Security Target for IBM z/OS Version 1 Release 7

Version 2.14

February 16, 2006

TABLE OF CONTENTS

1. Introduction

This is version 2.14 of the Security Target for IBM® z/OS® Version 1 Release 7.

1.1 Security Target (ST) identification

Title: IBM z/OS Version 1 Release 7
Version: 2.14
Keywords: access control, discretionary access control, general-purpose operating system, information protection, security labels, mandatory access control, security, UNIX®

This document is the Security Target for the Common Criteria (CC) evaluation of the IBM z/OS Version 1 Release 7 operating system. It is conformant to the Common Criteria for Information Technology Security Evaluation [CC].

1.2 ST overview

This Security Target (ST) documents the security characteristics of the IBM z/OS Version 1 Release 7 operating system with the additional required licensed programs (see section 2.3 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 @server® zSeries® or z9® mainframe computers, empowering the use of their most advanced features, such as the new 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 @server zSeries or z9 mainframe computer. Several zSeries or z9 computers 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 z800, the z900, the IBM @server zSeries (z890 and z990), and z9 (109) servers. 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 2.3.1 for the software configuration of z/OS used in this evaluation. The security functions subject to this evaluation are described in Chapters 5 and 6 of this document.

With such outstanding security features as multilevel security support, IBM z/OS meets all of the requirements of the Labeled Security Protection Profile (LSPP) and the Controlled Access Protection Profile (CAPP), which were developed by the Information Systems Security Organization within the National Security Agency to map the TCSEC B1 (LSPP) and C2 (CAPP) classes of the U. S. Department of Defense (DoD) Trusted Computer System Evaluation Criteria (TCSEC) to the Common Criteria framework. This Security Target therefore claims full compliance with the requirements of these Protection Profiles and also includes additional functional and assurance packages beyond those required by LSPP and CAPP.

1.3 Common Criteria conformance

This Security Target is *CC Part 2 extended* and *Part 3 conformant*, with a claimed Evaluation Assurance Level of EAL4, augmented by ALC_FLR.1.

1.4 Strength of function

The claimed minimum strength of function for this TOE is SOF-medium.

1.5 Structure

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

- Section 1 is the Introduction.
- Section 2 is the Target of Evaluation (TOE) description
- Section 3 provides the statement of TOE security environment
- Section 4 provides the statement of Security objectives
- Section 5 provides the statement of Security requirements
- Section 6 provides the TOE summary specification, which includes the detailed specification of the IT security functions
- Section 7 provides the Protection Profile claims
- Section 8 provides the Rationale for the security objectives, security requirements, and TOE summary specification

1.6 Terminology

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

abstract machine

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

access

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

access control policy

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

Accessor Environment Element

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

authorization

If an authorized administrator 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 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.

category

See *security category*.

classification (LSPP)

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

discretionary access control (DAC)

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

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.

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 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 nonhierarchical categories (security category). Security labels are used as the base for mandatory access control decisions. Security labels are sometimes referred to as *SECLABELs.*

security level (IBM)

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

security level (LSPP)

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 LSPP 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 that uniquely identifies a user. Users who may use UNIX services will additionally have a numerical user identifier (UID) that is used by the UNIX subsystem for access decisions. The user name is an additional attribute that usually holds the user's full name. While users can modify their user names, only administrators can change user IDs.

1.7 Abbreviations

ACEE	Accessor Environment Element
CC	Common Criteria
DAC	discretionary access control
IOCDS	input/output configuration data set
MAC	mandatory access control
PADS	program access to data sets
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

1.8 References

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

[CAPP] Controlled Access Protection Profile, Version 1.d, Information Systems Security Organization. 8 October 1999

[CC] Common Criteria for Information Technology Security Evaluation, Parts 1 to 3, CCMB-2005-08-001 to CCMB-2005-08-003, Version 2.3, August 2005

[CEM] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, CCMB-2005-08-004, Version 2.3, August 2005

[GUIDE] ISO/IEC PDTR 15446 Title: Information technology – Security techniques – Guide for the production of protection profiles and security targets, ISO/IEC JTC 1/SC 27 N 2449, 2000-01-04

[IPSEC] "Security Architecture for the Internet Protocol", ftp://ftp.rfc-editor.org/in-notes/rfc2401.txt

[LSPP] Labeled Security Protection Profile, Version 1.b, Information Systems Security Organization, 8 October 1999

[PMLS] z/OS V1R7.0 Planning for Multilevel Security and the Common Criteria, Sixth Edition, March, 2006, GA22-7509-05

[RFC3268] RFC 3268 - Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS), June 2002, ftp://ftp.rfc-editor.org/in-notes/rfc3268.txt

[SSLV3] "The SSL Protocol Version 3.0", http://wp.netscape.com/eng/ssl3/draft302.txt

[TLSV1] "The TLS Protocol Version 1.0", ftp://ftp.rfc-editor.org/in-notes/rfc2246.txt

[ZARCH] IBM: z/Architecture: Principles of Operation, SA22-7832-04, Fifth Edition, September 2005

1.9 Trademarks

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

- Advanced Function Presentation
- AFP
- DFS
- DFSORT
- @server
- IBM
- Infoprint
- MVS
- PR/SM
- Print Services Facility
- Processor Resource/Systems Manager
- RACF
- VTAM
- z/Architecture
- z/OS
- z/VM
- zSeries
- z9

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

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

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

2. Target of Evaluation (TOE) description

The Target of Evaluation (TOE) is the z/OS operating system with the software components as described in Section 2.3. z/OS is a general-purpose, multi-user, multi-tasking operating system for enterprise computing systems. Mutltiple 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:

- an IBM zSeries or z9 processor (z800, z890, z900, or z990, or z9 109)
- a logical partition of an IBM zSeries or z9 processor (PR/SM)
- z/VM® executing directly on a zSeries or z9 processor or in a logical partition of PR/SM

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

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: LSPP-compliant and CAPP-compliant. 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 LSPP mode only are marked accordingly.

2.1 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 of the z/OS UNIX System Services that provide similar functions as started procedures or tasks based on UNIX interfaces

These services can be accessed by users local to, or with otherwise protected access to, the computer systems.

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. The TOE authenticates the claimed identity of a user before allowing this user to perform any further security-relevant actions. Users are allowed to execute programs that accept network connections on ports the user has access to. An example would be an Anonymous FTP server. In this case the program has no knowledge about the external "user". The program is executing with the rights of the z/OS user that started it and this user has been authenticated.

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 (compliant to the requirements of the "Controlled Access Protection Profile" [CAPP]) and one with both discretionary and mandatory access control where the mandatory access control is fully enabled for all subjects and objects (compliant to the requirements of the "Labelled Security Protection Profile" [LSPP]). In commercial environments it is often useful to activate only part of the mandatory access control functions required in this Security Target for full compliance to LSPP. 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 LSPP 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 LSPP 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 LSPP 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.

2.2 Summary of security features

The primary security features of the product are:

- identification and authentication
- discretionary access control

- in LSPP mode: mandatory access control and support for security labels
- 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.

2.2.1 Identification and authentication

z/OS provides identification and authentication of users by the means of an alphanumeric user ID and a system-encrypted password.

In the evaluated configuration, all individual 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 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, an installation may decide to define a user ID that is used for access checking for users that have not been authenticated. This allows an installation to define resources unauthenticated users may access using the web browser of their client system that connects to the HTTP server executing on the TOE. Users may still authenticate to the HTTP server using their user ID and password 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 that must usually be changed by the user during initial logon.

2.2.2 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), 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 two DAC mechanisms. The z/OS standard DAC mechanism is used for most protected objects, except for UNIX file system objects, which are protected by the z/OS UNIX DAC mechanism.

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. These discrete profiles are valid for a single, named resource and generic profiles applicable to a group of resources. 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.

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.

2.2.3 Mandatory access control and support for security labels in LSPP mode

In addition to DAC, z/OS provides mandatory access control (MAC) in LSPP mode, which imposes access restrictions to information based 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.

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.

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.

2.2.4 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 the resource. 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 LSPP mode) MAC mechanisms.

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.

Auditors can unload selected parts of the SMF database for further analysis into human-readable formats or for upload to a query or reporting package, such as DFSORT™.

2.2.5 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.

2.2.6 Security management

z/OS provides a set of commands and options to adequately manage the TOE's security functions. 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 LSPP 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 can be delegated to group security administrators.
- Users can change their own passwords, their default groups, and their user names.
- In LSPP mode: users can choose their security labels at login.
- 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[1] or by security administrators.

2.2.7 Secure communication

z/OS provides means of secure communication between systems sharing the same security policy. In LSPP mode, communication within TOE parts coupled into a sysplex can be multilevel, whereas other communication channels are assigned a single security label. In CAPP 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 LSPP mode, communication is permitted between any two addresses that have equivalent labels. In LSPP 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.

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]), which can be applied transparently to communications without changing the applications using it.

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.

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. 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). Web Services are provided by the HTTP server. All those types of services use RACF services for identification, authentication, and access control.

2.2.8 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

[1] For named UNIX objects, the profile owner is the object owner.

- 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 that allow authorized administrators to check the correct operation of the underlying abstract machine.

2.3 Configurations

2.3.1 Software configuration

The Target of Evaluation, z/OS Version 1 Release 7, consists of:

- z/OS Version 1 Release 7 Common Criteria Evaluated Base Package:
  - z/OS Version 1 Release 7 (z/OS V1R7, program number 5694-A01),
  - Overlay Generation Language Version 1 (OGL V1R1, program number 5688-191)
  - IBM Print Services Facility™ Version 4 Release 1 for z/OS (PSF V4R1, 5655-M32).
- PTFs UA22055, UA22598, and UA90249.

The same software elements are used in the LSPP and CAPP mode of operation. 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 V1R7 Common Criteria Evaluated Base package must be installed according to the directions delivered with the media and configured according to the instructions in Chapter 7, "The evaluated configuration for the Common Criteria" in *z/OS Planning for Multilevel Security and the Common Criteria* ([PMLS]).

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

This explicitly excludes replacement of any element in the ServerPac providing security functions relevant to this evaluation by other third-party products.

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 component from the UNIX System Services Element
- The Common Information Model element
- The Distributed Computing Environment (DCE) component of the Integrated Security Services element
- DCE Base Services
- The DFS™ Server Message Block (SMB) and DFS DCE-DFS components of the Distributed File Service element
- The Enterprise Identity Mapping component of the Integrated Security Services element
- The Firewall Technologies Base component of the Integrated Security Services element
- Infoprint® Server
- The Integrated Cryptographic Service Facility (ICSF) component of the Cryptographic Services element
- JES3
- The Lightweight Directory Access Protocol (LDAP) server component of the Integrated Security Services element
- Managed System Infrastructure for Operations (msys for Operations)
- The Multiple Virtual Storage / Advanced Program-to-Program Communication (MVS/APPC) component of the BCP
- The Network Authentication Service component of the Integrated Security Services element
- Network File System (NFS Server and NFS Client)
- 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 cannot be used in the evaluated configuration.

The PassTicket authentication mechanism has not been part of the evaluation and cannot be used in the evaluated configuration.

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

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

Note: The evaluated software configuration is not invalidated by installing and operating other appropriately-certified components that possibly run authorized. However, the following restrictions apply:

- The security policies implemented by those components must not undermine those described in this document.
- The evaluation of those components must show that the component does not undermine the security policies described in this document.

2.3.2 Hardware configuration

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

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

- IBM zSeries model z800
- IBM zSeries model z890
- IBM zSeries model z900
- IBM zSeries model z990
- IBM z9 model 109

In addition, the TOE may run on a virtual machine provided by the 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 CAPP mode: any printer that is supported by the TOE.
  - in LSPP 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), with the exception of RVA devices.
  - Tape drives.
- 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.

3. TOE security environment

3.1 Introduction

The statement of the TOE security environment 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 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.

3.2.1 Physical assumptions

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.LOCATE

The processing resources of the TOE will be located within controlled access facilities that will prevent unauthorized physical access.

A.PROTECT

The TOE hardware and software critical to security policy enforcement will be protected from unauthorized physical modification.

3.2.2 Personnel assumptions

It is assumed that the following personnel conditions will exist:

A.MANAGE

There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.

A.NO_EVIL_ADM

The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation.

A.COOP

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 cooperative manner in a benign environment.

3.2.3 Procedural assumptions

The ability of the TOE to enforce the intent of the organizational security policy, especially with regard to the mandatory access controls, is dependent upon the establishment of procedures. It is assumed that the following procedural controls exist.

A.CLEARANCE (LSPP mode only)

Procedures exist for granting users authorization for access to specific security levels.

A.SENSITIVITY (LSPP mode only)

Procedures exist for establishing the security level of all information imported into the system, for establishing the security level for all peripheral devices (such as printers, tape drives, and disk drives) attached to the TOE, and marking a sensitivity label on all output generated.

3.2.4 Connectivity assumptions

For the TOE to operate in a network, it is assumed that the following assumptions hold:

A.PEER

Any other systems with which the TOE communicates are assumed to be under the same management control and operate under the same security policy constraints. The TOE may be deployed in networked or distributed environments only if the entire network operates under the same constraints and resides within a single management domain. There are no security requirements that address the need to trust external systems or the communications links to such systems.

A.CONNECT

All connections to peripheral devices and other systems reside within the controlled access facilities unless they are protected by the TLSv1, SSLv3, or IPSec protocol. The TOE only addresses security concerns related to the manipulation of the TOE through its authorized access points. Internal communication paths to access points such as terminals or job entry stations are assumed to be adequately protected.

3.3 Threats

In compliance with the Labeled Security Protection Profile (LSPP) and the Controlled Access Protection Profile (CAPP), this Security Target has derived all security objectives from the statement of Organizational Security Policy found in the following section. Therefore, there is no statement of the explicit threats countered by this Security Target.

The threats to be countered by the TOE are therefore those of the violations of the Organizational Security Policies defined in Section 3.4 of this document. The *IT assets* to be protected comprise the information stored, processed, or transmitted by the TOE. The term *information* is used here to refer to all data held within the TOE, including data in transit between different systems as part of a parallel sysplex.

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

- unauthorized users of the TOE (that is, individuals who have not been granted the right to access the system)
- authorized users of the TOE (that is, individuals who have been granted the right to access the system)

The threat agents are assumed to originate from a well-managed user community in a non-hostile working environment, and hence the product protects against threats of inadvertent or casual attempts to breach the system security. The TOE is not intended to be applicable to circumstances in which protection is required against determined attempts by hostile and well-funded attackers with a high level of expertise to breach system security.

3.4 Organizational security policies

The Controlled Access Protection Profile (CAPP) as well as the Labeled Security Protection Profile (LSPP) both define organizational security policies. The following text, which is identical in CAPP and LSPP, provides the rationale for this:

An organizational security policy is a set of rules or procedures imposed by an organization upon its operations to protect its sensitive data. Although the following organizational security policies are drawn from DoD Manual 5200.28-M (Techniques and procedures for Implementing, Deactivating and Evaluating Resource Sharing ADP Systems) [ADP], they apply to many non-DoD environments as well.

P.AUTHORIZED_USERS

Only those users who have been authorized to access the information within the system may access the system.

P.NEED_TO_KNOW

The system must limit the access to, modification of, and destruction of the information in protected resources to those authorized users who have a “need to know” for that information.

P.ACCOUNTABILITY

The users of the system shall be held accountable for their actions within the system.

P.CLASSIFICATION (LSPP mode only)

The system must limit the access to information based on sensitivity, as represented by a label, of the information contained in objects, and the formal clearance of users, as represented by subjects, to access that information. The access rules enforced prevent a subject from accessing information which is of higher sensitivity than it is operating at and prevent a subject from causing information from being downgraded to a lower sensitivity.

Note: The method for classification of information is made based on criteria set forth by the organization. This is usually done based on relative value to the organization and its interest in limiting dissemination of that information. The determination of classification of information is outside the scope of the IT system; the IT system is only expected to enforce the classification rules, not determine classification. The method for determining clearances is also outside the scope of the IT system. It is essentially based on the trust placed in individual users by the organization. To some extent, it is also dependent upon the individual’s role within the organization.

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 Security objectives for the TOE

The IT security objectives are:

O.AUTHORIZATION

The TSF must ensure that only authorized users gain access to the TOE and its resources.

O.DISCRETIONARY_ACCESS

The TSF must control access[2] to resources based on identity of users. The TSF must allow authorized users to specify which resources may be accessed by which users.

O.MANDATORY_ACCESS (LSPP mode only)

The TSF must control access to resources based upon the sensitivity and categories of the information being accessed and the clearance of the subject attempting to access that information.

O.AUDITING

The TSF must record the security relevant actions of users of the TOE. The TSF must present this information to authorized administrators.

O.RESIDUAL_INFORMATION

The TSF must ensure that any information contained in a protected resource is not released when the resource is recycled.

O.MANAGE

The TSF must provide all the functions and facilities necessary to support the authorized administrators that are responsible for the management of TOE security.

O.ENFORCEMENT

The TSF must be designed and implemented in a manner that ensures that the organizational policies are enforced in the target environment.

O.COMPROT

The TSF must be designed and implemented in a manner that allows for establishing a trusted channel between the TOE and another trusted IT product that protect the user data transferred over this channel from disclosure and undetected modification.

[2] a typographic error in [LSPP] has been corrected here.

4.2 Security objectives for the TOE environment

The TOE is assumed to be complete and self-contained and, as such, not dependent upon any other products to perform properly. However, certain objectives with respect to the general operating environment must be met. The following are the non-IT security objectives:

OE.INSTALL

Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner that maintains IT security objectives.

OE.PHYSICAL

Those responsible for the TOE must ensure that those parts of the TOE critical to security policy are protected from physical attack, which might compromise IT security objectives.

OE.CREDEN

Those responsible for the TOE must ensure that all access credentials, such as passwords or other authentication information, are protected by the users in a manner that maintains IT security objectives.

OE.HW_SEP

The underlying abstract machine must provide a separation mechanism that can be used by the TOE to protect the TSF and TSF data from unauthorized access and modification.

OE.CLASSIFICATION (LSPP mode only)

Those responsible for the TOE must ensure that users of the TOE are cleared for access to information depending on the classification of the information. They must also ensure that information is correctly classified to be protected by the security functions of the TOE.

5. Security requirements

5.1 TOE security: functional requirements

This chapter defines the functional requirements for the TOE. Functional requirement components in this Security Target were drawn from Part 2 of the CC. Some functional requirements are extensions to those found in the CC.

CC-defined operations for assignment, selection, and refinement were used to tailor the requirements to the level of detail necessary to meet the stated security objectives. The operations already performed in the Labeled Security Protection Profile (LSPP) -- assignments, selections, and refinements -- are shown in italics. Additional assignments, selections, and refinements made in this Security Target, as well as additional security functional requirements introduced as extensions to the LSPP in this Security Target, are shown in green italics.

SFRs are marked “LSPP mode only” if they are only applicable in the LSPP mode of operation. All other SFRs (or portions thereof) not marked as "LSPP mode only" are applicable in both LSPP and CAPP mode. Application notes marked “from LSPP” have been copied from this protection profile. For all SFRs not explicitly marked as “LSPP mode only”, these application notes are identical to the application notes found in CAPP.

5.1.1 Security audit (FAU)

5.1.1.1 Audit data generation (FAU_GEN.1)

FAU_GEN.1.1 The TSF shall be able to generate an audit record of ***the auditable events listed in column “Event” of Table 5-1 (Auditable events). This includes all auditable events for the basic level of audit, except FIA_UID.1’s user identity during failures and audit events for the security functional requirements added in addition to LSPP (FCS_CKM.1, FCS_CKM.2, FCS_COP.1, FMT_SMF.1, FPT_TDC.1, FTP_ITC.1).***

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

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

b) (in LSPP mode) The sensitivity labels of subjects, objects, or information involved; and

c) The additional information specified in the “Details” column of Table 5-1 (Auditable events).

Application note (from LSPP): For some situations, it is possible that some events cannot be automatically generated. This is usually due to the audit functions not being operational at the time these events occur. Such events need to be included in the administrator guidance documentation, along with recommendations on how manual auditing should be established to cover these events.

Rationale (from LSPP): This component supports O.AUDITING by specifying the detailed, security-relevant events and data that the audit mechanism must be capable of generating and recording. The “basic” level of auditing was selected as best representing the “mainstream” of contemporary audit practices used in the target environments.

Table 5-1 – 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.2	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 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.
FDP_ACC.1	None.	
FDP_ACF.1	All requests to perform an operation on an object covered by the Security Function Policy (SFP).	SMF type 80 record, event code 2.
FDP_ETC.1 (LSPP)	All attempts to export information.	SMF type 80 record, event code 2, for TAPEVOL class.
FDP_ETC.2 (LSPP)	All attempts to export information.	SMF type 80 record, event code 2, for TAPEVOL class.
FDP_ETC.2 (LSPP)	Overriding of human-readable output marking. (Additional)	SMF type 80 record, event code 2, for PSFMPL class.
FDP_IFC.1 (LSPP)	None.	
FDP_IFF.2 (LSPP)	All decisions on requests for information flow.	SMF type 80 record, event code 2, with reason indicating SECLABEL AUDIT.
FDP_ITC.1 (LSPP)	All attempts to import user data, including any security attributes.	SMF type 80 record, event code 2, associated with TAPEVOL profiles.
FDP_ITC.2 (LSPP)	All attempts to import user data, including any security attributes.	SMF type 80, event code 2, associated with TAPEVOL profiles.
FDP_RIP.2	None.	
Note1	None.	
FIA_ATD.1	None.	
FIA_SOS.1	Rejection or acceptance by the TSF of any tested secret.	SMF type 80 record, event code 1, qualifier 1 (password not valid).
FIA_UAU.1	All use of the authentication mechanism.	SMF type 80 record, event code 1, various qualifiers and SMF record type 30 subtypes 1 and 5).
FIA_UAU.7	None.	

