to the criteria checking step below {AC.4::AC.4-R8-RACF-2}.
3. RACF checks the user's access authority in the standard access list. If the user is in the list and if the specified access authority is sufficient to allow access, RACF grants the request {AC.4::AC.4.44}. If the user is in the list and if the specified access authority is less than the requested access, RACF continues processing at Step 7 (conditional access list checking) {AC.4::AC.4.45}. This prevents access based on ID(*), UACC, or the OPERATIONS attribute.

   This could happen if, for example, user JOE requests UPDATE access, and the standard access list includes ID(JOE) ACCESS(READ).
4. RACF determines whether the user has access to the resource because the user is a member of a group and the group is on the standard access list {AC.4::AC.4.46}. e)

   Which group is used depends on whether list-of-groups processing is in effect.

   (List-of-groups processing is in effect if the SETROPTS command has been issued with the GRPLIST operand.) RACF determines which group to use according to the following rules:

   - If list-of-groups processing is not in effect, RACF uses only the user's current connect group {AC.4::AC.4.47}.
   - If list-of-groups processing is in effect, RACF finds all of the groups to which the user is connected that are also in the access list. Of these groups, RACF uses the group that has the highest access authority to the resource {AC.4::AC.4.48}. (For example, assume that a user is a member of groups A, B, and C. If group A has NONE access authority, group B has READ access authority, and group C has UPDATE access authority, RACF uses group C to determine the user's access.)

If the highest access authority is sufficient to allow the requested access, RACF grants the request. If the highest group that was found in the list does not have the requested authority, RACF continues processing at Step 8 {AC.4::AC.4.49} (conditional access list checking). This prevents access based on ID(*), UACC, or the OPERATIONS attribute

5. If a user ID of * is found on the standard access list, the current user is defined to RACF without the RESTRICTED attribute, and the access authority granted to * is:
   - Sufficient to allow the requested access, RACF grants the request {AC.4::AC.4.50}
   - Not sufficient to allow the requested access, RACF continues processing at Step 7 {AC.4::AC.4.51} (OPERATIONS attribute checking)
6. If the universal access authority (UACC) for the resource provides sufficient access authority and the requesting user is not defined with the RESTRICTED attribute, RACF grants the request {AC.4::AC.4.52} f)
7. If the requesting user has the OPERATIONS attribute (or group-OPERATIONS if the resource is within the scope of that group) and OPERATIONS access is allowed for the class, RACF grants the request {AC.4::AC.4.53} g)
8. RACF checks the user's access authority in the conditional access list specified with WHEN(TERMINAL), WHEN(CONSOLE), WHEN(SERVAUTH), or WHEN(JESINPUT). If the user is in the list, if the user meets the specified condition (such as logged on at the specified terminal), and if the specified access authority is sufficient to allow access, RACF grants the request {AC.4::AC.4.54} h)
9. RACF determines whether the user has access to the resource because the user is a member of

a group that meets a condition specified on the conditional access list specified with WHEN(TERMINAL), WHEN(CONSOLE), WHEN(SERVAUTH), or WHEN(JESINPUT). Which group is used depends on whether list-of-groups processing is in effect.

If the group to be used according to the preceding rules has sufficient access authority to allow the requested access, RACF grants the request {AC.4::AC.4.55}. If none of the user's groups has sufficient authority, RACF continues with the next step i)

10. If a user ID of * is found on the conditional access list specified with WHEN(TERMINAL), WHEN(CONSOLE), WHEN(SERVAUTH), or WHEN(JESINPUT), and if the current user is defined to RACF without the RESTRICTED attribute, and if the current user meets the specified condition (such as logged on at the specified terminal), and the access authority granted to * is sufficient to allow the requested access, RACF grants the request {AC.4::AC.4.56} j)
11. RACF checks the user's access authority in the conditional access list specified with WHEN(PROGRAM). If the user is in the list, if the user meets the specified condition (such as running the specified program), and if the specified access authority is sufficient to allow access, RACF grants the request {AC.4::AC.4.57}. h)

    Note: For DASD data sets, if program control is active and a controlled program is executing, RACF performs authorization checking for program access to data sets. If the user/program combination is in the conditional access list with sufficient authority to allow access to the data sets, RACF grants the request {AC.4::AC.4.58}.
12. RACF determines whether the user has access to the resource because the user is a member of a group that meets a condition specified on the conditional access list (such as running a specified program). Which group is used depends on whether list-of-groups processing is in effect. i)

    If the group to be used according to the preceding rules has sufficient access authority to allow the requested access, RACF grants the request {AC.4::AC.4.59}. If the group is in the list and if the specified access authority is NONE, RACF denies the request {AC.4::AC.4.60}.
13. If a user ID of * is found on the conditional access list specified with WHEN(PROGRAM), and if the current user is defined to RACF without the RESTRICTED attribute, and if the current user meets the specified condition (such as logged on at the specified terminal or running the specified program), and the access authority granted to * is sufficient to allow the requested access, RACF grants the request {AC.4::AC.4.61} j)
14. Criteria Checking: For RACROUTE REQUEST=FASTAUTH, if the resource manager has asserted an SQL role name (SQLROLE) via the CRITERIA keyword, RACF checks for authority (via the user ID, a group, or * (for non-RESTRICTED users)) in the conditional access list specified with WHEN(SQLROLE(…)), and if the specified access authority is sufficient to allow access, RACF grants the request {AC.4::AC.4-R8-RACF-3}. If the resource manager has also specified AUTHCHKS=CRITONLY, and this step did not grant access, RACF denies the request {AC.4::AC.4-R8-RACF-4}. i)
15. For access to uncataloged data sets, if SETROPTS CATDSNS is in effect, and none of the following is true, then RACF denies the request {AC.4::AC.4.62}:
    - The data set is newly-created in this job, or is a system temporary data set;
    - The data set is protected by a discrete profile;
    - The data set is cataloged in the Master catalog;
    - The user has access to FACILITY resource ICHUNCAT.data set-name (truncated to 39 characters total, if needed);
    - The user has the SPECIAL attribute

16. For the DATASET class, if no profile is found and the SETROPTS PROTECTALL(FAILURES) option is in effect, RACF denies the request {AC.4::AC.4.63}.

If none of the above steps has granted access and the call to RACF has provided a nested ACEE and RACF is called with RACROUTE REQUEST=FASTAUTH and the object is eligible for nested ACEE processing, the algorithm for both mandatory and discretionary access control is repeated using the user ID specified in the nested ACEE {AC.4::AC.4-V1R7.1}. If audit is configured to audit the access attempt, both user IDs (the original and the nested) are contained in the audit record {AC.4::AC.4.V1R7.2}.

8.3.4.3 DAC for UNIX objects

DAC controls for UNIX objects involve the user's effective UID and effective GID (which may be different from the user's real UID and real GID) {AC.4::AC.4-R8-USS-1} and the user's supplemental GIDs. If the user is connected to 5 groups, and 3 of them have GIDs, then he would have one real GID and 2 supplemental GIDs {AC.4::AC.4-R8-USS-2}.

DAC checking for UNIX file objects (files, directories) involves permission bits that specify the permissions (read, write, execute/search) separately for the object's owner, the owning group, and everyone else (the world), and optional access list entries (ACLs) with similar permission settings.

DAC checking for UNIX IPC objects (semaphores, shared memory) involves only permission bits.

8.3.4.4 Algorithm to check DAC access to UNIX file system objects

The following algorithm is used in the evaluated configuration to check the access to UNIX file system objects. The checks are performed by RACF using the effective user and group ID respectively.

1. (Step performed in Labeled Security Mode only) Access to the file system object must be allowed by the mandatory access control function. If not, access is denied {AC.4::AC.4.21}. 0

2. If the FSACCESS class is active and SETROPTS RACLISTed, and if the user is accessing the root directory in a mounted zFS file system (but not the system root directory), then if the MVS data set name of the file system container is protected by a profile in the FSACCESS class the user must have UPDATE access to that FSACCESS profile or the request will fail. Note that this is a standard RACF access check using the capabilities described above for DAC for MVS resources. {AC.4:: AC.4-R13-UNIX-1} 0

3. If the user has the RACF AUDITOR attribute, and read or search access for a directory is requested, access is granted {AC.4::AC.4.22}. a

4. If the user has UID(0), or has the TRUSTED or PRIVILEGED attribute, then access is granted automatically unless the user is executing a file. If the user is executing a file, access is denied only if none of the permissions bits grant execute access, and, if an ACL is present and the FSSEC class is active, no ACL entry grants execute access. Otherwise, access is granted {AC.4::AC.4.23}. b

5. If the user does not have search permission to all directories in the path of the file system object, access is denied {AC.4::AC.4.24}. 0

6. If the UID matches the file owner UID, the file's "owner" permission bits are checked. If the "owner" bits allow the requested access, then access is granted {AC.4::AC.4.25}. If the UID matches the file owner UID and the owner bits do not allow the requested access, go to Step 15 {AC.4::AC.4.26}. c

7. If the FSSEC class is active, and an ACL exists, and there is an ACL entry for the requesting UID, then the permission bits of that ACL entry are checked. If the ACL entry allows the requested access, then access is granted {AC.4::AC.4.27}. Otherwise, if the ACL for the UID exists, but does not allow access, go to Step 14 {AC.4::AC.4.28}. d

8. If the GID matches the file owner GID, the file's "group" permission bits are checked. If the "group" bits allow the requested access, then access is granted {AC.4::AC.4.29}. f
9. If the FSSEC class is active, and an ACL exists, and there is an ACL entry for the requesting GID, then the permission bits of that ACL entry are checked. If the ACL entry allows the requested access, then access is granted {AC.4::AC.4.30}. If not, then the next ACL entry is checked until there are no more entries {AC.4::AC.4.31}. d
10. If any of the user's supplemental GIDs match the file owner GID, the file's "group" permission bits are checked. If the "group" bits allow the requested access, then access is granted {AC.4::AC.4.32}. f
11. If the FSSEC class is active, and an ACL exists, and there is an ACL entry for any of the user's supplemental GIDs, then the permission bits of that ACL entry are checked. If the ACL entry allows the requested access, then access is granted {AC.4::AC.4.33}. If not, then the next ACL entry is checked until there are no more entries {AC.4::AC.4.34}. d
12. If at least one matching ACL entry was found for the GID, or any of the supplemental GIDs, then processing continues with Step 14 {AC.4::AC.4.35}. If the GID, or any of the supplemental GIDs, matched the file owner GID, then processing continues with Step 15 {AC.4::AC.4.36}. Otherwise (neither the GID nor any of the supplemental GIDs matched either the file owner GID or an ACL entry), processing continues with the next step {AC.4::AC.4.37}.
13. If the requesting user has the RESTRICTED attribute, and the UNIXPRIV class is active and RACLISTed, and the RESTRICTED.FILESYS.ACCESS resource is protected by a profile in the UNIXPRIV class, and the user does not have at least READ access, then go to Step 15 {AC.4::AC.4.38}. i
14. The file's "other" permission bits are checked. If the "other" bits allow the requested access, then access is granted {AC.4::AC.4.39}. Otherwise, go to Step 15.
15. If the UNIXPRIV class is active and RACLISTed, and if the SUPERUSER.FILESYS.ACLOVERRIDE resource is protected by a profile in the UNIXPRIV class, then the user must have the correct access level as documented for the ck_access (IRRSKA00) callable service in z/OS Security Server RACF Callable Services. If the profile exists, it determines whether file access is granted or denied {AC.4::AC.4.40}. k
16. If the UNIXPRIV class is active and RACLISTed, and if the SUPERUSER.FILESYS resource is protected by a profile in the UNIXPRIV class, then the user must have the correct access level as documented for the ck_access (IRRSKA00) callable service in z/OS Security Server RACF Callable Services. If the profile exists, it determines whether file access is granted or denied {AC.4::AC.4.41}.

Access is denied, if none of the above steps has explicitly granted access {AC.4::AC.4.42}. l

8.3.4.5 Algorithm to check DAC access to UNIX IPC objects

The discretionary access control rules allow access to an IPC object,

- if the user has an effective user ID of zero {AC.4::AC.2.70}
- if the user is the owner or creator of the IPC object and the requested type of access is allowed by the owner related permission bits {AC.4::AC.2.71}
- if the user is neither the owner or creator of the IPC object but is a member of the IPC object's creating group or owning group and the requested type of access is allowed by the group related permission bits {AC.4::AC.2.72}
- if the user is neither owner nor creator of the IPC object and also is not a member of the IPC object's creating group or owning group and the access is allowed by the other related permission bits {AC.4::AC.2.73}

If none of the above mentioned conditions is satisfied, permission is denied by the discretionary access control rules for IPC objects {AC.4::AC.2.74}.

8.4 Security management

8.4.1 User and group management

8.4.1.1 Definition of users and groups

z/OS users and groups are defined in RACF.

Local Kerberos users are defined as z/OS users who also have a KERB segment in their RACF USER profile. A remote (foreign) Kerberos user may be defined locally by mapping the foreign principal name to a local z/OS (RACF) user via KERBLINK profiles.

To create a z/OS user, a user profile for the new user has to be created in RACF. Each user profile consists of a base segment and optional segments for the use of specific subsystems. In the evaluated configuration, the base segment, the KERB segment, and the OMVS segment for the specification of attributes for z/OS UNIX System Services contain the information required by the security functions defined in this Security Target. Other segments of the user profile may exist but the effects of any values in those segments do not influence the security policy defined in this Security Target. RACF also supports a special user profile segment, CSDATA, for which the security administrator can specify the format and content of the data fields using other profiles in the CFIELD class, as well as specifying access rules in the FIELD class to determine which users can view or update data in the segment {SM.1::SM.1-R10-RACF-19}.

To create or modify a user profile, a user must have one of the following authorities:

- the SPECIAL role as a general system administrator {SM.1::SM.1.1}
- the UPDATE authority to the fields in a non-base segment of the profile he wants to modify through field-level access checking {SM.1::SM.1.2}
- to create a new user: is connected to a group that has the group-SPECIAL role and has the CLAUTH attribute for the USER class and is the owner of or has JOIN authority in the new user's default group. Note that the following roles of the ADDUSER command can not be assigned in this case: OPERATIONS, SPECIAL, and AUDITOR {SM.1::SM.1.3}
- to modify the attribute of a user: the CLAUTH attribute for the user class {SM.1::SM.1.4}. Note that only the CLAUTH and NOCLAUTH attribute can be changed {SM.1::SM.1.5}.

To list the contents of a user (user-2) profile using the LISTUSER command, a user (user-1) must have one of the following authorities:

- The SPECIAL role as a general system administrator, or the group-SPECIAL role as a group-administrator for user-2, the AUDITOR role, the group-AUDITOR role as a group auditor for user-2, or user-1 must own user-2 {SM.1::SM.1-R10-RACF-1}
- READ authority to the fields in a non-base segment of the profile he wants to list through field-level access checking {SM.1::SM.1-R10-RACF-2}
- When user-2 does not have the SPECIAL, OPERATIONS, or AUDITOR roles:
  - READ authority to FACILITY resource IRR.LISTUSER {SM.1::SM.1-R10-RACF-3}
  - READ authority to FACILITY resource IRR.LU.OWNER.owner-of-profile to allow use of LISTUSER for any non-excluded user-2 owned by "owner-of-profile" (which specifies a user ID or group name). {SM.1::SM.1-R10-RACF-4}

- READ authority to FACILITY resource IRR.LU.TREE.owner-of-tree to allow use of LISTUSER for any non-excluded user-2 who would be in the group-SPECIAL scope of “owner-of-tree” (which specifies a user ID or group name). That is, users owned by “owner-of-tree” or owned by groups owned by “owner-of-tree” {SM.1::SM.1-R10-RACF-21}
- To exclude a user-2 from being listed using IRR.LU.OWNER.owner-of-profile or IRR.LU.TREE.owner-of-tree authority, the administrator can define a profile that protects the resource IRR.LU.EXCLUDE.excluded-user-2 in the FACILITY class. With such a profile defined, a user also needs READ authority to it in order to gain authority via IRR.LU.OWNER.owner-of-profile or IRR.LU.TREE.owner-of-tree {SM.1::SM.1-R10-RACF-5}.

To reset the password for another user to an expired value using the PASSWORD or PHRASE commands:

- The SPECIAL role as a general system administrator, the group-SPECIAL role as a group-administrator for user-2 ,or user-1 must own user-2 {SM.1::SM.1-R10-RACF-6}.

To reset the password or password phrase for another user (user-2) or to resume user-2 using the ALTUSER command, a user (user-1) must have one of the following authorities:

- To specify a new expired or non-expired password/phrase, the SPECIAL role as a general system administrator {SM.1::SM.1-R10-RACF-7}
- To specify a new expired password/phrase, the group-SPECIAL role as a group-administrator for user-2 ,or user-1 must own user-2 {SM.1::SM.1-R10-RACF-8}
- When user-2 does not have the SPECIAL, OPERATIONS, or AUDITOR roles, or the PROTECTED attribute, one of:
  - READ authority to FACILITY resource IRR.PASSWORD.RESET to specify a new expired password/phrase when not within the minimum change window for user-2, or resume user-2 without specifying a resume date. User-1 can not set a phrase for a user-2 who does not have one already {SM.1::SM.1-R10-RACF-9}
  - UPDATE authority to FACILITY resource IRR.PASSWORD.RESET to specify a new non-expired password/phrase when not within the minimum change window for user-2, or resume user-2 without specifying a resume date. User-1 can not set a phrase for a user-2 who does not have one already {SM.1::SM.1-R10-RACF-10}.

