Security Target for IBM z/OS Version 1 Release 12

Version:	8.4
Status:	Final
Last Update:	2011-06-17
Classification:	None: External Version

Trademarks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

- Advanced Function Presentation
- AFP
- BladeCenter
- DFS
- DFSORT
- eserver
- IBM
- Infoprint
- MVS
- PR/SM
- Print Services Facility
- Processor Resource/Systems Manager
- RACF
- System z
- VTAM
- z/Architecture
- zEnterprise
- 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.

Table of Contents

List of Tables

1 Introduction

1.1 Security Target Identification

Title:	Security Target for IBM z/OS Version 1 Release 12
Version:	8.4
Status:	Final
Date:	2011-06-17
Sponsor:	IBM Corporation
Developer:	IBM Corporation
Certification ID:	BSI-DSZ-CC-0701
Keywords:	access control, discretionary access control, general-purpose operating system, information protection, security labels, mandatory access control, security, UNIX®

1.2 TOE Identification

The TOE is z/OS Version 1 Release 12.

1.3 TOE Overview

This Security Target (ST) documents the security characteristics of the IBM z/OS Version 1 Release 12 operating system with the additional required licensed programs (see section Software configuration of this ST) configured in a secure manner as described in *z/OS Planning for Multilevel Security and the Common Criteria* ([PMLS]).

IBM z/OS, a highly-secure, robust, scalable, high-performance enterprise operating system on which to build and deploy mission-critical applications, provides a comprehensive and diverse application execution environment. IBM z/OS is the flagship operating system for IBM System z™ mainframe computers, empowering the use of their most advanced features, such as the 64-bit z/Architecture™. It delivers the highest qualities of service for enterprise transactions and data and extends these qualities to new applications using the latest software technologies. IBM z/OS serves as the heart of customers' IT infrastructures, helping to integrate their information strategy and business strategy.

IBM z/OS can be used on a single IBM System z mainframe computer, or several systems or logical partitions running the evaluated version of IBM z/OS can be connected to form a loosely-coupled complex of systems called a *sysplex*.

IBM z/OS provides such software technologies as Enterprise Java™ Beans, eXtensible Markup Language (XML), HyperText Markup Language (HTML), Unicode and distributed Internet Protocol (IP) networking. z/OS UNIX System Services allows customers to develop and run UNIX programs on z/OS and exploit the reliability and scalability of the System z processors. z/OS also incorporates cryptographic services, distributed print services, workload management, storage management, parallel sysplex availability, and automation capabilities. Not all of these functions have been analyzed in this evaluation; see section Software configuration for the software configuration of z/OS used in this evaluation. The security functions subject to this evaluation are described in chapter 8 of this document.

With such outstanding security features as multilevel security support, IBM z/OS meets all of the requirements of the Operating System Protection Profile base [OSPP], as well as its extended packages for extended identification and authentication [OSPP-EIA] and labeled security [OSPP-LS].

IBM z/OS provides identification and authentication of users using different authentication mechanisms, both discretionary and mandatory access control to a large number of different objects, a configurable audit functionality, protection of communication services, sophisticated security management functions, preparation of objects for reuse and functionality used internally to protect z/OS from interference and tampering by untrusted users or subjects.

1.4 TOE Description

The Target of Evaluation (TOE) is the z/OS operating system with the software components as described in section Software configuration. z/OS is a general-purpose, multi-user, multi-tasking operating system for enterprise computing systems. Multiple users can use z/OS simultaneously to perform a variety of functions that require controlled, shared access to the information stored on the system.

In this ST, the TOE is seen as one instance of z/OS running on an abstract machine as the sole operating system and exercising full control over this abstract machine. This abstract machine can be provided by one of the following:

- a logical partition provided by a certified version of PR/SM on an IBM System z™ processor (z890, z990, z9™ 109, z9™ BC, z9™ EC, IBM System z10™ Business Class, IBM System z10™ Enterprise Class, or zEnterprise 196).

- a certified version of z/VM® executing in a logical partition provided by PR/SM on one of the above-listed System z™ processors.

If the configuration includes a zEnterprise BladeCenter Extension (zBX), the operating systems running in the zBX are not part of the TOE. They are external systems, connected to z/OS only via the built-in TCP/IP networking facilities included in the zEnterprise System and zBX.

The abstract machine itself is not part of the TOE, rather, it belongs to the TOE environment. Nevertheless the correctness of separation and memory protection mechanisms implemented in the abstract machine is analyzed as part of the evaluation since those functions are crucial for the security of the TOE.

The TOE environment, as part of the System z processor, also includes specific hardware functions that provide support for the cryptographic operations involved in communications security and for the digital signature operations involved with X.509v3 digital certificates.

Multiple instances of the TOE may be connected in a basic sysplex or in a parallel sysplex with the instances sharing their RACF® database.

The platforms selected for the evaluation consist of IBM products that are available when the evaluation has been completed and will remain available for a substantial period of time afterward.

The individual TOEs can be run alone or within a network as a set of cooperating hosts, operating under and implementing the same set of security policies.

Transmission Control Protocol/Internet Protocol (TCP/IP) network services, connections, and communication that occur outside of a sysplex are restricted to one security label; that is, each system regards its peers as single-label hosts. Other network communication is disallowed, with the exception of the Job Entry System 2 (JES2) Network Job Entry (NJE) protocol.

Most of the TOE security functions (TSF) are provided by the z/OS operating system Base Control Program (BCP) and the Resource Access Control Facility (RACF), a z/OS component that is used by different services as the central instance for identification and authentication and for access control decisions. z/OS comes with management functions that allow configuring of the TOE security functions to tailor them to the customer’s needs.

Some elements have been included in the TOE that do not provide security functions. These elements run in authorized mode, so they could compromise the TOE if they do not behave properly. Because these elements are essential for the operation of many customer environments, the inclusion of these elements subjects them to the process of scrutiny during the evaluation and ensures that they may be used by customers without affecting the TOE’s security status.

In its evaluated configuration, the TOE allows two modes of operation: a standard mode meeting all requirements of the Operating System Protection Profile base [OSPP] and its extended package for Extended Identification and Authentication [OSPP-EIA], and a more restrictive mode called Labeled Security Mode, which additionally meets all requirements of the OSPP extended package for Labeled Security [OSPP-LS]. In both modes, the same software elements are used. The two modes have different RACF settings with respect to the use of security labels. All other configuration parameters are identical in the two modes.

Throughout this Security Target, all claims that are valid for the Labeled Security Mode only are marked accordingly. Any claim not marked for Labeled Security Mode applies to both modes.

1.4.1 Intended Method of Use

z/OS provides a general computing environment that allows users to gain controlled access to its resources in different ways:

- online interaction with users through Time Sharing Option Extensions (TSO/E) or z/OS UNIX System Services
- batch processing (JES2)
- services provided by started procedures or tasks
- daemons and servers utilizing z/OS UNIX System Services that provide similar functions as started procedures or tasks but based on UNIX interfaces

These services can be accessed by users local to the computer systems or accessing the systems via network services supported by the evaluated configuration.

All users of the TOE are assigned a unique user identifier (user ID). This user ID, which is used as the basis for access control decisions and for accountability, associates the user with a set of security attributes. In most cases the TOE authenticates the claimed identity of a user before allowing this user to perform any further security-relevant actions. Exceptions to this authentication policy include:

1. Pre-specified identities:
   a. The authorized administrator can specify an identity to be used by server or daemon processes or system address spaces, which may be started either automatically or via system operator commands;
   b. The authorized administrator may configure a trusted HTTP server to access selected data under a specified identity, rather than the identity of the end user making the request. The HTTP server may optionally authenticate the user in this case, or may serve the data to anyone asking for it, if the administrator has determined that such anonymous access is appropriate.
2. Users are allowed to execute programs that accept network connections on ports the user has access to. In this case the untrusted program has no knowledge about the external "user" and cannot perform authentication. The program executes with the rights of the z/OS user that started it, and any data access occurs using this user’s authenticated identity.

The TOE provides mechanisms for both mandatory and discretionary access control. This Security Target describes two modes of operation: one with discretionary access control only and one with both discretionary and mandatory access control where the mandatory access control is fully enabled for all subjects and objects . In commercial environments it is often useful to activate only part of the mandatory access control functions required in this Security Target . While such a mode may be useful for specific environments and the functions used have been evaluated, the claims about information flow control made in this Security Target for the Labeled Security Mode may not hold completely when only part of the mandatory access control functions are configured.

All TOE resources are under the control of the TOE. The TOE mediates the access of subjects to TOE-protected objects. Subjects in the TOE are called *tasks*. Tasks are the active entities that can act on the user’s behalf. Data is stored in named objects. The TOE can associate a set of security attributes with each named resource, which includes the description of the access rights to that object and (in Labeled Security Mode) a security label.

Objects are owned by users, who are assumed to be capable of assigning discretionary access rights to their objects in accordance with the organizational security policies. Ownership of named objects can be transferred under the control of the access control policy. In Labeled Security Mode, security labels are assigned by the TOE, either automatically upon creation of the object or by the trusted system administrator. The security attributes of users, data objects, and objects through which the information is passed are used to determine if information may flow through the system as requested by a user.

Apart from normal users, z/OS recognizes administrative users with special authorizations. These users are trusted to perform system administration and maintenance tasks, which includes configuration of the security policy enforced by the z/OS system and attributes related to it. Authorizations can be delegated to other administrative users by updating their security attributes.

The TOE also recognizes the role of an *auditor*, who uses the auditing system provided by z/OS to monitor the system usage according to the organizational security policies.

The TOE is intended to operate in a networked environment with other instantiations of the TOE as well as other well-behaved client systems operating within the same management domain. All of those systems need to be configured in accordance with a defined common security policy.

1.4.2 Summary of Security Features

The primary security features of the product are:

- identification and authentication
- discretionary access control
- in Labeled Security Mode: mandatory access control and support for security labels (Note that security labels can be used in standard mode, too, if allowed by the security administrator.)
- auditing
- object reuse
- security management
- secure communication
- TSF protection

These primary security features are supported by domain separation and reference mediation, which ensure that the features are always invoked and cannot be bypassed.

1.4.2.1 Identification and authentication

z/OS provides identification and authentication of users by the means of

- an alphanumeric RACF user ID and a system-encrypted password or (for applications that support it) password phrase.

- 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.

- an X.509v3 digital certificate presented to a server application 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.

- a Kerberos™ v5 ticket presented to a server application that supports the Kerberos mechanism, and then mapped by that application through the TOE-provided GSS-API programming services or alternate functions that are also provided by the TOE (specifically the R_ticketServ, and R_GenSec services). These functions 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.

- an LDAP LDBM bind DN (which is mapped to a RACF user ID by information in the LDAP directory) or an LDAP ICTX or SDBM bind DN (which contains a RACF user ID) together with a RACF password or password phrase. The bind processing then passes the derived RACF user ID, and the password/phrase, to RACF to complete the authentication process.

- a digital certificates presented to LDAP over SSL/TLS (LDAP SASL bind with EXTERNAL verification) which must map to a RACF USER ID.

In the evaluated configuration, all human users are assigned a unique user ID. This user ID supports individual accountability. The TOE security functions authenticate the claimed identity of the user by verifying the password/phrase (or other mechanism, as listed above) before allowing the user to perform any actions that require TSF mediation, other than actions that aid an authorized user in gaining access to the TOE.

In some cases of external access to the system, such as the HTTP server, or LDAP server, an installation may decide to define a user ID that is used for access checking of selected resources for users that have not been authenticated. This allows an installation to define resources unauthenticated users may access using that server via an appropriate client program. Users may still authenticate to the server using their user ID and password/phrase (or other authentication mechanism as above) to access additional resources they have been assigned access to.

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

z/OS supports access controls that are capable of enforcing access limitations on individual users and data objects. Discretionary access control (DAC) allows individual users to specify how such resources as direct access storage devices (DASDs), DASD and tape data sets, and tape volumes that are under their control are to be shared.

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.

z/OS provides three DAC mechanisms:

1. The z/OS standard DAC mechanism is used for most traditional (non-UNIX) protected objects.
2. The z/OS UNIX DAC mechanism is used for z/OS UNIX objects (files, directories, etc.)
3. The z/OS LDAP LDBM DAC mechanism is used to protect LDAP objects in the LDAP LDBM back-end data store.

z/OS standard DAC mechanism

Access types that can be granted are NONE, EXECUTE, READ, UPDATE, CONTROL, and ALTER, which form a hierarchical set of increasing access authorities.

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.

Profiles are assigned to a number of resources within z/OS. This Security Target defines the resource types analyzed during the evaluation. RACF profiles are also used to manage and control privileges in z/OS and resources of subsystems that are not part of the evaluated configuration (e. g. DB2, CICS, JES3).

Access rights for subjects to resources can be set by the profile owner and by the system administrator.

The TOE allows access decisions by this mechanism for local applications or remote applications. For local applications the application, or the TOE, uses the RACROUTE programming interface to perform the access check. Remote applications can perform similar access checking via LDAP interfaces, if the z/OS ITDS LDAP server is appropriately configured, by first authenticating (binding) with an ICTX-style identity (DN), and then providing an extended-operation request indicating that the applications wants do perform an access check. LDAP will then invoke the ICTX extended operation processing routine which will check the application's authority to make such a request, and then will process the request if authorized. The request specifies the resource to be checked and the RACF user ID or group name whose access should be checked.

z/OS UNIX DAC mechanism

z/OS implements POSIX-conformant access control 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 file system objects provide either access control 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.

z/OS LDAP DAC mechanism

The z/OS LDAP server supports several back-end data stores as well as plug-ins. Two of the data stores (LDBM, SDBM) and one plug-in (ICTX) can be used in the evaluated configuration. The SDBM back-end allows RACF administration by remote administrators for systems configured in standard operation mode. The ICTX plug-in allows remote servers to issue authorization check or auditing requests to RACF in either standard or Labeled Security Mode. The LDBM back-end allows storage of customer data in either standard or Labeled Security Mode, and this back-end supports a standard LDAP access control mechanism to control which authenticated users can access which data. It also supports the possibility of “public” data, accessed by unauthenticated users, when the administrator has configured this kind of data and access.

1.4.2.3 Mandatory access control and support for security labels

In addition to DAC, z/OS 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.

MAC checks are performed before DAC checks.

Note that security label checking will also occur in standard operation mode, if the administrator has configured security labels and if resources and users have labels assigned to them. The exact effects (e.g., whether write-down can occur) depend on several RACF options, and so the behavior may differ from that imposed by a Labeled Security configuration, which mandates the setting of certain options.

Users with clearance for multiple security classifications can choose their label at login time in TSO and for batch jobs submitted to JES, with appropriate defaults assigned if no labels are chosen. The choice may be restricted by the label assigned to the point of access (the logical or physical device the user has used to authenticate, e. g. the ID of the terminal, the IP address, or the ID of the job entry station).

TCP/IP applications that process user login requests must either be restricted to a single label or must restrict the user label by the label assigned to the point of access.

Specifically for the z/OS LDAP server:

- The LDBM back-end has no mechanisms to perform MAC checking. Instead, each LDAP server must run with a single security label, matching the classification of the data in the LDBM database. TCP/IP processing will then ensure that only users running with that security label will have access to the LDAP data, thus fulfilling the required MAC checking. As needed, customers may configure multiple z/OS LDAP servers, each running with a single security label, and users must connect to the appropriate server that matches their own security label when they want to access the data.

- The SDBM back-end is prohibited in Labeled Security Mode.

- The ICTX back-end does not provide any data access functions, and thus technically does not need to provide MAC checking. However, if the administrator configures ICTX in Labeled Security Mode then TCP/IP will still control an external server's connection to LDAP based on the server's security label, and any remote authorization checking requests will use that security label as part of the decision making process.

1.4.2.4 Auditing

The TOE 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 collected by the System Management Facilities (SMF) into an audit trail, which is protected from unauthorized modification or deletion by the DAC and (in Labeled Security Mode) MAC mechanisms. This audit trail can reside directly in MVS data sets, or in an MVS log stream (which can be automatically off-loaded into MVS data sets), as configured by the administrator.

The system can be configured to halt on exhaustion of audit trail space to prevent audit data loss.

Operators are warned when audit trail space consumption reaches a predefined threshold.

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) as well as for LDAP-based resources.

Remote applications can use an LDAP interface to request that RACF generate an SMF audit record, if the z/OS ITDS LDAP server is appropriately configured, by first authenticating (binding) with an ICTX-style identity (DN) and then providing a extended-operation request indicating that the applications wants do generate an audit record. LDAP will then invoke the ICTX extended operation processing routine, which will check the application's authority to make such a request, and then will process the request if authorized. The request specifies the information to be audited.

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 Object reuse functionality

Reuse of protected objects and of storage is handled by various hardware and software controls, and by administrative practices.

All memory content of non-shared page frames is cleared before making it accessible to other address spaces or data spaces. DASD data sets can be purged during deletion with the RACF ERASE option and tape volumes can be erased on return to the scratch pool. All resources allocated to UNIX objects are cleared before reuse. Other data pools are under strict TOE control and cannot be accessed directly by normal users.

1.4.2.6 Security management

z/OS provides a set of commands and options to adequately manage the TOE's security functions. Additionally, the TOE provides the capability of managing users, groups of users, general resource profiles, and RACF SETROPTS options via the z/OS LDAP server, which can accept LDAP-format requests from a remote administrator and transform them into RACF administrative commands via its SDBM backend processing. The TOE also provides a Java class that allows Java programs to issue commands to manage users and groups. Both the LDAP SDBM and the Java class ultimately create a RACF command and pass it to RACF using a programming interface, and then RACF runs the command using the identity associated with the SDBM session or the Java program. This behaves just the same as when a local administrator issues the command, including all the same security checking and auditing.

The TOE 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 standard operation 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.

1.4.2.7 Communications Security

z/OS provides means of secure communication between systems sharing the same security policy. In Labeled Security Mode, communication within TOE parts coupled into a sysplex can be multilevel, whereas other communication channels are assigned a single security label. In standard operation mode, labels need not to be assigned and evaluated for any communication channel.

z/OS TCP/IP provides the means for associating labels with all IP addresses in the network. In Labeled Security Mode, communication is permitted between any two addresses that have equivalent labels. In Labeled Security Mode, communication between two multilevel addresses requires the explicit labeling of each packet with the sending user's label and is only permitted over XCF links within the sysplex.

z/OS TCP/IP provides the means to define Virtual IP addresses (VIPAs) with specific labels on a multilevel system. z/OS TCP/IP considers the user's label when choosing a source address for communications. z/OS UNIX Systems Services also provides the means to run up to eight instances of the z/OS TCP/IP stack which can each be restricted to a single label. Either of these approaches can be used to ensure that most communications between multilevel systems do not use a multilevel address on both ends and thereby avoid the need for explicit labeling.

In its evaluated configuration, z/OS supports trusted communication channels for TCP/IP connections. The confidentiality and integrity of network connections are assured by Secure Sockets Layer / Transport Layer Security (SSL/TLS) encrypted communication for TCP/IP connections ([SSLV3], [TLSV1], [TLSV1.1]), which can be used explicitly by applications or applied transparently to their communications (AT-TLS) without changing the applications using it (assuming the applications that do not make use of the SSL/TLS capabilities that allow clients to authenticate to the system using a client-supplied X.509 digital certificate. If applications accept client certificates then they do need to have specific SSL/TLS-related processing within the applications.).

In addition to the SSL/TLS connection, z/OS also supports the IP Security (IPSec) protocol with Internet Key Exchange (IKE) as the key exchange method. This is an additional way to set up a trusted channel to another trusted IT product for IP-based connections. z/OS also provides centralized policy management for IPSec policies across multiple z/OS systems in the network. It also provides centralized management for digital certificates, message signing, and message verification for IPSec across multiple z/OS systems in the network.

z/OS also supports Kerberos™ version 5 networking protocols, via the Integrated Security Services Network Authentication Service component, hereafter called z/OS Network Authentication Service These protocols enable both the client and the server to mutually authenticate. This authentication mechanism can be utilized with the GSS-API services provided by the z/OS Network Authentication Service to provide security services to applications. These services enable encrypted communications channels between clients and servers that may reside on the same or on different systems.

z/OS also supports, via the optional add-on product IBM Ported Tools for z/OS, the SSH v2 protocol and the ssh-daemon provided services of ssh (secure shell), scp (secure copy), and sftp (secure ftp) ([SSHV2])

TCP/IP-based communication can be further controlled by the access control function for TCP/IP connections, which allows controlling of the connection establishment based on access to the TCP/IP stack in general, individual network address and individual ports on a per-application or per-user basis.

z/OS provides also a variety of network services, all of which use RACF for identification, authentication, and access control. In the evaluated configuration, terminal services are provided by TN3270, telnet, rlogin, rsh, and rexec. File transfer services are provided by the File Transfer Protocol (FTP), sftp and scp, Web serving functions are provided by the z/OS HTTP Server.

1.4.2.8 TSF protection

TSF protection is based on several protection mechanisms that are provided by the underlying abstract machine:

- 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

The TOE’s 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 the system, 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. The z/OS Base Control Program mediates all access to the TOE’s hardware resources themselves, other than program-visible CPU instruction functions.

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, the TOE 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.

1.4.3 Configurations

1.4.3.1 Software configuration

The Target of Evaluation, z/OS Version 1 Release 12, consists of:

- z/OS Version 1 Release 12 (V1R12) Common Criteria Evaluated Base Package:
    - z/OS Version 1 Release 12 (z/OS V1R12, program number 5694-A01),
    - Overlay Generation Language Version 1 (OGL V1R1, program number 5688-191)
    - IBM Print Services Facility™ Version 4 Release 3 for z/OS (PSF V4.3.0, program number 5655-M32)
- IBM Ported Tools for z/OS V1.2 (FMID HOS1120, program number 5655-M23, optional)
- The following APARs (or their associated PTFs):

APAR	PTF
PM21052	UK60305
OA33991	UA56370
OA34611	UA57861
OA33063	UA55172
OA33032	UA55781
OA33725	UA56195
OA33835	UA56629
OA31675	UA57144
OA34521	UA57354
OA32012	UA55967
OA34025	UA57068

-

Note: References in this document to “HTTP Server” refer to the IBM HTTP Server Base (FMID HIMW530) and IBM HTTP Server North America Secure (JIMW531) that ship as part of z/OS, and not to the IBM Ported Tools for z/OS HTTP Server, which must not be used in the evaluated configuration.

The same software elements are used in the Labeled Security and standard modes of operation, except as otherwise noted. The mode of operation is defined by the configuration of the labeling-related options in RACF. Details are described in *z/OS Planning for Multilevel Security and the Common Criteria* ([PMLS]).

The z/OS V1R12 Common Criteria Evaluated Base package, and (if used) IBM Ported Tools for z/OS 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]).

Installations may choose not to use any of the elements delivered within the ServerPac, but are required to install, configure, and use at least the RACF component of the z/OS Security Server element.

In addition, any software outside the TOE may be added without affecting the security characteristics of the system, if it cannot run:

- in supervisor state
- as APF-authorized
- with keys 0 through 7
- with UID(0),
- with authority to FACILITY resources BPX.DAEMON, BPX.SERVER, or BPX.SUPERUSER
- with authority to UNIXPRIV resources

This explicitly excludes:

- replacement of any element in the ServerPac providing security functions relevant to this evaluation by other third-party products;
- installing system exits that run authorized (supervisor state, system key, or APF-authorized), with the exception of the sample ICHPWX11 and its associated IRRPHREX routine;
- installing IBM Tivoli Directory Server plug-ins that have not been evaluated;
- using the Authorized Caller Table (ICHAUTAB) in RACF to allow unauthorized programs to issue RACROUTE REQUEST=VERIFY (RACINIT) or RACROUTE REQUEST=LIST (RACLIST).

Note: The evaluated software configuration is not invalidated by installing and operating other appropriately-certified components that possibly run authorized. However the evaluation of those components must show that the component and the security policies implemented by the component do not undermine the security policies described in this document.

The IBM Tivoli Directory Server for z/OS (FMID HRSL3C0) component may be used as the LDAP server, but:

- For client authentication via digital certificates the administrator must configure the LDAP server to map the certificate to a RACF user ID and to fail the bind if the certificate does not map to a RACF user ID. The allowable LDAP configuration provides three options for forming an LDBM subject:
- LDAP may use the original DN from the certificate; or
- LDAP may replace the original DN with an SDBM-format DN based on the RACF user ID; or LDAP may add the SDBM-format DN to the LDAP subject, giving a subject with two DNs, either of which will work in LDAP ACLs.
- client authentication using the Kerberos mechanism has not been evaluated for LDAP and cannot be used in the evaluated configuration.
- authentication via passwords stored in LDAP cannot be used. Authentication must occur using RACF passwords or password phrases. Note that if an LDBM bind DN is specified when binding to the server, the password/phrase specified must be for the RACF user ID associated with that bind DN by the LDAP administrator.;
- only the LDBM back-end and the ICTX plug-in may be used in Labeled Security Mode. In standard mode the LDBM and SDBM back-ends and the ICTX plug-in may be used. Other LDAP back-end configurations and plug-ins have not been evaluated and must not be used.
- (Labeled Security Mode only) Each running instance of the LDAP server must run with a single, non-SYSMULTI, non-SYSNONE, security label. Multiple server instances may run at the same time, with the same or different security labels.

Each running instance of the HTTP server must run with a security label that is neither SYSMULTI nor SYSNONE.

sshd (from IBM Ported Tools for z/OS), may be used, but if used:

- must be configured to use protocol version 2 and either TDES or one of the AES-based encryption suites,
- must be configured in privilege separation mode, and
- must be configured to allow only password-based (including password phrase) authentication of users or public-key based authentication of users with the public keys stored in RACF keyrings. Rhost-based and public-key based user authentication with the keys stored elsewhere may not be used in the evaluated configuration. In Labeled Security Mode sshd should be configured with the SYSMULTI security label.

The Network Authentication Service component of the Integrated Security Services component, if used, and applications exploiting it, must satisfy the following constraints:

- the Network Authentication Service must use the SAF (RACF) registry. The NDBM registry is not a valid configuration for this evaluation.

- Cross Realm Trust relationships with foreign Kerberos realms is allowed, but the foreign KDC must be capable of supporting the same cipher as does the z/OS KDC.

- In order to ensure strong cryptographic protection of Kerberos tickets, Triple DES or AES should be utilized by the z/OS KDC and any KDC participating in a cross-realm trust relationship with the z/OS KDC. DES should only be used in network environments where the threat of cryptographic attacks against the tickets and Kerberos-protected sessions is deemed low enough to justify the use of these weaker encryption protocols.

- Applications supporting Kerberos may use a combination of application specific protocols and the GSS-API functions or the equivalent native platform callable services (the SAF R_TicketServ and R_GenSec callable services) to authenticate clients, and in client-server authentication. Only the Kerberos mechanism may be used by applications that utilize GSS-API or the equivalent native platform functions. The GSS-API and R_GenSec services also enable the encryption of sensitive application messages passed via application specific protocols. These services enable the secure communication between client and server applications. The GSSAPI services include the message integrity and privacy functions that validate the authenticity and secure the communications between clients and servers.

The Network File System (NFS) Server may be used, but must be configured with the SAF or SAFEXPORT option, to ensure that all file and directory access (except possibly directory mounting) has appropriate RACF security checks made.

SSL (Secure Sockets Layer) processing, if used, must use SSLv3 protocols. SSL and TLS (Transport Layer Security), if used, must use use either TDES (168-bit keys) or AES (128- or 256-bit keys) encryption.

Any application performing client authentication using client digital certificates over SSL or TLS must be configured to use RACF profiles in the RACDCERT or DIGTRING classes or PKCS#11 tokens in ICSF 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 use of gskkyman for this purpose is not part of the evaluated configuration.

Any client that is delivered with the product that executes with the user's privileges must be used with care, since the TSF can not protect those clients from potentially hostile programs. Passwords/phrases a user enters into those client programs that those clients use to pass to the corresponding server to authenticate the user may potentially be spoofed by hostile programs running in the user's address space. This includes client programs for telnet, TN3270, ftp, r-commands, ssh, all LDAP utilities and Kerberos administration utilities that require the user to enter his password/phrase. When using those client programs the user should take care that no untrusted potentially hostile program has been called during his session.

The following elements and element components cannot be used in an evaluated system, either because they violate the security policies stated in this Security Target or because they have been removed from the evaluated configuration due to time and resource constraints of the evaluation. As they are part of the base system, either they must be not configured for use or they must be deactivated, as described in Chapter 7, “The evaluated configuration for the Common Criteria” in
z/OS Planning for Multilevel Security and the Common Criteria:

- All Bulk Data Transfer (BDT) elements: BDT, BDT File-to-File , and BDT Systems Network Architecture (SNA) NJE
- Connection Manager
- The Distributed Computing Environment (DCE) component (FMID HRSS190) of the Integrated Security Services element
- DCE Base Services (FMID HMB3190)
- The DFS™ Server Message Block (SMB) and DFS DCE-DFS (FMID H0H23B0) components of the Distributed File Service element
- The Enterprise Identity Mapping component of the Integrated Security Services element
- Infoprint® Server
- JES3
- The Advanced Program-to-Program Communication / Multiple Virtual Storage (APPC/MVS) component of the BCP
- Process Manager component from the UNIX System Services Element

The use of TCP/IP communication for JES2 NJE has not been part of the evaluation and must not be used in the evaluated configuration.

The JES2 Execution Batch Monitor (XBM) facility has not been part of the evaluation and must not be used in the evaluated configuration.

The RACF Remote Sharing Facility has not been part of the evaluation and must not be used in the evaluated configuration.

The Data Facility Storage Management Subsystem (DFSMS) Object Access Method for content management type applications must not be used.

The IBM Ported Tools for z/OS HTTP Server V7.0 (FMID HHAP700) must not be used.

For the Communications Server:

- The z/OS FTP server and client, and the z/OS TN3270 server, support both manually-configured SSL/TLS, or AT-TLS. This evaluation has considered only AT-TLS configurations, and as a result manual configuration of those components to use SSL or TLS is not allowed for evaluated configurations.

- The z/OS FTP server and client can support either the protocols from the draft standard for securing FTP with TLS/SSL, or the protocols from the formal RFC 4217 level of Security FTP with TLS/SSL [RFC4217]. This evaluation has considered only the formal RFC 4217 level of support, and as a result that option must be used in the evaluated configuration.

- The following applications must not be used in Labeled Security configurations, as noted in the Communications Server IP Configuration Guide: HOMETEST command, IUCV, LPD, LPQ command, LPR command, LPRM command, LPRSET command, NCPROUTE, NPF, Portmapper, SMTP, SNMP NetView client, TELNET client command, TESTSITE command, TNF, VMCF, z/OS UNIX Network SLAPM2 subagent, z/OS UNIX OMPROUTE SNMP subagent, z/OS UNIX popper, z/OS UNIX RSVP agent, z/OS UNIX SNMP client command, z/OS UNIX SNMP server and agent, z/OS UNIX Trap Forwarder Daemon.

1.4.3.2 Hardware configuration

The following assumptions about the technical environment in which the TOE is intended to be used are made:

The TOE is running within a logical partition provided by a certified version of PR/SM, on the z/Architecture as implemented by the following hardware platforms:

- IBM zSeries model z890, optionally with CryptoExpress2 card or PCIXCC and PCICA crypto cards

- IBM zSeries model z990, optionally with CryptoExpress2 card or PCIXCC and PCICA crypto cards

- IBM System z9 109, z9 BC, or z9 EC, optionally with CryptoExpress2 card.

- IBM System z10 Business Class, optionally with CryptoExpress2 or CryptoExpress3 card.

- IBM System z10 Enterprise Class, optionally with CryptoExpress2 or CryptoExpress3 card.

- IBM zEnterprise 196, optionally with CryptoExpress3 card, and with or without the zEnterprise BladeCenter Extension (zBX).

In addition, the TOE may run on a virtual machine provided by a certified version of z/VM.

The following peripherals can be used with the TOE, while still preserving the security functionality:

- All terminals that are supported by the TOE.

- Printers:
  - in standard operation mode: any printer that is supported by the TOE.
  - in Labeled Security Mode: any printer that is used to print output with different security labels must support the Guaranteed Print Labeling Function. Guaranteed print labeling works with a subset of Advanced Function Presentation™ (AFP™) printers and ensures the integrity of the identification label by preventing the user from changing the label. Review the printer hardware documentation or contact the printer vendor to determine if a printer supports this function.
- All storage devices and backup devices supported by the TOE, such as:
  - Direct access storage devices (DASDs), except RVA devices.
  - Tape drives (including encrypting tape drives, though this evaluation has not specifically examined those cryptographic functions).
- All Ethernet and token-ring network adapters that are supported by the TOE.

Note: The peripherals may be virtualized in the case of the TOE executing within a logical partition or z/VM. The logical partitioning software and z/VM software is part of the abstract machine and therefore part of the TOE environment. The logical partitioning software documentation as well as the z/VM documentation provides the required guidance on how to set up and configure the logical partitioning software or z/VM and how to define the logical peripheral devices so the TOE operates securely in the logical partitioning or z/VM environment.

1.4.4 Structure

The structure of this document is as defined by [CC] Part 1 :

- Chapter 1 provides the ST Introduction
- Chapter 2 provides the CC Conformance Claim
- Chapter 3 provides the Security Problem Definition
- Chapter 4 provides the Security Objectives
- Chapter 5 provides the Extended Components Definition
- Chapter 6 provides the Security Requirements for the Operational Environment
- Chapter 7 provides the Security Requirements
- Chapter 8 provides the TOE Summary Specification

- Chapter 9 provides Abbreviations, Terminology and References

2 CC Conformance Claim

This ST is CC Part 2 extended and CC Part 3 conformant, with a claimed Evaluation Assurance Level of EAL4, augmented by ALC_FLR.3.

This ST claims conformance to the following Protection Profiles:

- [OSPP] Operating System Protection Profile. Version 2.0 as of 2010-06-01; strict conformance.

- [OSPP-EIA]: OSPP Extended Package - Extended Identification and Authentication. Version 2.0 as of 2010-05-28; strict conformance.

- [OSPP-LS]: OSPP Extended Package - Labeled Security. Version 2.0 as of 2010-05-28; strict conformance.

Common Criteria [CC] and Common Evaluation Methodology [CEM] version 3.1 revision 3 have been taken as the basis for this conformance claim.

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:

- Persistent storage objects used to store user data and/or TSF data, where this data needs to be protected from any of the following operations:
    - Unauthorized read access
    - Unauthorized modification
    - Unauthorized deletion of the object
    - Unauthorized creation of new objects
    - Unauthorized management of object attributes
- Transient storage objects, including network data
- TSF functions and associated TSF data
- 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 an enhanced-basic attack potential.

The following threats are addressed by the TOE. All threats have been copied from the OSPP. As in the OSPP, there are no threats and policies to justify the assurance level.

3.2.3 Threats countered by the TOE

T.ACCESS.TSFDATA

A threat agent might read or modify TSF data without the necessary authorization when the data is stored or transmitted.

T.ACCESS.USERDATA

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

T.ACCESS.TSFFUNC

A threat agent might use or modify functionality of the TSF without the necessary privilege to grant itself or others unauthorized access to TSF data or user data.

T.ACCESS.COMM

A threat agent might access a communication channel that establishes a trust relationship between the TOE and another remote trusted IT system or masquerade as another remote trusted IT system.

T.RESTRICT.NETTRAFFIC

A threat agent might get access to information or transmit information to other recipients via network communication channels without authorization for this communication attempt by the information flow control policy.

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.

T.DATA_NOT_SEPARATED

The TOE may not adequately separate data on the basis of its sensitivity label, thereby allowing information to flow illicitly from or to users.

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 a cooperative non-hostile 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

3.3.1.1 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.

3.3.1.2 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.

3.3.1.3 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.

A.PEER.MGT

All remote trusted IT systems trusted by the TSF to provide TSF data or services to the TOE, or to support the TSF in the enforcement of security policy decisions are assumed to be under the same management control and operate under security policy constraints compatible with those of the TOE.

A.PEER.FUNC

All remote trusted IT systems trusted by the TSF to provide TSF data or services to the TOE, or to support the TSF in the enforcement of security policy decisions are assumed to correctly implement the functionality used by the TSF consistent with the assumptions defined for this functionality.

3.3.1.4 Connectivity

A.CONNECT

All connections to and from remote trusted IT systems and between physically-separate parts of the TSF not protected by the TSF itself are physically or logically protected within the TOE environment to ensure the integrity and confidentiality of the data transmitted and to ensure the authenticity of the communication end points.

Application note: If the TOE consists of separate parts and the TOE implements mechanisms ensuring the protection TSF data in transit between these parts, the ST author may consider claiming FPT_ITT.1 to supplement or replace A.CONNECT.

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.I&A.REMOTE

Remote trusted IT systems shall be able to obtain identification and authentication decisions from the TOE based on credentials transmitted by a remote trusted IT system to the TOE.

P.CLEARANCE

The system must limit the information flow between protected resources and authorized users based on whether the user's sensitivity label is appropriate for the labeled information.

P.LABELED_OUTPUT

The beginning and end of all paged, hardcopy output must be marked with sensitivity labels that properly represent the sensitivity label of the output.

P.RESOURCE_LABELS

All resources accessible by subjects and all subjects must have associated labels identifying the sensitivity levels of data contained therein.

P.USER_CLEARANCE

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, categorized as either IT security objectives or non-IT security objectives, reflect the stated intent to counter identified threats, comply with any organizational security policies identified, or both. All of the identified threats and organizational policies are addressed under one of the following categories.

4.1 Objectives for the TOE

O.AUDITING

The TSF must be able to record defined security-relevant events (which usually include security-critical actions of users of the TOE). The TSF must protect this information and present it to authorized users if the audit trail is stored on the local system. The information recorded for security-relevant events must contain the time and date the event happened and, if possible, the identification of the user that caused the event, and must be in sufficient detail to help the authorized user detect attempted security violations or potential misconfiguration of the TOE security features that would leave the IT assets open to compromise.

O.CRYPTO.NET

The TSF must allow authorized users to remotely access the TOE using a cryptographically-protected network protocol that ensures integrity and confidentiality of the transported data and is able to authenticate the end points of the communication. Note that the same protocols may also be used in the case where the TSF is physically separated into multiple parts that must communicate securely with each other over untrusted network connections.

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.NETWORK.FLOW

The TOE shall mediate communication between sets of TOE network interfaces, between a network interface and the TOE itself, and between subjects in the TOE and the TOE itself in accordance with its security policy.

O.SUBJECT.COM

The TOE shall mediate communication between subjects acting with different subject security attributes in accordance with its security policy.

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.TRUSTED_CHANNEL

The TSF must be designed and implemented in a manner that allows for establishing a trusted channel between the TOE and a remote trusted IT system that protects the user data and TSF data transferred over this channel from disclosure and undetected modification and prevents masquerading of the remote trusted IT system.

O.I&A.REMOTE

The TOE shall allow remote trusted IT systems to transmit user credentials to the TOE which are used to perform a local identification and authentication policy decision. This decision is communicated back to one or more remote trusted IT systems based on the identification and authentication policy.

O.I&A.MULTIPLE

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

O.LS.CONFIDENTIALITY

The TOE will control information flow between entities and resources based upon the sensitivity labels of users and resources.

O.LS.PRINT

The TOE will provide the capability to mark printed output with accurate labels based on the sensitivity label of the user causing the output.

O.LS.LABEL

The TOE will provide the capability to label all subjects, and all objects accessible by subjects, to restrict information flow based on the sensitivity labels.

4.2 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.REMOTE

If the TOE relies on remote trusted IT systems to support the enforcement of its policy, those systems provide the functions required by the TOE and are sufficiently protected from any attack that may cause those functions to provide false results.

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:

- All network and peripheral cabling must be approved for the transmittal of the most sensitive data held by the system. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted.
- DAC protections on security-relevant files (such as audit trails and authentication databases) shall always be set up correctly.
- Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data.

OE.INSTALL

Those responsible for the TOE must establish and implement procedures to ensure that the hardware, software and firmware components that comprise the system 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.TRUSTED.IT.SYSTEM

The remote trusted IT systems implement the protocols and mechanisms required by the TSF to support the enforcement of the security policy.

These remote trusted IT systems are under the same management domain as the TOE, are managed based on the same rules and policies applicable to the TOE, and are physically and logically protected equivalent to the TOE.

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.CRYPTO.NET	T.ACCESS.TSFDATA T.ACCESS.USERDATA T.ACCESS.TSFFUNC
O.DISCRETIONARY.ACCESS	T.ACCESS.TSFDATA T.ACCESS.USERDATA
O.NETWORK.FLOW	T.RESTRICT.NETTRAFFIC
O.SUBJECT.COM	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.TRUSTED_CHANNEL	T.ACCESS.COMM
O.I&A.REMOTE	P.I&A.REMOTE
O.I&A.MULTIPLE	P.I&A.REMOTE
O.LS.CONFIDENTIALITY	T.DATA_NOT_SEPARATED P.CLEARANCE P.USER_CLEARANCE
O.LS.PRINT	P.LABELED_OUTPUT

Objective	Threats / OSPs
O.LS.LABEL	P.RESOURCE_LABELS P.USER_CLEARANCE

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.REMOTE	A.CONNECT T.ACCESS.COMM
OE.INFO_PROTECT	A.PHYSICAL 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 A.DETECT

Objective	Assumptions / Threats / OSPs
OE.TRUSTED.IT.SYSTEM	A.PEER.MGT A.PEER.FUNC A.CONNECT

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.CRYPTO.NET requiring cryptographically-protected communication channels for data including TSF data controlled by the TOE in transit between trusted IT systems, • O.DISCRETIONARY.ACCESS requiring that data, including TSF data stored with the TOE, have discretionary access control protection, • O.SUBJECT.COM requiring the TSF to mediate communication between subjects.
T.ACCESS.USERDATA	The threat of accessing user data without proper authorization is removed by: • O.CRYPTO.NET requiring cryptographically-protected communication channels for data including user data controlled by the TOE in transit between trusted IT systems, • O.DISCRETIONARY.ACCESS requiring that data including user data stored with the TOE, have discretionary access control protection, • O.SUBJECT.COM requiring the TSF to mediate

Threat	Rationale for security objectives
	communication between subjects. • O.CRYPTO.BASIC requiring the TSF to provide cryptographic services for general use by authorized entities, including encryption, decryption, and message digest generation services.
T.ACCESS.TSFFUNC	The threat of accessing TSF functions without proper authorization is removed by: • O.CRYPTO.NET requiring cryptographically-protected communication channels to limit which TSF functions are accessible to external entities, • O.MANAGE requiring that only authorized users utilize management TSF functions.
T.ACCESS.COMM	The threat of accessing a communication channel that establishes a trust relationship between the TOE and another remote trusted IT system is removed by: • O.TRUSTED_CHANNEL requiring that the TOE implements a trusted channel between itself and a remote trusted IT system protecting the user data and TSF data transferred over this channel from disclosure and undetected modification, • OE.REMOTE requiring that those systems providing the functions required by the TOE are sufficiently protected from any attack that may cause those functions to provide false results.
T.RESTRICT.NETTRAFFIC	The threat of accessing information or transmitting information to other recipients via network communication channels without authorization for this communication attempt is removed by: • O.NETWORK.FLOW requiring the TOE to mediate the communication between itself and remote entities in accordance with its security policy.
T.IA.MASQUERADE	The threat of masquerading as an authorized entity in order to gain unauthorized access to user data, TSF data or

Threat	Rationale for security objectives
	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.
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.
T.DATA_NOT_SEPARATED	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 control information flow between entities and resources based upon the sensitivity labels of users and 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.INFO_PROTECT requiring the approval of network and peripheral cabling, • 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: • O.I_A ensuring that users have been successfully authenticated before allowing any action the TOE has defined to provide to authenticated users only, • 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

Assumption	**Rationale for security objectives**
	TOE, • O.MANAGE requiring the TSF to ensure that only authorized users are able to access such functionality, • 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.
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, • OE.RECOVER requiring an administrative user to 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.
A.PEER.MGT	The assumption on all remote trusted IT systems to be under the same management control and operate under security policy constraints compatible with those of the TOE is covered by: • OE.TRUSTED.IT.SYSTEM requiring that these remote trusted IT systems are under the same management domain as the TOE, and are managed based on the same rules and policies applicable to the TOE.
A.PEER.FUNC	The assumption on all remote trusted IT systems to correctly implement the functionality used by the TSF consistent with the assumptions defined for this functionality is covered by: • OE.TRUSTED.IT.SYSTEM requiring that the remote trusted IT systems implement the protocols and mechanisms required

Assumption	Rationale for security objectives
	by the TSF to support the enforcement of the security policy.
A.CONNECT	The assumption on all connections to and from remote trusted IT systems and between physically separate parts of the TSF not protected by the TSF itself are physically or logically protected is covered by: • OE.REMOTE requiring that remote trusted IT systems provide the functions required by the TOE and are sufficiently protected from any attack that may cause those functions to provide false results, • OE.TRUSTED.IT.SYSTEM demanding the physical and logical protection equivalent to the TOE.