Component	Event	Details
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_USB.1	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(1)	All modifications of the values of security attributes.	SMF type 80 record (generated by the RACF commands).
FMT_MSA.1(2) (LSPP)	All modifications of the values of security attributes.	SMF type 80 record (generated by the RACF commands).
FMT_MSA.3(1)	Modifications of the default setting of permissive or restrictive rules. All modifications of the initial value of security attributes.	SMF type 80 record (generated by the RACF commands).
FMT_MSA.3(2) (LSPP)	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_MTD.1(1)	All modifications to the values of TSF data.	SMF type 80 record (generated by the RACF commands).
FMT_MTD.1(2)	All modifications to the values of TSF data.	SMF type 80 record (generated by the RACF commands).
FMT_MTD.1(3)	All modifications to the values of TSF data.	SMF type 80 record (generated by the RACF commands).
FMT_MTD.1(4)	All modifications to the values of TSF data.	SMF type 80 record (generated by the RACF commands).
FMT_REV.1(1)	All attempts to revoke security attributes.	SMF type 80 record (generated by the RACF commands).
FMT_REV.1(2)	All modifications to the values of TSF data.	SMF type 80 record (generated by the RACF commands).
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_AMT.1	Execution of the tests of the underlying machine and the results of the test.	FPT_AMT.1 is satisfied by the TOE environment, so no audit record is produced.
FPT_RVM.1	None.	
FPT_SEP.1	None.	
FPT_STM.1	Changes to the time.	SMF type 80 record for MVS™ operator command SET CLOCK.

Application note: The TOE includes the MVS system management facilities (SMF) component of z/OS, which allows a large number of events to be audited. SMF is not dedicated solely to security auditing, but is used mainly for collecting information that can be used to charge users for the resources they have used. SMF is highly configurable and can be tuned to record events an installation considers to be important.

Application note: Labels are audited in LSPP mode only.

5.1.1.2 User identity association (FAU_GEN.2)

FAU_GEN.2.1 The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

Application note (from LSPP): There are some auditable events that may not be associated with a user, such as failed login attempts. It is acceptable that such events do not include a user identity. In the case of failed login attempts, it is also acceptable to not record the attempted identity in cases where that attempted identity could be misdirected authentication data (when the user may have been out of sync and typed a password in place of a user identifier, for example).

Rationale (from LSPP): O.AUDITING calls for individual accountability (that is, TOE users) whenever security-relevant actions occur. This component requires every auditable event to be associated with an individual user.

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.
User accessing the HTTP server without authenticating themselves are audited with the user ID the server uses for unauthenticated users.

5.1.1.3 Audit review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide ***authorized administrators*** with the ability 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 (from LSPP): The minimum information that must be provided is the same that which is required to be recorded in FAU_GEN.2.

The intent of this requirement is that a tool exists so an administrator can access the audit trail in order to assess it. Exactly what tool is provided is an implementation decision, but it needs to be implemented in a way that allows the administrator to make effective use of the information presented. This requirement is closely tied to FAU_SAR.3 and FAU_SEL.1. It is expected that a single tool will exist within the TSF that will satisfy all of these requirements.

Rationale (from LSPP): This component supports the O.AUDITING and O.MANAGE objectives by providing the administrator with the ability to assess the accountability information that is accumulated by the TOE.

Application note: LSPP has instantiated the term *authorized administrator,* neglecting the fact that a secure system might define additional roles to enhance the security model. In this case, the term *authorized administrator* maps to the AUDITOR role of z/OS or a user with SPECIAL.

5.1.1.4 Restricted audit review (FAU_SAR.2)

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

Application note (from LSPP): By default, authorized administrators may be considered to have been granted read access to the audit records. The TSF may provide a mechanism that allows other users to also read audit records.

Rationale (from LSPP): This component supports the O.AUDITING objective by protecting the audit trail from unauthorized access.

5.1.1.5 Selectable audit review (FAU_SAR.3)

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

a) user identity;

b) subject sensitivity label; (LSPP mode only)

c) object sensitivity label; (LSPP mode only)

d) object type and object name

Application note (from LSPP): The ST must state the additional attributes that audit selectivity may be based upon (object identity and type of event, for example), if any.

Rationale (from LSPP): This component supports both the O.AUDITING and O.MANAGE objectives, by providing a means for the administrator to assess the accountability information associated with an individual user.

5.1.1.6 Selective audit (FAU_SEL.1)

FAU_SEL.1.1 The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes:

a) user identity;

b) subject sensitivity label; (LSPP mode only)

c) object sensitivity label; (LSPP mode only)

d) object type and object name

Application note (from LSPP): The ST must state the additional attributes that audit selectivity may be based upon (object identity and type of event, for example), if any.

Rationale (from LSPP): This component supports both the O.AUDITING and O.MANAGE objectives, by providing a means for the administrator to assess the accountability information associated with an individual user.

Application note: RACF allows to include auditable events based on the criteria defined above.

5.1.1.7 Guarantees of audit data availability (FAU_STG.1)

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

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

Application note (from LSPP): On many systems, in order to reduce the performance impact of audit generation, audit records will be temporarily buffered in memory before they are written to disk. In these cases, it is likely that some of these records will be lost if the operation of the TOE is interrupted by hardware or power failures. The developer needs to document what the likely loss will be and show that it has been minimized.

Rationale (from LSPP): This component supports the O.AUDITING objective by protecting the audit trail from tampering, through deletion or modification of records in it. Further, it ensures that it is as complete as possible.

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

Application note: FAU_STG.1.2 has been modified in accordance with Common Criteria Version 2.3.

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

FAU_STG.3.1 The TSF shall ***generate an alarm to the authorized administrator*** if the audit trail exceeds ***the capacity of the current SMF data set***

Application note (from LSPP): For this component, an *alarm* is to be interpreted as any clear indication to the administrator that the predefined limit has been exceeded. The ST author must state the predefined limit that triggers generation of the alarm. The limit can be stated as an absolute value, or as a value that represents a percentage of audit trail capacity (the audit trail is 75% full, for example). If the limit is adjustable by the authorized administrator, the ST should also incorporate an FMT requirement to manage this function.

Rationale (from LSPP): This component supports the O.AUDITING and O.MANAGE objectives by providing the administrator with a warning that a pending failure due to the exhaustion of space available for audit information.

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. The term *authorized administrator* has been instantiated by LSPP, neglecting the fact that a more finely-grained role model may exist. In this case, the *z/OS operator* role needs to be instantiated.

5.1.1.9 Prevention of audit data loss (FAU_STG.4)

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

Application note (from LSPP): The selection of “preventing” auditable actions if audit storage is exhausted is minimal functionality; providing a range of configurable choices (for example: ignoring auditable actions, changing to a degraded mode, or both) is allowable, as long as “preventing” is one of the choices. If configurable, FMT_MOF.1 should be incorporated into the ST.

Rationale (from LSPP): This component supports the O.AUDITING and O.MANAGE objectives by providing that the audit trail is complete with respect to non-administrative users, while providing administrators with the ability to recover from the situation.

5.1.2 Cryptographic support (FCS)

5.1.2.1 Cryptographic key generation (TLS/SSL: symmetric algorithms) (FCS_CKM.1(1))

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm ***as defined in the SSLv3 and TLSv1 standards*** (***[SSLV3], [TLSV1]***) and specified cryptographic key sizes ***128 bit (AES), 256 bit (AES), 128-bit (RC4) and 168-bit (Triple DES)*** that meet the following: ***generation and exchange of session keys as defined in the SSLv3 [SSLV3] and TLSv1 [TLSV1] standards with the cipher suites defined in FCS_COP.1(2).***

5.1.2.2 Cryptographic key generation (IPSec: symmetric algorithms) (FCS_CKM.1(2))

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm ***product specific*** and specified cryptographic key sizes ***168-bit (Triple DES)*** that meet the following: ***FIPS 46-3.***

5.1.2.3 Cryptographic key distribution (TLS/SSL: RSA public keys) (FCS_CKM.2(1))

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method ***digital certificates for public RSA keys*** that meets the following: ***certificate format as defined in the standard X.509 Version 3***.

Application note: This requirement addresses the exchange of public RSA keys as part of the TLS/SSL client and server authentication. The RSA 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.

5.1.2.4 Cryptographic key distribution (TLS/SSL: symmetric keys) (FCS_CKM.2(2))

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method ***Secure Socket Layer handshake using RSA encrypted exchange of session keys*** that meets the following: ***SSLv3 [SSLV3] and TLSv1 [TLSV1]***.

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

5.1.2.5 Cryptographic key distribution (IPSec: Diffie-Hellman key exchange for symmetric session keys) (FCS_CKM.2(3))

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method ***Diffie-Hellman*** that meets the following: ***Internet Key Exchange standard as defined in IETF RFC 2409.***

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.

5.1.2.6 Cryptographic operation (TLS/SSL: RSA) (FCS_COP.1(1))

FCS_COP.1.1 The TSF shall perform ***digital signature generation and digital signature verification*** in accordance with a specified cryptographic algorithm ***RSA*** and cryptographic key sizes ***1024-bit*** that meet the following: ***SSLv3 [SSLV3], and TLSv1 [TLSV1].***

Application note: This requirement addresses the RSA digital signature generation and verification operations using the RSA algorithm as required by the SSL session establishment protocol (provided a cipher suite including RSA is used). 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 and TLSv1 standard ([SSLV3], [TLSV1]).

5.1.2.7 Cryptographic operation (TLS/SSL: symmetric operations) (FCS_COP.1(2))

FCS_COP.1.1 The TSF shall perform encryption and decryption in accordance with a specified cryptographic algorithm ***AES, RC4, and Triple DES*** and cryptographic key sizes ***128 and 256 bit (AES), 128-bit (RC4), and 168-bit Triple DES*** that meet the following*:* ***SSLv3 and the following cipher suites: SSL_RSA_WITH_RC4_128_SHA and SSL_RSA_TDES_168_SHA as defined in the SSLv3 standard [SSLV3] and TLSv1 and the following cipher suites: TLS_RSA_WITH_RC4_128_SHA, TLS_RSA_3DES_EDE_CBC_SHA,***

TLS_RSA_WITH_AES_128_CBC_SHA and TLS_RSA_WITH_AES_256_CBC_SHA as defined in the TLSv1 standard [TLSV1] and [RFC3268].

5.1.2.8 Cryptographic operation (IPSec: payload encryption) (FCS_COP.1(3))

FCS_COP.1.1 The TSF shall perform encryption and decryption in accordance with a specified cryptographic algorithm ***Triple DES*** and cryptographic key sizes ***168-bit*** that meet the following: ***encryption of the payload of IP packets with tunnel and transport mode as defined in IETF RFC 2406 (IP Encapsulating Security Payload (ESP)).***

5.1.2.9 Cryptographic operation (IPSec: HMAC-SHA) (FCS_COP.1(4))

FCS_COP.1.1 The TSF shall perform encryption and decryption in accordance with a specified cryptographic algorithm ***HMAC-SHA*** and cryptographic key sizes ***160-bit*** that meet the following: ***cryptographically securing the payload and the authentication header of an IP packet as defined in IETF RCF 2406 (IP Encapsulating Security Payload [ESP]) and IETF RFC 2402 (IP Authentication Header) using the specific method for HMAC-SHA as defined in IETF RFC 2404 (The Use of HMAC-SHA-196 within ESP and AH).***

5.1.3 User data protection (FDP)

5.1.3.1 Discretionary access control policy (FDP_ACC.1)

FDP_ACC.1.1 The TSF shall enforce the ***discretionary access control policy*** on ***jobs, started tasks, UNIX processes (whether initiated by rlogin, telnet, HTTP, FTP, or other method) and TSO sessions*** acting on behalf of users, ***and data sets, z/OS UNIX file system objects, z/OS UNIX IPC objects, terminals, devices, volumes, consoles, TCP/IP connections, operator commands, programs, and all operations among subjects and objects covered by the DAC policy***.

Application note (from LSPP): For most systems, there is only one type of subject, usually called a process or task, that needs to be specified in the ST.

Named objects are those objects that are used to share information among subjects acting on the behalf of different users, and for which access to the object can be specified by a name or other identity. Any object that meets this criterion, but is not controlled by the DAC policy, must be justified.

The list of operations covers all operations between the two previous lists. It may consist of a sublist for each subject-named-object pair. Each operation needs to specify which type of access right is needed to perform the operation, for example: read access or write access.

Rationale (from LSPP): This component supports the O.DISCRETIONARY_ACCESS objective by specifying the scope of control for the DAC policy.

5.1.3.2 Discretionary access control functions for non-z/OS UNIX objects (FDP_ACF.1(1))

FDP_ACF.1.1 The TSF shall enforce the ***discretionary access control policy for non-z/OS UNIX resources*** to 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:

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

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 (does not apply for user with the RESTRICTED attribute)

or, if a) is not true,

b) (LSPP 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 and the high-level qualifier of the data set name is identical to the user ID

if c) did not grant access,

d) if the requested type of access is allowed by an access control list (ACL) entry for this particular user

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.

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

f) if the user does not have the RESTRICTED attribute and the requested type of access is granted by the universal access authority (UACC) in the profile protecting the resource or granted by an ACL with ID(*)

if f) did not grant access,

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

if g) did not grant access,

h) if the user has an entry in the conditional access list for the profile that allows the requested type of access and the user meets the condition defined in this conditional access list entry

or, if h) did not grant access,

i) if the user is a member of a group that has an entry in the conditional access list for the profile that allows the requested type of access and the user meets the condition defined in this conditional access list entry. If list-of-groups processing is not in effect, this rule is evaluated only for the current connect group. Otherwise, this rule is evaluated for all groups to which the user is connected.

or, if i) did not grant access,

j) if a conditional access list entry for ID(*) exists with requested type of access, the user does not have the RESTRICTED attribute set and the user satisfies the condition of the conditional access list entry.

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

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

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

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

Application note (from LSPP): A LSPP-conformant TOE is required to implement a DAC policy, but the rules that govern the policy may vary between TOEs; those rules need to be specified in the ST. In completing the rule assignment above, the resulting mechanism must be able to specify access rules that apply to at least any single user. This single user may have a special status, such as the owner of the object. The mechanism must also support specifying access to the membership of at least any single group. Conformant implementations include self/group/public controls and access control lists.

A DAC policy may cover rules on accessing public objects, that is, objects that are readable to all authorized users, but that can only be altered by the TSF or authorized administrators. Specification of these rules should be covered under FDP_ACF.1.3 and FDP_ACF.1.4.

A DAC policy may include exceptions to the basic policy for access by authorized administrators or other forms of special authorization. These rules should be covered under FDP_ACF.1.3.

The ST must list the attributes which are used by the DAC policy for access decisions.

These attributes may include permission bits, access control lists, and object ownership.

A single set of access control attributes may be associated with multiple objects, such as all objects stored on a single disk. The association may also be indirectly bound to the object, such as access control attributes being associated with the name of the object rather than directly to the object itself.

Rationale (from LSPP): This component supports the O.DISCRETIONARY_ACCESS objective by defining the rules that will be enforced by the TSF.

Application note: The rules apply for the TOE in the evaluated configuration. Other configurations may have additional rules that need to be considered.
Further information on the RACF access control mechanisms are provided in Chapter 6, where the possible conditions for conditional access list entries are also defined. In LSPP mode, global access checking may be used to grant READ-type access to resources with a SYSLOW security level only as described in [PMLS].

5.1.3.3 Discretionary access control functions for z/OS UNIX objects (FDP_ACF.1(2))

FDP_ACF.1.1 The TSF shall enforce the ***discretionary access control policy for UNIX objects*** to 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:

- ***read***
- ***write***
- ***execute (ordinary files)***
- ***search (directories)***

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

File system objects are: regular files, directories and symbolic links, device special files, UNIX domain sockets and named pipes (FIFOs)

Access rights for IPC objects are:

- ***read***
- ***write***

Access is defined by permission bits only.

IPC objects are: shared memory segments, message queues, and semaphores

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 (LSPP mode) must allow access and the following algorithm for the discretionary access control must also result in granting access.

File system objects:

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 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.*

b. *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.*

c. *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.*

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

e. *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.*

f. *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.*

g. *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.*

h. *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.*

i. *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.*

j. *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.*

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

IPC objects:

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 (LSPP 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:

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

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: ***none.***

5.1.3.4 Export of unlabeled user data (FDP_ETC.1) (LSPP 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***, outside the TSF Scope of Control (TSC).

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

The TSF shall enforce the following rules when unlabeled user data is exported from the TSC:

a) devices used to export data without security attributes cannot be used to export data with security attributes unless the change in device state is performed manually and is auditable;

b) none.

Application note (from LSPP): An LSPP-conformant TOE must provide protections to data exported outside the control of the TSC through any communication mechanisms that do not provide security attributes along with the actual data. The device or mechanism used to export information must have security attributes that correspond to those of the information being exported. The ability to export information must be allowed under the existing rules that establish the MAC policy of the TOE.

Human-readable hardcopy output must be properly marked with appropriate labels on the top and bottom of pages and on the banner pages at the beginning and end of each output. The ST author must explicitly state the procedures under which this will be accomplished (use of pre-labeled paper is allowable, for example).

The ST author must also explicitly state the rules under which authorized users can designate the security attributes of the mechanisms, or devices, used to export data without security attributes. The ST author must also make it clear that mechanisms, or devices, used to export data without security attributes cannot also be used to export data with security attributes, unless this change in state can only be done manually and is audited.

Single-level input/output devices and single-level communication channels are not required to maintain the sensitivity labels of the information they process.

Rationale (from LSPP): This component supports the O.MANDATORY_ACCESS objective by defining the rules which will be enforced by the TOE.

Application note: Unlabeled data can be exported using tape volumes. Tape volumes that have a single security label can be used to write data to those volumes in accordance with the mandatory access control policy (the security label of the tape must dominate the security label of all data written to the tape). A change in the security label of a tape has to done manually by a system administrator and is audited. A properly authorized system administrator may assign a security label of SYSMULTI to the tape volume, which can then be used for the export of data with its label as required by FDP_ETC.2.

5.1.3.5 Export of labeled user data (FDP_ETC.2) (LSPP mode only)

FDP_ETC.2.1 The TSF shall enforce the ***mandatory access control policy*** when exporting ***labeled*** user data, controlled under the ***MAC policy***, outside the TSC.

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 TSC, 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 TSC:

a) when data is exported in a human-readable or printable form:

- ***the authorized administrator shall be able to specify the printable label that is assigned to the sensitivity label associated with the data.***
- ***each print job shall be marked at the beginning and end with the printable label assigned to the "least upper bound" sensitivity label of all the data exported in the print job.***
- ***each page of printed output shall be marked with the printable label assigned to the "least upper bound" sensitivity label of all the data exported to the page. By default, this marking shall appear on both the top and bottom of each printed page.***

b) devices used to export data with security attributes cannot be used to export data without security attributes unless the change in device state is performed manually and is auditable;

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

d) none.

Application note (from LSPP): The ST author may establish rules that control the export of information from the TSC. These rules must reflect the nature of both the object types and the actual object security attributes. In all cases, the TOE must export the security attributes with the corresponding information.

An LSPP-conformant TOE must only use protocols to export data with security attributes that provide unambiguous pairings of security attributes and the information being exported. Further, the ST author must make it clear that the mechanisms, or devices, used to export data with security attributes cannot be used to export data without security attributes unless this change in state can only be done manually and is audited. In addition, the security attributes must be exported to the same mechanism or device as the information. Also, any change in the security attributes settings of a device must be audited.

Explicit rules must exist in the ST for the export of information that represent hardcopy output. The rules must capture the labeling requirements that must be met for printing labels on the first and last pages, top and bottom of pages, and so forth, and any overriding of printed labels must be audited. Further, the ST must make certain that the external form of the security attributes, or label, must accurately and unambiguously represent the internal label.

Rationale (from LSPP): This component supports the O.MANDATORY_ACCESS objective by defining the rules that will be enforced by the TOE.

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.

5.1.3.6 Mandatory access control policy (FDP_IFC.1) (LSPP mode only)

FDP_IFC.1.1 The TSF shall enforce the ***mandatory access control policy*** on ***jobs, started tasks, UNIX sessions, and TSO sessions acting on behalf of users***, ***data sets, volumes, devices, z/OS UNIX file system objects, z/OS UNIX IPC objects, terminals, TCP/IP connections,*** , ***and all operations among subjects and objects covered by the MAC policy***.

Application note (from LSPP): For most systems, there is only one type of subject, usually called a process or task, which needs to be specified in the ST.

Named objects are those objects that are used to share information among subjects acting on the behalf of different users, and for which access to the object can be specified by a name or other identity. Any object that meets this criterion, but is not controlled by the DAC policy, must be justified.

The ST author must also explicitly list the objects that exist in the TOE. This list must include storage objects. Objects should include data storage resources as well as input/output devices, and so forth.

The operations, listed in the ST, among subjects and objects must explicitly define all relationships between subjects and objects in the TOE, and must be consistent with the list of objects defined in the earlier assignment.

A subject is an entity within the TSC that causes operations to be performed.

Rationale (from LSPP): This component supports the O.MANDATORY_ACCESS objective by specifying the scope of control for the MAC policy.

5.1.3.7 Mandatory access control functions (FDP_IFF.2) (LSPP mode only)

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

a) the sensitivity label of the subject; and

b) the sensitivity label of the object containing the information.

Sensitivity label of subjects and objects shall consist of the following:

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

FDP_IFF.2.2 The TSF shall permit an information flow between a controlled subject and controlled information through 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, 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; the flow of information from the subject to the object is permitted (a write operation);

c) if the sensitivity label of subject A is greater than or equal to the sensitivity label of subject B; the flow of information from subject B to subject A is permitted.