    CONTROL authority allows the same as UPDATE, but also allows changing the password/phrase even when within the minimum change window for user-2 {SM.1::SM.1-R10-RACF-11}.
  - READ authority to FACILITY resource IRR.PWRESET.OWNER.owner-of-profile to specify a new expired password/phrase or resume a user without specifying a resume date, for any non-excluded user-2 owned by “owner-of-profile” (which specifies a user ID or group name) {SM.1::SM.1-R10-RACF-12}

    UPDATE authority allows the same as READ, and also allows setting a non-expired password or password phrase {SM.1::SM.1-R10-RACF-13}.

    CONTROL authority allows the same as UPDATE, and also allows setting a new password/phrase even when within the minimum change window for user-2 {SM.1::SM.1-R10-RACF-14}.
  - READ authority to FACILITY resource IRR.PWRESET.TREE.owner-of-tree to specify a new expired password/phrase or resume a user without specifying a resume date, for any non-excluded user-2 who would be in the group-SPECIAL scope of “owner-of-tree” (which specifies a user ID or group name). That is, users owned by “owner-of-tree” or

owned by groups owned by “owner-of-tree” {SM.1::SM.1-R10-RACF-15}.

UPDATE authority allows the same as READ, and also allows setting a non-expired password or password phrase {SM.1::SM.1-R10-RACF-16}.

CONTROL authority allows the same as UPDATE, and also allows setting a new password/phrase even when within the minimum change window for user-2 {SM.1::SM.1-R10-RACF-17}.

- To exclude a user-2 from being altered using IRR.PWRESET.OWNER.owner-of-profile or IRR.PWRESET.TREE.owner-of-tree authority, the administrator can define a profile that protects the resource IRR.PWRESET.EXCLUDE.excluded-user-2 in the FACILITY class. With such a profile defined, a user also needs READ authority to it in order to gain authority via IRR.PWRESET.OWNER.owner-of-profile or IRR.PWRESET.TREE.owner-of-tree {SM.1::SM.1-R10-RACF-18}.

RACF allows groups of users to be defined, making the management of users and user attributes and roles easier. To create a new group, a group profile must be defined in RACF. A group profile (as a user profile) consists of a base segment and (optional) other segments. As with the user profiles all group attributes related to the Security Policy as defined in this Security Target are contained in the base segment and the OMVS segment of the group profile. Each group defined in RACF must be owned by a RACF-defined user or by its superior group. Ownership of a group is assigned with the ADDGROUP command when a new group profile is created and can be changed with the ALTGROUP command used to change an existing group profile {SM.1::SM.1.6}.

RACF also supports a special group profile segment, CSDATA, for which the security administrator can specify the format and content of the data fields using other profiles in the CFIELD class, as well as specifying access rules in the FIELD class to determine which users can view or update data in the segment {SM.1::SM.1-R10-RACF-20}.

The owner of a group or a user connected to a group that has the group-SPECIAL role can:

- Define new users to RACF (provided he also has the CLAUTH attribute for the USER class) {SM.1::SM.1.7}.
- Connect and remove users from the group {SM.1::SM.1.8}.
- Delegate and change group authorities and set the default UACC for all new resources belonging to members of the group {SM.1::SM.1.9}.
- Modify, list, and delete the group profile {SM.1::SM.1.10}.
- Define, delete, and list the names of the subgroups under the group {SM.1::SM.1.11}.
- Specify the group terminal option {SM.1::SM.1.12}.

Users can be connected to a number of groups and have the group-related authorities of all the groups they are connected to {SM.1::SM.1.13}.

The OMVS segment of a group profile contains the group’s z/OS UNIX group identifier.

Management of z/OS user and group profiles occurs primarily via the RACF commands described later (ADDUSER, ALTUSER, DELUSER, LISTUSER, ADDGROUP, ALTGROUP, DELGROUP, LISTGRP). Administrators enter these commands while running in a TSO session.

8.4.1.2 User profiles

The base segment of a user profile within RACF contains (among other data not relevant for the security functions defined in this Security Target) the following:

Name	Description
USERID	User’s identification (a maximum of 8 characters).

NAME	User's name (not security relevant, because the user is allowed to change his name).
OWNER	Owner of the user's profile.
DFLTGRP	User's default group. (Note: A user may specify, at login time, any group he or she is connected to as the current default group. This does not change the DFLTGRP value in the profile.)
AUTHORITY	User's authority in the default group (use, create, connect, join).
PASSWORD	User's password. The user ID is DES-encrypted using the password (padded with blanks) as a key. Users who have no password and no password phrase are said to have the PROTECTED attribute, and can not logon to the system via any mechanism that uses a password, password phrase, or PassTicket.
PHRASE	Optional password phrase. Users who have a phrase must also have a password.
REVOKE	This attribute consists of a flag and a date. The date parameter specifies the date on which the user is revoked. The flag indicates that the user is revoked. The user is revoked, if either the flag is set or the actual date is after the revoke date, if defined.
RESUME	Date on which RACF lets the user have access to the system again.
UACC	Default universal access authority for resource profiles that the user defines. Only applicable to DATASET and a few general resource classes).
WHEN	Days of the week and hours of the day during which the user has access to the system (applies only to login through a terminal, not to other ports-of-entry).
CLAUTH	Classes in which the user can define profiles.
SPECIAL	Gives the user the system-wide SPECIAL attribute.
AUDITOR	Gives the user the system-wide AUDITOR attribute.
OPERATIONS	Gives the user the system-wide OPERATIONS attribute.
MODEL	Name of the data set model profile to be used when creating new data set profiles, either generic or discrete.
SECLABEL	User's default security label (evaluated in Labeled Security Mode only).
CERTNAME	The names of the profiles in the DIGTCERT (digital certificate) class that are related this RACF user ID.
CERTLABL	The certificate labels associated with the profiles in the DIGTCERT class that are related to this RACF user ID.

Table 28: User Profile

The OMVS segment in a user profile contains the following fields (among other information not relevant for the security policy as defined in this Security Target:

HOME User's z/OS UNIX initial directory path name

PROGRAM User's z/OS UNIX program path name, such as a default shell program

UID User's z/OS UNIX user identifier

Administrators have several choices when establishing OMVS information for users:

1. They may define the OMVS segment for users completely manually, via ADDUSER or ALTUSER with the OMVS keyword, and explicit specifications for the value of HOME, PROGRAM, and UID {SM.1::SM.1-R11-RACF-1}.
2. They may define the OMVS segment via ADDUSER or ALTUSER with the OMVS keyword and explicit specifications for HOME and PROGRAM, but allowing RACF to automatically choose the UID via the AUTOUID keyword , in conjunction with the BPX.NEXT.USER profile in the FACILITY class, where the administrator specifies an APPLDATA field containing the allowable range of automatically-assigned UIDs. RACF will then assign the lowest available unique UID and update the APPLDATA information to indicate the UID it used {SM.1::SM.1-R11-RACF-2}.
3. They may define the OMVS information implicitly, through use of the BPX.DEFAULT.USER profile in the FACILITY class. With the profile, the APPLDATA specifies the RACF user ID of a user who has an OMVS segment, and when a user without an OMVS segment needs to run a UNIX process, the system will temporarily use the HOME, PROGRAM, and UID information from the user named in BPX.DEFAULT.USER {SM.1::SM.1-R11-RACF-3}.
4. They may define the OMVS information automatically, by specifying the BPX.NEXT.USER profile in the FACILITY class to record the allowable range of automatically-assigned UIDs (as above), and the BPX.UNIQUE.USER profile to indicate that whenever a user without an OMVS segment makes use of UNIX functions, RACF should automatically create a permanent OMVS segment for the user, with a unique UID and with HOME and PROGRAM information derived from the user named in BPX.DEFAULT.USER {SM.1::SM.1-R11-RACF-4}. This process will also occur if someone inquires about the UID for a user who does not have one using the getumap() callable service. {SM.1::SM.1-R11-RACF-10}

The KERB segment in a user profile contains the following fields :

ENCRYPT Encryption methods allowable for this user : DES, DES3 (Triple DES), DES with key derivation, AES128, or AES256. For this evaluation only DES3, AES128, or AES256 is allowable.

KERBNAME The Keberos principal ID for a locally-defined Kerberos user.

MAXTKTLFE The maximum lifetime of a Kerberos ticket for this user.

8.4.1.1 Digital Certificates, Key Rings, and Certificate Mappings in RACF and PKCS#11 Cryptographic Tokens

RACF provides the RACDCERT command which can be used to

1. create certificate requests to send to a Certifying Authority {SM.1::SM.1-R8-RACF-RACDCERT-1}
2. generate public/private key pairs and certificates (DIGTCERT class) {SM.1::SM.1-R8-RACF-RACDCERT-2}
3. export a certificate or certificate packages to a data set, optionally with the private key {SM.1::SM.1-R8-RACF-RACDCERT-3}
4. install certificates into the RACF database and register them as belonging to a user or to a certifying authority {SM.1::SM.1-R8-RACF-RACDCERT-4}. The __certificate() and InitACEE() services can also register/deregister certificates {SM.1::SM.1-R8-RACF-RACDCERT-5}, and administrators an allow users to register their own certificates by granting them READ access to FACILITY resource IRR.DIGTCERT.ADD {SM.1::SM.1-R8-RACF-RACDCERT-6}.
5. delete or list certificates in the RACF database {SM.1::SM.1-R8-RACF-RACDCERT-7}
6. maintain (create, list, delete) key rings containing certificates (DIGTRING class) {SM.1::SM.1-

R8-RACF-RACDCERT-8}

7. add certificates to or delete them from key rings {SM.1::SM.1-R8-RACF-RACDCERT-9}
8. create mapping rules (certificate name filters) that can map client certificates that are not installed/registered in the database to specified user IDs based on subject or issuer information (DIGTNMAP class) {SM.1::SM.1-R8-RACF-RACDCERT-10}. This can allow a many-to-one mapping for applications that do not need to have each user run under his own ID. In this case, accountability can be maintained for auditing purposes by having the application provide the subject's distinguished name via the X500Name parameter when creating the security environment (ACEE) for the user {SM.1::SM.1-R8-RACF-RACDCERT-11}. The mapping process can also make use of mapping criteria specified by the DIGTCRIT class when it is necessary to map a client certificate into different IDs depending on characteristics of the user's session (such as the application name, or system name where the application is running) {SM.1::SM.1-R8-RACF-RACDCERT-12}.
9. create and manage the contents of PKCS#11 cryptographic tokens contained in the ICSF TKDS {SM.1::SM-1.R9-RACF-RACDCERT-13}.

{SM.1:: SM-1.R12-RACF-RACDCERT-14} RACDCERT supports installing or generating certificates that have the following key characteristics, subject to US export regulations and the available cryptographic hardware present on the system:

- RSA keys up to 4096 bits;
- DSA keys up to 2048 bits;
- NIST ECC keys up to 521 bits;
- Brainpool ECC keys up to 512 bits.

{SM.1::SM-1.R13-RACF-RACDCERT-15} RSA keys can be generated through

- RACF software and stored in the RACF DB (clear key)
- RACF software and stored in the ICSF PKDS (clear key)
- Cryptographic cards and stored in the ICSF PKDS (secure key)

{SM1.::SM-1.R13-RACF-RACDCERT-16} NIST/Brainpool keys can be generated through

- ICSF software and stored in the ICSF PKDS (clear key)
- CryptoExpress3 coprocessor cards (z114 or z196 processor) and stored in the ICSF PKDS (secure key)

The rest of this section describes processing in RACF.

Profiles in the DIGTCERT class contain information about digital certificates contained in the RACF database, as well as the certificate itself and optionally the certificate's private key. Additionally, the user's USER profile will have information about a certificate associated with the user.

Profiles in the DIGTRING class contain information about key rings and the certificates contained in a key ring. Each key ring is a named collection of the personal, site, and CA certificates associated with a user. When the user represents a server, the key ring has the allowable CA certificates that must be used to sign certificates presented by clients of the server during SSL handshaking.

Profiles in the DIGTNMAP and DIGTCRIT classes contain profiles used during certificate name filtering, a process during client authentication that can derive a user ID to use for the session from a certificate that is not specifically registered in the RACF database.

Note that only the RACDCERT command may be used to administer profiles in the DIGTCERT, DIGTRING, and DIGTNMAP classes.

8.4.1.2 Management for RACF Digital Certificates, Key Rings, Certificate Mappings, and Criteria

Administrators can use the RACDCERT command to generate or delete digital certificates, generate certificate requests, maintain key rings, and maintain certificate mappings. RACF maintains certificates in the DIGTCERT class, key rings in the DIGTRING class, and certificate mappings in the DIGTNMAP class.

Additionally RACF provides programming interfaces to allow applications to maintain RACF key rings.

Management for RACF digital certificates, key rings, certificate mappings, and certificate mapping criteria occurs during processing of the RACDCERT command or the use of the associated programming interfaces as described above. It also occurs during SSL/TLS processing, Communications Server Network Security Server processing, or other processing using the R_datalib programming interfaces to read or update RACF key ring information.

The authority to perform the individual management operations is determined by checking the user's access to specific RACF profiles. This access check processing generally follows the normal MVS DAC algorithm for general resources described above in the section on discretionary access control, using specific resource names in the FACILITY class that depend on the function requested. It also allows users with SPECIAL to perform certain of the functions, as explained below.

8.4.1.3 Authority checking for RACDCERT Processing

Note: Since the check for sufficient authority to perform one of the management functions of RACDCERT is performed by checking the user's authority to specific profiles using the standard RACF access check algorithm, the claims in this section start with "AC" instead of "SM".

In general to use RACDCERT users need either the SPECIAL attribute (AC.4-R9-RACF-1) or

- READ access to FACILITY resource IRR.DIGTCERT.*function* to issue RACDCERT commands for themselves {SM.7::AC.4-R9-RACF-2};
- UPDATE access to FACILITY resource IRR.DIGTCERT.*function* to issue RACDCERT commands for other users {SM.7::AC.4-R9-RACF-3};
- CONTROL access to FACILITY resource IRR.DIGTCERT.*function* to issue RACDCERT commands for SITE and CERTAUTH certificates {SM.7::AC.4-R9-RACF-4}.