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: • 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.I&A.REMOTE	The policy to obtain identification and authentication decisions from the TOE based on credentials transmitted by a

OSP	Rationale for security objectives
	remote trusted IT system to the TOE is implemented by: • O.I&A.REMOTE allowing remote trusted IT systems to transmit user credentials to the TOE which are used to perform a local identification and authentication policy decision. This decision is communicated back to one or more remote trusted IT systems based on the identification and authentication policy. • O.I&A.MULTIPLE allowing the concurrent use of multiple identification and authentication mechanisms implementing the identification and authentication policy.
P.CLEARANCE	The policy to limit the information flow between protected resources and authorized users based on the user's sensitivity label being appropriate for the labeled information 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.
P.LABELED_OUTPUT	The policy to provide the capability to mark printed output with accurate labels based on the sensitivity label of the user causing the output is implemented by: • O.LS.PRINT providing the capability to mark printed output with accurate labels based on the sensitivity label of the user causing the output.
P.RESOURCE_LABELS	The policy that resources accessible by subjects and all subjects must have associated labels identifying the sensitivity levels of data contained therein 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:

OSP	Rationale for security objectives
	• 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 object and subject are accurately labeled for the TOE to enforce the label policy.

Table 5: Sufficiency of objectives enforcing Organizational Security Policies

5 Extended Components Definition

[OSPP] defines three extended components:

- FCS_RNG.1: Random number generation
- FDP_RIP.3: Full residual information protection of subjects , and
- FIA_USB.2: Enhanced user-subject binding.

[OSPP-EIA] defines two extended components:

- FIA_UAU.8: Authentication policy decisions, and
- FIA_UID.3: Identification policy decisions.

This Security Target does not define any additional extended components.

6 Security Requirements for the Operational Environment

Although CC Version 3.1 does not mandate the use of security requirements for the IT environment, it allows to define the security objectives for the IT environment to the level of detail useful for the understanding and evaluation of a TOE. In the case of z/OS the security functionality defined in chapter 7 of this Security Target depends on the supporting functionality defined in this section. The authors of this Security Target decided (also for compatibility with Security Targets used for previous versions of the TOE) to define this functionality using the structure of Security Functional Requirements.

There are several components in the IT environment that are used by the TOE to implement the security functional requirements. Those are:

- The instructions provided by the underlying processor (named z/Architecture)
- The “CP Assist for Cryptographic Functions” (CPACF). Although this feature is implemented as instructions of the processor and therefore is part of the z/Architecture, it has been decided by the authors of this Security Target to treat them separate from the other instructions. One reason is that some features of CPACF are available on selected processor types only. This is expressed in the SFRs related to CPACF.
- The PCIXCC, a PCI board with its own processor and cryptographic coprocessors. This board provides a set of cryptographic functions broader than the CPACF. The PCIXCC coprocessor provides a separate, physically protected environment to store cryptographic keys and perform cryptographic operations. This coprocessor is

optional. The ICSF component of the TOE checks for the availability of one or more of those boards.

- The PCICA, PCI board that provides functions for fast long integer arithmetic that can be used for fast implementation of asymmetric cryptographic algorithms like RSA and DSA.
- The “CryptoExpress 2” (CEX2) coprocessor board. This board can be operated in two modes:
    - a coprocessor mode (CEX2C), where it is functionally equivalent to the PCIXCC
    - an accelerator mode (CEX2A), where it is functionally equivalent to the PCICA

Note: *The latest versions of the z10 and z196 processors provide the CryptoExpress 3 (CEX3) coprocessor instead of the CEX2 coprocessor. In this Security Target, references to CryptoExpress2 or CEX2(CEX2A, CEX2C, etc.) apply to both CryptoExpress2 cards and CryptoExpress3 cards. References specifically to CryptoExpress3 or CEX3 (if any) apply only to CryptoExpress3 cards and not to CryptoExpress2 cards.*

The PCICA, PCIXCC and CEX2 coprocessors are used when they are installed in a way transparent to the user when he uses the ICSF component of the TOE. ICSF scans for the available cryptographic coprocessors and uses them accordingly. The security functional requirements listed here are related to the use of those coprocessors by the functions claimed in this Security Target that rely on cryptographic operations. While the coprocessors may implement more cryptographic functions than those claimed here, those are not used to support any of the claims made in chapter 7.1 of this Security Target.

While the functions of the coprocessors can only be called using ICSF, the processor instructions implemented by the CPACF are available for all programs. The claims made in this section are only for the use of those functions by the TSF. While this checks for the correct implementation of the basic cryptographic algorithms for those instructions, no claim can be made here for applications not part of the TSF that use those instructions. They may still use those instructions incorrectly or fail to protect cryptographic keys appropriately.

The other part of the IT environment where requirements are stated is the underlying abstract machine as implemented by the z/Architecture that has to provide the mechanism to protect the TSF and TSF data from unauthorized access and tampering. This is expressed with the following security functional requirement for the processor used to execute TOE software:

6.1 General security requirements for the abstract machine

6.1.1 Subset access control (FDP_ACC.1(E))

FDP_ACC.1.1 The abstract machine shall enforce the memory access control policy on instructions as subjects and memory locations and processor registers as objects.

6.1.2 Security-attribute-based access control (FDP_ACF.1(E))

FDP_ACF.1.1 The abstract machine shall enforce the memory access control policy to objects based on the processor state (problem or supervisor).

FDP_ACF.1.2 The abstract machine shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: access to memory locations and special registers is based on the processor state and the state of the memory management unit. Access to dedicated processor registers is allowed only if the processor is in supervisor state when the instruction accessing the register is executed.

FDP_ACF.1.3 The abstract machine shall explicitly authorize access of subjects to objects based on the following additional rules: some dedicated processor registers may be read but not modified when the instruction accessing the register is in problem mode.

FDP_ACF.1.4 The abstract machine shall explicitly deny access of subjects to objects based on the following rule: none.

Application note: *The precise definition of the objects and the rules for the access control policy differ slightly depending on the processor type. Although the underlying hardware / firmware that enforces this policy is part of the IT environment, it is analyzed and tested to provide the support required for the enforcement of the TOE's self-protection. The criteria for the analysis of the high-level design require the analysis of the underlying hardware and firmware and the security functional requirements stated here are taken as the basis for this analysis.*

6.1.3 Static attribute initialization (FMT_MSA.3(E))

FMT_MSA.3.1 The abstract machine shall enforce the memory access control policy to provide permissive default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The abstract machine shall allow the no role to specify alternative initial values to override the default values when an object or information is created.

Application note: *The “default” values in this case are seen as the values the processor has after startup. They have to be “permissive”, because the initialization routine needs to set up the memory management unit and the device register. With respect to the hardware, there is no “role” model implemented, but the access control policy is purely based on a single attribute (“user” or “supervisor” state) that can not be managed or assigned to a “user”. The attribute changes under well-defined conditions (when the* *processor encounters an exception an interrupt, or when a call gate for a higher ring of privilege is called). The security requirement FMT_MSA.1 was therefore not applicable because the security attribute cannot be "managed". For this reason, there is also no security requirement FMT_SMR.1 included, because there are no "roles" that need to be managed or assigned to "users". The dependency of FMT_MSA.3 to FMT_MSA.1 and FMT_SMR.1 is therefore unresolved.*

6.2 Security requirements for CPACF

The CP assist for cryptographic functions (CPACF) is a feature of the z/Architecture that provides instructions to perform cryptographic operations. Those instructions are part of the general instruction set of the processor and available to programs executing in any mode and with any PSW key. The instructions provide support for the basic cryptographic operations only. No support for key management, key protection or key generation is provided. This has to be performed by the software using the instructions. The instructions are specified in [ZARCH].

6.2.1 Cryptographic operation (DES) (FCS_COP.1(1E))

FCS_COP.1.1 The CPACF shall perform encryption and decryption in accordance with a specified cryptographic algorithm TDES and cryptographic key sizes 112 and 168 bit that meet the following: FIPS 46-3.

Application note: *This function is provided by the "Cipher Message" and "Cipher Message with Chaining" instructions. Function Code 1 specifies DES, function code 2 specifies two key TDES and function code 3 specifies 3 key TDES. The z890, z990, and later processors implement this function.*

6.2.2 Cryptographic operation (AES) (FCS_COP.1(2E))

FCS_COP.1.1 The CPACF shall perform encryption and decryption in accordance with a specified cryptographic algorithm AES and cryptographic key sizes 128 or 256 bit that meet the following: FIPS 197.

Application note: *This function is provided by the "Cipher Message" and "Cipher Message with Chaining" instructions. Function Code 18 specifies AES. This function is only implemented by the z9 and later processors. A z10 processor is required for 256 bit keys.*

6.2.3 Cryptographic operation (SHA-1) (FCS_COP.1(3E))

FCS_COP.1.1 The CPACF shall perform hashing in accordance with a specified cryptographic algorithm SHA-1 and cryptographic key sizes not applicable that meet the following: FIPS 180-2 (August 2002)

Application note: *This function is provided by the “Compute intermediate message digest” and “Compute last message digest” instructions. Function Code 1 specifies SHA-1. The z890, z990 and later processors implement this function.*

6.2.4 Cryptographic operation (SHA-2) (FCS_COP.1(4E))

FCS_COP.1.1 The CPACF shall perform hashing in accordance with a specified cryptographic algorithms SHA-224, SHA-256, SHA-384, and SHA-512 and cryptographic key sizes not applicable that meet the following: FIPS 180-3.

Application note: *This function is provided by the “Compute intermediate message digest” and “Compute last message digest” instructions. Function Code 2 specifies SHA-256. Only the z9 and later processors implement this function. Function Code 3 specifies SHA-512. Only the z10 processor implements this function. With appropriate input values and post-processing, the SHA-256 processing can produce SHA-224 results, and the SHA-512 processing can produce SHA-384 results.*

6.3 Security requirements for PCIXCC and CEX2 in CEX2C mode

PCIXCC as well as CEX2 in CEX2C mode are cryptographic coprocessors that provide the ability to perform both symmetric and asymmetric encryption. When configured in CEX2C mode the CEX2 is identical to the PCIXCC both from the hardware components as well as from the software functions provided. The coprocessors can be used via ICSF which uses the CCA functions to request services from the coprocessor. In the evaluated configuration only a subset of the functions provided by the coprocessors are used providing some of the basic encryption functions required by System SSL. The following SFRs therefore reflect only those functions and not the full set of capabilities of the PCIXCC or CEX2C. TSF functions In the evaluated configuration may use the PCIXCC or CEX2C for RSA key generation as well as RSA encryption and decryption. Both the clear key option (where the private key may be exported in clear from the coprocessor to the TOE) as well as the secure key option (where the private key is never exported in clear from the coprocessor) may be used. The secure key option is useful in environments where the risk of leakage of the private key from the TOE is viewed as unacceptable. This allows the TOE to securely use public key cryptography, since the PCIXCC and CEX2C with their physical security protection provide an additional barrier for an attacker.

Although the PCIXCC and the CEX2C are also capable to perform symmetric encryption operations using DES and TDES, those functions are not used by the TSF. Performing DES or TDES symmetric encryption using the CPACF is significantly more efficient than using those functions on the PCIXCC or CEX2C.

6.3.1 Cryptographic operation (RSA) (FCS_COP.1(5E))

FCS_COP.1.1 The PCIXCC/CEX2C when operating on processors earlier than the System z9 shall perform encryption and decryption in accordance with a specified cryptographic algorithm RSA and cryptographic key sizes 1024 to 2048 bit that meet the following: RSA encryption and decryption operation as defined in PKCS#1 using either non-CRT or CRT key format as defined in section 3.2 of PKCS#1, Version 2-1.

Application note: *This function is with both the clear key and the secure key option.*

6.3.2 Cryptographic operation (RSA) (FCS_COP.1(7E))

FCS_COP.1.1 The CEX2C when operating on the System z9 or later processors shall perform encryption and decryption in accordance with a specified cryptographic algorithm RSA and cryptographic key sizes 1024 to 4096 bit that meet the following: RSA encryption and decryption operation as defined in PKCS#1 using either non-CRT or CRT key format as defined in section 3.2 of PKCS#1, Version 2-1.

Application note: *This function is with both the clear key and the secure key option.*

6.3.3 Cryptographic key generation (Public/Private Keys) (FCS_CKM.1(1E))

FCS_CKM.1.1 The PCIXCC/CEX2C when operating on processors earlier than the System z9 shall generate RSA public/private cryptographic keys in accordance with a specified cryptographic key generation algorithm none specified and specified cryptographic key sizes between 1024 and 2048 bit that meet the following: none.

Application note: *Keys are either generated as "cleartext keys" where the private key can be extracted in clear by the system using the PCIXCC or CEX2C or they are generated as "secure keys" where the private key is never exported in clear from the PCIXCC / CEX2C.*

6.3.4 Cryptographic key generation (Public/Private Keys) (FCS_CKM.1(2E))

FCS_CKM.1.1 The CEX2C when operating on the System z9 or later processors shall generate RSA public/private cryptographic keys in accordance with a specified cryptographic key generation algorithm none specified and specified cryptographic key sizes between 1024 and 4096 bit that meet the following: none.

Application note: *Keys are either generated as "cleartext keys" where the private key can be extracted in clear by the system using the CEX2C or they are generated as "secure keys" where the private key is never exported in clear from the CEX2C.*

6.4 Security requirements for PCICA and CEX2 in CEX2A mode

The PCICA as well as the CEX2 in CEX2A mode are used as accelerator cards for asymmetric encryption/decryption operations. They provide the ability for fast RSA encryption and decryption operations. The coprocessor performs no key generation and does not provide any key storage capability. The PCICA basically includes the hardware cryptographic processor also integrated into the PCIXCC and CEX2 coprocessor cards. While in the PCIXCC this hardware processor can only be used by the software on the coprocessor, the PCICA does not include any software and just exposes the interface of the hardware cryptographic processor to the TOE. In the case of the CEX2 the card exposes both the interface to the full functions of the card (including the software) when in CEX2C mode and the direct interface of the hardware cryptographic coprocessor when in CEX2A mode.

6.4.1 Cryptographic operation (RSA) (FCS_COP.1(6E))

FCS_COP.1.1 The PCICA/CEX2A shall perform encryption and decryption in accordance with a specified cryptographic algorithm RSA and cryptographic key sizes 1024 to 2048 bit that meet the following: key representation can be either of both ways (non-CRT and CRT) as specified in section 3.2 of PKCS#1 Version 2-1.

Application note: *The control block passed to the coprocessor identifies the operation to be performed as well as the key size and the key format used.*

7 Security Requirements

7.1 TOE Security Functional Requirements

The following table shows the Security functional requirements for the TOE, and the operations performed on the components according to CC part 2: iteration (Iter.), refinement (Ref.), assignment (Ass.) and selection (Sel.).

Security functional class	Security functional requirement	Security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
FAU Security audit	FAU_GEN.1 Audit data generation		OSPP	No	No	Yes	No
	FAU_GEN.2 User identity association		OSPP	No	No	No	No
	FAU_SAR.1 Audit review		OSPP	No	No	Yes	No
	FAU_SAR.2 Restricted audit review		OSPP	No	No	No	No
	FAU_SAR.3 Selectable audit review		CC-PART2	No	No	Yes	No
	FAU_SEL.1 Selective audit		OSPP	No	Yes	Yes	No
	FAU_STG.1 Protected audit trail storage		OSPP	No	No	No	Yes
	FAU_STG.3 Action in case of possible audit data loss		OSPP	No	Yes	Yes	No
	FAU_STG.4 Prevention of audit data loss		OSPP	No	No	Yes	Yes
FCS Cryptographic support	FCS_CKM.1(SYM) Cryptographic key generation: symmetric algorithms	FCS_CKM.1	OSPP	Yes	Yes	Yes	No
	FCS_CKM.1(RSA) Cryptographic key	FCS_CKM.1	OSPP	Yes	Yes	Yes	No

Security functional class	Security functional requirement	Security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	generation: RSA						
	FCS_CKM.1(DSA) Cryptographic key generation: DSA	FCS_CKM.1	OSPP	Yes	No	Yes	Yes
	FCS_CKM.2(NET) Cryptographic key distribution	FCS_CKM.2	OSPP	No	Yes	Yes	Yes
	FCS_CKM.4 Cryptographic key destruction		OSPP	No	No	No	Yes
	FCS_COP.1(SGN) Cryptographic operation: signatures	FCS_COP.1	CC-PART2	Yes	Yes	Yes	No
	FCS_COP.1(NET) Cryptographic operation: network	FCS_COP.1	OSPP	Yes	Yes	Yes	Yes
	FCS_RNG.1 Random number generation		OSPP	No	No	Yes	Yes
FDP User data protection	FDP_ACC.1(PSO) Subset access control: persistent objects	FDP_ACC.1	OSPP	Yes	No	Yes	No
	FDP_ACC.1(TSO) Subset access control: transient objects	FDP_ACC.1	OSPP	Yes	No	Yes	No
	FDP_ACF.1(PSO-MVS) Security attribute based access control: MVS	FDP_ACF.1	OSPP	Yes	Yes	Yes	No
	FDP_ACF.1(PSO-UNIX) Security attribute based access control: UNIX	FDP_ACF.1	OSPP	Yes	Yes	Yes	No
	FDP_ACF.1(PSO-LDAP) Security attribute based access control: LDAP	FDP_ACF.1	OSPP	Yes	Yes	Yes	No

Security functional class	Security functional requirement	Security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FDP_ACF.1(TSO) Security attribute based access control: UNIX IPC	FDP_ACF.1	OSPP	Yes	Yes	Yes	No
	FDP_ETC.1 (Labeled Security Mode only) Export of user data without security attributes		CC-PART2	No	Yes	Yes	No
	FDP_ETC.2(LS) (Labeled Security Mode only) Export of user data with security attributes	FDP_ETC.2	OSPP-LS	No	Yes	Yes	No
	FDP_IFC.2(LS) (Labeled Security Mode only) Complete information flow control: labeled security	FDP_IFC.2	OSPP-LS	Yes	Yes	Yes	No
	FDP_IFC.2(NI) Complete information flow control: network	FDP_IFC.2	OSPP	Yes	No	Yes	No
	FDP_IFF.1(NI) Simple security attributes	FDP_IFF.1	OSPP	No	No	Yes	Yes
	FDP_IFF.2(LS) (Labeled Security Mode only) Hierarchical security attributes	FDP_IFF.2	OSPP-LS	No	Yes	Yes	No
	FDP_ITC.1(LS) (Labeled Security Mode only) Import of user data without security attributes	FDP_ITC.1	OSPP-LS	No	Yes	Yes	No
	FDP_ITC.2 Import of user data with security attributes		OSPP	Yes	No	Yes	No
	FDP_ITC.2(LS) (Labeled Security Mode only)	FDP_ITC.2	OSPP-LS	Yes	Yes	Yes	No

Security functional class	Security functional requirement	Security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	Import of user data with security attributes: labeled security						
	FDP_RIP.2 Full residual information protection		OSPP	No	No	No	Yes
	FDP_RIP.3 Full residual information protection of resources		OSPP	No	No	No	Yes
FIA Identification and authentication	FIA_AFL.1 Authentication failure handling		OSPP	No	Yes	Yes	Yes
	FIA_ATD.1(HU) User attribute definition: human users	FIA_ATD.1	OSPP	Yes	Yes	Yes	No
	FIA_ATD.1(TU) User attribute definition: technical users	FIA_ATD.1	OSPP	Yes	No	Yes	No
	FIA_ATD.1(EIA) User attribute definition: EIA	FIA_ATD.1	OSPP-EIA	Yes	No	Yes	No
	FIA_ATD.1(LS) (Labeled Security Mode only) User attribute definition: labeled security	FIA_ATD.1	OSPP-LS	Yes	Yes	No	No
	FIA_SOS.1 Verification of secrets		OSPP	No	No	No	No
	FIA_UAU.1 Timing of authentication		OSPP	No	No	Yes	No
	FIA_UAU.5 Multiple authentication mechanisms		OSPP	No	Yes	Yes	No
	FIA_UAU.7 Protected authentication feedback		OSPP	No	No	No	No
	FIA_UAU.8(EIA)	FIA_UAU.8	OSPP-EIA	No	No	Yes	No

Security functional class	Security functional requirement	Security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	Authentication policy decisions						
	FIA_UID.1 Timing of identification		OSPP	No	No	Yes	No
	FIA_UID.3(EIA) Identification policy decisions	FIA_UID.3	OSPP-EIA	No	No	Yes	No
	FIA_USB.1(LS) (Labeled Security Mode only) User-subject binding	FIA_USB.1	OSPP-LS	No	Yes	Yes	No
	FIA_USB.2 Enhanced user-subject binding		OSPP	No	Yes	Yes	No
FMT Security management	FMT_MSA.1(PSO) Management of object security attributes: persistent objects	FMT_MSA.1	OSPP	Yes	Yes	Yes	Yes
	FMT_MSA.1(TSO) Management of object security attributes: transient objects	FMT_MSA.1	OSPP	Yes	Yes	Yes	Yes
	FMT_MSA.1(LS) (Labeled Security Mode only) Management of object security attributes: labeled security	FMT_MSA.1	OSPP-LS	Yes	Yes	Yes	No
	FMT_MSA.3(PSO) Static attribute initialization: persistent objects	FMT_MSA.3	OSPP	Yes	No	Yes	No
	FMT_MSA.3(TSO) Static attribute initialization: transient objects	FMT_MSA.3	OSPP	Yes	No	Yes	No
	FMT_MSA.3(NI) Static attribute initialization: network	FMT_MSA.3	OSPP	Yes	No	Yes	Yes

Security functional class	Security functional requirement	Security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FMT_MSA.3(LS) (Labeled Security Mode only) Static attribute initialization: labeled security	FMT_MSA.3	OSPP-LS	Yes	Yes	Yes	No
	FMT_MSA.4(PSO) Security attribute value inheritance: persistent objects	FMT_MSA.4	OSPP	No	No	Yes	No
	FMT_MTD.1(AE) Management of TSF data: audit events	FMT_MTD.1	OSPP	Yes	No	Yes	No
	FMT_MTD.1(AS) Management of TSF data: audit storage	FMT_MTD.1	OSPP	Yes	No	Yes	Yes
	FMT_MTD.1(AT) Management of TSF data: audit trail threshold	FMT_MTD.1	OSPP	Yes	Yes	Yes	Yes
	FMT_MTD.1(AF) Management of TSF data: audit storage failure	FMT_MTD.1	OSPP	Yes	Yes	Yes	Yes
	FMT_MTD.1(NI) Management of TSF data: network filters	FMT_MTD.1	OSPP	Yes	No	Yes	Yes
	FMT_MTD.1(NI2) Management of TSF data: IPSec	FMT_MTD.1	CC-PART2	Yes	No	Yes	Yes
	FMT_MTD.1(IAT) Management of TSF data: authentication threshold	FMT_MTD.1	OSPP	Yes	No	Yes	No
	FMT_MTD.1(IAF) Management of TSF data: account re-	FMT_MTD.1	OSPP	Yes	No	Yes	No

Security functional class	Security functional requirement	Security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	enablement						
	FMT_MTD.1(IAU) Management of TSF data: user security attributes	FMT_MTD.1	OSPP	Yes	Yes	Yes	No
	FMT_MTD.1(IAU-AUTH) Management of TSF data: authentication data	FMT_MTD.1	OSPP	Yes	Yes	Yes	No
	FMT_MTD.1(EIA) Management of TSF data: EIA	FMT_MTD.1	OSPP-EIA	Yes	No	Yes	No
	FMT_MTD.1(CRYPTO1) Management of TSF data: key import	FMT_MTD.1	CC-PART2	Yes	No	Yes	Yes
	FMT_MTD.1(CRYPTO2) Management of TSF data: digital certificates	FMT_MTD.1	CC-PART2	Yes	No	Yes	Yes
	FMT_MTD.1(ADD) Management of TSF data: additional configuration	FMT_MTD.1	CC-PART2	Yes	Yes	Yes	Yes
	FMT_REV.1(OBJ) Revocation: objects	FMT_REV.1	OSPP	Yes	No	Yes	No
	FMT_REV.1(USR) Revocation: users	FMT_REV.1	OSPP	Yes	No	Yes	No
	FMT_SMF.1 Specification of Management Functions		OSPP	No	No	Yes	No
	FMT_SMR.1 Security roles		OSPP	No	No	Yes	No
FPT Protection of the TSF	FPT_STM.1 Reliable time stamps		OSPP	No	No	No	No

Security functional class	Security functional requirement	Security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FPT_TDC.1 Inter-TSF basic TSF data consistency		OSPP	Yes	No	Yes	No
	FPT_TDC.1(LS) (Labeled Security Mode only) Inter-TSF basic TSF data consistency: labeled security	FPT_TDC.1	OSPP-LS	Yes	No	Yes	No
FTA TOE access	FTA_SSL.1 TSF-initiated session locking		OSPP	No	No	Yes	No
	FTA_SSL.2 User-initiated locking		OSPP	No	No	Yes	No
FTP Trusted path/channels	FTP_ITC.1 Inter-TSF trusted channel		OSPP	No	Yes	Yes	Yes

Table 6: Security functional requirements for the TOE

7.1.1 Security audit (FAU)

7.1.1.1 Audit data generation (FAU_GEN.1)

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

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

b) All auditable events for the basic level of audit; and

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

d) all user authentication attempts;

e) all denied accesses to objects for which the access control policy defined in the OSPP base applies;

f) explicit modifications of access rights to objects covered by the access control policies; and

g) **the events listed in Table 7: Auditable Events**.

Component	Event	Details
FAU_GEN.1	Startup and shutdown of the audit functions.	SMF type 81 record (RACF initialization). **Note:** SMF type 90 record, subtypes 5 and 9, record SMF status. IFASMFDP and IDCAMS can be used to report on these records.
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_SAR.2	Unsuccessful attempts to read information from the audit records.	SMF type 80 record, event code 2 (rejected attempt to access a raw SMF data set or a saved SMF data set).
FAU_SAR.3	None.	
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 can be used to report on these records).
FAU_STG.1	None.	
FAU_STG.3	Actions taken due to exceeding of a threshold.	Not applicable due to implementation. (The TOE switches automatically to another empty data set once the current data set used for auditing is full. The TOE is able to start a program that is

Component	Event	Details
		defined in the audit configuration to process the audit records in the data set that got filled up.)
FAU_STG.4	Actions taken due to the audit storage failure.	The system enters a wait state.
FCS_CKM.1(SYM)	The object attribute(s), and object value(s) excluding any sensitive information (e.g. secret or private keys)	None.

FCS_CKM.1(RSA)
FCS_CKM.1(DSA)
	Cryptographic key generation	SMF type 80 record, event code 66 for RACDCERT command with the GENREQ keyword specified

FCS_CKM.2(NET)	None.	
FCS_CKM.4	None.	
FCS_COP.1(SGN)	None.	
FCS_COP.1(NET)	None.	
FCS_RNG.1	None.	

FDP_ACC.1(PSO)
FDP_ACC.1(TSO)
	None.	

FDP_ACF.1(PSO-MVS)	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(PSO-UNIX)	All requests to perform	SMF type 80 record,

Component	Event	Details
FDP_ACF.1(TSO)	an operation on an object covered by the Security Function Policy (SFP).	event codes 28-30 for access to UNIX resources.
FDP_ACF.1(PSO-LDAP)	All requests to perform an operation on an object covered by the Security Function Policy (SFP).	SMF type 83 record, subtype 3,, event codes 1,3,5,8,9,10 for access to LDAP LDBM resources.
FDP_ETC.1(Labeled Security Mode only)	All attempts to export information.	SMF type 80 record, event code 2, for TAPEVOL class.
FDP_ETC.2(LS) (Labeled Security Mode only)	All attempts to export information.	SMF type 80 record, event code 2, for TAPEVOL class.
FDP_ETC.2(LS) (Labeled Security Mode only)	Overriding of human-readable output marking. (Additional)	SMF type 80 record, event code 2, for PSFMPL class.
FDP_IFC.2(LS)(Labeled Security Mode only)	None.	
FDP_IFC.2(NI)	None.	
FDP_IFF.1(NI)	All permitted traffic, all denied traffic not silently discarded	All traffic logged with syslog. Rules explicitly discarding packets will not generate a syslog record.
FDP_IFF.2(LS)(Labeled Security Mode only)	All decisions on requests for information flow.	SMF type 80 record, event code 2, with reason indicating SECLABEL AUDIT.
FDP_ITC.1(LS)(Labeled Security Mode only)	All attempts to import user data, including any security attributes.	SMF type 80 record, event code 2, associated with TAPEVOL profiles.

Component	Event	Details
FDP_ITC.2 FDP_ITC.2(LS)(Labeled Security Mode only)	All attempts to import user data, including any security attributes.	SMF type 80, event code 2, associated with TAPEVOL profiles.
FDP_RIP.2	None.	
FDP_RIP.3	None.	
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(TU), FIA_ATD.1(EIA), FIA_ATD.1(LS)(Labeled Security Mode only)	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 Also SMF type 80, event code 70, qualifier 2 for R_PKIServ Export function with incorrect passphrase.
FIA_UAU.1 FIA_UAU.8(EIA)	All use of the authentication mechanism.	SMF type 80 record, event code 1, various qualifiers and SMF record type 30 subtypes 1 and 5). Also SMF type 80, event code 68, qualifier 0

Component	Event	Details
		(success) or 1 (failure) to generate a Kerberos TGT 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. Also, SMF type 30 record.
FIA_UID.3(EIA)	All use of the user identification mechanism, including the identity provided during successful attempts.	SMF type 80 record, event code 1, various qualifiers. Also, SMF type 30 record.
FIA_USB.1(LS)(Labeled Security Mode only) 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. Also, SMF type 30 record, subtypes 1 and 5.

FMT_MSA.1(PSO)
FMT_MSA.1(TSO)
FMT_MSA.1(LS)
(Labeled Security Mode only)
	All modifications of the values of security attributes.	SMF type 80 record (generated by the RACF commands).

Component	Event	Details

FMT_MSA.3(PSO)
FMT_MSA.3(TSO)
FMT_MSA.3(LS) (Labeled Security Mode only)
	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.4(PSO)	None.	

FMT_MTD.1(AE)
FMT_MTD.1(AF)
FMT_MTD.1(AS)
FMT_MTD.1(AT)
	All modifications to the values of TSF data.	SMF type 80 record (generated by the RACF commands).

FMT_MTD.1(NI)
FMT_MSA.3(NI)
	All modifications of TSF data (Management of IPSec, IP filtering, and Defensive Filtering configuration from the command line)	SMF type 80 record generated by access check to SERVAUTH resource that controls ability to use this administrative interface.

FMT_MTD.1(NI2)	All modifications of TSF data (Management of IPSec via network interfaces)	SMF type 80 record generated by access check to SERVAUTH resource that controls ability to use this administrative interface

FMT_MTD.1(IAU)
FMT_MTD.1(IAU-AUTH)
FMT_MTD.1(IAF)
FMT_MTD.1(IAT)
FMT_MTD.1(EIA)
	All modifications to the values of TSF data.	SMF type 80 record (generated by the RACF commands).

FMT_MTD.1(CRYPTO1)	All modifications to the values of TSF data	SMF type 80 record (generated by the RACF command RACDCERT).
FMT_MTD.1(CRYPTO2)	All modifications of TSF data (Management activities related to PKI	auditing performed by PKI Services.:

Component	Event	Details
	services)	SMF Type 80. Event code 72 : Cert admin READ record Event code 73 : Cert admin Update request record Event code 74 : Cert admin Update certificate record Event code 79 : CRL Publication Event code 80 : PKI response for cert status Event code 83 : SCEP request
FMT_MTD.1(ADD)	All modifications of TSF data (Management activities related to other TOE configuration data)	SMF Type 80 records associated with access checks for access to MVS data sets, UNIX files, or LDAP objects holding the configuration data.
FMT_REV.1(USR)	All attempts to revoke security attributes.	SMF type 80 record (generated by the RACF commands).
FMT_REV.1(OBJ)	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, FAU_STG.3, FAU_STG.4, and	

Component	Event	Details
	FMT_SMR.1 requirements which are implied by FMT_SMF.1 as discussed in chapter 8.	
FMT_SMR.1	Modifications to the group of users that are part of a role.	SMF type 80 record (generated by the RACF commands).
FMT_SMR.1	Every use of the rights of a role. (Additional / Detailed)	SMF type 80 record.
FPT_STM.1	Changes to the time.	SMF type 80 record for MVS™ operator command SET CLOCK.
FPT_TDC.1 FPT_TDC.1(LS) (Labeled Security Mode only)	None	
FTA_SSL.1 FTA_SSL.2	None	
FTP_ITC.1	None	

Table 7: Auditable Events

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

a) Date and time of the event, type of event, subject identity (if applicable), and outcome of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST;

1. User identity (if applicable); and
2. **(in Labeled Security Mode) The sensitivity labels of subjects, objects, or information involved; and**
3. **The additional information specified in the "Details" column of Table 7: 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. Users accessing the HTTP server or LDAP server without authenticating themselves are audited with the user ID the server is configured to use for unauthenticated users. Also, for the HTTP server, authenticated users running under an administrator-configured ID for data access are audited with that administrator-configured ID. Also, in some cases of client authentication via SSL, when 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.*

7.1.1.2 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.

7.1.1.3 Audit review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide **the authorized administrators** 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.

Application note: *In this case, the term "authorized administrators" maps to the AUDITOR role of z/OS or a user with SPECIAL.*

7.1.1.4 Restricted audit review (FAU_SAR.2)

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

7.1.1.5 Selectable audit review (FAU_SAR.3)

FAU_SAR.3.1 The TSF shall provide the ability to apply **searches** of audit data based on **the following attributes:**

a) **user identity;**

b) **subject sensitivity label; (Labeled Security Mode only)**

c) **object sensitivity label; (Labeled Security Mode only)**

d) **object type and object name**

7.1.1.6 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) ~~Subject or~~ 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)**

Application note: *RACF allows inclusion of auditable events based on the criteria defined above.*

7.1.1.7 Protected audit trail storage (FAU_STG.1)

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

FAU_STG.1.2 The TSF shall be able to **prevent** unauthorized modifications to the audit records in the audit trail.

Application note: *RACF data set protection needs to be used to protect the files containing audit records from unauthorized access and modification.*

7.1.1.8 Action in case of possible audit data loss (FAU_STG.3)

FAU_STG.3.1 The TSF shall **generate an alarm to the z/OS operator** if the audit trail exceeds **the capacity of the current SMF data set**. ~~or if any of the following list of conditions is detected that may result in a loss of audit records.~~

Application note: *The TOE switches to the next available SMF data set. Saving the SMF data set that got filled up can be done automatically or manually.*

7.1.1.9 Prevention of audit data loss (FAU_STG.4)

FAU_STG.4.1 The TSF shall **prevent audited events, except those taken by the authorized administrator** and **inform a z/OS operator** if the audit trail is full.

7.1.2 Cryptographic support (FCS)

7.1.2.1 Cryptographic key generation: symmetric algorithms (FCS_CKM.1(SYM))

FCS_CKM.1.1 The TSF shall generate symmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm capable of generating a random bit sequence and specified cryptographic key sizes:

a) 128 bits *(TLS/SSL: AES; IPSec: AES; SSH: AES; Kerberos: AES)*,

b) 168 bits *(TLS/SSL: TDES; IPSec: TDES; SSH: TDES; Kerberos: TDES)*

c) 192 bits (SSH: AES)

d) 256 bits *(TLS/SSL: AES; IPSec: AES; SSH: AES; Kerberos: AES)*,

that meet the following:

a) TLS/SSL: generation and exchange of session keys as defined in the SSLv3 [SSLV3] ,TLSv1 [TLSV1], and TLSv1.1 [TLSV1.1] standards with the cipher suites defined in FCS_COP.1(NET)

b) **IPSec: FIPS 46-3,FIPS 197 as defined in RFC4109, RFC4306, RFC2308, and RFC4835.**

c) SSH: generation and exchange of session keys using the Diffie-Hellman key negotiation protocol as defined in RFC4253

d) Kerberos: generation and exchange of TDES, or AES session keys as defined in the Kerberos v5 standards (RFC1510, RFC3961, and RFC3962).

Application note: *For details of the SSH key generation and key negotiation process see section 8 of [RFC4253]. The evaluation will assess that the keys are generated in accordance with the requirements defined in [RFC4253].*

7.1.2.2 Cryptographic key generation: RSA (FCS_CKM.1(RSA))

FCS_CKM.1.1 The TSF shall generate RSA cryptographic keys in accordance with a specified cryptographic key generation algorithm defined in U.S. NIST FIPS PUB 186-3 appendix B.3 and specified cryptographic key sizes:

a) 1024 bits, 2048 bits or 4096 bits (X.509 certificates, SSH keys)

~~b) 2048 bits~~

that meet the following:

a) U.S. NIST FIPS PUB 186-3,

b) X.509v3 certificate structure as defined in ITU-T X.509 and RFC2459

c) generation of SSH host keys as defined in the Secure Shell (SSH) Transport Layer Protocol, RFC4253.

Application note: *For SSH keys, this requirement addresses the generation of public/private keys for host authentication, i.e., using the ssh-keygen utility. Exchange of the public keys generated involves a manual process of the administrator making the public key file available to the client users, and the client users copying those key files. ssh has not been modified from the Open Source version with respect to the cryptographic functions used and will therefore always use the software functions of the OpenSSL library for key generation and cryptographic operations.*

7.1.2.3 Cryptographic key generation: DSA (FCS_CKM.1(DSA))

FCS_CKM.1.1 The TSF shall generate DSA cryptographic keys in accordance with a specified cryptographic key generation algorithm defined in U.S. NIST FIPS PUB 186-3 appendix B.1 and specified cryptographic key sizes:
a) L=1024, N=160 bits;
b) L=2048, N=224 bits;
c) L=2048, N=256 bits;
d) DSA domain parameter generation for specified values for L and N;

that meet the following:

a) U.S. NIST FIPS PUB 186-3,

b) X.509v3 certificate structure as defined in ITU-T X.509 and RFC2459

c) generation of SSH host keys as defined in the Secure Shell (SSH) Transport Layer Protocol, RFC4253.

Application note: *Public/private key pairs for X.509v3 certificates can be generated in software with DSA keys up to 2048 bits. For SSH, the default key length for DSA keys generated by the ssh-keygen utility is 1024 bits. Please see also the application notes for FCS_CKM.1(RSA)*

7.1.2.4 Cryptographic key distribution (FCS_CKM.2(NET))

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with the following specified cryptographic key distribution method ~~that meets the following~~:

a) **RSA and DSA public keys: digital certificates for public RSA and DSA keys that meet the following: certificate format as defined in the standard X.509 Version 3;**

b) TLS/SSL handshake: RSA encrypted exchange of session keys according to [SSLV3], [TLSV1], and [TLSV1.1];

c) ***IPSec:* Diffie-Hellman key agreement method defined for the IKE protocol by RFC2409*, RFC4753, and RFC4754*;**

d) ***SSH symmetric session keys:* Diffie-Hellman key agreement method defined for the SSH protocol by RFC4253;**

e) **Kerberos: Kerberos v5 key distribution as defined by RFC3961**.

Application note: *This requirement addresses the exchange of public RSA and DSA keys during SSL/TLS or IPSec session negotiation, or as distributed within X509.v3 digital certificates distributed via the CA functions in PKI Services. In TOE configurations that include a PCIXCC or CryptoExpress2 (CEX2) in PCIXCC mode (CEX2C), RSA public/private key pairs may be generated by the coprocessor in a form where the private key is never exported in cleartext from the coprocessor. This case is covered by a specific security functional requirement for the IT environment. The administrator who generates the RSA key pair can specify in the RACDCERT command used for this key generation, if the key pair is generated by a cryptographic coprocessor (as part of the IT environment) or by the TOE itself. The requirement here does not cover this case but only the case where the key pair is generated by the TSF software (which is always the case for DSA key pairs generated by the TOE). The public/private key pair may also be generated external to the TOE or a PCIXCC or CEX2 cryptographic coprocessor attached to the TOE, and in this case it needs to be imported using appropriate protection measures as defined in FDP_ITC.1. This SFR addresses only the RSA and DSA key pair generation in software within the TOE. RSA key pair generation by the PCIXCC or CEX2 coprocessor is addressed by SFRs for those components in the IT environment.*

Application note: *This requirement addresses the exchange of TLS/SSL session keys as part of the TLS/SSL handshake protocol.*

Application note: *This requirement addresses the negotiation of session keys as defined in the IKE standard. The Diffie-Hellman public/private key pair is generated external to the TOE and needs to be imported using appropriate protection measures as defined in FDP_ITC.1.*

7.1.2.5 Cryptographic key destruction (FCS_CKM.4)

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method of **zeroization** that meets the following: **vendor-specific zeroization**.