FDP_IFF.2.3 The TSF shall enforce the: ***none***

FDP_IFF.2.4 The TSF shall provide the following: ***none***

FDP_IFF.2.5 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.2.6 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.2.7 The TSF shall enforce the following relationships for any two valid sensitivity labels:

a) there exists an ordering function that, given two valid sensitivity labels, determines if the sensitivity labels are equal, if one sensitivity label is greater than the other, or if the sensitivity labels are incomparable; and

- ***sensitivity labels are equal if the hierarchical level of both labels are equal and the non-hierarchically category sets are equal.***
- ***sensitivity label A is greater than sensitivity label B if one of the following conditions exists:***
  - ***if the hierarchical level of A is greater than the hierarchical level of B, and the non-hierarchical category set of A is equal to the non-hierarchical category set of B.***
  - ***if the hierarchical level of A is equal to the hierarchical level of B, and the non-hierarchical category set of A is a proper super-set of the nonhierarchical category set of B.***

- ***if the hierarchical level of A is greater than the hierarchical level of B, and the non-hierarchical category set of A is a proper[3] superset of the nonhierarchical category set of B.***

▪ ***sensitivity labels are incomparable if they are not equal and neither label is greater than the other.***

b) there exists a "least upper bound" in the set of sensitivity labels, such that, given any two valid sensitivity labels, there is a valid sensitivity label that is greater than or equal to the two valid sensitivity labels; and

c) there exists a "greatest lower bound" in the set of the sensitivity labels, such that, given any two valid sensitivity labels, there is a valid sensitivity label that is not greater than the two valid sensitivity labels.

Application note (from LSPP): The terms *security attribute* and *information flow control security attribute* refer to the sensitivity labels of subjects and objects.

An LSPP-conformant TOE should support at least 16 site-definable hierarchical levels and 64 site-definable non-hierarchical categories. The implementation of sensitivity labels does not need to store labels in a format that has the components of the label explicitly instantiated, but may use some form of tag that maps to a level and category set.

Rationale (from LSPP): This component supports the O.MANDATORY_ACCESS objective by defining the rules that will be enforced by the TOE.

5.1.3.8 Import of unlabeled user data (FDP_ITC.1) (LSPP mode only)

FDP_ITC.1.1 The TSF shall enforce the ***mandatory access control policy*** when importing unlabeled user data, controlled under the ***MAC policy***, from outside the TSC.

FDP_ITC.1.2 The TSF shall ignore any security attributes associated with the ***unlabeled*** user data when imported from outside the TSC.

FDP_ITC.1.3 The TSF shall enforce the following rules when importing ***unlabeled*** user data controlled under the MAC policy from outside the TSC:

a) devices used to import data without security attributes cannot be used to import data with security attributes unless the change in device state is performed manually and is auditable.

b) none.

Application note (from LSPP): The LSPP-conformant TOE must provide protections for data imported from outside the control of the TSC through functions that do not provide reliable security attributes along with the actual data. The imported data must be assigned a sensitivity label that will be used to enforce the MAC policy. Further, the ability for a subject to import information must be controlled under the existing rules that establish the MAC policy of the TOE.

The ST author must explicitly state the rules under which authorized users can designate the security attributes of the mechanisms, or devices, used to import data without security attributes, and any attribute change must be audited. The ST author must also make it clear that mechanisms, or devices, used to import data without security attributes cannot also be used to import data with security attributes unless this change in state can only be done manually and is audited.

Rationale (from LSPP): This component supports the O.MANDATORY_ACCESS objective by defining the rules which will be enforced by the TOE.

[3] The word "proper" in this rule has been taken over from LSPP, but is definitively wrong in this rule. Because the hierarchical level of A is already greater than the hierarchical level of B, A is greater than B even if the sets of categories of A and B are identical.

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.

5.1.3.9 Import of labeled user data (FDP_ITC.2) (LSPP mode only)

FDP_ITC.2.1 The TSF shall enforce the ***mandatory access control policy*** when importing ***labeled*** user data, controlled under the ***MAC policy***, from outside the TSC.

FDP_ITC.2.2 The TSF shall use the 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 security attributes and the ***labeled*** user data received.

FDP_ITC.2.4 The TSF shall ensure that interpretation of the 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*** from outside the TSC:

a) devices used to import data with security attributes cannot be used to import data without security attributes unless the change in device state is performed manually and is auditable;

b) none.

Application note (from LSPP): The ST author must provide for the protection of data imported from outside the control of the TSC through any mechanisms that provide security attributes along with the information being imported. The security attributes received along with the data must accurately represent the security attributes of the data with which they are associated.

The ST author must make it clear that the mechanisms, or devices, used to import data with security attributes cannot be used to import data without security attributes unless this change in state can only be done manually and is audited. Also, any change in the security attributes of a device must be audited.

Rationale (from LSPP): This component supports the O.MANDATORY_ACCESS objective by defining the rules which will be enforced by the TOE.

c) sensitivity label, consisting of the following:

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

Application note (from LSPP): An LSPP-conformant TOE should support at least 16 site-definable hierarchical levels and 64 site-definable non-hierarchical categories. The implementation of sensitivity labels does not need to store labels in a format which has the components of the label explicitly instantiated, but may use some form of tag which maps to a level and category set.

Application note: See the application note on FDP_ETC.2 for export of labeled data.

5.1.3.10 Object 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.

Application note (from LSPP): This requirement applies to all resources governed by or used by the TSF; it includes resources used to store data and attributes. It also includes the encrypted representation of information.

Clearing the information content of resources on deallocation from objects is sufficient to satisfy this requirement, if unallocated resources will not accumulate new information until they are allocated again.

Rationale (from LSPP): This component supports the O.RESIDUAL_INFORMATION objective.

5.1.3.11 Subject residual information protection (Note 1)

NOTE 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.

Application note (from LSPP): This requirement applies to all resources governed by or used by the TSF; it includes resources used to store data and attributes. It also includes the encrypted representation of information.

Clearing the information content of resources on deallocation from subjects is sufficient to satisfy this requirement, if unallocated resources will not accumulate new information until they are allocated again.

Rationale (from LSPP): This component supports the O.RESIDUAL_INFORMATION objective.

5.1.3.12 Basic data exchange confidentiality (FDP_UCT.1)

FDP_UCT.1.1 The TSF shall enforce the ***discretionary access control policy and (in LSPP mode) mandatory access control policy*** to be able to ***transmit and receive*** objects in a manner protected from unauthorized disclosure.

Application note: Confidentiality of data during transmission is ensured when the secured protocols TLS, SSL, or IPSec are used. User processes are still bound by the mandatory and discretionary access control policy with respect to the data they are able to transfer.

5.1.3.13 Data exchange integrity (FDP_UIT.1)

FDP_UIT.1.1 The TSF shall enforce the ***discretionary access control policy and (in LSPP mode) mandatory access control policy*** to be able to ***transmit and receive*** user data in a manner protected from ***modification and insertion*** errors.

FDP_UIT.1.2 The TSF shall be able to determine on receipt of user data, whether ***modification or insertion*** has occurred.

Application note: Integrity of data during transmission is ensured when the secured protocols TLS, SSL, or IPSec are used. User processes are still bound by the mandatory and discretionary access control policy with respect to the data they are able to transfer.

5.1.4 Identification and authentication (FIA)

5.1.4.1 User attribute definition (FIA_ATD.1)

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

a) user identifier;
b) group memberships;
c) authentication data;
d) user clearances; (in LSPP mode)
e) security-relevant roles; and

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

Application note (from LSPP): The specified attributes are those that are required by the TSF to enforce the DAC policy, the generation of audit records, and proper identification and authentication of users. The user identity must be uniquely associated with a single individual user.

Group membership may be expressed in a number of ways: a list per user specifying to which groups the user belongs, a list per group which includes which users are members, or implicit association between certain user identities and certain groups.

A TOE may have two forms of user and group identities: a text form and a numeric form. In these cases, there must be unique mapping between the representations.

Rationale (from LSPP): This component supports the O.AUTHORIZATION and O.DISCRETIONARY_ACCESS objectives by providing the TSF with the information about users needed to enforce the TSP.

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

5.1.4.2 Strength of authentication data (FIA_SOS.1)

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

a) for each attempt to use the authentication mechanism, the probability that a random attempt will succeed is less than one in 1,000,000;

b) for multiple attempts to use the authentication mechanism during a one minute period, the probability that a random attempt during that minute will succeed is less than one in 100,000; and

c) any feedback given during an attempt to use the authentication mechanism will not reduce the probability below the above metrics.

Application note (from LSPP): The method of authentication is unspecified by the LSPP, but must be specified in a ST. The method which is used must be shown to have low probability that authentication data can be forged or guessed. For example, if a password mechanism is used a set of metrics needs to be specified and may include such things as minimum length of the password, maximum lifetime of a password, and the subjecting of possible passwords to dictionary attacks. The strength of whatever mechanism implemented must be subjected to a strength of function analysis.

Rationale (from LSPP): This component supports the O.AUTHORIZATION objective by providing an authentication mechanism with a reasonable degree of certainty that only authorized users may access the TOE.

5.1.4.3 Authentication (FIA_UAU.1)

FIA_UAU.1.1 The TSF shall allow ***no execution of a program or command for any user on behalf of the user to be performed before the user is authenticated except for pseudo-users for started procedures (started tasks) and access to the HTTP server restricted to the***

***functions and resources accessible to the pseudo user the server assigns to unauthenticated users*.**

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

Application note (from LSPP): The ST must specify the actions which are allowed by an unauthenticated user. The allowed actions should be limited to those things which aid an authorized user in gaining access to the TOE. This could include help facilities or the ability to send a message to authorized administrators.

Rationale (from LSPP): This component supports the O.AUTHORIZATION objective by specifying what actions unauthenticated users may perform.

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 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.

5.1.4.4 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 (from LSPP): Obscured feedback implies the TSF does not produce a visible display of any authentication data entered by a user, such as through a keyboard (echo the password on the terminal, for example). It is acceptable that some indication of progress be returned instead, such as a period returned for each character sent.

Some forms of input, such as card input based batch jobs, may contain human readable user passwords. The administrator and user guidance documentation for the product must explain the risks in placing passwords on such input and must suggest procedures to mitigate that risk.

Rationale (from LSPP): This component supports the O.AUTHORIZATION objective. Individual accountability cannot be maintained if the individual's authentication data, in any form, is compromised.

Application note: When entered during LOGIN or other command that initiates a session or when entered for a password change the user has the option to use the commands in a way that prohibits passwords to be displayed. Passwords a user enters using a

jobcard will be supressed in any output of the JCL statements to prohibit that the password can be obtained by anybody reading the output.

5.1.4.5 Identification (FIA_UID.1)

FIA_UID.1.1 The TSF shall allow ***access to the HTTP server restricted to the functions and resources accessible to the pseudo user the server assigns to unauthenticated users*** 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 the behalf of that user.

Application note (from LSPP): The ST must specify the actions which are allowed to an unidentified user. The allowed actions should be limited to those things which aid an authorized user in gaining access to the TOE. This could include help facilities or the ability to send messages to authorized administrators.

The method of identification is unspecified by this PP, but should be specified in a ST and it should specify how this relates to user identifiers maintained by the TSF.

Rationale (from LSPP): This component supports the O.AUTHORIZATION objective by specifying what actions unidentified users may perform.

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.

5.1.4.6 User-subject binding (FIA_USB.1)

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

a) The user identity that is associated with auditable events;
b) The user identity (or identities) used to enforce the discretionary access control policy;
c) The group membership or memberships used to enforce the discretionary access control policy;
d) In LSPP mode: The sensitivity label used to enforce the mandatory access control policy, which consists of the following:
- ***A hierarchical level; and***
- ***A set of non-hierarchical categories.***

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

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

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

c) A user that connects to the HTTP server will be bound to the user ID the installation has assigned for the unauthenticated user of the server until the user is successfully identified and authenticated using his user ID password combination.

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:***

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 for this user.

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

c) 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 , has the SURROGAT authority for the new UID or has BPX.SUPERUSER authority allowing him to switch to UID 0 without supplying a password.

d) 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.

e) When executing a program from a file with the set-user-ID-on-execution bit (S_ISUID) set, the subject's EUID 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 EGID is set to the group ID of the file being executed;

Application note (from LSPP): The DAC policy and audit generation require that each subject acting on the behalf of users have a user identity associated with the subject. This identity is normally the one used at the time of identification to the system.

The DAC policy enforced by the TSF may include provisions for making access decisions based on a user identity which differs from the one used during identification.

The ST must state, in 5.3.6.3, how this alternate identity is associated with a subject and justify why the individual user associated with this alternate identity is not compromised by the mechanism used to implement it.

Depending on the TSF's implementation of group membership, the associations between a subject and groups may be explicit at the time of identification or implicit in a relationship between user and group identifiers. The ST must specify this association.

Like user identification, an alternate group mechanism may exist, and parallel requirements apply.

Application note (LSPP mode): The "clearance range" supported by the TOE is a discrete set of allowable labels for each user. Because the set of security labels is only a partially-ordered list, the definition of a "range" makes no sense.

Rationale (from LSPP): This component supports the O.DISCRETIONARY_ACCESS and O.AUDITING objectives by binding user identities to subjects acting on their behalf.

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.

5.1.5 Security management (FMT)

5.1.5.1 Management of *object security* attributes (FMT_MSA.1(1))

FMT_MSA.1.1 The TSF shall enforce the ***discretionary access control policy*** to restrict the ability to ***modify*** the ***access control attributes associated with a named object*** to ***users with the SPECIAL attribute or the appropriate group-SPECIAL attribute, users who have ALTER authority to the object and the owner of the resource profile of the named object (for non-UNIX objects) and the owner of the named object and a user with z/OS UNIX superuser privilege (for z/OS UNIX objects).***

Application note (from LSPP): The ST must state the components of the access rights that may be modified, and must state any restrictions that may exist for a type of authorized user and the components of the access rights that the user is allowed to modify.

The ability to modify access rights must be restricted in that a user having access rights to a named object does not have the ability to modify those access rights unless granted the right to do so. This restriction may be explicit, based on the object ownership, or based on a set of object hierarchy rules.

Rationale (from LSPP): This component supports the O.DISCRETIONARY_ACCESS objective by providing the means by which the security attributes of objects are managed by a site.

5.1.5.2 Management of *object security* attributes for MAC (FMT_MSA.1(2)) (LSPP mode only)

FMT_MSA.1.1 The TSF shall enforce the ***mandatory access control policy*** to restrict the ability to ***modify*** the ***sensitivity label associated with an object*** to ***users with the SPECIAL attribute***.

Rationale: This component supports the O.MANDATORY_ACCESS objective by providing the means by which the security attributes of objects are managed by a site.

5.1.5.3 Secure security attributes (FMT_MSA.2)

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for security attributes.

Application note: This requirement is included as a dependency from the security functional requirements FCS_CKM.1, FCS_CKM.2, and FCS_COP.1. The assessment with respect to this requirement in the evaluation of this TOE does not include any assessment of the cryptographic strength of the keys generated or used. Instead, the assessment with respect to this requirement just includes an assessment that the TOE protects those keys from unauthorized access, disclosure, or tampering. This requirement is not applied to other security attributes, because there it is up to the system administrator to assign values to those attributes and there is no way for the TOE to decide if the values assigned are “secure” within the intended operational purpose of the TOE. For example, administrators should know about the potential consequences when they assign labels to objects or when they assign security attributes to users.

5.1.5.4 Static attribute initialization (FMT_MSA.3(1))

FMT_MSA.3.1 The TSF shall enforce the ***discretionary access control policy*** to provide ***restrictive*** default values for security attributes that are used to enforce the ***discretionary access control policy***.

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.

Application note (from LSPP): An LSPP-conformant TOE must provide protection by default for all objects at creation time. This may be done through the enforcing of a restrictive default access control on newly-created objects or by requiring the user to explicitly specify the desired access controls on the object at its creation. In either case, there shall be no window of vulnerability through which unauthorized access may be gained to newly-created objects.

Rationale (from LSPP): This component supports the O.DISCRETIONARY_ACCESS objective by requiring that objects are properly protected starting from the instant that they are created.

Application note: Because the option to assign a property other than "restrictive" or "permissive" was only introduced with final interpretation RI#202, the authors of LSPP and CAPP have selected "restrictive", but allowed an authorized administrator to override those default values. In reality, most systems will neither define the "restrictive" nor the "permissive" case as the default value, but the default values will be defined such that they match the intended operational policy in the best way. This also applies to 5.1.5.5.

5.1.5.5 Static attribute initialization for MAC (FMT_MSA.3(2)) (LSPP mode only)

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

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.

Application note (from LSPP): An LSPP-conformant TOE must provide protection by default for all objects at creation time. This may be done through the enforcing of a restrictive default access control on newly-created objects or by requiring the user to explicitly specify the desired access controls on the object at its creation. In either case, there shall be no window of vulnerability through which unauthorized access may be gained to newly-created objects.

Rationale: This component supports the O.MANDATORY_ACCESS objective by requiring that objects are properly protected with a security label starting from the instant that they are created.

Application note: LSPP has just iterated the element FMT_MSA.3.1 twice and not the component FMT_MSA.3 as a whole. Since the authors of this Security Target felt that this is not consistent with the requirements of the CC when having multiple iterations of a component, this Security Target defines two iterations of FMT_MSA.3, one for discretionary and one for mandatory access control. The rationale of the second iteration now mentions the support for O.MANDATORY_ACCESS, which the authors of LSPP have forgotten in their rationale.

5.1.5.6 Management of ***the audit trail*** (FMT_MTD.1(1))

FMT_MTD.1.1 The TSF shall restrict the ability to ***create, delete, and clear the audit trail*** to ***authorized administrators***.

Application note (from LSPP): The selection of "create, delete, and clear" functions for audit trail management reflect common management functions. These functions should be considered generic; any other audit administration functions that are critical to the management of a particular audit mechanism implementation should be specified in the ST.

Rationale (from LSPP): The component supports the O.AUDITING and O.MANAGE objectives by ensuring that the accountability information is not compromised by destruction of the audit trail.

Application note: The term *authorized administrators* has been instantiated by LSPP and has been included for this reason in this Security Target. z/OS allows a more finely-grained control of the management of the audit trail, which is explained in Chapter 6. In this case, the roles are *auditor* and *z/OS operator*.

5.1.5.7 Management of *audited events* (FMT_MTD.1(2))

FMT_MTD.1.1 The TSF shall restrict the ability to ***modify or observe the set of audited events*** to ***authorized administrators***.

Application note (from LSPP): The set of audited events are the subset of auditable events which will be audited by the TSF. The term *set* is used loosely here and refers to the total collection of possible ways to control which audit records get generated; this could be by type of record, identity of user, identity of object, and so forth.

An important aspect of auditing is that users should not be able to effect which of their actions are audited, and therefore must not have control over or knowledge of the selection of an event for auditing.

Rationale (from LSPP): This component supports the O.AUDITING and O.MANAGE objectives by providing the administrator with the ability to control the degree to which accountability is generated.

Application note: The management of audited events in z/OS is controlled by users in the role of auditors and by the owner of the profile for events related to a profile. The owner of a profile is viewed as an authorized administrator for that profile.

5.1.5.8 Management of *user attributes* (FMT_MTD.1(3))

FMT_MTD.1.1 The TSF shall restrict the ability to ***initialize and modify the user security attributes, other than authentication data***, to ***authorized administrators***.

Application note (from LSPP): This component only applies to security attributes that are used to maintain the TSP. Other user attributes may be specified in the ST, but control of those attributes are not within the scope of the LSPP.

Rationale (from LSPP): This component supports the O.MANAGE objective by providing the administrator with the means to manage who are authorized users and what attributes are associated with each user.

Application note: The term *authorized administrators* has been included from the instantiation made in LSPP. z/OS allows for a more finely-grained management of user attributes by users with the SPECIAL attribute, users with CLAUTH attribute for the USER class and, for users that are members of a specific group, by users with the group-SPECIAL attribute for this group. This is explained in more detail in Chapter 6.

5.1.5.9 Management of *authentication data* (FMT_MTD.1(4))

FMT_MTD.1.1 The TSF shall restrict the ability to ***initialize the authentication data*** to ***authorized administrators***.

FMT_MTD.1.1 The TSF shall restrict the ability to ***modify the authentication data*** to ***the following:***

a) authorized administrators; and

b) users authorized to modify their own authentication data

Application note (from LSPP): User authentication data refers to information that users must provide to authenticate themselves to the TSF. Examples include passwords, personal identification numbers, and fingerprint profiles. User authentication data does not include the user's identity. The ST must specify the authentication mechanism that makes use of the user authentication data to verify a user's identity.

This component does not require that users be authorized to modify their own authentication information; it only states that it is permissible. It is not necessary that requests to modify authentication data require reauthentication of the requester's identity at the time of the request.

Rationale (from LSPP): This component supports the O.AUTHORIZATION and O.MANAGE objectives by ensuring integrity and confidentiality of authentication data.

Application note: Users with the SPECIAL attribute can modify a user's password.

5.1.5.10 Management of *cryptographic keys* (FMT_MTD.1(5))

FMT_MTD.1.1 The TSF shall restrict the ability to ***import cryptographic keys*** to ***authorized administrators***.

FMT_MTD.1.1 The TSF shall restrict the ability to ***modify cryptograqphic keys*** to ***authorized administrators.***

5.1.5.11 Management of *network configuration* (FMT_MTD.1(6))

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

5.1.5.12 Revocation *of user attributes* (FMT_REV.1(1))

FMT_REV.1.1 The TSF shall restrict the ability to revoke security attributes associated with the ***users*** within the TSC to ***authorized administrators***.

FMT_REV.1.2 The TSF shall enforce the rules:

a) the immediate revocation of security-relevant authorizations; and

b) none.