Authority The following tables describe the basic functions and the authorities used for each RACDCERT function in more detail {SM.7::AC.4-R9-RACF-29}:

FUNCTION	READ	UPDATE	CONTROL
ADD	Add a certificate to one own's ID	Add a certificate to another user's ID	Add a site or certificate authority certificate
ADDRING	Create a key ring for one's own ID	Create a key ring for another user's ID	n/a
ADDTOKEN (controlled only via CRYPTOZ class)[1]	n/a	n/a	n/a
ALTER	Change the trust status or label of one's own certificate	Change the trust status or label of another user's certificate	Change the trust status or label of a site or certificate authority certificate

[1] See Authority Checking for PKCS#11 Cryptographic Tokens in the ICSF TKDS

ALTMAP	Alter a mapping associated with one's own ID	Alter a mapping associated with another user's ID or with MULTIID	n/a
BIND (Also see CRYPTOZ class)[4]	See BIND table	See BIND table	See BIND table
CHECKCERT (Note: uses LIST as the *function* in the DAC check)	Check one's own certificate	Check another user's certificate	Check a site or certificate authority certificate
CONNECT	See Connect tables	See Connect tables	See Connect tables
DELETE	Delete one's own certificate	Delete another user's certificate	Delete a site or certificate authority certificate
DELMAP	Delete a mapping associated with one's own ID	Delete a mapping associated with another user's ID or with MULTIID	n/a
DELRING	Delete one's own key ring	Delete another user's key ring	n/a
DELTOKEN (controlled only via CRYPTOZ class)[4]	n/a	n/a	n/a
EXPORT	See Export table	See Export table	See Export table
GENCERT	See Gencert table	See Gencert table	See Gencert table
GENREQ	Generate a request based on one's own certificate	Generate a request based on another user's certificate	Generate a request based on a site or certificate authority certificate
IMPORT (also see CRYPTOZ class)[4]	See ADD above.	See ADD above.	See ADD above.
LIST	List one's own certificate	List another user's certificate	List a site or certificate authority certificate
LISTMAP	List mapping information associated with one's own ID	List mapping information associated with another user's ID or MULTIID	n/a
LISTTOKEN (also see CRYPTOZ class)[4]	See LIST above	See LIST above	See LIST above
MAP	Create a mapping associated with one's own ID	Create a mapping associated with another user's ID or MULTIID	n/a
REMOVE	Remove a certificate from one's own key	Remove a site or certificate authority	Remove a certificate from another user's

	ring	certificate from one's own key ring	key ring
REKEY	Rekey one's own certificate	Rekey another user's certificate	Rekey a site or certificate authority certificate
ROLLOVER	Rollover one's own certificate	Rollover another user's certificate	Rollover a site or certificate authority certificate
UNBIND (controlled only via CRYPTOZ class) [4]	n/a	n/a	n/a

Table 29: RACDCERT Sunfunctions

This table describes the authorities needed to perform the BIND function to bind a certificate to a PKCS#11 token:

USAGE	One's own certificate	Another user's certificate	A site or certificate authority certificate
PERSONAL	READ authority to IRR.DIGTCERT.BIND	UPDATE authority to IRR.DIGTCERT.BIND	CONTROL authority to IRR.DIGTCERT.BIND
SITE CERTAUTH	CONTROL authority to IRR.DIGTCERT.ADD and READ authority to IRR.DIGTCERT.BIND	CONTROL authority to IRR.DIGTCERT.ADD and UPDATE authority to IRR.DIGTCERT.BIND	UPDATE authority to IRR.DIGTCERT.BIND

Table 30: RACDCERT BIND authorizations

This table describes the authorities needed to perform the CONNECT function to connect a certificate to one's own key ring:

USAGE	One's own certificate	Another user's certificate	A site or certificate authority certificate
PERSONAL	READ authority to IRR.DIGTCERT.CONNECT	UPDATE authority to IRR.DIGTCERT.CONNECT	CONTROL authority to IRR.DIGTCERT.CONNECT
SITE CERTAUTH	CONTROL authority to IRR.DIGTCERT.ADD and READ authority to IRR.DIGTCERT.CONNECT	CONTROL authority to IRR.DIGTCERT.ADD and UPDATE authority to IRR.DIGTCERT.CONNECT	UPDATE authority to IRR.DIGTCERT.CONNECT

Table 31: RACDCERT CONNECT authorizations for own key ring

This table describes the authorities needed to perform the CONNECT function to connect a certificate to another user's key ring:

USAGE	One's own certificate	Another user's certificate	A site or certificate authority certificate
PERSONAL	CONTROL authority to IRR.DIGTCERT.CONNECT	CONTROL authority to IRR.DIGTCERT.CONNECT	CONTROL authority to IRR.DIGTCERT.CONNECT

SITE CERTAUTH	CONTROL authority to IRR.DIGTCERT.ADD and CONTROL authority to IRR.DIGTCERT.CONNECT	CONTROL authority to IRR.DIGTCERT.ADD and CONTROL authority to IRR.DIGTCERT.CONNECT	CONTROL authority to IRR.DIGTCERT.CONNECT

Table 32: RACDCERT CONNECT authorizations for other key rings

This table describes the authorities needed to perform the EXPORT function:

Function	READ	UPDATE	CONTROL
EXPORT (in CERT format)	Export one's own certificate	Export another user's certificate	Export a site or certificate authority certificate
EXPORT (in PKCS#7 format)	Export one's own certificate but not the parent CA chain	Export another user's certificate but not the parent CA chain	Export site or certificate authority certificates or the entire parent CA chain for oneself or another user.
Function	**READ**	**UPDATE**	**CONTROL**
EXPORT (in PKCS#12 format. Note: uses EXPORTKEY as the *function* in the DAC check)	Export one's own certificate and the private key	Export another user's certificate and the private key	Export a site or certificate authority certificate and the private key

Table 33: RACDCERT EXPORT authorizations

This table describes the authorities needed to perform the GENCERT function:

SIGNWITH option chosen	To generate one's own certificate	To generate another user's certificate	To generate a site or certificate authority certificate
SIGNWITH one's own certificate	READ authority to IRR.DIGTCERT.ADD and READ authority to IRR.DIGTCERT.GENCERT	UPDATE authority to IRR.DIGTCERT.ADD and READ authority to IRR.DIGTCERT.GENCERT	CONTROL authority to IRR.DIGTCERT.ADD and READ authority to IRR.DIGTCERT.GENCERT
SIGNWITH a SITE or CERTAUTH certificate	READ authority to IRR.DIGTCERT.ADD and CONTROL authority to IRR.DIGTCERT.GENCERT	UPDATE authority to IRR.DIGTCERT.ADD and CONTROL authority to IRR.DIGTCERT.GENCERT	CONTROL authority to IRR.DIGTCERT.ADD and CONTROL authority to IRR.DIGTCERT.GENCERT
SIGNWITH not specified	READ authority to IRR.DIGTCERT.ADD and READ authority to IRR.DIGTCERT.GENCERT	UPDATE authority to IRR.DIGTCERT.ADD and UPDATE authority to IRR.DIGTCERT.GENCERT	CONTROL authority to IRR.DIGTCERT.ADD and CONTROL authority to IRR.DIGTCERT.GENCERT

Table 34: RACDCERT GENCERT authorizations

8.4.1.1 Authority Checking for R_datalib Processing

The R_datalib callable services provides access to some fields of certificates and key rings, including when appropriate the private keys when stored in RACF. R_datalib allows reading, creation, or modification of key rings As with RACDCERT functions, the SPECIAL attribute authorizes some functions. In addition, profiles in the RDATALIB class or in the FACILITY class can authorize various

R_datalib functions.

When using the FACILITY class, RACF will use resource names of the form IRR.DIGTCERT.*function* to authorize the processing, where the descriptions below will describe the applicable *function* values.

When using the RDATALIB class, RACF will use resource names of the form <***ringOwner***>.<***ringName***>.*function*, where the descriptions below will the describe the applicable *function* values.

The ***ringOwner*** must be in upper case. The ***ringName*** will be folded into upper cases during profile checking. Rings differ only in case will be using the same profile {SM.7::AC.4-R9-RACF-26}.

In the case the owner ID and the ring name are of their maximum limits, and you want to create a discrete profile, it can be done by truncating the ring name from the end so that the whole profile name length is 246 characters {SM.7::AC.4-R9-RACF-27}.

If the input Ring_name is of the virtual keyring form - a single '*', the ring name part in the resource will be IRR_VIRTUAL_KEYRING so that different profiles can be set up to control access on real and virtual keyrings {SM.7::AC.4-R9-RACF-28}.

If the caller of R_datalib provides an owner ID of *TOKEN*, then the request specifies use of a PKCS#11 cryptographic token in the ICSF TKDS, and all security checking occurs in ICSF using the CRYPTOZ class. R_datalib does not do any checking in the FACILITY or RDATALIB classes for these cases. For more information on this case see **Authority Checking** for PKCS#11 Cryptographic Tokens in the ICSF TKDS in the z/OS Security Target.

For the DatagetFirst, DataGetNext, and GetUpdateCode functions:

Using RDATALIB Checking for a Real Keyring {SM.7::AC.4-R9-RACF-5}:

Access to <ringOwner>.<ringName>.LST in the RDATALIB class, **Eg. SERVER1.FTPRING1.LST**	**Action able to perform**
READ	DataGetFirst, DataGetNext: list Server1's ring named FTPring1, and returns one's own private key if the usage is PERSONAL GetUpdateCode: return the sequence number of Server1's ring named FTPring1
UPDATE	DataGetFirst, DataGetNext: list Server1's ring named FTPring1, and returns other's private key if the usage is PERSONAL
CONTROL (or caller is RACF SPECIAL)	DataGetFirst, DataGetNext: list Server1's ring named FTPring1, and returns SITE/CA's private key if the usage is PERSONAL

Table 35: RDATALIB checking for real key ring

Using RDATALIB Checking for a Virtual Keyring {SM.7::AC.4-R9-RACF-6}:

Virtual keyring owner	**Resource Name**	**Access**	**Action able to perform**
Ordinary ID, eg. USER1	USER1.IRR_VIRTUAL_KEYRING.LST	READ	DataGetFirst, DataGetNext: list USER1's virtual keyring, and

			returns the private keys if the caller is USER1, ie. the owner of the virtual keyring GetUpdateCode: return the sequence number
		UPDATE	DataGetFirst, DataGetNext: list USER1's virtual keyring, and returns the private key GetUpdateCode: return the sequence number
CERTAUTH	CERTIFAUTH.IRR_VIRTUAL_KEYRING.LST	Read	DataGetFirst, DataGetNext: list CERTAUTH's virtual keyring GetUpdateCode: return the sequence number
SITE	SITECERTIF.IRR_VIRTUAL_KEYRING.LST	Read	DataGetFirst, DataGetNext: list SITE's virtual keyring GetUpdateCode: return the sequence number

Table 36: RDATALIB checking for virtual key ring

Using FACILITY Checking {SM.7::AC.4-R9-RACF-7}:

Access to IRR.DIGTCERT.LISTRING in the FACILITY class	Action able to perform
READ	DataGetFirst, DataGetNext: list one's own real or virtual ring, and returns one's own private key if the usage is PERSONAL list one's own real or virtual ring, and returns SITE/CA's private key if the usage is PERSONAL, if caller is SPECIAL or has CONTROL to IRR.DIGTCERT.GENCERT in the FACILITY class GetUpdateCode: return the sequence number of one's own real or virtual ring
UPDATE	DataGetFirst, DataGetNext: list other's real or virtual ring, and returns SITE/CA's private key if the usage is PERSONAL if caller is SPECIAL or has CONTROL to IRR.DIGTCERT.GENCERT in the FACILITY class GetUpdateCode:

	return the sequence number of other's real or virtual ring

Table 37: RDATALIB FACILITY class checking

For the CheckStatus function:

The call requires READ authority to resource IRR.DIGTCERT.LIST in the FACILITY class {SM.7::AC.4-R9-RACF-8}.

For the IncSerialNum function:

The call requires either the SPECIAL attribute {SM.7::AC.4-R9-RACF-9} or

- 35 17 READ authority to resource IRR.DIGTCERT.GENCERT in the FACILITY class if the caller owns the certificate {SM.7::AC.4-R9-RACF-10};
- 35 17 CONTROL authority to resource IRR.DIGTCERT.GENCERT in the FACILITY class for a site or certificate authority certificate {SM.7::AC.4-R9-RACF-11}.

For the NewRing function:

No checking will be performed if the caller has the RACF SPECIAL attribute {SM.7::AC.4-R9-RACF-12}.

Using RDATALIB Profile Checking: {SM.7::AC.4-R9-RACF-13}:

Access to <ringOwner>.<ringName>.UPD in the RDATALIB class, **Eg. SERVER1.FTPRING1.UPD**	**Action able to perform**
READ	35 17 add a new ring for Server1 named FTPring1 35 17 remove all certificates from the the existing ring named FTPring1 owned by Server1

Using FACILITY Profile Checking: {SM.7::AC.4-R9-RACF-14}:

Access to IRR.DIGTCERT.ADDRING in the FACILITY class	**Access to IRR.DIGTCERT.REMOVE in the FACILITY class**	**Action able to perform**
READ	n/a	create one's own new ring
UPDATE	n/a	create other's new ring
n/a	READ	remove certificates from one's ring
n/a	UPDATE	remove certificates from other's ring

For the DelRing Function:

No checking will be performed if the caller has the RACF SPECIAL attribute {SM.7::AC.4-R9-RACF-15}.

Using RDATALIB Profile Checking {SM.7::AC.4-R9-RACF-16}:

Access to <ringOwner>.<ringName>.UPD in the RDATALIB class,	**Action able to perform**

Eg. SERVER1.FTPRING1.UPD	
READ	delete a ring owned by Server1 named FTPring1

Using FACILITY Profile Checking {SM.7::AC.4-R9-RACF-17}:

Access to IRR.DIGTCERT.DELRING in the FACILITY class	**Action able to perform**
READ	delete one's own ring
UPDATE	delete other's ring

For the DataRemove Function:

No checking will be performed if the caller has the RACF SPECIAL attribute {SM.7::AC.4-R9-RACF-18}.

Using RDATALIB Profile Checking {SM.7::AC.4-R9-RACF-19}:

Access to <ringOwner>.<ringName>.UPD in the RDATALIB class, **Eg. SERVER1.FTPRING1.UPD**	**Action able to perform**
READ	remove one's own cert from Server1's ring named FTPring1
UPDATE	remove one's own or other's cert from Server1's ring named FTPring1
CONTROL	remove any type cert from Server1's ring named FTPring1

Using FACILITY Profile Checking {SM.7::AC.4-R9-RACF-20}:

Access to IRR.DIGTCERT.REMOVE in the FACILITY class	**Action able to perform**
READ	remove one's own cert from one's ring
UPDATE	remove any type cert from one's ring
CONTROL	remove any type cert from other's ring

In addition, if the DataRemove operation specifies CDDL_ATT_DEL_CERT_TOO, then RACF will also check, IRR.DIGTCERT.DELETE whether using RDATALIB or FACILITY profiles {SM.7::AC.4-R9-RACF-21}:

Access to IRR.DIGTCERT.DELETE in the FACILITY class	**Action able to perform**
READ	delete one's own cert from RACF if it is not connected to other rings
UPDATE	delete one's or other's cert from RACF if it is not connected to other rings
CONTROL	delete any type cert from RACF if it is not connected to other rings

For the DataPut Function:

No checking will be performed if the caller has the RACF SPECIAL attribute {SM.7::AC.4-R9-RACF-22}.

Note: In the following tables,

- 35 17 Any usage = PERSONAL, CERTAUTH or SITE
- 35 17 Any type cert = certificate is owned by any regular ID, or by the site or a certificate authority.

Using RDATALIB Profile Checiking {SM.7::AC.4-R9-RACF-23}:

With READ Access to <ringOwner>.<ringName>.UPD, eg. SERVER1.FTPRING1.UPD

<table>
<tr><td rowspan="2"></td><td rowspan="2">Input cert is not in RACF</td><td colspan="2">Input cert is already in RACF</td><td colspan="2">Input cert is in RACF and already connected to the ring</td></tr>
<tr><td>with no private key</td><td>with private key</td><td>With no private key</td><td>with private key</td></tr>
<tr><td>Input cert only</td><td rowspan="2">(a) add one’s own cert

(b) connect to Server1’s ring named FTPring1 one’s own cert with usage PERSONAL only</td><td colspan="2">if cert owned by caller
35 17 connect to Server1’s ring named FTPring1 with usage PERSONAL only, other usages cause error
35 17 change the NOTRUST status to TRUST if trust flag turns on
if cert is not owned by caller, error</td><td colspan="2">if cert owned by caller
35 17 re-connect to Server1’s ring named FTPring1 with usage PERSONAL only, other usages cause error, with new specified default value
35 17 change the NOTRUST status to TRUST if trust flag turns on
if cert is not owned by caller, error</td></tr>
<tr><td>Input cert and private key</td><td>if cert owned by caller
35 17 re-add cert with private key
35 17 connect to Server1’s ring named FTPring1 with usage PERSONAL only, other usages cause error
35 17 change the NOTRUST status to TRUST if trust flag turns on
if cert is not owned by caller,</td><td>if cert owned by caller
35 17 connect to Server1’s ring named FTPring1 with usage PERSONAL only, other usages cause error
35 17 change the NOTRUST status to TRUST if trust flag turns on
if cert is not owned by caller, error</td><td>if cert owned by caller
35 17 re-add cert with private key
35 17 re-connect to Server1’s ring named FTPring1 with usage PERSONAL only, other usages cause error, with new specified default value
35 17 change the NOTRUST status to TRUST if trust flag</td><td>if cert owned by caller
35 17 re-connect to Server1’s ring named FTPring1 with usage PERSONAL only, other usages cause error, with new specified default value
35 17 change the NOTRUST status to TRUST if trust flag turns on
if cert is not owned by caller, error</td></tr>
</table>

		error		turns on if cert is not owned by caller, error	

With UPDATE Access to <ringOwner>.<ringName>.UPD, eg. SERVER1.FTPRING1.UPD

	Input cert is not in RACF	Input cert is already in RACF		Input cert is in RACF and already connected to the ring	
		with no private key	with private key	with no private key	with private key
Input cert only	35 17 add any type cert 35 17 connect to Server1's ring named FTPring1 one's own cert with any usage or 35 17 connect other's or SITE/CA's cert with usage SITE or CERTAUTH only, PERSONAL usage causes error	35 17 connect to Server1's ring named FTPring1 one's own cert with any usage or 35 17 connect to Server1's ring named FTPring1 other's or SITE/CA's cert with usage SITE or CERTAUTH only, PERSONAL usage causes error 35 17 change the NOTRUST status to TRUST if trust flag turns on		35 17 re-connect to Server1's ring named FTPring1 one's own cert with any usage or 35 17 re-connect to Server1's ring named FTPring1 other's or SITE/CA's cert with usage SITE or CERTAUTH only, PERSONAL usage causes error, with new specified default value 35 17 change the NOTRUST status to TRUST if trust flag turns on	
Input cert and private key	35 17 add any type cert 35 17 connect to Server1's ring named FTPring1 any type cert with any usage	35 17 re-add any type cert with private key under original ID 35 17 connect to Server1's ring named FTPring1 any type cert with any usage 35 17 change the	35 17 connect to Server1's ring named FTPring1 any type cert with any usage 35 17 change the NOTRUST status to TRUST if trust flag turns on	35 17 re-add any type cert with private key under original ID 35 17 re-connect to Server1's ring named FTPring1 any type cert with any usage, with new specified default value	35 17 re-connect to Server1's ring named FTPring1 any type cert with any usage, with new specified default value 35 17 change the NOTRUST status to TRUST if trust