Application note: *Cryptographic session keys for the SSL/TLS, Kerberos, SSH, and IPsec sessions are protected by the TOE against unauthorized access and are destroyed by the object re-use functions of the TOE. Long-living private keys of a public/private key pair will also be destroyed by the object reuse function of the TOE when they are kept in memory.*

7.1.2.6 Cryptographic operation: signatures (FCS_COP.1(SGN))

FCS_COP.1.1 The TSF shall perform **digital signature generation and digital signature verification** in accordance with ~~a specified~~ *the following* cryptographic algorithm*s*, ~~and~~ cryptographic key sizes *and applicable standards*:

a) **DSA with L=1024, N=160 bits as defined in FIPS 186-3**

b) **RSA with 1024bit**

that meet the following: **[SSLV3], [TLSV1], [TLSV1.1], Internet Security Association and Key Management Protocol (ISAKMP) RFC2408.**

Application note: *This requirement addresses the RSA and DSA digital signature generation and verification operations using the RSA or DSA algorithm as required by the SSL session establishment protocol (provided a cipher suite including RSA or DSA is used), the IPSec ISAKMP session establishment protocol, and digital certificate generation by RACDCERT (RACF) and PKI Services. The details of the signature format, such as the use of the PKCS#1 block type 1 and block type 2, are defined in the SSLv3 , TLSv1, and TLSv1.1 standards ([SSLV3], [TLSV1], [TLSV1.1]). Note that for ISAKMP only RSA is supported as a signature algorithm. When a PCIXCC, PCICA, or CEX2 coprocessor is attached to the hardware the TOE is operating upon and ICSF is installed and operational, System SSL and IPSec will use this hardware for RSA encryption and decryption operations. In those cases the RSA cryptographic operations of System SSL (including AT-TLS) and IPSec will be performed by the IT environment.*

7.1.2.7 Cryptographic operation: network (FCS_COP.1(NET))

FCS_COP.1.1 The TSF shall perform encryption, decryption, integrity verification, peer authentication in accordance with the following cryptographic algorithms, cryptographic key sizes and applicable standards:

a) **SSH allowing the use of TDES in CBC mode with 168 bits key**

size, and HMAC-SHA1 defined by RFC 4253 *(cipher suite: 3des-cbc)*;

b) **SSH allowing the use of AES in CBC *and CTR* mode with 128 bits, *192 bits* and 256 bits key size, and HMAC-SHA1 defined by RFC 4253 *(cipher suites: aes128-cbc, aes192-cbc, aes256-cbc, aes128-ctr, aes192-ctr, or aes256-ctr)*;**

c) TLS allowing the use of TDES in CBC mode with 168 bits key size, and HMAC-SHA-1 defined by RFC4346 (cipher suite TLS_RSA_3DES_EDE_CBC_SHA as defined in the TLSv1 standard [TLSV1] , [TLSV1.1] and RFC3268);

d) TLS allowing the use of AES in CBC mode with 128 bits and 256 bits key size, and HMAC-SHA-1 defined by RFC4346 (cipher suite TLS_RSA_WITH_AES_128_CBC_SHA and TLS_RSA_WITH_AES_256_CBC_SHA as defined in the TLSv1 standard [TLSV1] , [TLSV1.1] and RFC3268);

e) SSLv3 allowing the use of TDES with 168 bits key size , and HMAC-SHA-1 (cipher suite SSL_RSA_TDES_168_SHA as defined in the SSLv3 standard [SSLV3]);

f) SSLv3 allowing the use of AES in CBC mode with 128 bits and 256 bits key size, and HMAC-SHA-1 (cipher suite SSL_RSA_WITH_AES_128_CBC_SHA and SSL_RSA_WITH_AES_256_CBC_SHA equivalent to ciphers defined in RFC3268);

g) IPSec with IKE allowing the use of AES in CBC mode with 128 bits or 256 bits key size, and either SHA-1, SHA-2, AES_GMAC or AES_XCBC_MAC_96 as defined by RFC4301, RFC4303, RFC3602, RFC4106, RFC2404, and RFC4868;

h) IPSec allowing the use of HMAC-SHA-1 for message authentication, cryptographically securing the payload and the authentication header of an IP packet as defined in IETF RCF2406 (IP Encapsulating Security Payload [ESP]) and IETF RFC2402 (IP Authentication Header) using the specific method for HMAC-SHA-1 as defined in IETF RFC2404 (The Use of HMAC-SHA-1-96 within ESP and AH);

i) IPSec allowing the use of AES in GCM mode with 128 bits and 256 bits key size for combined encryption and authentication as defined by RFC4301, RFC4303, RFC3602, RFC4106, RFC2404, and RFC4868;

j) Kerberos allowing TDES or AES with 168 bit (TDES), 128-bit (AES) or 256-bit (AES) for Encryption and Checksum specifications as defined in RFC3961.

Application note: *TLS/SSL: Triple DES and AES encryption may be performed using the* *supporting CPACF processor instructions of the z/Architecture. Note that hardware support for AES is available on the z9 and later processors only. System SSL will check for the availability of those functions in the underlying hardware and use the CPACF processor instructions when they support Triple DES or AES. In those cases the cryptographic operations related to those functions will be performed by the IT environment (the z/Architecture processor).*

Application note: *IPSec: SHA-1 hashing may be performed using the supporting CPACF processor instructions of the z/Architecture. CS390 will check for the availability of those functions via ICSF and use ICSF functions to perform those cryptographic operations. ICSF will use the CPACF processor instructions when they support SHA-1. If SHA-1 is not supported by the processor but a PCIXCC or CEX2C coprocessor is installed on the system operating the TOE, ICSF will use those for the cryptographic operations for SHA-1. In those cases the cryptographic operations related to those functions will be performed by the IT environment (the z/Architecture processor).*

Application note: *IPSec: Triple DES and AES encryption may be performed using the supporting CPACF processor instructions of the z/Architecture. Note that hardware support for AES is available on the z9 and newer processors only. CS390 will check for the availability of those functions via ICSF and use ICSF functions to perform those cryptographic operations. ICSF will use the CPACF processor instructions when they support Triple DES or AES. If Triple DES is not supported by the processor but a PCIXCC or CEX2C coprocessor is installed on the system operating the TOE, ICSF will use those for the cryptographic operations for Triple DES. In those cases the cryptographic operations related to those functions will be performed by the IT environment (the z/Architecture processor). If AES is not supported by the processor, ICSF will use its software implementation of the AES algorithm (AES is currently not supported by the PCIXCC or CEX2C coprocessors).*

Application note: *Kerberos: AES or Triple DES encryption may be performed using the supporting CPACF processor instructions of the z/Architecture. Network Authentication Service will check for the availability of those functions via ICSF and use ICSF functions to perform those cryptographic operations. ICSF will use the CPACF processor instructions when they support AES or Triple DES. If Triple DES is not supported by the processor but a PCIXCC or CEX2C coprocessor is installed on the system operating the TOE, ICSF will use those for the cryptographic operations for Triple DES. In those cases the cryptographic operations related to those functions will be performed by the IT environment (the z/Architecture processor). If AES is not supported by the processor, ICSF will use a software implementation.*

7.1.2.8 Random number generation (FCS_RNG.1)

FCS_RNG.1.1 The TSF shall provide a **deterministic** random number generator that implements: **forward secrecy and backward secrecy**.

FCS_RNG.1.2 The TSF shall provide random numbers that meet **the requirements of functionality class K3 for a medium strength of function as defined in [AIS20]**.

7.1.3 User data protection (FDP)

7.1.3.1 Subset access control: persistent objects (FDP_ACC.1(PSO))

FDP_ACC.1.1 The TSF shall enforce the Persistent Storage Object Access Control Policy on

a) **jobs, started tasks, UNIX processes (whether initiated by rlogin, telnet, HTTP, FTP, or other method), and TSO sessions acting on behalf of users**

b) Objects: Persistent Storage Objects of the following type

MVS objects:

data sets, terminals, devices, volumes, consoles, operator commands, programs, System Logger objects, Communications Server Policy Agent data;

UNIX objects:

z/OS UNIX file system objects (regular files, directories and symbolic links, character special files, UNIX domain sockets and named pipes (FIFOs);

LDAP objects:

LDAP LDBM objects;

c) Operations: **all operations among subjects and objects covered by the Persistent Storage Object Access Control Policy.**

Application note: *A persistent storage object establishes a data storage or data exchange link between two or more subjects. Examples of persistent storage objects are: files, directories.*

7.1.3.2 Subset access control: transient objects (FDP_ACC.1(TSO))

FDP_ACC.1.1 The TSF shall enforce the Transient Storage Object Access Control Policy on

a) **jobs, started tasks, UNIX processes (whether initiated by rlogin, telnet, HTTP, FTP, or other method), and TSO sessions acting on behalf of users**

b) Objects: Transient Storage Objects of the following type

z/OS UNIX IPC objects:

shared memory segments, message queues,

semaphores

TCP/IP connections;

c) Operations: **all operations among subjects and objects covered by the Transient Storage Object Access Control Policy**.

Application note: *A transient storage object establishes a data exchange link between two or more subjects or users. Examples of transient storage objects are: shared memory, semaphores, message queues, named/unnamed pipes.*

7.1.3.3 Security attribute based access control: MVS (FDP_ACF.1(PSO-MVS))

FDP_ACF.1.1 The TSF shall enforce the Persistent Storage Object Access Control Policy to *MVS* objects based on the following:

a) **The user identity and group memberships associated with a subject; and**

b) **The following access control attributes associated with an object:**

1. **an access control list capable of defining the access rights read, update, execute, alter, control, and none for individual users and groups**
2. **a default access right (defined by the UACC attribute in the resource profile) for users who are not addressed in the access control list**
3. **an entry for the resource containing the object in the global access checking table.**

Application note: *The semantics of "read", "update", "execute", "alter", and "control" are defined by the resource manager and follow the intuitive semantics of those terms. In the case of the Communication Server Policy Agent data, the resource manager implements only "read" access to this data.*

Any access right hierarchical to read for the profile protecting this data will therefore still result only in read access to this data. In the case of Operator Commands, the semantics of the different access rights is defined as part of the description of the command.

FDP_ACF.1.2 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 protected resource, if the resource is protected by RACF and**

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) td 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 td 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 td 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 td (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 td 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 td. 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 td 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 td. (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 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: *The term "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 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 only be accessed by users with the SPECIAL role**.

7.1.3.4 Security attribute based access control: UNIX (FDP_ACF.1(PSO-UNIX))

FDP_ACF.1.1 The TSF shall enforce the Persistent Storage Object Access Control Policy to *UNIX* 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 and (for file system objects) an access control list capable of defining access rights read, write, execute, or search. Default access rights are defined by a system management attribute.**

Access rights for file system objects are:

a) **read**

b) **write**

c) **execute (ordinary files)**

d) **search (directories)**

Access is defined by POSIX ACLs and permission bits. ACLs are evaluated only when the FSSEC class is active in RACF.

Users who have the AUDITOR attribute have implicit SEARCH and READ access for directories, without needing explicit permission via the permission bits or ACLs.

FDP_ACF.1.2 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.

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 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, the requested type of access is execute, there is no permission bit, and there is no ACL that provides execute access to any user, access is denied.**
- **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) **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 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 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **none**.

7.1.3.5 Security attribute based access control: (FDP_ACF.1(PSO-LDAP))

FDP_ACF.1.1 The TSF shall enforce the Persistent Storage Object Access Control Policy to *LDAP* objects based on the following:

a) **The z/OS LDAP Bind DN identity associated with a subject, together with the subject’s LDAP groups derived during bind processing; and**

b) **LDAP ACLs or ACL Filters that determine whether the access is allowed or not, and**

c) **The entryOwner attribute that applies to the object.**

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

a) **The owner of the LDAP object as well as the LDAP administrator (identified by the administrator DN) are always allowed full access to the object**

b) **In the case the z/OS LDAP user identity is neither the owner nor the LDAP administrator access is determined by the LDAP ACL associated with the LDAP object. This ACL is determined as follows:**

   1. **If the LDAP object has an explicit AclEntry, the ACLs in this entry are used to determine access**
   2. **If the LDAP object has no explicit AclEntry, the next entry found when traversing up the directory tree that has an explict AclEntry and has the AclPropagate attribute set to TRUE, defines the AclEntry used to determine access**
   3. **If no LDAP object with an explicit AclEntry is found by the above two steps, the default ACL is used to determine access**

c) **ACLs in the AclEntry are evaluated as follows to determine access:**

   1. **if there is a specific value for the DN of the LDAP user, the LDAP user gets those permissions only**
   2. **else if there is a cn=this value and the DN of the LDAP user is the distinguished name of the entry, the LDAP user gets those permissions only**
   3. **else if there are one or more group values that the LDAP user is a member of, the LDAP user gets the union of the permissions for those groups**
   4. **else if there is a cn=authenticated value and the LDAP user is authenticated to the directory with an LDAP bind operation, the LDAP user gets those permissions only**
   5. **else if there is a cn=anybody value, the LDAP user gets those permissions only**
   6. **otherwise the LDAP user gets no permissions**

d) **ACLs in the AclEntry may specify “grant” or “deny” permissions for the object as a whole, for specific named**

attributes within the object, or for attribute classes within the object. The LDAP server will process the ACLs in a precedence order to determine which ACL best applies to the user's request. The higher priority of the following list have preference over lower priorities (listed from highest to lowest):

- **a) attribute-level deny permissions**
- **b) attribute-level grant permissions**
- **c) access-class deny permissions**
- **d) access-class grant permissions**

e) **After a user's base access has been determined, it may be modified by Filter ACL entries. Filter ACLs can set permissions based on any of the following:**

- **a) bind DN**
- **b) alternate DNs**
- **c) pseudo DNs**
- **d) groups that the bind or alternate DNs belong to**
- **e) IP address of the client connection**
- **f) time of day that directory entry was accessed**
- **g) day of week that directory entry was accessed**
- **h) the bind mechanism used**
- **i) whether or not bind encryption was used**

Filters support wildcards. Filters have the same syntax support as LDAP search filters, where logical rules can be specified, such as "&"(and), "|"(or), and "!"(not).

To allow flexibility, the aclEntry filtering mechanism also supports three operation values that allow administrators to specify the way in which filtered ACLs will take effect:

- **a) replace - the base effective ACL is replaced by the filtered ACLs.**

  If administrators want a client from a given IP address to only have a specific set of permissions, they would use replace.

- **b) union - the base effective ACL is unioned with the filtered ACLs, resulting in a new effective ACL. This would be used to expand permissions.**

  If administrators want a client from a given IP address to have a specific set of permissions, at a

minimum, they would use union.

c) **intersect - the base effective ACL is intersected with the filtered ACLs. This would be used to reduce permissions.**

   If administrators want a client from a given IP address to have a specific set of permissions, if and only if they already have the permissions, they would use intersect.

As with the operator precedence described above, filter ACL entries specifying “deny” take precedence over entries specifying “grant”.

Application note: *The owner of an LDAP object is determined by the entryOwner attribute, or (if this does not exist for the LDAP object) by the ownerSource attribute. The ownerSource attribute is not modifiable and is managed by the TOE. It indicates the DN of the td that holds the entryOwner attribute that applies to this object. This is the first td encountered, while traveling up the directory tree from the object toward the root, which has an entryOwner attribute and has the ownerPropagate attribute set to TRUE.*

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

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

7.1.3.6 Security attribute based access control: UNIX IPC (FDP_ACF.1(TSO))

FDP_ACF.1.1 The TSF shall enforce the Transient Storage Object Access Control Policy to *UNIX* 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. Default access rights are defined by a system management attribute.**

Access rights for z/OS UNIX IPC objects are:

a) **read**

b) **write**

Access is defined by permission bits only.

FDP_ACF.1.2 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:**

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

c) **the user is neither the owner nor the creator of the IPC object and 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,**

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

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

7.1.3.7 Export of user data without security attributes (FDP_ETC.1 (Labeled Security Mode only))

FDP_ETC.1.1 The TSF shall enforce the **mandatory access control policy** when exporting *unlabeled* user data, controlled under the *MAC policy*~~SFP(s)~~, outside of the TOE.

FDP_ETC.1.2 The TSF shall export the *unlabeled* user data without the user data's associated security attributes

7.1.3.8 Export of user data with security attributes (FDP_ETC.2(LS) (Labeled Security Mode only))

FDP_ETC.2.1 The TSF shall enforce the *mandatory access control policy*~~Multilevel Confidentiality Information Flow Control Policy~~ when exporting *labeled* user data, controlled under the *MAC-policy*~~SFP(s)~~, outside of the TOE.

FDP_ETC.2.2 The TSF shall export the *labeled* user data with the user data's associated security attributes.

FDP_ETC.2.3 The TSF shall ensure that the security attributes, when exported outside the TOE, are unambiguously associated with the exported *labeled* user data.

FDP_ETC.2.4 The TSF shall enforce the following rules when *labeled* user data is exported from the TOE:

a) When data is exported in hardcopy form, each page shall be marked with a printed representation of the sensitivity label of the subject requesting the export of the page. By default, this marking shall appear on both the top and bottom of each printed page.

b) When the data is exported to a device the security attributes shall be exported with the data using **the printable label that is assigned to the sensitivity label associated with the data by the authorized administrator**.

c) devices used to export data with security attributes shall completely and unambiguously associate the security attributes with the corresponding data.

Application note: *A properly-authorized system administrator can export data with its labels by placing all of the data to be exported in a multi-level zFS UNIX file system. The z/OS data set that contains the zFS file system must be classified as SYSHIGH, which ensures that only a system administrator who is authorized to work with this data can directly read the z/OS data set containing the zFS UNIX file system.*

The security labels of each file in the zFS file system are stored as extended attributes in the file system and exported with the file system when the z/OS data set containing the file system is written to a tape volume. When importing such a file system, it is the task of the system administrator to ensure that the importing system is set up in a way that it correctly interprets the labels.

It also possible to set up a zFS UNIX file system within a z/OS data set that has a dedicated security label. The TOE then enforces that all zFS files within this file system have the same security label as the z/OS data set containing the zFS file system. In this case, any user who has read access to the z/OS data set may export the data set to a tape volume in accordance with the security policy enforced by the TOE. When this tape *volume is read in another system, the labels of the files in the zFS file system (which are all identical) can also be imported and interpreted.*

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

FDP_IFC.2.1 The TSF shall enforce the *mandatory access control policy (MAC-policy)* ~~Multilevel Confidentiality Information Flow Control Policy~~ on

a) Subjects: **jobs, started tasks, UNIX sessions, and TSO sessions acting on behalf of users**;

b) Objects: **data sets, volumes, devices, z/OS UNIX file system objects, z/OS UNIX IPC objects, terminals, TCP/IP connections,**

and all operations that cause that information to flow among them.

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*~~Multilevel Confidentiality Information Flow Control Policy~~.

7.1.3.10 Complete information flow control: network (FDP_IFC.2(NI))

FDP_IFC.2.1 The TSF shall enforce the Network Information Flow Control Policy on

a) Subjects:

1. unauthenticated external IT entities that send and receive information mediated by the TOE;
2. **none** that send and receive information mediated by the TOE;

b) Information:

1. Network data routed through the TOE;
2. **no other information**;

and all operations that cause that information to flow to and 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 to and from any subject in the TOE are covered by an information flow control SFP.

Application note: *This requirement covers IPv4 and IPv6 traffic.*

7.1.3.11 Simple security attributes (FDP_IFF.1(NI))

FDP_IFF.1.1 The TSF shall enforce the Network Information Flow Control Policy based on the following types of subject and information security attributes:

a) Object security attribute: the logical or physical network interface through which the network data entered the TOE;

b) TCP/IP information security attributes:

1. **Source and destination IP address,**
2. **Source and destination TCP port number,**
3. **Source and destination UDP port number,**
4. **Network protocol of IP, TCP, UDP, ICMP,**
5. **TCP header flags of SYN, ACK.**

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

For z/OS TCP/IP stacks configured for IP security, the TOE allows an IP packet to be sent or to be received by a subject if a rule described in FDP_IFF.1.3 applies and explicitly allows the packet flow.

FDP_IFF.1.3 The TSF shall enforce the following rules:

Identification of IP packets using one or more of the following concepts:

a) Information security attribute matching;

b) **Matching based on the state of a TCP connection**;

Performing one or more of the following actions with identified network data:

a) Discard the network data **without any further processing**;

b) Allow the network data to be processed unaltered by the TOE according to the routing information maintained by the TOE;

c) **no other actions**.

FDP_IFF.1.4 The TSF shall explicitly authorize an information flow based on the following rules: **no additional rules.**

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: **no additional rules.**

Application note: *This requirement covers IPv4 and IPv6 traffic.*

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

FDP_IFF.2.1 The TSF shall enforce the *mandatory access control policy* ~~Multilevel Confidentiality Information Flow Control Policy~~ based on the following types of subject and object security attributes:

a) Subject security attributes:

1. Sensitivity label of the subject consisting of at least 8 site definable hierarchical levels and a set of 60 site definable non-hierarchical categories;
2. **none**;

b) Object security attributes:

1. 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;
2. **none**.

FDP_IFF.2.2 The TSF shall permit an information flow between a controlled subject and 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 the **none**.

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**.

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:

1. Sensitivity labels are equal if the hierarchical level of both labels are equal and the non-hierarchical category sets are identical;
2. 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;
3. 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.

7.1.3.13 Import of user data without security attributes (FDP_ITC.1(LS) (Labeled Security Mode only))

FDP_ITC.1.1 The TSF shall enforce the *mandatory access control policy* ~~Multilevel Confidentiality Information Flow Control Policy~~ when importing unlabeled user data controlled under the *mandatory access control policy* ~~SFP~~, from outside of the TOE.

FDP_ITC.1.2 The TSF shall ignore any label-related security attributes associated with the unlabeled user data when imported from outside the TOE.

FDP_ITC.1.3 The TSF shall enforce the following rules when importing unlabeled user data controlled under the *mandatory access control policy* ~~SFP~~ from outside the TOE:

a) When importing unlabeled data, the TSF shall allow the **authorized administrator** to specify that the data is to be labeled with: **a label manually chosen by the authorized administrator**.

Application note: *See the Application note on FDP_ETC.1 for export of unlabeled data. The requirement also applies for the import of RSA key pairs or Diffie-Hellman key pairs imported to be used for the cryptographic operations of the TOE. The administrators need to ensure using the MAC and DAC policy enforced by the TOE that this key material is imported in a secure way and can not be imported by unauthorized users.*

7.1.3.14 Import of user data with security attributes (FDP_ITC.2)

FDP_ITC.2.1 The TSF shall enforce the Persistent Storage Access Control Policy, Network Information Flow Control Policy, **none** when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.2.2 The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3 The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4 The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: **none**.

Application note: *If, for example, file names or file name extensions are used for access control decisions, they are security attributes. In the case that an external file system is mounted, this is considered an import of user data with security attributes, and therefore, FDP_ITC rules must be defined and satisfied.*

Application note: *Based on the wording of FDP_ITC.2.1, the TOE complies with this SFR even when it does not allow import of objects covered by the persistent or transient storage object control policy.*

However, network information flow control policy must always be covered by the TOE, as it applies to the networking capability of the TOE to control traffic originating from outside the TOE. In this case, the interpretation of security attributes is defined by the respective protocol family.

7.1.3.15 Import of user data with security attributes: labeled security (FDP_ITC.2(LS) (Labeled Security Mode only))

FDP_ITC.2.1 The TSF shall enforce the *mandatory access control policy* ~~Multilevel Confidentiality Information Flow Control Policy~~ when importing labeled user data, controlled under the *mandatory access control policy* ~~SFP~~, from outside of the TOE.

FDP_ITC.2.2 The TSF shall use the label-related security attributes associated with the imported labeled user data.

FDP_ITC.2.3 The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the *labeled* user data received.

FDP_ITC.2.4 The TSF shall ensure that interpretation of the label-related security attributes of the imported *labeled* user data is as intended by the source of the user data.

FDP_ITC.2.5 The TSF shall enforce the following rules when importing *labeled* user data controlled under the *MAC policy* ~~SFP~~ from outside the TOE:

a) **devices used to import data with security attributes shall unambiguously associate security labels with the corresponding data.**

Security labels consist of the following:

1. **a hierarchical level; and**
2. **a set of non-hierarchical categories.**

7.1.3.16 Full residual information protection (FDP_RIP.2)

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the **allocation of the resource to** all objects.

7.1.3.17 Full residual information protection of resources (FDP_RIP.3)

FDP_RIP.3.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the **allocation of the resource to** all subjects or users.

7.1.4 Identification and authentication (FIA)

7.1.4.1 Authentication failure handling (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when an administrator-configurable number of *consecutive* unsuccessful authentication attempts for the authentication method*s* **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**.

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

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

a) User identifier;

b) Group memberships;

c) User password *or password phrase*;

~~d) Software token verification data;~~

e) Security roles;

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

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

h) 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);

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

j) z/OS UNIX group memberships;

k) 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);

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

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

n) **X.509v3 certificate(s)**.

Application note: *The software token verification data can be implemented as transient in nature. For example, a Kerberos ticket granting ticket or Kerberos ticket is created when the user requests these tickets. The ticket granting server has the data to verify the Kerberos ticket granting ticket, whereas the application server has the data to verify the Kerberos ticket.*

Application note: *Software token verification data for FIA_ATD.1(HU) is implemented by the Kerberos principal name and X.509v3 certificates.*

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

7.1.4.3 User attribute definition: technical users (FIA_ATD.1(TU))

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

a) the logical or physical network interface through which the network data entered the TOE;

b) identity of the logical or physical external interface through which the user connected to the TOE;

c) Source IP address;

d) Destination IP address;

e) **Destination port**.

Application note: *Bullet a) of this SFR relates to FDP_IFC.2(NI) and FDP_IFF.1(NI). In the Common Criteria scheme, external entities are always considered to be users. Therefore, every network data entity must be specified as user in this ST.*

7.1.4.4 User attribute definition: EIA (FIA_ATD.1(EIA))

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users for remote identification and authentication:

a) z/OS LDAP user (bind) identifier (for users also defined to LDAP LDBM); and

b) z/OS LDAP group memberships (for users also defined to LDAP LDBM).

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

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

a) Sensitivity label *(in Labeled Security Mode)*,

b) *user clearances (in Labeled Security Mode)*.

7.1.4.6 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.*

7.1.4.7 Timing of authentication (FIA_UAU.1)

FIA_UAU.1.1 The TSF shall allow

a) the information flow covered by the Network Information Flow Control Policy;

b) all functions allowed to be performed by the individual pseudo-user assigned by the authorized administrator for started procedures (started tasks);

c) administrator-specified anonymous access to specific data via HTTP, FTP, or LDAP

on behalf of the user to be performed before the user is authenticated.

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.

Application note: *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. z/OS 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 used to log in under TSO or UNIX. 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. Write access to those libraries needs to be restricted to system administrators.*

This concept does not allow an unauthenticated user to execute any program or command on the TOE. Instead this concept allows an authenticated and properly authorized user to start specific tasks that have previously been defined by an authorized administrator and that operate under a pseudo–user-ID. The user that started this task usually has no influence on what the task is doing. The fact that he started the Started Procedure is auditable which ensures that the individual accountability for starting the started procedure is given. The ID of the pseudo-user listed in the JOB statement of the started procedure is not authenticated.

Also, z/OS allows an authorized administrator to configure the HTTP server, the FTP server, or the LDAP server to allow “anonymous” access to selected data. Such access occurs for HTTP or FTP using an administrator-specified user ID, which also is a form of pseudo-user, and the administrator controls which data that user has access to, and whether such anonymous access is enabled or not. For LDAP, the administrator can control whether a particular LDAP LDBM server allows unauthenticated access or not, and can further control which data in the LDBM database the unauthenticated user can access. For LDAP, the default is to allow anonymous access, and so the administrator who chooses to enable LDAP access must usually disable the default anonymous access.

7.1.4.8 Multiple authentication mechanisms (FIA_UAU.5)

FIA_UAU.5.1 The TSF shall provide the following authentication mechanisms:

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

b) Authentication based on software token verification data *(digital certificates, Kerberos tickets)*;

c) RACF PassTickets

to support user authentication.

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 TOE-originated requests and credentials stored by the TSF;

b) Authentication based on software token verification data is performed for TOE-originated requests;

c) TSF applications that perform user authentication accept any of the above listed authentication mechanisms provided they are configured for this mechanism (in the case of digital certificates or Kerberos tickets) and the mechanism is also supported for the user that attempts to authenticate.

Attempts to authenticate to such an application using a mechanism the application is not configured for or where the mechanism is not supported for the user will result in the rejection of the authentication attempt.

Application note: *For the term “software token verification data”, see the Application note for FIA_ATD.1(HU).*

7.1.4.9 Protected authentication feedback (FIA_UAU.7)

FIA_UAU.7.1 The TSF shall provide only obscured feedback to the user while the authentication is in progress.

Application note: *When entered during TSO LOGON the user has the option to use those TSF functions in a way that prohibits passwords/phrases from being displayed. Passwords for Operator LOGON are not displayed. Passwords a user enters via a JCL JOB statement will be suppressed in any output of the JCL statements to prohibit the password from being obtained by anybody reading the output.*

For authentication performed by servers where the userid and password/phrase is transferred over the network, the servers ensure that no feedback is provided as long as the authentication is in progress. For protocols where the server can request the client to suppress the display of characters entered by the user, such a request is sent before passwords/phrases are requested to be entered by the user. This is done for telnet, TN3270, and the r-commands. This still requires that the clients used implement those controls (e. g. switching to no-echo mode) correctly. In the case of FTP, SSH, Kerberos, and LDAP the protocols do not have any control statements that can be sent to the client to suppress the display of characters when a user enters a password/phrase. In those cases the TSF have no control how the client obtains a user's password/phrase and just ensures that no password/phrase related information is sent back to the client.

In all cases where clients operating as regular user programs are used it is outside of the control of the TSF how those clients handle the password/phrase. Where those interfaces are defined as part of the communication protocol, the TSF interfaces of the servers just ensure that the clients get the required information to suppress displaying passwords/phrases.

Client programs supplied by the TOE that operate as regular user programs (su, kinit, kpasswd, ssh, etc.) do not echo the password/phrase, but as they are user programs they are not part of the TSF.

Note that in the case of authentication via digital certificates, Kerberos tickets or PassTickets, no feedback is provided during the time authentication is in progress.

7.1.4.10 Authentication policy decisions (FIA_UAU.8(EIA))

FIA_UAU.8.1 The TSF shall accept an authentication request holding the user credentials from **previously authenticated XMLAppliance instances that are connected to NSS**.

FIA_UAU.8.2 The TSF shall perform the authentication operation based on the transmitted user credentials according to **the rules for user authentication enforced by RACF**.

FIA_UAU.8.3 The TSF shall transmit the result of the authentication operation to **the XMLAppliance that submitted the authentication request** according to the **following process: NSS calls RACF to perform the user authentication and transmits the result returned by RACF to that XMLAppliance**.

7.1.4.11 Timing of identification (FIA_UID.1)

FIA_UID.1.1 The TSF shall allow **access to the HTTP server, FTP server, or LDAP server (restricted to the functions and resources accessible to the pseudo user the administrator assigned for that purpose)** 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.

Application note: *The pseudo-user of a started task is identified within the JOB statement of the JCL defining the started task. Users who start a started task (which will not be executed with the ID of the user that started the task) need to be identified and authenticated before they can perform this action. The FTP, LDAP, and HTTP server will assign an administrator defined ID of a pseudo user to users that connect to those servers without authenticating themselves. In this case all security related decisions are based on this ID.*

7.1.4.12 Identification policy decisions (FIA_UID.3(EIA))

FIA_UID.3.1 The TSF shall accept an identification request holding the user credentials from **previously authenticated XMLAppliance instances that are connected to NSS**.

FIA_UID.3.2 The TSF shall perform the identification operation based on the transmitted user credentials according to **the rules for user identification enforced by RACF**.

FIA_UID.3.3 The TSF shall transmit the result of the identification operation to **the XMLAppliance that submitted the authentication request** according to the **following process: NSS calls RACF to perform the user authentication and transmits the result returned by RACF to that XMLAppliance**.

7.1.4.13 User-subject binding (FIA_USB.1(LS) (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* ~~Multilevel Confidentiality Information Flow Control Policy~~ *which consists of the following:*

b. A hierarchical level; and

c. 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.**

7.1.4.14 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 or LDAP* user identity that is associated with auditable events;

b) The *RACF or LDAP or UNIX* user security attributes that are used to enforce the Persistent Storage Object Access Control Policy;

c) The *RACF or LDAP or UNIX* user security attributes that are used to enforce the Transient Storage Object Access Control Policy;

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

e) Active roles;

f) Active groups;

g) **the RACF attributes/roles SPECIAL, group-SPECIAL, AUDITOR, group-AUDITOR, CLAUTH OPERATIONS, and group-OPERATIONS.**

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) **A user that connects to the HTTP server or LDAP server without authenticating will be bound to the identity the installation has assigned for the unauthenticated user of the server, and limited to accessing data that user is allowed to access, unless and until the user is successfully identified and authenticated using his own authentication information.**

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) **A z/OS administrator may define specific z/OS applications to execute with an administrator defined user ID.**

b) **A z/OS administrator may use the SURROGAT authority mechanism to allow a user to switch his identify to another defined user (e. g. submitting jobs or changing the ID with the su command in the z/OS UNIX System Services environment) without specifying the password/phrase for this user.**

In z/OS UNIX, the following additional rules apply:

a) **The su command provides the ability to create a new session with a new set of credentials (to be inherited by subjects created within this session). The credentials are set to the UID (RUID and EUID), GID (RGID and EGID), and supplementary groups of the user requested. The user issuing the su command must have the authority to use this command, have the authority to switch to the specified UID and either authenticates properly for this UID with the password/phrase , has the SURROGAT authority for the new UID or has BPX.SUPERUSER authority allowing him to switch to UID 0 without supplying a password/phrase.**

b) **If the BPX.DAEMON profile exists in the FACILITY class of RACF, a user with UID 0 needs to have authority other than NONE to this profile to change his UID using the setuid or seteuid system calls.**

Application note: *In the z/OS BCP, a temporary change of the user ID is not implemented. In z/OS UNIX System Services this is possible with a slightly-modified semantic compared to other UNIX systems.*

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) **When executing a program from a file with the set-user-ID-on-execution bit (S_ISUID) set, the subject’s effective UID is**

set to the owner ID of the file being executed; when executing the program from a file with the set-group-ID-on-execution bit (S_ISGID) set, the subject's effective GID is set to the group ID of the file being executed;

b) **The Port of Entry (POE) is set to TERMINAL (when the user has started his session from a terminal), CONSOLE (when the user has started his session from z console), JESINPUT (when the subject is started as a job entered via JES), or SERVAUTH (when the user has started his session via a network server application). Additional data is associated with this security attribute depending on the input class (e. g. the terminal ID in case of a terminal or the network zone in case the Port of Entry type is SERVAUTH).**

7.1.5 Security management (FMT)

7.1.5.1 Management of object security attributes: persistent objects (FMT_MSA.1(PSO))

FMT_MSA.1.1

1. **The TSF shall enforce the Persistent Storage Object Access Control Policy to restrict the ability to modify ~~*[assignment: other operations]*~~ the security attributes of the objects covered by the SFP to the owner of the object and**

a) **For non-UNIX, non-LDAP objects:**

   1. **users with the SPECIAL attribute or the appropriate group-SPECIAL attribute and**
   2. **users who have ALTER authority to the object**

b) **For UNIX objects a user with z/OS UNIX superuser privilege**

c) **For LDAP LDBM objects:**

   1. **The directory Administrator**
   2. **Users with DAC authority to move or rename an object and**
   3. **Users with write authority to restricted attributes in the object.**

Application note: *The required selection operation was performed by selecting the assignment operation available. As no other actions are foreseen, a respective refinement operation deleting that assignment was performed to demonstrate the absence of such operation.*

7.1.5.2 Management of object security attributes: transient objects (FMT_MSA.1(TSO))

FMT_MSA.1.1 The TSF shall enforce the Transient Storage Object Access Control Policy to restrict the ability to modify ~~*[assignment: other operations]*~~ the security attributes of the objects covered by the SFP to the owner of the object and **for z/OS UNIX IPC objects to a user with z/OS UNIX superuser privilege**.

Application note: *The required selection operation was performed by selecting the assignment operation available. As no other actions are foreseen, a respective refinement operation deleting that assignment was performed to demonstrate the absence of such operation.*

7.1.5.3 Management of object security attributes: labeled security (FMT_MSA.1(LS) (Labeled Security Mode only))

FMT_MSA.1.1 The TSF shall enforce the *mandatory access control policy* ~~Multilevel Confidentiality Information Flow Control Policy~~ to restrict the ability to modify the label-related object security attributes to **users with the SPECIAL attribute**.

7.1.5.4 Static attribute initialization: persistent objects (FMT_MSA.3(PSO))

FMT_MSA.3.1 The TSF shall enforce the Persistent Storage Object Access Control Policy to provide restrictive default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 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.

7.1.5.5 Static attribute initialization: transient objects (FMT_MSA.3(TSO))

FMT_MSA.3.1 The TSF shall enforce the Transient Storage Object Access Control Policy to provide restrictive default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 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.

7.1.5.6 Static attribute initialization: network (FMT_MSA.3(NI))

FMT_MSA.3.1 The TSF shall enforce the Network Information Flow Control Policy to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the **users with READ access to the appropriate profiles in the SERVAUTH class as specified in section "Communication Server ipsec Command Interface"** to specify alternative initial values to override the default values when an object or information is created.

Application note: *The ability to perform IPSec, IP filtering, and defensive filtering related network management functions can be delegated to users by providing them READ access to the profile EZB.IPSECCMD.sysname.tcpprocname.CONTROL in the SERVAUTH class of RACF. A list of profiles and the network management functions they protect can be found in Network configuration and management.*

7.1.5.7 Static attribute initialization: labeled security (FMT_MSA.3(LS) (Labeled Security Mode only))

FMT_MSA.3.1 The TSF shall enforce the *mandatory access control policy* ~~Multilevel Confidentiality Information Flow Control Policy~~ to provide restrictive default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 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.

7.1.5.8 Security attribute value inheritance: persistent objects (FMT_MSA.4(PSO))

FMT_MSA.4.1 The TSF shall use the following rules to set the value of security attributes for Persistent Storage Objects:

MVS objects:

Object security attributes for MVS objects are stored in RACF profiles. MVS objects inherit their object security attributes from the RACF profile that most closely matches the object's name.

LDAP LDBM objects:

LDAP LDBM objects can inherit their ACLs and owner attributes from nodes above them in the directory hierarchy. Inheritance is controlled by the aclPropagate and ownerPropagate attributes. An LDBM object's ACL and owner attribute marked with these values inherit their respective attribute to objects below in the hierarchy, unless another ACL or owner attribute with the aclPropagate or ownerPropagate value is encountered, or the ACL or owner attributes for the object have been set explicitly.

7.1.5.9 Management of TSF data: audit events (FMT_MTD.1(AE))

FMT_MTD.1.1 The TSF shall restrict the ability to query, modify the set of audited events to

a) **users with the AUDITOR role**

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

Application note: *This SFR applies to FAU_SEL.1.*

7.1.5.10 Management of TSF data: audit storage (FMT_MTD.1(AS))

FMT_MTD.1.1 The TSF shall restrict the ability to clear, **create, delete** the audit storage to **users that satisfy the following rules:**

a) **users with the AUDITOR role**

b) **z/OS operators.**

Application note: *This SFR applies to FAU_STG.1.*

Application note: *The required selection was not performed since the term modify already includes add and delete of parameters. To demonstrate the absence of that selection, a respective refinement operation was performed.*

7.1.5.11 Management of TSF data: audit trail threshold (FMT_MTD.1(AT))

FMT_MTD.1.1 The TSF shall restrict the ability to modify~~, *[selection: add, delete]*~~ the

a) threshold of the audit trail when an action is performed;

b) action when the threshold is reached

to **the authorized administrator**.

Application note: *This SFR applies to FAU_STG.3.*

7.1.5.12 Management of TSF data: audit storage failure (FMT_MTD.1(AF))

FMT_MTD.1.1 The TSF shall restrict the ability to modify~~, [selection: add, delete]~~ the actions to be taken in case of audit storage failure to **the authorized administrator**.

Application note: *This SFR applies to FAU_STG.4.*

Application note: *The required selection was not performed since the term modify already includes add and delete of parameters. To demonstrate the absence of that selection, a respective refinement operation was performed.*

7.1.5.13 Management of TSF data: network filters (FMT_MTD.1(NI))

FMT_MTD.1.1 The TSF shall restrict the ability to query, modify, delete, **perform command-driven management functions for IPSec, IP filtering, and defensive filtering configuration related to** the security attributes for the rules governing the

a) identification of network data;

b) actions performed on the identified network data

to **users with READ access to the appropriate profiles in the SERVAUTH class as specified in section "Communication Server ipsec Command Interface".**

Application note: *This SFR applies to FDP_IFF.1(NI).*

Application note: *The ability to perform IPSec, IP filtering, and defensive filtering related network management functions can be delegated to users by providing them READ access to profiles of the form EZB.IPSECCMD.sysname.[one of: tcpname, clientname, sysname, DMD, GLOBAL] (potentially followed by a function name) in the SERVAUTH class of RACF. A list of profiles and the network management function they protect can be found in section "Network configuration and management".*

7.1.5.14 Management of TSF data: IPSec (FMT_MTD.1(NI2))

FMT_MTD.1.1 The TSF shall restrict the ability to **perform management functions for** the **IPSec network configuration** to **users with READ access to the appropriate profiles in the SERVAUTH class as specified in section Communication Server Network Management Interface.**

Application note: *The ability to perform IPSec related network management functions can be delegated to users by providing them READ access to profiles of the form EZB.NETMGMT.sysname.[one of: tcpname, clientname, sysname] (potentially followed by a function name) in the SERVAUTH class of RACF. A list of profiles and the network management function they protect can be found in section "Network configuration and management".*

7.1.5.15 Management of TSF data: authentication threshold (FMT_MTD.1(IAT))

FMT_MTD.1.1 The TSF shall restrict the ability to modify the threshold for unsuccessful authentication attempts to **users that satisfy 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.**

Application note: *This SFR applies to FIA_AFL.1.*

7.1.5.16 Management of TSF data: account re-enablement (FMT_MTD.1(IAF))

FMT_MTD.1.1 The TSF shall restrict the ability to re-enable the authentication to the account subject to authentication failure to **users that satisfy 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**

In addition, the following users may change the REVOKE status and the user's password or password phrase:

- **c) user has READ access to FACILITY resource IRR.PASSWORD.RESET assuming the revoked user does not have the SPECIAL, OPERATIONS, or AUDITOR attributes.**
- **d) user has READ access to FACILITY resource**

IRR.PWRESET.OWNER.owner-of-profile where "owner-of-profile" specifies the user or group that owns the revoked user, and the revoked user does not have the SPECIAL, OPERATIONS, or AUDITOR attributes.

e) **user has READ access to FACILITY resource IRR.PWRESET.TREE.owner-of-tree where "owner-of-tree" specifies a user or group that would have group-SPECIAL over the revoked user, and the revoked user does not have the SPECIAL, OPERATIONS, or AUDITOR attributes.**

Application note: *This SFR applies to FIA_AFL.1.*

7.1.5.17 Management of TSF data: user security attributes (FMT_MTD.1(IAU))

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

a) **user has SPECIAL**

b) **users with CLAUTH attribute for the USER class**

c) **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**

d) **LDAP administrators for administration of LDAP-based users, groups, and roles.**

Application note: *This SFR applies to FIA_ATD.1, FIA_UAU.1, FIA_UID.1.*

7.1.5.18 Management of TSF data: authentication data (FMT_MTD.1(IAU-AUTH))

FMT_MTD.1.1 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.**

Application note: *This SFR applies to FIA_ATD.1, FIA_UAU.1, FIA_UID.1.*

7.1.5.19 Management of TSF data: EIA (FMT_MTD.1(EIA))

FMT_MTD.1.1 The TSF shall restrict the ability to initialize, modify, delete the user security attributes used for the remote identification and authentication policy to **the authorized administrator, or users that satisfy 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**

In addition, the following users may reset the REVOKE status of a revoked user account (re-enable the account) and the user's password or password phrase:

c) **user has READ access to FACILITY resource IRR.PASSWORD.RESET assuming the revoked user does not have the SPECIAL, OPERATIONS, or AUDITOR attributes.**

d) **user has READ access to FACILITY resource IRR.PWRESET.OWNER.owner-of-profile where "owner-of-profile" specifies the user or group that owns the revoked user, and the revoked user does not have the SPECIAL, OPERATIONS, or AUDITOR attributes.**

e) **user has READ access to FACILITY resource IRR.PWRESET.TREE.owner-of-tree where "owner-of-tree" specifies a user or group that would have group-SPECIAL**

over the revoked user, and the revoked user does not have the SPECIAL, OPERATIONS, or AUDITOR attributes.

7.1.5.20 Management of TSF data: key import (FMT_MTD.1(CRYPTO1))

FMT_MTD.1.1 The TSF shall restrict the ability to **import or modify** the **cryptographic keys** to **the authorized administrator**.

