Security Target for IBM z/VM Version 5 Release 3

Version 2.6

August 5, 2008

Table of Contents

1. Introduction ... 5
1.1 ST Identification ... 5
1.2 ST Overview ... 5
1.3 CC Conformance ... 5
1.4 Strength of Function ... 6
1.5 Structure ... 6
1.6 Terminology ... 6
1.7 Abbreviations ... 8
1.8 References ... 8
1.9 Trademarks ... 9
2. TOE Description ... 10
2.1 Structure and concept of z/VM ... 11
2.1.1 Differences to other general purpose operating systems ... 11
2.1.2 z/VM's Kernel and non-kernel software ... 12
2.1.3 User's management of virtual machines using the Control Program ... 12
2.1.4 Communication between virtual machines and the Control Program ... 13
2.2 Intended Method of Use ... 13
2.2.1 Conversational Monitor System (CMS) ... 14
2.3 Summary of Security Features ... 15
2.3.1 Identification and Authentication ... 15
2.3.2 Discretionary Access Control ... 15
2.3.3 Mandatory Access Control and Support for Security Labels in LSPP mode ... 16
2.3.4 Separation of virtual machines ... 16
2.3.5 Audit ... 16
2.3.6 Object reuse functionality ... 16
2.3.7 Security Management ... 16
2.3.8 TSF Protection ... 17
2.4 Configurations ... 17
2.4.1 Software Components ... 17
2.4.2 Software Configuration ... 18
2.4.3 Hardware configurations ... 18
3. TOE Security Environment ... 19
3.1 Introduction ... 19
3.2 Assumptions ... 19
3.2.1 Physical Assumptions ... 19
3.2.2 Personnel Assumptions ... 19
3.2.3 Procedural Assumptions ... 20
3.2.4 Connectivity Assumptions ... 20
3.3 Threats ... 20
3.4 Organizational Security Policies ... 21
4. Security Objectives ... 22
4.1 Security Objectives for the TOE ... 22
4.2 Security Objectives for the TOE Environment ... 23
5. Security requirements ... 24
5.1 TOE Security Functional Requirements ... 24
5.1.1 Security Audit (FAU) ... 24
5.1.2 User Data Protection (FDP) ... 29
5.1.3 Identification and Authentication (FIA) ... 39
5.1.4 Security Management (FMT) ... 42
5.1.5 Protection of the TOE Security Functions (FPT) ... 46
5.1.6 Resource Utilisation (FRU) ... 48
5.2 TOE Security Assurance Requirements ... 48
5.3 Security Requirements for the IT Environment ... 48
5.3.1 Domain Separation (FPT_SEP.1) ... 48
5.3.2 Abstract Machine Testing (FPT_AMT.1) ... 49

6. TOE Summary Specification 50
6.1 Overview of the TOE architecture 50
6.2 F.AU: Auditing 51
6.2.1 F.AU.1 - Generation of Audit Records 51
6.2.2 F.AU.2 - Protection of the Audit Trail 51
6.2.3 F.AU.3 - Audit Configuration and Management 52
6.3 F.AC: Access Control 52
6.3.1 F.AC.1 - General Operation 52
6.3.2 F.AC.2 - Profiles 53
6.3.3 F.AC.3 - Access control enforcement 55
6.3.4 F.AC.4 - Access Control Configuration and Management 59
6.3.5 F.AC.5 - Protected Resources 60
6.3.6 F.AC.6 - Access control enforcement by CP 60
6.4 F.I&A: Identification and Authentication 61
6.4.1 F.I&A.1 - Identification and authentication mechanism 61
6.4.2 F.I&A.2 - Passwords 62
6.4.3 F.I&A.3 - Identity Change 62
6.5 F.IP: Interference Protection between virtual machines 62
6.6 F.OR: Object re-use 65
6.7 F.SM: Security Management 65
6.7.1 F.SM.1 - Management of user security attributes 65
6.7.2 F.SM.2 - Management of object security attributes 66
6.7.3 F.SM.3 - Management of audit 66
6.7.4 F.SM.4 - Management of system assurance testing 67
6.8 F.TP: TOE Self Protection 67
6.8.1 F.TP.1 - Supporting Mechanisms of the Abstract Machine 67
6.8.2 F.TP.2 - Structure of the TOE 69
6.9 Assurance Measures 70
6.10 Self-test functions 72
7. Protection Profile Claims 73
7.1 PP Reference 73
7.2 PP Tailoring 73
8. Rationale 75
8.1 Security Objectives Rationale 75
8.1.1 Complete Coverage – Organizational Security Policies 75
8.1.2 Complete Coverage – Environmental Assumptions 77
8.2 Security requirements Rationale 79
8.2.1 Internal Consistency of Requirements 79
8.2.2 Complete Coverage – Security Objectives 82
8.2.3 Security Requirements Instantiation Rationale 85
8.2.4 Explicit stated security requirements 87
8.2.5 Security Requirements Coverage 87
8.2.6 Rationale for Security Requirements for the IT Environment 89
8.2.7 Security Requirement Dependency Analysis 89
8.2.8 Strength of Function 91
8.2.9 Evaluation Assurance Level 92
8.3 TOE Summary Specification Rationale 92
8.3.1 Security Functions Justification 92
8.3.2 Mutual Support of the Security Functions 94
8.3.3 Strength of Function 95
8.4 PP Claims Rationale 95

Index of Tables

Table 5-1 Auditable Events 27
Table 6-1 RACF user profile 54
Table 6-2 RACF group profile 54
Table 6-3 RACF resource profile 55
Table 6-4 Communication channel usage 64
Table 6-5 Assurance Measures 71
Table 8-1 Mapping from OSP to objectives 76
Table 8-2 Mapping from objectives to OSP 77
Table 8-3 Mapping Non-IT Security Objectives to Environmental Assumptions and Policies 78
Table 8-4 Non-IT Security Objectives to Environmental Assumptions 79
Table 8-5 Mapping Security Objectives to Security Functional Requirements 83
Table 8-6 Mapping Security Functional Requirements to Objectives 89
Table 8-7 Mapping Security Functional Requirements for the IT Environment to Objectives 89
Table 8-8 IT Security Functional Requirements dependency analysis 91
Table 8-9 Mapping from SFR to Security Funtions 94

1. Introduction

This is version 2.6 of the Security Target document for IBM z/VM Version 5 Release 3.

1.1 ST Identification

Title:	Security Target for IBM z/VM Version 5 Release 3
Version:	2.6
Keywords:	access control, discretionary access control, general-purpose operating system, information protection, security labels, mandatory access control, security, virtual machine

This document is the security target for the CC evaluation of the IBM z/VM Version 5 Release 3 operating system product, and 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/VM Version 5 Release 3 operating system product with the additional required software products (see section 2.4 on page 17 of this ST) configured in a secure manner according to the supplied security guide.

IBM z/VM is a highly secure, robust, scalable, high-performance enterprise operating system on which to build and deploy mission-critical applications, providing a comprehensive and diverse application execution environment. IBM z/VM is the virtual machine operating system for IBM @server zSeries™ mainframe computers, empowering the use of their most advanced features such as providing separated virtual machines on top of 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/VM can be used on a single IBM @server zSeries™ mainframe computer. Several zSeries computers running the evaluated version of IBM z/VM can be connected to form a networked system. The communication aspects within IBM z/VM used for this connection are also part of the evaluation. External communication links can be protected against loss of confidentiality and integrity by cryptographic protection mechanisms not part of the TOE.

With its outstanding security features such as multilevel security support, IBM z/VM meets all of the requirements of the Labeled Security Protection Profile (LSPP) and the Controlled Access Protection Profile, both developed by the Information Systems Security Organization within the National Security Agency to map the TCSEC B1 (LSPP) and C2 (CAPP) class 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 those Protection Profiles and also includes additional functional and assurance packages beyond those required by LSPP and CAPP.

Although intended for the development of protection profiles, the manual on “Basic Robustness Environments”, release 2.0 from March 1, 2004 has been considered during the development of this Security Target. Modifications suggested in this manual are marked throughout this document.

1.3 CC Conformance

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

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.

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 2 is the 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 6.9 provides the Assurance Measures
- Section 7 provides the Protection Profile claims
- Section 8 provides the Rationale for the security objectives, security requirements and the TOE summary specification.

1.6 Terminology

This section contains definitions of technical terms that are used with a meaning specific to this document. Terms defined in the [CC] are not reiterated here, unless stated otherwise. This ST uses the following terms consistently with [LSPP]; they are described in this section to aid in the understanding by readers of this ST. Readers should be aware that some terms are used differently in other z/VM documents. The following glossary points out different usage where appropriate:

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; typical ones include read access and write access, which allow the reading and writing of objects respectively.

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: those which apply to the behavior of *authorized users* (termed *access* rules) and those which apply to the behavior of *authorized administrators* (termed *authorization* rules).

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 User

An authorized user is a *user* who has been properly identified and authenticated. These *users* are considered to be legitimate *users* of the TOE.

Authorized Administrator

An authorized administrator is an *authorized user* who has been granted the authority to manage the TOE. These *users* are expected to use this authority only in the manner prescribed by the guidance given them.

Category

See *security category*.

Control Program (IBM)

The Control Program provides the kernel or nucleus of z/VM running in supervisor state outside the SIE instruction environment. It controls and manages the SIE instruction provided by the underlying processor providing a restricted computing environment for the virtual machines.

Discretionary Access Control (DAC)

An *access control policy* that allows *authorized users* and *authorized administrators* to control *access* to objects on the basis of individual user identity or membership in a group (e.g., PROJECTA).

Logical Processor (IBM)

A logical processor (also called virtual processor) is a computing unit usable by virtual machines. Logical processors have the same behaviour as *real processors* and they are mapped to *real processors*. The main difference to *real processors* comes from the fact, that z/VM schedules logical processors, hence there can be less, more or equal numbers of logical processors compared to *real processors*.

Mandatory Access Control (MAC)

An *access control policy* that determines *access* based upon the sensitivity (e.g., SECRET) or *category* (e.g., PERSONNEL, MEDICAL) of the information being accessed and the access authority of the *user* attempting to access that information.

Mediation

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

Real Processor (IBM)

A real processor is either a physically installed processor (native mode) or a processor made available to the logical partition z/VM is running in (PR/SM mode). Real processors are all computing units available to and usable by the z/VM *Control Program*.