		NOTRUST status to TRUST if trust flag turns on		35 17 change the NOTRUST status to TRUST if trust flag turns on	flag turns on

With CONTROL Access to <ringOwner>.<ringName>.UPD, eg. SERVER1.FTPRING1.UPD

	Input cert is not in RACF	Input cert is already in RACF		Input cert is in RACF and already connected to the ring	
		with no private key	with private key	with no private key	with private key
Input cert only	35 17 add any type cert 35 17 connect to Server1's ring named FTPring1 any type cert with any usage	35 17 connect to Server1's ring named FTPring1 any type cert with any usage 35 17 change the NOTRUST status to TRUST if trust flag turns on		35 17 re-connect to Server1's ring named FTPring1 any type cert with any usage, with new specified default value 35 17 change the NOTRUST status to TRUST if trust flag turns on	
Input cert and private key		35 17 re-add cert with private key under original ID 35 17 connect to Server1's ring named FTPring1 any type cert with any usage 35 17 change the NOTRUST status to TRUST if trust flag turns on	35 17 connect to Server1's ring named FTPring1 any type cert with any usage 35 17 change the NOTRUST status to TRUST if trust flag turns on	35 17 re-add cert with private key under original ID 35 17 re-connect to Server1's ring named FTPring1 any type cert with any usage, with new specified default value 35 17 change the NOTRUST status to TRUST if trust flag turns on	35 17 re-connect to Server1's ring named FTPring1 any type cert with any usage, with new specified default value 35 17 change the NOTRUST status to TRUST if trust flag turns on

Using FACILITY Profile Checking {SM.7::AC.4-R9-RACF-24}:

Certificate does not exist in RACF Database

Access to IRR.DIGTCERT.ADD in the FACILITY class	**Access to IRR.DIGTCERT.CONNECT in the FACILITY class**	**Action able to perform**

READ	READ	35 17 add one's own cert 35 17 connect one's own cert with usage PERSONAL to one's own ring
CONTROL	READ	35 17 add one's own cert 35 17 connect one's own cert with any usage to one's own ring
UPDATE	UPDATE	35 17 add one's own or other's cert 35 17 connect one's own or other's cert with usage PERSONAL to one's ring or 35 17 connect SITE/CA's cert with SITE/CERTAUTH usage to one's own ring
CONTROL	UPDATE	35 17 add any type cert 35 17 connect one's own or other's cert with usage PERSONAL to one's ring or 35 17 connect any type cert with usage SITE/CERTAUTH to one's ring
UPDATE	CONTROL	35 17 add one's own or other's cert 35 17 connect any type cert with usage PERSONAL to any ring or 35 17 connect SITE/CA's cert with any usage to any ring
CONTROL	CONTROL	35 17 add any type cert 35 17 connect any type cert with any usage to any ring

Certificate exists in RACF Database with no private key but private key is specified

Access to IRR.DIGTCERT.ADD in the FACILITY class	**Access to IRR.DIGTCERT.CONNECT in the FACILITY class**	**Action able to perform**
READ	READ	35 17 re-add one's own cert with private key 35 17 change the NOTRUST status of the connected cert to TRUST if trust flag turns on 35 17 connect one's own cert with usage PERSONAL to one's own ring
CONTROL	READ	35 17 re-add one's own cert with private key 35 17 change the NOTRUST status of the connected cert to TRUST if trust flag turns

		on 35 17 connect one's own cert with any usage to one's own ring
UPDATE	UPDATE	35 17 re-add one's own or other's cert with private key 35 17 change the NOTRUST status of the connected cert to TRUST if trust flag turns on 35 17 connect one's own or other's cert with usage PERSONAL to one's ring or 35 17 connect SITE/CA's cert with SITE/CERTAUTH usage to one's own ring
CONTROL	UPDATE	35 17 re-add any type cert with private key 35 17 change the NOTRUST status of the connected cert to TRUST/HIGHTRUST if trust flag turns on 35 17 connect one's own or other's cert with usage PERSONAL to one's ring or 35 17 connect any type cert with usage SITE/CERTAUTH to one's ring
UPDATE	CONTROL	35 17 re-add one's own or other's cert with private key 35 17 change the NOTRUST status of the connected cert to TRUST if trust flag turns on 35 17 connect any type cert with usage PERSONAL to any ring or 35 17 connect SITE/CA's cert with any usage to any ring
CONTROL	CONTROL	35 17 re-add any type cert with private key 35 17 change the NOTRUST status of the connected cert to TRUST/HIGHTRUST if trust flag turns on 35 17 connect any type cert with any usage to any ring

Certificate already exists in RACF Database and no private key is input

Access to IRR.DIGTCERT.ADD in the FACILITY class	**Access to IRR.DIGTCERT.CONNECT in the FACILITY class**	**Access to IRR.DIGTCERT.ALTER in the FACILITY class (will be checked if changing status from**	**Action able to perform**

		NOTRUST to TRUST/HIGHTRUST is requested)	
n/a	READ	READ	35 17 connect one's own cert with usage PERSONAL to one's own ring 35 17 change the NOTRUST status of the connected cert to TRUST if trust flag turns on
CONTROL	READ	READ	35 17 connect one's own cert with any usage to one's own ring 35 17 change the NOTRUST status of the connected cert to TRUST if trust flag turns on
n/a	UPATE	READ – one's own cert UPDATE – other's cert CONTROL – SITE/CA's cert	35 17 connect one's own or other's cert with usage PERSONAL to one's ring or 35 17 connect SITE/CA's cert with SITE/CERTAUTH usage to one's own ring 35 17 change the NOTRUST status of the connected cert to TRUST/HIGHTRUST if trust flag turns on
CONTROL	UPDATE	READ – one's own cert UPDATE – other's cert CONTROL – SITE/CA's cert	35 17 connect one's own or other's cert with usage PERSONAL to one's own ring or 35 17 connect SITE/CA's cert with SITE/CERTAUTH

			usage to one's own ring 35 17 change the NOTRUST status of the connected cert to TRUST/HIGHTRUST if trust flag turns on
n/a	CONTROL	READ – one's own cert UPDATE – other's cert CONTROL – SITE/CA's cert	35 17 connect any type cert with usage PERSONAL to any ring or 35 17 connect SITE/CA's cert with any usage to any ring 35 17 change the NOTRUST status of the connected cert to TRUST/HIGHTRUST if trust flag turns on
CONTROL	CONTROL	READ – one's own cert UPDATE – other's cert CONTROL – SITE/CA's cert	35 17 connect any type cert with any usage to any ring 35 17 change the NOTRUST status of the connected cert to TRUST/HIGHTRUST if trust flag turns on

For the DataRefresh Function:

No checking will be performed if the caller has the RACF SPECIAL attribute, otherwise if the DIGTCERT class is SETR RACLISTed then the caller needs class authority (CLAUTH) to the DIGTCERT class {SM.7::AC.4-R9-RACF-25}.

8.4.1.1 Group profiles

The base segment of a group profile within RACF contains (among other data not relevant for the security functions defined in this Security Target) the following:

Name	Description
GROUPNAME	Name of the group
OWNER	Owner of the group profile
SUPGROUP	The profile's superior group
MODEL	Name of a profile to be used as a model

TERMUACC or NOTERMUACC	The group's terminal authorization

The OMVS segment of the group profile contains the group's z/OS UNIX group identifier in the GID field.

Administrators have several choices when establishing OMVS information for groups:

1. They may define the OMVS segment for groups completely manually, via ADDGROUP or ALTGROUP with the OMVS keyword, and explicit specification of the value of the GID {SM.1::SM.1-R11-RACF-5}.
2. They may define the OMVS segment via ADDGROUP or ALTGROUP with the OMVS keyword but allowing RACF to automatically choose the GID via the AUTOGID keyword , in conjunction with the BPX.NEXT.USER profile in the FACILITY class, where the administrator specifies an APPLDATA field containing the allowable range of automatically-assigned GIDs. RACF will then assign the lowest available unique GID and update the APPLDATA information to indicate the GID it used {SM.1::SM.1-R11-RACF-6}.
3. They may define the OMVS information implicitly, through use of the BPX.DEFAULT.USER profile in the FACILITY class. With the profile, the APPLDATA specifies the RACF group name of a group that has an OMVS segment, when the system needs to determine the GID of a user's default group, and the group does not have an OMVS segment, the system will temporarily use the GID information from the group named in BPX.DEFAULT.USER {SM.1::SM.1-R11-RACF-7}.
4. They may define the OMVS information automatically, by specifying the BPX.NEXT.USER profile in the FACILITY class to record the allowable range of automatically-assigned GIDs (as above), and the BPX.UNIQUE.USER profile to indicate that whenever a user whose default (or current connect) group has no OMVS segment makes use of UNIX functions, RACF should automatically create a permanent OMVS segment for that group. {SM.1::SM.1-R11-RACF-8}. This process will also occur if someone inquires about the GID for a group that does not have one using the getgmap() callable service {SM.1::SM.1-R11-RACF-9}.

8.4.1.2 User roles and attributes

User roles and attributes are extraordinary capabilities, restrictions, or environments that can be assigned to a user, either all of the time or when the user is connected to a specific group or groups. User attributes are stored and managed within the RACF database.

When a role or attribute is to apply only to a specific group or groups, it is specified at the group level and is called a group-related user attribute. For example, user attributes that are specified in an ADDUSER or ALTUSER command are stored in the user's profile and are in effect regardless of the group to which the user is connected {SM.1::SM.1.14}.

RACF maintains the roles and attributes specified in this section in fields in the user profile. The distinction between roles and attributes in this Security Target is artificial and reflects the definition in Chapter 5 for roles and user attributed. RACF does not make this distinction and the IBM guidance describes all of the following as user attributes.

Apart from the explicitly mentioned roles and attributes described below, users are assigned certain roles implicitly:

- Users implicitly are in the "user" role which allows them to change their own authentication data
- Users can be assigned the operator role by authorizing them to issue an operator command in the command's own profile.
- Ownership of objects entitles users to change the object's security attributes. Ownership for

non-UNIX objects is identical to ownership of the profile protecting the object.

8.4.1.2.1 RACF Roles

SPECIAL and group-SPECIAL

A user who has the SPECIAL attribute at the system level can issue all RACF commands (but not all operands. There are AUDITOR-only operands related to the configuration of the audit function that only a user with the AUDITOR attribute is allowed to use) {SM.1::SM.1.15}. The SPECIAL attribute gives the user full control over all of the RACF profiles in the RACF database. The SPECIAL attribute can also be assigned at the group level. Such a user with the group-SPECIAL attribute has full control over all of the profiles within the scope of the group.

A user with the SPECIAL role in his user profile is regarded as a system administrator. He can:

- add, delete, list and modify user, group, DATASET and other profiles {SM.1::SM.1.16}
- list and define RACF general options (except options related to auditing) {SM.1::SM.1.17}

A system administrator can delegate administrative activities to users such that they can administer profiles belonging to a defined group. He does this by assigning such users the group-SPECIAL attribute. Those users then have administrative capabilities within the group they were assigned the group SPECIAL attribute {SM.1::SM.1.18}. Users with the attribute group-SPECIAL can not use general RACF options of the SETROPTS command (except for the REFRESH GENERIC and LIST operands) {SM.1::SM.1.19}.

AUDITOR and group-AUDITOR

The AUDITOR attribute is given only to users who are responsible for auditing RACF security controls and functions. To provide a check and balance on RACF security measures, the AUDITOR attribute should be given to security or group administrators other than those who have the SPECIAL attribute. The AUDITOR attribute can also be assigned at the group level. Such a user with the group-AUDITOR attribute can control the audit configuration within the scope of the group where the attribute was assigned {SM.1::SM.1.20}.

A user with the AUDITOR attribute can define and modify the audit related options in user and the auditor related options for resource profiles {SM.1::SM.1.21}. This allows him to define which activities are to be recorded in the audit trail. He can also list the content of any profile and set the system wide audit related options using the SETROPTS command. Those options are:

- AUDIT or NOAUDIT (for each profile class) {SM.1::SM.1.22}
- CMDVIOL or NOCMDVIOL {SM.1::SM.1.23}
- LOGOPTIONS (for each profile class) {SM.1::SM.1.24}
- OPERAUDIT or NOOPERAUDIT {SM.1::SM.1.25}
- SAUDIT or NOSAUDIT {SM.1::SM.1.26}
- SECLABELAUDIT or NOSECLABELAUDIT {SM.1::SM.1.27}

Audit configuration can also be delegated at the group level by giving the group-AUDITOR attribute to a user.

A user with the group-Auditor attribute can define and modify the audit related options in user, and resource profiles associated with his group {SM.1::SM.1.28}. He can not modify or set audit related attributes that operate system-wide {SM.1::SM.1.29}. Note that a user with SPECIAL controls the activation/deactivation of the OMVS audit related classes (DIRACC, DIRSRCH, FSOBJ, FSSEC, IPOBJ, PROCACT and PROCESS)

OPERATIONS and group-OPERATIONS

A user who has the OPERATIONS attribute has full access authorization to all RACF-protected

resources in the DATASET, DASDVOL, GDASDVOL and TAPEVOL classes except when restricted by an access list entry granting less authority {SM.1::SM.1.30}. The OPERATIONS attribute can also be assigned at the group level {SM.1::SM.1.31}.

z/OS UNIX superuser

A user operating with an effective UID of zero or a user that has been authorized to the BPX.SUPERUSER profile in the FACILITY class is defined to have the role of a z/OS UNIX superuser.

Pseudo user

A user defined with the NOPASSWORD, NOPHRASE, and NOOIDCARD parameter in his user profile is defined as having the role of a "pseudo-user". The TOE prohibits that a user with those attributes can log into the TOE. Those IDs can be used by SURROGAT-submitted batch jobs or by started procedures defined in the STARTED class or the started procedures table.

8.4.1.2.2 RACF Attributes

CLAUTH

If a user has the CLAUTH attribute in a class, RACF allows the user to define profiles in that class {SM.1::SM.1.32}.

Users receive the CLAUTH attribute on a class-by-class basis. {SM.1::SM.1.33}.

A user with the CLAUTH(USER) attribute can add and modify users except for setting or modifying the following attributes:

- SPECIAL or NOSPECIAL {SM.1::SM.1.34}
- AUDITOR or NOAUDITOR {SM.1::SM.1.35}
- OPERATIONS or NOOPERATIONS {SM.1::SM.1.36}

REVOKE

A user can be prevented from entering the system by assigning the REVOKE attribute {SM.1::SM.1.37}. This attribute is useful when a user needs to be prevented from entering the system, but cannot be deleted using the DELUSER command because the user still owns RACF resource profiles. It is also useful when a user must be temporarily prevented from using the system for some reason.

User accounts can be revoked automatically after a period of inactivity {SM.1::SM.1.38}. This applies also to accounts that have never been active {SM.1::SM.1.39}.

8.4.1.3 User Revocation

User revocation can take two forms in the TOE:

1. Revocation of the RACF user ID associated with a user: As all user authentication occurs via RACF, and all users have a RACF identity, the administrator can revoke a user by using the ALTUSER command with the REVOKE operand {SM.1::SM.1-R8-REV-1}. Note that this will not cover immediate revocation, but it will prevent the user from entering the system in the future.

2. Revocation of a user's digital certificate: For certificates registered in RACF via the RACDCERT command, the administrator can delete the certificate using RACDCERT {SM.1::SM.1-R8-REV-2}. This will prevent the system from recognizing that certificate in the future and associating it with the user's RACF identity.

For immediate revocation of a user in extreme situations a simple ALTUSER or certificate revocation may not suffice. In that case the administrator may determine which applications the user has access to (e.g., TSO/E, z/OS UNIX System Services, FTP server, HTTP server, LDAP). The administrator can then issue appropriate system or application commands to determine if the user is active in the system, and if so issue the appropriate system or application commands to terminate the user's sessions.

For example, for a TSO/E user the administrator could issue the CANCEL U=user-ID command. For a batch job the administer could issue CANCEL jobname.

As a final resort the administrator could stop servers such as the HTTP server, FTP server, or LDAP server if the administrator is not sure how to locate the user's sessions on the system, as well as stopping all UNIX processing, TSO/E processing, and batch processing.

8.4.2 Resource management

RACF makes access decisions based on information stored in profiles or in the metadata associated with z/OS UNIX objects. RACF manages the following resource profiles:

- Data set profiles
- General resource profiles

General resource profiles apply to a number of resources defined as protected resources in this Security Target. The structure of the profiles in RACF used to protect those resources is identical, but the semantics of specific access rights is defined by the manager of the resource and may therefore differ depending on the type of resource.