Application note (from LSPP): Many security-relevant authorizations could have serious consequences if misused, so an immediate revocation method must exist, although it does not need to be the usual method. For example, the usual method may be editing the trusted user's profile, but the change doesn't take effect until the user logs off and logs back on. The method for immediate revocation might be to edit the trusted user's profile and "force" the trusted user to log off. The immediate method must be specified in the ST and in administrator guidance. In a distributed environment, the developer must provide a description of how the "immediate" aspect of this requirement is met.

Rationale (from LSPP): This component supports the O.MANAGE objective by controlling access to data and functions that are not generally available to all users.

Application note: User attributes are evaluated when they are used. Revocation of such security relevant authorizations as the user's role or security attributes are therefore immediate, because even if the attribute is revoked when the user is active in a TSO session or a job, or as a z/OS UNIX user, the next time he used his authorization,

RACF performs the checks against the up-to-date RACF database. Note that revocation is restricted to users with defined roles who are allowed to perform the revocation of specific attributes. See Chapter 6 for details.

5.1.5.13 Revocation *of object attributes* (FMT_REV.1(2))

FMT_REV.1.1 The TSF shall restrict the ability to revoke security attributes associated with ***objects*** within the TSC to ***users authorized to modify the security attributes by the discretionary access control policy or (in LSPP mode) the mandatory access control policy***.

FMT_REV.1.2 The TSF shall enforce the rules:

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

b) LSPP mode only: the rules of the mandatory access control policy are enforced on all future operations; and

c) none.

Application note (from LSPP): The DAC policy may include immediate revocation (for example: Multics immediately revokes access to segments) or delayed revocation (most UNIX systems do not revoke access to already opened files, for example). The DAC access rights are considered to have been revoked when all subsequent access control decisions by the TSF use the new access control information. It is not required that every operation on an object make an explicit access control decision as long as a previous access control decision was made to permit that operation. It is sufficient that the developer clearly states in guidance documentation how revocation is enforced.

Rationale (from LSPP): This component supports the O.DISCRETIONARY_ACCESS objective by providing that specified access control attributes are enforced at some fixed point in time.

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.

5.1.5.14 Specification of management functions (FMT_SMF.1)

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

- **object security attributes management**
- **user security attribute management**
- **authentication data management**
- **audit event management**
- **key management for cryptographic keys**

5.1.5.15 Security management roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles:

a) ***authorized administrator[4];***

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

c) ***in LSPP mode: users authorized by the mandatory access control policy to modify object security attributes;***

d) ***users authorized to modify their own authentication data; and***

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

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

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

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

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

j) ***z/OS pseudo-user (protected user IDs used for executing defined started tasks as well as the surrogate-user ID used by the HTTP server for unauthenticated users)***

k) ***z/OS UNIX superuser***

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

Application note (from LSPP): An LSPP-conformant TOE only needs to support a single administrative role, referred to as the authorized administrator. If a TOE implements multiple independent roles, the ST should refine the use of the term authorized administrators to specify which roles fulfill which requirements.

The LSPP specifies a number of functions that are required of or restricted to an authorized administrator, but there may be additional functions that are specific to the TOE. This would include any additional function that would undermine the proper operation of the TSF. Examples of such functions include: the ability to access certain system resources, such as tape drives or vector processors, the ability to manipulate the printer queues, the ability to run real-time programs, and the overriding of sensitivity labels on printed output.

Rationale (from LSPP): This component supports the O.MANAGE objective.

5.1.6 Protection of the TOE security functions (FPT)

5.1.6.1 Reference mediation (FPT_RVM.1)

FPT_RVM.1.1 The TSF shall ensure that the TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

Application note (from LSPP): This element does not imply that there must be a reference monitor. Rather this requires that the TSF validates all actions between subjects and objects that require policy enforcement.

[4] LSPP uses the term *authorized administrators* in a number of SFRs. Literally, this would prohibit a more finely-grained role model as implemented in z/OS, allowing to bind some of the rights defined in the set of SFR to roles that only have some limited administration capability. Because such a finely-grained administration model is generally viewed as superior to a model with only one single “superuser”, such as an administration model, the authors of this Security Target have taken the freedom to define a more finely-grained administration model. Allowing the ability to define additional roles, but fixing the assignments of privileges and administration tasks to one already-defined role, is regarded as a failure of LSPP.

Rationale (from LSPP): This component supports O.ENFORCEMENT objective by ensuring that the TSP is not being bypassed.

5.1.6.2 Domain separation (FPT_SEP.1)

FPT_SEP.1.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

Application note (from LSPP): This component does not imply a particular implementation of a TOE. The implementation needs to exhibit properties that the code and the data upon which TSF relies are not alterable in ways that would compromise the TSF and that observation of TSF data would not result in failure of the TSF to perform its job. This could be done either by hardware mechanisms or hardware architecture. Possible implementations include multi-state CPUs that support multiple task spaces and independent nodes within a distributed architecture.

The second element can also be met in a variety of ways also, including CPU support for separate address spaces, separate hardware components, or entirely in software.

The latter is likely in a layered application such as a graphic user interface system that maintains separate subjects.

Rationale (from LSPP): This component supports O.ENFORCEMENT objectives by ensuring that a TSF exists within the TOE and that it can reliably carry out its functions.

5.1.6.3 Reliable time stamps (FPT_STM.1)

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps for its own use.

Application note (from LSPP):: The generation of audit records depends on having a correct date and time. The ST needs to specify the degree of accuracy that must be maintained in order to maintain useful information for audit records.

Rationale (from LSPP): This component supports the O.AUDITING objective by ensuring that accountability information is accurate.

5.1.6.4 Inter-TSF basic TSF data consistency (FPT_TDC.1) (LSPP mode only)

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: This requirement is required as a dependency from FDP_ITC.2. Although FDP_ITC.2 is included in LSPP, this dependency has been neither resolved nor has been any rationale provided as to why this dependency does not apply for LSPP. Because the authors of this Security Target do not have access to the evaluation technical report of the LSPP evaluation, the authors of this Security Target don't know if there was a reason for not resolving this dependency. The authors of this Security Target would have expected in any case that the rationale in LSPP provide an explanation why the dependency has not been resolved.

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 or when UNIX file system objects with their

extended attributes are exported from one system and imported into another system. In order to do this, at least the definition of the security labels between the systems involved have to be identical. In addition, 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.

5.1.7 Trusted path/channel

5.1.7.1 Inter-TSF trusted channel (FTP_ITC.1)

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and a remote 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 or disclosure.

FTP_ITC.1.2 The TSF shall permit ***the TSF or the remote trusted IT product*** to initiate communication by way of the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication by way of the trusted channel for ***when the communication uses the SSLv3, TLSv1, or IPSec protocols offered by TOE services***.

5.2 TOE security assurance requirements

The target evaluation assurance level for the product is EAL4 [CC] augmented by ALC_FLR.1.

5.3 Security requirements for the IT environment

The only 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:

5.3.1.1 Subset access control (FDP_ACC.1)

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

5.3.1.2 Security-attribute-based access control (FDP_ACF.1)

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

FDP_ACF.1.2 The TSF 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**.

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 FPT_SEP.1 and FPT_RVM.1 in section 5.1 of this Security Target. 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..

FDP_ACF.1.3 The TSF 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 TSF shall explicitly deny access of subjects to objects based on the **following rule: none.**

5.3.1.3 Static attribute initialization (FMT_MSA.3)

FMT_MSA.3.1 The TSF 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 TSF 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.

5.3.1.4 Abstract machine testing (FPT_AMT.1)

FPT_AMT.1.1 The TSF shall run a suite of tests ***periodically during normal operation and at the request of IBM field service personnel*** to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

Application note (from LSPP): In general, this component refers to the proper operation of the hardware platform on which a TOE is running. The test suite needs to cover only aspects of the hardware on which the TSF relies to implement required functions, including domain separation. If a failure of some aspect of the hardware would not result in the TSF compromising the functions it performs, testing of that aspect is not required.

Rationale: This component supports the OE.HW_SEP objective by demonstrating that the underlying mechanisms are working as expected.

Rationale: In contrast to the PP specification, abstract machine testing has been put into the TOE environment, because the TOE’s underlying hardware provides extensive testing of the abstract machine and intercepts possible failures at a level that cannot be observed or tested from within the TOE. The reader is referred to chapter 11 of [ZARCH] for a description of the continuous self-test and error reporting function of the underlying hardware platform. Figure 11-3 in [ZARCH] lists the possible interrupt codes for the machine check interrupt. Those codes and the malfunction they indicate are described in detail in the text following the figure. Testing the correct functionality of the underlying abstract machine by software running on this machine therefore

makes no sense, since this software will not be able to detect an error the underlying hardware has not detected and reported already.

5.4 Security requirements for the non-IT environment

All the security objectives for the TOE environment address physical protection of the TOE or procedures that need to be obeyed by administrative users.

6. TOE summary specification

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 derived from the Controlled Access Protection Profile (CAPP) and the Labeled Security Protection Profile (LSPP) with the extensions defined in Chapter 5 of this document are described in this chapter.

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

6.1 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 LSPP mode: mandatory access control based on security attributes of subjects and objects
4. Management functions to administer auditing, discretionary access control, and (in LSPP mode) mandatory access control, as well as users and groups with their related attributes
5. An audit trail for security relevant events
6. Secure communication
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 Communication Server, which is responsible for network communication using SNA- or IP-based protocols
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 and user interfaces
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.

The TOE also supports UNIX terminals through telnet, rlogin, and other TCP/IP-based network protocols.

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.

More information on how the TOE identifies, manages, and protects authorized programs can be found in Section 6.6.

6.1.1 Main trusted subsystems of the evaluated configuration

Some programs are started with authorization (see also section 6.8.3) during system startup. Those include the Job Entry Subsystem (JES2), the Time Sharing Option Extensions (TSO/E) subsystem, the Communication Subsystem (CS), and the z/OS UNIX System Services.

6.1.1.1 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 needs 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 LSPP 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.

6.1.1.2 Time Sharing Option Extensions (TSO/E)

TSO/E is the 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 in which the user is identified and authenticated. After successful authentication, a user can issue TSO commands, execute programs, or submit jobs to JES2.

TSO/E also uses RACF for authentication of users and to control access of users to terminal devices and data sets.

6.1.1.3 Communication Server

z/OS provides networking functions with the Communication 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.

6.1.1.4 z/OS UNIX System Services

z/OS also provides users and programs with a UNIX environment. Users who are also defined as UNIX users in RACF (they must have a UID and GID for the default group assigned in the OMVS segment in the RACF user profile) can use this environment to operate in a UNIX shell environment and use UNIX commands and program library interfaces.

RACF is used by the UNIX system services to:

- authenticate users
- control access to UNIX files, directories, and z/OS data sets
- 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 LSPP 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 LSPP 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 LSPP mode) the mandatory access control rules defined by RACF.

In LSPP 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.

6.1.1.5 Print Services Facility

z/OS provides printing functions with JES2 and PSF. The PSF subsystem provides support for printing output on a large variety of print peripherals. In LSPP mode, PSF must be used in conjunction with JES2 to enforce printing of security labels on all pages of print jobs containing labelled data.

6.2 Identification and authentication

6.2.1 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
- As a user connecting to the HTTP server

In all cases (except for the HTTP server that allows access of unauthenticated users as described later), users are identified and authenticated by a user ID and password combination (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 TSO (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.1.3). They must only be started from trusted data sets.

When authenticating a user, RACF checks:

- If the user is defined to RACF (IA.1.4)
- If the user has supplied a valid password and a valid group name. Otherwise, the user's default group is selected. Note that telnet, rlogin, rsh, rexec, http and ftp do not allow a user to select a group. If those functions are used for login, the user will always get his default group assigned for the session. In LSPP 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 range of labels the user is allowed to use. 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.1.5)
- If the user has a valid UID and his default group has a valid GID (if UNIX services are requested) (IA.1.6)
- If the user ID is in REVOKE status, which prevents a RACF-defined user from entering the system at all or entering the system with certain groups (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 when he exceeds the number of failed login attempts.
- If the user in the TSO environment can use the system on this day and at this time of the day (an installation can impose restrictions). This is checked only when using a terminal from a defined set. This does not apply to telnet, rlogin, rsh, rexec, http, ftp, or to batch jobs (IA.1.8)
- 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.1.9)
- If the user is authorized to use the application (if specified) (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.1.11). In LSPP mode, the surrogate user who submits the job must have read access to the security label under which the job runs (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.1.13).

6.2.2 Passwords

In RACF, the user selects his own password and only the user knows his own password. If a password needs to be reset, the security administrator will reset the password (IA.2.1). When the system administrator follows the rules for the evaluated configuration, this new password should be in an expired state, thus forcing the user to enter a new password on the first logon (IA.2.2). Only when a new user ID for a pseudo-user is created that is not a protected user ID, the initial password may not be marked as expired (IA.2.3).

A system administrator can set a variety of rules for forming valid passwords using the SETROPTS command (for system-wide settings) or with 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.2.4).

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.2.5)
- Maximum password lifetime (INTERVAL suboption) (IA.2.6) and minimum password change time (MINCHANGE option) (IA.2.V1R7.1)
- Number of passwords from the user's password history that are not allowed for a new password (HISTORY suboption) (IA.2.7)
- Maximum number of consecutive failed authentication attempts until the REVOKE attribute is set in the user's profile (REVOKE suboption) (IA.2.8)
- Differentiate between upper- and lowercase characters with the PASSWORD(MIXEDCASE) option (IA.2.V1R7.2)
- Type of character for each character position of a password. Possible types are (IA.2.9):
  - ALPHA
  - ALPHANUM (which includes also the special characters $, # and @)
  - VOWEL
  - NOVOWEL
  - CONSONANT
  - NUMERIC
  - MIXEDCONSONANT
  - MIXEDVOVEL
  - 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.

  Passwords are not displayed when entered at a TSO terminal as part of the login process (IA.2.10), when using the TOE supplied clients for ftp or telnet, rlogin, rsh and rexec (IA.2.11), when entered in order to change a password (IA.2.12), or when the content of a jobcard is displayed as part of a job's output (IA.2.13).

6.2.3 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.3.1).

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, DFSMS 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.3.2). 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.3.4). 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)

6.2.3.1 Assigning RACF user IDs to started procedures

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

6.2.3.2 Protected user IDs

The user IDs that an administrator assigns to started procedures should have the PROTECTED attribute unless the started procedure is required to have a user ID with a password defined. Protected user IDs are user IDs that have both the NOPASSWORD and NOOIDCARD attributes (IA.3.5). 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.3.6).

6.2.3.3 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.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.V1R7.2).

The HTTP server also provides a function to identify and authenticate users using their user ID and password (IA.3.V1R7.3). Once authenticated successfully the access rights of the authenticated user are checked when the HTTP server attempts to access protected resources on behalf of the user (IA.3.V1R7.4). The HTTP server uses RACF for user identification and authentication (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.V1R7.6).

6.2.4 Special handling in z/OS UNIX

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.4.1).

2. Handling of the su command

   The handling of the su command depends on the existence of specific profiles in RACF.

   Case 1: 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.4.2), or if the original user ID has read access to the BPX.SRV.newuser resource profile in the SURROGAT class (IA.4.3).

   Note that, unlike in most other UNIX systems, this also applies to subjects running with UID 0.

   Case 2: 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.4.4)

b) the original user ID has read access to the BPX.SUPERUSER resource profile in the FACILITY class (IA.4.5).

The shell started by the su command inherits the security label of the user who issued the command (LSPP mode only) (IA.4.6). The new user must be authorized to the inherited security label or the su command fails (LSPP mode only) (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.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.V1R7.1) nor by executing a program that has the setuid or setgid bit set (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.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.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.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 process is executing with UID 0 or the current subject has the trusted or privileged attribute (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.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.V1R7.7).
- The target user ID has been successfully authenticated by the password service (IA.4.V1R7.8) or has SURROGAT authority to the new user ID (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.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.V1R7.11).

The setgid service does not change any supplementary group IDs of the calling process (IA.4.V1R7.12).

User identification and authentication are also performed by the telnet, rlogin, rsh, rexec, ftp, and (optionally) http z/OS UNIX services as described in Section 6.2.1.

6.2.4.1 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.V1R7.13). This identity change can only occur if the target user ID has an OMVS segment defined (IA.4.V1R7.14).

Any program loaded into an address space that requires daemon level authority must be defined to program control. If the BPX.DAEMON FACILTY 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
- _password()

(IA.4.V1R7.15)

Daemon authority is required only when a program does a setuid(), seteuid(), setreuid(), or spawn() user ID to change the current UID without first having issued a __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.V1R7.16). If a program comes from a controlled library and knows the target UID's password, it can change the UID without having daemon authority (IA.4.V1R7.17).

6.3 Access control

6.3.1 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 LSPP 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.

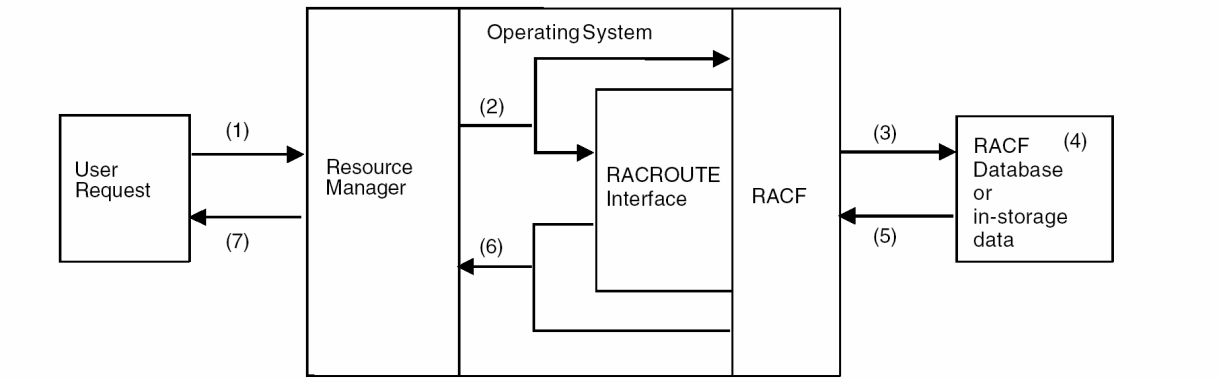

Figure 1: RACF and its relationship to the operating system

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

The z/OS component responsible for managing the resource calls the RACF component using the internal interface to RACF (mainly the RACROUTE interface) to check the access rights of the user that initiated the user request and passes the name and type of the resource and the requested type of access to RACF (AC.1.1). 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.1.2).

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.1.3). 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.1.4). 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.1.5).

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.1.6). 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.

6.3.2 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

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.

6.3.2.1 Data sets

6.3.2.1.1 Standard data set naming conventions

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

If an implementation team has chosen to define data set profiles under the standard RACF naming conventions, one 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.2.1).

6.3.2.1.2 Table-driven data set naming conventions

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

- Supply a qualifier to be used as the high-level qualifier for authorization checking (AC.2.3)
- Convert data set names to RACF naming convention form for RACF use (AC.2.4)
- Convert names in RACF form to the installation’s format for external display (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.2.6)
- 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.2.7). An installation can use the table to selectively rearrange data set names to “fit” the RACF convention without actually changing those names.

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

If some of the data sets in an installation have names that consist of a single qualifier, one can still RACF-protect those data sets (AC.2.8). 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.2.9). 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.2.10). All

SMF log records and all messages from RACF contain the RACF-modified version of the data set name (AC.2.11).

6.3.2.1.4 Protecting user data sets

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

- In general, all RACF-defined users can protect their own data sets (AC.2.12)
- 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.2.13)
  - 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.2.14)
  - 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.2.15).

    In this case, the user can create a discrete profile:
    - Through ADSP (AC.2.16)
    - By specifying the PROTECT operand on the TSO ALLOCATE command that creates the data set (AC.2.17)
    - By specifying the PROTECT=YES OR SECMODEL= profile-name operands on the JCL DD statement that creates the data set (AC.2.18)

6.3.2.1.5 Protecting group data sets

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

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

6.3.2.1.6 Controlling the creation of new data sets

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

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

- To add entries to the catalog, users need authority to create the data set as specified below and UPDATE authority to the catalog (AC.2.23)
- To delete entries from the catalog, users need ALTER authority to the protecting profile or to the catalog (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 protected by an existing generic profile and the user does not have ADSP (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.2.26)
- The data set name is not covered by an existing generic profile and the user does not have ADSP and the data set is protected by the Global Access check table. (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.2.28)
- The user has the OPERATIONS attribute. If the user has the group-OPERATIONS attribute (that is, the user is connected to a group with the OPERATIONS attribute), the high-level qualifier of the new data set must be the ID of a user who is within the scope of that group (AC.2.29)

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

- The data set name is protected by an existing generic profile and the user does not have ADSP.
  The creation is allowed if at least one of the following is true:
  - The user has ALTER authority to the data set through the generic profile or global access checking (AC.2.30)
  - The user has CREATE authority in the group (AC.2.31)
- The data set name is not covered by an existing generic profile and the user does not have ADSP (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.2.33)
- The user has the OPERATIONS attribute except when both of the following are true:
  1. The user is connected to the group with less than CREATE authority (AC.2.34)
  2. The user has less than ALTER access to the data set if it protected by a generic profile (AC.2.35)

  If the user has the group-OPERATIONS attribute (that is, the user is connected to a superior group with the OPERATIONS attribute), the group for which the new data set is being created must be within the scope of that superior group (AC.2.36).

6.3.2.1.7 Data set profile ownership

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

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.2.38).

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 LSPP mode) MAC policy rules (AC.2.39).

6.3.2.2 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.2.40). 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.2.41).

Tape volumes are protected by profiles in the TAPEVOL class (AC.2.42).

6.3.2.3 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.2.43). 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.

6.3.2.4 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.2.45). 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.2.46). 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.2.47). If this parameter is NONE, a user can not use such terminals to log in (AC.2.48). If the parameter is READ, a user can use those terminals to log in (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.2.50).

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.2.51).

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 (LSPP mode only) (AC.2.52).