Application note: *The process of a user requesting a certificate from PKI Services involves the user sending a public key to the PKI server. Similarly, authentication of a client via SSL/TLS involves the client sending a public key to the server. For the purposes of this ST, neither of those operations, nor other operations similar to them, are considered to be importation of a cryptographic key.*

7.1.5.21 Management of TSF data: digital certificates (FMT_MTD.1(CRYPTO2))

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

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 Authority Checking for PKCS#11 Cryptographic Tokens in the ICSF TKDS. That chapter 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(1).*

7.1.5.22 Management of TSF data: additional configuration (FMT_MTD.1(ADD))

FMT_MTD.1.1 The TSF shall restrict the ability to **initialize or change** ~~the~~ **additional TOE configuration parameters** to **authorized administrators**.

Application note: *This includes configuration information such as network configuration associated with the TCP/IP stack, as well as LDAP server configuration, FTP server configuration, HTTP server configuration, PKI Services configuration and management,* *basic system configuration information, etc.*

7.1.5.23 Revocation: objects (FMT_REV.1(OBJ))

FMT_REV.1.1 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:**

a) users authorized to modify the security attributes covered by the Persistent Storage Object Access Control Policy or the Transient Storage Object Access Control Policy, or (in Labeled Security Mode) the mandatory access control policy.

FMT_REV.1.2 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.

Application note: *For the access rights to data sets, z/OS UNIX file system objects, volumes, terminals, and TCP/IP connections, the access checks are performed once when the user starts to use the resource and are not checked again until the user releases the resource and attempts to use it again. Immediate revocation for these attributes can be achieved by terminating all active jobs of the user, his TSO sessions and all the z/OS UNIX processes acting on behalf of this user.*

7.1.5.24 Revocation: users (FMT_REV.1(USR))

FMT_REV.1.1 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 roles as defined by FMT_SMR.1 bullets a), d), f), l), n), o) and r)**.

Application note: *z/OS has several kinds of authorized administrators, including users with SPECIAL and group-SPECIAL attributes, as well as owners and users with authority to change another user's password/phrase. All of these can, in some sense, revoke some or all of a user's security attributes. Additionally, via PKI Services, users who own a digital certificate may request revocation of their certificate, and posting of that certificate to the certificate revocation list (CRL) maintained by PKI Services.*

FMT_REV.1.2 The TSF shall enforce the following rules:

a) The enforcement of the revocation of security-relevant authorizations with the next user-subject binding process during the next authentication of the user;

b) **the immediate revocation of security-relevant authorization**.

7.1.5.25 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;

b) Management of cryptographic network protocols;

c) Management of Persistent Storage Object Access Control Policy;

d) Management of Transient Storage Object Access Control Policy;

e) Management of Network Information Flow Control Policy;

f) Management of identification and authentication policy;

g) Management of user security attributes;

h) Management of cryptographic keys;

i) Management of digital certificates;

j) Management of other TOE configuration data.

7.1.5.26 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles:

a) User role with the following rights:

1. Users are authorized to modify their own user password;
2. Users are authorized to modify the access control permissions for the named objects they own;

b) users authorized by the discretionary access control policy to modify object security attributes;

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

d) users authorized to modify their own authentication data;

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

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

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

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

i) **Operations roles (users with the OPERATIONS attribute)**

j) **z/OS operators (users who are allowed to issue operator commands)**

k) **z/OS pseudo-user (protected user IDs used for executing defined started tasks, and for “anonymous” access to administrator-specified data via HTTP or LDAP)**

l) **z/OS UNIX superuser**

m) **LDAP Administrator (as specified in the LDAP configuration file)**

n) **PKI Services Administrator (as specified in the PKI Services configuration file)**

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

p) **Users authorized to perform IPSec network management functions based on access rights to RACF profiles protecting the individual management operations**

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

r) **authorized administrator (user with the SPECIAL attribute).**

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

7.1.6 Protection of the TSF (FPT)

7.1.6.1 Reliable time stamps (FPT_STM.1)

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

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

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

FPT_TDC.1.2 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.*

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

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret label-related security attributes, **no other TSF data** when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 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.*

7.1.7 TOE access (FTA)

7.1.7.1 TSF-initiated session locking (FTA_SSL.1)

FTA_SSL.1.1 The TSF shall lock an interactive session to a human user maintained by the TSF after **an administrator-defined time interval of user inactivity** by:

a) clearing or overwriting TSF controlled display devices, making the current contents unreadable;

b) disabling any activity of the user's TSF controlled data access/TSF controlled display devices other than unlocking the session.

FTA_SSL.1.2 The TSF shall require the following events to occur prior to unlocking the session:

a) Successful re-authentication with the credentials of the user owning the session using **one of the authentication methods out of the list of allowed methods specified in FIA_UAU.5**;

b) **no other events**.

Application note: *This SFR applies to directly attached terminals, not networked sessions.*

Application note: *It is possible that the TSF establishes a connection to a session on a remote trusted IT system, for example when using SSH. This remote trusted IT system maintains the session established with the communication channel. The locking requirement however applies to the session maintained by the TSF only as the TSF can only exercise control of the sessions it maintains.*

7.1.7.2 User-initiated locking (FTA_SSL.2)

FTA_SSL.2.1 The TSF shall allow user-initiated locking of the user's own interactive session maintained by the TSF, by:

a) clearing or overwriting TSF controlled display devices, making the current contents unreadable;

b) disabling any activity of the user's TSF controlled data access/TSF controlled display devices other than unlocking the session.

FTA_SSL.2.2 The TSF shall require the following events to occur prior to unlocking the session:

a) Successful re-authentication with the credentials of the user owning the session using **one of the authentication methods from the list of allowed methods specified in FIA_UAU.5**;

b) no other events.

Application note: *This SFR applies to directly attached terminals, not networked sessions.*

Application note: *See also application note for FTA_SSL.1*

7.1.8 Trusted path/channels (FTP)

7.1.8.1 Inter-TSF trusted channel (FTP_ITC.1)

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification and disclosure using the following mechanisms:

a) Cryptographically-protected communication channel using **SSLv3, TLSv1, TLSv1.1, SSHv2, GSSAPI with message privacy functions using the Kerberos v5 mechanism, or IPSec protocols offered by TOE services**;

b) **physically protected communication channel*****s provided by the TOE environment***.

FTP_ITC.1.2 The TSF shall permit **the TSF, another trusted IT product** to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for all security functions specified in the ST that interact with remote trusted IT systems and **no other functions and conditions**.

7.2 Security Functional Requirements Rationale

7.2.1 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.1	O.AUDITING
FAU_GEN.2	O.AUDITING
FAU_SAR.1	O.AUDITING
FAU_SAR.2	O.AUDITING
FAU_SAR.3	O.AUDITING
FAU_SEL.1	O.AUDITING
FAU_STG.1	O.AUDITING
FAU_STG.3	O.AUDITING
FAU_STG.4	O.AUDITING

Security Functional Requirements	**Objectives**
FCS_CKM.1(SYM)	O.CRYPTO.NET
FCS_CKM.1(RSA)	O.CRYPTO.NET
FCS_CKM.1(DSA)	O.CRYPTO.NET
FCS_CKM.2(NET)	O.CRYPTO.NET
FCS_CKM.4	O.CRYPTO.NET
FCS_COP.1(SGN)	O.CRYPTO.NET
FCS_COP.1(NET)	O.CRYPTO.NET
FCS_RNG.1	O.CRYPTO.NET
FDP_ACC.1(PSO)	O.DISCRETIONARY.ACCESS
FDP_ACC.1(TSO)	O.SUBJECT.COM
FDP_ACF.1(PSO-MVS)	O.DISCRETIONARY.ACCESS
FDP_ACF.1(PSO-UNIX)	O.DISCRETIONARY.ACCESS
FDP_ACF.1(PSO-LDAP)	O.DISCRETIONARY.ACCESS
FDP_ACF.1(TSO)	O.SUBJECT.COM
FDP_ETC.1 (Labeled Security Mode only)	O.LS.CONFIDENTIALITY
FDP_ETC.2(LS) (Labeled Security Mode only)	O.LS.CONFIDENTIALITY, O.LS.PRINT
FDP_IFC.2(LS) (Labeled Security Mode only)	O.LS.CONFIDENTIALITY
FDP_IFC.2(NI)	O.NETWORK.FLOW
FDP_IFF.1(NI)	O.NETWORK.FLOW

Security Functional Requirements	**Objectives**
FDP_IFF.2(LS) (Labeled Security Mode only)	O.LS.CONFIDENTIALITY
FDP_ITC.1(LS) (Labeled Security Mode only)	O.LS.CONFIDENTIALITY, O.LS.LABEL
FDP_ITC.2	O.DISCRETIONARY.ACCESS, O.NETWORK-FLOW, O.SUBJECT.COM
FDP_ITC.2(LS) (Labeled Security Mode only)	O.LS.CONFIDENTIALITY, O.LS.LABEL
FDP_RIP.2	O.AUDITING, O.CRYPTO.NET, O.DISCRETIONARY.ACCESS, O.I&A, O.NETWORK.FLOW, O.SUBJECT.COM
FDP_RIP.3	O.AUDITING, O.CRYPTO.NET, O.DISCRETIONARY.ACCESS, O.I&A, O.NETWORK.FLOW, O.SUBJECT.COM
FIA_AFL.1	O.I&A
FIA_ATD.1(HU)	O.I&A
FIA_ATD.1(TU)	O.NETWORK.FLOW
FIA_ATD.1(EIA)	O.I&A.REMOTE
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

Security Functional Requirements	**Objectives**
FIA_UAU.7	O.I&A
FIA_UAU.8(EIA)	O.I&A.REMOTE
FIA_UID.1	O.I&A, O.NETWORK.FLOW
FIA_UID.3(EIA)	O.I&A.REMOTE
FIA_USB.1(LS) (Labeled Security Mode only)	O.LS.LABEL
FIA_USB.2	O.I&A
FMT_MSA.1(PSO)	O.MANAGE
FMT_MSA.1(TSO)	O.MANAGE
FMT_MSA.1(LS) (Labeled Security Mode only)	O.LS.LABEL
FMT_MSA.3(PSO)	O.MANAGE
FMT_MSA.3(TSO)	O.MANAGE
FMT_MSA.3(NI)	O.MANAGE
FMT_MSA.3(LS) (Labeled Security Mode only)	O.LS.LABEL
FMT_MSA.4(PSO)	O.MANAGE
FMT_MTD.1(AE)	O.MANAGE
FMT_MTD.1(AS)	O.MANAGE
FMT_MTD.1(AT)	O.MANAGE
FMT_MTD.1(AF)	O.MANAGE
FMT_MTD.1(NI)	O.MANAGE
FMT_MTD.1(NI2)	O.MANAGE

Security Functional Requirements	Objectives
FMT_MTD.1(IAT)	O.MANAGE
FMT_MTD.1(IAF)	O.MANAGE
FMT_MTD.1(IAU)	O.MANAGE
FMT_MTD.1(IAU-AUTH)	O.MANAGE
FMT_MTD.1(EIA)	O.I&A.REMOTE
FMT_MTD.1(CRYPTO1)	O.MANAGE
FMT_MTD.1(CRYPTO2)	O.MANAGE
FMT_MTD.1(ADD)	O.MANAGE
FMT_REV.1(OBJ)	O.MANAGE
FMT_REV.1(USR)	O.MANAGE
FMT_SMF.1	O.MANAGE
FMT_SMR.1	O.MANAGE
FPT_STM.1	O.AUDITING
FPT_TDC.1	O.DISCRETIONARY.ACCESS, O.NETWORK.FLOW, O.SUBJECT.COM
FPT_TDC.1(LS) (Labeled Security Mode only)	O.LS.CONFIDENTIALITY, O.LS.LABEL
FTA_SSL.1	O.I&A
FTA_SSL.2	O.I&A
FTP_ITC.1	O.TRUSTED_CHANNEL

Table 8: Mapping of security functional requirements to security objectives

7.2.2 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.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), while all other users are denied access to the audit records (FAU_SAR.2). Audit trails can be searched for events belonging to users, objects, or labels(FAU_SAR.3). The authorized user must have the capability to specify which audit records are generated (FAU_SEL.1). The TOE prevents the audit log from being modified or deleted (FAU_STG.1) and ensures that the audit log is not lost due to resource shortage (FAU_STG.3, FAU_STG.4). To support auditing, the TOE is able to maintain proper time stamps (FPT_STM.1). The protection of reused resources ensures that no data leaks from other protected sources (FDP_RIP.2, FDP_RIP.3).
O.CRYPTO.NET	The cryptographically-protected network protocol (FCS_COP.1_NET) is supported by the generation of symmetric keys (FCS_CKM.1_SYM), as well as asymmetric keys (FCS_CKM.1_RSA, FCS_CKM.1_DSA). Key generation is supported by the provision of good-quality random numbers (FCS_RNG.1). As part of the cryptographic network protocol, the TOE securely exchanges the symmetric key with a remote trusted IT system (FCS_CKM.2_NET). The TOE ensures that all keys are zeroized upon de-allocation (FCS_CKM.4). The integrity protection is supported by the appropriate digital signatures (FCS_COP.1_SGN). The protection of reused resources ensures that no data leaks from other protected sources (FDP_RIP.2, FDP_RIP.3).
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. The access control policy must have a defined scope of

Security objectives	Rationale
	control (FDP_ACC.1_PSO). The rules for the access control policy are defined (FDP_ACF.1_PSO-MVS, FDP_ACF.1_PSO-UNIX, FDP_ACF.1_PSO-LDAP). When import of user data is allowed, the TOE must ensure that user data security attributes required by the access control policy are correctly interpreted (FDP_ITC.2, FPT_TDC.1). The protection of reused resources ensures that no data leaks from other protected sources (FDP_RIP.2, FDP_RIP.3).
O.NETWORK.FLOW	The packet filter mechanism controls the information flowing between different entities (FDP_IFC.2_NI). The TOE implements a rule-set governing the information flow (FDP_IFF.1_NI). To facilitate the information flow control, the information must be identified (FIA_UID.1) based on security attributes the TOE can maintain (FIA_ATD.1_TU). The TOE must ensure that security attributes of the network data required by the information flow control policy are correctly interpreted (FDP_ITC.2, FPT_TDC.1). The protection of reused resources ensures that no data leaks from other protected sources (FDP_RIP.2, FDP_RIP.3).
O.SUBJECT.COM	The TSF must control the exchange of data using transient storage objects between subjects based on the identity of users. The access control policy must have a defined scope of control (FDP_ACC.1_TSO). The rules for the access control policy are defined (FDP_ACF.1_TSO). When import of user data is allowed, the TOE must ensure that user data security attributes required by the access control policy are correctly interpreted (FDP_ITC.2, FPT_TDC.1). The protection of reused resources ensures that no data leaks from other protected sources (FDP_RIP.2, FDP_RIP.3).
O.I&A	The TSF must ensure that only authorized users gain access to the TOE and its resources. Users authorized to access the TOE must use an identification and authentication process (FIA_UID.1, FIA_UAU.1). 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

Security objectives	Rationale
	is capable of identifying and reacting to unsuccessful authentication attempts (FIA_AFL.1). In addition, user-initiated and TSF-initiated session locking (FTA_SSL.1, FTA_SSL.2) protect the authenticated user's session. The protection of reused resources ensures that no data leaks from other protected sources (FDP_RIP.2, FDP_RIP.3).
O.MANAGE	The TOE provides management interfaces globally defined in FMT_SMF.1 for: • the access control policies (FMT_MSA.1_PSO, FMT_MSA.1_TSO, FMT_MSA.3_PSO, FMT_MSA.3_TSO); • the information flow control policy (FMT_MSA.3_NI, FMT_MTD.1_NI, FMT_MTD.1_NI2); • the auditing aspects (FMT_MTD.1_AE, FMT_MTD.1_AS, FMT_MTD.1_AT, FMT_MTD.1_AF); • the identification and authentication aspects (FMT_MTD.1_IAT, FMT_MTD.1_IAF, FMT_MTD.1_IAU, FMT_MTD.1_IAU-AUTH); • the management of cryptographic functions, especially for cryptographic keys (FMT_MTD.1_CRYPTO1) and digital certificates (FMT_MTD.1_CRYPTO2); • the configuration of networks and network services (FMT_MTD.1_ADD). Persistently stored user data is stored either in hierarchical or relational fashion, which implies an inheritance of security attributes from parent objects (FMT_MSA.4_PSO). 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_OBJ, FMT_REV.1_USR).

Security objectives	Rationale
O.TRUSTED_CHANNEL	The TOE provides a trusted channel protecting communication between a remote trusted IT system and itself (FTP_ITC.1).
O.I&A.REMOTE	The remotely triggerable I&A policy is defined by the identification mechanism (FIA_UID.3_EIA), the authentication mechanism (FIA_UAU.8_EIA) and the specification of the security attributes applicable to this policy (FIA_ATD.1_EIA). The management aspect of this I&A policy is covered by FMT_MTD.1_EIA.
O.I&A.MULTIPLE	The TOE shall provide multiple I&A policies, at least one for local and one for remote I&A which is specified with FIA_UAU.5.
O.LS.CONFIDENTIALITY	The information flow control policy is defined by specifying the subjects, objects, security attributes and rules in FDP_IFC.2_LS and FDP_IFF.2_LS. Supportive to the enforcement of the policy are the automated label assignment when exporting data (FDP_ETC.1, FDP_ETC.2_LS) and during the import of data (FDP_ITC.1_LS, FDP_ITC.2_LS). For assigning labels to imported data, the label information transmitted with the data must be interpretable by the TOE (FPT_TDC.1_LS).
O.LS.PRINT	The addition of label information on exported data during printing is directly addressed by FDP_ETC.2_LS.
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). Object labels are assigned to objects when importing them into the TOE (FDP_ITC.1_LS, FDP_ITC.2_LS, FPT_TDC.1_LS). The management of labels is allowed for the TOE with (FMT_MSA.1_LS, FMT_MSA.3_LS).

Table 9: Security objectives for the TOE rationale

7.2.3 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.1	FPT_STM.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1	FAU_GEN.1
	FIA_UID.1	FIA_UID.1
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1
FAU_SAR.2	FAU_SAR.1	FAU_SAR.1
FAU_SAR.3	FAU_SAR.1	FAU_SAR.1
FAU_SEL.1	FAU_GEN.1	FAU_GEN.1
	FAU_MTD.1	FMT_MTD.1(AE)
FAU_STG.1	FAU_GEN.1	FAU_GEN.1
FAU_STG.3	FAU_STG.1	FAU_STG.1
FAU_STG.4	FAU_STG.1	FAU_STG.1
FCS_CKM.1(SYM)	FCS_COP.1	FCS_COP.1(NET)
	FCS_CKM.4	FCS_CKM.4
	FCS_RNG.1	FCS_RNG.1

Security Functional Requirement	**Dependencies**	**Resolution**
FCS_CKM.1(RSA)	FCS_COP.1	FCS_COP.1(NET)
	FCS_CKM.4	FCS_CKM.4
	FCS_RNG.1	FCS_RNG.1
FCS_CKM.1(DSA)	FCS_COP.1	FCS_COP.1(NET)
	FCS_CKM.4	FCS_CKM.4
	FCS_RNG.1	FCS_RNG.1

FCS_CKM.2(NET)	FCS_CKM.1	FCS_CKM.1(SYM)
		FCS_CKM.1(RSA)
		FCS_CKM.1(DSA)

	FCS_CKM.4	FCS_CKM.4
	FCS_RNG.1	FCS_RNG.1
FCS_CKM.4	FCS_CKM.1	FCS_CKM.1(SYM)

FCS_COP.1(SGN)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(RSA)
		FCS_CKM.1(DSA)

	FCS_CKM.4	FCS_CKM.4

FCS_COP.1(NET)	FCS_CKM.1	FCS_CKM.1(SYM)
		FCS_CKM.1(RSA)
		FCS_CKM.1(DSA)

Security Functional Requirement	Dependencies	Resolution
	FCS_CKM.4	FCS_CKM.4
FCS_RNG.1	No dependencies.	
FDP_ACC.1(PSO)	FDP_ACF.1	FDP_ACF.1(PSO-UNIX)
FDP_ACC.1(TSO)	FDP_ACF.1	FDP_ACF.1(TSO)
FDP_ACF.1(PSO-MVS)	FDP_ACC.1	FDP_ACC.1(PSO)
	FDP_MSA.3	FMT_MSA.3(PSO)
FDP_ACF.1(PSO-UNIX)	FDP_ACC.1	FDP_ACC.1(PSO)
	FDP_MSA.3	FMT_MSA.3(PSO)
FDP_ACF.1(PSO-LDAP)	FDP_ACC.1	FDP_ACC.1(PSO)
	FDP_MSA.3	FMT_MSA.3(PSO)
FDP_ACF.1(TSO)	FDP_ACC.1	FDP_ACC.1(TSO)
	FDP_MSA.3	FMT_MSA.3(TSO)
FDP_ETC.1 (Labeled Security Mode only)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(PSO)

Security Functional Requirement	**Dependencies**	**Resolution**
FDP_ETC.2(LS) (Labeled Security Mode only)	FDP_IFC.1	FDP_IFC.2(LS) (Labeled Security Mode only)
FDP_IFC.2(LS) (Labeled Security Mode only)	FDP_IFF.1	FDP_IFF.2(LS) (Labeled Security Mode only)
FDP_IFC.2(NI)	FDP_IFF.1	FDP_IFF.1(NI)
FDP_IFF.1(NI)	FDP_IFC.1	FDP_IFC.2(NI)
	FDP_MSA.3	FMT_MSA.3(NI)
FDP_IFF.2(LS) (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)
FDP_ITC.1(LS) (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)

FDP_ITC.2	FDP_ACC.1	FDP_ACC.1(PSO)
		FDP_ACC.1(TSO)

	FDP_IFC.1	FDP_IFC.2(NI)
	FDP_ITC.1	FTP_ITC.1
	FDP_TDC.1	FPT_TDC.1

Security Functional Requirement	**Dependencies**	**Resolution**
FDP_ITC.2(LS) (Labeled Security Mode only)	FDP_IFC.1	FDP_IFC.2(LS) (Labeled Security Mode only)
	FTP_ITC.1	FTP_ITC.1
	FPT_TDC.1	FPT_TDC.1(LS) (Labeled Security Mode only)
FDP_RIP.2	No dependencies.	
FDP_RIP.3	No dependencies.	
FIA_AFL.1	FIA_UAU.1	FIA_UAU.1
FIA_ATD.1(HU)	No dependencies.	
FIA_ATD.1(TU)	No dependencies.	
FIA_ATD.1(EIA)	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_UAU.8(EIA)	FTP_ITC.1	FTP_ITC.1
FIA_UID.1	No dependencies.	
FIA_UID.3(EIA)	FTP_ITC.1	FTP_ITC.1
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)
		FIA_ATD.1(TU)

FMT_MSA.1(PSO)	FMT_ACC.1	FDP_ACC.1(PSO)
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.1(TSO)	FDP_ACC.1	FDP_ACC.1(TSO)
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.1(LS) (Labeled Security Mode only)	FMT_IFC.1	FDP_IFC.2(LS) (Labeled Security Mode only)
	FMT_SMR.1	FMT_SMR.1

Security Functional Requirement	**Dependencies**	**Resolution**
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.3(PSO)	FMT_MSA.1	FMT_MSA.1(PSO)
	FMT_SMR.1	FMT_SMR.1
FMT_MSA.3(TSO)	FMT_MSA.1	FMT_MSA.1(TSO)
	FMT_SMR.1	FMT_SMR.1
FMT_MSA.3(NI)	FMT_MSA.1	FMT_MTD.1(NI) is specified to require the management of security attributes for the Network Information Flow Control Policy, just as a potential FMT_MSA.1(PF) would have been specified. However, the Network Information Flow Control Policy is not required to be enforced when managing the security attributes, as the management aspect of the packet filtering functionality is not protected by the packet filter. Therefore, FMT_MSA.1 is not applicable and is replaced with FMT_MTD.1(NI).
	FMT_SMR.1	FMT_SMR.1
FMT_MSA.3(LS) (Labeled Security Mode only)	FMT_MSA.1	FMT_MSA.1(LS) (Labeled Security Mode only)
	FMT_SMR.1	FMT_SMR.1
FMT_MSA.4(PSO)	FMT_ACC.1	FDP_ACC.1(PSO)

Security Functional Requirement	Dependencies	Resolution
FMT_MTD.1(AE)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(AS)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(AT)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(AF)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(NI)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(NI2)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(IAT)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1

Security Functional Requirement	Dependencies	Resolution
FMT_MTD.1(IAF)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(IAU)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(IAU-AUTH)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(EIA)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(CRYPTO1)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(CRYPTO2)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(ADD)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1

Security Functional Requirement	Dependencies	Resolution
FMT_REV.1(OBJ)	FMT_SMR.1	FMT_SMR.1
FMT_REV.1(USR)	FMT_SMR.1	FMT_SMR.1
FMT_SMF.1	No dependencies.	
FMT_SMR.1	FMT_UID.1	FIA_UID.1
FPT_STM.1	No dependencies.	
FPT_TDC.1	No dependencies.	
FPT_TDC.1(LS) (Labeled Security Mode only)	No dependencies.	
FTA_SSL.1	FTA_UAU.1	FIA_UAU.1
FTA_SSL.2	FTA_UAU.1	FIA_UAU.1
FTP_ITC.1	No dependencies.	

Table 10: TOE SFR dependency analysis

7.2.4 TSF Rationale

The following table maps the security functional requirements to the security functions as defined in the TOE summary specification to show that all security functional requirements are addressed by the security functions.

SFR	Security Functions
FAU_GEN.1	Section Generation of audit records explains how audit

SFR	Security Functions
	records are generated. This section also explains the structure of the audit records.
FAU_GEN.2	Section Generation of audit records 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 User roles and attributes and section AUDITOR and group-AUDITOR explain the auditor role. Section Protection of the audit trail 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_SAR.2	Section Protection of the audit trail explains how to protect the audit trail from unauthorized access.
FAU_SAR.3	Section Generation of audit records explains how to search the audit records. Section Using MVS Data Sets for SMF and Using a System Log Stream for SMF explain the IFASMFDP and IFASMFDL programs for unloading selected audit records.
FAU_SEL.1	Sections Audit configuration and management and AUDITOR and group-AUDITOR explain how the auditor role can configure the events that are audited. These chapters also explain that the owner of a profile can define which events related to the profile are audited.
FAU_STG.1	Section Protection of the audit trail explains how to protect the audit trail from unauthorized access.
FAU_STG.3	Section Using MVS Data Sets for SMF explains how the operator is informed about the fact that a SMF data set is full and the TOE has switched to the next non-full SMF data set.
FAU_STG.4	Section Audit configuration and management explains how the TOE prevents the loss of audit data by halting the system on audit trail exhaustion.
FCS_CKM.1(SYM), FCS_CKM.2(NET)	Section System SSLSystem SSL explains the use of the SSL/TLS protocols for the protection of communication links. Section Communications Server explains the use of the IPSec protocol for the protection of communication links by

SFR	Security Functions
	reference to the appropriate IETF standards. This discussion includes (by reference to the IETF standards) usage of HMAC-SHA-1 for integrity protection of the communication links. Section IBM Ported Tools for z/OS (OpenSSH) explains the use of the SSH protocols for the protection of communication links. Section Network Authentication Service explains the use of the Kerberos and GSSAPI protocols for the protection of communication links.
FCS_CKM.1(DSA), FCS_CKM.1(RSA)	Generation of RSA and DSA host keys for SSH is explained in IBM Ported Tools for z/OS (OpenSSH) The generation of RSA and DSA public/private key pairs using the RACDCERT command is explained in section Digital Certificates, Key Rings, and Certificate Mappings in RACF and PKCS#11 Cryptographic Tokens.
FCS_CKM.4	As pointed out in the SFR's application note, cryptographic keys for SSL/TLS, Kerberos, SSH, and IPsec sessions are protected by the TOE against unauthorized access and are destroyed by the object re-use functions of the TOE. The object reuse function is described in Object reuse.
FCS_COP.1(SGN)	Signature creation and verification is explained for the different usage scenarios as follows: • TLS/SSL: System SSL • IPSec ISAKMP session establishment: Communications Server by reference to the applicable RFCs (2408 and others) and mentioning that IKE daemons can use NSS to perform RSA signature generation and verification. • RACDCERT: Digital Certificates, Key Rings, and Certificate Mappings in RACF and PKCS#11 Cryptographic Tokens • PKI Services PKI Services In addition, Communications Server explains that signature creation and verification can also be performed through the Network Security Services (NSS) on behalf of trusted remote XMLAppliance clients.
FCS_COP.1(NET)	Cryptographic operations are explained for the different

SFR	Security Functions
	protocols in the respective TSS sections for these protocols: • TLS/SSL: System SSL • IPSec: Communications Server • SSH: IBM Ported Tools for z/OS (OpenSSH) • Kerberos: Network Authentication Service
FCS_RNG.1	Random numbers are provided either through the TOE environment by the functionality of the crypto cards, or by the TOE's software. The two software components providing random numbers for key generation are System SSL and OpenSSH. An analysis of the functionality and quality of the random number generators in these components is provided in a separate document.

FDP_ACC.1(PSO)
FDP_ACC.1(TSO)
	The general operation of access control is explained in section Access control principles. The possible access rights for discretionary access control are explained in section DAC for MVS resources. Since access control to TCP/IP stacks, addresses and ports is managed with RACF profiles, this explanation is also applicable to these transient objects.
	The protected resources are explained in section Protected resources.
	Unix and LDAP objects and their access control attributes are explained in DAC for UNIX objects and DAC for LDAP LDBM objects, respectively

FDP_ACF.1(PSO-MVS)	Discretionary access control for z/OS MVS objects is explained in section Discretionary Access control (DAC for MVS resources and DAC for System Logger Objects in the LOGSTRM class), listing all the different types of objects and the specifics of their access control mechanisms.
FDP_ACF.1(PSO-UNIX)	Section DAC for UNIX objects explains access control for z/OS UNIX objects. The access control algorithm for persistent objects is found in subsection Algorithm to check DAC access to UNIX file system objects.
FDP_ACF.1(PSO-LDAP)	Section DAC for LDAP LDBM objects explains access control to objects in the LDAP LDBM database.

SFR	Security Functions
FDP_ACF.1(TSO)	Section DAC for UNIX objects explains access control for z/OS UNIX objects. The access control algorithm for temporary objects is found in subsection Algorithm to check DAC access to UNIX IPC objects.
FDP_ETC.1 (Labeled Security Mode only)	Export of non-labeled user data is performed by tapes or through network connections. It is not mentioned explicitly that those connections can be used for this purpose, but this should be clear. Access control to these export channels in explained in section DAC for MVS resources.
FDP_ETC.2(LS) (Labeled Security Mode only)	Export of labeled data is explained in section Mandatory Access Control (Labeled Security Mode only).

FDP_IFC.2(LS) (Labeled Security Mode only)
FDP_IFF.2(LS) (Labeled Security Mode only)
	The mandatory access control policy is explained in section Mandatory Access Control (Labeled Security Mode only).

FDP_IFC.2(NI)
FDP_IFF.1(NI)
	The packet filtering functions of the Communications Server are explained in section Communications Server in the paragraphs on Packet filtering

FDP_ITC.2 FDP_ITC.1(LS) (Labeled Security Mode only)	Import of unlabeled user data is the inverse of export and is explained in the same sections as the export.
FDP_ITC.2(LS) (Labeled Security Mode only)	Import of labeled user data is the inverse of export and is explained section Mandatory Access Control (Labeled Security Mode only).
FDP_RIP.2 FDP_RIP.3	Object reuse is described in section Object reuse.
FIA_AFL.1	The system-wide attribute REVOKE for the number of failed consecutive authentication attempts is explained in Password Quality and Password Phrase Quality. The effect of a user ID being revoked is described in subsection REVOKE of RACF Roles.

FIA_ATD.1(HU)
FIA_ATD.1(EIA)
	User attributes for human users are defined in section User and group management, subsections Definition of users and groups, User profiles, Defining Kerberos Users, LDAP LDBM Users, Group Profiles, LDAP LDBM GroupsLDAP LDBM Groups

SFR	Security Functions
	and User roles and attributes,
FIA_ATD.1(TU)	Technical users are explained in section .. Other than that, technical users have the same attributes as human users (see the entry FIA_ATD.1(HU) above).
FIA_ATD.1(LS) (Labeled Security Mode only)	Labels as users attributes are described in sectkion Mandatory Access Control (Labeled Security Mode only).
FIA_SOS.1	The password and password phrase specifics are defined in section Password Quality and Password Phrase Quality.
FIA_UAU.1	User authentication is explained in section Identification and Authentication, subsections Authentication function, RACF Passwords and Password Phrases, RACF Pass Tickets, Authentication via Client Digital Certificates, Authentication via Kerberos, Authentication by trusted servers, Authentication Method Summary, Handling of Groups During Authentication and Authentication-related differences between z/OS UNIX and typical non-z/OS UNIX systems.
FIA_UAU.5	Authentication using passwords and password phrases is explained in section RACF Passwords and Password Phrases. Authentication using digital certificates is explained in section Authentication via Client Digital Certificates. Authentication using Kerberos tickets is explained in section Authentication via Kerberos. Authentication using RACF PassTickets is explained in section RACF Pass Tickets.
FIA_UAU.7	Section RACF Passwords and Password Phrases describes that passwords are not displayed when entered during authentication.

FIA_UAU.8(EIA)
FIA_UID.3(EIA)
	In section Authentication function, the XMLAppliance discipline explains how trusted, authenticated XMLAppliance clients can request an authentication decision from the TOE based on userIDs and passwords or PassTickets presented to RACF.

FIA_UID.1	User identification is explained in Identification and Authentication.

SFR	Security Functions
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 Mandatory Access Control (Labeled Security Mode only).
FIA_USB.2	User subject binding for z/OS is explained in section Identification and Authentication, which describes protected user IDs in section Protected user IDs. Specifics of the z/OS UNIX su command are explained in section Authentication-related differences between z/OS UNIX and typical non-z/OS UNIX systems, exemptions for started tasks in section Started procedures.
FMT_MSA.1(PSO), FMT_MSA.1(TSO)	Management of object security attributes is explained in section Resource management (and subsections) where the different RACF profiles and their management is described, along with descriptions for z/OS UNIX objects and LDAP LDBM objects. Section RACF configuration and management explains the RACF configuration.
FMT_MSA.1(LS) (Labeled Security Mode only)	Management of security labels being restricted to users with the SPECIAL attribute is described in section Mandatory Access Control (Labeled Security Mode only).

FMT_MSA.3(PSO)
FMT_MSA.3(TSO)
	Default values for the access control are defined in the UACC attribute in the resource profiles as explained in section Resource management (and subsections) in the description of the resource profiles. Defaults for z/OS UNIX and LDAP LDBM objects are discussed in sections z/OS UNIX file system resources and LDAP LDBM resources.

FMT_MSA.3(NI)	In section Network configuration and management, the configuration files for the Communications Server are described.
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 Resource management (and subsections) in the description of the resource profiles and in section z/OS UNIX file system resources for z/OS UNIX objects.
FMT_MSA.4(PSO)	The inheritance of MVS object attributes from the corresponding RACF profile is described in section Discretionary Access control. The inheritance mechanism for LDAP LDBM objects is described in DAC for LDAP LDBM

SFR	Security Functions
FMT_MTD.1(AE)	Management of audited events is explained in section Audit configuration and management and in the section on the AUDITOR role (AUDITOR and group-AUDITOR).

FMT_MTD.1(AF)
FMT_MTD.1(AS)
FMT_MTD.1(AT)
	Audit trail management is explained in section Audit configuration and management. Halting the system in case of audit trail exhaustion is described in Using MVS Data Sets for SMF. The threshold to warn about audit trail exhaustion is implemented by the process of switching audit data sets and dumping them to external storage, as described in Using MVS Data Sets for SMF. The management of Log Streams is described in section Security Management for System Logger Log Streams.

FMT_MTD.1(NI)	Configuration and management of IPSec configuration data via the command line is explained in section Communication Server ipsec Command Interface.
FMT_MTD.1(NI2)	Configuration and management of additional IPSec configuration data via network interfaces is explained in section Communication Server Network Management Interface.

FMT_MTD.1(IAF)
FMT_MTD.1(IAT)
	The system-wide attribute REVOKE for the number of failed consecutive authentication attempts is explained in Password Quality and Password Phrase Quality. The effect of a user ID being revoked is described in subsection REVOKE of RACF Roles. The management authorities to resume revoked users and/or to reset their passwords is described in section Definition of users and groups.

FMT_MTD.1(IAU)
FMT_MTD.1(IAU_AUTH)
FMT_MTD.1(EIA)
	Management of user attributes is explained in section User and group management, subsections Definition of users and groups, User profiles, Defining Kerberos Users and LDAP LDBM Users.

FMT_MTD.1(CRYPTO1)	Management of cryptographic keys is explained in section Communication security, Digital Certificates, Key Rings, and Certificate Mappings in RACF and PKCS#11 Cryptographic Tokens and PKI Services.
FMT_MTD.1(CRYPTO2)	Configuration and management of digital certificates is

SFR	Security Functions
	explained in section PKI Services. Management of the mapping to RACF user is explained in section Digital Certificates, Key Rings, and Certificate Mappings in RACF and PKCS#11 Cryptographic Tokens.
FMT_MTD.1(ADD)	Configuration and management of additional TOE configuration data is explained throughout section Network configuration and management. Section Authentication by trusted servers describes specifies of the configuration of the HTTP server , FTP server, CIM server, and LDAP server. PKI Services administration is described in section Security Administration for PKI Services.
FMT_REV.1(OBJ)	Revocation of object attributes is explained as part of the management of access control to objects in sections (or subsections of) Discretionary Access Control, Mandatory Access Control (Labeled Security Mode only) and Discretionary Access control.
FMT_REV.1(USR)	Revocation of user attributes is explained as part of the management of user attributes in section User Revocation and the subsection REVOKE_ of RACF Roles.
FMT_SMF.1	See SFRs FMT_MTD.1(x)
FMT_SMR.1	The roles are explained in sections User roles and attributes.
FPT_STM.1	The time mechanism is explained in section Processor features.
FPT_TDC.1 FPT_TDC.1(LS) (Labeled Security Mode only)	The capability to provide inter-TSF data consistency for the RACF database and the extended attributes of z/OS UNIX file system objects is explained with the description of the structure of the RACF database and their profiles in section Management(and subsections) and the description of the extended attributes for z/OS UNIX file system objects in sections UNIX file system objects, Mandatory Access Control (Labeled Security Mode only) and z/OS UNIX file system resources, which allows consistent interpretation of this data in different instantiations of the TOE. This also covers security labels.
FTA_SSL.1 FTA_SSL.2	Trivially satisfied: As stated in Session Locking, the TOE does not have direct interactive human connections, since the

SFR	Security Functions
	hardware does not allow direct connections of 3270 terminals any more, so session locking is irrelevant to this TOE.
FTP_ITC.1	The different ways of establishing a trusted channel (SSL/TLS, IPSec, SSH, Kerberos are explained in section Communication security

Table 11: Mapping security functional requirements to security functions

7.2.5 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 identification and authentication (as explained) above and also 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. Access control also supports communication security by protecting access to the TCP/IP stack in general as well as individual network ports.

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.

Communication security provides support for identification and authentication because it allows to protect the transfer of authentication information. It also supports discretionary access control to communication links, because the confidentiality and integrity protection provided by the cryptographic functions prohibit spoofing attacks.

Security management is required to manage the users, groups and the privileges of users. This is supporting identification and authentication as well as access control. Different aspects of security management support each other. For example user and group management supports the management of access control, because the definition of access rights can be simplified by defining access on a group level and assign users that require access to the appropriate groups. Security management also supports auditing because it allows to define the events to be audited based on individual users, individual protected objects, privileges of the users, type of event, and (in Labeled Security Mode) security label. In addition the security management of the audit data (especially dumping the SMF data sets when they get full) also supports audit. Security management also includes the management of access rights including (in Labeled Security Mode) the definition of the security labels and the definition how they get printed on a printer that supports multiple 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. Security management also supports communication security by providing the ability to configure the different protection mechanisms SSL/TLS, IPSec, SSH, Kerberos, and AT-TLS.

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.

Object reuse supports access control to avoid that users get access to information related to system internals like authentication information(passwords) and access information in contradiction to the mandatory access control. Object reuse therefore supports TOE self-protection, identification and authentication and (in Labeled Security Mode) mandatory access control.

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

7.3 Security Assurance Requirements

The following SAR was included from the OSPP base. It does not put an additional requirement on the product but requires the evaluator to check that all ST author notes from the PP have bean dealt with in this security target.

The security assurance requirements for the TOE are the Evaluation Assurance Level 4 components, augmented by ALC_FLR.3, as specified in [CC] part 3.

The following table shows the Security assurance requirements for the TOE that are specifically instantiated for the TOE through the operations on assurance components allowed in CC part 3: iteration (Iter.), refinement (Ref.), assignment (Ass.) and selection (Sel.).

Security assurance class	Security assurance requirement	Security assurance component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
ASE Security Target evaluation	ASE_CCL.1(CCR) Conformance claims	ASE_CCL.1	CC Part 3	No	Yes	No	No

Table 12: Security assurance requirements for the TOE

7.3.1 Security Target evaluation (ASE)

7.3.1.1 Conformance claims (ASE_CCL.1(CCR))

Content and presentation elements:

ASE_CCL.1.10C The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs *including the statements marked as “ST-Author Note” and the specification given in section 8.1 of the OSPP base* for which conformance is being claimed.

Application note: ASE_CCL.1 specified in CC Part 3 is refined as follows: All Developer Action Elements, Content and Presentation Elements, Evaluator Action Elements remain unaltered, except for ASE_CCL.1.10C as refined above.

7.4 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

8.1 TOE Security Functionality

This chapter provides a summary of the security functions of z/OS that are subject to the evaluation. z/OS has more security functions than described in this chapter; only those that implement the security requirements claimed in chapter 7are described here.

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 curly braces using an identifier for the security function and a number.

8.1.1 Overview of the TOE architecture

z/OS is an operating system that runs on the IBM z/Architecture processors. Those processors provide a separate problem and supervisor state and memory protection functions that allow z/OS to prohibit direct access from untrusted applications to I/O devices, protected memory areas used by the TOE, and memory areas used by other applications. The underlying firmware also allows the definition of separate logical partitions where several instances of the TOE can execute in parallel on the same hardware. The TOE may also be loaded in one logical partition while other non-TOE software is loaded in other logical partitions. The logical partitioning function is part of the TOE environment and has been evaluated separately.

The TOE provides an interface to applications by allowing them to request TOE services.

The TOE provides the following security functions:

1. Identification and authentication
2. Discretionary Access Control based on access control lists associated with objects
3. In Labeled Security Mode: Mandatory Access Control based on security attributes of subjects and objects
4. Management functions to administer auditing, discretionary access control, and (in Labeled Security Mode) mandatory access control, as well as users and groups with their related attributes
5. Secure communication
6. Auditing
7. Object reuse
8. TOE self-protection functions based on security features provided by the underlying hardware including memory protection and the provision of a privileged state that allows the TOE to reserve and protect a domain for its own execution.