Profiles consists of a base segment and optionally a set of non-base segments. Fields within non-base segments can be individually protected using the field-level access control possibilities provided by RACF.

Field-level access control allows the control of READ and UPDATE access to individual fields within a segment other than the base segment of a RACF profile. This access is based on RACF profiles in the FIELD class. Profiles in this class have the structure of *profile-type.segment-name.field-name* where *profile-type* is either the class name of a general resource profile or one of DATASET, GROUP, or USER. Different profile classes/types can have different segments and the name of the segment that contains the field for which access is controlled is specified as the second part of the profile. Different segments have different fields identified by their name and the name of the field is the third part of the profile controlling access to the field. The access control algorithm for access to general resources is used also for the FIELD class.

Fields in segments are related to operands of RACF commands or the R_admin callable service used to manage profiles and the purpose of field level access control is to provide a mechanism that allows the definition of fine-grained access control to use those command operands or list the content of individual fields. Table 18 in [RACF.SAG] provides a complete list of the segment names and the fields within each of those segments that are subject to field-level access control. The table also maps the field names to the command operands that are used to update the fields. Note that use of those operands requires to have at least UPDATE authority to the field through field-level access control unless the user has the SPECIAL attribute {SM.4::SM.4-R12-RACFEAL5-FLA-1}. When profiles are listed, the user will only get information about the content of fields in segments protected by field-level access control, if he has at least READ access to the field, unless the user has the SPECIAL or AUDITOR attribute {SM.4::SM.4-R12-RACFEAL5-FLA-2}.

In order to use field-level access control, the FIELD class needs to be active and SETROPTS RACLIST needs to be activated for the FIELD class. Otherwise only a user with the SPECIAL or AUDITOR attribute has access to fields {SM.4::SM.4-R12-RACFEAL5-FLA-3}.

When the FIELD class is active and SETROPTS RACLIST is activated for the FIELD class, users with the SPECIAL or AUDITOR attribute can list all fields regardless of the access definition in the

profile protecting access to the field {SM.4::R12-RACFEAL5-FLA-4}.

To allow users to read or update fields in their own user profile protected by field-level access control a userid of &RACUID can be specified in the PERMIT command for the profiles in the FIELD class related to the fields in profiles of type USER {SM.4::R12-RACFEAL5-FLA-5}. This does not allow this user access to those fields in the user profiles of other users {SM.4::R12-RACFEAL5-FLA-6}.

For information on z/OS UNIX objects see z/OS UNIX file system resources.

8.4.2.1 Data set profiles

A data set profile within RACF contains (among other data not relevant for the security functions defined in this Security Target) the following:

Name	Description
Profile name	Name of the data set profile
GENERIC, MODEL, or TAPE	Indicates if it is a generic, a model or a tape data set profile
OWNER	Owner of the data set profile
NOTIFY	The TSO user who is to be notified whenever RACF uses this profile to deny access to a data set
UACC	The universal access authority for the data set or data sets protected by the profile
AUDIT	The type of auditing to be performed for the data set or data sets protected by the profile
CATEGORY	The security categories to be assigned to the data set or data sets protected by the profile
SECLABEL	The security label of the data set or data sets protected by the profile (evaluated in Labeled Security Mode only)
SECLEVEL	The security level of the data set or data sets protected by the profile (evaluated in Labeled Security Mode only)
ERASE	A setting that indicates whether the data set or data sets protected by the profile are to be erased when they are scratched
UNIT	The unit type on which the data set resides (for discrete profiles only)
VOLUME	The volume on which the data set resides (for discrete profiles only)

Associated with those profiles is the access control list (ACL) for the profile. Each ACL entry defines the access rights of a user or a group with respect to the resource protected by the profile.

Attributes within an ACL entry are:

- access type (none, execute, read, update, control, alter)
- user IDs and group IDs allowed for the access type
- conditions of access (among other):
    - WHEN(CONSOLE(console-id ...))
      Modifies the access authority. Specifies that the identified users or groups have the specified access authority when executing commands originating from the specified system console

- WHEN(JESINPUT(device-name ...))
  Modifies the access authority. Specifies that the identified users or groups have the specified access authority when entering the system through the specified JES input device
- WHEN(PROGRAM(program-name...))
  Modifies the access authority. Specifies that the identified users or groups have the specified access authority when executing the specified program
- WHEN(TERMINAL(terminal-id ...))
  Modifies the access authority. Specifies that the identified users or groups have the specified access authority when logged on to the specified terminal

8.4.2.2 General resource profiles

Other protected resources defined in this Security Target (except the z/OS UNIX file system objects and z/OS UNIX IPC objects) are protected by general resource profiles that contains the resource class and the resource attributes. As with profiles for z/OS data sets, an access control list with entries defining the access types for individual users and / or groups can be defined for each such resource profile. The semantics of the individual access rights are defined by the resource manager responsible for the management of the resources protected by such a profile. Different resource classes may have different resource managers responsible for the protection and management of the resources within the class.

The structure of a general resource profile is defined in the following table (omitting fields that are not relevant for the Security Policy as defined in this Security Target:

Name	Description
Class name	Name of the resource class the profile belongs to
Profile name	Name of the generic resource profile
OWNER(user ID or groupname)	The owner of the profile
NOTIFY	The user who is to be notified whenever RACF uses this profile to deny access to a resource
UACC	The universal access authority for the resource or resources protected by the profile
AUDIT	The type of auditing to be performed for the resource or resources protected by the profile
FROM	The name of a profile that is to be used as a model
FCLASS	The class of the model profile
FGENERIC	A setting that indicates that the model profile name is to be treated as a generic name
FVOLUME	The volume that is to be used to locate the model profile
CATEGORY	The security categories to be assigned to the resource or resources protected by the profile (evaluated in Labeled Security Mode only)
SECLABEL	The security label of the resource or resources protected by the profile (evaluated in Labeled Security Mode only)
SECLEVEL	The security level of the resource or resources protected by the profile (evaluated in Labeled Security Mode only)

LEVEL	An installation-defined level
SINGLEDSN	The tape volume protected by this profile can contain only one data set (TAPEVOL class only)
TIMEZONE	The time zone in which a terminal resides (TERMINAL class only)
TVTOC	A setting that specifies that RACF is to create a tape volume table of contents (TVTOC) when a user creates the first output data set on the tape volume (TAPEVOL class only)
WHEN	The times when the terminal or terminals protected by the profile can be used to access the system (TERMINAL class only)

8.4.2.3 z/OS UNIX file system resources

z/OS UNIX file system resources are not protected by RACF profiles but by permission bits and extended attributes stored in the z/OS UNIX file system. The evaluated configuration supports two different z/OS UNIX file system types: zFS and HFS. A file system for both file system types is always implemented in a single z/OS data set.

In the case of zFS the extended attributes also contain the security label (evaluated in Labeled Security Mode only); therefore, a zFS file system can have different security labels associated with different files. If varying security labels are to be used within one zFS file system, the data set containing the zFS file system must be created with the SYSMULTI security label. After creation of the file system, the security label of the data set must then be set to SYSHIGH.

In the case of HFS, the extended attributes do not contain a security label and therefore in Labeled Security Mode a HFS file system must be contained in a z/OS data set with a defined security label. All z/OS UNIX files in this HFS will then automatically inherit the security label of the hosting z/OS data set.

See DAC for UNIX objects for details of the access control strategy for z/OS UNIX file system objects.

8.4.3 RACF configuration and management

8.4.3.1 Configuring RACF with the SETROPTS command

The SPECIAL and AUDITOR roles can define system wide-options of RACF with the SETROPTS command. This command can be used (among other actions) to:

- Choose the resource classes that RACF is to protect. {SM.3::SM.3.1}
- Set the universal access authority (UACC) for otherwise undefined terminals. {SM.3::SM.3.2}
- Specify logging of certain RACF commands and events. {SM.3::SM.3.3}
- Enable or disable list-of-groups access checking. {SM.3::SM.3.4}
- Display options currently in effect. {SM.3::SM.3.5}
- Enable generic profile checking for all active classes. {SM.3::SM.3.6}
- Establish password syntax rules. {SM.3::SM.3.7}
- Activate password processing for checking previous passwords, limit invalid password attempts, and warn of password expiration. {SM.3::SM.3.8}
- Control global access checking for selected individual resources or generic names with selected generalized access rules. {SM.3::SM.3.9}
- Set the passwords for authorizing use of the RVARY command. {SM.3::SM.3.10}

- Initiate refreshing of in-storage generic profile lists and global access checking tables. {SM.3::SM.3.11}
- Enable or disable shared profiles through RACLIST processing for general resources. {SM.3::SM.3.12}
- Activate auditing of access attempts to RACF-protected resources based on installation-defined security levels. {SM.3::SM.3.13}
- Activate enhanced generic naming. {SM.3::SM.3.14}
- Activate profile modeling for GDG, group, and user data sets. {SM.3::SM.3.15}
- Activate protection for data sets with single-level names. {SM.3::SM.3.16}
- Control logging of real data set names. {SM.3::SM.3.17}
- Control the job entry subsystem (JES) options implemented in RACF. {SM.3::SM.3.18}
- Activate tape data set protection. {SM.3::SM.3.19}
- Enable protection of data sets by default (PROTECTALL(FAILURES)). {SM.3::SM.3.20}
- Enable the erasure of scratched DASD data sets. {SM.3::SM.3.21}
- Activate program control. {SM.3::SM.3.22}
- Control whether a profile creator's user ID is automatically added to the profile's access list. {SM.3::SM.3.23}

Some administration activities can be delegated to user with other roles. See the definition of those roles for the administrative options that can be set or defined by those roles.

To operate in correspondence with the requirements in this Security Target, the system administrator needs to configure RACF (using the SETROPTS command) with the following options: CATDSNS(FAILURES), NOCOMPATMODE, ERASE(ALL), GENERIC(*), PROTECTALL(FAILURES), CLASSACT (TEMPDSN), JES(BATCHALLRACF). In Labeled Security Mode the following options need to be set in addition: MLACTIVE(FAILURES), MLFSOBJ(ACTIVE), MLIPCOBJ(ACTIVE), MLS(FAILURES), MLSTABLE, SECLABELCONTROL. {SM.3::SM.3.24}. Additional parameter for the PASSWORD operand need to be set to define the password policy. See RACF Passwords and Password Phrases for more information.

8.4.3.2 RACF commands

The administration of RACF is performed by a set of commands. Users need the required authorities or roles to issue those commands or specific parameter of those commands, which are defined in a table later in this section. The main RACF commands are:

- ADDGROUP, ALTGROUP, DELGROUP
  Commands to define a new group profile, modify an existing group profile or delete a group profile {SM.3::SM.3.25}
- ADDUSER, ALTUSER, DELUSER
  Commands to define a new user profile, modify an existing user profile or delete a user profile {SM.3::SM.3.26}
- ADDSD, ALTDSD, DELDSD
  Commands to define a new z/OS data set profile, modify an existing z/OS data set profile or delete an existing z/OS data set profile {SM.3::SM.3.27}
- CONNECT, REMOVE
  Command to connect a user to or remove a user from a group {SM.3::SM.3.28}
- LISTGROUP, LISTUSER, LISTDSD

Commands to list user, group or z/OS data set profiles {SM.3::SM.3.29}

- RDEFINE, RALTER, RDELETE
  Commands to define, modify or delete a general resource profile {SM.3::SM.3.30}
- RLIST
  Command to list a general resource profile {SM.3::SM.3.31}
- PASSWORD
  Command to specify a user's password {SM.3::SM.3.32}
- PHRASE
  Command to specify a user's password phrase {SM.3::SM.3-R10-RACF-1}
- PERMIT
  Command to maintain the access list of a resource profile {SM.3::SM.3.33}
- RACDCERT
  Command to maintain X5.09v3 digital certificates, certificate mapping filters, certificate mapping criteria, and key rings in the RACF database. {SM.3::SM.3-R13-RACF-3}
- RACMAP
  Command to establish mappings between distributed user identities and local RACF user IDs {SM.3::SM.3-R12-RACF-2}.
- SETROPTS
  Command to set specific RACF options (see section above for details) {SM.3::SM.3.34}

There are the following options how a RACF command can be issued:

- as a TSO command
- as an operator command
- with command direction (command can be directed to run under the authority of a user ID on a remote node, or the same node using the AT operand. Use of this operand is usually restricted as indicated in the table below). [Note: This method is part of the RACF Remote Sharing Facility (RRSF) which is not a part of this evaluation, and will not be considered further.]
- with automatic command direction (activated using the SET AUTODIRECT command. Use is restricted by RACF profiles as for command direction)[Note: This method is part of the RACF Remote Sharing Facility (RRSF) which is not a part of this evaluation, and will not be considered further.]
- from the RACF parameter library
- via the R_admin callable service

Not all commands can be issued in all options. For example some commands can not be used as TSO commands, and some can not be used as operator commands.

{SM.3::SM.3-R12-RACFEAL5-30} For commands (except RVARY) that can be issued as either a TSO command or as an operator command, the following security requirements apply in addition to any specified by the commands themselves when the command is issued as an operator command:

- The command issuer must be logged on to the console.
- If the OPERCMDS class is active and SETR RACLISTed then the command issuer must have READ authority to the OPERCMDS profile protecting the command, if one exists:

Command Name	***OPERCMDS Resource name***
ADDGROUP	Subsystem-name.ADDGROUP

Command Name	*OPERCMDS Resource name*
	(Note: for all the RACF TSO commands, when issued as an operator command the OPERCMDS resource name uses the full command name, even if the user issued the command using an abbreviation such as AG)
ADDSD	Subsystem-name.ADDSD
ADDUSER	Subsystem-name.ADDUSER
ALTDSD	Subsystem-name.ALTDSD
ALTGROUP	Subsystem-name.ALTGROUP
ALTUSER	Subsystem-name.ALTUSER
CONNECT	Subsystem-name.CONNECT
DELDSD	Subsystem-name.DELDSD
DELGROUP	Subsystem-name.DELGROUP
DELUSER	Subsystem-name.DELUSER
LISTDSD	Subsystem-name.LISTDSD
LISTGRP	Subsystem-name.LISTGRP
LISTUSER	Subsystem-name.LISTUSER
PASSWORD	Subsystem-name.PASSWORD
PHRASE	Subsystem-name.PASSWORD
PERMIT	Subsystem-name.PERMIT
RALTER	Subsystem-name.RALTER
RDEFINE	Subsystem-name.RDEFINE
RDELETE	Subsystem-name.RDELETE
REMOVE	Subsystem-name.REMOVE
RLIST	Subsystem-name.RLIST
SEARCH	Subsystem-name.SEARCH
SETROPTS	Subsystem-name.SETROPTS (Note: READ authority is required to issue SETROPTS LIST, but UPDATE is required if

Command Name	*OPERCMDS Resource name*
	any other operands are specified.)

{SM.3::SM.3-R12-RACFEAL5-31} For commands that can be issued as operator commands but not as TSO commands, the following security requirements apply:

- If you are logged on to the console, and the command is protected by an OPERCMDS profile, then you must have READ authority to that profile:

Command Name	*OPERCMDS Resource name*
DISPLAY	Subsystem-name.DISPLAY.SIGNON Note: No OPERCMDS authority check is performed when this command is issued from a RACF parameter library member.
RESTART	Subsystem-name.RESTART
SET	*subsystem-name*.SET.AUTOAPPL s*ubsystem-name*.SET.AUTODIRECT *subsystem-name*.SET.AUTOPWD *subsystem-name*.SET.INCLUDE *subsystem-name*.SET.JESNODE *subsystem-name*.SET.LIST *subsystem-name*.SET.PWSYNC *subsystem-name*.SET.TRACE Note: No OPERCMDS authority check is performed when this command is issued from a RACF parameter library member.
SIGNOFF	Subsystem-name.SIGNOFF Note: No OPERCMDS authority check is performed when this command is issued from a RACF parameter library member.
STOP	Subsystem-name.STOP
TARGET	*subsystem-name*.TARGET.DESCRIPTION *subsystem-name*.TARGET.LIST *subsystem-name*.TARGET.LOCAL *subsystem-name*.TARGET.NODE *subsystem-name*.TARGET.OPERATIVE **Note:** TARGET.OPERATIVE also protects the DELETE and DORMANT operands. *subsystem-name*.TARGET.PREFIX *subsystem-name*.TARGET.PROTOCOL *subsystem-name*.TARGET.PURGE

Command Name	*OPERCMDS Resource name*
	subsystem-name.TARGET.WDSQUAL *subsystem-name*.TARGET.WORKSPACE Note: No OPERCMDS authority check is performed when this command is issued from a RACF parameter library member.

- If you are not logged on to the console, or the command is not protected by an OPERCMDS profile, then (except for the DISPLAY command) you must issue the command from a console with MASTER or SYSTEM authority.