6.3.2.5 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.2.53).
- 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.zone-name" where system-name is the name of the TOE image, stack-name is the job name of the particular stack and zone-name is the name of the security zone containing the IP address (AC.2.54).
- 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-name*", where *system-name* is the name of the TOE image, *stack-name* is the job name of the particular stack, and *SAF-name* is the name configured on the Port statement (AC.2.55).

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.2.56):

- 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.

TCP/IP performs additional access control when the RACF option MLACTIVE is set (in LSPP 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 LSPP 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.2.57).
- In LSPP 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.2.58).
- In LSPP 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.2.59).
- In LSPP 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.2.60). 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.2.61).

The Communications Server subsystem of the TOE provides numerous commands and applications. For LSPP mode: There are documented restrictions on usage and configuration of these when RACF option MLACTIVE is set. The FTP and TN3270 Server applications may be configured to use SystemSSL services to provide end-to-end data channels that are authenticated and encrypted (AC.2.62). Application Transparent Transport Layer Security may be configured to use SystemSSL service to provide end-to-end data channels that are authenticated and encrypted for most TCP applications.

The Communication Server element of the TOE provides support for IPSec-protected communication in accordance with RFCs 2401 through 2406 and 2410, 3947 and 3948 as well as the key management RFCs 2407 trough 2409. (AC.2.63). It also provides the IKE application that negotiates IPSec security association parameters with communication peers.

6.3.2.6 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.2.64). Note that if the class is active and a command is not protected by a profile it is not allowed to be executed.

6.3.2.7 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.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.V1R7.2).

Program control (as desribed 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.V1R7.3). An empty value or any other value than 'BASIC' or 'ENHANCED' will result in the ENHANCED-WARNING mode (AC.2.V1R7.4). If the IRR.PGMSECURITY profile is not defined, BASIC mode is used (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.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.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.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.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.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.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.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.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.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.V1R7.15)

With program control enabled, z/OS provides the ability to allow users access to data sets which they are not allowed to access directly by using program-controlled programs (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.V1R7.17).

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

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

6.3.2.8 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.V1R7.24). Using RACF, the use of system consoles can be controlled (AC.2.V1R7.25).

6.3.2.9 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 LSPP 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 LSPP mode) security labels are stored and managed as extended attributes of the file system object and are not stored in the RACF database (AC.2.65). RACF is still involved when an access decision is made to a UNIX file system object (AC.2.66). The UNIX System Services subsystem of the TOE extracts the permission bits, access control list entries and (in LSPP mode) the security label from the file system object as well as the effective user ID and (in LSPP 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.2.67).

Besides the access control lists and (in LSPP 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.2.68). In LSPP mode, a z/OS UNIX superuser is also bound by the mandatory access control rules when accessing z/OS UNIX files (AC.2.69).

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

- if the user has an effective user ID of zero (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.2.71)
- if the user is neither the owner or creator of the IPC object but is a member of the IPC object's group and the requested type of access is allowed by the group related permission bits (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 group and the access is allowed by the other related permission bits (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.2.74).

6.3.3 Mandatory access control (LSPP mode only)

Label based mandatory access control is supported by z/OS. User profiles contain one or more SECLABEL names, 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.3.1). He 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.3.2)
- 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.3.3)
- 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.3.4)

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.V1R7.3).

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

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

Printers (as examples of devices) and terminals can be restricted to the security labels allowed to be used with them (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.3.14). The exact text of this label can be defined during system configuration (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.3.16).

A user can define the security label of a session when he performs his TSO login or when submitting a batch job (AC.3.17). 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.3.18). 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.3.19). 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.3.20). 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.3.21).

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.3.22). 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.3.23).

6.3.4 Discretionary access control

Discretionary access control to RACF resources is controlled by the user, group, and resource profiles stored and managed by RACF. See the sections above on the different profiles for details on what is stored in those profiles. RACF controls the types of access to all non-UNIX resources. The access types are ordered hierarchically, an access type listed higher in the list implies all the access types lower in this list (except for NONE access). The full semantics of each access type are defined by the resource manager.

- **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.4.1).

  ALTER does not allow users to change the owner of the profile using the ALTDSD command (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.4.3).
  When specified in a generic profile, ALTER gives users no authority over the profile itself (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.4.5).

  For non-VSAM data sets, CONTROL is equivalent to UPDATE (AC.4.6)

- **UPDATE**

  Allows users to read from, copy from, or write to the data set (AC.4.7). UPDATE does not, however, authorize a user to delete, rename, move, or scratch the data set (AC.4.8)

- **READ**

  Allows users to access the data set for reading only (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.4.10)

- **NONE**

  The specified user or group is not permitted to access the resource or list the profile (AC.4.11)

These access types can be defined per user, group or generic for all users not addressed specifically by a user or group access entry (“universal access”) (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.4.13). 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.4.14).

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.4.15)
- the user must be logged on to a defined console (AC.4.16)
- the batch job requesting access must have been submitted from a defined JES input device (AC.4.17)
- the user must have entered the system from a defined network port (AC.4.18)

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.4.19). If more than one generic profiles would apply, z/OS always chooses the most specific profile applicable based on a matching algorithm (AC4.20).

6.3.4.1 Algorithm to check 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 using the effective user and group ID respectively.

- (Step performed in LSPP mode only) Access to the file system object must be allowed by the mandatory access control function. If not, access is denied (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.4.22)
- 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.4.23)
- If the user does not have search permission to all directories in the path of the file system object, access is denied (AC.4.24)
- 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.4.25). If the UID matches the file owner UID and the owner bits do not allow the requested access, go to Step 15 (AC.4.26)
- 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.4.27). Otherwise, if the ACL for the UID exists, but does not allow access, go to Step 14 (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.4.29)

- 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.4.30). If not, then the next ACL entry is checked until there are no more entries (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.4.32)
- 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.4.33). If not, then the next ACL entry is checked until there are no more entries (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.4.35). If the GID, or any of the supplemental GIDs, matched the file owner GID, then processing continues with Step 15 (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.4.37)
- 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.4.38)
- The file's "other" permission bits are checked. If the "other" bits allow the requested access, then access is granted (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.4.40)
- 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.4.41)
- Access is denied, if none of the above steps has explicitly granted access (AC.4.42)

6.3.4.2 Algorithm to check for access to non-UNIX 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 LSPP 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.4.43). 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. 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.4.44). 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.4.45). 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)
3. 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.4.46).

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.) RACF determines which group to use according to the following rules:

- If list-of-groups processing is not in effect, RACF uses only the user's current connect group (AC.4.47)
- 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.4.48). (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 7 (AC.4.49) (conditional access list checking). This prevents access based on ID(*), UACC, or the OPERATIONS attribute

4. 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.4.50)
   - Not sufficient to allow the requested access, RACF continues processing at Step 6 (AC.4.51) (OPERATIONS attribute checking)
5. 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.4.52)
6. 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.4.53)
7. RACF checks the user's access authority in the conditional access list specified with WHEN(TERMINAL), WHEN(CONSOLE), or WHEN(JESINPUT). If the user is in the list, if the user meets the specified condition (such as logged on at the specified terminal), and if the specified access authority is sufficient to allow access, RACF grants the request (AC.4.54)
8. 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), 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.4.55). If none of the user's groups has sufficient authority, RACF continues with the next step
9. If a user ID of * is found on the conditional access list specified with WHEN(TERMINAL), WHEN(CONSOLE), 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.4.56)
10. 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.4.57).

    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.4.58)

11. 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.4.59). If the group is in the list and if the specified access authority is NONE, RACF denies the request (AC.4.60)

12. 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.4.61)
13. For access to uncataloged data sets, if SETROPTS CATDSNS is in effect, and none of the following is true:
    - 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 data set is cataloged in a STEPCAT, and the user has READ access to FACILITY resource ICHUSERCAT;
    - The user has access to FACILITY resource ICHUNCAT.dataset-name (truncated to 39 characters total, if needed);
    - The user has the SPECIAL attribute

    RACF denies the request (AC.4.62).

14. For the DATASET class, if no profile is found and the SETROPTS PROTECTALL(FAILURES) option is in effect, RACF denies the request (AC.4.63)
15. 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.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.V1R7.2).

6.4 Communication security

z/OS provides networking functions with the Communication Subsystem. 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 communication server uses RACF to protect access of users to the following resources:

- the TCP/IP stack in general (CS.1.1)
- TCP and UDP ports (CS.1.2)
- IP addresses (CS.1.3)

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.

- 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 encrypted using the configured algorithms. See Section 6.3.2.5 for details of the standards supported and the associated security claim.
- SSL / TLS layer to set up a trusted channel to another trusted IT product:
  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 Triple DES with 168-bit key length, AES with either 128 or 256-bit key length, as well as RC4 with 128-bit key length. Application Transparent Transport Layer Security (AT-TLS) supports the use of all cipher suites supported by System SSL (CS.1.4). The TN3270 and FTP protocols are enabled to use System SSL and can be tunneled through SSL/TLS to establish a trusted channel to another trusted IT product that also implements this protocol (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.V1R7.1).

  AT-TLS can be configured in the PROFILE.TCPIP configuration file (see Section 6.5.4).

In addition, the communication server provides the following application protocols that include user authentication using RACF:

- FTP (user authentication is required) (CS.1.6)
- telnet (CS.1.7)
- rlogin, rsh, and rexec (CS.1.8)
- TN3270 (CS.1.9)
- HTTP (user authentication is an option) (CS.1.V1R7.2)

Access control to resources used within a FTP, HTTP or telnet session is also performed using RACF (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.1.11).

6.5 Security management

6.5.1 User and group management

6.5.1.1 Definition of users and groups

z/OS users and groups are defined in RACF. To create a 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 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.

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

- the SPECIAL role as a general system administrator (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.1.2)

- 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.1.3)
- to modify the attribute of a user: the CLAUTH attribute for the user class (SM.1.4). Note that only the CLAUTH and NOCLAUTH attribute can be changed (SM.1.5)

RACF 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.1.6).

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.1.7)
- Connect and remove users from the group (SM.1.8)
- Delegate and change group authorities and set the default UACC for all new resources belonging to members of the group (SM.1.9)
- Modify, list, and delete the group profile (SM.1.10)
- Define, delete, and list the names of the subgroups under the group (SM.1.11)
- Specify the group terminal option (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.1.13).

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

6.5.1.2 User profiles

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

Name	Description
USERID	User's identification (a maximum of 8 characters).
NAME	User's name (not security relevant, because the user is allowed to change his name).
OWNER	Owner of the user's profile.
DFLTGRP	User's default group. (Note: A user may specify, at login time, any group he or she is connected to as the current default group. This does not change the DFLTGRP value in the profile.)
AUTHORITY	User's authority in the default group (use, create, connect, join).
PASSWORD	User's password. The user ID is DES-encrypted using the password (padded with blanks) as a key.
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.

Name	Description
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 LSPP 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

6.5.1.3 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.

6.5.1.4 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.1.14).

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.

6.5.1.4.1 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.1.15). 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.1.16)
- list and define RACF general options (except options related to auditing) (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.1.18). 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.1.19).

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.1.20).

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.1.21). 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.1.22)
- CMDVIOL or NOCMDVIOL (SM.1.23)
- LOGOPTIONS (for each profile class) (SM.1.24)
- OPERAUDIT or NOOPERAUDIT (SM.1.25)
- SAUDIT or NOSAUDIT (SM.1.26)
- SECLABELAUDIT or NOSECLABELAUDIT (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.1.28). He can not modify or set audit related attributes that operate system-wide (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.1.30). The OPERATIONS attribute can also be assigned at the group level (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 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 SUUROGAT-submitted batch jobs or by started procedures defined in the STARTED class or the started procedures table.

6.5.1.4.2 Attributes

CLAUTH

If a user has the CLAUTH attribute in a class, RACF allows the user to define profiles in that class (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.1.33).

A user with the CLAUTH(USER) attribute can add and modify users except for setting or modifying the following attributes:

- SPECIAL or NOSPECIAL (SM.1.34)
- AUDITOR or NOAUDITOR (SM.1.35)
- OPERATIONS or NOOPERATIONS (SM.1.36)

REVOKE

A user can be prevented from entering the system by assigning the REVOKE attribute (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.1.38). This applies also to accounts that have never been active (SM.1.39).

6.5.2 Resource management

RACF makes access decisions based on information stored in profiles. 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.

6.5.2.1 Data set profiles

A data set profile within RACF contains (among other data not relevant for the security functions defined in this Security Target) the following:

Name	Description
Profile name	Name of the data set profile
GENERIC, MODEL, or TAPE	Indicates if it is a generic, a model or a tape data set profile
OWNER	Owner of the data set profile
NOTIFY	The TSO user who is to be notified whenever RACF uses this profile to deny access to a data set
UACC	The universal access authority for the data set or data sets protected by the profile
AUDIT	The type of auditing to be performed for the data set or data sets protected by the profile
CATEGORY	The security categories to be assigned to the data set or data sets protected by the profile
SECLABEL	The security label of the data set or data sets protected by the profile (evaluated in LSPP mode only)
SECLEVEL	The security level of the data set or data sets protected by the profile (evaluated in LSPP 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

6.5.2.2 General resource profiles

Other protected resources defined in this Security Target (except the z/OS UNIX file system objects and z/OS UNIX IPC objects) are protected by general resource profiles that contains the resource class and the resource attributes. As with profiles for z/OS data sets, an access control list with entries defining the access types for individual users and / or groups can be defined for each such resource profile. The semantics of the individual access rights are defined by the resource manager responsible for the management of the resources protected by such a profile. Different resource classes may have different resource managers responsible for the protection and management of the resources within the class.

The structure of a general resource profile is defined in the following table (omitting fields that are not relevant for the Security Policy as defined in this Security Target:

Name	Description
Class name	Name of the resource class the profile belongs to
Profile name	Name of the generic resource profile
OWNER(user ID or groupname)	The owner of the profile
NOTIFY	The user who is to be notified whenever RACF uses this profile to deny access to a resource
UACC	The universal access authority for the resource or resources protected by the profile
AUDIT	The type of auditing to be performed for the resource or resources protected by the profile
FROM	The name of a profile that is to be used as a model
FCLASS	The class of the model profile
FGENERIC	A setting that indicates that the model profile name is to be treated as a generic name
FVOLUME	The volume that is to be used to locate the model profile
CATEGORY	The security categories to be assigned to the resource or resources protected by the profile (evaluated in LSPP mode only)
SECLABEL	The security label of the resource or resources protected by the profile (evaluated in LSPP mode only)

Name	Description
SECLEVEL	The security level of the resource or resources protected by the profile (evaluated in LSPP mode only)
LEVEL	An installation-defined level
SINGLEDSN	The tape volume protected by this profile can contain only one data set (TAPEVOL 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)
TIMEZONE	The time zone in which a terminal resides (TERMINAL class only)
WHEN	The times when the terminal or terminals protected by the profile can be used to access the system (TERMINAL class only)

6.5.2.3 z/OS UNIX file system resources

z/OS UNIX file system resources are not protected by RACF profiles but by permission bits and extended attributes stored in the z/OS UNIX file system. The evaluated configuration supports two different z/OS UNIX file system types: zFS and HFS. A file system for both file system types is always implemented in a single z/OS data set.

In the case of zFS the extended attributes also contain the security label (evaluated in LSPP 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 LSPP 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 section 6.2.3.8 for details of the access control strategy for z/OS UNIX file system objects.

6.5.2.4 RACF classes

For the evaluation the protection of the following classes are considered:

CONSOLE
Controlling access to operator consoles. Also, conditional access to other resources for commands originating from an operator console. (SM.2.1)

DASDVOL
DASD volumes. See also the GDASDVOL class. (SM.2.2)

DEVICES
Used to control access to unit record devices, teleprocessing or communication devices, and graphic devices. (SM.2.3)

DIRAUTH (used in LSPP 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.2.4)

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.

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.2.5)

GTERMINL
Resource group class for TERMINAL class. (SM.2.6)

JESINPUT
Port of entry class to control which JES2 input devices a user can use to submit batch work to the system. (SM.2.7)

JESJOBS
Controlling the submission and cancellation of jobs by job name. (SM.2.8)

JESSPOOL
Controlling access to job data sets on the JES spool (that is, SYSIN and SYSOUT data sets). (SM.2.9)

NODES
Controlling the following on MVS systems:

- Whether jobs are allowed to enter the system from other JES2 nodes (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.2.11)

OPERCMDS
Controlling who can issue operator commands (for example, JES and MVS, and operator commands). (SM.2.12)

PROGRAM
Controlled programs (load modules). (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.2.14)

SDSF
Controls the use of authorized commands in the System Display and Search Facility (SDSF). (SM.2.15)

SECDATA (used in LSPP mode only)
Security classification of users and data (security levels and security categories). (SM.2.16)

SECLABEL (used in LSPP mode only)
If security labels are used, and, if so, their definitions. (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.2.18)

SERVER
Controlling the server's ability to register with the daemon. (SM.2.19)

SMESSAGE
Controlling to which users a user can send messages (TSO only). (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.2.21)

TAPEVOL
Tape volumes. (SM.2.22)

TERMINAL
Terminals (TSO). SM.2.23)

TSOPROC
TSO logon procedures. (SM.2.24)

UNIXPRIV
Contains profiles that are used to grant z/OS UNIX privileges. (SM.2.25)

VTAMAPPL
Controlling who can open ACBs from non-APF authorized programs. This prevents programs from counterfeiting login screens. (SM.2.26)

WRITER
Controlling the use of JES writers. (SM.2.27)

6.5.3 RACF configuration and management

6.5.3.1 Configuring RACF with the SETROPTS command

The SPECIAL and AUDITOR roles can define system wide-options of RACF with the SETROPTS command. This command can be used (among other actions) to:

- Choose the resource classes that RACF is to protect. (SM.3.1)
- Set the universal access authority (UACC) for otherwise undefined terminals. (SM.3.2)
- Specify logging of certain RACF commands and events. (SM.3.3)
- Permit list-of-groups access checking. (SM.3.4)
- Display options currently in effect. (SM.3.5)
- Enable or disable generic profile checking on either a class-by-class or system-wide level. (SM.3.6)
- Establish password syntax rules. (SM.3.7)
- Activate password processing for checking previous passwords, limit invalid password attempts, and warn of password expiration. (SM.3.8)
- Control global access checking for selected individual resources or generic names with selected generalized access rules. (SM.3.9)
- Set the passwords for authorizing use of the RVARY command. (SM.3.10)
- Initiate refreshing of in-storage generic profile lists and global access checking tables. (SM.3.11)
- Enable or disable shared profiles through RACLIST processing for general resources. (SM.3.12)
- Activate auditing of access attempts to RACF-protected resources based on installation-defined security levels. (SM.3.13)

- Activate enhanced generic naming. (SM.3.14)
- Activate profile modeling for GDG, group, and user data sets. (SM.3.15)
- Activate protection for data sets with single-level names. (SM.3.16)
- Control logging of real data set names. (SM.3.17)
- Control the job entry subsystem (JES) options. (SM.3.18)
- Activate tape data set protection. (SM.3.19)
- Control whether or not data sets must be RACF-protected. (SM.3.20)
- Control the erasure of scratched DASD data sets. (SM.3.21)
- Activate program control. (SM.3.22)
- Control whether a profile creator's user ID is automatically added to the profile's access list. (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(*), GRPLIST, PROTECTALL(FAILURES), CLASSACT (TEMPDSN), JES(BATCHALLRACF). In LSPP mode the following options need to be set in addition: MLACTIVE(FAILURES), MLFSOBJ(ACTIVE), MLIPCOBJ(ACTIVE), MLS(FAILURES), MLSTABLE, SECLABELCONTROL. (SM.3.24). Additional parameter for the PASSWORD operand need to be set to define the password policy. See section 6.2.2 of this Security Target.

6.5.3.2 RACF commands

The administration of RACF is performed by a set of commands. Users need the required authorities or roles to issue those commands or specific parameter of those commands. 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.3.25)
- ADDUSER, ALTUSER, DELUSER
  Commands to define a new user profile, modify an existing user profile or delete a user profile (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.3.27)
- CONNECT, REMOVE
  Command to connect a user to or remove a user from a group (SM.3.28)
- LISTGROUP, LISTUSER, LISTDSD
  Commands to list user, group or z/OS data set profiles (SM.3.29)
- RDEFINE, RALTER, RDELETE
  Commands to define, modify or delete a general resource profile (SM.3.30)
- RLIST
  Command to list a general resource profile (SM.3.31)
- PASSWORD
  Command to specify a user's password (SM.3.32)
- PERMIT
  Command to maintain the access list of a resource profile (SM.3.33)

- SETROPTS
  Command to set specific RACF options (see section above for details) (SM.3.34)

Other RACF commands not related to the Security Policy as defined in this Security Target exist, but are not mentioned here.

6.5.3.3 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.3.35). 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.3.36). 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.3.37).

6.5.4 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.
- 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.

6.6 Auditing

6.6.1 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.1.1). 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.1.2).

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, 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 LSPP mode only)

Each record contains further data specific to the event code and qualifier (AU.1.3).

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 (LSPP mode) or specific object sensitivity label (LSPP mode) can be extracted from the audit trail (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.1.5). Tools exist that transfer the audit data into human readable format (AU.1.6).

6.6.2 Protection of the audit trail

SMF writes audit records into 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.

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.2.2). 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. 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.2.3). 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.2.4). 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.2.5). If the TOE is configured according to the administrative guidance, the system will halt if no buffer space is left (AU.2.6).

6.6.3 Audit configuration and management

Within the system configuration it needs to be decided, which SMF records shall be generated by z/OS. Three record types (type 80, 81, and 83) are dedicated to RACF and are the most important ones for security. Which events are actually recorded with those records can be configured by a user with the AUDITOR attribute in his RACF user profile (AU.3.1). 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.3.2). 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.3.3).