Seclabel

Synonym for *security label*

Sensitivity Label (IBM)

A specific marking attached to subjects or objects denoting the *security level.*

Security Category (IBM)

A special designation for data at a given level that indicates that only people properly briefed and cleared can receive permission for access to the information.

Security Label (IBM)

A name that represents the combination of a hierarchical level of *classification* (*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 also 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 term for *security level* in LSPP is classification (see *Security Level (LSPP)*)

Security Level (LSPP)

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

User

An individual attempting to invoke a service offered by the TOE.

1.7 Abbreviations

CC	Common Criteria
CP	Control Program
DAC	Discretionary Access Control
IPL	Initial Program Load
MAC	Mandatory Access Control
PSW	Program Status Word
PR/SM™	Processor Resource/Systems Manager™
RACF	Resource Access Control Facility
TOE	Target of Evaluation
TSP	TOE Security Policy

1.8 References

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

[CC] Common Criteria for Information Technology Security Evaluation, Version 2.3, August 2005, Parts 1 to 3

[CEM] Common Methodology for Information Technology Security Evaluation, Part 2 - Evaluation Methodology, Version 2.3, 2005

[CPPA] CP Planning and Administration, Version 5 Release 1.0, Document Number SC24-6083-00

[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

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

[RACFSAG] Resource Access Control Facility Security Administrator's Guide, Version 1 Release 10, Document Number SC28-1340-14

[SCG] Secure Configuration Guide

1.9 Trademarks

The e-business logo, IBM, IBM @server, IBM eServer, IBM logo, DirMaint, HiperSockets, Processor Resource/Systems Manager, PR/SM, RACF, S/390, Enterprise Systems Architecture/390, ESA/390, VM/ESA, z/Architecture, z/VM and zSeries are trademarks or registered trademarks of International Business Machines Corporation of the United States, other countries or both.

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

2. TOE Description

The Target of Evaluation (TOE) is the z/VM virtual machine operating system with the software components as described in section 2.4. z/VM is a general-purpose, multi-user, multi-tasking operating system designed for enterprise computing systems. z/VM can be used by multiple users simultaneously to perform a variety of functions requiring controlled, separated access to the information stored on the system. The TOE provides a virtual machine for each logged in user, separating the execution domain of each user from other users as defined in the virtual machine definitions stored in the system directory. In addition, the system directory contains access control information for privileged functions (such as codes delivered through the processors DIAGNOSE instruction). In addition to the system directory, RACF is employed to perform access control to resources.

For the purpose of this ST, the TOE is seen as one instance of z/VM running on an abstract machine as the sole operating system on the level of the abstract machine and exercising full control over this abstract machine regardless which software runs inside of virtual machines. This abstract machine can be provided by one of the following:

- a logical partition of an IBM zSeries machine (PR/SM)
- native mode (no PR/SM logical partition) on z800 and z900

The abstract machine itself is not part of the TOE, but belongs to the TOE environment. It is to be noted that although a z/VM instance can be run within a z/VM instance, the evaluated configuration is restricted to one z/VM instance running directly on an abstract machine as defined above. A z/VM instance running within a virtual machine is allowed, but this z/VM instance is not in an evaluated configuration (some security functionality may be implemented differently, in particular with respect to the usage of the processor's Start Interpretive Execution (SIE) instruction).

Multiple instances of the TOE may be connected with the instances sharing their RACF database. This can done by sharing the DASD (direct access storage device) volume keeping the RACF database between the different z/VM instances. Although sharing of one RACF database between z/VM and z/OS is technically feasible, it is explicitly excluded from this evaluation.

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

The TOE security functions (TSF) are provided by the z/VM operating system core (called Control Program – CP), by applications running within virtual machines, and by the Resource Access Control Facility (RACF), which is used by different services as the central instance for identification and authentication and for access control decisions. z/VM provides management functions that allow configuring the TSF and tailor them to the customer's needs.

Some elements have been included in the TOE which do not provide security functions, but run in authorized mode and could therefore, if misbehaved, compromise the TOE. Since these elements are substantial for the operation of many customer environments, they are included as trusted applications within the TOE.

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 Structure and concept of z/VM

z/VM presents an approach to computer operating systems. It provides each end user with an individual working environment known as a virtual machine. The virtual machine simulates the existence of a dedicated real machine, including server functions, storage, and input/output (I/O) resources.

But virtual machines support much more than just end users. Guest operating systems can run in virtual machines. z/VM can support multiple z/Architecture™ and Enterprise Systems Architecture/390® (ESA/390) operating systems. (To provide full support for z/Architecture guests, z/VM must be running on a zSeries server.) For example, multiple Linux and z/OS images can be run on the same z/VM system that is supporting z/VM applications and end users. As a result, application development, testing, and production environments can share a single physical computer.

2.1.1 Differences to other general purpose operating systems

z/VM is similar to other general purpose operating systems (such as UNIX or Windows) by implementing the concepts of:

- Users logging into the system and controlling software acting on behalf of the user
- Access control to memory objects or devices based on rules enforced on users and their associated group or (in LSPP mode) security label assignment
- Nucleus or kernel software running in a privileged and protected environment, controlling and enforcing rules upon subjects and objects
- Management of real and virtual memory and separation of address spaces between different virtual machines
- Scheduling of user software to run multiple software concurrently on one or more processors in a serialized manner

The major difference to those general-purpose operating systems is the concept of virtual machines implemented by z/VM. Upon login, each user is provided with a virtual machine that is capable of running arbitrary software. A virtual machine differs from application environment of other general-purpose operating systems the following:

- Predefined limits of processors (i.e. definition of logical processors whose number may differ from the number of real processors), processing time (i.e. processing power of logical processors), memory ranges accessible from inside the virtual machine and access to devices are enforced on every virtual machine.
- Virtual machines allow software to run in problem and supervisor state provided by the underlying processors restricted by the limits of the virtual machine z/VM defined by the administrator.
- Hardware can be virtualized. Access to hardware not dedicated for one virtual machine only is virtualized by the TOE (such access to the timer of the abstract machine is mediated by the TOE). Virtualized devices can be accessed the same way, as they would be accessed natively by software inside virtual machines.
- Hardware can be simulated. In some case there is no hardware, but the TOE simulates a hardware device (such as a virtual LAN adapter for providing virtual machines access to the virtual LAN maintained by the TOE). This device can be accessed like any other real hardware from inside the virtual machine using a device driver.
- Pre-defined processor instructions are simulated by the Control Program (CP) to ensure strict separation of virtual machines. z/VM defines the limits of each virtual machine. A set of parameters for the virtual machine environment is loaded into a table within the processor when z/VM passes control to a virtual machine. Whenever the processor detects that an instruction cannot be handled within this “interpreted environment”, it generates an interrupt into the z/VM kernel (CP) which then handles the

instruction by simulating the processor instruction in addition to well-defined sanity checks. To activate the interpreted environment, the processor provides the Start Interpretive Execution (SIE) instruction.

By using the processor's SIE instruction, the Control Program (CP) is capable of maintaining a m:n association between the number of real processors installed in the processor complex (native mode) or real processors allocated to the logical partition z/VM is running in (PR/SM mode), and the number of logical processors assigned to a virtual machine. Hence, it is possible to have more, less or equal numbers of logical processors configured than real processors available. This is implemented in CP by having a scheduler for virtual machines. In particular, this is implemented by utilizing the SIE's time mechanism, which allows CP to specify how long the SIE environment is executed by one processor. After the expiry of the timer, the processor's control is returned to CP.

2.1.2 z/VM's Kernel and non-kernel software

The z/VM Control Program (CP) is primarily a real-machine resource manager. CP provides each user with an individual working environment known as a virtual machine. Each virtual machine is a functional equivalent of a real system, sharing the real processor instructions and its functionality, storage, console, and input/output (I/O) device resources.

CP provides connectivity support that allows application programs running within virtual machines to exchange information with each other and to access resources residing on the same z/VM system or on different z/VM systems.

In order to create and maintain these rules (virtual machine definitions), additional management software is employed, that runs outside the CP, but is part of the TOE. Hence, each component of the management software runs within a virtual machine. The following list illustrates, which functionality runs within virtual machines:

- CMS (a general purpose operating system that is employed to run all the following software components within a virtual machine – see section 2.2.1 for details on the intended usage of CMS in the evaluated configuration)
- RACF (provides authorization and authentication services to CP and to other authorized CMS applications)
- TCP/IP stack application

It is to be noted that the TCP/IP stack application contains a Telnet service for users to log on via network and a terminal service (called console) for local log-ins. In particular, this Telnet service receives requests from the network and forwards them into CP using the DIAGNOSE processor instruction. CP generates a virtual console session as a memory object. All outgoing information is sent from the CP back to the Telnet service, which encapsulates the information in the Telnet protocol and sends to the client.

2.1.3 User's management of virtual machines using the Control Program

The login facility is provided by the Control Program (CP). CP validates the user's credentials and establishes the virtual machine environment. For logging into z/VM, users have to access a virtual machine operator's console (or simply "console"). Prior to the IPL (Initial Program Load) of an operating system, this console is always in the CP environment. After IPLing an operating system, the console is in the virtual machine environment unless the user specifically returns to the CP environment. Thus the console serves two purposes after IPL. In case two or more consoles are available for one virtual machine, one console can be used for communicating with CP and the other for communication with the software running inside the virtual machine. However, there can only be one "virtual operator console" to access CP.

After successful login using the console, CP provides a shell style prompt on the login terminal, in case no software is automatically loaded upon establishment of the virtual machine. This shell allows management of the virtual machine within the boundary defined by the virtual machine definition, including IPL of an operating system.

The virtual machine is initialized with an administrator-predefined virtual machine definition. During runtime of the virtual machine, the user of the virtual machine may be allowed to alter the virtual machine definition by using the console interface to the CP. These changes are not stored; hence they are in effect until the user logs off from his virtual machine.

Interfaces to the CP for software running in virtual machines are provided using processor instructions.

2.1.4 Communication between virtual machines and the Control Program

z/VM offers the following communication facilities:

- A Guest LAN provides a simulated Ethernet or zSeries HiperSockets network.
- VMCF (Virtual Machine Communication Facility) provides bidirectional communication channels.
- IUCV (Inter-User Communication Vehicle) offers bidirectional communication channels.
- CP commands MESSAGE (MSG), SMSG, and WARNING (WNG) provide unidirectional communication channels (users can send messages to each other, but there is no "reply" mechanism).
- VCTC (Virtual Channel-To-Channel) provides bidirectional communication channels.