To use a specific RACF command, a user needs the specific authority to use the command or specific parameter of the command. The authorities required are listed in the following table:

Command	Authorities required for use
ADDGROUP	{SM.3::SM.3-R12-RACFEAL5-1} General: • SPECIAL or • group-SPECIAL and the superior group is within the scope of group-SPECIAL authority, or • owner of the superior group, or • JOIN authority to the superior group Special parameter authorization: • SPECIAL or UPDATE authority via field-level access control to add segments to a group's profile • For use of the SHARED keyword: SPECIAL or READ authority to the SHARED.IDS resource in the UNIXPRIV class
ADDSD	{SM.3::SM.3-R12-RACFEAL5-2} The level of authority required to use the ADDSD command and the types of profiles the caller can define are: To protect a user data set with RACF, one of the following must be true: • The high-level qualifier of the data set name (or the qualifier supplied by the RACF naming conventions table) must match the caller's user ID, or • SPECIAL, or • The user ID for the data set profile must be within the scope of a group in which the caller has the group-SPECIAL attribute. To protect a group data set with RACF, one of the following must be true: • CREATE authority in the group, or • SPECIAL, or • OPERATIONS attribute and not be connected to the group, or • The data set profile must be within the scope of the group in which the caller has the group-SPECIAL attribute. • The data set profile must be within the scope of the group in which

Command	Authorities required for use

	the caller has the group-OPERATIONS attribute, and he must not be connected to the group.
	Special parameter authorization:
	• To assign a security category to a profile, you must have the SPECIAL attribute or have the category in your user profile.
	• To assign a security level to a profile, you must have the SPECIAL attribute or, in your own profile, a security level that is equal to or greater than the security level you are defining.
	• To assign a security label to a profile, you must have the SPECIAL attribute or (when SETROPTS SECLABELCONTROL is not active) READ authority to the security label profile.
	• To add non-base segments, SPECIAL or UPDATE authority via field-level access control.
	• To add a discrete profile for a VSAM data set already RACF-protected by a generic profile, you must have ALTER access authority to the catalog or to the data set through the generic profile.
	• To specify a model data set profile (using, as required, FROM, FCLASS, FGENERIC, and FVOLUME), you must have sufficient authority over the model profile (the *from* profile). RACF makes the following checks until one of the conditions is met:
	  • You have the SPECIAL attribute.
	  • The *from* profile is within the scope of a group in which you have the group-SPECIAL attribute.
	  • You are the owner of the *from* profile.
	  • The high-level qualifier of the profile name (or the qualifier supplied by the naming conventions routine) is your user ID.
	  • For a discrete profile, you are on the access list in the *from* profile with ALTER authority. (If you have any lower level of authority, you cannot use the profile as a model.)
	  • For a discrete profile, your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is on the access list in the *from* profile with ALTER authority.
	  • For a discrete profile, the UACC is ALTER.

ADDUSER	{SM.3::SM.3-R12-RACFEAL5-3} General:
	• SPECIAL or
	• CLAUTH for the USER class and one of the following is true:
	  ◦ The caller is the owner of the default group specified
	  ◦ The caller has JOIN authority in the default group specified
	  ◦ The default group specified is within the scope of agroup where the caller has group-SPECIAL
	Special parameter authorization:
	• SPECIAL to give the new user the OPERATIONS, SPECIAL or AUD-

Command	Authorities required for use
	ITOR attribute • SPECIAL to assign a security category to a profile or the category must be in the caller's user profile • SPECIAL to assign a security level to a profile or the security level in the caller's profile is equal or greater than the security level assigned to the new user • • When defining information within a segment other than the base segment: ◦ SPECIAL, or ◦ UPDATE authority to the desired field through field-level access control • For use of the SHARED keyword: SPECIAL or READ authority to the SHARED.IDS resource in the UNIXPRIV class
ALTDSD	{SM.3::SM.3-R12-RACFEAL5-4} The level of authority required to use the ALTDSD command and the types of profiles the caller can define are: To protect a user data set with RACF, one of the following must be true: • The high-level qualifier of the data set name (or the qualifier supplied by the RACF naming conventions table) must match the caller's user ID, or • SPECIAL, or • The user ID for the data set profile must be within the scope of a group in which the caller has the group-SPECIAL attribute. • The caller is the owner of the profile • For a discrete profile: the caller has ALTER authority to the profile or the caller's current connect group (or, if list-of-groups checking is active) has ALTER authority Special parameter authorization: • For use of the GLOBALAUDIT keyword: AUDITOR or the data set profile is within the scope of a group in which the caller has the group-AUDITOR attribute • For access to the DFP or TME segment: field-level access authority to those segments • SPECIAL to assign a security label, or when SETR SECLABELCONTROL is not active, READ authority to the specified SECLABEL. • SPECIAL to assign a security category to a profile or remove a security category from a profile, or the category must be in the caller's user profile. SPECIAL to assign a security level to a profile or the security level in the caller's profile is equal or greater than the security level assigned to the new user
ALTGROUP	{SM.3::SM.3-R12-RACFEAL5-5} General: • To issue the command,, except as specified below: • SPECIAL

Command	Authorities required for use
	• The group profile is within the scope of a group in which you have the group-SPECIAL attribute • You are the current owner of the group. Special parameter authorization: • To change the superior group of a group: • SPECIAL or • group-SPECIAL and the current and new superior group is within the scope of group-SPECIAL authority, or • owner of the current and new superior group, or • JOIN authority to the current and new superior group • A combination of group-SPECIAL, owner or JOIN authority where one of those applies for the current and one of those applies for the new superior group • SPECIAL or permission through field-level access checking in order to add, delete, or alter segments of a group's profile • For use of the SHARED keyword: SPECIAL or READ authority to the SHARED.IDS resource in the UNIXPRIV class
ALTUSER	{SM.3::SM.3-R12-RACFEAL5-6} General: • SPECIAL (allows use of all operands except UAUDIT/NOUAUDIT) Special parameter authorization: • If the owner of the user profile is within the scope of a group in which the group-SPECIAL attribute, he can use all of the operands except SPECIAL, AUDITOR, OPERATIONS, NOEXPIRED and UAUDIT/NOUAUDIT. • The owner of the user's profile can use any of the following operands for user-related attributes: ◦ ADSP | NOADSP ◦ DATA | NODATA ◦ DFLTGRP ◦ GRPACC | NOGRPACC ◦ MODEL | NOMODEL ◦ NAME ◦ OIDCARD | NOOIDCARD ◦ OWNER ◦ PASSWORD | NOPASSWORD ◦ PHRASE | NOPHRASE ◦ RESTRICTED | NORESTRICTED ◦ RESUME | NORESUME

Command	Authorities required for use
	◦ REVOKE | NOREVOKE ◦ WHEN • Each user can change his or her name field, default group or model data set profile name (using the NAME, DFLTGRP, or MODEL operand, respectively). • You can use the GROUP, AUTHORITY, and UACC operands for group-related user attributes if you have JOIN or CONNECT authority, if the group profile is within the scope of a group in which you have the group-SPECIAL attribute, or if you are the owner of the specified group. • To specify the AUDITOR/NOAUDITOR, SPECIAL/NOSPECIAL, and OPERATIONS/NOOPERATIONS operands as system-wide user attributes, the caller must have the SPECIAL attribute. • To specify the UAUDIT/NOUAUDIT operand, either the caller must have the AUDITOR attribute, or the user profile must be within the scope of a group in which the caller has the group-AUDITOR attribute. • The CLAUTH and NOCLAUTH operands can be specified if the caller is the owner of the user's profile and has the CLAUTH attribute for the class to be added or deleted. • To assign a security category to a profile, or to delete a category from a profile, one of the following must be true: • If the user profile is within the scope of a group in which you have the group-SPECIAL attribute, or if you are the owner of the specified user, the category you are adding or deleting must be in your user profile. • You have the SPECIAL attribute. • To assign a security level to a profile, or to delete a security level from a profile, one of the following must be true: • If the user profile is within the scope of a group in which you have the group-SPECIAL attribute, or if you are the owner of the specified user, the security level in your user profile must be equal to or greater than the security level you are assigning or deleting. • You have the SPECIAL attribute. • • To change information within a segment other than the base segment, the caller must have one of the following: ◦ The SPECIAL attribute ◦ At least UPDATE authority to the desired field within the segment through field-level access control. • To reset passwords and password phrases or to resume user IDs, the caller must have at least one of the following authorizations: ◦ SPECIAL.

Command	Authorities required for use
	◦ group-SPECIAL authority over the user profile. ◦ The caller is the OWNER of the user profile. ◦ The caller has sufficient access to the IRR.PASSWORD.RESET resource in the FACILITY class. ◦ The caller has sufficient access to an appropriate resource in the FACILITY class (IRR.PWRESET.OWNER.owner or IRR.PWRESET.TREE.owner), and both of the following conditions are also true: ▪ The other user does not have the SPECIAL, OPERATIONS, AUDITOR, or PROTECTED attribute. ▪ The caller is not excluded from altering the user by the IRR.PWRESET.EXCLUDE.excluded-user resource in the FACILITY class. • When the caller's reset and resume authority is through access to the IRR.PASSWORD.RESET resource, the IRR.PWRESET.OWNER.owner resource, or the IRR.PWRESET.TREE.owner resource, the following requirements apply: ◦ If the caller has READ access, he can: ▪ Use the PASSWORD operand to reset a password (to an expired password) for a user who does not have the SPECIAL, OPERATIONS, AUDITOR, or PROTECTED attribute. ▪ Use the PHRASE operand to reset a password phrase (to an expired password phrase) for a user with an assigned password phrase who does not have the SPECIAL, OPERATIONS, AUDITOR, or PROTECTED attribute. Note: The caller cannot use the PHRASE operand to add a password phrase for a user who does not have one. ▪ Use the RESUME operand, without specifying a date, for a user who does not have the SPECIAL, OPERATIONS, AUDITOR, or PROTECTED attribute. ◦ If the caller has UPDATE access, he can: ▪ Use the PASSWORD, PHRASE, and RESUME operands as noted for READ access. ▪ Use the NOEXPIRED operand (with PASSWORD or PHRASE) for a user who does not have the SPECIAL, OPERATIONS, AUDITOR, or PROTECTED attribute. ◦ If the caller has CONTROL access, he can: ▪ Use the PASSWORD, PHRASE, RESUME, and NOEXPIRED operands as noted for READ and UPDATE access. ▪ Reset the password or password phrase within the minimum change interval for a user who does not have the SPECIAL, OPERATIONS, AUDITOR, or PROTECTED attribute. • For use of the SHARED keyword: SPECIAL or READ authority to the SHARED.IDS resource in the UNIXPRIV class
CONNECT	{SM.3::SM.3-R12-RACFEAL5-7} General:

Command	Authorities required for use
	• SPECIAL, or • Group-SPECIAL in the group • Ownership of the group • JOIN authority in the group • CONNECT authority in the group • A caller can not give a higher level of authority in the group than he has himself.
DELDSD	{SM.3::SM.3-R12-RACFEAL5-8} General: • SPECIAL, or • The data set profile is within the scope of a group in which the caller has the group-SPECIAL attribute, or • The high-level qualifier of the profile name or the qualifier supplied by the RACF naming conventions table is the caller's user ID, or • The caller is the owner of the profile • For a discrete profile: the caller is on the access list with ALTER authority, or • For a discrete profile, the caller's group or one of the caller's groups (if checking list of groups is active) is on the access list and has ALTER authority. • For a discrete profile, the universal access authority is ALTER.
DELGROUP	{SM.3::SM.3-R12-RACFEAL5-9} General: • SPECIAL, or • The group to be deleted is within the scope of a group where the caller has the group-SPECIAL in the group, or • The caller is the Owner of the group to be deleted, or • The caller is the Owner of a superior group of the group to be deleted • JOIN authority in the superior group
DELUSER	{SM.3::SM.3-R12-RACFEAL5-10} General: • SPECIAL, or • Group-SPECIAL for the group where the user profile to be deleted is within the scope of the group, or • Ownership of the user profile Other: • A user profile that has a distributed identity filter associated with it can only be deleted after the distributed identity filter has been deleted.
DISPLAY	{SM.3::SM.3-R12-RACFEAL5-25} None (can be used an operator command only. RACF allows restricting the use of specific operator commands to defined users).
HELP	No restrictions
LISTDSD	{SM.3::SM.3-R12-RACFEAL5-11} For listing the RACF segment of a data set

Command	Authorities required for use
	profile: • SPECIAL, or • The data set profile is within the scope of a group in which the caller has the group-SPECIAL attribute, or • OPERATIONS, or • The data set profile is within the scope of a group in which the caller has the group-OPERATIONS attribute, or • AUDITOR, or • The data set profile is within the scope of a group in which the caller has the group-AUDITOR attribute, or • The high-level qualifier of the profile name or the qualifier supplied by the RACF naming conventions table is the caller's user ID, or • The caller is the owner of the profile • The caller is on the access list with READ authority, or • The caller's group or one of the caller's groups (if checking list of groups is active) is on the access list and has READ authority, or • The universal access authority is READ, or • The caller has READ access for the profile name from the GLOBAL ENTRY TABLE (if the table contains an entry for the profile) For listing the DFP or TME segment of a data set profile: • SPECIAL, or • AUDITOR, or • READ authority to the desired field within the segment through field-level access control. Special parameter authorization: • To specify the AUTHUSER operand to display the access list for a profile, one of the following conditions must be met for each profile to be listed: ◦ SPECIAL, or ◦ The data set profile is within the scope of a group in which the caller has the group-SPECIAL attribute, or ◦ OPERATIONS, or ◦ The data set profile is within the scope of a group in which the caller has the group-OPERATIONS attribute, or ◦ AUDITOR, or ◦ The data set profile is within the scope of a group in which the caller has the group-AUDITOR attribute, or ◦ The high-level qualifier of the profile name or the qualifier supplied by the RACF naming conventions table is the caller's user ID, or ◦ The caller is the owner of the profile

Command	Authorities required for use
	◦ The caller has ALTER access for the profile name from the GLOBAL ENTRY TABLE (if this table contains an entry for the profile). ◦ For a discrete profile, the caller is on the profile's access list with ALTER authority. ◦ For a discrete profile, the caller's current connect group (or, if list-of-groups checking is active, any group to which the caller is connected) is in the access list and has ALTER authority. ◦ For a discrete profile, the universal access authority is ALTER. Other: • To display the type of access attempts (as specified by the GLOBALAUDIT operand on the ALTDSD command) that are being logged on the SMF data set, either the caller must have the AUDITOR attribute or the profile must be within the scope of a group in which the caller has the group-AUDITOR attribute.
LISTGRP	{SM.3::SM.3-R12-RACFEAL5-12} For listing the RACF segment of a group profile: • SPECIAL, or • AUDITOR, or • For each group to be listed at least one of the following is true: ◦ The caller has group-SPECIAL the group or the group profile is within the scope of a group in which the caller has the group-SPECIAL attribute, or ◦ The caller has group-AUDITOR for the group or the group profile is within the scope of a group in which the caller has the group-AUDITOR attribute, or ◦ The caller is the owner of the group, or ◦ The caller has JOIN or CONNECT authority for the group For listing the other segments of a group profile: • SPECIAL, or • AUDITOR, or • READ access to the desired field through field-level access control
LISTUSER	{SM.3::SM.3-R12-RACFEAL5-13} For listing the RACF segment of a user profile: • SPECIAL, or • AUDITOR, or • For each user to be listed at least one of the following is true: ◦ the user profile is within the scope of a group in which the caller

Command	Authorities required for use
	has the group-SPECIAL attribute, or ◦ the user profile is within the scope of a group in which the caller has the group-AUDITOR attribute, or ◦ The caller is the owner of the user profile, or ◦ The caller has READ access to the IRR.LISTUSER resource in the FACILITY class and the user does not have the SPECIAL, AUDITOR, or OPERATIONS attribute. ◦ The caller has READ access to an appropriate resource (IRR.LU.OWNER.owner or IRR.LU.TREE.owner) in the FACILITY class, and both of the following conditions are also true: ▪ The user does not have the SPECIAL, AUDITOR, or OPERATIONS attribute. ▪ The caller is not excluded from listing the user by the IRR.LU.EXCLUDE.excluded-user resource in the FACILITY class. For listing other segments of a user profile: • SPECIAL, or • AUDITOR, or • READ authority to the desired field within the segment through field-level access checking Other: • If the caller has the group-SPECIAL or group-AUDITOR attribute and the installation has assigned security levels and security categories to user profiles, the caller must have the following to be able to display the RACF segment of a user's profile: ◦ A security level equal to, or greater than, that in the user profile the caller is trying to display ◦ All security categories contained in the user profile the caller is trying to display are contained in the caller's own user profile. • The value of the UAUDIT/NOUAUDIT operand is only listed if the authorization is given by the AUDITOR or group-AUDITOR authorities.
PASSWORD	{SM.3::SM.3-R12-RACFEAL5-14} General: • To change a user's password or password phrase or change interval: ◦ The caller is the user himself, or ◦ SPECIAL, or ◦ the user's profile is within the scope of a group in which the caller has group-SPECIAL • To reset another user's password to the default value: ◦ The caller is the owner of the user's profile ◦ SPECIAL, or ◦ the user's profile is within the scope of a group in which the caller has group-SPECIAL

Command	Authorities required for use
PERMIT	{SM.3::SM.3-R12-RACFEAL5-15} General: • SPECIAL, or • The profile is within the scope of a group in which the caller has the group-SPECIAL attribute, or • The caller is the owner of the resource, or • For a discrete profile: the caller has ALTER authority (either via the standard access list, or via group access or via UACC) • For profiles in the DATASET class: the high-level qualifier of the profile name (or the qualifier supplied by the naming conventions routine) is the caller's user ID. • For profiles in the FILE or DIRECTRY class: the second qualifier of the profile name is the caller's user ID. (Note: the FILE and DIRECTRY classes apply only in z/VM systems, and will not be considered in this evaluation even though the classes exist in RACF for z/OS.)
PHRASE	See PASSWORD
RACDCERT	See separate table
RACMAP	{SM.3::SM.3-R13-RACFEAL5-16} General: • SPECIAL, or • Sufficient authority to the IRR.IDIDMAP.*function* resource in the FACILITY class, where *function* is MAP, DELMAP, QUERY, or LISTMAP. READ authority is required for the caller to create, delete, or list a distributed identity filter for his own RACF user ID. UPDATE authority is required for the caller to create, delete, or list a distributed identity filter for another RACF user ID.
RACPRIV	{SM.3::SM.3-R12-RACFEAL5-17} General: • READ access to the profile IRR.WRITEDOWN.BYUSER in the FACILITY class.
RALTER	{SM.3::SM.3-R12-RACFEAL5-18} General (with the exception of the GLOBALAUDIT parameter): • SPECIAL, or • The profile is within the scope of a group in which the caller has the group-SPECIAL attribute, or • The caller is the owner of the profile, or • For a discrete profile: the caller has ALTER authority (either via the standard access list, or via group access or via UACC) • For profiles in the FILE or DIRECTRY class: the second qualifier of the profile name is the caller's user ID. (Note: the FILE and DIRECTRY classes apply only in z/VM systems, and will not be considered in this evaluation even though the classes exist in RACF for z/OS.)