The TOE itself is logically structured into the following major units:

1. The Hardware Configuration Definition (HCD), which mirrors the IOCDS definition of the underlying abstract machine.
2. The Base Control Program (BCP), which is responsible for handling supervisor call interrupts, program call interrupts, and all other interrupts, and task scheduling and memory management, including the management of address spaces
3. The Data Facility Storage Management Subsystem (DFSMS), which is responsible for accessing and managing disk and tape devices, including the data sets on those devices
4. The Communications Server, which is responsible for network communication using SNA- or IP-based protocols , and which provides TN3270, FTP, telnet, rsh, IKE, Network Security Services (NSS), Centralized Policy, and DCAS servers.
5. The Job Entry Subsystem (JES2), which is responsible for scheduling jobs and handling spool files (for the purpose of the evaluation, the SDSF display facility is considered to be part of JES2)

6. The UNIX System Services, which provides UNIX programming, user interfaces, and rlogin support. For the purposes of this ST, zFS and HFS are considered to be part of UNIX System Services.
7. The Resource Access Control Facility (RACF), which is the central system for discretionary and mandatory access control to resources
8. The Time Sharing Option Extensions (TSO/E) system, which is responsible for handling of commands issued by users at TSO/E terminals
9. The Print Services Facility (PSF) provides services for printing of output, and prints proper security labels on pages.
10. IBM Ported Tools for z/OS which provides OpenSSH functions (e.g, sshd, scp, sftp). The evaluated version includes the following:
    a. OpenSSH 5.0
    b. OpenSSL 0.9.8k (statically linked; not available to applications)
    c. zlib 1.2.3 (statically linked; not available to applications)
11. The z/OS Network Authentication Service and associated GSSAPI programming services that provides authentication and message privacy and integrity functions.
12. The IBM HTTP Server z/OS Cryptographic Services Integrated Cryptographic Services Facility, which provides management of secure crypto keys used with the PCIXCC and CryptoExpress2 hardware cards, and management of the cryptographic hardware. It also implements storage, retrieval, and maintenance of information contained in PKCS#11 cryptographic tokens.
13. z/OS Cryptographic Services PKI Services, which provides digital certificate management (CA and RA) functions.
14. z/OS Network File System (NFS), which provides access to MVS data sets and UNIX files to clients over the TCP/IP network.
15. IBM Tivoli Directory Server (also called z/OS LDAP Server in this ST), which provides LDAP support and also an interface allowing remote administration of RACF users and groups in non-MLS environments.
16. IBM z/OS Common Information Model (CIM) Server, which provides CIM data and services to help manage z/OS in a distributed network, and is based on OpenPegasus CIM Server.
17. The System REXX (AXR) server address space which can run REXX execs upon request from other parts of the TOE. Such execs run using the identity of the requester, and run in an authorized state (unlike REXX execs run in batch, TSO, or z/OS UNIX shell environments).

The TOE itself consists of a “nucleus” operating in the supervisor state of the underlying abstract machine and a set of “trusted processes” that either also operate in supervisor state or operate as “authorized programs”. Those authorized programs start their operation in problem state, but can switch into supervisor state, operate with storage key 0, or both, so are therefore not limited in their capabilities by any element of the system security policy. Therefore, all authorized programs allowed to be executed in the evaluated configuration are considered to be part of the TOE. Additionally, any program running with

UID(0) or with access to the FACILITY class resources BPX.SUPERUSER, BPX.DAEMON, or BPX.SERVER, or with access to any UNIXPRIV class resources named SUPERUSER.*function-name*, or with access to any PTKTDATA class profiles named IRRPTAUTH.*function-name*, are considered “authorized” for this evaluation, and thus are also considered to be part of the TOE.

More information on how the TOE identifies, manages, and protects authorized programs can be found in Authorized programs.

8.1.1.1 Main trusted subsystems of the evaluated configuration

Some programs are started with authorization (see also Authorized programs) during system startup. Those include the Job Entry Subsystem (JES2), PSF, the Time Sharing Option Extensions (TSO/E) subsystem, the Communication Subsystem (CS), and the z/OS UNIX System Services.

Job Entry Subsystem (JES2)

The Job Entry Subsystem is responsible for starting jobs that have been entered at remote or local entry stations, submitted by TSO or UNIX users or submitted by batch jobs themselves. A job consists of a set of individual job steps described in the Job Control Language (JCL). There, the name of the job, the user ID the job will have during execution (usually inherited from the submitting user), the data sets used by each job step, and the first program to be started for each job step are defined.

JES2 is responsible for scheduling those jobs, that is, for transforming the JCL statements into internal control blocks and initiating each job step in cooperation with the “initiator”. As described above, a job step may execute with the authorization bit set in the Job Step Control Block (JSCB) if the conditions mentioned above are satisfied.

JES2 uses RACF to authenticate users. If they are not already authenticated by another subsystem, users need to specify their passwords in the job card, which is the first JCL statement in a job. JES2 also uses RACF to control access to data sets and printers.

JES2 is responsible for managing spool files for job input and job output. JES2 also manages printers attached to it. In Labeled Security Mode and in the case of a multilevel printer device, JES2 in cooperation with the printer system ensures that each page of printer output is marked with the security label of the job step that produced the output.

Time Sharing Option Extensions (TSO/E)

TSO/E is the main dialog system within a primary user interface to the z/OS system. This interface provides many capabilities such as allowing users to execute commands and programs as well as write programs in a high-level procedural language known as REXX. VTAM creates a separate address space for each TSO/E user. TSO/E requires user authentication before allowing users to issue TSO commands, execute programs, or submit jobs to JES2. RACF provides the user authentication and resource access controls for TSO/E users, as for all other users on the system.

TSO commands and services can also run outside of a TSO/E login environment, such as in batch jobs or in UNIX processes. In such cases, TSO/E has not performed the authentication functions, which took place either in JES or in the UNIX processing that established the process. However, all access checking for resources used by the TSO commands and services happens just as for commands and services invoked in the standard TSO/E login environment.

Communications Server

z/OS provides networking functions with the Communications Server. This subsystem provides support for network communication using the IBM SNA protocols and the TCP/IP protocol suite. APIs for both protocol stacks are provided. For IP, both IPv4 and IPv6 are supported. For the evaluated configuration, use of SNA networking by user programs has been excluded. Only those parts of SNA that are required for TN3270 are part of the TOE. Those parts do not export a direct interface for the use by untrusted programs.

z/OS UNIX System Services

z/OS also provides users and programs with a UNIX environment. RACF-defined users who also have a UNIX UID and whose default group (at least) has a GID can use this environment to operate in a UNIX shell environment and use UNIX commands and program library interfaces.

Additionally, users defined as UNIX users can also use UNIX-based programs, and access UNIX data, while running in other environments, such as TSO/E or batch, and users running in a UNIX environment can access non-UNIX data (e.g., MVS data sets).

RACF is used by the UNIX system services to:

- authenticate users
- control access to UNIX files and directories
- control access to UNIX IPC objects

UNIX files have the traditional access permission bits and POSIX-compatible access control lists. To manage an ACL for a file, one must either be the file owner or have superuser authority (UID=0 or have READ access to SUPERUSER.FILESYS.CHANGEPERMS in the UNIXPRIV class). In Labeled Security Mode, UNIX files and directories are also subject to the mandatory access control function of the TOE. File permission bits and access control lists are stored with the files as part of the UNIX file system. In all attempts to access a UNIX file, the UNIX system services will call RACF and provide the permission bits, access control list and (in Labeled Security Mode) security label as an additional input to the call.

UNIX IPC objects are controlled by the access permission bits for IPC objects and (in Labeled Security Mode) the mandatory access control rules defined by RACF.

In Labeled Security Mode: For full support of mandatory access control, the evaluated configuration only supports zFS as a UNIX file system. A read-only hierarchical file system (HFS) can also be used if the contained data is at the same security level.

Print Services Facility

z/OS provides printing functions with JES2 and PSF. These subsystems provide support for printing output on a large variety of print peripherals. In Labeled Security Mode, PSF must be used in conjunction with JES2 to enforce printing of security labels on all pages of print jobs containing labeled data.

8.1.2 Identification and Authentication

8.1.2.1 Authentication function

A user can interact with the TOE in one of the following ways:

- As a TSO user
- As an operator at a console
- By submitting a job to be initiated and scheduled by the Job Entry Subsystem (JES2)
- As a UNIX user, including access via the UNIX shell or as a client of a UNIX-based server such as FTP, HTTP, SSH, TN3270, rsh, rexec, etc.
- As an LDAP user
- By using a NFS client supporting the z/OS extensions
- As a CIM user
- As a user of the RMF Distributed Data Server
- As a Communication Server Policy Agent or Network Security Server or Load Balancing Advisor client
- As a Kerberos principal

In all cases (except for the HTTP server or LDAP server that the administrator may optionally configure to allow selected access by unauthenticated users as described elsewhere) users are identified and authenticated by the TOE {IA.1::IA.1.1} before being authorized to perform any other security relevant action. In the case of jobs submitted by an already-authenticated user, no additional authentication is required for jobs running with the ID of the user who submitted them. The internal reader accepts (and relies) in this case on the authentication performed when the user has logged on to the system {IA.1::IA.1.2}.

An exception to this rule are started tasks, which operate under a protected user ID and are started either at system startup or through an operator command. Those tasks are not executing on behalf of a human user and their protected user IDs are exempt from authentication {IA.1::IA.1.3}. They must only be started from trusted data sets.

When authenticating a user the TOE allows applications to accept:

- A user ID defined to RACF {IA.1::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}.

- For applications supporting TLSv1.1-, TLSv1- or SSLv3-based client authentication, a valid X.509v3 digital certificate (see Authentication via Client Digital Certificates) that the application (or AT-TLS) has mapped to a RACF user ID via __certificate() or R_usermap() {IA.1::IA.1.4-R8-MULTI-1}

- For applications supporting Kerberos (see Authentication via Kerberos), a valid Kerberos service ticket for the client Kerberos principal that the application has mapped to a local user ID via R_usermap() {IA.1::IA.1.4-R8-MULTI-2}. The application may also request entry of a valid RACF user ID and password/phrase {IA.1::IA.1.4-R10-MULTI-3} and if so the application must run the user's session under that ID {IA.1::IA.1.4-R8-MULTI-5}.

- For SSH login functions (ssh, scp, sftp) RACF will also verify the specified password/phrase {IA.1::IA-1.4-R10-SSH-1}. For clients authenticating using public/private keys, SSH will verify the private key using information from the RACF keyring when configured to allow this authentication method {IA.1::IA-1.4-R12-SSH-2}

- The NFS server must be configured with SECURITY(SAF or SAFEXP) and in this mode will support authentication of the client via either Kerberos or the mvslogin command. If the client uses mvslogin and does not use Kerberos, then the NFS server will validate the user ID and password/phrase using RACF {IA.1::IA.1.4-R12-NFS-1} . If the client uses Kerberos then the NFS server will use SAF and RACF functions to map the Kerberos principal to a local RACF user ID and will use that ID for all NFS security functions {IA.1::IA.1.4-R11-NFS-2}.

- For LDAP authentication:

  - With an SDBM DN, the z/OS LDAP server accepts the DN and a RACF password/phrase, and presents the user ID from the DN, together with the password/phrase, to RACF for authentication. {IA.1::IA.1.4-R10-LDAP-1}.

  - With an LDBM DN, the z/OS LDAP server accepts the DN and a RACF password/phrase. It transforms the LDAP-style DN into a RACF user ID by lookup within the LDBM database, and presents the resulting RACF user ID and the supplied password/phrase to RACF for verification {IA.1::IA.1.4-R10-LDAP-2}.

  - With the ICTX plug-in (for remote authorization or remote auditing extended-operation requests) the z/OS LDAP server accepts an ICTX-format DN of the form racfid=userid,cn=ictx and the RACF user's password/phrase, and the plug-in presents the user ID from the DN, together with the password, to RACF for authentication. {IA.1::IA.1.4-R10-EIM-1}.

  - Additionally, LDAP will allow authentication via a digital certificate presented over an SSL or TLS connection when doing an external SASL bind, and will map the certificate to a RACF user ID using TOE functions, failing the bind if RACF does not recognize the certificate. This will work for access to SDBM or

LDBM data {IA.1::IA.1.4-R10-LDAP-3}. For SDBM, LDAP will provide that RACF user ID when accessing the SDBM back-end {IA.1::IA.1.4-R10-LDAP-4}. For LDBM, LDAP will transform the RACF user ID into an SDBM-style DN which (based on administrator-supplied LDAP configuration options) can either replace or supplement the DN contained in the certificate {IA.1::IA.1.4-R10-LDAP-5}.

- For authentication to the CIM server, CIM accepts a RACF user ID and password or PassTicket and uses RACF to validate them before allowing connection {IA.1::IA.1.4-R8-CIM-1}. Subsequently, if RACF validates the ID and password, the CIM server continues authentication by ensuring that the user has access to the CIMSERV resource in the (customer-defined) WBEM RACF class according to the type of request {IA.1::IA.1.4-R8-CIM-2}. In addition the CIM server uses pthread_security to process requests that access/manipulate system resources under the requestor's user ID {IA.1::IA.1.4-R8-CIM-3}. For use of PassTickets the administrator can configure CIM to use either the standard z/OS UNIX application ID of OMVSAPPL, or to use a CIM-specific application ID of CFZAPPL {IA.1::IA.1.4-R10-CIM-4}.
- The RMF Distributed Data Server (DDS) authenticates users via a RACF user ID and password or PassTicket {IA.1::IA.1.4-R11-RMF-1}.
- The Communication Server Policy Agent Server {IA.1::IA.1.4-R9-CS-POLCEN-1} and Network Security Server {IA.1::IA.1.4-R9-CS-NSS-1} accept a RACF user ID and password or PassTicket from their clients during session initiation and use RACF to validate them before allowing connection.
- The Communication Server Network Security Services (NSS) XMLAppliance discipline SAFAccess service accepts a RACF user ID and password or PassTicket and validates them at the request of the XML Appliance client {IA.1::IA.1.4-R10-CS-XMLApp-1}. The SAFAccess service can also perform an access control check for a specified resource when requested to do so {AC.4::AC-R10-CS-XMLApp-4}.
- The Communication Server NSS also provides XMLAppliance services to:
    - Provide certificate management operations {SM.4::SM-R11-CS-XMLApp-5}
    - Provide private key operations to allow retrieval of private keys from RACF certificates {SM.4::SM-R11-CS-XMLApp-6}, and to perform RSA signature creation and RSA decryption using ICSF-protected keys {SM.4::SM-R11-CS-XMLApp-7}.
- The Communication Server Load Balancing Advisor accepts a client digital certificate via SSL/ TLS from its clients (Load Balancing Agents, external load balancers) and uses RACF to validate them before allowing connection {IA.1::IA.1.4-R10-CS-LBA-1}. Following a successful authentication, the Load Balancing Advisor further restricts access by requiring that the user have READ access to SERVAUTH resource EZB.LBA.[one of: AGENTACCESS, LBACCESS]. <system-name>.<tcp-sysplex-group-name> {IA.1::IA.1.4-R10-CS-LBA-2}.

Some additional considerations:

- If security label (SECLABEL) processing is active (mandatory in Labeled Security Mode), 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. This does not apply to operator console login, telnet, rlogin, rsh, rexec, ssh, scp, sftp, LDAP, z/OS Network Authentication Service, HTTP, ftp, or to batch jobs {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 (e.g, HTTP) 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 job runs with the user ID that the job card specifies, not the surrogate user's user ID. The audit record for surrogate job submission identifies both the surrogate user and the jobcard user ID {IA.1::IA.1.13}.

8.1.2.2 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}.

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 passwordchange 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.

When the commands are called in a way that allows the TOE to suppress printing, passwords are not displayed:

- when entered at a TSO terminal as part of the login process {IA.2::IA.2.10}, or
- when entered at a TSO terminal as part of the ADDUSER, ALTUSER, or PASSWORD commands when the command contains the PASSWORD keyword but no value {IA.2::IA.2-R10-RACF-21}, or
- when entered into one of the RACF-supplied ISPF panels that allows specification of a password {IA.2::IA.2-R10-RACF-22}, or
- when entered at a system operator console as part of the operator logon {IA.2::IA.2-R8-BCP-1}, or
- when the content of a jobcard is displayed as part of a job’s output {IA.2::IA.2.13}.

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.

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}

When the commands are called in a way that allows the TOE to suppress printing, the phrase is not displayed:

- when entered at a TSO terminal as part of the login process {IA.2::IA.2-R10-TSO-23}, or

- when entered into one of the RACF-supplied ISPF panels that allows specification of a password phrase {IA.2::IA.2-R10-RACF-25}.

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

8.1.2.3 RACF Pass Tickets

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}

z/OS UNIX uses, by default, an application name (APPLID) of OMVSAPPL {IA.2::IA.2.14-R10-USS-1} when authenticating users via:

- The __login(), or pthread_security_np() services.

- The _passwd() service if issued from a thread created by pthread_create() which subsequently issued pthread_security_np(), and if the _passwd() call does not specify a new password.

The application may override this default in one of these ways:

- For pthread_security_np() and __passwd(), the application can

  - update the BPXYTHLI control block to indicate that z/OS UNIX should instead use the job name as the APPLID value {IA.2::IA.2.14-R10-USS-2}, or

  - update the BPXYTHLI control block to indicate a specific APPLID to use {IA.2::IA.2.14-R10-USS-3}.

- By changing to use one of the corresponding new services pthread_security_applid_np(), __login_applid(), and __passwd_applid() the application can specify an APPLID value directly as a parameter on the call {IA.2::IA.2.14-R10-USS-4}.

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}.

z/OS also allows Java applications running on z/OS to generate or evaluate PassTickets, using a JNI interface to R_ticketserv() and R_gensec() {IA.2::IA.2.14-R8-RACF-14}.

The Communications Server uses PassTickets as part of its participation in the Express Logon Facility (ELF) and Web Express Logon (WEL) single signon solutions.

- Express Logon Facility (ELF) --This function is provided in a Two-Tier or Three-Tier model for single-signon to a z/OS application. With either model, a user presents an X.509v3 digital certificate to the z/OS ELF service, which in the two-tier model is a TN3270 server and in the three-tier model is a Digital Certificate Access Server (DCAS). When the TN3270 server or DCAS receives the certificate and a target application name, it will invoke RACF to : 1) map the certificate to a RACF user ID 2) Generate a PassTicket for the user ID and target application. {IA.2::IA.2.14-R9-ELF-1} In the two-tier model DCAS is not involved and the TN3270 server runs on z/OS. Here, the ELF function is agreed upon by the TN3270 sever and client. When the TN3270 server receives the logon panel (by examining the input data), it will invoke the RACF services to map the certificate to a user ID and generate a PassTicket, which it will then insert the user ID and PassTicket into the logon panel, subsequently passing the panel to the target application for logon. (IA.2.61) {IA.2.14-R9-ELF-2} In the case of the three-tier model, the TN3270 server does not run on z/OS, but runs on a distributed platform. In this case, the distributed TN3270 server (upon receipt of the logon panel), invokes DCAS, passing it the certificate and target application name (on behalf of the end user). DCAS then invokes RACF to map the certificate to a User ID and generate a PassTicket. DCAS passes this information back to the TN3270 server which inserts the User ID and PassTicket into the logon panel, and subsequently passes the panel to the target application for logon. {IA.2::IA.2.14-R9-ELF-3}

- Web Express Logon (WEL) --In this model (non certificate-based), a DCAS client is requesting a PassTicket on behalf of an end user. Note that as part of the single-signon architecture, that end user has already been authenticated at some point prior to the DCAS client requesting the PassTicket. In this case, the DCAS server supports two types of requests:
  1. It can receive a valid z/OS user ID from the client and the target application name. It will pass these to RACF requesting a PassTicket for that user ID and application. {IA.2::IA.2.14-R9-WEL-1}
  2. It can receive a principal name from the client along with the target application name. It will pass these to RACF requesting a z/OS user ID that has been mapped to the principal name and a PassTicket which will be returned to the requesting client. In this case, it is required that the z/OS user ID be mapped to a principal name using the RACF KERBLINK class. {IA.2::IA.2.14-R9-WEL-2}

- Additional details:

  - For ELF, in the two-tier model, communication between the TN3270 client and server requires SSL with client authentication at a minimum. {IA.2::IA.2.14-R9-ELF-4}

  - For ELF, in the three-tier model, communication between the distributed TN3270 server and DCAS requires SSL with client authentication at a minimum. The SSL and DCAS client in this case is the TN3270 server itself. In the evaluated configuration the DCAS server will also verify that that its client (the TN3270 server) is authorized to SERVAUTH resource EZA.DCAS.system-name. {IA.2::IA.2.14-R9-ELF-5}

  - For WEL, communication between the DCAS server and client (WEL server) requires SSL with client (WEL server) authentication at a minimum. In the evaluated configuration the DCAS server will also verify that its client (the WEL server) is authorized to SERVAUTH resource EZA.DCAS.system-name. {IA.2::IA.2.14-R9-WEL-3}

Additionally, the Communications Server provides the DCAS server (Digital Certificate Application Server) which can be used by applications running in the network, perhaps as part of a single-signon service. DCAS provides two functions:

1. Generate a PassTicket for an application-specified user ID and application name; {IA.2::IA.2.14-R8-DCAS-1}
2. Map an application-specified digital certificate for the server’s client to a RACF user ID, and generate a PassTicket for that user and an application-specified application name. {IA.2::IA.2.14-R8-DCAS-2}In order to use DCAS, the network-based application must connect to DCAS using an SSL session with client authentication, and provide its own digital certificate that maps to a RACF user ID {IA.2::IA.2.14-R8-DCAS-3}. In the evaluated configuration that mapped user ID must be authorized to resource EZA.DCAS.system-name in the SERVAUTH class {IA.2::IA.2.14-R8-DCAS-4}.

The Communication Server's Network Security Services Server provides a SAFAccess service as part of its XMLAppliance discipline. The SAFAccess performs RACF userid authentication on behalf of the XMLAppliance client and supports passwords and PassTickets as authentication tokens. NSS clients must connect to the NSS server using a TLS-protected session and must also authenticate themselves to the server using their own RACF userid and password or PassTicket {IA.2::IA.2.14-R10-CS-XMLApp-2}.

PassTickets are also used internally by the Kerberos KDC server as part of the processing when users change their Kerberos passwords.

8.1.2.4 Authentication via Client Digital Certificates

In the evaluated configuration, SSL- or TLS-aware applications, or the Application-Transparent TLS (AT-TLS) functions of the Communications Server, 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 server or AT-TLS to acquire the client certificate via the standard SSLv3 or TLS data flows. As part of that processing, 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.

System SSL can validate certificates using either the processing specified by [RFC2459], or by [RFC3280], under control of an environment variable or specifications provided by the application. In the absence of an environment variable or application-provided specification, System SSL will first validate using [RFC2459] and if that fails will then retry using [RFC3280] {IA.2::IA.2.15-R11-SSL-20}.

gsk_validate_certificate_mode will perform the following checks against the client certificate and certification chain:

1. The certificate's subject name must be identified by either a non-empty distinguished name (with an optional SubjectAltname certificate extension) or an empty distinguished name with a SubjectAltName certificate extension: RFC2459: {IA.2::IA.2.15-R8-SSL-1}.

   RFC3280: {IA.2::IA.2.15-R11-SSL-15} SubjectAltName Extension must be critical.
2. Certificate Authority certificates must have a non-empty subject name {IA.2::IA.2.15-R8-SSL-2}.
3. The certificate issuer name must not be an empty distinguished name {IA.2::IA.2.15-R8-SSL-3}.
4. The CertificatePolicies extension, if present, must not be marked critical (RFC2459) {IA.2::IA.2.15-R8-SSL-4}. For RFC3280, if the certificate policies extension is present, it must be marked critical and must satisfy the policies defined by the issuing certificate chain. {IA.2::IA.2.15-R11-SSL-16}
5. The current time must not be earlier than the start of the certificate validity period {IA.2::IA.2.15-R8-SSL-5}.
6. The issuer's certificate must be a valid CA certificate, and the root certificate and any intermediate signing certificates not in the client’s message must be present in the server’s key ring {IA.2::IA.2.15-R8-SSL-7}. The server's certificate store can either be a RACF key ring (DIGTCERT class) or a PKCS#11 token in ICSF TKDS (CRYPTOZ class) {IA.2::IA.2.15-R9-SSL-14}.
7. The certificate signature must be correct and using a supported signature (RSA 1024-4096 bit key with hashing algorithm --MD5, SHA-1, SHA-224, SHA-256, SHA-384 or SHA-512 or DSA 1024-bit key with hashing algorithm SHA-1) {IA.2::IA.2.15-R10-SSL-8}
   or ECDSA 160-521 bit keys with hashing algorithm SHA-1, SHA-224, SHA-256, SHA-384, SHA-512) {IA.2::IA.2.15-R12-SSL-21}
8. No certificate in the certification chain can be revoked or expired {IA.2::IA.2.15-R8-SSL-10}.
9. For CA certificates, the BasicConstraints extension, if present, must have the CA indicator set and the path length constraint must not be violated by subordinate certificates in the certification chain RFC2459 – {IA.2::IA.2.15-R8-SSL-11}

   RFC3280 -- {IA.2::IA.2.15-R11-SSL-17} --Basic Constraints extension must be present.
10. If the issuing certificate chain has defined any name constraints through Name Constraints extensions, the constraints must not be violated by the subject certificate {IA.2::IA.2.15-R8-SSL-12}.

11. The key usage extension, if present in a CA certificate, must specify signing capability RFC2459 – {IA.2::IA.2.15-R8-SSL-13}

    RFC3280 -- {IA.2::IA.2.15-R11-SSL-18} --Key Usage extension must be present.

12. Certificate Authority certificates with a Key Usage extension present which has the keyCertSign bit set must have a Basic Constraints extension present which has the CA indicator set. RFC3280 -- {IA.2::IA.2.15-R11-SSL-19}

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 pthread_security_np() service {IA.2::IA.2.16-R8-USS-1}, the InitACEE() service {IA.2::IA.2.16-R8-RACF-3}, or the _certificate() service {IA.2::IA.2-16-R9-USS-1} which will accept the certificate as input. In either 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, 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, 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}.

8.1.2.5 Authentication via Public/Private Key (SSH)

OpenSSH supports authentication via public/private keys, however for the evaluated configuration OpenSSH on z/OS must be configured to obtain those public/private key pairs from digital certificates associated with RACF key rings. The existing RACDCERT command can be used to generate the keys and a certificate, or the certificates may be generated elsewhere and imported into RACF using RACDCERT. Public/private keys stored directly in the UNIX file system must not be used.

When a remote user authenticates to the OpenSSH server, the server will use the public key, obtained via a digital certificate which is associated with the user's configured key ring, to perform the authentication. {IA.2::IA.2-R12-SSH-KEY-1}

When a z/OS user acts as an SSH client, connecting to an SSH server, the client will obtain the necessary private key via a digital certificate which is associated with the user's configured key ring to perform the authentication. {IA.2::IA.2-R12-SSH-KEY-2}

The private key is not needed at the server when a client authenticates. The public keys must be distributed to remote hosts. When stored in a key ring, the certificate must be exported (via RACDCERT) and manually copied (e.g. via FTP) to the remote host, where it will be imported into that system's key ring.

When configured to use key rings, the OpenSSH server and client code will use existing System SSL interfaces to pull the keys from the RACF key ring, and the server and client will need authority to those key rings to use the RDATALIB service. {IA.2::IA.2-R12-SSH-KEY-3}

8.1.2.6 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 authentication via Kerberos:

1. The client (principal) will obtain a Ticket Granting Ticket (TGT) by authenticating to the assigned Kerberos registry, which may be a z/OS Network Authentication Service instance KDC {IA.2::IA.1.4-R8-KERB-1} or some non-z/OS KDC. This initial authentication will follow standard Kerberos protocols, using one of the encryption protocols specified for the KDC {IA.2::IA.1.4-R8-KERB-2}. If the z/OS Network Authentication Service KDC is used for initial principal authentication, the z/OS Network Authentication Service will map the Kerberos principal name to a RACF user ID and the password used to derive the key info for the Kerberos authentication exchanges will be the user’s RACF password or phrase, whichever was last established for the user {IA.2::IA.1.4-R10-KERB-3}.
2. As is standard with the Kerberos protocol, the client will then acquire a service ticket for the desired server, and will present that ticket to the server for validation and mapping to a RACF identity.
3. {IA.2::IA.1.4-R12-KERB-8} If the service principal in a request is different from the principal specified on the krb5_rd_req, krb5_rd_req_verify, or gss_accept_sec_context

API call, but the request is otherwise valid, the request will still be approved if and only if all the following are true:

- use_dvipa_override=1 is specified in the libdefaults section of the krb5.confconfiguration file,
- both principals are in the standard “Primary/Instance@Realm”format (with only one instance),
- both the primary and the realm in the request are the same as those provided in the API call, and
- the KRB5_SERVER_KEYTAB environmental variable is set to 2 and the application server that invoked the API call has at least READ access in the KERBLINK class to the principal specified in the request
  -OR-
  an entry in the keytab file matches the service principal name, version number, and encryption type from the incoming service ticket.

4. If the user is assigned to a foreign Kerberos realm (with respect to the TOE server application), the user will first use kinit to acquire a TGT from his local KDC. If a peer trust relationship is defined between the two KDCs, the client application can use this initial TGT to obtain a TGT for the remote z/OS KDC from its local KDC, which is then used by the client application to obtain a service ticket from the remote z/OS KDC. The z/OS KDC will only issue a service ticket for a TGT produced by a KDC in another realm if the administrator for each realm has configured a trust relationship between the two KDCs {IA.2::IA.1.4-R8-KERB-4}. This trust relationship may be transitive and involve the client contacting a series of KDCs before finally obtaining the TGT for the remote z/OS KDC {IA.2::IA.1.4-R8-KERB-5}.
4. If the application server is running on z/OS, once it has validated the client principal’s service ticket, it uses the R_usermap() service to determine the local RACF user ID associated with the Kerberos principal that may be defined to the z/OS Network Authentication Service {IA.2::IA.1.4-R8-KERB-6} or foreign {IA.2::IA.1.4-R8-KERB-7} principal that is defined to another Kerberos realm with an established trust relationship with the z/OS Network Authentication Service.

8.1.2.7 Started procedures

With the concept of a started procedure, the TOE 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 {IA.3::IA.3.1}.

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 {IA.3::IA.3.4}. 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)

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.

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 used to log on to the system, and are protected from being revoked through incorrect password attempts {IA.3::IA.3.6}.

8.1.2.8 Authentication by trusted servers

Trusted servers of z/OS may be required to perform user authentication. They all use RACF to verify the credentials presented by a user for authentication. Those trusted servers may have some special configuration options that are explained in this section.

Handling of user authentication in the HTTP server

Users may connect to the HTTP server of the TOE. The server will assign an installation-defined pseudo-user ID to a user unless the user is authenticated with his user ID and password {IA.3::IA.3.V1R7.1}. Access checks to protected resources the HTTP server accesses on behalf of an unauthenticated user will be performed using the access rights of this installation-defined pseudo user ID {IA.3::IA.3.V1R7.2}.

The HTTP server also provides a function to identify and authenticate users using their user ID and password when the PROTECT directive specifies UserID %%CLIENT%% {IA.3::IA.3.V1R7.3}. Once authenticated successfully, the access rights of the authenticated user are checked when the HTTP server attempts to access resources protected by that PROTECT directive {IA.3::IA.3.V1R7.4}. The HTTP server uses RACF for user identification and authentication {IA.3::IA.3.V1R7.5}. Once the user has been successfully authenticated the HTTP server, when acting on behalf of the user, switches to the MVS user ID of the authenticated user and all access checks to protected resources are performed by RACF checking the access rights of this user {IA.3::IA.3.V1R7.6}.

The HTTP Server also supports client authentication via SSL/TLS client authentication using digital certificates. {IA.3::IA.3-R8-HTTP-1}. To enable this support, the administrator would specify UserID %%CERTIF%% on the PROTECT directive {IA.3::IA.3-R8-HTTP-2}. The HTTP server will present the certificate to RACF to map into a RACF user ID {IA.3::IA.3-R8-HTTP-3} and then proceed with access checking using that RACF identity as above for UserID %%CLIENT%% {IA.3::IA.3-R8-HTTP-4}.

Handling of user authentication in the FTP server

Users may connect to the FTP server and authenticate with a user ID and password or a Kerberos service ticket, or a digital certificate as previously described. The FTP server also supports unauthenticated, or anonymous, access to data. Administrators who have certain data that they want to serve to unauthenticated users via FTP may enable this anonymous access. Data access will then occur under a RACF ID that the administrator has specified in the FTP server configuration file, and only data accessible to that user will be served to the FTP client. Additionally, as this is intended to be “public” data with unrestricted access, no audit logs showing the actual human user who accessed the data can be maintained, but the administrator will have accepted the loss of auditing by configuring anonymous access.

In the evaluated configuration, if the administrator wishes to allow anonymous FTP access, the following parameters must be specified:

1. ANONYMOUSLEVEL 3
2. ANONYMOUS user-id/SURROGAT (Note: the administrator can choose any user-id he wants, but the user must have the RESTRICTED attribute, and an OMVS segment with a unique UID, a default group with a unique GID, a home directory to which the user has access, and should have no other group connections.).
3. ANONYMOUSFILEACCESS HFS or MVS or BOTH
4. ANONYMOUSFILETYPEJES FALSE
5. ANONYMOUSFILETYPESQL FALSE

With these settings:

- When the user specifies USER ANONYMOUS the FTP server will prompt for an email address {IA.3::IA.3-R9-FTP-1}.
- The FTP server will then establish a security environment for the chosen user ID from the ANONYMOUS statement {IA.3::IA.3-R9-FTP-2}.
- The FTP server’s ID must have SURROGAT authority to BPX.SRV.user-ID {IA.3::IA.3-R9-FTP-3}.

- The user ID must have an OMVS UID and its default group must have a GID {IA.3::IA.3-R9-FTP-4}.

- If starting in the UNIX file system or if the user issues a “cd” to switch to the UNIX file system, the FTP server will issue chroot() to restrict the user to his home directory as specified in the user’s OMVS segment {IA.3::IA.3-R9-FTP-5}.

- The user will only be able to access data in that home directory {IA.3::IA.3-R9-FTP-6}, or if the user switches to the MVS file system (assuming the administrator specified ANONYMOUSFILEACCESS MVS or BOTH) the user will have access to only that data to which the user ID or his group(s) are explicitly permitted {IA.3::IA.3-R9-FTP-7}. No access to other MVS data via UACC or ID(*) or GLOBAL will be permitted {IA.3::IA.3-R9-FTP-8}.

The user will not be able to specify SITE FILETYPE JES nor SITE FILETYPE SQL (IA.3.24) (IA.3-R9-FTP-9)

If desired, the administrator can configure the FTP server to verify an authenticated user's authority to access the server via a SERVAUTH resource check before allowing access to data. To do this, the administrator would specify VERIFYUSER TRUE in the FTP configuration parameters and define a RACF SERVAUTH profile to protect the resource EZB.FTP.<system-name>.<ftp-daemon-name>.PORT<nnnn> where nnnn represents the port number assigned to that FTP daemon. The user will need READ access to that resource if it is protected by a SERVAUTH profile {IA.3::IA.3-R10-FTP-9}.

The administrator may also prevent the user from accessing data in the UNIX file system (thus restricting the user to accessing traditional MVS data sets). To do this, the administrator would define a profile in the SERVAUTH class to protect the resource EZB.FTP.<system-name>.<ftp-daemon-name>.ACCESS.HFS and deny the user access to the resource via the profile UACC or access list {IA.3::IA.3-R10-FTP-10}.

Handling of user authentication in the CIM server

Users may connect to the CIM server and authenticate with a RACF user ID and password or with a RACF user ID and PassTicket {IA.3::IA.3-R10-CIM-1}. The CIM server first uses RACF services to validate the user ID and password or PassTicket. In addition for all user requests that are to obtain or manipulate system management data, the CIM server dispatches the request to an extra thread, for which the effective userid is switched to that of the requestor using the pthread_security_np() service . This way the access to system resources occurs on behalf of the user's identity rather than under the identity of the CIM server {AC.1::AC.1-R10-CIM-2}.

Depending on the type of request the CIM server then ensures that the user has the proper level of access to the CIMSERV resource in the (customer defined) WBEM RACF class. For read access to the system data exposed by the CIM server the user requires READ access, for manipulation of system resources the user requires UPDATE access and for performing administrative tasks against the CIM server itself the user requires CONTROL access to the CIMSERV RACF resource {AC.1::AC.1-R10-CIM-3}.

Handling of user authentication in the LDAP server

LDAP user authentication in the evaluated configuration will occur via digital certificates over SSL/TLS (LDAP SASL bind with EXTERNAL verification) or via an LDAP DN and a RACF password/phrase.

For users initiating a bind operation with a DN and a password/phrase, the processing that occurs will depend on the style of DN presented: LDBM, SDBM, or ICTX:

- If the user presents an SDBM-style DN (such as racfid=ID1,profiletype=user,SDBM-suffix) then LDAP will extract the racfid value. Note that the SDBM-suffix is configured by the administrator. LDAP will pass that user ID and password/phrase to RACF for authentication {IA.3::IA.3-LDAP-1}.
- Similar processing happens when users present an ICTX-style DN. Again, LDAP recognizes this based on a configured suffix value, and invokes the the ICTX plug-in, which will pass the RACF user ID from the DN and the password to RACF for authentication {IA.3::IA.3-R9-EIM-1}.
- For LDBM users, the “native authentication” functions of the server are required for any authenticated access to LDBM in the evaluated configuration. For each LDBM user, the LDAP administrator will define the user’s distinguished name (DN) in the LDBM database, together with the RACF user ID that corresponds to that DN. The LDAP LDBM user will provide his LDAP DN and the RACF password for the user ID specified by the administrator. LDAP will find the user-specified DN, then call RACF passing the administrator-specified user ID and the user-specified password {IA.3::IA.3-R8-LDAP-2}. Note that the evaluated configuration allows the administrator to configure selected LDBM data for access by users who have not authenticated, if the administrator decides that such access meets the security policies in effect for that data

For a SASL bind with a digital certificate (possible only for SDBM or LDBM in this evaluation), the evaluated configuration requires the administrator to configure LDAP to map the certificate to a RACF user ID. The administrator must specify the configuration option sslMapCertificate with a first operand of CHECK, ADD, or REPLACE and a second operand of FAIL. With this configuration in effect:

- If no RACF user ID is associated with the certificate, LDAP will fail the bind operation {IA.3::IA.3-R10-LDAP-2}.
- The mapped RACF user ID will be used for any access to the SDBM back-end {IA.3::IA.3-R10-LDAP-3}.
- For LDBM operations:
    - With ADD specified, LDAP will convert the mapped RACF user ID into an SDBM DN, and will add that DN to the DN from the certificate, using both DNs for group gathering and access decisions {IA.3::IA.3-R10-LDAP-4}.

- o With REPLACE specified, LDAP will convert the mapped RACF user ID into an SDBM DN, and will use only that DN for group gathering and access decisions {IA.3::IA.3-R10-LDAP-5}.

8.1.2.9 Authentication Method Summary

The following TOE applications support client authentication via Kerberos in the evaluated configuration:

- FTP {IA.3::IA.3-R8-FTP-AUTHKERB}
- ORSH {IA.3::IA.3-R8-RSH-AUTHKERB}
- OTELNET {IA.3::IA.3-R8-TELNET-AUTHKERB}
- NFS {IA.3::IA.3-R8-NFS-AUTHKERB}

The following TOE applications support client authentication via digital certificates when using SSL/TLS sessions in the evaluated configuration:

- TN3270, when using a TN3270 emulator that supports the Express Logon Facility (ELF) {IA.3::IA.3-R8-TN3270-AUTHSSL}
- FTP {IA.3::IA.3-R9-FTP-AUTHSSL}
- HTTP Server {IA.3::IA.3-R8-HTTP-AUTHSSL}
- LDAP Server, for SDBM or LDBM access {IA.3::IA.3-R10-LDAP-AUTHSSL}

The following TOE functions support authentication using passwords/phrases in the evaluated configuration:

- TSO/E {IA.3::IA.3-R10-TSO-AUTHPHRASE}
- NFS {IA.3::IA.3-R12-NFS-AUTHPHRASE}
- OpenSSH {IA.3::IA.3-R10-SSH-AUTHPHRASE}
- The z/OS UNIX shell commands su and passwd {IA.3::IA.3-R10-USS-AUTHPHRASE-1}
- The z/OS UNIX rlogin command {IA.3::IA.3-R10-USS-AUTHPHRASE-2}
- The C runtime functions __login(), __passwd(), pthread_security_np() (and the variants that accept an APPL ID), and getpass() {IA.3::IA.3-R10-LE-AUTHPHRASE}
- LDAP Server for SDBM or LDBM (via native authentication) access {IA.3::IA.3-R10-LDAP-AUTHPHRASE-1}

OpenSSH supports authentication via public/private key pairs stored in digital certificates when it is configured to store the certificates in RACF key rings {IA.3::IA.3-R12-SSH-AUTHRINGS}.

8.1.2.10 Handling of Groups During Authentication

During authentication, RACF and LDAP construct 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 user during logon if the application allows that specification. 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-R10-RACF-1}.

- When a user attempts to use 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}.

- For access to LDAP LDBM resources, the LDAP server gathers a list of groups based on the authentication data supplied by the user.

    - If the user supplied an LDBM-format DN and a RACF password/phrase, LDAP uses that DN to determine the LDBM groups to use on subsequent LDBM access checks {IA.3::IA.1.14-R10-LDAP-1}.

    - If the user supplied an SDBM-format DN and a RACF password/phrase, LDAP retrieves the subject's group(s) from the ACEE, and converts them into SDBM-format DNs, which become the groups to use for subsequent LDBM access checks {IA.3::IA.1.14-R10-LDAP-2}. Additionally, if LDAP is configured for extended group searching, LDAP derives additional groups by determining the LDBM groups to which the SDBM user belongs {IA.3::IA.1.14-R10-LDAP-3}.

    - If the user supplied a digital certificate for a SASL external bind, LDAP maps the DN in the certificate to a RACF user ID {IA.3::IA.1.14-R10-LDAP-4}. Subsequent processing depends on the value of the LDAP sslMapCertificate configuration parameter.

        - If “check” is specified, the certificate DN becomes the LDBM bind DN, and LDAP derives LDBM groups solely from that DN {IA.3::IA.1.14-R10-LDAP-5}.

- If “add” is specified, LDAP creates an SDBM DN from the mapped RACF user ID, and adds that SDBM DN as an alternate bind DN for authorization processing. LDBM processing derives LDBM groups from both the LDBM DN (in the certificate) and the SDBM DN {IA.3::IA.1.14-R10-LDAP-6}.

- If “replace” is specified, LDAP creates an SDBM DN from the mapped RACF user ID, and that SDBM DN becomes the only bind DN for LDBM authorization processing. LDAP gathers the mapped user's RACF groups, converts them to SDBM DNs, and uses them as LDBM groups for authorization checking {IA.3::IA.1.14-R10-LDAP-7}.

- Additionally, if configured to do so, LDAP also derives LDBM groups from the SDBM DN for the RACF user {IA.3::IA.1.14-R10-LDAP-8}.

8.1.2.11 Authentication-related differences between z/OS UNIX and typical non-z/OS UNIX systems

There are a few security aspects that are handled different in z/OS than in “standard” UNIX implementations. Those differences are:

1. Definition of users in /etc/passwd. In other UNIX systems, the file /etc/passwd contains the users defined and some of the user’s attributes. Within z/OS, the file /etc/passwd does not exist (or if it exists, does not contain any values used by the system). All user attributes are stored in the RACF user profile and managed solely by RACF {IA.4::IA.4.1}.
2. Handling of the su command: the handling of the su command depends on the existence of specific profiles in RACF.
3. Switching to a user identity by specifying a new user ID.

The su command allows the change if the user provides the correct password (like most other UNIX systems) {IA.4::IA.4.2}, or if the original user ID has read access to the BPX.SRV.newuser resource profile in the SURROGAT class {IA.4::IA.4.3}.

Note that, unlike in most other UNIX systems, this also applies to subjects running with UID 0.

1. Switching to a superuser identity (UID 0) without specifying a new user ID.

The su command allows the change if

a) the user is already running with UID 0 {IA.4::IA.4.4}

b) the original user ID has read access to the BPX.SUPERUSER resource profile in the FACILITY class {IA.4::IA.4.5}.