All listed communication channels are established and maintained by CP. CP protects them against spoofing, eavesdropping and sniffing.

In addition to dedicated communication channels, CP allows the configuration of:

- Sharing of disk space between virtual machines (CP does not control the access to data stored in these shared devices but performs access control when initially linking to the disk; hence, the software inside the accessing virtual machines must have some sort of synchronization mechanism to avoid data inconsistencies on shared disk space).
- Sharing of memory between virtual machines. CP allows the following types of sharing:
    - Private memory: this memory is not shared
    - Shared exclusive write: shared memory is allocated once and accessible from virtual machines. Upon first write access, the complete memory area is copied to the write-accessible virtual machine memory. The copied memory is marked as private memory and access to the shared memory area is prohibited.
    - Shared write: a memory area is shared between virtual machines. All virtual machines with access have read and write access to this memory area.
    - Read only: a memory area is shared between virtual machines. However, all virtual machines with access have read-only access to this memory area.

It is to be noted that processor signaling using the SIGP processor instruction is limited to logical processors belonging to the signaling virtual machine. CP ensures that these signals do not traverse the virtual machine boundary.

2.2 Intended Method of Use

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

- Using Control Program (CP) commands from the virtual machine console accessible locally or remotely by Telnet connections via the Telnet service provided by the TCP/IP stack application running in a dedicated virtual machine.

- Access of resources assigned to this virtual machine (the operating system just “sees” those resources assigned to the virtual machine).
- Execution of a processor instruction by software running inside a virtual machine causing the SIE instruction to terminate and to return the processor control to the CP for simulating the instruction.
- Communication with CP from inside the virtual machine using the processor’s DIAGNOSE instruction.

All users of the TOE are assigned a unique user identifier (user ID). This user ID is used as the basis for access control decisions and for accountability purposes and 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 actions. After successful authentication, the user’s associated virtual machine is created based on the virtual machine definition. The virtual machine identifier is identical with the user ID. Hence, the virtual machine ID is used as a synonym to the user ID and managed identically by the TOE.

All TOE resources are under control of the TOE. The TOE mediates access of subjects to TOE-protected objects based on discretionary and/or mandatory access rights. Subjects in the TOE are called virtual machines. They are the active entities that may act on behalf of users. 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/VM recognizes administrative users with special authorizations. They are trusted to perform system administration and maintenance tasks, which includes configuration of the security policy enforced by the z/VM 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 audit system provided by z/VM 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 systems operating within the same management domain. All those systems need to be configured in accordance with a defined common security policy.

2.2.1 Conversational Monitor System (CMS)

CMS is used as operating systems for TOE applications (such as TCP/IP). The following information concludes that no functionality of CMS is security relevant as it can be considered as a form of library to mediate operations from the TOE applications to the CP.

CMS is a general-purpose operating system delivered with z/VM. It is to be used to run the TCP/IP application in a virtual machine. The RACF system can also be run on CMS. Customers can write their own applications to be run on CMS either using the native API or using the POSIX compatible OpenExtensions application programming interface.

Although being a general-purpose operating system, CMS offers no security functionality claimed in this document. Security functions are implemented by servers that run as applications on top of CMS. CMS uses CP communication channels (such as IUCVs) for ensuring the confidentiality and integrity of the communication with the servers. In addition, these communication channels ensure that the communication partner is really the expected partner (i.e. the communication channels ensure that when CMS assumes to communicate with the Shared File System server, it really speaks with it). Security functions such as listed the following are provided by servers:

- Access control and audit for the Shared File System (SFS)
- Access control for the Byte File System (BFS)

However, when using CMS to run TOE components, the following restrictions apply:

- CMS is configured to run TOE components individually in different virtual machines.
- Each CMS instance running a TOE component must only be used to run this component. No other service must be provided by this CMS instance.
- Each CMS instance running a TOE component must be restricted to be manageable by authorized users only.
- Each CMS instance running a TOE component must not use SFS (i.e. the virtual machine definition assigns only exclusive minidisks to the virtual machine).
- The virtual machine definition only assigns private memory for the virtual machines running CMS with a TOE component. It is allowed to boot CMS from the commonly shared read only code segment containing the CMS binary object code.

These restrictions allow considering CMS as a supporting library for this evaluation, since no security functionality required for the operation of the TOE is provided by CMS. CMS is only required to provide a computing environment for TOE applications.

2.3 Summary of Security Features

The primary security features of the product are:

- Identification and authentication
- Discretionary access control
- Mandatory access control and support for security labels in LSPP mode
- Separation of virtual machines
- Audit
- Object reuse functionality
- Security management
- 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.

2.3.1 Identification and Authentication

z/VM provides identification and authentication of users by the means of an alphanumeric user ID and a system-encrypted password. The following parts of the TOE perform identification and authentication independently:

- Control Program
- RACF

For performing identification and authentication, z/VM employs RACF managing resource profiles and user profiles.

2.3.2 Discretionary Access Control

For implementation of extended DAC rules, RACF provides the capability and flexibility as required by the evaluation compared to the usage of the system. Hence, the evaluated configuration of z/VM includes RACF. Basically, a user's authority to access a resource while operating in a RACF-protected system at any time is determined by a combination of these factors:

- User's identity and group membership

- User's attributes including group-level attributes
- User's group authorities
- Security classification of the user and the resource profile (this specified in section 2.3.3)
- Access authority specified in the resource profile

2.3.3 Mandatory Access Control and Support for Security Labels in LSPP mode

In addition to DAC, z/VM provides Mandatory Access Control (MAC) in LSPP mode, which imposes access restrictions to information based on security classification. Each user and each RACF controlled object can have a security classification specified in its profile. The security classification can be a security level and zero or more security categories. Security labels are maintained separately from privilege classes in RACF.

The access control enforced by the TOE ensures that users may only read labeled information if their security label dominates the information's label, and that they may only write to labeled information containers if the container's label dominates the subject's.

2.3.4 Separation of virtual machines

Operating system failures that occur in virtual machines do not normally affect the z/VM operating system running on the real processor. If the error is isolated to a virtual machine, only that virtual machine fails, and the user can re-IPL without affecting the testing and production work running in other virtual machines.

Supported by the underlying processor, the TOE restricts results of software failures (such as program checks) occurring in a virtual machine to this machine, thus not affecting other virtual machines or the CP.

Failures of CP that cannot be isolated to a particular virtual machine result in the abnormal termination ("abend") of the Control Program. In the event of such an abend, the system will re-initialize itself, if possible. Special abend code numbers are used to identify the specific reason for the abend.

2.3.5 Audit

The TOE provides an audit 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. The audit records generated by RACF are collected into files residing on disks that are protected from unauthorized modification or deletion by the DAC and (in LSPP mode) MAC mechanism.

2.3.6 Object reuse functionality

The TOE provides a facility clearing protected objects and storage previously used by virtual machines or the TOE itself prior to reassignment to other virtual machines or the TOE. This ensures confidentiality of data maintained either by the TOE or by virtual machines.

DASD devices and their derivatives (such as minidisks or temporary disks) are to be cleared manually by the administrator in accordance with the organizational policies. There is additional software support by the IBM Directory Maintenance Facility (DirMaint), which however is not part of this evaluation

2.3.7 Security Management

z/VM provides a set of commands and options to adequately manage the TOE's security functions. The TOE recognizes several roles that are able to perform the different management tasks related to the TOE's security:

- General security options are managed by security administrators.
- Management of MAC attributes is performed by security administrators in LSPP mode.

- Management of users and their security attributes is performed by security administrators. Management of groups can be delegated to group security administrators.
- Management of virtual machine definitions is performed by security administrators.
- Users are allowed to change their own password, their default group, and their user name.
- Users may choose their security label from the range defined in their profile at login time in LSPP mode.
- Auditors manage the parameters of the audit system (e.g. list of audited events) and can analyse the audit trail.

2.3.8 TSF Protection

The z/VM control program enforces integrity of its own domain. No virtual machine can access TOE resources without appropriate authorization. This prevents tampering with TOE resources by untrusted subjects.

Supportive to this functionality are hardware implemented facilities, namely the SIE instruction and the Set Address Limit facility provided by the underlying processor. Therefore the hardware and firmware components providing the abstract machine for the TOE are required to be physically protected from unauthorized access.

2.4 Configurations

2.4.1 Software Components

The Target of Evaluation, IBM z/VM Version 5 Release 3, requires the following software elements to be installed:

- Conversational Monitor System (CMS) for operating RACF and TCP/IP.
- Control Program (CP).
- RACF
- TCP/IP for z/VM
- RSU 5302
- PTF UM32174
- PTF UV61016

Apart from these required elements, the following optional elements may be used in the system without changing the security characteristics as described in this Security Target:

- SSL support for the network communication

The following description defines an unprivileged and a privileged user.

An unprivileged user is defined as a virtual machine which (these options are documented in the guidance):

- Has AT MOST the CP commands available in IBM-defined privilege class G (it may have fewer)
- Does not have SPECIAL, group-SPECIAL, CLAUTH, AUDITOR or group-AUDITOR, OPERATIONS or group-OPERATIONS authority to RACF
- Does not have COMSRV, DIAG88, DIAG98, DEVMAINT, MAINTCCW, or SETORIG options in its CP directory entry.
- Does not have access to the VM directory (source or object forms)
- Does not have read-write access to the PARM disk(s), or other system areas of CP-owned volumes
- Does not have read-write access to the source or object code of CP, CMS, RACF, or VM TCP/IP.

- Does not have read-write access to the RACF database.
- Does not have read-write access to the RACF audit trail.
- Does not have OBEY authority for VM TCP/IP or other form of administrative authority over a virtual machine that has any of the special privileges described above.

All other virtual machines are considered to be Trusted Users or Administrators. A Trusted User has access to additional sensitive resources, system services or commands, but cannot alter it's own configuration or bypass DAC controls of resources it does not own, change ownership of system resource, and cannot disable system MAC controls which is possible to an administrator.

2.4.2 Software Configuration

The TOE software components allow a broad range of configuration possibilities. However, to implement all security requirements, restrictions on the configuration must be made.