Command	Authorities required for use
	• To assign a security category to a profile, or to delete a category from a profile, the caller must have the SPECIAL attribute, or the category must be in the caller's user profile. • To assign a security level to a profile, or to delete a security level from a profile, the caller must have the SPECIAL attribute, or, in his own profile, a security level that is equal to or greater than the security level he is assigning or deleting. • To assign a security label, SPECIAL, or when SETROPTS SECLABELCONTROL is not active, the caller must have READ access to the security label profile. Special parameter authorization: • For defining a delegated resource (by specifying the RACF-DELEGATED string in the APPLDATA of the profile), when the profile has a SECLABEL and SETR SECLBELCONTROL is active, SPECIAL • For modifying information in segments other than the base segment: ◦ SPECIAL, or ◦ UPDATE Access allowed by field-level access control for the fields to be modified • For use of the GLOBALAUDIT parameter: ◦ AUDITOR, or ◦ The profile is within the scope of a group for which the caller has the group-AUDITOR attribute. • For use of the ADDMEM parameter: ◦ For classes other than SECLABEL, PROGRAM, SECDATA, GLOBAL, RACFVARS, and NODES, if the member resources are already RACF-protected by a member class profile or as a member of a profile in the same grouping class, one of the following must be true: ▪ The caller has ALTER access authority to the member, or ▪ The caller is the owner of the member resource, or ▪ The member resource is within the scope of a group in which the caller has the group-SPECIAL attribute, or ▪ The caller has the SPECIAL attribute. ◦ For classes other than SECLABEL, PROGRAM, SECDATA, GLOBAL, RACFVARS, and NODES, if the member resources are not RACF-protected (that is, there is no profile defined for that member), one of the following must be true: ▪ The caller has CLAUTH authority to define resources in the member resource class. ▪ The caller has the SPECIAL attribute. ◦ To add a member to a profile in the RACFVARS or NODES class, one of the following must be true:

Command	Authorities required for use
	▪ The caller has CLAUTH authority to define resources in the specified class ▪ The caller has the SPECIAL attribute. ▪ The caller is the owner of the profile indicated by profile-name. ▪ The caller has ALTER access authority to the profile indicated by profile-name. ◦ To add a member to a profile in the PROGRAM or SECDATA class, one of the following must be true: ▪ You have CLAUTH authority to define resources in the specified class (for example, PROGRAM or SECDATA). ▪ You have the SPECIAL attribute. ◦ To add a member to a profile in the GLOBAL class (other than the GLOBAL DATASET, GLOBAL DIRECTRY, or GLOBAL FILE profile) ▪ If the profile resource-name is already RACF-protected by a profile in class class-name: • ALTER access authority to the profile resource-name in class class-name, or • The caller is the OWNER of the profile resource-name, or • The profile resource-name in class class-name is within the scope of a group in which the caller has the group-special attribute, or • The caller has the SPECIAL attribute. ▪ If the profile resource-name is not already RACF-protected (that is, there is no profile defined for that member in class class-name): • The caller has CLAUTH authority to define resources in the class class-name, or • The caller has the SPECIAL attribute. ◦ To add a member to the GLOBAL DATASET profile, one of the following must be true: ▪ The caller is the owner of the DATASET profile in the GLOBAL class. ▪ The member is within the scope of a group in which the caller has the group-SPECIAL attribute. ▪ The caller has the SPECIAL attribute. ◦ To add a member to the GLOBAL DIRECTRY or GLOBAL FILE profile, the caller must have the SPECIAL attribute. (Note: the FILE and DIRECTRY classes apply only in z/VM systems, and will not be considered in this evaluation even though the classes exist in RACF for z/OS.) • For use of the DELMEM operand:

Command	Authorities required for use
	◦ If class-name is specified as GLOBAL, the rules for member are the same as given for ADDMEM. If class-name is specified as SECDATA, member should be a valid SECLEVEL name or category name. • For use of the ADDVOL operand: ◦ SPECIAL, or ◦ CLAUTH attribute for the TAPEVOL resource class in addition to the other authorization requirements for using the RALTER command. • For use of the GLOBALAUDIT operand: ◦ AUDITOR, or ◦ The resource profile must be within the scope of a group in which the caller has the group-AUDITOR attribute.
RDEFINE	{SM.3::SM.3-R12-RACFEAL5-19} General (with the exception of the GLOBALAUDIT parameter): • SPECIAL, or • If the caller has CLAUTH authority for the GLOBAL class, and group-SPECIAL authority in a group, he can add members whose high-level qualifier is the group name or a user ID in the scope of the group. This applies only to classes that are sensitive to high-level qualifiers, such as DATASET. • If the name to be defined is not already defined to RACF as a member of a resource group and the caller is defining a profile in a normal (non-member, non-grouping) class, a member class, or a member of a grouping class, he must have CLAUTH authority for the specified class. • If the resource to be defined is a discrete name already defined to RACF as a member of a resource group, the caller can define it as a resource to RACF if he has ALTER authority, or if the resource group profile is within the scope of a group in which he has the group-SPECIAL attribute, or if he is the owner of the resource group profile. If authority conflicts arise because the resource is a member of more than one group and the user's authority in those groups differs, RACF resolves the conflict by using the least restrictive authority. • If the caller defines a profile in the FILE or DIRECTRY class, one of the following must be true: ◦ The second qualifier of the profile name must match the caller's user ID ◦ The caller has the SPECIAL attribute ◦ The profile name must be within the scope of a group in which the caller has the group-SPECIAL attribute. (Note: the FILE and DIRECTRY classes apply only in z/VM systems, and will not be considered in this evaluation even though the classes exist in RACF for z/OS.) • If the caller does not have the SPECIAL attribute and the SETROPTS GENERICOWNER option is in effect, and if an existing generic profile

Command	Authorities required for use
	protects the profile name the caller is defining, he needs to own the less specific profile. GENERICOWNER does not apply to the PROGRAM class. • To assign a security category to a profile, the caller must have the category in his user profile. • To assign a security level to a profile, the caller's own profile must have a security level that is equal to or greater than the security level he is defining. • To define segments other than the base segment, the caller must have the SPECIAL attribute or he must be permitted to do so through field-level access checking. Other: • Only a SPECIAL user can define a delegated resource (by specifying the RACF-DELEGATED string in the APPLDATA of the profile protecting the resource) when the resource has a SECLABEL and SETROPTS SECLABELCONTROL is in effect. • For Model profiles: To specify a model profile (using, as required, FROM, FCLASS, FGENERIC, and FVOLUME), the caller must have sufficient authority over the model profile (the from profile). RACF makes the following checks until one of the conditions is met: ◦ The caller has the SPECIAL attribute. ◦ The from profile is within the scope of a group in which the caller has the group-SPECIAL attribute. ◦ The caller the owner of the from profile. ◦ If the FCLASS operand is DATASET, the high-level qualifier of the profile name (or the qualifier supplied by the naming conventions routine) is the caller's user ID. ◦ For a discrete profile, the caller is on the access list in the from profile with ALTER authority. ◦ For a discrete profile, the caller's current connect group (or, if list-of-groups checking is active, any group to which the caller is connected) is in the access list in the from profile with ALTER authority. ◦ For a discrete profile, the universal access authority (UACC) is ALTER. Special parameter authorization: • For use of the ADDMEM operand: In addition to the authority needed to issue the RDEFINE command, the caller needs one of the following authorities to add members using the RDEFINE command: ◦ See the requirements specified for the use of the ADDMEM operand for the RALTER command.
RDELETE	{SM.3::SM.3-R12-RACFEAL5-20} General:

Command	Authorities required for use
	• SPECIAL, or • The profile is within the scope of a group in which the caller has the group-SPECIAL attribute, or • The caller is the owner of the profile, or • For a discrete profile: the caller has ALTER authority (either via the standard access list, or via group access or via UACC) • For profiles in the FILE or DIRECTRY class: the second qualifier of the profile name is the caller's user ID. (Note: the FILE and DIRECTRY classes apply only in z/VM systems, and will not be considered in this evaluation even though the classes exist in RACF for z/OS.)
REMOVE	{SM.3::SM.3-R12-RACFEAL5-21} General: • SPECIAL, or • The profile is within the scope of a group in which the caller has the group-SPECIAL attribute, or • The caller is the owner of the group, or • The caller has JOIN or CONNECT authority in the group.
RESTART	None (can be used an operator command only. RACF allows restricting the use of specific operator commands to defined users).
RLIST	{SM.3::SM.3-R12-RACFEAL5-22} General: • SPECIAL, or • The resource profile is within the scope of a group in which the caller has the group-SPECIAL attribute, or • OPERATIONS, or • The resource profile is within the scope of a group in which the caller has the group-OPERATIONS attribute, or • AUDITOR, or • The resource profile is within the scope of a group in which the caller has the group-AUDITOR attribute, or • The caller is the owner of the resource. • If the profile is in the FILE or DIRECTRY class and the second qualifier of the profile name is the caller's user ID. (Note: the FILE and DIRECTRY classes apply only in z/VM systems, and will not be considered in this evaluation even though the classes exist in RACF for z/OS.) • To list the contents of segments other than the base segment, the caller must have the SPECIAL or AUDITOR attribute or the installation must permit the caller to do so through field-level access checking. • If the caller is on the access list for the resource and has at least

Command	Authorities required for use
	READ authority. If you specify ALL, RACF lists only information pertinent to the caller'suser ID. • If the caller's current connect group (or, if list-of-groups checking is active, any group to which the caller is connected) is in the access list and has at least READ authority. • The universal access authority of the resource is at least READ. • The caller has at least read access for the profile name from the GLOBAL ENTRY TABLE (if this table contains an entry for the profile). Other: • The caller will see the type of access attempts, as specified by the GLOBALAUDIT operand, only if he has the AUDITOR attribute or if the resource profile is within the scope of a group in which he has the group-AUDITOR attribute.
RVARY	General: none {SM.3::SM.3-R12-RACFEAL5-32} Other : • No special authority is needed to issue the RVARY command. However, the operator (at the operator console or security console) must approve a change in RACF status or the RACF data sets - or a change in the operational mode if RACF is enabled for sysplex communication - before RACF allows the command to complete. • If the RVARY command changes RACF or database status (ACTIVE/INACTIVE), RACF issues an informational message and the operator is required to enter the password defined by RVARYPW STATUS(*status-pw*) to authorize the change. If the RVARY command switches the RACF data sets (SWITCH) or changes the RACF operating mode (DATASHARE/NODATASHARE), RACF issues an informational message and the operator is required to enter the password defined by RVARYPW SWITCH(*switch-pw*). When RVARY is issued as a RACF operator command from a console with master authority, the default password YES is also accepted for RVARY ACTIVE, RVARY NODATASHARE or RVARY SWITCH commands.
SEARCH	{SM.3::SM.3-R12-RACFEAL5-23} General: • SPECIAL, or • AUDITOR, or • The profile is within the scope of a group in which the caller has the group-SPECIAL or group-AUDITOR attribute, or • If the profile is for a DASD data set, the high-level qualifier of the data set name or the qualifier supplied by the RACF naming conventions table is the caller's user ID, or • If the profile is in the FILE or DIRECTRY class, the second qualifier of the profile name is the caller's user ID, (Note: the FILE and DIRECTRY classes apply only in z/VM systems, and will not be considered in this evaluation even though the classes exist in RACF for z/OS.) or • The caller is on the access list for the profile and has at least READ

Command	Authorities required for use
	authority, or • The caller's current connect group (or, if list-of-groups checking is active, any group to which the caller is connected) is on the access list and has at least READ authority, or • The caller has the OPERATIONS attribute, or the profile is within the scope of a group in which the caller has the group-OPERATIONS attribute, and the class is DATASET or a general resource class that specifies OPER=YES in the static class descriptor table or OPERATIONS(YES) in the dynamic class descriptor table, or • The universal access authority is at least READ (or GLOBAL when listing discrete profiles). Special parameter authorization: • In order to use the USER operand, one of the following must be true: ◦ SPECIAL, or ◦ AUDITOR, or ◦ The caller is the owner of the user profile, or ◦ The parameter entered for the USER operand is the caller's user ID ◦ The caller has group-SPECIAL or group-AUDITOR authority in a group that owns the user profile. Other: • If the SECLABEL class is active, the caller's current security label must dominate the security label of the general resource profile or data set profile, (unless the high-level qualifier of the data set profile matches the issuer's user ID). • No authorization is required to the user or group profiles that are listed when the UID or GID keyword is specified. • Inactive SECLABEL profiles and profiles that contain inactive security labels may not be listed if SETROPTS SECLBYSYSTEM is active because only users with SPECIAL or AUDITOR authority are allowed to view inactive security labels.
SET	{SM.3::SM.3-R12-RACFEAL5-26} None (can be used an operator command only. RACF allows restricting the use of specific operator commands to defined users).
SETROPTS	{SM.3::SM.3-R12-RACFEAL5-24} General: • SPECIAL, for all operands except: ◦ APPLAUDIT | NOAPPLAUDIT ◦ AUDIT | NOAUDIT ◦ CMDVIOL | NOCMDVIOL ◦ LOGOPTIONS ◦ OPERAUDIT | NOOPERAUDIT

Command	Authorities required for use
	◦ SAUDIT | NOSAUDIT ◦ SECLABELAUDIT | NOSECLABELAUDIT ◦ SECLEVELAUDIT | NOSECLEVELAUDIT • AUDITOR, for use of the following operands: ◦ APPLAUDIT | NOAPPLAUDIT ◦ AUDIT | NOAUDIT ◦ CMDVIOL | NOCMDVIOL ◦ LOGOPTIONS ◦ OPERAUDIT | NOOPERAUDIT ◦ SAUDIT | NOSAUDIT ◦ SECLABELAUDIT | NOSECLABELAUDIT ◦ SECLEVELAUDIT | NOSECLEVELAUDIT Special parameter authorization: • For using the LIST operand: SPECIAL or AUDITOR Other: In some situations, a caller can use SETROPTS even if he does not have the SPECIAL or AUDITOR attributes. These situations are: • The LIST operand can be used if the caller has the group-SPECIAL or group-AUDITOR attribute in the current connect group or (if GRPLIST is active) in any group that the caller is connected to. • The REFRESH operand together with GENERIC can be used if the caller has the group-SPECIAL, AUDITOR, group-AUDITOR, OPERATIONS, group-OPERATIONS attribute, or CLAUTH authority for the classes specified. • The REFRESH operand together with GLOBAL can be used if the caller has the OPERATIONS attribute or CLAUTH authority for the classes specified. • The REFRESH operand together with RACLIST can be used if the caller has CLAUTH authority to the specified class. • The REFRESH operand together with WHEN(PROGRAM) can be used if the caller has CLAUTH authority for the program class or the OPERATIONS attribute.
SIGNOFF	{SM.3::SM.3-R12-RACFEAL5-27} None (can be used an operator command only. RACF allows restricting the use of specific operator commands to defined users).
STOP	{SM.3::SM.3-R12-RACFEAL5-28} None (can be used an operator command only. RACF allows restricting the use of specific operator commands to defined users).
TARGET	{SM.3::SM.3-R12-RACFEAL5-29} None (can be used an operator command

Command	Authorities required for use
	only. RACF allows restricting the use of specific operator commands to defined users).