The system can be configured to send certain audit messages to the security console to immediately alert operators of detected policy violations (AU.3.4)

6.7 Object reuse

z/OS provides explicit object reuse functionality for the following objects:

- memory objects
- z/OS data sets
- z/OS tape volumes
- z/OS UNIX file system objects
- z/OS UNIX IPC objects

In the evaluated configuration, z/OS ensures that those 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.1.1). DASD data from z/OS data sets is erased when the data is released when the erase-on-scratch option is active (OR.1.2). 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.v18.1). z/OS UNIX file system objects and z/OS UNIX IPC objects are cleared before they are made accessible to a new subject (OR.1.3).

6.8 TOE self-protection

6.8.1 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.

6.8.1.1 Processor features

The zSeries or z9 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.1.1).

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.1.2). 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.1.3). 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.1.4).

6.8.1.2 Abstract machine modes of operation

z/OS may execute in one of the three modes:

- native hardware mode
- logical partition mode
- VM guest mode

In all of those cases, z/OS operates on an abstract machine that implements the z/Architecture.

In native hardware mode, z/OS has full control of all the resources of the physical processor.

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.

6.8.2 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.

6.8.3 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.3.1)
- 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.3.2)
- The authorization bit is set in the Job Step Control Block (JSCB) under which the program is executing (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.3.4). 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.3.5)

- By starting a started task from an authorized library using the operator START command (SP.3.6)
- By starting an authorized program from a zFS file system (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.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.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.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.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.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.

6.8.3.1 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.3.7)

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.3.8). 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.3.9).

6.9 Assurance measures

The following table provides an overview, how the assurance measures of EAL4 augmented by ALC_FLR.1 are met by z/OS.

Table 6-1: Mapping Assurance Components to Assurance Measures

Assurance Component	Documentation describing how the requirements are met
ACM_AUT.1	All configuration management of z/OS source code uses automated CM systems
ACM_CAP.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.
ACM_SCP.2	Source code, generated binaries, documentation, test plan, test cases and test results are all maintained under configuration management.
ADO_DEL.2	z/OS is delivered through sales channels controlled by IBM.
ADO_IGS.1	Guidance for installation and system configuration is provided in a number of documents that are part of the zSeries z/OS Collection.
ADV_FSP.2	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 subset of the implementation representation includes all modules that implement TSFI and all modules that call those modules, allowing the evaluators to trace the flow from a TSFI to the enforcement of the security functional requirement.
ADV_HLD.2	A high-level design of the security functions of z/OS is provided. 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.
ADV_LLD.1	IBM provides dedicated low-level design documentation for all subsystems of the TOE related to security functions.
ADV_RCR.1	The correspondence information is provided in the form of a spreadsheet showing the correspondence between the TOE summary specification and the functional specification and the functional specification and the high level design.
ADV_SPM.1	An informal security policy model is provided by the developer defining the security policy of the TOE in an informal way.
AGD_ADM.1	A number of documents exist that provide guidance for the system administrator. This includes guides for the overall system configuration and management as well as the configuration and management for individual components of z/OS. Especially for the configuration and management of RACF a System Administrator Guide exists, that describes and explains in detail the administration commands and parameters.

Assurance Component	Documentation describing how the requirements are met
AGD_USR.1	User guidance is provided in a number of documents related to the individual components of z/OS. Those documents explain in detail the security functions a normal user can use and manage.
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.1	z/OS Development within IBM has a well-defined system for reporting flaws and tracing the status of the corrective actions for those flaws.
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.
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_MSU.2	A Misuse Analysis will be provided by the sponsor.
AVA_SOF.1	The Strength of Function Analysis will be provided for the mechanism based on permutational or probabilistic algorithms as part of the developer's vulnerability analysis document.
AVA_VLA.2	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.

6.10 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. For this reason the security functional requirement FPT_AMT.1 of CAPP and LSPP is already satisfied by the underlying hardware as part of the IT environment.

7. Protection Profile claims

7.1 Reference

This Security Target claims conformance with the “Labeled Security Protection Profile” (LSPP), Version 1.b, 8 October 1999, and the “Controlled Access Protection Profile” (CAPP) Version 1.d, 8 October 1999. Both Protection Profiles were developed by the “Information System Security Organization” of the National Security Agency of the United States of America.

Both Protection Profiles are listed on the TPEP web site of NSA as “Certified Protection Profile”.

7.2 Tailoring and additions

Security functional requirements have been refined where required by the Protection Profile.

The following security objective for the TOE has been added:

- O.COMPROT

This objective addresses the ability of the TOE to set up a trusted channel to another trusted IT product as expressed with security functional requirements FTP_ITC.1, FDP_UTC.1 and FDP_UIT.1, which have been included as an extension to the requirements defined in CAPP and LSPP.

The following security objectives for the TOE environment have been added:

- OE.HW_SEP
- OE.CLASSIFICATION (LSPP mode)

These objectives are required to cover the specific assumptions and organization security policies addressing the TOE environment. All objectives are related to physical and procedural security measures and therefore address the TOE non-IT environment. LSPP mode: Note that OE.CLASSIFICATION has been added to address the assumptions A.SENSITIVITY and A.CLEARANCE listed in LSPP in Chapter 3, but were not addressed in the rationale section provided in LSPP.

The assumption A.CONNECT of CAPP and LSPP has been modified to reflect the capability of the TOE to protect communication to other systems outside secured premises.

In addition, the Security Target has added security requirements for the IT environment (the underlying abstract machine) to define the requirement for the underlying processor to provide the functions to implement effective separation of the TSF from untrusted software. This includes the requirements FDP_ACC.1, FDP_ACF.1 and FMT_MSA.3 for the IT environment.

The assurance requirements of the Protection Profiles are those defined in the Evaluation Assurance Level EAL3 of the Common Criteria, augmented by ADV_SPM.1 for LSPP. This Security Target specifies an Evaluation Assurance Level EAL 4 augmented by ALC_FLR.1, which is specific for this Security Target. Because the Evaluation Assurance Levels in the Common Criteria define a hierarchy, with EAL4 already including ADV_SPM.1, all assurance requirements of the Protection Profiles are included in this Security Target. ALC_FLR.1, which has been added to the assurance requirements defined in the CAPP and LSPP, has no dependency on any other security functional requirement or security assurance requirement and is therefore an augmentation that has no effect on the security functional requirements or security assurance requirements stated in the Protection Profile.

Security functional requirement (FMT_SMF.1) has been added to those defined in LSPP and CAPP. The reason is CC version 2.3 (released after CAPP and LSPP), where the new family FMT_SMF is defined and dependencies from FMT_MSA.1 and FMT_MTD.1 to the new component FMT_SMF.1 have been added. To

resolve those new dependencies, FMT_SMF.1 has been added as a security functional requirement in addition to those defined in LSPP and CAPP.

Security functional requirements for cryptographic operations and for a trusted channel to another trusted IT product have been added for the additional security function of the TOE with respect to protected communication through the SSL/TLS protocol. These include FCS_CKM.1, FCS_CKM.2 (multiple instantiations) and FCS_COP.1 (multiple instantiations). These requirements address the cryptographic function for the protection of the communication links using the SSL/TLS protocol. Requirements FDP_UCT.1 and FDP_UIT.1 address the ability to protect communication links for confidentiality and integrity when using SSL/TLS. In addition FMT_MSA.2 was included, because it is required as a dependency from the requirements in the FCS class. FPT_TDC.1 was included due to a dependency from FDP_ITC.2, which is included in LSPP. The authors of the LSPP neither resolved this dependency nor provided any argument in the rationale as to why this dependency does not need to be resolved. FTP_ITC.1 was added to reflect the requirement for a trusted path to another trusted IT product that can be established through the SSL /TLS protocol implemented in the TOE.

FMT_MSA.1 and FMT_MSA.3 have been changed into two iterations each. In LSPP, both SFRs have “inline iterations”, which are not allowed in the CC model. The now correctly iterated SFRs have been titled “Management of *object security* attributes (FMT_MSA.1(1))” and “Management of *object security* attributes for MAC (FMT_MSA.1(2))” for the LSPP SFR FMT_MSA.1, and “Static attribute initialization (FMT_MSA.3(1))” and “Static attribute initialization for MAC (FMT_MSA.3(2))” for the LSPP SFR FMT_MSA.3, respectively.

FPT_AMT.1 has been moved into the TOE environment, because the required functionality is provided within the TOE’s underlying abstract machine (see also sections 6.10 and 8.4).

FIA_USB.1 has been updated according to RI#137 without changing the wording from CAPP/LSPP by sorting the PP statements into the appropriate functional elements FIA_USB.1.1 to FIA_USB.1.3.

8. Rationale

This chapter provides the rationale for the selection, creation, and use of the threats, security policies, objectives, and components. It demonstrates that the security objectives and the security functions defined in the previous chapters are consistent and sufficient to counter the threats and to implement the organizational security policies defined in Chapter 3.

Section 8.1 provides the rationale for the existence of the security objectives based upon the assumed threats and stated security policies while Section 8.2 provides the lower-level rationale for the existence of functional and assurance components based upon the stated security objectives. Section 8.3 provides an analysis that maps given security objectives to components as well as mapping given components to security objectives. In providing a mapping in both directions for the components and objectives, assurance is gained that the objectives were entirely met. This is further detailed in Section 8.4.

In addition to providing a complete rationale, Chapters 5 and 6 also provide the necessary application notes needed to understand how a TOE must meet the stated security objectives. These application notes provide additional information about a particular family/component/element that a developer or evaluator may need in order to fully understand how the component is to be applied.

8.1 Security objectives rationale

This section provides a rationale for the existence of each threat, policy statement, security objective, and component that comprise the protection profile.

8.1.1 Complete Coverage: organizational security policies

This section provides evidence demonstrating coverage of the Organizational Security Policies (OSPs) by both the IT and non-IT security objectives. The following table shows this objective to policy mapping, and the table is followed by a discussion of the coverage for each OSP.

Table 8-1: Mapping OSPs to objectives

Organizational Security Policy	Objective
P.AUTHORIZED_USERS	O.AUTHORIZATION O.MANAGE O.ENFORCEMENT OE.HW_SEP
P.NEED_TO_KNOW	O.DISCRETIONARY_ACCESS O.RESIDUAL_INFORMATION O.MANAGE O.ENFORCEMENT O.COMPROT OE.HW_SEP
P.ACCOUNTABILITY	O.AUDITING O.MANAGE O.ENFORCEMENT OE.HW_SEP

Organizational Security Policy	Objective
P.CLASSIFICATION (LSPP mode only)	O.MANDATORY_ACCESS O.RESIDUAL_INFORMATION O.MANAGE O.ENFORCEMENT O.COMPROT OE.HW_SEP OE.CLASSIFICATION

The following discussion provides detailed evidence of coverage for each organizational security policy:

P.AUTHORIZED_USERS

Only those users who have been authorized to access the information within the system may access the system.

This policy is implemented by the O.AUTHORIZATION objective. O.MANAGE supports this policy by requiring authorized administrators to be able to manage the functions provided for O.AUTHORIZATION. O.ENFORCEMENT ensures that the functions provided for O.AUTHORIZATION are invoked and operate correctly, and OE.HW_SEP ensures that the underlying abstract machine supports this enforcement.

P.NEED_TO_KNOW

The system must limit the access to, modification of, and destruction of the information in protected resources to those authorized users who have a “need to know” for that information.

This policy is implemented by the O.DISCRETIONARY_ACCESS objective, which ensures that authorized users have appropriate permissions before being granted access to protected information. The O.RESIDUAL_INFORMATION objective ensures that information will not be given to users which do not have a need to know, when resources are reused. O.MANAGE ensures that permissions can be managed properly. O.ENFORCEMENT ensures that the access control functions are invoked and operate correctly, and OE.HW_SEP ensures that the underlying abstract machine supports this enforcement. In addition O.COMPROT ensures that information is protected while being transferred to another trusted IT product.

P.ACCOUNTABILITY

The users of the system shall be held accountable for their actions within the system.

This policy is implemented by the O.AUDITING objective by requiring that actions are recorded in an audit trail. The O.MANAGE objective supports this policy by requiring an authorized administrator be able to manage the audit system. O.ENFORCEMENT ensures that functions provided for O.AUDITING are invoked and operate correctly, while OE.HW_SEP ensures that the underlying abstract machine supports this enforcement.

P.CLASSIFICATION (LSPP mode only)

The system must limit the access to information based on sensitivity, as represented by a label, of the information contained in objects, and the formal clearance of users, as represented by subjects, to access that information. The access rules enforced prevent a subject from accessing information which is of higher sensitivity than it is operating at and prevent a subject from causing information from being downgraded to a lower sensitivity.

This policy is implemented by the O.MANDATORY_ACCESS objective, which ensures that authorized users have appropriate clearance before being granted access to labeled information. The objective O.RESIDUAL_INFORMATION ensures that information will not be given to users which do not have a cleared access, when resources are re-used. O.MANAGE ensures that labels and functions provided for O.MANDATORY_ACCESS can be managed properly. O.ENFORCEMENT ensures that the mandatory access control functions are invoked and operate correctly, and OE.HW_SEP ensures that the underlying abstract

machine supports this enforcement. OE.CLASSIFICATION provides for the organizational aspects of managing the mandatory access controls.

For completeness, the following table provides the inverse mapping from Table 8-1, demonstrating that every objective maps to at least one threat or OSP:

Table 8-2: Mapping objectives to threats and policies

Objective	Threat / Policy
O.AUTHORIZATION	P.AUTHORIZED_USERS
O.DISCRETIONARY_ACCESS	P.NEED_TO_KNOW
O.MANDATORY_ACCESS	P.CLASSIFICATION
O.AUDITING	P.ACCOUNTABILITY
O.RESIDUAL_INFORMATION	P.NEED_TO_KNOW P.CLASSIFICATION
O.MANAGE	P.AUTHORIZED_USERS P.NEED_TO_KNOW P.CLASSIFICATION P.ACCOUNTABILITY
O.ENFORCEMENT	P.AUTHORIZED_USERS P.NEED_TO_KNOW P.CLASSIFICATION P.ACCOUNTABILITY
O.COMPROT	P.NEED_TO_KNOW P.CLASSIFICATION

8.1.2 Complete coverage: environmental assumptions

This section provides evidence demonstrating coverage of the non-IT security objectives by the environmental assumptions. The following table shows this assumption to objective mapping.

Table 8-3: Mapping non-IT security objectives to environmental assumptions

Non-IT Security Objectives	Environmental Assumptions / Organizational Security Policies
OE.INSTALL	A.MANAGE A.NO_EVIL_ADMIN A.PEER
OE.PHYSICAL	A.LOCATE A.PROTECT A.CONNECT
OE.CREDEN	A.COOP
OE.HW_SEP	P.AUTHORIZED_USERS P.NEED_TO_KNOW, P.CLASSIFICATION, P.ACCOUNTABILITY
OE.CLASSIFICATION	A.SENSITIVITY, A.CLEARANCE, P.CLASSIFICATION

The following discussion provides detailed evidence of coverage for each Non-IT Security Objective:

OE.INSTALL

Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner that maintains IT security objectives.

The TOE requires proper installation to operate in a secure way. This is addressed by the assumption that the TOE is managed by personnel with the required knowledge to perform the installation in the required way (A.MANAGE), that management personnel does not deliberately undermine the security (A.NO_EVIL_ADMIN) and that the TOE is installed in line with the configuration of other systems the TOE is connected to (A.PEER).

OE.PHYSICAL

Those responsible for the TOE must ensure that those parts of the TOE critical to security policy are protected from physical attack, which might compromise IT security objectives.

The objective for the physical protection of the TOE is addressed by the assumption that the TOE is in a protected environment (A.LOCATE), is protected from unauthorized physical access (A.PROTECT) and has physically protected network connections for those network links where the communication is not logically protected by security functions of the TOE itself.

OE.CREDEN

Those responsible for the TOE must ensure that all access credentials, such as passwords or other authentication information, are protected by the users in a manner that maintains IT security objectives.

The objective of users handling their access credentials in a secure way addresses the assumptions that users are co-operative and do not deliberately undermine the security of the TOE by passing their passwords to others or define access rights to objects they own or have control of such that the overall objective of protecting information within an organization is undermined (A.COOP).

OE.HW_SEP

The underlying abstract machine must provide a separation mechanism that can be used by the TOE to protect the TSF and TSF data from unauthorized access and modification.

The objective of having hardware support to assist the TOE to protect the TSF data from unauthorized access and modification (O.ENFORCMENT) addresses the organizational security policies for controlled access to the TOE (P.AUTHORIZED_USERS), need-to-know separation (P.NEED_TO_KNOW), classification of information (P.CLASSIFICATION) and individual user accountability (P.ACCOUNTABILITY). The enforcement of those policies within the TOE requires the protection of the TSF data used to implement the policies within the TOE.

OE.CLASSIFICATION (LSPP mode only)

Those responsible for the TOE must ensure that users of the TOE are cleared for access to information depending on the classification of the information. They must also ensure that information is correctly classified to be protected by the security functions of the TOE.

The objective of having appropriate classification of users and data addresses the policy to enforce information flow policy based on the classification of data and the clearance level of users (P.CLASSIFICATION).

For completeness, the following table provides the inverse mapping from Table 8-3, demonstrating that every environmental assumption maps to at least one Non-IT security objective:

Table 8-4: Mapping non-IT security objectives to environmental assumptions

Environmental Assumptions	Non-IT Security Objectives
A.MANAGE	OE.INSTALL
A.NO_EVIL_ADMIN	OE.INSTALL

Environmental Assumptions	Non-IT Security Objectives
A.PEER	OE.INSTALL
A.LOCATE	OE.PHYSICAL
A.PROTECT	OE.PHYSICAL
A.CONNECT	OE.PHYSICAL
A.COOP	OE.CREDEN
A.CLEARANCE	OE.CLASSIFICATION
A.SENSITIVITY	OE.CLASSIFICATION

OE.CLASSIFICATION was introduced in this Security Target to address a flaw in LSPP.

8.2 Security requirements rationale

This section provides the rationale for the internal consistency and completeness of the security functional requirements defined in this Security Target.

8.2.1 Internal consistency of requirements

This section describes the mutual support and internal consistency of the components selected for this Security Target. These properties are discussed for both functional and assurance components.

The functional components were selected from CC components defined in Part 2 of the Common Criteria. The use of component refinement was accomplished in accordance with CC guidelines.

An additional component was included by the [LSPP] to clarify the relationship of objects and security attributes.

Assignment, selection, and refinement operations were carried out among components using consistent computer security terminology. This helps to avoid the ambiguity associated with interpretations of meanings of terms between related components.

Multiple instantiation of identical or hierarchically-related components was used to clearly state the required functionality that must exist in a TOE conformant with this profile.

For internal consistency of the requirements, the following rationale is provided:

Auditing

The requirements for auditing have been completely derived from [LSPP] and [CAPP]. The rationale for those requirements is:

FAU_GEN.1 defines the events that the TOE is required to be able to audit. Those events are related to the other security functional requirements showing which event contributes to make users accountable for their actions with respect to the requirement. FAU_GEN.2 requires that the events are associated with the identity of the user that caused the event. The identity has been associated with the subject that causes an auditable event by FIA_USB.1. Of course this can only be accomplished if the user is already known, which may not be the case for failed login attempts.

FAU_SAR.1 ensures that authorized administrators are able to evaluate the audit records, while FAU_SAR.2 requires that no other users can read the audit records (because they may contain sensitive information). Taking into account that the amount of audit records gathered may be very large, FAU_SAR.3 requires that the TOE provides the ability to search the audit records for a set that satisfies defined attributes.

To avoid all possible audit records always being generated (which would result in an unacceptable overhead to the system performance and might easily fill up the available audit trail space) the TOE is required in FAU_SEL.1 to provide the possibility to restrict the events to be audited based on a set of defined attributes.

Requirement FAU_STG.1 defines that audit records need to be protected from unauthorized deletion and modification to ensure their completeness and correctness. Requirement FAU_STG.3 addresses the aspect that the system detects a shortage in the audit trail space. This can be used to take preventive action, e.g. backup the audit trail and release the space to avoid a critical situation.

FAU_STG.4 addresses the problem that the TOE might not be able to record further audit records (e. g. due to the shortage of some resources). Also in this case the TOE needs to ensure that such a situation cannot be misused by a user to bypass the auditing of critical activities. Otherwise a user might deliberately bring the TOE into a situation where it is no longer able to audit critical events just to avoid that a critical action he performs is audited.

Because accountability also requires the ability to prove when and in which sequence security relevant events occurred, FPT_STM.1 provides for a reliable time reference.

Management of audit is addressed by FMT_MTD.1 for both the audit trail and audited events.

Discretionary access control

FDP_ACC.1 requires the existence of a Discretionary Access Control Policy for named objects in z/OS, including named objects within the UNIX realm. The rules of this policy are described in FDP_ACF.1 in iterations for UNIX and non-UNIX objects. Discretionary access control rules are partly based on user security attributes provided through FIA_ATD.1. Management of access rights is defined in FMT_MSA.1(1)and FMT_REV.1. When initialized, object attributes are initialized to restrictive values (FMT_MSA.3(1)), to avoid breaches of the security policy.

Because access decisions are based on user attributes, subjects must be bound to users on whose behalf they take action (FIA_USB.1). This must be supported by proper identification and authentication.

Other supportive requirements are from TOE self-protection, where reference mediation and domain separation assure that these mechanisms are always invoked and cannot be tampered with.

Discretionary access control is also supported by the requirements for residual information protection, which prevent users from accessing information they are not authorized to by way of residual information remaining in objects that they allocate.

Mandatory access control (LSPP mode only)

FDP_IFC.1 requires the existence of a mandatory access control policy for named objects in z/OS. The rules of this policy are described in FDP_IFF.2. Mandatory access control rules are partly based on user security attributes provided through FIA_ATD.1. Management of labels attached to objects is defined in FMT_MSA.1(2) and FMT_REV.1(2). When new objects are created, proper attribute initialization is ensured by FMT_MSA.3(2).