The Secure Configuration Guide provides instructions and constraints for the evaluated configuration.

2.4.3 Hardware configurations

The following assumptions about the technical environment of the TOE are made:

The TOE is running on the abstract machine defined as a z/Architecture compliant platform.

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

- all terminals and printers supported by the TOE
- all storage devices and backup devices supported by the TOE
- all network adapters supported by the TOE

3. TOE Security Environment

3.1 Introduction

The statement of 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 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 the TOE is intended to be used in. This includes information about the physical, personnel, procedural, and connectivity aspects of the environment.

The TOE is assured to provide effective security measures in a non-hostile environment only if it is installed, managed, and used correctly. The operational environment must be managed in accordance with assurance requirements documentation for delivery, operation, and user/administrator guidance. The following specific conditions are assumed to exist in an environment where the TOE is employed.

3.2.1 Physical Assumptions

The TOEs 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 which 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 cooperating 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 (e.g., printers, tape drives, disk drives) attached to the TOE, and marking a sensitivity label on all output generated.

3.2.4 Connectivity Assumptions

This ST contains no explicit network or distributed system requirements. However, it is assumed that the following connectivity conditions exist:

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 TOEs 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 which address the need to trust external systems or the communications links to such systems.

A.CONNECT

All connections to peripheral devices reside within the controlled access facilities. 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 are assumed to be adequately protected.

3.3 Threats

In compliance with the Labeled Security Protection Profile (LSPP), 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 **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 a server, including data in transit between virtual machines and external entities.

The **threat agents** can be categorized as either:

- unauthorized users of the TOE, i.e. individuals who have not been granted the right to access the system; or

- authorized users of the TOE, i.e. 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

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 organizational security policies described below are drawn from DoD Manual 5200.28-M (Techniques and procedures for Implementing, Deactivating and Evaluating Resource Sharing ADP Systems) it applies to many non-DoD environments.

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 which 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 on a basis of relative value to the organization and its interest to limit 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 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 comply with any organizational security policies identified. All of the identified organizational policies are addressed under one of the categories below.

4.1 Security Objectives for the TOE

The following are the IT security objectives:

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 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 which ensures that the organizational policies are enforced in the target environment.

O.NONINTERFERE

The TOE must ensure that software running in one virtual machine cannot interfere with software running in another virtual machine in a way that causes an interrupt or exception in this virtual machine, except for communication and interrupts mediated and authorized by the TOE.

O.NO_COMM

The TOE must ensure that no information can be transferred between different virtual machines, except for communication and interrupts mediated and authorized by the TOE.

O.PARTIAL_SELF_PROTECTION

The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering or unauthorized disclosure, through its own interfaces.

Application Note: This security objective has been derived from the manual on “Basic Robustness Environments”.

4.2 Security Objectives for the TOE Environment

The TOE is assumed to be complete and self-contained and, as such, is 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 which 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 which 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 requirements components in this profile were drawn from the LSPP and Part 2 of the CC (all security functional requirements not present in LSPP are listed in chapter 7). 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. These operations are indicated through the use of underlined (assignments and selections) and italicized (refinements) text. Operations that are performed for this ST required either by the protection profile or the CC are marked in green letters. SFRs added to this ST, which are not already present in LSPP or CAPP are marked green entirely.

All SFRs were drawn from CC Part 2, except Note1 (required from LSPP), FPT_SEP_(EXP).1 and FPT_SEP_(EXP).2. Hence these SFRs are explicitly stated.

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.*

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: 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 documented in the Administrative Guidance, along with recommendation on how manual auditing should be established to cover these events.

Rationale: 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.

Component	Event	Details
FAU_GEN.1	Start-up and shutdown of the audit functions.	SMF Record Type 81: RACF Initialization Record
FAU_GEN.2	None	

Component	Event	Details
FAU_SAR.1	Reading of information from the audit records.	SMF Record Type 80, Event type 2, qualifier 0, for the RACF SMF minidisks (RACFVM.301 and RACFVM.302, by default)
FAU_SAR.2	Unsuccessful attempts to read information from the audit records.	SMF Record Type 80, Event type 2 qualifier non-zero, for RACF SMF minidisks
FAU_SAR.3	None	
FAU_SEL.1	All modifications to the audit configuration that occur while the audit collection functions are operating.	SMF Record Type Type 80, event types 3 through 24
FAU_STG.2	None	
FAU_STG.3	Actions taken due to exceeding of a threshold.	Not applicable due to implementation (Note that the TOE switches automatically to another empty minidisk once the current minidisk used for auditing is full. The TOE is able to start a program defined in the audit configuration to process the audit records in the minidisk that got filled up)
FAU_STG.4	Actions taken due to the audit storage failure.	RACF will stop making security decisions (automatic SETRACF INACTIVE) until the SMF logs are cleared and SETRACF ACTIVE is issued
FDP_ACC.1	None	
FDP_ACC.1	None	
FDP_ACF.1	All requests to perform an operation on an object covered by the SFP (RACF).	SMF Record Type 80, Event type 2
FDP_ACF.1	All requests to perform an operation on an object covered by the SFP (CP).	SMF Record Type 80, event type 2. Classes are VMXEVENT and VMCMD
FDP_ETC.1 (LSPP)	All attempts to export unlabeled information.	SMF Record Type 80, event type 2. Class=VMXEVENT Event=UTLPRINT or SPLDUMP
FDP_ETC.2 (LSPP)	All attempts to export labeled information	Neither printers, nor SPXTAPE DUMP will export automatically labeled data
FDP_ETC.2 (LSPP)	Overriding of human-readable output marking. (Additional)	Neither printers, nor SPXTAPE DUMP will export automatically labeled data
FDP_IFC.2 (LSPP)	None	
FDP_IFF.2 (LSPP)	All decisions on requests for information flow.	SMF Record type 80, event type 2

Component	Event	Details
FDP_ITC.1 (LSPP)	All attempts to import user data, including any security attributes.	
FDP_ITC.2 (LSPP)	All attempts to import user data, including any security attributes.	
FDP_RIP.2	None	
Note1	None	
FIA_ATD.1	None	
FIA_SOS.1	Rejection or acceptance by the TSF of any tested secret.	SMF Record Type 80, event type 1
FIA_UAU.1	All use of the authentication mechanism.	SMF Record type 80 event type 0
FIA_UAU.7	None	
FIA_UID.1	All use of the user identification mechanism, including the identity provided during successful attempts.	SMF Record Type 80, Event type 1
FIA_USB.1	Success and failure of binding user security attributes to a subject (e.g. success and failure to create a subject).	SMF Record type 80, event 0 for VMBATCH profiles.
FMT_MSA.1	All modifications of the values of security attributes.	SMF Record Type 80 (generated by the RACF commands)
FMT_MSA.3	Modifications of the default setting of permissive or restrictive rules. All modifications of the initial value of security attributes.	SMF Record Type 80 (generated by the RACF commands)
FMT_MTD.1	All modifications to the values of TSF data.	SMF Record Type 80 (generated by the RACF commands)
FMT_MTD.1	All modifications to the values of TSF data.	SMF Record Type 80 (generated by the RACF commands)
FMT_MTD.1	All modifications to the values of TSF data.	SMF Record Type 80 (generated by the RACF commands)
FMT_MTD.1	All modifications to the values of TSF data.	SMF Record Type 80 (generated by the RACF commands)
FMT_REV.1	All attempts to revoke security attributes.	SMF Record Type 80 (generated by the RACF commands)
FMT_REV.1	All modifications to the values of TSF data.	SMF Record Type 80 (generated by the RACF commands)
FMT_SMF.1	None (covered in other management functions)	
FMT_SMR.1	Modifications to the group of users that are part of a role.	SMF Record Type 80 (generated by the RACF commands)
FMT_SMR.1	Every use of the rights of a role. (Additional / Detailed)	SMF Record Type 80

Component	Event	Details
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, therefore not audit record is produced
FPT_FLS.1	None	
FPT_RVM.1	None	
FPT_SEP.1	None	
FPT_STM.1	Changes to the time.	z/VM does not permit modification of the time except during system initialization.
FRU_FLT.1	None	

Table 5-1 Auditable Events

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: There are some auditable events which 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 not to record the attempted identity in cases where that attempted identity could be misdirected authentication data; for example when the user may have been out of sync and typed a password in place of a user identifier.

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

5.1.1.3 Audit Review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide authorized administrators with the capability to read all audit information from the audit records:

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

Application Note: The minimum information which must be provided is the same that which is required to be recorded in FAU_GEN.2.

The intent of this requirement is that there exists a tool for administrator to be able to access the audit trail in order to assess it. 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 which will satisfy all of these requirements.

Rationale: This component supports the O.AUDITING and O.MANAGE objectives by providing the administrator with the ability to assess the accountability information 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/VM.

5.1.1.4 Restricted Audit Review (FAU_SAR.2)

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

Application Note: By default, authorized administrators may be considered to have been granted read access to the audit records. The TSF may provide a mechanism which allows other users to also read audit records.

Rationale: 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: The ST must state the additional attributes that audit selectivity may be based upon (e.g., object identity, type of event), if any.

Rationale: 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: The ST must state the additional attributes that audit selectivity may be based upon (e.g., object identity, type of event), if any.

Rationale: 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.

5.1.1.7 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 in the audit trail.

Application Note: 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: This component supports the O.AUDITING objective by protecting the audit trail from tampering, via deletion or modification of records in it. Further it ensures that it is as complete as possible.

Application Note: Final Interpretation RI141 has been considered for this SFR.

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 SMF disks.

Application Note: For this component, an “alarm” is to be interpreted as any clear indication to the administrator that the pre-defined limit has been exceeded. The ST author must state the pre-defined 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 (e.g., audit trail 75% full). If the limit is adjustable by the authorized administrator, the ST should also incorporate an FMT requirement to manage this function.

Rationale: 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.

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 the audit system operator if the audit trail is full.

Application Note: The selection of “preventing” auditable actions if audit storage is exhausted is minimal functionality; providing a range of configurable choices (e.g., ignoring auditable actions and/or changing to a degraded mode) is allowable, as long as “preventing” is one of the choices. If configurable, then FMT_MOF.1 should be incorporated into the ST.