Table 38: RACF command authorizations

8.4.3.1 Management of z/OS UNIX file system objects and IPC objects

Access permissions to z/OS UNIX file system objects and IPC objects are managed by functions of the RACF callable services. For example the function R_setfacl can be used to manage the access control lists of UNIX file system objects and the function R_IPC_ctl can be used to manage access rights to IPC objects and the function R_chmod can be used to change the permission bits {SM.3::SM.3.35}

An application executing in supervisor state may change the group IDs of a subject representing a user with the R_setegid, R_setgid, or R_exec callable services according to the following rules:

- R_setegid: If the high-order bit of the input GID is on, the real, effective, and saved GIDs are changed for the current process.
- R_setegid: If the high-order bit of the input GID is off and if the user is the superuser or if the input GID is equal to the real or saved GID of the calling process, the effective GID of the process is changed to the input GID. The real and saved GIDs are not changed.
- R_setgid: If the calling process is a superuser, the real, saved, and effective GIDs are changed. If the calling process is not a superuser but the input GID is equal to the real or saved GID, the effective GID of the process is changed. If neither condition is met, the GIDs of the process are not changed.
- R_exec: sets the effective and saved UNIX group identifier to the specified values (if the call requested a change of the GID)
- The new GID must be known to RACF.

An application executing in supervisor state may change the user IDs of a subject representing a user with the R_seteuid, R_setuid, or R_exec callable services according to the following rules.

- R_seteuid: If the high-order bit of the input UID is on, the real, effective, and saved UIDs are changed for the current process.
- R_seteuid: If the high-order bit of the input z/OS UNIX user identifier (UID) is off and if the user is the superuser or if the input UID is equal to the real or saved UID of the calling process, the effective UID of the process is changed to the input UID. The real and saved UIDs are not changed.
- R_setuid: If the calling process is a superuser, the real, saved, and effective z/OS UNIX user identifiers (UIDs) are changed. If the calling process is not a superuser, but the input UID is equal to the real or saved UID, the effective UID of the process is changed. If neither condition is met, the UIDs of the process are not changed.
- R_exec: sets the effective and saved UNIX user identifier to the specified values (if the call requested a change of the UID)..

8.5 Auditing

8.5.1 Generation of audit records

The TOE provides a general facility to collect data required for auditing and accounting services, whic the TOE forwards to a component of z/OS for recording. This component of the TOE environment, the System Management Facilities (SMF), collects and records system and job-related information that an installation can use for such tasks as the following:

- Billing users
- Reporting reliability
- Analyzing the configuration
- Scheduling jobs
- Summarizing direct access volume activity
- Evaluating data set activity
- Profiling system resource use
- Maintaining system security

This component is used by the TOE to store and manage the security-related auditing information the TOE has generated as required by FAU_GEN.1 and FAU_GEN.2.

Each SMF record consists of a standard header which contains (among other information) the type of the record and the time the record was produced {AU.1::AU.1.1}. The standard header is produced by the calls to the SMFWTM or SMFEWTM services or the smf_record UNIX System Services callable service. Especially the time and date are filled in by SMF and not by the caller.

One record type is usually reserved for a whole class of events where the individual events are identified by the record subtype or event code in the header of the SMF record.

RACF as the central access control function has three SMF record types reserved for its use (80, 81, 83), with record type number 80 being the most important one. The information recorded in this record type contains (among other non security related information):

- The record type
- Time stamp (time and date) (filled in by the SMF component of z/OS)
- System identification
- Event code and qualifier
- User identification
- Group name
- Authorities used to successfully execute commands or access resources
- Reasons for logging
- Command processing error flag
- Foreground user terminal ID or other port-of-entry information
- Job log number (job name, entry time, and date)
- RACF version, release, and modification number
- SECLABEL of user (relevant in Labeled Security Mode only)

Each record contains further data specific to the event code and qualifier {AU.1::AU.1.3}.

The administrator can configure RACF and other elements of the TOE to generate audit records for all events listed in Table 6, Auditable Events {AU.1::AU.1-R9-MULTI-1}.

z/OS provides the capability to search the audit trail for specific events and relate them such that events related to a specific user, specific user/job sensitivity label (Labeled Security Mode) or specific object sensitivity label (Labeled Security Mode) can be extracted from the audit trail {AU.1::AU.1.4}.

Tools exist that allow user with access to the audit trail data to search the audit trail for specific events, for audit events related to specific jobs / users and other criteria {AU.1::AU.1.5}. Tools exist that transfer the audit data into human readable format {AU.1::AU.1.6}.

RACF also allows LDAP clients (typically servers outside of the TOE, residing on the network) that have authenticated using an ICTX-style DN to request RACF to generate audit records to record events that have occurred externally to the TOE. The requester provides information about the user involved with the event, the kind of event, and the resource name and resource class name (any class except DATASET) associated with the event.

If an application has created an ACEE and specified ICTX= on the RACROUTE REQUEST=VERIFY to associate a X.500-format distributed identity with the RACF user's ACEE, RACF will include that distributed identity in the SMF records that it creates. {AU.1::AU.1-R12-RACF-1}

8.5.2 Event Notifications generated by RACF

RACF can send an ENF type 62 signal to listeners when a SETROPTS RACLIST command affects in-storage profiles used for authorization checking. RACF sends a signal when a SETROPTS RACLIST, SETROPTS NORACLIST, or SETROPTS RACLIST REFRESH command is issued for a class, activating, deactivating, or updating the profiles. Signals are sent for a class in the static class descriptor table if SIGNAL=YES was specified on the ICHERCDE macro that defined the class. Signals are sent for a class in the dynamic class descriptor table if SIGNAL(YES) was specified on the CDTINFO keyword of the RDEFINE or RALTER command that defined the class.{SM.3::SM.3-R12-RACFEAL5-33}

8.5.3 Audit configuration and management

Within the system configuration it needs to be decided, which SMF records shall be generated by z/OS. Three record types (type 80, 81, and 83) are dedicated to RACF and are the most important ones for security. Which events are actually recorded with those records can be configured by a user with the AUDITOR attribute in his RACF user profile {AU.3::AU.3.1}. In addition record type 30 is generated for a number of security related events.

Because a set of mandatory events is always audited, not all audit records (such as unauthorized attempts to access the system or changes to the status of the RACF database) can be configured.

In addition, resource profiles can define which events related to this resource are audited {AU.3::AU.3.2}. The owner of a resource profile as well as a user in the AUDITOR role are able to change the entries related to auditing within the resource profile {AU.3::AU.3.3}.

The system can be configured to send certain audit messages to the security console to immediately alert operators of detected policy violations {AU.3::AU.3.4}.

RACF uses the interfaces defined by the SMF component of z/OS to have SMF finalize and store the audit records. SMF adds a standard header to the audit record, which contains the time and date the record was produced.

The RACF SMF data unload utility (IRRADU00) can be used to create a sequential file from the security relevant audit events stored in the SMF audit trail. The resulting sequential file can either be viewed directly or used for further processing like with the utility DFSORT (which is part of the TOE environment), which can be used to create reports by selecting specific records and further structure

the reports.

8.6 RACF configuration

Upon installation RACF is configured using

- the RACF data set name table (ICHRDSNT load module), which provides the data set names for the RACF database and for each of them the backup data set.
- the RACF range table (load module ICHRRNG), which determines in which data set of the RACF database RACF places each profile.
- the class descriptor table (CDT), which describes the RACF general resource classes known to RACF. This table has two parts: the static class descriptor table and the dynamic class descriptor table. The static class descriptor table consists of a load module describing the classes supplied by IBM (load module ICHRRCDX) and an optional load module (ICHRRCDE) that contains installation defined classes.

8.7 RACF support for program signing and verification

RACF provides the R_PgmSignVer callable service to support the signing and signature verification of z/OS program objects. The function can be used for both signing a program object and verifying the signature of a program object. The function is intended to be used by the z/OS program binder (for signing program objects) and the z/OS loader (to verify the signature of a program object). Callers of this service need to have sufficient authority to use the key ring, which is either specified in the parameter list or using the APPLDATA field of FACILITY class profile IRR.PROGRAM.SIGNATURE.VERIFICATION) and the private contained within it as determined by the R_datalib RACF callable service. The signature will be generated using SHA256 as the hash function and RSA as the public key encryption algorithm. The maximum RSA key size is 4096 bit.

{SP.4::SP.4-R12-RACF-2} A key ring used for program signing needs to contain all of the following certificate objects:

- An RSA private key to apply the digital signature.
- The X.509 certificate, called a code-signing certificate, that corresponds to the RSA private key.
- Each certificate-authority (CA) certificate (up to and including the root CA certificate) in the certificate chain of the code-signing certificate.

The code-signing certificate and each CA certificate in the chain must be signed using one of the following signature algorithms:

- sha256WithRSAEncryption
- sha1WithRSAEncryption

The code-signing certificate must have code-signing capability in one of the following ways:

- Either the certificate has no KeyUsage extension, or the certificate has a KeyUsage extension with at least the digitalSignature and nonRepudiation indicators enabled.

Each CA certificate in the chain must have certificate-signing capability in both of the following ways:

- Either the certificate has no BasicConstraints extension, or the certificate has a BasicConstraints extension with the cA indicator enabled.
- Either the certificate has no KeyUsage extension, or the certificate has a KeyUsage extension

with at least the keyCertSign indicator enabled.

{SP.4::SP.4-R12-RACF-1}The Security Administrator authorizes users to perform program signing by

(a) creating FACILITY class profiles of the form IRR.PROGRAM.SIGNING.groupname.userid or IRR.PROGRAM.SIGNING.userid or IRR.PROGRAM.SIGNING.groupname or IRR.PROGRAM.SIGNING (listed here in priority order, and where groupname is the user's current connect group) and

(b) providing APPLDATA information in that profile that specifies a hashing algorithm to use and the key ring (both owning userid and key-ring-name) that contains the code signing digital certificate to use, and

(c) authorizing the user to access that key ring.

{SP.4::SP.4-R12-RACF-3}The Security Administrator enables program signature verification by:

- Defining the FACILITY class profile IRR.PROGRAM.SIGNATURE.VERIFICATION and specifying in the APPLDATA the owning user ID and key-ring name of the key ring that holds the CA certificates associated with the various code signing certificates, including the IBM-supplied certificate with the label 'STG Code Signing CA', and
- Defining a PROGRAM profile for IRRPVERS, e.g.,

  RDEFINE PROGRAM IRRPVERS ADDMEM('SYS1.SIEALNKE'//NOPADCHK) UACC(READ) SIGVER(SIGREQUIRED(YES) FAILLOAD(ANYBAD) SIGAUDIT(ANYBAD))
- Defining PROGRAM profiles to protect each signed program that the administrator wants verified during use, and specifying in that profile's SIGVER segment various options that tell RACF how to process the verification for the protected programs:
  - SIGREQUIRED (YES | NO):
    - YES indicates that the program must have a digital signature;
    - NO indicates that it might have one, but is not required to have one.
  - FAILLOAD(ANYBAD | BADSIGONLY | NEVER):
    - ANYBAD indicates that if any failures occur during program signature verification (including administrative setup errors such as missing or incorrectly defined keyrings, signatures by untrusted signers, or incorrect or missing signatures) the system should disallow use of the program.
    - BADSIGONLY indicates that if the signature itself is incorrect or missing the system should disallow use of the program.
    - NEVER (the default) indicates that a problem with signature verification should not prevent use of the program.
  - SIGAUDIT (ALL | SUCCESS | ANYBAD | BADSIGONLY | NONE):
    - ALL indicates that RACF should audit the result of all signature verification operations using an SMF type 80 audit record.
    - SUCCESS indicates that RACF should audit any successful signature verification operations.
    - ANYBAD indicates that RACF should audit any failing signature verification operation.
    - BADSIGONLY indicates that RACF should audit signature verification operations that fail due to an incorrect or missing signature.
    - NONE indicates that RACF should not audit any program verification operations.

- Refreshing the PROGRAM profiles using SETR WHEN(PROGRAM) REFRESH, and running program IRRVERLD. (Note: IRRVERLD must be run at least once per IPL per system.)

8.8 TOE Assurance Measures

The assurance measures provided by the developer to meet the security assurance requirements for the TOE are based on the developer action elements and the requirements on content and presentation of evidence elements defined for the individual assurance requirements in CC Part 3:

SAR	Assurance Measure
ADV_ARC.1	The architecture is described in [ZARCH], which describes the protection functionality provided by the underlying hardware and firmware, in [z/OS Concepts], which describes the concepts of z/OS and in [ABC-V6], which describes how RACF is integrated in z/OS. Further details are described in a dedicated RACF architecture document. The structure of the RACF database is described as part of the TOE design.
ADV_FSP.5	The functional specification of RACF consists of all the interfaces that are callable by external entities. This includes programming interfaces as well as RACF commands and utilities and also includes configuration data RACF reads from storage managed by z/OS (e. g. the RACF related members of SYS1.PARMLIB). Interfaces are described partly in external documents, partly in design documents (for interfaces intended to be used IBM internally only). The description of those interfaces uses a semi-formal style.
ADV_IMP.1	IBM provides access to the source code for the evaluation team in the IBM environment. The implementation representation includes all modules that comprise the TSF.
ADV_INT.2	IBM provides detailed design information showing the internal structure of the entire TSF.
ADV_TDS.4	A high-level design of the security functions of RACF is provided which describes the TOE design at the subsystem level. This document provides an overview of the implementation of the security functions within the subsystems of RACF and points to other existing documents for further details where appropriate. In addition IBM provides dedicated low-level design documentation for the whole TOE. Part of this design documentation has been extracted from the source code, which contains structured module headers with detailed interface and design information for all source code modules of RACF as well as a generalized description of the control flow within each module. A semi-formal description of the subsystems exists. The correspondence information is provided in the form of a spreadsheet showing the correspondence between the functional specification and the TOE design.
AGD_OPE.1	A number of documents exist that provide operational guidance for the user and the system administrators. This includes guides for the management of RACF.

SAR	Assurance Measure
	Especially for the management of RACF a System Administrator Guide and an Auditor Guide exists, that describes and explains in detail the administration commands and parameters.
AGD_PRE.1	Guidance is provided in a number of documents related to the individual components of RACF describing the configuration parameter required to configure the TOE to prepare for a secure operation.
ALC_CMC.4	All configuration management of z/OS source code uses automated CM systems. This has been analyzed as part of the z/OS evaluation and as far as possible the results of the analysis performed for z/OS will be reused.
ALS_CMS.5	RACF is developed at the IBM Poughkeepsie site each using a well defined and highly automated configuration management system. The site has a detailed description of how the configuration management for the z/OS parts maintained at the site is performed. Source code, generated binaries, documentation, test plan, test cases, test results, and development tools are all maintained under configuration management.
ALC_DEL.1	RACF is delivered together with z/OS through sales channels controlled by IBM.
ALC_DVS.1	IBM has a set of guidance documents for physical, logical and procedural security measures that all IBM facilities have to use in their specific implementation of a Security Plan. Each site then has their specific Site Security Plan as a site specific instantiation of those global guidelines. Several sites of IBM (including the site in Poughkeepsie) have been subject to an analysis of the developer security measures in other evaluations. Where possible this evaluation will re-use the results of those evaluations.
ALC_FLR.3	RACF follows the development practices for z/OS and the z/OS Development within IBM has a well-defined system for reporting flaws and tracing the status of the corrective actions for those flaws. In addition, well-defined procedures exist for IBM's z/OS clients to report security problems via the IBM Support Center, and for IBM to distribute security fixes to clients, and clients can register with IBM to receive special notification of security flaws and fixes.
ALC_LCD.1	IBM's Integrated Product Development (IPD) fulfils the requirements for the development life cycle model and the life cycle related processes.
ALC_TAT.2	The tools used in the development process and product generation are documented with their behavior, options and usage assumptions. The developer provides a description of the implementation standards applied.
ATE_COV.2	IBM has detailed test plans to test the functions of z/OS. Those test plans include an analysis of the test coverage, an analysis of the functional interfaces tested and an analysis of the testing against the high level design.
ATE_DPT.3	Testing of internal interfaces is defined and described in the test plan documents and the test case descriptions.
ATE_FUN.1	Testing has been performed on the platforms that are defined in the Security

SAR	Assurance Measure
	Target. Test results are documented such that the tests can be repeated.
ATE_IND.2	All the required resources to perform their own tests will be provided to the evaluation facility to perform their test. The evaluation facility will perform and document the tests they have created and performed as part of the evaluation technical report for testing. Due to the size of the systems the evaluator tests will be performed at the appropriate IBM development sites.
AVA_VAN.4	IBM has its own team that performs vulnerability analysis and penetration testing for z/OS, which also covers RACF. This team has a long term experience with potential security problems within z/OS and RACF and is also integrated in the design reviews. The developer vulnerability analysis will report the activities and findings of this team. The evaluator will perform a methodical vulnerability analysis.

End of document