Import and export of labeled and unlabeled data (FDP_ETC.1, FDP_ETC.2, FDP_ITC.1, FDP_ITC.2) can be provided over a trusted channel (FTP_ITC.1). FPT_TDC.1 ensures that labels can be consistently interpreted when labeled data is transferred from one system to another (provided the two systems have been configured with compatible definitions of the security labels).

Because access decisions are based on user attributes, subjects must be bound to users on whose behalf they take action (FIA_USB.1). This must be supported by proper identification and authentication.

Other supportive requirements are from TOE self-protection, where reference mediation and domain separation assure that these mechanisms are always invoked and cannot be tampered with.

Mandatory access control is also supported by the requirements for residual information protection, which prevent users from accessing information they are not authorized to by way of residual information remaining in objects that they allocate.

Identification and authentication

Identification and authentication are required for discretionary and mandatory access control as well as for auditing, which are based on the identity of individual users. FIA_UAU.1 and FIA_UID.1 require that users are authenticated before they can perform any critical action on the TOE. Access of unauthenticated users is restricted to resources the installation has defined to be accessible by the pseudo user ID the HTTP server uses for unauthenticated users. FIA_SOS.1 ensures that the mechanism used for authentication (passwords) has a minimum strength. FIA_UAU.7 provides some level of protection against simple spoofing in the TOE environment. FIA_USB.1 ensures that a TOE subject (z/OS task) is properly bound to the user for whom it runs. This association also provides the user attributes (defined by FIA_ATD.1) necessary to take policy decisions. Management of the user attributes and authentication data is provided by FMT_MTD.1(3), FMT_MTD.1(4), and FMT_REV.1(1).

Object reuse

Object reuse (as required by FDP_RIP.2 and Note 1) is a supporting function that prevents unauthorized access to information through residuals left in objects when they are reallocated to another subject or object.

Object reuse therefore supports the intention of the discretionary and (in LSPP mode) mandatory access control policies as well as identification and authentication and secure communication (for the protection of keys and data).

Security management

The functions defined so far require several management functions as defined by FMT_SMF.1.

Management of access rights and (in LSPP mode) labels attached to objects is necessary to configure the DAC and (in LSPP mode) MAC mechanisms; it is defined by FMT_MSA.1 and FMT_REV.1(2) “Revocation of Object Attributes”. In addition new objects are required to have default access rights and security labels which are required by FMT_MSA.3.

Management of users and groups is defined in FMT_MTD.1(3) “Management of User Attributes” and FMT_REV.1(1) “Revocation of User Attributes”. Because passwords are used for authentication, the management of authentication data is also required in FMT_MTD.1(4) “Management of Authentication Data”.

Management of the audit system is covered by the requirements for the management of the audit trail (FMT_MTD.1(1) “Management of the Audit Trail”) and the management of the audit events (FMT_MTD.1(2) “Management of the Audit Events”). Audit trail management is supported by the requirements for the audit review (FAU_SAR.1 and FAU_SAR.3) as well as the requirements for the protection of the audit trail (FAU_STG.3 and FAU_STG.4). Management of the audit events is supported by the ability to select the events to be audited (FAU_SEL.1).

In addition the TOE supports several roles, which is expressed by FMT_SMR.1.

Security management requirements therefore provide support for auditing, discretionary and (in LSPP mode) mandatory access control, and identification and authentication.

TSF protection

The TOE needs to ensure that users are limited in their activities by the boundaries defined by the access control policies. To ensure this the TSF need to check all access of subjects to protected objects (as required by FPT_RVM.1) and maintain a domain for its own execution that protects it from interference and tampering by any subject that is not part of the TSF. This is expressed with the requirement FPT_SEP.1.

Meeting these requirements provides the basis for all other security functions.

The underlying hardware of the TOE performs extensive and continuous self tests to ensure the correct operation of the TOE. In the case when an error is detected, the TOE is informed by way of a machine-check interrupt about the problem, allowing the TOE to react to the error like shut down in a controlled way (provided the error does not lead to an immediate stop of the machine).

Secure communication

The TOE provides a protocol that allows applications or users to securely communicate with other trusted IT products (which may be other instantiations of the TOE). This protocol uses cryptographic functions to ensure the confidentiality and integrity of the user data during transmission as required. The requirements for those cryptographic functions are defined in FCS_CKM.1, FCS_CKM.2 and FCS_COP.1.

The protocol provides the ability to establish an Inter-TSF trusted channel, as required by FTP_ITC.1. Within this channel, user data transferred is protected for confidentiality (as required by FDP_UCT.1) and integrity (as required by FDP_UIT.1).

Management of parameters required for secure communication is addressed by FMT_MTD.1(6).

The secure generation of cryptographic keys used for secure communications is addressed by FMT_MSA.2.

8.2.2 Complete coverage: security objectives

This section demonstrates that the functional components selected for this profile provide complete coverage of the defined security objectives. The mapping of components to security objectives is depicted in the following table. Note the green coloring as an indication for applicability to LSPP only.

Table 8-5: Mapping security objectives to security functional requirements

Security Objective	Security Functional Requirement
O.AUTHORIZATION	User attribute definition (FIA_ATD.1) Strength of authentication data (FIA_SOS.1) Authentication (FIA_UAU.1) Protected authentication feedback (FIA_UAU.7) Identification (FIA_UID.1) User subject binding (FIA_USB.1) Management of user attributes (FMT_MTD.1(3)) Management of authentication data (FMT_MTD.1(4)) Revocation of user attributes (FMT_REV.1(1))
O.DISCRETIONARY_ACCESS	Discretionary access control policy (FDP_ACC.1) Discretionary access control functions for non-z/OS UNIX objects (FDP_ACF.1(1)) Discretionary access control functions for z/OS UNIX objects (FDP_ACF.1(2)) User attribute definition (FIA_ATD.1) User subject binding (FIA_USB.1) Management of object security attributes (FMT_MSA.1(1)) Static attribute initialization (FMT_MSA.3(1)) Revocation of object attributes (FMT_REV.1(2))
O.MANDATORY_ACCESS	Export of unlabeled user data (FDP_ETC.1) Export of labeled user data (FDP_ETC.2) Mandatory access control policy (FDP_IFC.1) Mandatory access control functions (FDP_IFF.2) Import of unlabeled user data (FDP_ITC.1) Import of labeled user data (FDP_ITC.2) User attribute definition (FIA_ATD.1) User subject binding (FIA_USB.1) Management of object security attributes for MAC (FMT_MSA.1(2)) Static attribute initialization for MAC (FMT_MSA.3(2)) Revocation of object attributes (FMT_REV.1(2)) Inter-TSF basic TSF data consistency (FPT_TDC.1) Inter-TSF trusted channel (FTP_ITC.1)

Security Objective	Security Functional Requirement
O.AUDITING	Audit data generation (FAU_GEN.1) User identity association (FAU_GEN.2) Audit review (FAU_SAR.1) Restricted audit review (FAU_SAR.2) Selectable audit review (FAU_SAR.3) Selective audit (FAU_SEL.1) Guarantees of audit data availability (FAU_STG.1) Action in case of possible audit data loss (FAU_STG.3) Prevention of audit data loss (FAU_STG.4) User subject binding (FIA_USB.1) Management of the audit trail (FMT_MTD.1(1)) Management of audited events (FMT_MTD.1(2)) Reliable time stamps (FPT_STM.1)
O.RESIDUAL_INFORMATION	Object residual information protection (FDP_RIP.2) Subject residual information protection (Note 1)
O.MANAGE	Audit review (FAU_SAR.1) Selectable audit review (FAU_SAR.3) Selective audit (FAU_SEL.1) Action in case of possible audit data loss (FAU_STG.3) Prevention of audit data loss (FAU_STG.4) Management of object security attributes (FMT_MSA.1(1)) Management of object security attributes for MAC (FMT_MSA.1(2)) Static attribute initialization (FMT_MSA.3(1)) Static attribute initialization for MAC (FMT_MSA.3(2)) Management of the audit trail (FMT_MTD.1(1)) Management of audited events (FMT_MTD.1(2)) Management of user attributes (FMT_MTD.1(3)) Management of authentication data (FMT_MTD.1(4)) Management of cryptographic keys (FMT_MTD.1(5)) Management of network configuration (FMT_MTD.1(6)) Revocation of user attributes (FMT_REV.1(1)) Revocation of object attributes (FMT_REV.1(2)) Specification of management functions (FMT_SMF.1) Security management roles (FMT_SMR.1)
O.ENFORCEMENT	Abstract machine testing (FPT_AMT.1)[5] Reference mediation (FPT_RVM.1) Domain separation (FPT_SEP.1)

Security Objective	Security Functional Requirement
O.COMPROT	Cryptographic key generation (SSL/TLS: Symmetric algorithms) (FCS_CKM.1(1)) Cryptographic key generation (IPsec: Symmetric algorithms)(FCS_CKM.1(2)) Cryptographic key distribution (SSL/TLS: RSA public keys) (FCS_CKM.2(1)) Cryptographic key distribution (SSL/TLS: Symmetric keys) (FCS_CKM.2(2)) Cryptographic key distribution (IPsec: DH key exchange) (FCS_CKM.2(3)) Cryptographic operation (SSL/TLS: RSA) (FCS_COP.1(1)) Cryptographic operation (SSL/TLS: Symmetric operations) (FCS_COP.1(2)) Cryptographic operation (IPsec: Payload encryption) (FCS_COP.1(3)) Cryptographic operation (IPsec: HMAC-SHA) (FCS_COP.1(4)) basic data exchange Confidentiality (FDP_UCT.1) data exchange integrity (FDP_UIT.1) Secure security attributes (FMT_MSA.2) Management of cryptographic keys (FMT_MTD.1(5)) Management of network configuration (FMT_MTD.1(6)) Inter-TSF trusted channel (FTP_ITC.1)

The following discussion provides detailed evidence of coverage for each security objective:

O.AUTHORIZATION

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]. To ensure authorized access to the TOE, authentication data and other relevant user attributes are protected [FIA_ATD.1, FIA_UAU.7] and can be managed appropriately [FIA_MTD.1(4) ”Management of Authentication Data”, FIA_MTD.1(3) “Management of User Attributes”, FMT_REV.1(1) “Revocation of User Attributes”]. The strength of the authentication mechanism must be sufficient to ensure unauthorized users cannot easily pose as authorized users [FIA_SOS.1]. Proper authorization for subjects acting on behalf of users is also ensured [FIA_USB.1].

O.DISCRETIONARY_ACCESS

The TSF must control access to resources based on identity of users. The TSF must allow authorized users to specify which resources may be accessed by which users.

Discretionary access control must have a defined scope of control [FDP_ACC.1]. The rules of the DAC policy must be defined [FDP_ACF.1]. The security attributes of objects used to enforce the DAC policy must be defined. The security attributes of subjects used to enforce the DAC policy must be defined [FIA_ATD.1, FIA_USB.1]. Authorized users must be able to control who has access to objects [FMT_MSA.1(1)] and be able to revoke that access [FMT_REV.1(2) “Revocation of Object Attributes”]. Protection of named objects must be continuous, starting from object creation [FMT_MSA.3(1)].

O.MANDATORY_ACCESS (LSPP mode only)

The TSF must record the security relevant actions of users of the TOE. The TSF must present this information to authorized administrators.

Mandatory access control attributes and rules must be defined [FDP_IFF.2] and must have a defined scope of control [FDP_IFC.1]. The rules for importing unlabeled data [FDP_ITC.1] and labeled data [FDP_ITC.2] must be covered, as must the exporting of unlabeled data [FDP_ETC.1] and labeled data [FDP_ETC.2], ensuring

that a consistent interpretation of the TSF attributes be achieved [FPT_TDC.1] and providing a trusted channel for data exchange [FTP_ITC.1]. Finally, if the MAC policy is to be correctly enforced, it is required that correct and sufficient static attributes be associated with each object [FMT_MSA.3(2), FMT_MSA.1(2) “Management of Object Security Attributes for MAC”, FMT_REV.1 “Revocation of Object Security Attributes”], and that the binding between processes and the attributes of the user on whose behalf they operate be correct and unforgable [FIA_ATD.1, FIA_USB.1].

O.AUDITING

> The TSF must record the security relevant actions of users of the TOE. The TSF must present this information to authorized administrators.

Security-relevant actions must be defined, auditable [FAU_GEN.1], and capable of being associated with individual users [FAU_GEN.2, FIA_USB.1]. The audit trail must be protected so that only authorized users may access it [FAU_SAR.2]. The TSF must provide the capability to audit the actions of an individual user [FAU_SAR.3, FAU_SEL.1, FIA_USB.1]. The audit trail must be complete [FAU_STG.1, FAU_STG.4]. The time stamp associated must be reliable [FPT_STM.1]. An authorized administrator must be able to review [FAU_SAR.1] and manage [FAU_STG.3, FMT_MTD.1(1) “Management of the Audit Trail”, FMT_MTD.1(2) “Management of Audited Events”] the audit trail.

O.RESIDUAL_INFORMATION

> The TSF must ensure that any information contained in a protected resource is not released when the resource is recycled.

Residual information associated with defined objects in the TOE must be purged prior to the re-use of the object containing the residual information [FDP_RIP.2] and before a resource is re-allocated to another subject [Note 1].

O.MANAGE

> The TSF must provide all the functions and facilities necessary to support the authorized administrators that are responsible for the management of TOE security.

Aspects that need to be managed must be defined [FMT_SMF.1] The TSF must provide for an authorized administrator to manage the TOE [FMT_SMR.1]. The administrative user must be able to administer the audit system [FAU_STG.3, FAU_STG.4, FMT_MTD.1(1) “Management of the Audit Trail”, FMT_MTD.1(2) “Management of the Audit Events”] and review it [FAU_SAR.1, FAU_SAR.3, FAU_SEL.1], to manage user accounts [FMT_MTD.1(3) “Management of User Attributes”, FMT_MTD.1(4) “Management of Authentication Data”, FMT_REV.1(1) “Revocation of User Attributes”] to manage cryptographic keys [FMT_MTD.1(5) “Management of Cryptographic Keys”], network security configuration [FMT_MTD.1(6) “Management of *network configuration*”] and to manage object security attributes [FMT_MSA.1, FMT_REV.1(2) “Revocation of Object Attributes”]. In addition the default values for access control need to be defined [FMT_MSA.3].

O.ENFORCEMENT

> The TSF must be designed and implemented in a manner that ensures that the organizational policies are enforced in the target environment.

The TSF must make and enforce the decisions of the TSP [FPT_RVM.1]. It must be protected from interference that would prevent it from performing its functions [FPT_SEP.1]. Additionally, the TOE must provide the capability to demonstrate correct operation of the TSF’s underlying abstract machine [FPT_AMT.1] which is satisfied by the TOE environment. The correctness of this objective is further met through the assurance requirements defined in this Security Target.

This objective provides global support to other security objectives for the TOE by protecting the parts of the TOE which implement policies and ensures that policies are enforced.

O.COMPROT

The TSF must be able to establish an Inter-TSF trusted channel between itself and another trusted IT product [FTP_ITC.1] protecting the user data transferred from disclosure [FDP_UCT.1] and undetected modification [FDP_UIT.1]. This TSF uses cryptographic functions in the implementation that require securely generating keys [FCS_CKM.1(1), FCS_CKM.1(2)], distributing keys [FCS_CKM.2(1), FCS_CKM.2(2), FCS_CKM.2(3)] and performing the required cryptographic operations on the user data [FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4)]. Keys used must be secure enough such that they can not be guessed [FMT_MSA.2]. Certificates and keys as well as network configuration parameters can only be managed by authorized administrators [FMT_MTD.1(5), FMT_MTD.1(6)].

No security functions for the non-IT environment have been added, because the procedures that need to be implemented can (and probably will) be different for each site running the evaluated version of the TOE. Therefore no specific security functional requirements and security functions for the non-IT environment have been defined in this Security Target. Individual sites running z/OS should validate that the procedures and physical security measures they have put in place are sufficient to cover the security objectives defined for the environment of the TOE in this Security Target.

Security requirements for the IT environment have been added to define the support required by the TOE from the underlying processor. As with every operating system that also runs untrusted software, some kind of separation mechanism must exist that prohibits the untrusted software from tampering with trusted software and TSF data. In the case of this TOE the processor must supply a separation mechanism such that memory areas as well as hardware privileges required to directly access devices or memory management functions are protected from direct access by untrusted software. This is defined with a *memory access control policy* that the underlying processor must support. This policy is expressed using FDP_ACC.1 and FDP_ACF.1, as well as FMT_MSA.3 from Part 2 of the Common Criteria.

8.2.3 Security requirements instantiation rationale

This section provides the rationale for the selections and instantiations made in the security requirements section for the security requirements taken from Part 2 of the Common Criteria. A rationale is given only for those requirements where selections and instantiations in addition to the ones defined in [LSPP] and [CAPP] are provided. For the selections and instantiations performed in [LSPP] and [CAPP], the reader is referred to the rationale provided there.

In FAU_GEN.1, the different events that the TOE is able to audit are defined with respect to the SFR they belong to. This list has been taken from [LSPP] (which is a strict superset of [CAPP]) and extended with the names of the events and with the SFR that are additional to the ones required by [LSPP].

In FAU_SAR.1, it is expressed that an authorized administrator is able to read all the audit data from the audit log and therefore is able to evaluate the information of the audit trail.

In FAU_SAR.3, it is expressed that an authorized administrator is able to search the audit trail for events matching defined selection criteria where the selection can be performed based on the list of attributes defined in the SFR.

In FAU_STG.1, the requirement for preventing unauthorized modifications of the audit records is expressed.

In FAU_STG.3, the requirement for timely notification of the authorized administrator about a potential shortage in the disk space for the audit trail is expressed, allowing the administrator to take the appropriate measures to overcome the situation before it gets critical.

FCS_CKM.1 reflects the requirements for the generation of symmetric keys to be used by the SSL/TLS protocol to set up and maintain a trusted channel between the TOE and another trusted IT product.

FCS_CKM.2 has multiple instantiations to reflect the different ways for public key exchange and session key exchange.

FCS_COP.1 has multiple instantiations to define the different cryptographic algorithms used within the SSL/TLS protocol (with the cipher suites configured for the TOE, which are a subset of the cipher suites allowed in the standards defining those protocols).

In FDP_ACC.1, the different objects that z/OS controls with a discretionary access control function are listed.

FDP_ACF.1 gets somewhat complicated with expressing the different policies for discretionary access control for the different types of objects. It was decided to list the rules for z/OS and z/OS UNIX objects separately, because they differ significantly.

In FIA_ATD.1,nothing has been added as additional security attribute of users within the evaluated configuration of z/OS.

In FIA_USB.1, the way z/OS associates real users with tasks is expressed.

In FMT_MSA.1(1), the ability of the authorized administrator and the profile owner to modify access rights for objects is expressed. In addition, the special role of the owner in the case of UNIX objects is expressed.

In FMT_REV.1, “Revocation of User Attributes” the delayed revocation method has been added, because this is the standard way z/OS behaves. To get immediate revocation the administrative user has to force the user to log off after he has made the modifications to the users attribute.

In FMT_REV.1, “Revocation of Object Attributes” the z/OS implementation of delayed revocation is defined.

FMT_SMF.1 has been added to comply with CC version 2.3 and the dependencies defined there. The Security Target defines management requirements in the iterations of FMT_MSA.1 and the iterations of FMT_MTD.1 for

- Audit trail management
- Audit event management
- User attribute management
- Authentication data management
- Cryptographic key management
- Network configuration management

Those aspects are listed in this security functional requirement.

FMT_SMR.1 defines the roles of authorized administrators, users authorized by DAC or MAC policies to modify object security attributes, users authorized to modify their own authentication data, users authorized to perform administrative actions within a group, RACF auditors and RACF group auditors.

FPT_AMT.1 expresses the ability of the authorized administrator to perform the tests of the underlying abstract machine on his demand, this requirement is satisfied by the TOE environment.

FPT_TDC.1 expresses the ability to consistently interpret labels when labeled data is transferred between different systems.

In FTP_ITC.1, the ability to set up a trusted channel between the TOE and another trusted IT product is expressed where either the TOE or the other trusted IT product is allowed to initiate the communication over the trusted channel.

8.2.4 Security requirements coverage

The following table shows that each security functional requirement addresses at least one objective.