The shell started by the su command inherits the security label of the user who issued the command (Labeled Security Mode only) {IA.4::IA.4.6}). The new user must be authorized to the inherited security label or the su command fails (Labeled Security Mode only) {IA.4::IA.4.7}.

When a user executes a program that has the setuid bit set, only the effective user ID is changed to that of the owner of the file containing the program while the real user ID remains that of the caller {IA.4::IA.4.v111.1}. The RACF user ID is neither changed by the su command when changing to UID 0 using the su command without specifying a user ID {IA.4::IA.4.V1R7.1} nor by executing a program that has the setuid or setgid bit set {IA.4::IA.4.v111.2}.

When executing the su command to a user with a non-zero UID, or when specifying the userid and password with the su command when switching to a user with UID 0, all credentials including the RACF user ID are reset to the new user {IA.4::IA.4.V1R7.2}.

An executable file can have additional attributes (setuid and setgid bits) used to allow a program temporary access to files that are not normally accessible to other users. Those permission bits sets the effective user ID or group ID of the user process executing a program to that of the file whenever the file is run {IA.4::IA.4.V1R7.3}. The setuid and setgid bits are only honored for executable files containing load modules or REXX execs. These bits are not honored for shell scripts that reside in the file system {IA.4::IA.4.V1R7.4}.

When authorized to do so, a process executing in the z/OS UNIX System Services environment can change its real, effective, and saved set user IDs or the real, effective and saved user ID of process spawned off using dedicated system services. The following restrictions apply:

- The handling of the su command depends on the existence of specific profiles in RACF. the process is executing with UID 0 or the current subject has the trusted or privileged attribute {IA.4::IA.4.V1R7.5} or

- If User_ID is the same as the real UID of the process or the saved set UID, the setuid service sets the effective UID to be the same as User_ID {IA.4::IA.4.V1R7.6}.

The RACF user ID is changed if one of the following conditions is satisfied

- The calling process is executing with an effective UID 0, the calling user ID has been authorized to the BPX.DAEMON profile in the FACILITY class and the calling program has been loaded from a controlled library in a clean environment {IA.4::IA.4.V1R7.7}.

- The target user ID has been successfully authenticated by the password service {IA.4::IA.4.V1R7.8} or has SURROGAT authority to the new user ID {IA.4::IA.4.V1R7.9}. The TOE may also allow to change the real, effective, and saved set group IDs (GIDs) for the calling process. The following restrictions apply:

    - the process is executing with UID 0 or the current RACF user ID has the trusted or privileged attribute (\{IA.4::IA.4.V1R7.10} or

    - If Group_ID is equal to the real group ID or saved set group ID of the process, the effective group ID is set to Group_ID the process is executing with UID 0 or the current RACF user ID has the trusted or privileged attribute {IA.4::IA.4.V1R7.11}.

The setgid service does not change any supplementary group IDs of the calling process {IA.4::IA.4.V1R7.12}.

User identification and authentication are also performed by the telnet, rlogin, rsh, rexec, and ftp z/OS UNIX services (as described in Authentication function), the LDAP server, the HTTP Server, and the SSH daemon (sshd).

The BPX.DAEMON Profile in the FACILITY Class

When the BPX.DAEMON profile is defined in the FACILITY class of RACF, z/OS allows for a finer granularity of handling privileges of z/OS UNIX System Services.

Any superuser permitted to this profile has the daemon authority to change MVS identities via z/OS UNIX services without knowing the target user ID’s password {IA.4::IA.4.V1R7.13}. This identity change can only occur if the target user ID has an OMVS segment defined {IA.4::IA.4.V1R7.14}.

If the BPX.DAEMON FACILITY class profile is defined, then z/OS UNIX will verify that the address space has not loaded any executables that are uncontrolled before it allows any of the following services that are controlled by z/OS UNIX to succeed:

- seteuid
- setuid
- setreuid
- pthread_security_np()
- auth_check_resource_np()
- __login()
- spawn with user ID change
- __passwd()
- __certificate()

{IA.4::IA.4.V1R7.15}

Daemon authority is required only when a program does a setuid(), seteuid(), setreuid(), or spawn with user ID to change the current UID without first having issued an ___certificate() or an __passwd() call to the target user ID. In order to change the MVS identity without knowing the target user ID’s password, the caller of these services must be a superuser. Additionally, if a BPX.DAEMON FACILITY class profile is defined and the FACILITY class is active, the caller must be permitted to use this profile {IA.4::IA.4.V1R7.16}. If a program comes from a controlled library and knows the target UID’s password, or supplied the target’s certificate, it can change the UID without having daemon authority {IA.4::IA.4.V1R7.17}.

8.1.2.12 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 new 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.

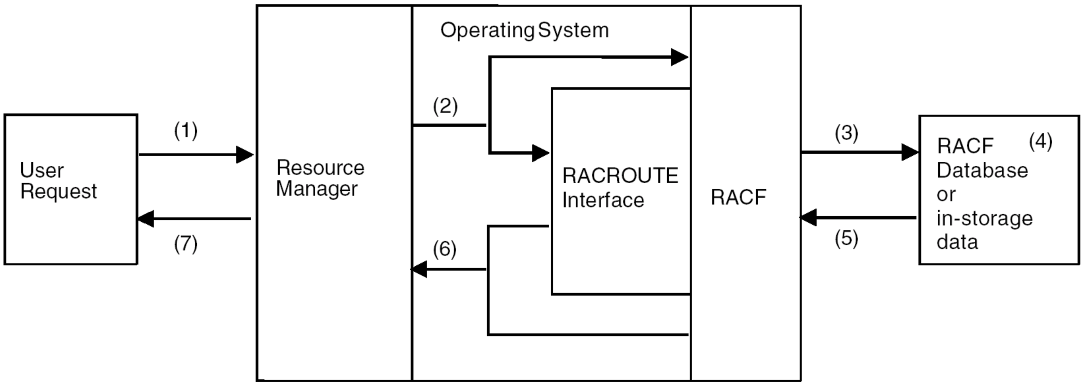
{IA.5::IA.5-R12-IDPROP-USS-1} The UNIX System Services __passwd (BPX1PWD) and pthread_security_np() (BPX1TLS) function allows appropriately authorized servers to assert a user identity and create a security environment by specification of the pseudo-userID and pseudo-password obtained via a prior authentication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

8.1.3 Discretionary Access Control

8.1.3.1 Access control principles

z/OS provides the Resource Access Control Facility (RACF) as the component that performs access control between subjects acting on behalf of a user and resources protected by the discretionary and (in Labeled Security Mode) 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)

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.



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.

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 perform an access check on its own behalf or on behalf of another user (typically a client of the server making the request). Note that these requests do not represent actual resource accesses that will occur within the TOE, but merely allow the TOE to provide access controls to processes running externally within the network if desired. Additionally, the client can specify any resource class known to RACF, except DATASET, and any resource name with legal RACF syntax that it chooses.

The LDAP client uses an LDAP extended-operation (which gets routed by the LDAP server to the ICTX plugin) to request this remote authorization function which can:

- Check the client's own authority to access a specified resource name in a specified RACF resource class {AC.2::AC.2-R9-EIM-1}. This usage of the remote authorization service requires the LDAP client to have READ authority to FACILITY resource IRR.LDAP.REMOTE.AUTH {AC.2::AC.2-R9-EIM-2}.

- Check a specified user's or group's authority to access a specified resource name in a specified RACF resource class {AC.2::AC.2-R10-EIM-3}. This usage of the remote authorization service requires the LDAP client to have UPDATE authority to FACILITY resource IRR.LDAP.REMOTE.AUTH {AC.2::AC.2-R9-EIM-4}.

8.1.3.2 Protected resources

The protected resources considered in this Security Target are:

- Data sets
- Volumes
- Devices
- Terminals
- TCP/IP connections
- Operator commands
- Programs
- Consoles
- UNIX file system objects
- UNIX IPC objects
- LDAP LDBM objects
- System logger objects
- Communication Server Policy Agent data
- Hardware management interface functions

As a general-access control system, RACF is capable of protecting a number of other resources, but those are not included in this evaluation. The reader should note that some other RACF classes are included in this evaluation that do not 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. Those classes and profiles are described in the relevant subsection of the access control section in this Security Target. The reader should also understand that granting privileges that are not described in this document should be done with care, and only for trusted users, as those privileges may allow administrative functions or extraordinary resource accesses.

Data sets

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}.

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::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.

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}.

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}

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}.

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 instead of OPERATIONS, 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-R12-RACF}.

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}
-

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}.

Volumes

By defining profiles in the DASDVOL class, the system administrator can define non-SMS-managed DASD volumes to RACF and authorize users to perform maintenance operations (such as dump, restore, scratch, and rename) without having access to the data set profiles protecting the data sets on the volume {AC.2::AC.2.40}. If a user does not have the necessary DASDVOL authority to a non-SMS-managed volume, he or she must have the necessary authority in the DATASET class to each of the data sets on the volume {AC.2::AC.2.41}.

Tape volumes are protected by profiles in the TAPEVOL class in the following circumstances:

- when the RACF TAPEVOL class is active and the IEHINITT utility is used to reinitialize a tape volume that contains a standard label {AC.2::AC.2.42-R8-1}

- when the RACF TAPEVOL class is active, and SETR NOTAPEDSN is in effect, and TAPEAUTHDSN=NO is specified in SYS1.PARMLIB(DEVSUPxx), and the tape contains standard labels, and a user accesses data on the tape {AC.2::AC.2.42-R8-2}.

Special Considerations for Data on Tape

A Data file located on tape can be protected in several different ways, depending on RACF and system options:

a) TAPEVOL class active, and SETROPTS NOTAPEDSN, and TAPEAUTHDSN=NO in SYS1.PARMLIB(DEVSUPxx): In this mode the data is protected by the TAPEVOL profile for the standard-labeled tape {AC.2::AC.2-R8-TAPE-1} or is unprotected if no profile exists or the tape has no labels {AC.2::AC.2-R8-Tape-2}.

b) TAPEVOL class inactive, and SETROPTS TAPEDSN, and TAPEAUTHDSN=NO in SYS1.PARMLIB(DEVSUPxx): In this mode the data is protected by the DATASET profile for the data set if the tape has standard labels or is unprotected if the tape has no labels {AC.2::AC.2-R8-TAPE-3}. However, in this mode, protection may be ineffective for data sets with names longer than 17 characters, and the physical tape volume labels record only the last 17 characters of a data set name. Therefore, this mode should be used only if an active tape management system (DFSMSrmm for the evaluated configuration) is keeping track of tape contents, and will reject the tape volume request if the data set name does not match the name specified by the user {AC.2::AC.2-R8-TAPE-4}.

c) TAPEVOL class active, and SETROPTS TAPEDSN, and TAPEAUTHDSN=NO in SYS1.PARMLIB(DEVSUPxx), and with TAPEVOL profiles that contain RACF TVTOCs: In this mode RACF verifies that the user has specified the correct data set name, and then security for the data set is provided by the DATASET profile for the data set, if the tape has standard labels {AC.2::AC.2-R8-TAPE-5}.

d) TAPEAUTHDSN=YES specified in SYS1.PARMLIB(DEVSUPxx): In this mode the system will check access based on the data set name specified by the user, regardless of the SETROPTS tape-related options in effect {AC.2::AC.2-R8-TAPE-6}.

e) TAPEAUTHF1=YES specified in SYS1.PARMLIB(DEVSUPxx) and either SETROPTS TAPEDSN specified or TAPEAUTHDSN=YES specified in SYS1.PARMLIB(DEVSUPxx): In this mode, in addition to the access check for the data set name specified by the user, the system will perform an additional check for the first data set on the tape {AC.2::AC.2-R8-TAPE-7}. Note: This mode requires an active tape management system (DFSMSrmm for the evaluated configuration) which provides the data set name for the first fileon the tape.

Note: For systems configured in Labeled Security Mode, configuration option (a) above must be used to ensure proper auditing of data export and import.

Devices

A user authorized to define profiles in the DEVICES class can use this class to control which users can allocate unit record devices, teleprocessing or communications devices, and graphics devices {AC.2::AC.2.43}. For example, the DEVICES class can be used to ensure that only authorized users can allocate devices by name. The DEVICES class can not be used to protect other kinds of devices, such as tape or DASD devices.

Terminals

Terminals are protected by profiles in the TERMINAL or GTERMINL class. A user must have at least read access authority assigned to a profile representing a terminal to be able to use the terminal {AC.2::AC.2.45}. The GTERMINL class is provided to protect a class of terminals in the same way without the need to define discrete profiles for each terminal in the TERMINAL class {AC.2::AC.2.46}. User access to terminals that are not protected by a profile in one of those classes is defined by the parameter in the TERMINAL operand in the SETROPTS command {AC.2::AC.2.47}. If this parameter is NONE, a user can not use such terminals to log in {AC.2::AC.2.48}. If the parameter is READ, a user can use those terminals to log in {AC.2::AC.2.49}.

Access to terminals can also be controlled for groups of users. If the option NOTERMUACC is defined in the group profile, users within this group can only use terminals to which they are specifically authorized on the access list in the TERMINAL profile protecting the terminal {AC.2::AC.2.50}.

The use of a terminal can also be restricted to specific days and a time period within those days using the WHEN and TIME options in the RDEFINE and RALTER command {AC.2::AC.2.51}.

If both the TERMINAL and the SECLABEL class are active, RACF checks a user’s authority to use a terminal. When RACF checks a user’s authority to use the terminal, the user must log on with a security label that is less than or equal to the security label of the terminal (Labeled Security Mode only) {AC.2::AC.2.52}.

TCP/IP connections

TCP/IP is a component of the Communications Server subsystem of the TOE. TCP/IP runs as a started task and provides the TCP, UDP, RAW, ICMP and IP functions. TCP/IP loads an INET Physical File System into the UNIX System Services kernel to handle socket requests. TCP/IP connects to the VTAM® component of the Communications Server subsystem of the TOE for physical communications device management services. Up to eight instances of the TCP/IP started task may be run concurrently on one instance of the TOE to isolate networks or stacks by security label. Socket applications may be directed to a particular stack or may transparently span multiple stacks.

Several TCP/IP resources can be protected by resources in the SERVAUTH class:

- Access to a particular TCP/IP stack is controlled when an application opens a socket by read access to a profile in the form “EZB.STACKACCESS.system-name.stack-name” where system-name is the name of the TOE image and stack-name is the job name of the particular stack {AC.2::AC.2.53}.

- Access to a particular IP address is controlled when an application explicitly binds a socket to a local address and when an application sends data to or receives data from a peer address. IP addresses are configured into named security zones within the stack using NETACCESS profile statements. Access to a particular security zone is controlled by read access to a profile in the form “EZB.NETACCESS. system-name.stack-name.SAF-resname” where system-name is the name of the TOE image, stack-name is the job name of the particular stack and SAF-resname is the name configured on the NetAccess statement {AC.2::AC.2.54}.

- Access to the intranode managementnetwork iscontrolled when an application attempts to start a TCP connection, or read or write any data, that traverses the intranode management network using an OSA-Express chpid of type OSM. Access to the intranode management network is controlled by read access to a profile in the form “EZB.OSM.system-name.stack-name” where system-name is the name of the TOE image, and stack-name is the job name of the particular stack. {AC.2::AC.2-R12-CS-62}

TCP/IP makes point of access information available on sockets for use when processing user login requests. This information may be requested by applications. The UNIX Systems

Services subsystem will request this information on behalf of an application when it invokes the __poe() service. The information provided by TCP/IP includes {AC.2::AC.2.56}:

- The fully-qualified SERVAUTH resource name of the NETACCESS security zone containing the peer IP address, if it is in a security zone.
- The TERMINAL resource name of the peer IP address, if it is an IPv4 address.
- The security label to use if the RACF option MLACTIVE is set and the peer security zone has a SYSMULTI security label.
- Access to a particular port is controlled when an application explicitly binds a socket to a local port. Applications binding to low ports (below 1024) must be a UNIX superuser or APF-authorized. Port usage may also be controlled by configuring the Port statement in the TCP/IP profile. Control may be by user ID, job name, or read access to a profile in the form “EZB.PORTACCESS.system-name.stack-name.SAF-resname”, where system-name is the name of the TOE image, stack-name is the job name of the particular stack, and SAF-resname is the name configured on the Port or Portrange statement {AC.2::AC.2.55}. The port access functions will work for both reserved and (if configured via PORT UNRSV) for unreserved ports {AC.2::AC.2-R10-CS-PORT-1}.

TCP/IP performs additional access control when the RACF option MLACTIVE is set (in Labeled Security Mode). All profiles in the SERVAUTH class must have security labels defined. Sockets are always considered to be read/write objects so all MAC checks on SERVAUTH profiles require equivalent security labels.

- In Labeled Security Mode: The security label on the STACKACCESS profile must be identical to the security label of the stack job. Only applications running under an equivalent security label may access a given stack. A stack running under the SYSMULTI label may be accessed by applications with any security label but communications will be allowed only between applications with equivalent security labels {AC.2::AC.2.57}.
- In Labeled Security Mode: The security label on the NETACCESS profile for each local interface address must be identical to the security label of the stack job. This ensures that all implicit address assignments are equivalent to the application security label {AC.2::AC.2.58}.
- In Labeled Security Mode: The security label on the NETACCESS profile for each local VIPA must be equivalent to the stack security label of the stack job and may be SYSMULTI only when the stack job is also SYSMULTI. When SourceVIPA processing is enabled, a VIPA with a security label equivalent to the application will be chosen as the implicit source address {AC.2::AC.2.59}.
- In Labeled Security Mode: Communications will only be permitted when the source IP address and the destination IP address are in NETACCESS security zones with equivalent security labels {AC.2::AC.2.60}. Additionally, when both security zones have SYSMULTI labels, the security label of the sending application will be recorded

in the IP header using a proprietary format. These proprietary packets are restricted to IUTSAMEHOST links between stacks on the same TOE or XCF links between stacks on the same sysplex {AC.2::AC.2.61}.

The Communications Server subsystem of the TOE provides numerous commands and applications. For Labeled Security Mode: There are documented restrictions on usage and configuration of these when RACF option MLACTIVE is set.

Operator commands

Operator commands can be protected by resources in the OPERCMDS class. Resources in this class are the individual commands specified in the form “subsystem-name.command-name” where subsystem-name is the name of the processing environment of the command (JES2, RACF, or MVS, for example). Access to an operator command protected by a RACF profile requires the appropriate access authority in the access control list of the profile for the command {AC.2::AC.2.64}. Note that if the class is active and a command is not protected by a profile it is not allowed to be executed.

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:

- 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}.
- The user's group or user ID is associated with the program name in the conditional access list {AC.2::AC.2.V1R7.19}.
- 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}.
- 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}.
- 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}.

- 1.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}.

Consoles

When the CONSOLE class is active and a console being used is protected by a profile in the CONSOLE class, RACF ensures that the person attempting to logon at this console has the proper authority to do so {AC.2::AC2.V1R7.24}. Using RACF, the use of system consoles can be controlled {AC.2::AC2.V1R7.25}.

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 of the TOE {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}.

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.

LDAP LDBM objects

LDAP LDBM objects (objects in an LDBM backend for a z/OS LDAP server) exist in a single administrator-configured file (LDBM database) in the UNIX file system for each suffix the LDBM backend supports in each server {AC.2::AC.2-R8-LDAP-1} and are subject to discretionary access control by the LDAP server itself (not by RACF) using standard LDAP ACLs {AC.2::AC.2-R8-LDAP-2}and/or LDAP Filter ACLs {AC.2::AC.2-R12-LDAP-13}. LDAP objects are organized hierarchically in a tree format, and each object has a distinguished name (DN) which both names the object and locates it within the tree {AC.2::AC.2-R8-LDAP-3}.

Users do not have direct access to the data (in the sense that they have for, say, data access via FTP or NFS). Rather, users make requests to the LDAP server specifying the named objects to retrieve, and the server interprets those requests, locates the named objects, and acts on them if the user has the proper authority {AC.2::AC.2-R8-LDAP-4}.

Users who have not performed a bind or have performed an anonymous bind are called unauthenticated or anonymous. There is no difference between the access rights given to unauthenticated and anonymous user {AC.2::AC.2-R8-LDAP-6}. Administrators may allow access to anonymous users {AC.2::AC.2-R8-LDAP-7} or deny access to anonymous users {AC.2::AC.2-R8-LDAP-8} anywhere they choose within the LDAP tree {AC.2::AC.2-R8-LDAP-9}. By default anonymous access is allowed {AC.2::AC.2-R8-LDAP-10}.

For further information see Algorithm to check for DAC access to LDAP LDBM objects.

System Logger objects

System logger resources, such as log streams and the coupling facility structures associated with them are subject to discretionary access control. For more information about those objects and RACF profiles used to protect them, see the section on the management of system logger objects in the management section

Communication Server Policy objects

Communication Server Policy objects can be read by users that have at least read access to the profiles protecting those objects. For more information about those objects and the RACF profiles that protect them see the section on the Communication Server Policy Agent later in this document.

Hardware Management Interface Functions

The System z processors provide privileged instructions that allow an operating system running in a logical partition (LPAR) to perform hardware management functions (e.g., activating or deactivating processors, IPLing an operating system into an LPAR, adjusting LPAR performance characteristics, adjusting LPAR definitions, etc.) When defining an LPAR to PR/SM an administrator specifies whether the LPAR can control other LPARs by setting the Cross-Partition Authority Flag in the new LPAR definition. LPARs without that flag set can not control other LPARs {AC.2::AC.2-R11-BCPii-1}.

z/OS provides a function called BCPii which makes hardware management functions available to authorized (supervisor state or having a PKM allowing system key (0-7)) programs {AC.2::AC.2-R11-BCPii-2}. After the previous authorization check succeeds, BCPii also uses RACF resources in the FACILITY class to control which users can connect to it and to determine which users can use the hardware management interfaces:

- The HWICONN function allows an application to establish a logical connection to a Central Processor Complex (CPC), a CPC image (LPAR), a Capacity Record or different types of activation profiles.

When the application is done with its processing, it would use the HWIDISC function to disconnect.

Use of these functions requires READ access to HWI.APPLNAME.HWISERV {AC.2::AC.2-R11-BCPii-3}. Then, for HWICONN users also need:

  - READ access to HWI.TARGET.netid.nau to establish a CPC or activation profile connection {AC.2::AC.2-R11-BCPii-4};
  - READ access to HWI.TARGET.netid.nau.imagename to establish an image connection {AC.2::AC.2-R11-BCPii-5}; or
  - READ access to HWI.CAPREC.netid.nau.caprecid for capacity record connections {AC.2::AC.2-R11-BCPii-6}.

- The HWIQUERY function allows an application to retrieve information about Hardware Management Console (HMC) managed objects associated with Central Processor Complexes (CPCs), CPC images (LPARs), capacity records, or different types of activation profiles.

Use of HWIQUERY requires READ access to FACILITY resource HWI.APPLNAME.HWISERV {AC.2::AC.2-R11-BCPii-7} and also

  - READ access to HWI.TARGET.netid.nau for CPC connection or activation profile queries {AC.2::AC.2-R11-BCPii-8};
  - READ access to HWI.TARGET.netid.nau.imagename for image queries {AC.2::AC.2-R11-BCPii-9};
  - READ access to HWI.CAPREC.netid.nau.caprecid for capacity record queries {AC.2::AC.2-R11-BCPii-10}

- The HWICMD function allows an application to perform a command against a Hardware Management Console (HMC) managed object associated with Central Processor Complexes (CPCs), CPC images (LPARs) and capacity records.

Use of HWICMD requires READ access to FACILITY resource HWI.APPLNAME.HWISERV {AC.2::AC.2-R11-BCPii-11} and also

  - CONTROL access to HWI.TARGET.netid.nau for ConnectTokens representing CPC connections {AC.2::AC.2-R11-BCPii-12};
  - CONTROL access to HWI.TARGET.netid.nau.imagename for ConnectTokens representing image connections {AC.2::AC.2-R11-BCPii-13};

- The HWILIST function allows an application to retrieve Hardware Management Console (HMC) and/or Hardware Management Interface (HWI) configuration-related information. Depending on which information is requested, the data returned by this service can be used on subsequent HWI service calls to connect to a Central Processor Complex (CPC), image (LPAR), Capacity record (CAPREC), Reset activation profile, Image activation profile, or Load activation profile, or to register for the proper events (HWIEVENT callable service).

Use of HWILIST requires READ access to FACILITY resource HWI.APPLNAME.HWISERV {AC.2::AC.2-R11-BCPii-14} and also

  - READ access to the HWI.TARGET.netid.nau for each non-local CPC to be listed {AC.2::AC.2-R11-BCPii-15};
  - READ access to the HWI.TARGET.netid.nau.imagename for each image to be listed {AC.2::AC.2-R11-BCPii-16};
  - READ access to the HWI.TARGET.netid.nau for each CPC whose activation profiles are to be listed {AC.2::AC.2-R11-BCPii-17};
  - READ access to the HWI.CAPREC.netid.nau.caprecid for each CPC capacity record to be listed {AC.2::AC.2-R11-BCPii-18};
  - READ access to the HWI.TARGET.netid.nau for each CPC whose events are to be listed {AC.2::AC.2-R11-BCPii-19};
  - READ access to the HWI.TARGET.netid.nau.imagename for each image whose events are to be listed {AC.2::AC.2-R11-BCPii-20}.

- The HWIEVENT function allows an application to register for notification of hardware and software events related to a CPC or image, or to delete such a registration.

  Use of HWIEVENT requires

  - READ access to HWI.TARGET.netid.nau for a ConnectToken representing a CPC connection {AC.2::AC.2-R11-BCPii-23};

  - o READ access to HWI.TARGET.netid.nau.imagename for a ConnectToken representing an image connection {AC.2::AC.2-R11-BCPii-24}.

- The HWISET function allows an application to change or set data for Hardware Management Console (HMC) managed objects associated with Central Processor Complexes (CPCs), CPC images (LPARs), or activation profiles.

  Use of HWISET requires requires

  - READ access to FACILITY resource HWI.APPLNAME.HWISERV {AC.2::AC.2-R11-BCPii-27} and also UPDATE access to HWI.TARGET.netid.nau for values related to a CPC or an activation profile {AC.2::AC.2-R11-BCPii-21};
  - UPDATE access to HWI.TARGET.netid.nau.imagename for image-related values {AC.2::AC.2-R11-BCPii-22}.

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}

LDAP LDBM objects are not subject to mandatory access control in the same way as other resources. Rather, a complete LDBM database has a single SECLABEL, neither SYSMULTI nor SYSNONE, derived from the label of the UNIX file that contains the database {AC.3::AC.3-R8-LDAP-1}. The LDAP/LDBM server runs with a specific security label, matching that of the database it will read/write, and serves data with that specific label to users with the same label {AC.3::AC.3-R8-LDAP-3}. This satisfies the overall data flow requirements of MAC processing. To serve data with different labels, the administrator may configure multiple LDAP/LDBM servers, each running with the appropriate label, and the client must connect to the appropriate server {AC.3::AC.3-R8-LDAP-2}.

Printers (as examples of devices) and terminals can be restricted to the security labels allowed to be used with them {AC.3::AC.3.13}. This allows for example to restrict user logon or printer output with critical security labels to defined terminals resp. printers.

Each page of printer output is labeled with the security label of the subject that initiated the output. The printed security label is in human readable format {AC.3::AC.3.14}. The exact text of this label can be defined during system configuration {AC.3::AC.3.15}.

Communication channels within a TOE, even for a TOE consisting of multiple systems coupled into a sysplex can be multi-level, whereas other communication channels are assigned a single security label {AC.3::AC.3.16}.

A user can define the security label of a session when he performs his TSO login or when submitting a batch job {AC.3::AC.3.17}. At that time he can specify the security label of the session / job to any security label assigned for him by the system administrator {AC.3::AC.3.18}. A user needs to start a new session or job when he wants to work with a different security label (from the set of security labels allowed for him). In all other cases the security label is defined by the user’s default label, by the port-of-entry or by the application {AC.3::AC.3.19}. The user’s security label can be restricted by the allowed security label for the port-of-entry or it can be restricted by the application he is connecting to.

Data can be exported with its labels attached by storing the data in a z/OS UNIX zFS file system {AC.3::AC.3.20}. Each zFS file system is implemented within a single z/OS data set. To be able to create files and directories with different security labels in the zFS file system, the z/OS data set hosting the zFS file system must be labeled as SYSMULTI {AC.3::AC.3.21}.

When the z/OS data set containing the zFS file system is exported, all the security labels associated with the files and directories in this zFS file system are exported because they are included as extended attributes in the i-nodes of the file system {AC.3::AC.3.22}. The importing system needs to define the security labels compatible with the exporting system to ensure that the security labels are interpreted consistently.

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}.Discretionary access control

8.1.3.3 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, UNIX resources, and LDAP LDBM resources. RACF provides the discretionary access controls for MVS and UNIX resources; the LDAP server provides the discretionary access controls for LDAP LDBM objects. See the sections above on the different profiles for details on what is stored in those profiles.

DAC for MVS resources

RACF controls the types of access to all MVS (non-UNIX, non-LDAP) 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. The semantics for MVS data sets are:

- ALTER

ALTER allows users to read, update, delete, rename, move, or scratch the data set.

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

For VSAM data sets, CONTROL is equivalent to the VSAM CONTROL password; that is, it allows users to perform improved control interval processing. This is control-interval access (access to individual VSAM data blocks), and the ability to retrieve, update, insert, or delete records in the specified data set {AC.4::AC.4.5}.

For non-VSAM data sets, CONTROL is equivalent to UPDATE {AC.4::AC.4.6}.

- UPDATE

Allows users to read from, copy from, or write to the data set {AC.4::AC.4.7}. UPDATE does not, however, authorize a user to delete, rename, move, or scratch the data set {AC.4::AC.4.8}.

- READ

Allows users to access the data set for reading only {AC.4::AC.4.9}. (Note that users who can read the data set can copy or print it.)

- EXECUTE

For a private load library, EXECUTE allows users to load and execute, but not to read or copy programs (load modules) in the library {AC.4::AC.4.10}.

- 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::AC4.20}.

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.

Algorithm to check for DAC access to MVS resources

RACF performs the following checks to identify, if a subject has the requested type of access to an object 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}. For example:

   - For tape and DASD data sets, if the user ID of the requesting user is the high-level qualifier of the data set name, RACF grants the request

   - For spool data sets, if the JESSPOOL class is active, RACF compares the user ID and node of the requester with the user ID and node of the creator of the spool data set (using the security token). If the user IDs match, RACF grants the request.

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}.

   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.)

   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}
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}
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. If the user is in the list with insufficient authority RACF continues processing at step 11. {AC.4::AC.4-R12-RACF-54}
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.

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}
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}.

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.

    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}

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}.

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.dataset-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}.

8.1.3.4 DAC for System Logger Objects in the LOGSTRM class

DAC for System Logger objects in the LOGSTRM class uses the basic MVS DAC algorithm explained above. The DAC algorithms apply in two cases:

- application programs that merely need to read or write to a log stream. The standard MVS DAC algorithm applies, using READ access for reading only, or UPDATE access for reading and writing, to resource log_stream_name in the LOGSTRM class {AC.4::AC.4-R10-Logger-1}.
- application programs that want to perform system management functions: defining, deleting, or updating the log stream definitions. The Security Management section will cover those usages.

8.1.3.5 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.

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.

- (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}.
- 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}.
- 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}.

- 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}.
- 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}.
- 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}.
- 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}.
- 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}.
- 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}.
- 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}.
- 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}.
- 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}.
- 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.
- 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}.

- 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}.

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}.

8.1.3.6 DAC for LDAP LDBM objects

Access to LDAP directory entries and attributes is defined by Access Control Lists (ACLs). Each entry in the directory contains a special set of attribute/value pairs which describe who is allowed to access information within that entry. Attributes associated with access control are **aclEntry, aclPropagate, aclSource, entryOwner, ownerPropagate,** and **ownerSource**. The aclEntry and entryOwner attributes appear to be part of the entry, but may in fact be logically associated with an entry, but physically present in some parent entry higher in the directory tree. When we talk about an LDAP ACL (Access Control List) we mean the combination of the entryOwner and aclEntry attribute values. If the user is the entryowner they have administrator level permissions to the entry. If they are not the entryOwner then we look to the aclEntry attribute values to determine the access.

The TOE controls access to all directory entry objects based on the following security attributes:

- Entry Owner Information

  - entryOwner: defines the DN(s) of the LDAP user(s) or group(s) considered to own this entry.
  - ownerPropagate: indicates whether to propagate the ownership of the entry to all descendant entries, until another entry with ownerPropagate is found.

- Access Control Attributes (ACA)
  - aclEntry: defines the access control information, which can specify access permissions (grant, deny) for LDAP users or groups that control access to the complete entry, specific named attributes in the entry, or all attributes in the entry that belong to a specific attribute class

    aclEntry specifications can also filter the user's access based on various characteristics such as the bind DN , alternate Dns , pseudo Dns , groups that the bind/alternate DNs belongs to , IP address of the client connection , time of day that directory entry was accessed , day of week that directory entry was accessed , the bind mechanism used , whether or not bind encryption was used.
  - aclPropagate: indicates whether to propagate access control information of the entry to all descendant entries, until another entry with aclPropagate is found.

Algorithm to check for DAC access to LDAP LDBM objects

The Access Control List for an LDAP LDBM object (entry DN) is determined in the following way:

a) If there is a set of explicit access control attributes for the object , then the object’s Access Control List applies {AC.4::AC.4-R8-LDAP-1}.

b) If there is no explicitly defined set of access control attributes, then traverse the directory tree upwards until an ancestor node is reached with a set of propagating access control attributes {AC.4::AC.4-R8-LDAP-2}.

If no such ancestor node is found, the default access rights will apply {AC.4::AC.4-R8-LDAP-3}. The default access rights are predefined as aclEntry: group:CN=ANYBODY:normal:rsc:system:rsc and cannot be changed by the Directory Administrator {AC.4::AC.4-R8-LDAP-4}.

When determining base access (before filtering), processing stops as soon as access can be determined {AC.4::AC.4-R12-LDAP-5} based on access evaluation as described below:

- The first check for access is done by comparing the subject’s LDAP user ID (bind DN) and LDAP groups with the effective entryOwner attribute values. If there is a match with any of the entryOwner values then the subject has full access to the object {AC.4::AC.4-R8-LDAP-6}. The LDAP Administrator is additionally considered to have ownership authority for all objects in the directory tree {AC.4::AC.4-R8-LDAP-7}.
- The subject may be granted different access permissions to an object, from specific access permissions for the subject’s DN and from group memberships (including the

authenticated and anybody groups). The LDAP server uses the following algorithm to determine which permissions to grant a DN based on the values in the aclEntry attribute:

- if there is a specific value for the subject’s DN, the subject gets those permissions only {AC.4::AC.4-R8-LDAP-8}
- else if there is a cn=this value and the subject’s DN is the distinguished name (DN) of the object, the subject gets those permissions only {AC.4::AC.4-R8-LDAP-9}
- else if there are one or more group values that the subject is a member of, the subject gets the union of the permissions for those groups {AC.4::AC.4-R8-LDAP-10}
- else if there is a cn=authenticated value and the subject is authenticated to the directory with an LDAP bind operation, the subject gets those permissions only {AC.4::AC.4-R8-LDAP-11}
- else if there is a cn=anybody value, the subject gets those permissions only {AC.4::AC.4-R8-LDAP-12}
- otherwise the subject gets no permissions {AC.4::AC.4-R8-LDAP-13}

Permissions may be add (a) or delete (d) or both at the object level {AC.4::AC.4-R8-LDAP-17}, or read (r), write (w), search (s), or compare (c) or a combination of these at the attribute {AC.4::AC.4-R8-LDAP-18} or attribute class {AC.4::AC.4-R8-LDAP-19} level.

Permissions may specify grant or deny for any of the above {AC.4::AC.4-R8-LDAP-23}.

Each of the access permissions is discrete. One permission does not imply another. {AC.4::AC.4-R8-LDAP-14}

Permissions may be specified for the attribute classes normal, sensitive, critical, restricted, or system {AC.4::AC.4-R8-LDAP-20}

Administrator-defined attributes may be specified to be in the normal, sensitive, or critical attribute classes {AC.4::AC.4-R8-LDAP-21}. The default attribute class for administrator-defined attributes is normal {AC.4::AC.4-R8-LDAP-22}.

With the support for attribute-level permissions as well as grant/deny support, the order of evaluation of the separate permissions clauses is important. The access control permissions clauses are evaluated in a precedence order, not in the order in which they are found in the ACL entry value {AC.4::AC.4-R8-LDAP-15}. With this support, there are four types of permissions settings: access-class grant permissions, access-class deny permissions, attribute-level grant permissions, and attribute-level deny permissions. The precedence for these types of permissions is as follows (from highest precedence to lowest): {AC.4::AC.4-R8-LDAP-16}

- attribute-level deny permissions

- attribute-level grant permissions
- access-class deny permissions
- access-class grant permissions

Using this precedence, a deny permission takes precedence over a grant permission (for the same item specified) while attribute-level permissions take precedence over access-class permissions.

After a user's base access has been determined, it may be modified by Filter ACL entries.

{AC.4::AC.4-R12-LDAP-24}Filter ACLs can set permissions based on any of the following:

- bind DN
- alternate Dns
- pseudo Dns
- groups that the bind or alternate DNs belong to
- IP address of the client connection
- time of day that directory entry was accessed
- day of week that directory entry was accessed
- the bind mechanism used
- whether or not bind encryption was used

Filters support wildcards. Filters will have the same syntax support as LDAP search filters, where logical rules can be specified, such as “&” (and), “|” (or), and “!” (not), to name a few {AC.4::AC.4-R12-LDAP-25}.

{AC.4::AC.4-R12-LDAP-26} To allow flexibility, the new aclEntry filtering mechanism will also support three operation values that allow users to specify the way in which filtered ACLs will take effect:

- replace --the base effective ACL is replaced by the filtered ACLs.

  If administrators want a client from a given IP address to only have a specific set of permissions, they would use replace. (This is similar to what Sun ONE supports.)

- union --the base effective ACL is unioned with the filtered ACLs, resulting in a new effective ACL. This would be used to expand permissions.

  If administrators want a client from a given IP address to have a specific set of permissions, at a minimum, they would use union.

- intersect --the base effective ACL is intersected with the filtered ACLs. This would be used to reduce permissions.

  If administrators want a client from a given IP address to have a specific set of permissions, if and only if they already have the permissions, they would use intersect.

As with the operator precedence described above, filter ACL entries specifying “deny” take precedence over entries specifying “grant” {AC.4::AC.4-R12-LDAP-27}.

8.1.3.7 Access Control Considerations for the ISPF Client Gateway

The ISPF Client Gateway allows two additional ways for users to execute TSO/E commands and ISPF functions:

- The administrator can configure the HTTP server to allow the HTTP client to request invocation of the ISPF Client Gateway control program ISPZINT. In this case, ISPZINT and any commands it invokes will run with the user ID configured by the administrator (which may be the client user's identity or an identity specified by the administrator) {AC.4::AC-R10-ISPF-1}.

Additionally, in Labeled Security Mode the commands will run with the security label of the HTTP server, which the specified or client identity must have access to {AC.3::AC-R10-ISPF-2}.

- An existing user session (batch job, UNIX process, etc.) can invoke ISPZINT directly. In this case, ISPZINT and any comands it invokes will run with the user ID of the invoking user {AC.4::AC-R10-ISPF-3}.

Additionally, in Labeled Security Mode the commands will run with the security label of the invoking user session {AC.3::AC-R10-ISPF-4}.

Based on information supplied by its client, ISPZINT will either run a single command or it will run a command and leave the session active to run subsequent commands. When leaving the session active for subsequent commands, ISPZINT will ensure that those commands run with the same user ID as the original command {AC.4::AC-R10-ISPF-5}.

Additionally, in Labeled Security Mode the subsequent commands will run with the same security label as the original command {AC.3::AC-R10-ISPF-6}.

8.1.4 Communication security

z/OS provides communications security functions in several system components:

- Communications Server (stack access control, packet filter, IPSec, Application Transparent TLS),
- System SSL (SSL, TLS)
- Network Authentication Service (Kerberos, GSSAPI)

- NFS client and server (using Kerberos and GSSAPI functions supplied by the Network Authentication Service)
- IBM Ported Tools for z/OS (OpenSSH)

8.1.4.1 Communications Server

z/OS provides basic networking functions with the Communication Server component. This subsystem provides support for network communication using the IBM SNA protocols as well as the TCP/IP protocol suite. APIs for both protocol stacks are provided. For IP, both IPv4 and IPv6 are supported.

The Communications Server uses RACF to protect access of users to the following resources:

- the TCP/IP stack in general {CS.1::CS.1.1}
- TCP and UDP ports {CS.1::CS.1.2}
- IP addresses {CS.1::CS.1.3}
- Centralized policy information for QoS (Qualities of Service), PBR (Policy-Based Routing), IPSec, IDS (Intrusion Detection Services), and AT-TLS policy {CS.1::CS.1-R9-CS-POLCEN-1}.
- Network management information related to IP Filters and IPSec security associations {CS.1::CS.1-R9-CS-SECMON-1}
- Network Security Services, which IKE daemons can use to perform RSA signature generation and verification, and RSA and ECDSA (ECDSA supported only with IKE v2) certificate management/validation functions {CS.1::CS.1-R12-CS-NSS-5} at a centralized server, and which XMLAppliance clients can use to perform remote RACF authentication and access control functions {CS.1::CS.1-R10-CS-NSS-1}, certificate management functions {CS.1::CS.1-R11-CS-NSS-2}, private key retrieval {CS.1::CS.1-R11-CS-NSS-3}, and RSA signature creation and RSA decryption {CS.1::CS.1-R11-CS-NSS-4}.

z/OS provides the following security functions as part of the Communications Server:

- Access Control for the IP stack and access control to ports and port ranges

The IP stack as well as TCP/UDP ports and port ranges can be protected with RACF. Users can be granted or denied access to the IP stack in general as well as to individual ports and port ranges. See TCP/IP connections for the associated security claims.

- z/OS Communications Server provides packet filtering functionality that can control information flow into or out of the system based on security characteristics of the packets or of the network interface they use.

- IPSec security associations

The Communications Server can be configured to establish IPSec security associations at the IP layer. All packets transmitted between security association endpoints will be authenticated, encrypted, or both using the configured algorithms. The Communications Server provides support for IPSec-protected communication in accordance with RFCs 4301 through 4305, 4308, and 4835 {CS.1::CS.1-R12-IPSec-1} and IKEv2 in accordance with RFCs 4306 through 4308, 4718, 4753, 4754, 4809, 4868, 4869 and 4945 {CS.1::CS.1-R12-IPsec-6}. It also provides the IKE application that negotiates IPSec security association parameters with communication peers {CS.1::CS.1-R8-IPSec-2}. IKE is configured through the PROFILE.TCPIP configuration and the Policy Agent (see section Network configuration and management).

IPSec when authenticating using certificates will obtain the subject alternate extension present in the client's certificate and compare it contents to the identity defined in the IKE security policy {CS.1::CS.1-R12-IPSec-7}.

- A Network Security Services (NSS) server that can be used by:
    - IKE daemons to perform RSA signature generation and verification and certificate management/validation from a centralized location, minimizing the number of systems on which digital certificates for the IKE daemons must be installed t
    - Network management applications to monitor and manage ipsec on NSS client nodes (see the Network Management section)
    - XMLAppliance applications to remotely peform RACF user authentication and access control calls for RACF resources that the application specifies, as well as certificate management operations, retrieval of private keys from RACF certificates, and RSA signature generation and RSA decryption using ICSF-protected keys.

The Network Security Server will authenticate its clients using the RACF user ID and password or PassTicket that they provide and will ensure that the connection is protected by AT-TLS {IA.1::IA-R9-CS-NSS-1}.