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

Application note: Final interpretation RI#202 would allow a selection of “prevent auditable events, except those taken by the authorized user with special rights", while LSPP and CAPP state “to be able to prevent auditable events, except those taken by the authorized administrator”. While RI#202 would not allow a system administrator to override the policy, LSPP and CAPP (as any useful system) allows for such an overwrite, giving the person responsible for the operation of the system the decision on the policy the system is going to enforce when the audit trail is full.

5.1.2 User Data Protection (FDP)

5.1.2.1 Discretionary Access Control Policy by RACF (FDP_ACC.1)

FDP_ACC.1.1 The TSF shall enforce the Discretionary Access Control Policy on software running inside of virtual machines acting on the behalf of users, and the following list of objects:

- Minidisks

- Real DASD volumes
- Restricted DCSS
- Restricted NSS
- Spool files
- Guest LANs
- Virtual Switches
- NJE network nodes
- CP-controlled printers
- Virtual point-to-point communication paths (IUCV, VMCF, APPC, virtual CTC, MSG, WNG, MSGNOH, SMSG)
- POSIX information database
- User authentication service
- RACROUTE macro
- CP real memory
- Alternate userids
- RACF database
- Human-readable security labels
- Virtual machine console
- System access
- Objects accessible through the following interfaces:
  - CP commands listed in table 4, Appendix A [SCG]
  - DIAGNOSE codes listed in table 5, Appendix A [SCG]
  - System functions listed in table 6, Appendix A [SCG]

and all operations among subjects and objects covered by the DAC policy.

Application Note: Interfaces that are marked as being mandatory DAC checked in the LSPP section of each table in Appendix A [SCG] are applicable to this DAC policy.

5.1.2.2 Discretionary Access Control Policy by CP (FDP_ACC.1)

FDP_ACC.1.1 The TSF shall enforce the Discretionary Access Control Policy on software running inside of virtual machines acting on the behalf of users, and the following list of objects:

- CP commands belonging to one or more privilege classes other than privilege class any
- DIAGNOSE code belonging to one or more privilege classes other than privilege class any or can be access restricted with a system directory statement
- Following processor instruction causing the SIE instruction to terminate:
  - IUCV processor instruction (0xB2F0)

and all operations among subjects and objects covered by the DAC policy.

Application Note: 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 which 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 above two 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: This component supports the O.DISCRETIONARY_ACCESS objective by specifying the scope of control for the DAC policy.

5.1.2.3 Discretionary Access Control Functions by RACF (FDP_ACF.1)

FDP_ACF.1.1 The TSF shall enforce the Discretionary Access Control Policy to objects based on the following:

a) The user identity and group membership(s) 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

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

or, if a) is not true,

b) if the requested type of access is allowed by an access authority for 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 the user is connected to.

or, if none of the above is true,

c) if the requested type of access is granted by the universal access authority (UACC) in the profile protecting the resource.

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

- Assignment the OPERATIONS attribute to users or groups allow access to any resource in a class defined in the Class Descriptor Table with OPER=YES (assigning attributes to groups provide the user with the same set of rights restricted to the scope of the group)
- By adding resource profiles to the global access table with a UACC other than NONE, this resource is always allowed access with the access level specified by the UACC.

Application Note: Other attributes, such as the SPECIAL, AUDITOR, or CLAUTH attributes, or the group authority of CONNECT/JOIN allow accessing the resource profile only. Only when changing these profiles to allow the user bearing these attributes access to the resource, access is granted. Therefore, these attributes do not overwrite the DAC policy specified here.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following rules:

- Assignment of the REVOKE attribute to users
- By adding resource profiles to the global access table with a UACC of NONE, this resource is always denied access to.

5.1.2.4 Discretionary Access Control Functions by CP (FDP_ACF.1)

FDP_ACF.1.1 The TSF shall enforce the Discretionary Access Control Policy to objects based on the following:

a) The user identity associated with a subject; and

b) The following access control attributes associated with an object: a privilege class.

Application Note: The membership of the user to groups defined in RACF is not applicable as the access control mechanism only uses the user ID for access validation.

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: if the user belongs to the same privilege class the CP command, DIAGNOSE code, or protected processor instruction is assigned to.

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

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

Application Note: A LSPP conformant TOE is required to implement a DAC policy, but the rules which 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 which 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; i.e., objects which are readable to all authorized users, but which 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 floppy 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.

Application Note: FDP_ACC.1 (RACF) applies to FDP_ACF.1 (RACF) and is implemented by the trusted application RACF. FDP_ACC.1 (CP) applies to FDP_ACF.1 (CP) and is implemented by the TOE kernel (Control Program).

Application Note: Both DAC mechanism implemented in RACF and CP are enforced on identical objects: the CP commands and DIAGNOSE codes listed in FDP_ACC.1 (RACF). The access check on those objects is sequential: first the CP check is being performed and RACF authorizes second. In case the CP check denies access, no further RACF check is performed. In contrast, if the CP check accepts the request from the user, RACF performs its access check. Only if both access checks succeed, the request is being allowed to proceed.

Application Note: The REVOKE attribute prevents a user from logging into the system.

Rationale: This component supports the O.DISCRETIONARY_ACCESS objective by defining the rules which will be enforced by the TSF.

5.1.2.5 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 TSC.

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

FDP_ETC.1.3 The TSF shall enforce the following rules when *unlabeled* user data is exported from the TSC:

a) Devices used 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: An LSPP-conformant TOE must provide protections to data exported outside the control of the TSC via any communications mechanisms that do not provide security attributes along with the actual data. The device, or mechanism, used to export information must, itself, 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 hard copy 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 (e.g., use of pre-labeled paper is allowable).

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: This component supports the O.MANDATORY_ACCESS objective by defining the rules which will be enforced by the TOE.

5.1.2.6 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 which 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: 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, etc.; 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: This component supports the O.MANDATORY_ACCESS objective by defining the rules which will be enforced by the TOE.

Application Note: The TOE does not provide multi-level devices, therefore this SFR stated in the LSPP does not apply. The TOE supports single-labeled printers only that need human interaction when changing the associated security label.

5.1.2.7 Mandatory Access Control Policy (FDP_IFC.1) (LSPP mode only)

FDP_IFC.1.1 The TSF shall enforce the Mandatory Access Control Policy on software running inside of virtual machines acting on the behalf of users, the following list of objects:

- Minidisks
- Real DASD volumes
- Restricted DCSS
- Restricted NSS
- Spool files
- Guest LANs
- Virtual Switches
- NJE network nodes
- CP-controlled printers
- Virtual point-to-point communication paths (IUCV, VMCF, APPC, virtual CTC, MSG, WNG, MSGNOH, SMSG)
- POSIX information database
- User authentication service
- RACROUTE macro
- CP real memory
- Alternate userids
- RACF database
- Human-readable security labels
- Virtual machine console
- System access
- Objects accessible through the following interfaces:
  - CP commands listed in table 4, Appendix A [SCG]
  - DIAGNOSE codes listed in table 5, Appendix A [SCG]
  - System functions listed in table 6, Appendix A [SCG]

and all operations among subjects and objects covered by the MAC policy.

Application Note: Interfaces that are marked as being mandatory MAC checked in the LSPP section of each table in Appendix A [SCG] are applicable to this MAC policy.

Application Note: 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 which 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, etc.

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: This component supports the O.MANDATORY_ACCESS objective by specifying the scope of control for the MAC policy.

5.1.2.8 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 via a controlled operation if the following rules, based on the ordering relationships between security attributes hold:

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

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

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

FDP_IFF.2.3 The TSF shall enforce the following additional information flow control SFP rules: security label SYSNONE excludes a user or resource from mandatory access control verification.

FDP_IFF.2.4 The TSF shall provide the following additional SFP capabilities: none.

FDP_IFF.2.5 The TSF shall explicitly authorize an information flow based on the following rules: None.

FDP_IFF.2.6 The TSF shall explicitly deny an information flow based on the following rules:

- Objects with the security label "no seclabel specified" cause all MAC access checks to fail for the corresponding subject or object.

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 super-set 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: The terms “security attribute” and “information flow control security attribute” refer to the sensitivity labels of subjects and objects.

A 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.

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

5.1.2.9 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: The LSPP-conformant TOE must provide protections for data imported from outside the control of the TSC via 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: This component supports the O.MANDATORY_ACCESS objective by defining the rules which will be enforced by the TOE.

5.1.2.10 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: The ST author must provide for the protection of data imported from outside the control of the TSC via 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: 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.

A 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: The TOE does not provide multi-level devices, therefore this SFR stated in the LSPP does not apply.

5.1.2.11 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: 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: This component supports the O.RESIDUAL_INFORMATION objective.

5.1.2.12 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: 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: This component supports the O.RESIDUAL_INFORMATION objective.

5.1.3 Identification and Authentication (FIA)

5.1.3.1 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;

e) Security-relevant Roles; and

f) default access rights for objects created by the user (UACC) in the user's default group

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

h) User's attributes including group-level attributes;

i) User's group authorities.

Application Note: 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.

Application Note: The attributes f) to i) are optional attributes that can be maintained for each individual user. These attributes present additional authorities and default values for the user. If they are not present, this user either does not have these authorities or system global default values are used.

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

5.1.3.2 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: 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: 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.

5.1.3.3 Authentication (FIA_UAU.1)

FIA_UAU.1.1 The TSF shall allow the following actions in addition to providing credentials and selection of security label on behalf of the user to be performed before the user is authenticated:

- Use of the LOGON and LOGOFF command

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: 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: This component supports the O.AUTHORIZATION objective by specifying what actions unauthenticated users may perform.

5.1.3.4 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: Obscured feedback implies the TSF does not produce a visible display of any authentication data entered by a user, such as through a keyboard (e.g., echo the password on the terminal). 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: This component supports the O.AUTHORIZATION objective. Individual accountability cannot be maintained if the individual's authentication data, in any form, is compromised.

5.1.3.5 Identification (FIA_UID.1)

FIA_UID.1.1 The TSF shall allow the following actions in addition to providing credentials and (in LSPP mode) selection of security label on behalf of the user to be performed before the user is identified:

- Use of the LOGON and LOGOFF command

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: 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: This component supports the O.AUTHORIZATION objective by specifying what actions unidentified users may perform.