Table 8-6: Mapping security functional requirements to objectives

CC Identifier	Security Objective
FAU_GEN.1	O.AUDITING
FAU_GEN.2	O.AUDITING
FAU_SAR.1	O.AUDITING, O.MANAGE
FAU_SAR.2	O.AUDITING
FAU_SAR.3	O.AUDITING, O.MANAGE
FAU_SEL.1	O.AUDITING, O.MANAGE

CC Identifier	Security Objective
FAU_STG.1	O.AUDITING
FAU_STG.3	O.AUDITING, O.MANAGE
FAU_STG.4	O.AUDITING, O.MANAGE
FCS_CKM.1(1)	O.COMPROT
FCS_CKM.1(2)	O.COMPROT
FCS_CKM.2(1)	O.COMPROT
FCS_CKM.2(2)	O.COMPROT
FCS_CKM.2(3)	O.COMPROT
FCS_COP.1(1)	O.COMPROT
FCS_COP.1(2)	O.COMPROT
FCS_COP.1(3)	O.COMPROT
FCS_COP.1(4)	O.COMPROT
FDP_ACC.1	O.DISCRETIONARY_ACCESS
FDP_ACF.1(1)	O.DISCRETIONARY_ACCESS
FDP_ACF.1(2)	O.DISCRETIONARY_ACCESS
FDP_ETC.1	O.MANDATORY_ACCESS
FDP_ETC.2	O.MANDATORY_ACCESS
FDP_IFC.1	O.MANDATORY_ACCESS
FDP_IFF.2	O.MANDATORY_ACCESS
FDP_ITC.1	O.MANDATORY_ACCESS
FDP_ITC.2	O.MANDATORY_ACCESS
FDP_RIP.2	O.RESIDUAL_INFORMATION
Note 1	O.RESIDUAL_INFORMATION
FDP_UCT.1	O.COMPROT
FDP_UIT.1	O.COMPROT
FIA_ATD.1	O.AUTHORIZATION, O.DISCRETIONARY_ACCESS, O.MANDATORY_ACCESS
FIA_SOS.1	O.AUTHORIZATION
FIA_UAU.1	O.AUTHORIZATION
FIA_UAU.7	O.AUTHORIZATION
FIA_UID.1	O.AUTHORIZATION
FIA_USB.1	O.AUTHORIZATION, O.DISCRETIONARY_ACCESS, O.MANDATORY_ACCESS, O.AUDITING

CC Identifier	Security Objective
FMT_MSA.1(1)	O.DISCRETIONARY_ACCESS, O.MANAGE
FMT_MSA.1(2)	O.MANDATORY_ACCESS, O.MANAGE
FMT_MSA.2	O.COMPROT
FMT_MSA.3(1)	O.DISCRETIONARY_ACCESS, O.MANAGE
FMT_MSA.3(2)	O.MANDATORY_ACCESS, O.MANAGE
FMT_MTD.1(1)	O.AUDITING, O.MANAGE
FMT_MTD.1(2)	O.AUDITING, O.MANAGE
FMT_MTD.1(3)	O.AUTHORIZATION, O.MANAGE
FMT_MTD.1(4)	O.AUTHORIZATION, O.MANAGE
FMT_MTD.1(5)	O.MANAGE
FMT_MTD.1(6)	O.MANAGE, O.COMPROT
FMT_REV.1(1)	O.MANAGE
FMT_REV.1(2)	O.DISCRETIONARY_ACCESS, O.MANDATORY_ACCESS, O.MANAGE
FMT_SMF.1	O.MANAGE
FMT_SMR.1	O.MANAGE
FPT_AMT.1[6]	O.ENFORCEMENT
FPT_RVM.1	O.ENFORCEMENT
FPT_SEP.1	O.ENFORCEMENT
FPT_STM.1	O.AUDITING
FPT_TDC.1	O.MANDATORY_ACCESS
FTP_ITC.1	O.MANDATORY_ACCESS, O.COMPROT

8.2.5 Rationale for security requirements for the IT environment

These requirements define the need for an access control policy implemented in the underlying abstract machine that allows reserving the access and manipulation of critical processor and memory resources to special software (instructions) operating with a defined privilege attribute (usually called "supervisor" or "system" mode). The TSF have to ensure that no untrusted software will ever execute with this privilege. Based on this the TSF can then control the access to memory objects and other processor resources and implement the high level access control functions as well as the TSF self protection.

To do this the underlying processor has to provide a basic access control mechanism where access to processor resources (like registers) and memory areas is controlled based on a processor attribute where the implementation of the TSF ensures that untrusted software never executes with this attribute. This is expressed with FDP_ACC.1 and FDP_ACF.1. Because the processor may allow read access to specific registers for software running without “supervisor” privilege, FDP_ACF.1.3 is used to define this.

[6] Note that FPT_AMT.1 is satisfied by the TOE environment.

The requirements don't define the exact rules because they may differ slightly for different processor types. For example a new processor may implement additional instructions and additional registers but still be fully downwards compatible. Because software developed for the older versions of the processor will not use the additional instructions and will not touch the additional registers, the claims for the software still hold although the objects controlled by the new processor differ from those controlled by the old processor. Of course, if anybody wants to evaluate the underlying processor those rules have to be defined precisely for the specific processor type that is the target of the hardware evaluation.

The "static attribute initialization" (FMT_MSA.3) is defined here as the value of the processor attribute ("user" or "supervisor") at start-up of the processor (after reset or power-up). This has to be "permissive" because the registers and memory areas need to be initialized. It is therefore necessary that the software that performs those initialization activities is part of the TSF.

The security requirements for the IT environment address the security objective OE.HW_SEP because the memory access control policy allows the TOE to protect the TSF and the TSF data from unauthorized access by untrusted software. The TOE has to use the memory access control policy to allow memory access by untrusted software just to those memory areas that belong to the untrusted software itself. Access to special hardware registers will be managed by the TSF such that this access will always be reserved to trusted software. This shows that the security requirements for the IT environment are sufficient to protect the TSF and TSF data from unauthorized access and modification when used correctly by the TOE.

Abstract machine testing (FPT_AMT.1) addresses the security objective OE.HW_SEP as follows: It provides assurance that the separation mechanisms of the abstract machine operate correctly, as required by the TOE for the protection of its TSFs.

The following table shows the mapping of the security functional requirements for the IT environment to the security objectives for the IT environment:

Table 8-7: Mapping security functional requirements for the IT environment to objectives

SFR	Objective
FDP_ACC.1	OE.HW_SEP
FDP_ACF.1	OE.HW_SEP
FMT_MSA.3	OE.HW_SEP
FMT_AMT.1	OE.HW_SEP

8.2.6 Security requirement dependency analysis

The following table shows the dependencies which exist. A box with an X in it indicates a dependency which has been satisfied. A box with an O in it indicates an optional dependency where one of the options has been satisfied. A box with an N indicates a dependency that has not been resolved with arguments provided in the text following the table, why this dependency does not apply for the TOE.

Table 8-8: Dependencies between security functional requirements

CC Identifier	ADV_SPM.1	FAU_GEN.1	FAU_SAR.1	FAU_STG.1	FCS_CKM.1	FCS_CKM.2	FCS_CKM.4	FCS_COP.1	FDP_ACC.1	FDP_ACF.1	FDP_IFC.1	FDP_IFF.1	FDP_ITC.1	FDP_ITC.2	FIA_ATD.1	FIA_UAU.1	FIA_UID.1	FMT_MSA.1	FMT_MSA.2	FMT_MSA.3	FMT_MTD.1	FMT_SMF.1	FMT_SMR.1	FPT_STM.1	FPT_TDC.1	FTP_ITC.1	FTP_TRP.1
FAU_GEN.1																								X			
FAU_GEN.2		X															X										
FAU_SAR.1		X																									
FAU_SAR.2			X																								

CC Identifier	ADV_SPM.1	FAU_GEN.1	FAU_SAR.1	FAU_STG.1	FCS_CKM.1	FCS_CKM.2	FCS_CKM.4	FCS_COP.1	FDP_ACC.1	FDP_ACF.1	FDP_IFC.1	FDP_IFF.1	FDP_ITC.1	FDP_ITC.2	FIA_ATD.1	FIA_UAU.1	FIA_UID.1	FMT_MSA.1	FMT_MSA.2	FMT_MSA.3	FMT_MTD.1	FMT_SMF.1	FMT_SMR.1	FPT_STM.1	FPT_TDC.1	FTP_ITC.1	FTP_TRP.1
FAU_SAR.3			X																								
FAU_SEL.1		X																			X						
FAU_STG.1		X																									
FAU_STG.3				X																							
FAU_STG.4				X																							
FCS_CKM.1(1)						O	N	O											X								
FCS_CKM.1(2)						O	N	O											X								
FCS_CKM.2(1)					O		N						O	O					X								
FCS_CKM.2(2)					O		N						O	O					X								
FCS_CKM.2(3)					O		N						O	O					X								
FCS_COP.1(1)					O		N						O	O					X								
FCS_COP.1(2)					O		N						O	O					X								
FCS_COP.1(3)					O		N						O	O					X								
FCS_COP.1(4)					O		N						O	O					X								
FDP_ACC.1										X																	
FDP_ACF.1(1)									X												X						
FDP_ACF.1(2)									X												X						
FDP_ETC.1									O		O																
FDP_ETC.2									O		O																
FDP_IFC.1												X															
FDP_IFF.2											X									X							
FDP_ITC.1									O		O									X						O	O
FDP_ITC.2									O		O																
FDP_RIP.2																											
Note 1																											
FDP_UCT.1									O		O		O														O
FDP_UIT.1									O		O		O														O
FIA_ATD.1																											
FIA_SOS.1																											
FIA_UAU.1																	X										
FIA_UAU.7																X											
FIA_UID.1																											
FIA_USB.1															X												

CC Identifier	ADV_SPM.1	FAU_GEN.1	FAU_SAR.1	FAU_STG.1	FCS_CKM.1	FCS_CKM.2	FCS_CKM.4	FCS_COP.1	FDP_ACC.1	FDP_ACF.1	FDP_IFC.1	FDP_IFF.1	FDP_ITC.1	FDP_ITC.2	FIA_ATD.1	FIA_UAU.1	FIA_UID.1	FMT_MSA.1	FMT_MSA.2	FMT_MSA.3	FMT_MTD.1	FMT_SMF.1	FMT_SMR.1	FPT_STM.1	FPT_TDC.1	FTP_ITC.1	FTP_TRP.1
FMT_MSA.1(1)									O		O											X	X				
FMT_MSA.1(2)									O		O											X	X				
FMT_MSA.2	X																		X				X				
FMT_MSA.3(1)																			X				X				
FMT_MSA.3(2)																			X				X				
FMT_MTD.1(1)																						X	X				
FMT_MTD.1(2)																						X	X				
FMT_MTD.1(3)																						X	X				
FMT_MTD.1(4)																						X	X				
FMT_MTD.1(5)																						X	X				
FMT_MTD.1(6)																						X	X				
FMT_REV.1(1)																							X				
FMT_REV.1(2)																							X				
FMT_SMF.1																											
FMT_SMR.1																		X									
FPT_AMT.1																											
FPT_RVM.1																											
FPT_SEP.1																											
FPT_STM.1																											
FPT_TDC.1																											
FTP_ITC.1																											

Remarks

The dependencies of FMT_MSA.1 and FMT_MSA.3 on FMT_SMF.1 are defined in version 2.3 of the Common Criteria, which has been introduced after the release of LSPP and CAPP. These dependencies have been considered here.

The multiple instantiations of FMT_MTD.1 and FMT_REV.1 have been included in this table, because a multiple instantiation of one security functional requirement may in some cases result in the requirement for multiple instantiations of depending requirements. This is not the case here, because they all rely on the same simple role model of the TOE.

For the iterations of FMT_MSA.3, dependencies to FMT_MSA.1 are satisfied by the respective DAC- or MAC-related iteration.

The dependencies for the multiple instantiations of FCS_CKM.1, FCS_CKM.2 and FCS_COP.1 on FCS_CKM.4 (Cryptographic key destruction) have not been resolved because cryptographic session keys for the SSL/TLS 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.

All dependencies on FTP_TRP.1 are optional and are resolved by the inclusion of a corresponding optional component. FTP_TRP.1 therefore does not need to be included as a security functional requirement.

FPT_AMT.1 is satisfied by the TOE environment.

This table shows that no other unresolved dependencies exist between security functional requirements.

There are also no unresolved dependencies between security assurance requirements. This is because the evaluation assurance level EAL4 has been defined such that no unresolved dependencies exist. ALC_FLR.1 has no dependencies. Therefore there are no unresolved dependencies for assurance components.

8.2.7 Strength of function

This Security Target claims a SOF rating of SOF-medium. This claim applies for FIA_SOS.1, whereby it is stated that a 'one off' probability of guessing the password in 1,000,000 is given. The SFR is in turn consistent with the security objectives. A claim of SOF-medium is also consistent with the assumption of a non-hostile user community and the assumption on physical protection which prevents well-skilled, hostile attackers from getting physical access to the TOE.

No strength of function analysis is performed for the cryptographic algorithms supported by the TOE as well as the process of the generation of the keys used by those cryptographic algorithms.

8.2.8 Evaluation assurance level

This Security Target claims EAL4 augmented with ALC_FLR.1, which is considered appropriate for a well-controlled, non-hostile environment.

8.3 TOE summary specification rationale

8.3.1 Security functions justification

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.

Table 8-9: Mapping security functional requirements to security functions

SFR	Security Functions
FAU_GEN.1	Section 6.6.1 explains how audit records are generated. This section also explains the structure of the audit records.
FAU_GEN.2	Section 6.6.1 explains the information contained in the audit records. Tools to export audit records in human-readable format are mentioned in Section 6.6.1.
FAU_SAR.1	Section 6.5.1.4.1 explains the auditor role. Section 6.6.2 describes the purpose of the audit dump program that reads audit records from the audit trail and stores them in a data set where they can be evaluated.
FAU_SAR.2	Section 6.6.2 explains how to protect the audit trail from unauthorized access.
FAU_SAR.3	Section 6.6.1 explains how to search the audit records.
FAU_SEL.1	Sections 6.6.3 and 6.5.1.4.1 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 6.6.2 explains how to protect the audit trail from unauthorized access.

SFR	Security Functions
FAU_STG.3	Section 6.6.2 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 6.6.2 explains how the TOE prevents the loss of audit data by halting the system on audit trail exhaustion.
FCS_CKM.1(1) FCS_CKM.2(1) FCS_CKM.2(2) FCS_COP.1(1) FCS_COP.1(2)	Sections 6.4 and 6.3.2.5 explain the use of the SSL/TLS protocols for the protection of communication links.
FCS_CKM.1(2) FCS_CKM.2(3) FCS_COP.1(3)	Section 6.5.2.5 explains the use of the IPSec protocol for the protection of communication links by reference to the appropriate IETF standards.
FCS_COP.1(4)	Section 6.3.2.5 explains the use of HMAC-SHA-1 for the integrity protection as part of the IPSec protocol for the protection of communication links by reference to the appropriate IETF standards.
FDP_ACC.1	The general operation of access control is explained in Section 6.3.1. The possible access rights for discretionary access control are explained in Section 6.3.4. The protected resources are explained in Section 6.3.2
FDP_ACF.1(1)	Discretionary access control for z/OS objects is explained in Section 6.3.2 and 6.3.4.2 listing all the different types of objects and the specifics of their access control mechanisms.
FDP_ACF.1(2)	Section s 6.3.4.1, 6.3.2.7, and 6.3.2.8 explain access control for z/OS UNIX objects.
FDP_ETC.1	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 6.3.2.
FDP_ETC.2	Export of labeled data is explained in Section 6.3.3.
FDP_IFC.1	The mandatory access control policy is explained in Section 6.3.3.
FDP_IFF.2	The mandatory access control policy is explained in Section 6.3.3.
FDP_ITC.1	Import of unlabeled user data is the inverse of export and is explained in the same sections as the export.
FDP_ITC.2	Import of labeled user data is the inverse of export and is explained Section 6.3.3.
FDP_RIP.1	Object reuse is described in Section 6.7.
Note 1	Object reuse is described in Section 6.7.
FDP_UCT.1 FDP_UIT.1	The use of the SSL/TLS and IPsec protocols is explained in Sections 6.4 and 6.3.2.5.
FIA_ATD.1	User attributes are defined in Sections 6.5.1.2 and 6.5.1.4.2.
FIA_SOS.1	The password specifics are defined in Section 6.2.2.
FIA_UAU.1	User authentication is explained in Section 6.2. The special case of the HTTP server that allows installation defined limited access for unauthenticates users is described in section 6.2.3.3.
FIA_UAU.7	Section 6.2.2 describes that passwords are not displayed when entered at a TSO terminal.
FIA_UID.1	User identification is explained in 6.2

SFR	Security Functions
FIA_USB.1	User subject binding for z/OS is explained in Section 6.2, which describes protected user IDs in Section 6.2.2.2. Specifics of the z/OS UNIX su command are explained in Section 6.2.4, exemptions for started tasks in Section 6.2.3.
FMT_MSA.1(1)	Management of object security attributes is explained in Section 6.5.2 where the different RACF profiles and their management is described. Sections 6.5.1.4 and 6.5.3 explain the RACF configuration.
FMT_MSA.1(2)	Management of security labels being restricted to users with the SPECIAL attribute is described in section 6.3.3
FMT_MSA.2	This aspect is explained together with the description of the individual attributes.
FMT_MSA.3(1)	Default values for the access control are defined in the UACC attribute in the resource profiles as explained in Section 6.5.2 in the description of the resource profiles.
FMT_MSA.3(2)	Default values for the security label are defined in the SECLABEL attribute in the resource profiles as explained in Section 6.5.2 in the description of the resource profiles.
FMT_MTD.1(1)	Audit trail management is explained in Section 6.6.2.
FMT_MTD.1(2)	Audit event management is explained in Section 6.6.3.
FMT_MTD.1(3)	Management of user attributes is explained in Section s 6.5.1.1 and 6.5.1.4.
FMT_MTD.1(4)	Management of authentication data is explained in Section 6.2.2.
FMT_MTD.1(5)	Management of cryptographic keys is explained in Section 6.4.
FMT_MTD.1(6)	Configuration and management of network security aspects is explained in 6.5.4
FMT_REV.1(1)	Revocation of user attributes is explained as part of the management of user attributes in Section 6.5.1.4.
FMT_REV.1(2)	Revocation of object attributes is explained as part of the management of access control to objects in Sections 6.3.2 (DAC) and 6.3.3 (MAC).
FMT_SMF.1	See SFRs FMT_MTD.1(1-6)
FMT_SMR.1	The roles are explained in Section 6.5.1.4.
FPT_AMT.1	The TOE hardware has extensive measures to check for the correct operation of the underlying z/Architecture.
FPT_RVM.1	The reference mediation property is explained in Section 6.8, with emphasis on Sections 6.8.2 and 6.8.3.
FPT_SEP.1	The separation mechanism within the hardware is explained in Sections 6.8.1 and 6.8.2. The separation of authorized programs from unauthorized programs is explained in Section 6.8.3.1.
FPT_STM.1	The time mechanism is explained in Section 6.8.1.1.
FPT_TDC.1	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 6.5 and the description of the extended attributes for z/OS UNIX file system objects in Section 6.8.4.4, which allows consistent interpretation of this data in different instantiations of the TOE.
FPT_ITC.1	The trusted channel is explained in Section 6.4.

8.3.2 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 LSPP 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.

LSPP 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 LSPP mode) security label. In addition the security management of the audit data (especially dumping the SMF data sets when they get full) also supports audit. Security management also includes the management of access rights including (in LSPP mode) the definition of the security labels and the definition how they get printed on a printer that supports multiple labels. 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 to delegate 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, 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 to identify security problems and allows to define 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 LSPP mode) mandatory access control.

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

8.3.3 Assurance measures justification

The assurance measures and how they are satisfied are explained in the table in Section 6.9. The authors of this Security Target view this table as sufficient justification for the individual assurance measures.

8.3.4 Strength of function

The password mechanism used for authentication is the only mechanism in the TSF that is implemented by a permutational or probabilistic mechanism subject to a strength-of-function analysis within the evaluation of this TOE. For the password-based authentication mechanism of the security function (see 6.2.2), a minimum strength of SOF-medium is claimed. This is done in accordance with the SOF claim for the related security functional requirement FIA_SOS.1. This claim is consistent with the security objective O.AUTHORIZATION and the statement in Section 3.3, which states that the TOE "protects against threats of inadvertent or casual attempts to breach the system security". A highly-skilled and well-funded attacker is explicitly excluded from the threat scenario described in Section 3.3.

The SOF-medium claim does not apply to the cryptographic algorithms, the process of generating keys for those cryptographic algorithms (including the random number generator), or the cryptographic hash functions implemented in the TOE. Excluding cryptographic algorithms and related functions from the strength of function analysis is in compliance with the [CEM], remarks on ASE_REQ.1.15, paragraph 424.

8.4 PP claims rationale

The TOE is conformant to the Labeled Security Protection Profile, as referenced in [LSPP], and to the Controlled Access Protection Profile, as referenced in [CAPP]. Conformance to CAPP is only claimed when the TOE is operated in CAPP mode.

One additional security objective for the TOE (O.COMPROT) has been defined to reflect the ability of the TOE to connect with trusted IT products through trusted channels. Objectives for the TOE environment have been added to this ST in addition to the ones contained in LSPP to allow a more distinguished description of the TOE environment; this does not impact the conformance of this ST to the PP.

Except for FCS_CKM.1, FCS_CKM.2, FCS_COP.1, FDP_UCT.1, FDP_UIT.1, FMT_MSA.2, FMT_SMF.1, FPT_TDC.1, and FTP_ITC.1, all security functional requirements in this ST are inherited from the LSPP and the operations allowed/required by the PP are performed and indicated in **bold.**

FMT_SMF.1 has been added to comply with CC version 2.3, which defines dependencies of two security functional requirements (FMT_MSA.1 and FMT_MTD.1) included in the PP. To satisfy those requirements, the new security functional component FMT_SMF.1 has been added to the Security Target (anticipating that this security functional requirement will be added in an update to the Labeled Security Protection Profile and the Controlled Access Protection Profile).

LSPP mode only: FPT_TDC.1 has been added to this Security Target as a result of an unresolved (and undiscussed) dependency already in LSPP.

FCS_CKM.1, FCS_CKM.2, FCS_COP.1, FDP_UCT.1, FDP_UIT.1, FMT_MSA.2, and FTP_ITC.1 have been added to address the ability of the TOE to set up a trusted channel to another trusted IT product using the SSLv3 or TLSv1 protocol. This protocol uses cryptographic functions to protect the trusted channel.

FPT_AMT.1 has been moved into the TOE's environment. The hardware implementing the TOE's underlying abstract machine provides extensive testing of the abstract machine that cannot be achieved from within the TOE, because many failure modes are intercepted at a level which does not affect the abstract machine's interface at all. Moving FPT_AMT.1 into the TOE environment for this TOE and its underlying hardware therefore provides all of the security required by CAPP and LSPP for this specific aspect.

Additional SFRs for the TOE IT environment have been defined to cope with the more distinguished description of the TOE environment. This does not impact the conformance of this ST to the PP.

End of document