For the IKE certificate-based processing, the Network Security Server will authorize use of its services via resources in the SERVAUTH class:

- EZB.NSS.sysname.clientname.IPSEC.CERT to control whether the client can request certificate services {AC.4::AC-R9-CS-NSS-1}
- EZB.NSSCERT.sysname.mappedlabelname.CERTAUTH to control whether the client can access a CERTAUTH certificate on the NSS server’s key ring {AC.4::AC-R9-CS-NSS-2}.
- EZB.NSSCERT.sysname.mappedlabelname.HOST to control whether the client can access a personal or SITE certificate on the NSS server’s key ring {AC.4::AC-R9-CS-NSS-3}.

For the XMLAppliance certificate processing, the Network Security Server will authorize use of certificates and private keys via resources in the SERVAUTH class:

- EZB.NSSCERT.sysname.mappedlabelname.CERTAUTH or EZB.NSSCERT.sysname.mappedlabelname.HOST to control whether the client can access a certificate on the NSS server's keyring. {AC.4::AC-R11-CS-NSS-9}
- EZB.NSSCERT.sysname.mappedlabelname.PRIVKEY to control whether the client can access a private key, either through direct retrieval or for usage in RSA signature generation or RSA decryption.{AC.4::AC-R11-CS-NSS-10}

The Network Security Server will authorize use of the network management service via the EZB.NSS.sysname.clientname.IPSEC.NETMGMT resource in the SERVAUTH class. NSS IPSec clients thata connect with a user ID permitted to this resource are allowed to utilize the IPSec monitoring and management service provided by the NSS server {AC.4::AC-R12-CS-NSS-4}.

The Network Security Server will authorize the use of XMLAppliance services as follows:

- RACF user authentication and access control calls require READ access to the EZB.NSS.sysname.clientname.XMLAPPLIANCE.SAFACCESS resource in the SERVAUTH class. {AC.4::AC-R10-CS-NSS-5}.
- Certificate management calls require READ access to the EZB.NSS.sysname.clientname.XMLAPPLIANCE.CERT resource in the SERVAUTH class {AC.4::AC-R11-CS-NSS-6}
- Private key retrieval calls require READ access to the EZB.NSS.sysname.clientname.XMLAPPLIANCE.PRIVKEY resource in the SERVAUTH class {AC.4::AC-R11-CS-NSS-7}

- RSA signature generation and RSA decryption calls require READ access to the EZB.NSS.sysname.clientname.XMLAPPLIANCE.PRIVKEY resource in the SERVAUTH class {AC.4::AC-R11-CS-NSS-8}
- SSL / TLS layer to set up a trusted channel to another trusted IT product, in a way transparent to the application (called Application Transparent TLS, or AT-TLS). The selectable algorithms can be limited by configuring a subset of allowable algorithms at the server. The SSL/TLS protocol can be used to set up a trusted channel to another system through a potentially insecure network. SSL/TLS protects the data against disclosure and attacks related to integrity like undetectable modifications or replay. Servers can support encryption using TDES with 168-bit key length or AES with either 128- or 256-bit key length. Application Transparent Transport Layer Security (AT-TLS) supports the use of all cipher suites supported by System SSL {CS.1::CS.1.4}. The TN3270 and FTP protocols are enabled to use AT-TLS and can be tunneled through SSL/TLS to establish a trusted channel to another trusted IT product

that also implements this protocol {CS.1::CS.1.5}. Applications that AT-TLS has been configured to support, can be tunneled through SSL/TLS to establish a trusted channel to another trusted IT product that also implements this protocol {CS.1::CS.1-V1R7.1}.

- An rpcbind application with the following characteristics:

- Control over which users can register or deregister application port information, which prevents unauthorized users from directing application RPC requests to the wrong TCP/IP port. To implement this security control, administrators define a RACF SERVAUTH profile to protect EZB.RPCBIND.<system-name>.<rpc-bind-name>.REGISTRY and give appropriate users READ access. This control protects the registerrpc(), svc_register(), pmap_set(), and pmap_unset() services {CS.1::CS.1-R10-CS-1}. In a multilevel secure environment no application can regiister or deregister with rpcbind unless this profile exists and grants access {CS.1::CS.1-R10-CS-2}.

- When operating in a multilevel secure environment, the rpcbind target assistance functions will assume the SECLABEL of the requesting process before forwarding the request to the target server. This will ensure that the target server knows the proper SECLABEL for the data it receives {CS.1::CS.1-R10-CS-3}.

AT-TLS is configured through the PROFILE.TCPIP configuration file and the Policy Agent. This configuration may also specify a list of LDAP servers for certificate revocation information (see Section Network configuration and management).

Note: When hardware crypto has been activated, the cryptographic operations performed by IPSec {CS.1::CS.1-R8-IPSec-3} and System SSL {CS.1::CS.1-R8-SSL-1} will make use of the hardware crypto when appropriate, either through ICSF or the CPACF processor instructions. In the absence of hardware crypto support, IPSec {CS.1::CS.1-R9-IPSec-4} and System SSL {CS.1::CS.1-R9-SSL-2} will use software algorithms for cryptographic operations, although in the case of AES (CBC, GCM, and GMAC) encryption IPSec will still make use of ICSF {CS.1::CS.1-R12-IPSec-5}. For symmetric (TDES, AES) and hashing (SHA-1, SHA-2) functions, ICSF will invoke the CPACF if it supports the function or will provide the functions via software within ICSF {CS.1::CS.1-R12-ICSF-1}.

Packet filtering functionality that can control information flow into or out of the system based on security characteristics of the packets or of the network interface they use, as follows:

- {CS.1::CS.1-R12-PF-1} Filter rules can apply to a packet based on information within the packet or information external to the packet.

  - Internal information: source address, destination address, protocol, source port, or destination port, ICMP type and code, OSPF type, and mobility header type.

        - External information: the direction of packet flow routing attribute, security class (determined by the network interface used by the packet).

    - {CS.1::CS.1-R12-PF-2} When a filter rule applies to a packet, it can discard the packet silently, discard the packet with ICMP notification to the sender, permit the packet flow, or permit the packet flow and enforce IPSec processing.

    - {CS.1::CS.1-R11-PF-3} A z/OS TCP/IP stack configured for IP security implements a default “deny” policy in the absence of any configured filter rules.

In addition, the Communications Server provides the following application protocols that include user authentication using RACF:

- FTP (user authentication is optional) {CS.1::CS.1.6}
- telnet {CS.1::CS.1.7}
- rlogin, rsh, and rexec {CS.1::CS.1.8}
- TN3270 {CS.1::CS.1.9}
- Network Security Services Server {CS.1::CS.1-R10-NSS-6}
- Policy Agent Server {CS.1::CS.1-R10-CS-POLCEN-12}
- Load Balancing Advisor {CS.1::CS.1-R10-CS-LBA-3}

z/OS also provides an HTTP server that uses RACF for authentication, (though the administrator can also configure anonymous access if necessary) {CS.1::CS.1.V1R7.2}

Access control to resources used within a FTP, HTTP, or telnet session is also performed using RACF {CS.1::CS.1.10}.

Import of certificates and key pairs used for authentication and key exchange for the SSL/TLS and IPSec protocols is restricted to authorized administrators {CS.1::CS.1.11}.

The FTP and TN3270 Server applications can use AT-TLS services to provide end-to-end data channels that are authenticated and encrypted {CS.1::CS.1-R8-CS-1}. AT-TLS (Application Transparent Transport Layer Security) uses System SSL services to provide end-to-end data channels that are authenticated and encrypted for most TCP applications.

8.1.4.2 System SSL

z/OS provides SSL/TLS functions via the System SSL component for applications wishing to use SSL/TLS directly (without taking advantage of the AT-TLS functions of the Communications Server). The selectable algorithms can be limited by configuring a subset of allowable algorithms at the server {CS.2::CS.1-R8-SSL-2}. The SSL/TLS protocol can be used to set up a trusted channel to another system through a potentially insecure network. SSL/TLS protects the data against disclosure and attacks related to integrity like undetectable modifications or replay. Servers can support encryption using TDES with 168-bit key length {CS.2::CS.1-R8-SSL-3} or AES with either 128- or 256-bit key length {CS.2::CS.1-R8-SSL-4}.

8.1.4.3 Network Authentication Service

The z/OS Network Authentication Service provides communication security via the Kerberos and GSS-API protocols, which use one of the supported encryption protocols (TDES, AES-128, AES-256) to encrypt application messages when requested by applications that support Kerberos and GSS-API functions {CS.3::CS.1-R8-KERB-1}.

8.1.4.4 NFS Client and Server

The z/OS NFS client and server support the use of Kerberos (via the Network Authentication Service) to provide integrity and confidentiality for authentication credentials and data as they flow over the network. NFS server configuration parameters allow the administrator to configure use of Kerberos for network traffic and the z/OS NFS client and server support the following Kerberos V5 security mechanisms {CS.4::CS.1-R10-NFS-1}:

- krb5, which provides Kerberos V5 based integrity on the RPC credentials (but not data) using the DES_MAC_MD5 integrity algorithm and uses the RPCSEC_GSS service of rpc_gss_svc_none.

- krb5i, which provides Kerberos V5 based integrity on both the RPC credentials and data using the DES_MAC_MD5 integrity algorithm and uses the RPCSEC_GSS service of rpc_gss_svc_integrity.

- krb5p, which provides Kerberos V5 based integrity and privacy on both the RPC credentials and data using the DES_MAC_MD5 algorithm for integrity and 56 bit DES for privacy. It uses the RPCSEC_GSS service of rpc_gss_svc_privacy.

When acquiring Kerberos tickets the z/OS NFS client supports the following encryption options {CS.4::CS-1-R10-NFS-5}:

- ENCTYPE_DES_CBC_MD5

8.1.4.5 IBM Ported Tools for z/OS (OpenSSH)

Additionally, the IBM Ported Tools for z/OS provide OpenSSH functionality, with an sshd daemon that supports the SSHv2 protocol {CS.5::CS.1-R8-SSH-1} and these commands to allow remote users to perform work on the z/OS system:

- ssh, to establish a UNIX shell environment {CS.5::CS.1-R8-SSH-2}

- scp to perform remote file copying operations {CS.5::CS.1-R8-SSH-3}

- sftp to perform file transfer operations (similar to ftp) {CS.5::CS.1-R8-SSH-4}
- ssh-keygen to generate the host key files and the RSA or DSA key pairs {CS.5::CS.1-R8-SSH-7}

The SSH protocol can be used to set up a trusted channel to another system through a potentially insecure network. SSH protects the data against disclosure and attacks related to integrity like undetectable modifications or replay. SSH supports encryption using TDES with 168-bit key length {CS.5::CS.1-R8-SSH-5} and AES with 128-, 192-, or 256-bit key length {CS.5::CS.1-R8-SSH-6}, {CS.5::CS.1-R9-OpenSSH-1}.

8.1.5 Management

8.1.5.1 User and group management

Definition of users and groups

z/OS users and groups are defined in RACF.

LDAP LDBM users and groups are defined in the LDAP server, but the LDAP users must be mapped one-to-one to RACF z/OS users. See LDAP LDBM Users for info on defining LDAP users.

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. See Defining Kerberos Users for more discussion of this topic.

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.

Additionally, for administrative convenience, the z/OS LDAP server and RACF provide an administrative backend to LDAP known as SDBM. RACF administrators can authenticate to LDAP using a RACF identity and password or using a digital certificates over SSL/TLS (LDAP

SASL bind with EXTERNAL verification) mapped to a RACF USER ID, then make requests to the SDBM backend via LDAP programming protocols. These requests allow the administrator to view or update RACF USER, GROUP, CONNECT and general resource profile information, and SETROPTS class-related options. LDAP transforms those requests into a tagged format supported by the R_admin() callable service, and uses that service to pass them to RACF for processing, just as though they were entered via TSO. Because the LDAP mechanisms merely provide a transformation of the administrator's LDAP request into a different format (RACF command structure), and RACF performs the authentication, and all security checking and administrative actions occur within RACF just as for the TSO commands, we do not view this LDAP mechanism as relevant to security. Therefore we do not address it further in this document.

The TOE also provides an interface via Java classes and methods that allows Java programs to perform RACF user and group administration in a manner similar to that used for the LDAP SDBM backend processing. The Java program invokes the provided Java methods, which transform the provided data into RACF commands and issues them via R_admin(). RACF then processes the commands as though they were entered via TSO, using the identity of the user running the Java program {SM.1::SM.1-R9-JSEC-1}.

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.

Name	Description
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.

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:

- 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}.
- 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}.
- 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}.
- 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 (TDES), 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.

Defining Kerberos Users

z/OS recognizes two kinds of Kerberos users: local and foreign. To define a local Kerberos user, add a KERB segment to the USER profile. Specify the encryption type as DES3 (TDES), NODES, NODESD NOAES128 NOAES256 to ensure that TDES encryption processing is used for this user. Specify the encryption type as AES128 AES256 NODES3 NODES NODESD to ensure use of AES encryption for this user. Specify the user's Kerberos principal name. When the user next changes his/her password/phrase, the user's encryption keys will be generated from the new RACF password/phrase {SM.1::SM.1-R10-KERB-1}.

To allow a foreign Kerberos user to authenticate, define a trust relationship between the local Kerberos realm and the foreign realm, using either the peer or transitive trust methods, by defining REALM profiles with passwords in RACF as described in the Network Authentication Service Administration guide {SM.1::SM.1-R8-KERB-2}. Kerberos passwords up to 128 characters in length may be specified in the REALM profiles {SM.1::SM.1-R10-KERB-4}. Then, for each foreign principal you want to accept, define a KERBLINK profile in RACF specifying the name of the local user in the APPLDATA field, as described in the RACF Security Administrator's Guide {SM.1::SM.1-R8-KERB-3}.

LDAP LDBM Users

LDAP has the ability to authenticate to RACF through LDBM by supplying a RACF password/phrase on a simple bind to the LDBM backend or by a digital certificates over SSL/TLS (LDAP SASL bind with EXTERNAL verification) mapped to a RACF USER ID. Authorization information is still gathered by the LDAP server backend based on the DN that performed the bind operation. The LDAP administrator defines the authorized LDAP LDBM users by defining "subject distinguished names" DNs in the LDBM directory. Additionally, for the evaluated configuration, the administrator must define the DN as using what LDAP calls native authentication (i.e.RACF authentication) rather than LDAP authentication, and must provide the RACF user ID that represents this LDAP subject. During the bind operation, the client user will provide his/her subject DN and the RACF password for the RACF user ID that corresponds to that subject DN. The LDAP server will then use z/OS authentication functions to validate the specified password against the configured RACF user ID. (Note: Security claims appear earlier under Identification and Authentication functions.)

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

RACF provides the RACDCERT command which can be used to

- create certificate requests to send to a Certifying Authority {SM.1::SM.1-R8-RACF-RACDCERT-1}
- generate public/private key pairs and certificates (DIGTCERT class) {SM.1::SM.1-R8-RACF-RACDCERT-2}
- export a certificate or certificate packages to a data set, optionally with the private key {SM.1::SM.1-R8-RACF-RACDCERT-3}
- 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}.
- delete or list certificates in the RACF database {SM.1::SM.1-R8-RACF-RACDCERT-7}

- maintain (create, list, delete) key rings containing certificates (DIGTRING class) {SM.1::SM.1-R8-RACF-RACDCERT-8}
- add certificates to or delete them from key rings {SM.1::SM.1-R8-RACF-RACDCERT-9}
- 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}.
- create and manage the contents of PKCS#11 cryptographic tokens contained in the ICSF TKDS {SM.1::SM-1.R9-RACF-RACDCERT-13}

Note: {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.

z/OS also provides the PKI Services component which provides a full-function Certificate Authority and certificate life-cycle management process. Certificates that PKI Services issues are not (by default) placed in the RACF database, but may be put there manually by users or administrators. See PKI Services for additional details.

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.

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 Authority checking for RACDCERT Processing or the use of the associated programming interfaces as described above. It also occurs during SSL/TLS processing, Communication 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.

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)	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
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)	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

FUNCTION	READ	UPDATE	CONTROL
DELTOKEN (controlled only via CRYPTOZ class)	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)	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)	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 ring	Remove a site or certificate authority certificate from one's own key ring	Remove a certificate from another user's key ring
REKEY	Rekey one's own certificate	Rekey another user's certificate	Rekey a site or certificate authority certificate

FUNCTION	READ	UPDATE	CONTROL
ROLLOVER	Rollover one's own certificate	Rollover another user's certificate	Rollover a site or certificate authority certificate
UNBIND (controlled only via CRYPTOZ class)	n/a	n/a	n/a

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

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

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

These tables describe 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.
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

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	READ authority to	UPDATE authority to	CONTROL authority to

SIGNWITH option chosen	To generate one's own certificate	To generate another user's certificate	To generate a site or certificate authority certificate
a SITE or CERTAUTH certificate	IRR.DIGTCERT.ADD and CONTROL authority to IRR.DIGTCERT.GENCERT	IRR.DIGTCERT.ADD and CONTROL authority to IRR.DIGTCERT.GENCERT	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

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 {SM.7::AC.4-R9-RACF-30}. For more information on this case see Authority Checking for PKCS11 Cryptographic Tokens in the ICSF TKDS.

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, e.g. 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

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

Virtual keyring owner	Resource Name	Access	Action able to perform
			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

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

Access to IRR.DIGTCERT.LISTRING in the FACILITY class	**Action able to perform**
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

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

- READ authority to resource IRR.DIGTCERT.GENCERT in the FACILITY class if the caller owns the certificate {SM.7::AC.4-R9-RACF-10};

- 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, e.g. SERVER1.FTPRING1.UPD	**Action able to perform**
READ	• add a new ring for Server1 named FTPring1 • 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, e.g. SERVER1.FTPRING1.UPD	Action able to perform
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,

- Any usage = PERSONAL, CERTAUTH or SITE
- Any type cert = certificate is owned by any regular ID, or by the site or a certificate authority.

Using RDATALIB Profile Checking {SM.7::AC.4-R9-RACF-23}:

With READ 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 Input cert and private key	• add one's own cert • connect to Server1's ring named FTPring1 one's own cert with usage PERSONAL only	if cert owned by caller • connect to Server1's ring named FTPring1 with usage PERSONAL only, other usages cause error • change the NOTRUST status to TRUST if trust flag turns on if cert is not owned by caller, error		if cert owned by caller • re-connect to Server1's ring named FTPring1 with usage PERSONAL only, other usages cause error, with new specified default value • change the NOTRUST status to TRUST if trust flag turns on if cert is not owned by caller, error	
		if cert owned by caller • re-add cert with private key • connect to	if cert owned by caller • connect to Server1's ring named FTPring1 with usage PERSONAL	if cert owned by caller • re-add cert with private key	if cert owned by caller • re-connect to Server1's ring named FTPring1

	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
		Server1's ring named FTPring1 with usage PERSONAL only, other usages cause error • change the NOTRUST status to TRUST if trust flag turns on if cert is not owned by caller, error	only, other usages cause error • change the NOTRUST status to TRUST if trust flag turns on if cert is not owned by caller, error	• re-connect to Server1's ring named FTPring1 with usage PERSONAL only, other usages cause error, with new specified default value • change the NOTRUST status to TRUST if trust flag turns on if cert is not owned by caller, error	with usage PERSONAL only, other usages cause error, with new specified default value • change the NOTRUST status to TRUST if trust flag 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	• add any type cert • connect to Server1's ring named FTPring1 one's own cert with any usage or • connect other's or SITE/CA's cert with usage SITE or CERTAUTH only, PERSONAL usage causes error	• connect to Server1's ring named FTPring1 one's own cert with any usage or • connect to Server1's ring named FTPring1 other's or SITE/CA's cert with usage SITE or CERTAUTH only, PERSONAL usage causes error • change the NOTRUST status to TRUST if trust flag turns on		• re-connect to Server1's ring named FTPring1 one's own cert with any usage or • 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 • change the NOTRUST status to TRUST if trust flag turns on	
Input cert and private key	• add any type cert • connect to Server1's ring named FTPring1 any type cert with any usage	• re-add any type cert with private key under original ID • connect to Server1's ring named FTPring1 any type cert with any usage	• connect to Server1's ring named FTPring1 any type cert with any usage • change the NOTRUST status to TRUST if trust flag turns on	• re-add any type cert with private key under original ID • re-connect to Server1's ring named FTPring1 any type cert with any usage,	• re-connect to Server1's ring named FTPring1 any type cert with any usage, with new specified default value • change the NOTRUST

	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
		• change the NOTRUST status to TRUST if trust flag turns on		with new specified default value • change the NOTRUST status to TRUST if trust flag turns on	status to TRUST if trust 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 Input cert and private key	• add any type cert • connect to Server1's ring named FTPring1 any type cert with any usage	• connect to Server1's ring named FTPring1 any type cert with any usage • change the NOTRUST status to TRUST if trust flag turns on		• re-connect to Server1's ring named FTPring1 any type cert with any usage, with new specified default value • change the NOTRUST status to TRUST if trust flag turns on	
		• re-add cert with private key under original ID	• connect to Server1's ring named FTPring1 any type	• re-add cert with private key under	• re-connect to Server1's ring named

	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**
		• connect to Server1's ring named FTPring1 any type cert with any usage • change the NOTRUST status to TRUST if trust flag turns on	cert with any usage • change the NOTRUST status to TRUST if trust flag turns on	original ID • re-connect to Server1's ring named FTPring1 any type cert with any usage, with new specified default value • change the NOTRUST status to TRUST if trust flag turns on	FTPring1 any type cert with any usage, with new specified default value • 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	• add one's own cert • connect one's own cert with usage PERSONAL to one's own ring
CONTROL	READ	• add one's own cert • connect one's own cert with

		any usage to one's own ring
UPDATE	UPDATE	• add one's own or other's cert • connect one's own or other's cert with usage PERSONAL to one's ring or • connect SITE/CA's cert with SITE/CERTAUTH usage to one's own ring
CONTROL	UPDATE	• add any type cert • connect one's own or other's cert with usage PERSONAL to one's ring or • connect any type cert with usage SITE/CERTAUTH to one's ring
UPDATE	CONTROL	• add one's own or other's cert • connect any type cert with usage PERSONAL to any ring or • connect SITE/CA's cert with any usage to any ring
CONTROL	CONTROL	• add any type cert • 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	• re-add one's own cert with private key • change the NOTRUST status of the connected cert to TRUST if trust flag turns on • connect one's own cert with usage PERSONAL

		to one's own ring
CONTROL	READ	• re-add one's own cert with private key • change the NOTRUST status of the connected cert to TRUST if trust flag turns on • connect one's own cert with any usage to one's own ring
UPDATE	UPDATE	• re-add one's own or other's cert with private key • change the NOTRUST status of the connected cert to TRUST if trust flag turns on • connect one's own or other's cert with usage PERSONAL to one's ring or • connect SITE/CA's cert with SITE/CERTAUTH usage to one's own ring
CONTROL	UPDATE	• re-add any type cert with private key • change the NOTRUST status of the connected cert to TRUST/HIGHTRUST if trust flag turns on • connect one's own or other's cert with usage PERSONAL to one's ring or • connect any type cert with usage SITE/CERTAUTH to one's ring

UPDATE	CONTROL	• re-add one's own or other's cert with private key • change the NOTRUST status of the connected cert to TRUST if trust flag turns on • connect any type cert with usage PERSONAL to any ring or • connect SITE/CA's cert with any usage to any ring
CONTROL	CONTROL	• re-add any type cert with private key • change the NOTRUST status of the connected cert to TRUST/HIGHTRUST if trust flag turns on • 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 NOTRUST to TRUST/HIGHTRUST is requested)**	**Action able to perform**
n/a	READ	READ	• connect one's own cert with usage PERSONAL to one's own ring • change the NOTRUST status of the connected cert to TRUST if trust flag turns on

CONTROL	READ	READ	• connect one’s own cert with any usage to one’s own ring • 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	• connect one’s own or other’s cert with usage PERSONAL to one’s ring or • connect SITE/CA’s cert with SITE/CERTAUTH usage to one’s own ring • 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	• connect one’s own or other’s cert with usage PERSONAL to one’s own ring or • connect SITE/CA’s cert with SITE/CERTAUTH usage to one’s own ring • 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	• connect any type cert with usage PERSONAL to any ring or • connect SITE/CA's cert with any usage to any ring • 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	• connect any type cert with any usage to any ring • 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}.

Authority Checking for PKCS11 Cryptographic Tokens in the ICSF TKDS

DAC for PKCS#11 Cryptographic Tokens in the ICSF TKDS occurs using profiles in the CRYPTOZ resource class, using the basic MVS DAC algorithm described above.

The access control defined in the PKCS#11 standard was designed for systems that have no security manager. Access to token information in the standard is granted based on the knowledge of a PIN. In the definition there are two types of users, the standard user (User) and the security officer (SO). Each has their own PIN. The SO can initialize a token (zero the contents) and set the User's PIN. The SO can also access the public objects on the token but not the private ones. The User has access to the private objects on a token and has the power to change his or her own PIN. The User cannot reinitialize the token. The role one is allowed to take depends on the PIN entered. Thus a single person can fill both roles by having knowledge of both PINs.

On z/OS these two roles will be simulated by using SAF profiles in a new Class called CRYPTOZ. There will be no PINs. Each token defined will have a unique token name (label) up to 32 characters in length. The permitted characters are alphanumeric, national (@,#,$) or period (.). The first character must be alphabetic or national. Lowercase letters are permitted but will be folded to uppercase. (This is the same naming restriction as PKDS labels.) There will be two CRYPTOZ Class resources checks performed for tokens:

- USER.token-name -- Controls the User role
- SO.token-name -- Controls the SO role

The different access levels provide the following functionality:

- The 3 standard PKCS#11 access types (User R/W, SO R/W, User R/O)
  - R/O vs R/W not end-user controlled
- Plus 3 z/OS unique access types
  - Weak SO -- An SO that can modify CAs contained in a token but not initialize the token
  - Strong SO -- An SO that can add or remove private objects in a token (e.g., a server administrator)
  - Weak User -- A user that cannot change the trusted CAs contained in a token

CRYPTOZ DAC Table {SM.7::AC.4-R9-ICSF-1}:

CRYPTOZ Resource Name	**Access of: READ**	**Access of: UPDATE**	**Access of: CONTROL**
SO.token-label	Weak SO -- read / create / delete / modify / use public objects	SO R/W -- Weak SO plus create / delete token	Strong SO -- SO RW plus read (but not use) private objects, create / delete / modify private objects
USER.token-label	User R/O -- read / use public and private objects.	Weak User -- User R/O plus create / delete / modify private and public objects (cannot add / delete / modify certificate authority objects)	User R/W -- Weak User plus add / delete / modify certificate authority objects

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:

- 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}.
- 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}.
- 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}.
- 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}.

LDAP LDBM Groups

LDBM supports group definitions. These group definitions allow for a collection of names to be easily associated for access control checking. LDBM supports static (where the members are defined individually {SM.2::SM.2-R8-LDAP-1}), dynamic (where membership is determined using one or more LDAP search expressions {SM.2::SM.2-R8-LDAP-2}), and nested (a group that references other group entries that can be static, dynamic or nested groups {SM.2::SM.2-R8-LDAP-3}) group entries.

When configured to search SDBM for a user's groups, LDAP will convert those groups into SDBM DNs and add them to the LDBM user's set of groups, making them available for use in LDBM ACLs for authorization checking {SM.2::SM.2-R10-LDAP-4}.

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.

For LDAP LDBM users, the LDAP server maintains the roles and attributes specified below (in LDAP Roles and LDAP Attributes) in the LDAP LDBM database.

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}.

Operator

A user who is allowed to issue operator commands has the role of an operator. To be able to issue operator commands a user must have been authorized to the profiles in the OPERCMDS class protecting the operator commands. Permission to issue operator commands can be given on a per command basis. For the purpose of this Security Target a user who has been authorized to at least one profile in the OPERCMDS class protecting MVS and JES2 operator commands is defined to have the role of an operator.

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.

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. The CLAUTH attribute can be assigned at the user or group level {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}.

LDAP Roles

The TOE supports the LDAP roles:

1. administrator {SM.1::SM.1-R8-LDAP-1},
2. for basic replication :masterServer {SM.1::SM.1-R8-LDAP-2} (used as the master in LDAP replication processing), and peerServer {SM.1::SM.1-R8-LDAP-3} (used as a peer in LDAP replication processing),
3. for advanced replication: supplier (a server thast sends changes to another (consumer) server , consumer (a server that receives changes via replication from another (supplier) server; {SM.1::SM.1-R12-LDAP-9}
4. and (by default) “end user”.{SM.1::SM.1-R12-LDAP-10}

Three non-default roles (administrator, and for basic replication, masterServer and peerServer) are defined within the LDAP configuration file.

For advanced replication, a replication context is created as an entry in the directory with the auxiliary objectclass ibm-replicationContext that identifies the root of a replicated subtree. It can be added to any entry. The replication configuration information is maintained in a set of entries created below the base of a replication context. If there is more than one replication context present in the same subtree, the replication configuration information under the child replication context entry is used while the replication configuration under the parent entry is ignored. If the ibm-replicationContext auxiliary objectclass is added to a non-suffix level entry in the directory, explicit aclEntry and entryOwner attribue values are required. For advanced replication, supplier credentials are created in the directory associated with the replica subtree that identify the servers that are supplied (replicated to) by each server, and on a consumer server a credentials entry is required for each supplier and verified using a simple or SASL EXTERNAL bind. {SM.1::SM.1-R12-LDAP-11}

End users have no pre-defined administrative rights, though under the control of access lists in the LDAP directory they may be allowed to create,or delete objects, or even manipulate the access lists for objects. The Directory Administrator has the ability to define LDAP groups to assist in the management of access rights and privileges {SM.1::SM.1-R8-LDAP-4}. Those administrator defined groups are not considered to be roles in the sense of the CC requirement FMT_SMR.1 but are just ways to manage access rights more easily.

The administrator also has complete access rights to all data in the LDAP LDBM database.

When configuring LDAP LDBM basic replication, replicas may be read-write, or read-only {SM.1::SM.1-R8-LDAP-5}. A peerServer can replicate its changes to other read-write replicas, and has the ability to update all data, bypassing all access list (ACL) controls {SM.1::SM.1-R8-LDAP-6}. A masterServer can replicate its changes to other read-write or read-only replicas {SM.1::SM.1-R8-LDAP-7}. A particular server may be both a peerServer to other read-write replicas, and a masterServer to read-only replicas {SM.1::SM.1-R8-LDAP-8}.

For advanced replication, a read-only replica is a consumer replica. If the replica can both read and write, the replica is both a consumer and supplier replica. {SM.1::SM.1-R12-LDAP-12}

LDAP Attributes

The ibm-nativeId LDAP attribute specifies the RACF user ID associated with an LDAP user authenticating to LDAP to access LDAP LDBM data.

Several attributes and object classes determine group membership for LDAP groups:

- For static groups in the accessGroup, groupOfNames, ibm-staticGroup object classes, the values of the member attribute determine group membership.
- For static groups in the groupOfUniquenames object class the values of the uniqueMember attribute determine group membership.
- For dynamic groups the scope and search filters contained in the values of the memberURL attribute determine group membership.
- For nested groups the values of the ibm-memberGroup attribute determine the groups that are members of the nested group.

User Revocation

User revocation can take two forms in the TOE:

- 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.
- 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 certificates supplied by PKI Services, the administrator can publish the certificate on the Certification Revocation List (CRL) which will signal to applications that support CRLs or the Online Certificate Status Protocol that the certificate is no longer valid and may not be used for authentication {SM.1::SM.1-R8-REV.3}.

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.

8.1.5.2 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.

For information on z/OS UNIX objects see z/OS UNIX file system resources.

Additionally, the LDAP server makes access decisions based on information stored in the LDBM database. For information on LDBM resources see LDAP LDBM Users.

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

Name	**Description**
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

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

Name	**Description**
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)

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 dataset containing the zFS file system must be created with the SYSMULTI security label. After creation of the file system, the security label of the dataset 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.

LDAP LDBM resources

The LDAP administrator can configure some LDAP resources as requiring user authentication prior to access, and others (representing public data which anyone should be able to access) as not requiring authentication.

Additionally, the LDAP server maintains the following attributes for LDBM data objects, using them in making access decisions. The TOE controls access to all directory entry objects based on the following security attributes:

- Entry Owner Information:
    - entryOwner: defines entry owner.
    - ownerPropagate: indicates whether to propagate the ownership of the entry to all descendant entries, until another entry with ownerPropagate is found.
- Access Control Attributes(ACA):
    - aclEntry: defines the access control information.
    - aclPropagate: indicates whether to propagate access control information of the entry to all descendant entries, until another entry with aclPropagate is found.

RACF General Resource classes

For the evaluation the protection of the following classes are considered:

CFIELD

Allows definition of fields in the CSDATA segment of USER and GROUP profiles {SM.1::SM.2-R10-RACF-1}

CONSOLE

Controlling access to operator consoles. Also, conditional access to other resources for commands originating from an operator console. {SM.2::SM.2.1}

CRYPTOZ

Controls access to PKCS#11 cryptographic tokens in the ICSF TKDS. {SM.2::SM.2-R9-CRYPTOZ}

DASDVOL

DASD volumes. See also the GDASDVOL class. {SM.2::SM.2.2}

DEVICES

Used to control access to unit record devices, teleprocessing or communication devices, and graphic devices. {SM.2::SM.2.3}

DIGTCERT

Used to register x.509v3 digital certificates in the RACF database.

DIGTCRIT

Used to define additional mapping criteria for the interpretation of x.509v3 digital certificates presented by clients when the certificates are not specifically registered in the RACF database, and to assign a RACF user ID to the client’s session as part of the client authentication process.

DIGTNMAP

Used to define the primary mapping rules for the interpretation of x.509v3 digital certificates presented by clients when the certificates are not specifically registered in the RACF database, and to assign a RACF user ID to the client’s session as part of the client authentication process.

DIGTRING

Implements key rings for servers or users in the RACF database, holding information about allowable Certificate Authority (CA) certificates and private keys for locally defined personal certificates and local signing certificates.

DIRAUTH (used in Labeled Security Mode only)

This class ensures that security label authorization checking is done when a user receives a message sent through the TPUT macro or the TSO SEND, or LISTBC commands. Profiles are not allowed in this class. {SM.2::SM.2.4}

FACILITY

This class is used by various components of the TOE to manage specific privileges that could be assigned to users such that they do not need the SPECIAL attribute or the z/OS UNIX superuser privilege. Only a few profiles in this class are relevant for the claims in this Security Target. Access to the relevant profiles in this class is covered by individual claims for those profiles when appropriate..

GDASDVOL

Grouping class for DASDVOL {SM.2::SM.2-R8-RACF-GDASDVOL}

GLOBAL

Global access checking table entry. Provides the ability for fast access check for user that don’t have the RESTRICTED attribute. Can be used for defined resource classes only. Must be used to allow READ access to resources classified as SYSLOW only. {SM.2::SM.2.5}

GTERMINL

Resource group class for TERMINAL class. {SM.2::SM.2.6}

GXFACILI

Grouping class for XFACILIT {SM.2::SM.2-R8-RACF-GXFACILI}

IDIDMAP

Class to provide mappinginformation from a distributed identity to a local RACF user ID {SM.2::SM.2-R12-RACF-IDIDMAP}

JESINPUT

Port of entry class to control which JES2 input devices a user can use to submit batch work to the system. {SM.2::SM.2.7}

JESJOBS

Controlling the submission and cancellation of jobs by job name. {SM.2::SM.2.8}

JESSPOOL

Controlling access to job data sets on the JES spool (that is, SYSIN and SYSOUT data sets). {SM.2::SM.2.9}

KERBLINK

Used to map user identities of local and foreign user IDs {SM.2::SM.2-R8-KERBLINK}

LOGSTRM

Used to control access to system logger resources, such as log streams and the coupling facility structures associated with them {SM.2::SM.2-R9-LOGGER-LOGSTRM}

NODES

Controls the following on MVS systems:

- Whether jobs are allowed to enter the system from other JES2 nodes {SM.2::SM.2.10}

- Whether jobs that enter the system from other nodes have to pass user identification and password verification checks associated with JES/NJE {SM.2::SM.2.11}

OPERCMDS

Controls who can issue operator commands (for example, JES and MVS, and operator commands). {SM.2::SM.2.12}

PROGRAM

Controlled programs (load modules). {SM.2::SM.2.13}

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. {SM.2::SM.2.14}

PTKTDATA

Used to configure PassTicket processing {SM.2::SM.2-R8-PTKTDATA}

RDATALIB

Used to peform authorization checking for the R_datalib callable service {SM.2::SM.2-R9-RDATALIB}

REALM

Used to define local and foreign Kerberos realms {SM.2::SM.2-R8-REALM}

SDSF

Controls the use of authorized commands in the System Display and Search Facility (SDSF). {SM.2::SM.2.15}

SECDATA (used in Labeled Security Mode only)

Security classification of users and data (security levels and security categories). {SM.2::SM.2.16}

SECLABEL (used in Labeled Security Mode only)

If security labels are used, and, if so, their definitions. {SM.2::SM.2.17}

SERVAUTH

Contains profiles that are used by servers to check a client's authorization to use the server or to use resources managed by the server. {SM.2::SM.2.18}

SERVER

Controlling the server's ability to register with the daemon. {SM.2::SM.2.19}

SMESSAGE

Controlling to which users a user can send messages (TSO only). {SM.2::SM.2.20}

STARTED

Used in preference to the started procedures table to assign an identity during the processing of an MVS START command. Part of the Identification of STCs. {SM.2::SM.2.21}

TAPEVOL

Tape volumes. {SM.2::SM.2.22}

TERMINAL

Terminals (TSO). {SM.2::SM.2.23}

TSOPROC

TSO logon procedures. {SM.2::SM.2.24}

UNIXPRIV

Contains profiles that are used to grant z/OS UNIX privileges. {SM.2::SM.2.25}

VTAMAPPL

Controlling who can open ACBs from non-APF authorized programs. This prevents programs from counterfeiting login screens. {SM.2::SM.2.26}

WRITER

Controlling the use of JES writers. {SM.2::SM.2.27}

XFACILIT

Analogous to the FACILITY class, but supporting longer resource and profile names (246 characters vs 39 for FACILITY) {SM.2::SM.2-R8-XFACILIT}

8.1.5.3 RACF configuration and management

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.

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. 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 x.509v3 digital certificates, certificate mapping filters, certificate mapping criteria, and key rings in the RACF database.

- SETROPTS

Command to set specific RACF options (see section above for details) {SM.3::SM.3.34}

- RACMAP

Command to establish mappings between distributed user identities and local RACF user IDs {SM.3::SM.3-R12-RACF-2}.

Other RACF commands not related to the Security Policy as defined in this Security Target exist, but are not mentioned here.

Administrators can also use the LDAP SDBM backend {SM.3::SM.3-R9-LDAP-1} or the Java JSEC interfaces {SM.3::SM.3-R9-JSEC-1} to issue the RACF commands ADDUSER, ALTUSER, DELUSER, LISTUSER, ADDGROUP, ALTGROUP, DELGROUP, LISTGRP, CONNECT, and REMOVE. Additionally, administrators can use the LDAP SDBM backend to issue RDEFINE, RDELETE, RALTER, RLIST, and SETROPTS commands for class-related options {SM.3::SM.3-R11-LDAP-2}.

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 in the z/OS UNIX System Services environment {SM.3::SM.3.35}. The standard functions to set or modify permission bits to file system objects and IPC objects also exist in the z/OS UNIX environment and allow users with the required permission to perform those actions {SM.3::SM.3.36}. In addition functions exist that allow the owner of a file system object to set or modify the access control list entries of this file system object {SM.3::SM.3.37}.

8.1.5.4 Network configuration and management

z/OS provides some basic configuration data sets for TCP/IP and TCP/IP based protocols. Those configuration data sets that are also related to security are:

- PROFILE.TCPIP

Provides TCP/IP initialization parameters and specifications for network interfaces and routing.

- TCPIP.DATA

Provides parameters for TCP/IP based client and server programs.

- Additional Communication Server configuration information (e.g., IPSec and AT-TLS) exists in policy files accessed via the Communication Server Policy Agent.

- The IKE daemon, NSS server, Defense Manager daemon, and Policy Agent also have their own configuration files.

- The HTTP server configuration file (default: httpd.conf)

Configuration statements in those data sets define the properties (including security properties) of the TCP/IP protocol itself as well as the main protocol server.

Communication Server ipsec Command Interface

The Communication Server provides a command named ipsec that allows authorized users to query information about IP filters, defensive filters and IPSec security associations. It also allows authorized users to activate or deactivate IPSec functions, affect which IP filters are loaded, and create, update and delete defensive filters. The administrator can control access to this command by granting READ access to the following resources in the SERVAUTH class:

- EZB.IPSECCMD.sysname.tcpname.DISPLAY allows clients to display information about IP filters, per-stack defensive filters and IPSec security associations {SM.4::SM-R10-CS-IPSECCMD-1}.

- EZB.IPSECCMD.sysname.tcpname.CONTROL allows clients to reload or refresh IP filters, create, update or delete per-stack defensive filters and activate or deactivate IPSec security associations {SM.4::SM-R10-CS-IPSECCMD-3}.

- EZB.IPSECCMD.sysname.DMD_GLOBAL.DISPLAY allows clients to display information about global defensive filters {SM.4::SM-R10-CS-IPSECCMD-4}.

- EZB.IPSECCMD.sysname.DMD_GLOBAL.CONTROL allows clients to create, update or delete global defensive filters {SM.4::SM-R10-CS-IPSECCMD-6}.

Communication Server Network Management Interface

The Communication Server provides, via the IKE daemon, a network management interface (NMI) that allows local applications to query information about IP filters and IPSec security associations. It also allows applications to activate or deactivate IPSec functions. The IKE daemon provides this information via a UNIX (not TCP/IP) socket. The administrator can control access to this interface by granting READ access to the following resources in the SERVAUTH class:

- EZB.NETMGMT.*sysname.tcpname*.IPSEC.DISPLAY allows clients to display information about IPSec filtering and security associations. If not defined, applications must run with UID(0) or access to BPX.SUPERUSER in order to use the interface {SM.4::SM-R10-CS-SECMON-1}.

- EZB.NETMGMT.*sysname.tcpname*.IPSEC.CONTROL allows clients to issue management requests to activate, deactivate, or modify IPSec security associations. If not defined, applications must run with UID(0) or access to BPX.SUPERUSER in order to use the interface. {SM.4::SM-R10-CS-SECMON-3}.

- EZB.NETMGMT.*sysname.sysname*.IKED.DISPLAY allows clients to display information about IKE daemon usage of the Network Security Services (NSS) client functions via the NMI or the ipsec command with the -w option. If not defined, applications must run with UID(0) or access to BPX.SUPERUSER in order to use the interface. {SM.4::SM-R10-CS-SECMON-4}.

Additionally, the Network Security Services (NSS) server provides a network management interface that allows a central administrator to monitor and control NSS and IPSec information in a manner similar to that provided by the IKE daemon. For these network management requests, the administrator can use the following SERVAUTH resources to provide protection:

- EZB.NSS.*sysname.clientname*.IPSEC.NETMGMT allows clients to register with the NSS server for IPSec network management services SM.4::SM-R9-CS-NSS-1}.

- EZB.NETMGMT.*sysname.clientname*.IPSEC.DISPLAY allows clients to display IPSec-related information via the NSS NMI or the ipsec command with the –z option {SM.4::SM-R9-CS-NSS-2}.

- EZB.NETMGMT.*sysname.clientname*.IPSEC.CONTROL allows clients to issue management requests to activate, deactivate, or modify IPSec security associations via the NSS NMI or the ipsec command with the –z option {SM.4::SM-R9-CS-NSS-3}.