5.1.3.6 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 which is associated with auditable events;

b) The user identity or identities which are used to enforce the Discretionary Access Control Policy;

c) The group membership or memberships used to enforce the Discretionary Access Control Policy;

d) The sensitivity label used to enforce the Mandatory Access Control Policy, which consists of the following (LSPP mode):

- A hierarchical level; and
- A set of non-hierarchical categories.

e) Attributes associated with the user or any of the user's groups.

The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of a user:

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

b) A started virtual machine executes with the user ID of the logged in user it has been defined for.

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

a) *A z/VM user can change his z/VM user ID with the DIAGNOSE code D4, provided the user is authorized to use this DIAGNOSE code and has been given explicit authorization to assume the identity of a given user.*

Application Note: 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.

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.

Rationale: This component supports the O.DISCRETIONARY_ACCESS, O.MANDATORY_ACCESS and O.AUDITING objectives by binding user identities to subjects acting on their behalf.

5.1.4 Security Management (FMT)

5.1.4.1 Management of Object Security Attributes (FMT_MSA.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,
- the CLAUTH attribute for the class the resource is assigned to,
- the owner of the resource profile of the named object,
- and users who have ALTER authority to the object.

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 a user with the SPECIAL attribute or the appropriate group-SPECIAL attribute.

Application Note: 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: This component supports the O.DISCRETIONARY_ACCESS and O.MANDATORY_ACCESS objectives by providing the means by which the security attributes of objects are managed by a site.

5.1.4.2 Static Attribute Initialization (FMT_MSA.3)

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.

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, the appropriate group-SPECIAL attribute, or the owner (non-LSPP mode only) 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: A 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.DISCRETIONARY_ACCESS and O.MANDATORY_ACCESS objectives by requiring that objects are properly protected starting from the instant that they are created.

5.1.4.3 Management of the Audit Trail (FMT_MTD.1)

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

Application Note: 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: 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.

5.1.4.4 Management of Audited Events (FMT_MTD.1)

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

Application Note: 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, etc.

It is an important aspect of audit that users 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: 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.

5.1.4.5 Management of User Attributes (FMT_MTD.1)

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: This component only applies to security attributes which 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: 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.

5.1.4.6 Management of Authentication Data (FMT_MTD.1)

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: 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 users 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 any user 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: This component supports the O.AUTHORIZATION and O.MANAGE objectives by ensuring integrity and confidentiality of authentication data.

5.1.4.7 Revocation of User Attributes (FMT_REV.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) Revocations/modifications made by an authorized administrator to security attributes of a user like the user identifier, user name, user group(s), user password or assigned security labels shall be effective the next time the user logs in.

Application Note: Many security-relevant authorizations could have serious consequences if misused, so an immediate revocation method must exist, although it need not be the usual method (e.g., The usual method may be editing the trusted users 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 users 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: This component supports the O.MANAGE objective by controlling access to data and functions which are not generally available to all users.

5.1.4.8 Revocation of Object Attributes (FMT_REV.1)

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 or (in LSPP mode) Mandatory Access Control policies.

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) The rules of the Mandatory Access Control policy (5.2.6) are enforced on all future operations; and

c) none.

Application Note: The DAC policy may include immediate revocation (e.g., Multics immediately revokes access to segments) or delayed revocation (e.g., most UNIX systems do not revoke access to already opened files). 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 documents in guidance documentation how revocation is enforced.

Rationale: This component supports the O.DISCRETIONARY_ACCESS and O.MANDATORY_ACCESS objectives by providing that specified access control attributes are enforced at some fixed point in time.

5.1.4.9 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 attribute management
- Authentication data management
- Audit event management

Rationale: This component supports the O.MANAGE objective by providing the management of security functions.

5.1.4.10 Security Management Roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles:

a) authorized administrator;

b) users authorized by the Discretionary Access Control Policy to modify object security attributes;

c) users authorized by the Mandatory Access Control Policy to modify object security attributes; (LSPP mode)

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

e) users authorized to perform administrative actions (SPECIAL or group-SPECIAL attribute in their profile)

f) RACF auditors (users that have the AUDITOR or group-AUDITOR attribute in their profile).

g) Operations roles (users with the OPERATIONS or group-OPERATIONS attribute).

h) users authorized to define profiles in a class (CLAUTH attribute in their profile for the particular class).

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

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

Application Note: Users configured with the SPECIAL attribute (either on user or group level) are to be considered as authorized administrator. However, in addition to the global administrator, the TOE maintains administrative roles limited in their scope and allowed tasks (such as an auditor is allowed to fully manage the audit functionality, but nothing else). Therefore the additional attributes are provided.

Rationale: This component supports the O.MANAGE objective.

5.1.5 Protection of the TOE Security Functions (FPT)

5.1.5.1 Failure with preservation of secure state (FPT_FLS.1)

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: software failure in a virtual machine, hardware failure of a hardware component.

Rationale: This component supports the O.NONINTERFERE objective by ensuring that no software instruction can interfere with the operation of other subjects, except through specifically configured communication channels.

5.1.5.2 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: 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.

Rationale: This component supports O.ENFORCEMENT and O.NO_COMM objective by ensuring that the TSP is not being bypassed and that no communication between subjects is allowed except if specifically configured.

5.1.5.3 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.

FPT_SEP_(EXP).1 The TSF shall maintain a security domain that protects it from interference and tampering by untrusted subjects initiating actions through its own TSFI.

FPT_SEP_(EXP).2 The TSF shall enforce separation between the security domains of subjects in the TOE Scope of Control.

Application Note: 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 CPU's which 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 layered application such as a graphic user interface system which maintains separate subjects.

Application Note: The TCP/IP application, containing the Telnet server, that is part of the TOE supports this SFR by separating multiple Telnet connections and forward those connections to virtual console provided by CP. This application ensures that established connections are always forwarded to the virtual console, where the identification and authentication took place. Hence this application therefore provides the maintenance of the subject-object binding of the TOE. To separate multiple Telnet connections, the TCP/IP application uses the TCP sequence and acknowledgment numbers.

Application Note: The explicitly stated SFR have been derived from the manual on "Basic Robustness Environments".

Rationale: This component supports O.ENFORCEMENT and O.NO_COMM objectives by ensuring that a TSF exists within the TOE and that it can reliably carry out its functions and that no communication between subjects is allowed except specifically configured. In addition, this component supports the O.PARTIAL_SELF_PROTECTION objective by maintaining a security domain for its own use.

5.1.5.4 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: 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.

Application Note: The TOE uses a hardware timer to maintain its own time stamp. This hardware timer is protected from tampering by untrusted subjects. The start value for this timer may be set by the system administrator, but the system administrator may also start a program that uses an external trusted time source to set this initial value.

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

5.1.6 Resource Utilisation (FRU)

5.1.6.1 Degraded fault tolerance (FRU_FLT.1)

FRU_FLT.1.1 The TSF shall ensure the operation of programs executing in other virtual machines when the following failures occur: software failure in a virtual machine other than the RACF or TCP/IP virtual machines, and hardware failure of a hardware component not required for the operation of the TOE.

Rationale: This component supports the O.NONINTERFERE objective by ensuring that no software instruction can interfere with the operation of other subjects, except through specifically configured communication channels.

5.2 TOE Security Assurance Requirements

The target evaluation assurance level for the product is EAL4 [CC] augmented by ALC_FLR.2.

5.3 Security Requirements for the IT Environment

The assumptions stated for the environment need to be satisfied by the IT environment. It is expected that any system integrating the TOE will provide documentation and procedures as well as technical measures (e.g. within the host system) to demonstrate that the assumptions are fulfilled and the policies are implemented.

The SFR for the IT environment, FPT_SEP_ENV.1, is an explicit stated SFR, which was not derived from CC Part 2.

The only IT environment where requirements are stated for are the underlying processor, 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 Domain Separation (FPT_SEP.1)

FPT_SEP_ENV.1 The TSF Environment shall provide hardware that provides virtual memory management and at least two execution rings for executing software.

Application Note: The SIE processor instruction causes the processor to adhere to memory definitions supplied with the invocation of SIE. The processor only allows access to the defined memory. The "SIE mode" sets special purpose registers in the processor, which are not visible to any application running on the processor. Also, the SIE instruction is capable of returning the processor control to the caller of the SIE instruction in case a predefined privileged instruction was executed by software running under the SIE instruction restriction.

Application Note: ART (access register translation) and DAT (dynamic address translation) are performed by the underlying processor for translating virtual addresses into real addresses for the address space referenced by the user from inside a virtual machine.

Application Note: The abstract machine supports the flagging of memory pages as read only memory enforced upon subjects. Privileged subjects have the ability to alter this flag to get write access to the memory page. Subjects are authorized if their storage key maintained in their PSW equals zero.

Application Note: When using the LPAR mechanism of the abstract machine, this LPAR mechanism must also support the domain separation by preventing access to any object managed by the TOE from software in other logical partitions. This supports the differentiation into two execution rings.

Application Note: The explicitly stated SFR have been derived from the manual on “Basic Robustness Environments”.

5.3.2 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: 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, then 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.

6. TOE Summary Specification

6.1 Overview of the TOE architecture

z/VM is an operating system operating on IBM zSeries architecture processors. Those processors provide the Start Interpretive Executive (SIE) environment and memory protection functions that allow z/VM to prohibit direct access from untrusted virtual machines to I/O devices used by other virtual machines, protected memory areas used by the TOE and memory areas used by other virtual machines. The underlying firmware also allows defining separate logical partitions allowing execution of several instances of the TOE on the same hardware as well as having the TOE execute in one logical partition while other non-TOE software is executing in other logical partitions. The logical partitioning function is part of the TOE environment and has been evaluated separately.

The TOE itself provides interfaces to applications and users allowing them to request TOE services.

The TOE provides the following security functions:

1. An audit trail for security relevant events (F.AU)
2. Discretionary and (in LSPP mode) Mandatory access control (F.AC)
3. Identification & authentication (F.I&A)
4. Interference Protection between virtual machines (F.IP)
5. Object re-use (F.OR)
6. Security management functions to administer audit, discretionary access control and (in LSPP mode) mandatory access control as well as users and groups with their related attributes (F.SM)
7. TOE self protection functions based on security features provided by the underlying hardware including memory protection and the provision of a privileged state allowing the TOE to reserve and protect a domain for its own execution (F.TP)

The TOE itself is structured into the following major units:

1. The Control Program (CP) responsible for handling virtual machine environments, interrupts, logical processor scheduling, memory management including the management of address spaces.
2. The Communication Server responsible for network communication using TCP/IP based protocols (the TCP/IP stack application also provides the Telnet service)
3. The Resource Access Control Facility (RACF) as the central system for discretionary and mandatory access control to resources

The TOE itself consists of a “nucleus” operating in the supervisor state and outside the SIE instruction environment of underlying abstract machine and a set of “trusted applications” that operate in dedicated virtual machines communicating with the nucleus over dedicated communication channels. Those trusted applications are granted access to specifically restricted interfaces provided by CP. The functionality behind these interfaces provides the capability of overriding or modifying system security policies. Therefore all trusted applications allowed to be executed in the evaluated configuration are considered to be part of the TOE.

Trusted applications are executed in virtual machines dedicated for this task, i.e. no other functionality must be present in the respective virtual machine. These dedicated virtual machines are separated from other virtual machines using the security functionality provided by the nucleus. In addition, all storage area configured for these virtual machines are dedicated, hence no other virtual machine can access any portion of this storage area. Communication between trusted applications and the nucleus is established using the communication channels provided by the nucleus.

6.2 F.AU: Auditing

6.2.1 F.AU.1 - Generation of Audit Records

The TOE provides a general facility to collect data required for auditing. This function provided by RACF collects and records system audit data.

This component is used by the TOE to collect also security related audit 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. SMF supports up to 256 different record types where record types 0 to 127 are reserved for the Control Program.

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 several 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:

- The record type
- Time stamp (time and date)
- System identification
- Event code and qualifier
- User identification
- Group name
- A count of the relocate sections
- Authorities used to successfully execute commands or access resources
- Reasons for logging
- Command processing error flag
- Foreground user terminal ID
- Foreground user terminal level number
- Job log number (job name, entry time, and date)
- RACF version, release and modification number
- SECLABEL of user (in LSPP mode)

Each record contains further data specific to the event code and qualifier.

6.2.2 F.AU.2 - Protection of the Audit Trail

RACF writes SMF audit records into dedicated CMS files that have been defined during system configuration. At least two minidisks must be defined holding the CMS files. Those CMS file need to be protected against unauthorized access by appropriate RACF profiles.

At initialization, RACF uses the SMF CONTROL file to determine on which of two minidisks to record SMF records. When RACF fills up the minidisk on which it began recording, it uses the SMF CONTROL file to determine the location of the alternate minidisk. When it switches minidisks, RACF updates the CURRENT field in the SMF CONTROL file (on RACF's A-disk) to reflect the minidisk that it is now recording on.

For archiving SMF audit records once the SMF minidisk fills up, RACF executes SMFPROF to archive the data to another location.

If no non-full minidisk is found, RACF will disable itself and all requests to access protected resources will fail. Only certain users will be permitted to logon and access resources for the purposes of clearing the system logs and re-enabling RACF. Once RACF is re-enabled, normal processing resumes.

6.2.3 F.AU.3 - Audit Configuration and Management

The system can be configured to halt on exhaustion of audit trail space in order to prevent audit data loss. Operators are warned when audit trail space consumption reaches a pre-defined threshold. With the initial configuration, RACF continues operation even if the SMF disk space is exhausted. Setting the SEVER keyword to YES, RACF severs the path between CP and RACF when the SMF disks are full, and RACF is unable to continue recording SMF records. To manage the audit subsystem in this state, the TOE provides an administrative ID for RACF that can log into the system without RACF being online. The credentials for this user are stored in the system directory.

RACF always generates audit records for events like unauthorized attempts to access the system or changes to the status of the RACF database. The security administrator, auditors and non-SPECIAL users with appropriate authorization can configure which additional optional security events are to be logged. In addition to writing records to the audit trail, messages can be sent to the security console to immediately alert operators of detected policy violations. RACF writes records for detected, unauthorized attempts to enter the system. Optionally, RACF writes records to SMF for authorized attempts and/or detected, unauthorized attempts to:

- Access RACF-protected resources
- Issue RACF commands
- Modify profiles on the RACF database

RACF writes SMF records to a CMS file. To list SMF records, either the RACF report writer or the RACF SMF data unload utility (IRRADU00) can be used. With the report writer, RACF SMF records can be selected to produce the reports. With the SMF data unload utility, RACF SMF records can be translated into a browsable format or uploaded to a database, query, or reporting package, such as DB2.

RACF sends messages to the security console for detected, unauthorized attempts to enter the system and for detected, unauthorized attempts to access RACF-protected resources or modify profiles on the RACF database. The security console is the user defined in RACF CSTCONS macro (OPERATOR by default). As well as sending resource access violation messages only to the security console, RACF can send a message to a RACF-defined VM user. Each resource profile can contain the name of a user to be notified when RACF denies access to the resource. If the user is not logged on to the system at the time of the violation, the user receives a reader file that contains the notification information.

If access attempts are audited, and if the RACF function that issues a warning message instead of failing an invalid access attempt is selected (to allow for a more orderly migration to a RACF-protected system), RACF records each attempted access. For each access attempt that would have failed, RACF sends a warning message (ICH408I) to the accessor, but allows the access. If a "notify" user is specified in the resource profile, RACF also sends a message to that user. If you are deferring access authorization to VM through the use of the SYSSEC macro, and are auditing access attempts, RACF writes SMF records for access attempts that would have failed if you were not deferring.

6.3 F.AC: Access Control

6.3.1 F.AC.1 - General Operation

z/VM provides the Resource Access Control Facility (RACF) as the component that performs access control between software running in virtual machines 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 stored in the RACF database to decide if a subject has access to a resource. In addition to RACF, CP itself provides discretionary access control to CP commands and DIAGOSE codes, which is documented in section 6.3.6.

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

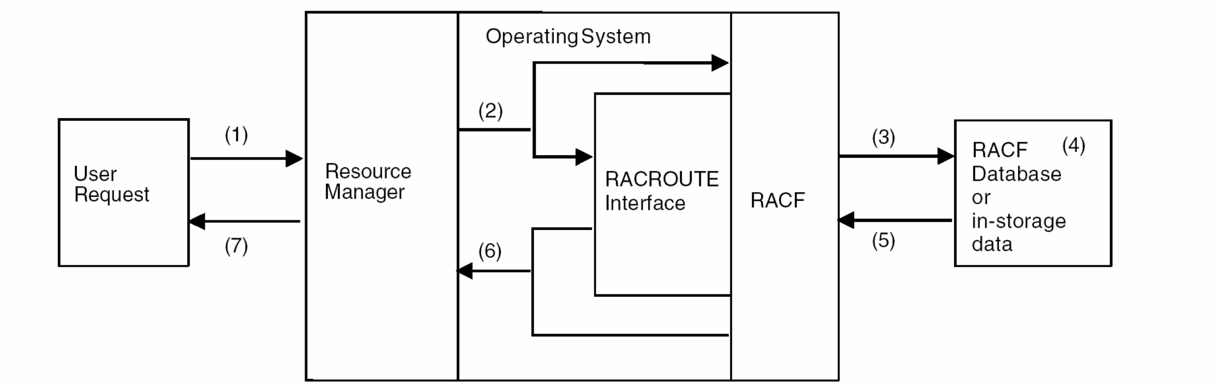

Figure 1: RACF and its relationship to the operating system

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

CP calls the RACF component using the internal interface to RACF (the *RPI interface that connects to the RACROUTE interface) to check the access rights of the user that initiated the user request and passes the ID of the user and user attributes like the security label (in LSPP mode), the name and type of the resource and the requested type of access to RACF (2). In addition to the RACROUTE interface, RACF also provides a resource check interface to CP to communicate more complex access control questions to RACF. As this resource check interface also transports queries to RACF, it is considered to be structurally equivalent as the RACROUTE interface.

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

RACF returns either a “yes” or a “no” decision for the access request in case the user and the resource are both known to RACF. If either of them is not known RACF returns a “don’t know” return code (6). In the later case 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).

6.3.2 F.AC.2 - Profiles

RACF makes access decisions based on information stored in profiles. RACF manages the following profiles:

- User profiles
- Group profiles
- General resource profiles

6.3.2.1 User Profiles

A user profile within RACF contains the following data:

Name	Description
USERID	User’s identification (maximum 8 characters)
NAME	User’s name (not security relevant, since the user is allowed to change his name)
OWNER	Owner of the user’s profile

Name	Description
DFLTGRP	User's default group (a user may change his default group to any group he is connected to)
AUTHORITY	User's authority in the default group (use, create, connect, join)
PASSWORD	User's password (Userid DES encrypted using the password - padded with blanks) as a key.
REVOKE	Date on which RACF prevents the user from having access to the system (also an indicator if the user completely revoked)
RESUME	Date on which RACF lets the user have access to the system again
UACC	Default universal access authority for resource profiles that the user defines. Only applicable to DATASET and a few general resource classes).
WHEN	Days of the week and hours of the day during which the user has access to the system (applies only to login via 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
SECLABEL	User's default security label

Table 6-1 RACF user profile

Note that there is other security relevant user data that is not stored in the RACF user profile but in the user's VM directory entry.

6.3.2.2 Group Profiles

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
TERMUACC or NOTERMUACC	The group's Terminal Authorization
GID	the group's OpenExtension group identifier

Table 6-2 RACF group profile

6.3.2.3 General Resource Profiles

A general resource profile – also called universal access authority (UACC) – in 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 profile

Name	Description
GENERIC or MODEL or TAPE	indicates if it is a generic, a model or a tape profile
OWNER	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 protected by the profile
AUDIT	The type of auditing to be performed for the resource protected by the profile
CATEGORY	The security categories to be assigned to the resource protected by the profile
SECLABEL	The security label of the resource protected by the profile
SECLEVEL	The security level of the resource protected by the profile
ACLs	Access control information (see definition below on the content of an individual ACL)

Table 6-3 RACF resource profile

Attributes within an ACL 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(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.
  - WHEN(DAYS(day-info))
  - WHEN(TIME(time-info))

UACC applies to all users, whether they are RACF-defined or not. If no access type for a UACC is defined, RACF uses NONE as a user's default universal access authority.

The default security label is “no seclabel specified”. This security label causes all MAC access checks to fail for that subject or object.