- EZB.NETMGMT.*sysname.sysname*.NSS.DISPLAY allows clients to display information about current NSS client connections to the NSS server via the NSS NMI or the ipsec command with the –x option {SM.4::SM-R9-CS-NSS-4}.

Communication Server Policy Agent

The Communication Server provides a Policy Agent that can act in any of several roles, depending on configuration options:

- The Policy Agent may act as the Policy Definition Point (PDP) on a single system, installing policies in one or more z/OS Communications Server stacks {SM.4::SM-R9-CS-POLCEN-1}.

- The Policy Agent may act as a centralized *policy server*, providing PDP services for one or more remote policy clients {SM.4::SM-R9-CS-POLCEN-2}.

- The Policy Agent may act as a *policy client*, retrieving remote policies from the policy server. Each stack in a Common INET (CINET) environment acts as a separate policy client {SM.4::SM-R9-CS-POLCEN-3}. Communications between the policy client and the policy server may optionally be secured by AT-TLS {SM.4::SM-R10-CS-POLCEN-11}.

A single Policy Agent may act as a policy client or a policy server, but not both {SM.4::SM-R9-CS-POLCEN-11}.

Policies may be defined in several different ways. When acting as the PDP for a single system, Policy Agent can read policy definitions from local configuration files, a central repository that uses the Lightweight Directory Access Protocol (LDAP), or both {SM.4::SM-R9-CS-POLCEN-4}.

The Policy Agent also installs policies in one or more z/OS Communications Server stacks. It can be used to replace existing policies or update them as necessary {SM.4::SM-R9-CS-POLCEN-5}.

When acting as a policy server, Policy Agent also acts as a PDP for the local system, and so can read policies from local configuration files or an LDAP server, and install them in local stacks {SM.4::SM-R9-CS-POLCEN-6}. But it also reads policies from local configuration files on behalf of policy clients. These policies are retrieved by policy clients, but are not installed in the local stacks on the policy server {SM.4::SM-R9-CS-POLCEN-7}.

When acting as a policy client, Policy Agent retrieves remote policies from the policy server, and can also use local policies from configuration files or an LDAP server {SM.4::SM-R9-CS-POLCEN-8}.

The choice of local or remote policies can be made separately for each type of supported policy: Quality of Service (Qos), Intrusion Detection (IDS), Policy-Based Routing (PBR), IPSec, or AT-TLS {SM.4::SM-R9-CS-POLCEN-9}. For a given policy type, all policies are obtained either locally or remotely {SM.4::SM-R9-CS-POLCEN-10}.

When acting as a policy server, the policy agent will:

- First, authenticate its clients using a RACF user ID and password or PassTicket {SM.4::IA-R9-CS-POLCEN-1}.

- Then, authorize retrieval of policy data, requiring READ access to policy agent resources in the SERVAUTH class. These resources must be protected or retrieval will fail {SM.4::AC-R9-CS-POLCEN-1}. They have the form EZB.PAGENT.*sysname.image.ptype* {SM.4::AC-R9-CS-POLCEN-2} where

    - *Sysname* is the system name defined in the sysplex

    - *Image* is the TCP name or policy client name

    - *Ptype* is either QOS, IDS, IPSEC, or (for AT-TLS) TTLS.

8.1.5.5 PKI Services

PKI Services allows an installation to establish a Public Key Infrastructure (PKI) and serve as a certificate authority for its internal and external users, issuing and administering digital certificates in accordance with the organization's policies. Users can use a PKI Services application to request and obtain certificates through their own Web browsers {SM.5::SM-R8-PKI-1}, while authorized PKI administrators approve, modify, or reject these requests through their own Web browsers, Microsoft Internet Explorer version 5.x or higher {SM.5::SM-R8-PKI-2} or Netscape Communicator version 4.x or higher {SM.5::SM-R8-PKI-3}. The Web applications provided with PKI Services are highly customizable. An installation can allow automatic approval for certificate requests from certain users {SM.5::SM-R8-PKI-4} and, to provide additional authentication, add host IDs, such as RACF user IDs, to certificates issued for certain users {SM.5::SM-R8-PKI-5}. Installations can also issue certificates for browsers, servers, and other purposes, such as virtual private network (VPN) devices, smart cards, and secure e-mail.

PKI Services CA’s signing key length can be up to 4096 bits for RSA, up to 1024 bit for DSA, 521 bits for NIST ECC, or 512 bits for Brainpool ECC {SM.5::SM-R12-PKI-6}.

PKI Services can generate RSA keys up to 4096 bits for the certificates if requested {SM.5::SM-R12-PKI-37}.

PKI Services can generate NIST ECC keys with maximum length 521 bits for the certificates if requested {SM.5::SM-R12-PKI-41}.

PKI Services can generate Brainpool ECC keys with maximum length 512 bits for the certificates if requested {SM.5::SM-R12-PKI-42}.

PKI Services may be configured to accept and process certificate requests and revocation requests through the Certificate Management Protocol (CMP). Certificate requests may contain the public key of a public/private key pair, or may be omitted to instruct the PKI Service CMP CGI program to generate the key pair. {SM.5::SM-R12-PKI-43}

Supported Certificate Fields and Extensions

PKI Services certificates support fields and extensions defined in the X.509 version 3 (X.509v3) standard. It can include the following types of extensions:

Standard extensions {SM.5::SM-R8-PKI-7}

The standard X.509v3 certificate extensions:

- authority information access
- authority key identifier
- basic constraints
- certificate policies
- certificate revocation list (CRL) distribution points
  - Distinguish Name format
  - Uniform Resource Identifier format using LDAP or HTTP protocol
- key usage
  - digitalSignature
  - nonRepudiation
  - keyEncipherment
  - dataEncipherment
  - keyAgreement
  - keyCertSign
  - CRLSign
- extended key usage
  - serverauth
  - clientauth
  - codesigning
  - emailprotection

  - timestamping
  - ocspsigning
  - mssmartcardlogon
- subject alternate name
  - email
  - domain
  - IPAddress
  - uniformResourcesIdentifier
  - OtherName
- subject key identifier

Other extensions

- host identity mapping {SM.5::SM-R8-PKI-8}. This extension associates the subject of a certificate with a corresponding identity on a host system, such as with a RACF user ID.
- Custom extensions {SM.5::SM-R12-PKI-44}. This allows users to create any extensions conformed to the Extension structure: a sequence of OID, critical flag and value.

Supported Certificate Revocation List Fields and Extensions

PKI Services generates CRLs that comply with the X.509 version 3 (X.509v3) standard. The following extensions are included:

CRL extensions: {SM.5::SM-R8-PKI-9}

- AuthorityKeyIdentifier
- CRLNumber
- IssuingDistributionPoint

CRL entry extensions: {SM.5::SM-R8-PKI-10}

- CertificateIssuer
- CRLReason
  - Unspecified

  - keyCompromise
  - cACompromise
  - affiliationChanged
  - superseded
  - cessationOfOperation
  - certificateHold
- InvalidityDate

Certificate Templates

PKI Services will only generate certificates that are consistent with the currently defined Certificate templates. PKI Services shipped with sample certificate templates of the most commonly requested certificate types. You can add, modify, and remove certificate templates to customize the variety of certificate types you offer to your users.

PKI Services templates support generating certificates for the following uses or with the following characteristics:

- SSL Client authentication {SM.5::SM-R8-PKI-11}.
  - key usage: digitalSignature and keyEncipherment
  - extended key usage: clientauth
- SSL Server authentication using SSL {SM.5::SM-R8-PKI-12}.
  - Key usage: digitalSignature and keyEncipherment
  - Extended key usage: serverauth
- IPSEC Firewall server {SM.5::SM-R8-PKI-13}.
  - Key usage: digitalSignature, keyEncipherment and dataEncipherment
- Certificate Authority {SM.5::SM-R8-PKI-14}.
  - Key usage: keyCertSign and CRLSign
- z/OS authentication {SM.5::SM-R8-PKI-15}.
  - Key usage: digitalSignature and keyEncipherment

  - Extended key usage: clientauth
  - Host Identity Mapping
- S/MIME email protection {SM.5::SM-R8-PKI-16}.
  - Key usage: digitalSignature and keyEncipherment
  - Subject alternate name: email
- Code signing {SM.5::SM-R8-PKI-17}.
  - Key usage: digitalSignature and docSign
  - Extended key usage: codeSigning
  - Subject alternate name: email
  - Authority Information Access: basic
- Windows logon {SM.5::SM-R8-PKI-18}.
  - Key usage: digitalSignature
  - Extended key usage: clientauth, mssmartcardlogon
- Network device using the Simple Certificate Enrollment Protocol (SCEP) {SM.5::SM-R8-PKI-19}.
- Certificate with key pair generated by PKI Services {SM.5::SM-R11-PKI-38}.

Distribution of certificates

Other than sending the certificate back to the requestor through the browser, PKI Services can also post the issued certificates to LDAP according to the LDAP standard for communications with the Directory {SM.5::SM-R8-PKI-20}.

If the key pair for the certificate was generated by PKI Services, an email with a link embedded with the transaction ID will be sent to the requestor for him to pick up the certificate packaged with the private key {SM.5::SM-R11-PKI-39}.

Providing Certificate status

PKI Services provides certificate status information through Certificate Revocation Lists (CRLs) whose format complies with the X.509 standard and, the Online Certificate Status Protocol (OCSP) standard as defined by RFC 2560 for a “basic” OCSP responder {SM.5::SM-R8-PKI-21}.

The CRLs can be posted to LDAP according to the LDAP standard for communications with the Directory {SM.5::SM-R8-PKI-22}, or posted to an HFS file {SM.5::SM-R8-PKI-23}.

End User Functions

The end user can use the end user web pages to perform the following tasks:

- Install a CA certificate into the browser {SM.5::SM-R8-PKI-24}
- Request a new certificate {SM.5::SM-R8-PKI-25}
- Pick up a previously requested certificate {SM.5::SM-R8-PKI-26}
- Renew or revoke a previously issued browser certificate {SM.5::SM-R8-PKI-27}
- Recover a previously requested certificate and its private key , which requires specification of the email address and passphrase of the original request{SM.5::SM-R11-PKI-40}

Administrator Functions

The administrator can use the administration web pages to perform the following tasks:

- Process a certificate request
  - Approve a request without making changes {SM.5::SM-R8-PKI-28}
  - Approve a request with changes {SM.5::SM-R8-PKI-29}
  - Reject a request {SM.5::SM-R8-PKI-30}
  - Delete a request {SM.5::SM-R8-PKI-31}
- Process a certificate
  - Revoke a certificate {SM.5::SM-R8-PKI-32}
  - Suspend a certificate {SM.5::SM-R8-PKI-33}
  - Resume a certificate {SM.5::SM-R8-PKI-34}
  - Delete a certificate {SM.5::SM-R8-PKI-35}
- Perform searches for certificate requests and certificates {SM.5::SM-R8-PKI-36}

Security Administration for PKI Services

PKI Services security administration comprises the following tasks:

- Authorizing users for the PKI Services administration group (connecting and deleting members)

- Authorizing users for inquiry access

8.1.5.6 Security Management for System Logger Log Streams

Applications can read and write to defined log streams as explained in the DAC section of this document. However, before they can do this an administrator or an application must define the log stream and the policies that apply to it.

The system policy for log streams exists in an MVS data set known as the “LOGR couple data set”. Administrators who need to define or view the logger policy information use the IXCMIAPU utility program to do so. They require:

- READ authority to the MVSADMIN.LOGR resource in the FACILITY class in order to generate reports about the logger policy {SM.6::SM-R9-LOGGER-1}.

Additionally, logger administrators who need to define, in the CFRM policy, coupling facility structures that will be utilized by log streams will also need UPDATE authority to the MVSADMIN.XCF.CFRM resource in the FACILITY class {SM.6::SM-R9-LOGGER-3}.

Additionally, logger administrators who need to define, delete, or modify the definitions of log streams will need:

- ALTER authority to resource *log_stream_name* in class LOGSTRM to define, delete, or update the stream {SM.6::SM-R9-LOGGER-4}

- ALTER authority to resource *MVSADMIN.LOGR* in class FACILITY to define or delete a coupling facility structure for use by a log stream {SM.6::SM-R9-LOGGER-6}.

- UPDATE authority to resource IXLSTR.*structure_name* in class FACILITY to associate the named coupling facility structure with a log stream {SM.6::SM-R9-LOGGER-7}.

Applications wishing to administer log streams using the programming interfaces will need:

- ALTER authority to resource *log_stream_name* in class LOGSTRM to define, update the definition of, or delete a log stream {SM.6::SM-R9-LOGGER-8}.

- Additionally, UPDATE to resource name IXLSTR.*structure_name* in class FACILITY to define a log stream that uses a coupling facility structure {SM.6::SM-R9-LOGGER-9}

- Additionally, when defining a log stream modeled upon the definition of another log stream, UPDATE access to resource IXLSTR.*model_structure_name* in class FACILITY (when the model stream has a structure) {SM.6::SM-R9-LOGGER-10}.

- ALTER to resource MVSADMIN.LOGR in class FACILITY if they wish to use logger interfaces to define coupling facility structures {SM.6::SM-R9-LOGGER-11}

8.1.6 Auditing

8.1.6.1 Generation of audit records

The TOE provides a general facility to collect data required for auditing and accounting services. This function, 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 collect security-related auditing information 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}. SMF supports up to 256 different record types. SMF records can only be generated by authorized processes or processes specifically authorized to generate specific types of SMF records under the mediation of the TOE {AU.1::AU.1.2}.

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)
- 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 7: 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.

The LDAP client uses an LDAP extended-operation to request this auditing function. Usage of the remote auditing service requires the LDAP client to have READ authority to FACILITY resource IRR.LDAP.REMOTE.AUDIT {AU.1::AC.2-R9-EIM.5}. The audit record will be created as an SMF type 83 subtype 4 record {AU.1::AU.1-R9-EIM-1}.

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}

8.1.6.2 Protection of the audit trail

SMF writes audit records into either

17. Dedicated SMF data sets that have been defined during system configuration. At least two SMF data sets must be defined by the administrator for compliance with the evaluated configuration. Those data sets need to be protected against unauthorized access by appropriate RACF access control lists. The administrator guidance documentation provides specific guidelines for the protection of the audit trail using RACF.

Or

18. A system log stream, which may reside solely in DASD data sets, or in a combination of data sets and a coupling facility structure for better performance, as specified by the administrator. The administrator configures profiles in the LOGSTRM class to control who can access the data while it exists in the managed log stream, and profiles in the DATASET class to control access to any data extracted from the log stream.

8.1.6.3 Using MVS Data Sets for SMF

When the system is started SMF searches for the first non-full data set in the list of SMF data sets defined. This data set becomes the active SMF data set used to store audit records. Once this data set is full, SMF marks the data set to be processed by the SMF Dump program and takes the next empty data set as the active, searching the list of SMF data sets in a wraparound way {AU.2::AU.2.2}. The operator is also alerted to switch the data set.

SMF data sets that are full need to be processed by the SMF Dump program, IFASMFDP. This program copies the content of a full SMF data set to another data set (the “dump data set”) defined by the installation and marks the SMF data set as empty {AU.2::AU.2.3}. The SMF Dump program itself creates two SMF records (Dump Header and Dump Trailer) that are stored in the beginning and at the end of the dump data set {AU.2::AU.2.4}. Dump data sets must be protected by RACF access control lists.

If no non-full data set is found, SMF stores the records in its buffers until a data set is made available {AU.2::AU.2.5}. If the TOE is configured according to the administrative guidance, the system will halt if no buffer space is left {AU.2::AU.2.6}.

8.1.6.4 Using a System Log Stream for SMF

In contrast to using MVS data sets directly, when using a log stream for the SMF data only one logical stream exists. Although this stream may reside in multiple MVS data sets as determined by system logger processing, the administrator will view the stream as one logical entity, starting with the earliest available data and ending with the current data, rather than dealing with the individual data sets.

Operators do not need to switch SMF data sets, nor dump them to archive storage, nor clear them. Rather, the data can simply reside in the logger-managed data sets.

z/OS provides the IFASMFDL utility program that can extract an administrator-specified set of SMF data from the log stream, based on time/date, system ID, and/or SMF record type and write that extracted data to a standard MVS data set for later processing {AU.2::AU-R9-SMF-1}.

IFASMFDL can invoke exit routines, just as IFASMFDP can, and so the RACF SMF Unload routine will work with IFASMFDL just as with IFASMFDP, providing an interpreted flat-file of RACF-relevant security records for subsequent analysis {AU.2::AU-R9-RACF-1}.

8.1.6.5 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}.

8.1.7 Object reuse

z/OS provides explicit object reuse functionality for the following objects, and z/OS ensures that these objects are prepared for reuse before they are allocated to another subject:

- Memory objects are filled with zeros before they are allocated for the first time to a subject {OR.1::OR.1.1}.

- z/OS data sets are erased when the data is released when the erase-on-scratch option is active {OR.1::OR.1.2}.

- z/OS system log streams that reside in z/OS data sets are cleared by the system logger before it writes any data into them. Similarly, for z/OS log stream data residing in a coupling facility the system logger clears the structure data in the coupling facility before writing any data into the structure {OR.1::OR.1-R9-LOGGER-1}.

- z/OS tape volumes are erased when they are returned to the scratch pool by appropriately configuring the SECCLS parmlib option for the parmlib member EDGRMMxx {OR.1::OR.1-R8-RMM-1} or under control of the appropriate data set profile’s ERASE option when TAPEAUTHDSN=YES is specified in SYS1.PARMLIB(DEVSUPxx) {OR.1::OR.1-R8-RMM-2}.

- z/OS UNIX file system objects and z/OS UNIX IPC objects are cleared before they are made accessible to a new subject (for zFS files, this requires that the zFS IOEFSPRM parameter file has the NBS option defaulted or set to enabled, and that any mount commands or multi-file-system aggregates also have the NBS option set) {OR.1::OR.1.3}.

- LDAP LDBM objects are not specifically cleared when they are deleted, but LDAP does ensure that any data returned from an object is not residual data from some previous object that may have occupied the same physical space in the LDBM database {OR.1::OR.1-R8-LDAP-1}.

8.1.8 TOE self-protection

8.1.8.1 Supporting mechanisms of the abstract machine

The following section provides a short overview of the supporting protection mechanisms of the abstract machine on which z/OS is running. The purpose of this section is to better understand how z/OS uses those mechanisms to protect itself against tampering and bypassing of the security functions of z/OS.

Processor features

The System z processors have two distinctive states: problem and supervisor. A bit in a processor internal special register, the program status word (PSW) indicates if the processor is in problem or supervisor state. When in problem state the processor will not execute so called “privileged instructions”. Those include instructions to perform I/O operations, modify the content of processor control registers, set storage keys for pages within real memory, modify the hardware support tables for virtual memory management or modify critical parts of the PSW like the problem/supervisor bit or the storage key mask bits. When a program in problem state tries to execute one of those instructions, the processor generates a program check interrupt {SP.1::SP.1.1}.

Pages within real storage can be protected using a so-called “storage key” that can be associated with each page of real storage. Programs can modify data within a page only if the storage key in the current PSW matches the storage key of the page or if the storage key in the current PSW is zero {SP.1::SP.1.2}. In addition pages can have an indicator, stating if the page is fetch protected. If this is the case, a program can read data from the page only if the storage key of the page and the storage of the program in the PSW match or if the storage key in the PSW is zero {SP.1::SP.1.3}. Storage protection is in effect whether the processor is in problem or supervisor state. There is one exemption from the rules stated above: If the “Storage Protection Override Control” bit is set in control register 0 of the processor, programs executing with storage key 8 are allowed to store and fetch into storage and from storage with a key of 9.

All processors within a machine share the real storage except for the first 8 KB, which are individual for each processor. The first 8 KB contain the PSWs loaded upon an interrupt.

When a program issues a supervisor call instruction the processor stores the current PSW of the calling program (which contains the instruction pointer pointing to the instruction following the supervisor call instruction) into a fixed location in the processor individual real storage in the first 8KB and loads a dedicated PSW from another location within the first 8 KB. The same procedure applies for interrupts, where each type of interrupt has dedicated locations for the “old” PSW to store and the “new” PSW to fetch. All those locations are within the first 8 KB. Program Call instructions save the current PSW (plus some other information on the caller's context) in the linkage-stack program-call state entry. Control Register 15 serves as a stack pointer to the linkage-stack.

The processor also contains support for virtual memory management. This support allows z/OS to define separate virtual address spaces and define the protection within those address spaces on a per page basis.

In addition to the main processor there is a dedicated I/O hardware subsystem, the "Channel" subsystem that allows I/O operations to be performed in parallel to the normal processor operation. Configuring and programming the I/O subsystem is restricted to programs operating in supervisor state.

The hardware also provides a single time reference within a machine that can be used by all processors. Different time references within different processors in a parallel sysplex may also be synchronized by the hardware. Only users with the privileges to use the operator command to set and change the time may modify the time and date in the TOE {SP.1::SP.1.4}.

Abstract machine modes of operation

z/OS may execute in one of these modes:

- logical partition mode
- VM guest mode

In all of those cases, z/OS operates on an abstract machine that implements the z/Architecture.

In logical partition mode, z/OS has full control of all of the resources allocated to the partition when it has been set up on the hardware management console. The logical partitioning software (PR/SM) starts the processors allocated to a partition in the "interpretative execution" mode using the SIE instruction. Each processor is then "confined" into the boundaries specified for the logical partition with respect to the physical memory and the channels it can access. Whenever a resource "virtualized" by PR/SM is accessed by an instruction on a processor, the processor breaks out of the interpretative environment into the PR/SM code which then services the request in accordance with its own policy. For z/OS this operation is transparent. PR/SM is part of the TOE environment that provides the abstract machine for the operation. PR/SM has been evaluated separately.

In VM guest mode, z/OS is operating within the boundaries defined by the z/VM operating system. z/VM is similar to PR/SM but provides more virtualization functions and more services a guest operating system may request from the virtual machine monitor. Like PR/SM z/VM also uses the SIE instruction to run a guest operating system within the boundaries of the virtual machine. z/VM itself may operate within a logical partition. When z/OS is operating in VM guest mode, the virtual machine monitor system z/VM is part of the TOE environment. z/VM itself is subject to a separate evaluation.

8.1.8.2 Supervisor state routines in z/OS

System services offered by z/OS can be invoked from programs running in problem state using the supervisor call (SVC) and Program Call (PC) instructions of the processor. When the SVC instruction is executed, the executing processor generates an interrupt, stores the current PSW at a fixed location in absolute memory, loads a new PSW from another fixed location in absolute storage and proceeds execution at the address and with the privilege settings defined in this new PSW. During system startup z/OS has defined the new PSW to be loaded into the absolute storage in case of an interrupt or exception for all interrupts and exceptions that may occur. The new PSW contains the address of the SVC interrupt handler and z/OS checks if the caller has the required privileges to obtain the requested service before providing it.

When a Program Call instruction is executed, the hardware checks the authorization of the caller to call the requested PC routine. A program-call number specified by the second operand address is used in a multi-level lookup to locate an entry-table entry (ETE). The program is authorized to use the ETE when the AND of the PSW-key mask in control register 3 and the authorization key mask in the ETE is nonzero or when the CPU is in the supervisor state. The ETE also defines the entry point address of the PC routine and if the PC routine will run in supervisor or problem state.

A number of SVC and PC system services as well as specific parameters of system services are restricted to authorized programs and the service will be rejected if the caller is not authorized. The concept of authorization is discussed in more detail in the next two sections.

8.1.8.3 Authorized programs

In addition to supervisor and PC routines, z/OS has a number of “authorized programs” that need to be trusted because they are not restricted by the security policy defined in this Security Target. An authorized program may call a number of program calls or supervisor calls or use supervisor call parameters that are reserved for authorized programs. In particular, it is authorized to call the MODESET SVC used to switch into supervisor state. With this function, authorized programs can execute any privileged instruction.

A program is authorized if at least one of the following conditions is true:

- The program is executing in supervisor state {SP.3::SP.3.1}
- The program is executing with a PSW key of 0 to 7 or a PSW key mask value that supports at least one key in the range of 0 to 7 in control register 3 {SP.3::SP.3.2}.
- The authorization bit is set in the Job Step Control Block (JSCB) under which the program is executing {SP.3::SP.3.3}.

Whenever a supervisor routine reserved for authorized programs is called or when a parameter reserved for authorized programs is used, the routine invoked to service the request checks if one of the above listed conditions is satisfied. Only if this is true, the request is honored {SP.3::SP.3.4}. Note that the hardware performs some checks when a supervisor routine is called with a Program Call (PC) instruction. In this case the routine implementing the service only needs to perform its own checks if additional restrictions to those implied by the hardware checks apply. Note also that some supervisor routine may be more restrictive, i. e. only a subset of the three conditions mentioned above is checked and the request is rejected if not one of the conditions in the subset apply. For example the hardware can not check if a program running in problem state with a PSW key of 8 is authorized by the authorization bit in the JSCB.

An authorized program can be started in one of the following ways:

- By starting a program from a dedicated program library (defined in the system configuration data set SYS1.PARMLIB) that has the authorization bit set in the directory entry of the member of the partitioned data set (library) containing the program. This program has to be the one started with the EXEC JCL statement of the job step, as a TSO command, as a UNIX process using exec(), or started as a dedicated task by an authorized program using the ATTACH supervisor call with parameters reserved for authorized programs {SP.3::SP.3.5}

  Note: TSO commands might be entered directly by the user at a terminal, executed in a batch job that runs TSO TMP, or entered programmatically using the TSO IKJEFTSR service. They may also be executed by any service that uses either the TMP or IKJEFTSR service such as the REXX 'address tso' function or the Unix shell 'tsocmd' function.

- By starting a started task from an authorized library using the operator START command {SP.3::SP.3.6}

- By starting an authorized program from a zFS file system {SP.3::SP.3.V1R7.1}. A program in a zFS file system is authorized when the authorization bit has been set using the extattr –a command for the file containing the program {SP.3::SP.3.V1R7.2}. A user needs to have been authorized to the BPX.FILEATTR.APF profile in the FACILITY class to set the authorization bit {SP.3::SP.3.V1R7.3}. If a program running in an APF-authorized address space attempts to load a program from zFS that does not have the APF-extended attribute set, the load is rejected {SP.3::SP.3.V1R7.4}. Sanction lists can be defined that restrict access of authorized programs in the z/OS Unix System Services environment to files and directories defined in those sanction lists {SP.3::SP.3.V1R7.5}.

Libraries that can contain authorized programs need to be protected from unauthorized modifications including the possibility to add new programs to the library. zFS files containing authorized programs also need to be protected from unauthorized modifications. The discretionary and mandatory access control features of z/OS have to be used to protect those libraries.

The IKJTSOxx member of SYS1.PARMLIB can be used to define the authorized programs and commands that can be executed in the TSO environment {SP.3::SP.3.V1R7.6}.

Some trusted subsystems of z/OS are started as part of the standard startup procedure or may be later started by explicit request of a properly authorized user.

Protection of authorized programs

Authorized programs need to be trusted because they are allowed to increase their privileges up to running in supervisor mode with a storage key of zero. Authorized programs therefore must be carefully protected from unauthorized modification and the system must be protected from adding authorized programs other than those allowed in the evaluated configuration.

A program executes with authorization when:

- the program was linked with an authorization code into an authorized library or assigned the authorization attribute in the zFS file system and

- the program is the first program started within a job step or is started as an authorized TSO command. All programs started within the same job step by this program also run authorized {SP.3::SP.3.7}.

To protect the integrity of the TOE the following security measures must be in place:

- all program libraries that are authorized libraries must be protected from update or alter access by other than the system administrators using the discretionary and mandatory access control functions and

- the system configuration library needs to be protected from any modification by other than the system administrators using the discretionary and mandatory access control functions

No program other than the programs allowed in the evaluated configuration should be linked with an authorization code in the authorized libraries or specified in the PPT as having a system key or supervisor state

Note that once a job step is authorized all programs called as part of the execution of the job step run with authorization and need to be trusted. The TOE protects trusted programs from accidentally executing any program from an untrusted library {SP.3::SP.3.8}. Trusted programs can take deliberate actions to bypass this protection.

Note that when within a non-authorized (untrusted) job step a program linked with authorization code into an authorized library is called, the program executes without authorization and will fail if it attempts to use privileges allowed only for programs executing with authorization {SP.3::SP.3.9}.

8.1.8.4 Program Signing and Verification

The TOE supports digital signatures for program objects stored in PDSE data sets. The digital signature for a program object is optionally created during the process of "binding" the object modules into the form of an executable program object and storing it into the PDSE data set. Later, when a user attempts to load or execute that program the system can optionally validate the signature and use the result of the validation to determine whether to allow use of the program or not.

Program Signing

{SP.4::SP.4-R12-Binder-1} Users executing the Binder to create a program object in a PDSE may optionally specify the SIGN=YES option to request that the Binder attempt to digitally sign the program object and save the signature in the PDSE directory when it saves the program object.

{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-2} The specified key ring must contain a digital certificate to use for signing, together with the RSA private key for that certificate, and the CA certificate(s) for the certificate. Certificate requirements:

- The code signing certificate and each CA certificate in the CA chain must be signed using either the sha256WithRSAEncryption or sha1WithRSAEncryption signature algorithms.
- 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.

Program Signature Verification

{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: Running IRRVERLD is required only when the administrator has made signature verification setup changes since the last IPL.)

8.1.9 Session Locking

The hardware available for the TOE (see section HW-CONFIG) does not provide means for direct connections of synchronous terminals any more. Since the requirements stated in the

SFRs FTA_SSL.1 and FTA_SSL.2 are targetted towards such direct connections, they are irrelevant to this TOE and therefore trivially met.

8.1.10 Implementation of cryptographic functions

Several components of the TOE use cryptographic functions as part of their security functions. With the inclusion of the Integrated Cryptographic Services Facility (ICSF) the cryptographic functions may be provided by hardware coprocessors attached to the TOE. ICSF checks for the availability of hardware support for individual cryptographic functions and uses this when appropriate. In the case where no cryptographic coprocessor is attached to the TOE, the components that use ICSF for cryptographic operations (IPSec, System SSL, z/OS Network Authentication Service) will use software implementation of the cryptographic algorithms. IPSec always requires ICSF for AES support, whether using the hardware or software. SSH will always use its own software implementation of the cryptographic algorithms and will use hardware support only for the key generation process. For the RACDCERT command, the command issuer chooses, by the keywords chosen, whether to use ICSF (if available) or a software implementation.

Note that CPACF is not considered a cryptographic coprocessor but a native capability of the z/Architecture processor. While the functions provided by CPACF may differ by different processor models, the functions provided by the CPACF instructions may be used by any application.

8.1.10.1 FIPS 140-2

In addition to providing cryptographic support, several components of z/OS have been designed to meet the Federal Implementation Processing Standard (FIPS) 140-2 Level 1 criteria. FIPS is a standard that has been issued by the National Institute of Standards and Technology and Communications Security Establishment (CSE) of the Government of Canada . This standard specifies security requirements for a cryptographic module which is utilized within a security system to protect sensitive or valuable information. To meet this standard, cryptographic modules are tested against requirements defined in the FIPS PUB 140-2, Security Requirements for Cryptographic Modules. These security requirements cover 11 areas related to the design and implementation of a cryptographic module.

The z/OS components designed to meet FIPS 140-2 Level 1 are ICSF and System SSL. When utilizing these components in FIPS mode, they will restrict cryptographic processing to what is approved or allowed in FIPS mode.

For System SSL, The algorithms/protocols will be restricted to: symmetric TDES (3-key), AES CBC (128 and 256 bit); hashing algorithms SHA-1 (160-bit) and SHA-2 (224, 256, 384, 512 bit); asymmetric algorithms RSA (1024-4096 bit) and DSA (1024 bit); Diffie-Hellman key agreement algorithm DH (2048 bit); TLS V1.0 and TLS V1.1 protocols. {CR.1::CR-R12-SSL-1}

For ICSF, the algorithms will be restricted to: symmetric algorithms TDES (3-key), AES ECB, CBC, GCM (128, 192 and 256 bit); hashing algorithms SHA-1 (160 bit) and SHA-2 (224, 256, 384 and 512 bit); asymmetric algorithms RSA (1024-2048 bit), DSA (1024 bit) and ECDSA (160-521 bit); Diffie-Hellman key agreement algorithms DH (1024-2048 bit) and EC-DH (160-521 bit).{CR.1::CR-R12-ICSF-1}

Approved cryptographic algorithms are tested for conformance by inputting defined test vectors and comparing the output results against expected output vectors. No conformance testing exists for Diffie-Hellman. {CR.1::CR-R12-SSL-2}{CR.1::CR-R12-ICSF-2}

Each component has unique methods to control FIPS mode execution.

System SSL applications that execute in FIPS mode must utilize the gsk_fips_state_set API prior to other System SSL functions {CR.1::CR-R12-SSL-3}. System SSL allows switching from FIPS to non-FIPS mode but not vice versa {CR.1::CR-R12-SSL-4}.

ICSF applications has 2 methods: 1) Utilizing the FIPSMODE start up option for the ICSF started task. FIPSMODE allows ICSF to be started in strict FIPS mode, compatibility mode which allows select applications to execute in FIPS mode or non-FIPS mode {CR.1::CR-R12-ICSF-3}. Compatibility mode applications by default execute in FIPS mode {CR.1::CR-R12-ICSF-4}. FIPSEXEMPT.<token-label> in the cryptoz class allows a user (application) to be exempt from FIPS when using a specified PKCS#11 token {CR.1::CR-R12-ICSF-5} or the calling application has indicated that the specified key must always be used in a FIPS 140-2 compliant fashion {CR.1::CR-R12-ICSF-6}.

ICSF and System SSL modules that execute within FIPS mode have been digitally signed during the bind process using RSA and SHA-256 signatures. Upon validation of the signatures, the modules are verified to ensure they have not been tampered with since being built. Verification is performed during the module load process {CR.1::CR-R12-SSL-5}{CR.1::CR-R12-ICSF-7}.

In the evaluated configuration, Application Transparent TLS (AT-TLS) {CR.1::CR-R12-CS-1}, Network Security Services daemon (NSSD) {CR.1::CR-R12-CS-2}, IKED {CR.1::CR-R12-CS-3} and IPSec support in the Communications Server stack {CR.1::CR-R12-CS-4} allow for execution in either FIPS or non-FIPS mode.

The following hardware support options for cryptographic functions are available:

8.1.10.2 CPACF

This feature is part of the instruction set of the z/Architecture. Instructions are available for DES encryption and decryption, TDES encryption and decryption and SHA-1 and SHA-2 hashing. In addition a DES based pseudo-random number generator is provided. The instructions for those operations are part of the general instructions of a z/Architecture processor and may therefore be used by programs in any processor state. The instructions are:

- CIPHER MESSAGE (KM)
- CIPHER MESSAGE WITH CHAINING (KMC)
- COMPUTE INTERMEDIATE MESSAGE DIGEST (KIMD)
- COMPUTE LAST MESSAGE DIGEST (KLMD)

The KMC instruction also provides a DES based pseudo random number generator. For details of those instructions see [ZARCH].

Specific z/Architecture processor models also support 128-bit or 256-bit AES encryption/decryption, and SHA-224, SHA-256, SHA-384, or SHA-512 message digests.

8.1.10.3 PCIXCC

The PCIXCC is a PCI based coprocessor card with its own main processor (a pSeries processor), a cryptographic hardware coprocessor and its own memory. It contains an operating system (Linux) on top of which application programs implement the functions of IBM's Common Cryptographic Architecture (CCA). Basically CCA commands are passed by the TOE to the coprocessor, processed there and the result is passed back to the TOE. Logical access to the coprocessor functions is controlled by the TSF and unprivileged programs can access those functions only through the ICSF component of the TSF and only for services they are allowed to use.

The coprocessor has the ability to generate RSA key pairs and retain the private key in the coprocessor. When generating such a key pair the coprocessor would only pass back the public key and a key identifier that can be used to request the coprocessor to use a specific private key. The private key will never leave the coprocessor in clear. Only export in encrypted form for backup purposes is possible.

8.1.10.4 PCICA

The PCICA is a PCI based cryptographic coprocessor card that only contains the cryptographic hardware coprocessor but no own general purpose processor, memory or operating system. The cryptographic coprocessor is the same as the one used in the PCIXCC. This coprocessor is only used as an accelerator for RSA encryption and decryption. RSA encryption and decryption are the only cryptographic functions the coprocessor can perform. Since the PCICA has no own storage, the key has to be provided by the TOE each time it uses the coprocessor. The coprocessor can accept keys both in "normal" format as well as in CRT format (as defined in PKCS#1). The operation code submitted to the card identifies the operation and the key format. Operation code, input data, output data, data length, key length and the key are passed in a block to the coprocessor, which then performs its operation and passed the result back in the output data field. For applications that just need fast RSA encryption and decryption (e. g. a server that allows a lot of SSL based connections), this provides a significantly faster method for RSA operations than using the PCIXCC and the overhead associated with the operations on the PCIXCC card. Of course the PCICA does not provide an option for "secure" private keys.

8.1.10.5 CryptoExpress2 (CEX2)

The CryptoExpress2 is basically a PCIXCC coprocessor with an additional direct interface to the cryptographic coprocessor. The configuration of the card determines if it operates like a PCIXCC (CEX2C) or in PCICA (CEX2A) mode. The hardware and the software on the card are identical to the PCIXCC and therefore (depending on the configuration) the coprocessor acts behaves either identical to a PCIXCC (in CEX2C mode) or identical to a PCICA (in CEX2A mode), except that on the System z9 or later the CEX2C supports 4096-bit RSA key operations for clear and secure keys.

8.1.11 Self-test functions

The underlying hardware of the TOE includes a large set of self-test functions for the correct operation of the functions of the processor, the memory and the attached I/O devices. Errors detected by those functions result in a machine-check interrupt (for errors in the processor or the memory) or an error indicator in the information returned by the TEST SUBCHANNEL instruction in the case of an error within an I/O device. The conditions that are checked internally by the underlying hardware are listed in chapter 11 of [ZARCH]. Errors detected by the hardware will result in the error being reported to the TOE in the machine-check interruption code. The hardware will determine if the problem allows for a safe handling by the software running on the hardware (the TOE) and pass control to this software by generating a machine check interrupt. This is the case where either the hardware could correct the error or where the error is related to a piece of the hardware that still allows a CPU to safely treat the error.

Errors from I/O devices are detected and reported by the channel subsystem of the hardware. Chapter 16 of [ZARCH] describes in the section on the Subchannel-Status Word the Subchannel-Status Field values that indicate an error detected by the channel subsystem including device errors or errors detected in the data being transferred (using error detection and correction codes as part of the data).

In addition IBM field service has specific utilities that allow to locate the hardware error. Those include a utility that performs a subset the test performed by the System Assurance Kernel (SAK) tool used within IBM to verify full compliance to the z/Architecture. Neither the hardware nor the utilities used by the IBM service personnel are part of the TOE but extensive and continuous abstract machine testing is performed by the TOE environment.

Due to the extensive self-test functions of the underlying hardware the TOE does not provide self-test functions of the underlying hardware. Those functions would not be able to identify and report a problem the self-test functions of the hardware had not already identified and handled.

8.2 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 Measures
ADV_ARC.1	The architecture is described in [ZARCH], which describes the protection functionality provided by the underlying hardware and firmware, and in [ABC-V10], which describes how z/OS uses the supporting hardware functionality to define separate address spaces and to provide a separated execution environment for the TSF. In addition the start-up process is described.
ADV_FSP.4	The functional specification for z/OS consists of the description of the supervisor calls (as the description of the macros used to generate the code for calling the system function), the description of the commands provided to users, system administrators and auditors to use and manage the security functions and the description of the system configuration data sets. In addition there is a document providing an overview of the system functions with separate parts for functions available to all programs and functions or parameters of functions available to authorized programs only.
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_TDS.3	A high-level design of the security functions of z/OS 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 z/OS and points to other existing documents for further details where appropriate. In addition IBM provides dedicated low-level design documentation for the modules of all SFR enforcing subsystems of the TOE and summary descriptions for all modules of the SFR-supporting and SFR-non-interfering subsystems.. 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 overall system management as well as the management for individual components of z/OS. Especially for the management of RACF a System Administrator 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 z/OS 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

SAR	**Assurance Measures**
	systems
ALC_CMS.4	z/OS is developed at different sites each using a well defined and highly automated configuration management system. Each 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 and test results are all maintained under configuration management.
ALC_DEL.1	z/OS is delivered 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 for example 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	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.1	The tools used in the development process and product generation are documented with their behavior, options and usage assumptions.
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.1	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 Target. Test results are documented such that the tests can be repeated.

SAR	Assurance Measures
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.3	IBM has its own team that performs vulnerability analysis and penetration testing for z/OS. This team has a long term experience with potential security problems within z/OS and is also integrated in the design reviews. The developer vulnerability analysis will report the activities and findings of this team.

Table 13: Assurance measures meeting the TOE security assurance requirements

9 Abbreviations, Terminology and References

9.1 Abbreviations

ACEE

Accessor Environment Element

AT-TLS

Application-Transparent TLS

CC

Common Criteria

cn

common name

DAC

discretionary access control

DN

distinguished name

IOCDS

input/output configuration data set

LDAP

Lightweight Directory Access Protocol

MAC

mandatory access control

PADS

program access to data sets

PKI

Public Key Infrastructure

PP

Protection Profile

PR/SM

Processor Resource/Systems Manager™

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

9.2 Terminology

This section contains definitions of technical terms that are used with a meaning specific to this document. Terms defined in the [CC] are not reiterated here, unless stated otherwise.

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*.

common name (cn)

One component of an LDAP object’s complete name, usually specified as cn=*name*.

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).

distinguished name (DN)

The complete name of an object in an LDAP directory, or the complete name of the subject or issuer of a digital certificate.

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

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.

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 *SECLABEL*s.

security level (IBM)

A hierarchical designation for data that represents the sensitivity of the information. Security levels are sometimes referred to as *SECLEVEL*s. 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.
The equivalent term in other IBM documentation is *security label*.

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.

9.3 References

ABC-V10 **ABCs of z/OS System Programming Volume 10**

Version SG24-6990-03

Date September 2008

Location http://www.redbooks.ibm.com/redbooks/pdfs/sg246990.pdf

AIS20 **AIS-20, Functionality classes and evaluation methodology for deterministic random number generators**

Version 1

Date December, 1999

CC **Common Criteria for Information Technology Security Evaluation**

Version 3.1R3

Date July 2009

Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R3.pdf

Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R3.pdf

Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R3.pdf

CEM **Common Methodology for Information Technology Security Evaluation**

Version 3.1R3

Date July 2009

Location http://www.commoncriteriaportal.org/files/ccfiles/CEMV3.1R3.pdf

OSPP **Operating System Protection Profile**

Version 2.0

Date 2010-06-01

OSPP-EIA **OSPP Extended Package -- Extended Identification and Authentication**

Version 2.0

Date 2010-05-28

OSPP-LS **OSPP Extended Package -- Labeled Security**

Version 2.0

Date 2010-05-28

PMLS **z/OS V1R12 Planning for Multilevel Security and the Common Criteria**

Version Twelfth Edition

Date July, 2011

File name GA22-7509-11

RFC4217 **Securing FTP with TLS**

Date received

SSHV2 **see RFC4251 to RFC4253**

Date received

SSLV3 **The SSL Protocol Version 3.0**

Date received

Location http://wp.netscape.com/eng/ssl3/draft302.txt

TLSV1 **The TLS Protocol Version 1.0**

Date received

Location ftp://ftp.rfc-editor.org/in-notes/rfc2246.txt

TLSV1.1 **See [RFC4346]**

Date received

ZARCH **IBM: z/Architecture: Principles of Operation**

Version Ninth Edition

Date August, 2010

File name SA22-7832-08