6.3.3 F.AC.3 - Access control enforcement

Basically, a user's authority to access a resource while operating in a RACF-protected system at any time is determined by a combination of these factors:

- User's identity
- User's attributes including group-level attributes
- User's group authorities
- Security classification (in LSPP mode)
- The access authority specified in the resource profile

6.3.3.1 User identity

A z/VM user is identified by an alphanumeric user ID that is associated with the user by RACF. Note, however, that a user need not be an individual. For example, a user ID can be associated with a disconnected service machine. In addition, in many systems today a "user" is equated with a function, rather than an individual. For example, a service bureau customer may comprise several people who submit work as a single user. Their jobs are simply charged to a single account number. From the security standpoint, equating a user ID with anything other than an individual can be undesirable because individual accountability is lost. It is up to the installation, to decide how much individual accountability is required. When defining a user, the administrator assigns a 1- to 8-character user ID. With this user ID, the user logs on to the system (or submits a batch job). When a user attempts to access RACF-protected resources, RACF uses the user ID to determine the user's access to those resources.

A RACF group is normally a collection of users with common access requirements. As such, it is an administrative convenience, because it can simplify the maintenance of access lists in resource profiles. By adding a user to a group, user access is given to all the resources that the group has access to. Likewise, by removing a user from a group, the user is prevented from accessing those resources. Individual users can be connected to any number of groups. Membership and authority in these groups can be used to control the scope of a user's activity. Each user must be assigned (connected) to at least one group (called the user's default group).

6.3.3.2 User's attributes

The administrator can assign attributes to each RACF-defined user. The attributes determine various extraordinary privileges and restrictions a user has when using the system. Attributes are classified as either user-level attributes (or, simply, user attributes) or group-level attributes. User attributes override DAC and MAC rules (except explicitly stated).

SPECIAL Attribute

A user with the SPECIAL attribute in his user profile is regarded as a system administrator. He can:

- add, delete and modify user, group, DATASET and other profiles
- define RACF general options (except options related to auditing)

Group-SPECIAL

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 have then administrative capabilities within the scope of the group they belong to. Users with the attribute group-SPECIAL cannot define general RACF options using the SETROPTS command (except for the REFRESH GENERIC, REFRESH RACLIST and LIST operands).

AUDITOR Attribute

A user with the AUDITOR attribute can define and modify the audit related options in user, group and resource profiles. This allows him to define which activities are to be recorded in the audit trail. The AUDITOR attribute at the system level gives the user the authority to specify logging options on the ALTUSER, RALTER, SETROPTS, ALTDIR and ALTFILE commands. In addition, the auditor can list auditing information with the LISTGRP, LISTUSER, RLIST, SEARCH, LDIRECT, LFILE, SRDIR, and SRFILE commands and the IRRUT100 utility program.

The user with the AUDITOR attribute 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)
- CMDVIOL or NOCMDVIOL
- LOGOPTIONS (for each profile class)
- OPERAUDIT or NOOPERAUDIT

- SAUDIT or NOSAUDIT
- SECLABELAUDIT or NOSECLABELAUDIT (in LSPP mode)
- SECLEVELAUDIT or NOSECLEVELAUDIT (in LSPP mode)

Audit configuration can also be delegated at the group level by giving the group-AUDITOR attribute to a user.

Group-AUDITOR

A user with the group-Auditor attribute can define and modify the audit related options in user, group and resource profiles in his group. The user's authority is limited to profiles that are within the scope of that group.

OPERATIONS Attribute

A user with the system-OPERATIONS attribute has full authorization to all RACF-protected resources in the following classes:

- VMBATCH
- VMCMD
- VMMDISK
- VMNODE
- VMRDR

However specifically configured access control lists for the resources to be accessed and MAC rules have precedence over this attribute.

Group-OPERATIONS

The group-OPERATIONS user's authority is restricted to resources within the scope of the group.

CLAUTH Attribute

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

- SPECIAL or NOSPECIAL
- AUDITOR or NOAUDITOR
- OPERATIONS or NOOPERATIONS

The CLAUTH attribute is assignable on a class-by-class basis; hence it cannot be assigned at the group level.

REVOKE attribute

RACF prevents user from entering the system when the user is assigned the REVOKE attribute. The REVOKE attribute can also be assigned on a group level by using the CONNECT command. If the user has the REVOKE attribute for a group, the user cannot enter the system by connecting to that particular group, or access resources as a member of that group. RACF allows specifying a future date for a REVOKE to occur (at both the system and the group level). Also a future date to remove the REVOKE attribute by using the RESUME operand can be specified.

6.3.3.3 User's group authorities

The administrator can assign a specific level of "group authority" to each user of a group. The group authorities are:

- USE – the user can access resources to which the group is authorized to
- CONNECT – access rights of USE, and ability of connect other users to the group and assign USE or CONNECT authorities
- JOIN – access rights of CONNECT, and the ability to define new users and groups and assign any level of group authority. To define new users, the users with JOIN authority must also have the CLAUTH user attribute for the USER class. When a user defines a new group, it becomes a subgroup of the group in which the user has JOIN authority.

6.3.3.4 Security classification (LSPP mode)

Label based mandatory access control is supported by z/VM using RACF. User profiles contain a SECLABEL name, which is the name of a profile of the SECLABEL class. This profile contains the security classification consisting of a hierarchical security level and a set of non-hierarchical categories. The values for the levels and the categories can be defined by the administrator. The administrator can then also define resources in the SECLABEL resource class as a combination of one security level and zero or more categories. Such a resource is called a “security label”.

The system defines a set of predefined security labels:

- SYSHIGH
  This label consists of the highest security level and all categories defined for the system.
- SYSLOW
  This label consists of the lowest security level defined for the system and no categories.
- SYSNONE
  This is used for resources that need to be excluded from MAC checking. It is used in the evaluated configuration for TCPIP. It must be defined as SYSNONE so that any user can login using telnet. If not defined as SYSNONE, then only users that have the same security label as user TCPIP can log on. It is to be applied only to trusted userids that perform system-wide functions on behalf of all users.

The access control enforced by the TOE ensures that users may only read labeled information if their security label dominates the information’s label, and that they may only write to labeled information containers if the container’s label dominates the subject’s.

For evaluating RACF access control rules, MAC rules are evaluated prior to DAC rules. When MAC rules deny access, no further evaluation of DAC rules is done. If MAC rules allow access, DAC rules are consulted afterwards to finally decide the access allowance. MAC rules are checked at access time of the object (i.e. in case of a change in MAC rules, changes affect only new access attempts).

During logon, users can select one privilege class out of all classes they are assigned to. At runtime of the virtual machine, restrictions from this privilege class only apply (other privilege classes the user is assigned to, but not selected during logon are not enforced). Users can alter their selected privilege class using the “logon USERID seclabel SECLABEL” command. The privilege class can only be changed to other classes during logon, the users is configured access to (i.e. these privilege classes are already selectable by the user during logon).

6.3.3.5 Access authority

The access authority determines to what extent the specified user or group can use the resource. The owner of a profile protecting a general resource (such as a tape volume or terminal) can grant or deny a user or group access to that resource by including the user ID or group ID in the resource profile's access list. Associated with each user ID or group ID is an access authority that determines whether the user or group can access the resource, and if they can access the resource, how they can use it. Access types that may be granted are NONE, READ, UPDATE, CONTROL, and ALTER, which form a hierarchical set of increasing access authorities.

- **NONE**
  The specified user or group is not permitted to access the resource or list the profile.
- **READ**
  Allows users to access the resource for reading only. (Note that users who can read the minidisk can copy or print it.) For minidisks, link modes R, RR, SR, and ER are permitted.
- **UPDATE**
  Allows users to read from, copy from, or write to the resource. For minidisks, link modes W, WR, SW, or EW are permitted in addition to those allowed for READ.
- **CONTROL**
  Allows users to read from, copy from, or write to the resource. For minidisks, link modes M, MR, and SM are permitted, in addition to those allowed for UPDATE.
- **ALTER**
  Allows user to read from, copy from, or write to the resource. For minidisks, link mode MW is permitted in addition to those allows for CONTROL.

  When specified in a discrete profile, ALTER allows users to read, alter, and delete the profile itself, including the access list. However, ALTER does not allow users to change the owner of the profile.

  When specified in a generic profile, ALTER gives users no authority over the profile itself.

In some cases the resource may not implement read-only or read-write capabilities and in such cases, the level of access required to permit use is resource-specific and is documented in the RACF Security Administrator's Guide [RACFSAG].

It is to be noted that MAC rules take precedence over DAC rules in case they contradict each other. DAC rules are checked at access time of the object (i.e. in case of a change in DAC rules, changes affect only new access attempts).

6.3.3.6 Deferring access control decisions

In case RACF is unable to validate the requested access, RACF notifies CP that it cannot perform the access control decisions. Inability of validating access is possible for RACF in case there is no profile for the calling subject or the requested object. CP validates access based on the directory entries for the calling subject and the requested object

In the evaluated configuration, the RACF – CP interface is configured in a way that any deferred operations are automatically and unconditionally denied by CP.

Please note that in case RACF severed the connection to CP due to the audit trail is full, no notification about RACF deferring the access control decision to CP can be made. Therefore, no CP based access control is conducted. This state causes CP to fail any request that requires RACF intervention.

6.3.4 F.AC.4 - Access Control Configuration and Management

Management of the access control facility is restricted to users with specific authorities defined in their user profile. The following list shows those authorities:

- SPECIAL Attribute
- AUDITOR Attribute
- CLAUTH Attribute

6.3.4.1 System wide configuration of RACF

The system administrator can define system wide-options of RACF with the SETROPTS, SETEVENT and SETRACF commands.

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(*), GLOBAL(*), GRPLIST.

6.3.5 F.AC.5 - Protected Resources

On z/VM, RACF can be used to control access to all objects listed in FDP_ACC.1 (RACF) with discretionary and in FDP_IFC.1 with mandatory access control checks.

For the evaluation the protection of the following resource classes is considered:

FIELD
Fields in RACF profiles (field-level access checking).

GLOBAL
Global access checking table entry. fastpath DAC rules for other classes. only for syslow MAC level.

GTERMINL
Resource group class for TERMINAL class. See below for terminal class

SECDATA
Security classification of users and data (security levels and security categories).

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

SURROGAT
If surrogate submission is allowed, and if allowed, which user IDs can act as surrogates.

TERMINAL
Terminals.

WRITER
Controlling the use of CP controlled printers.

6.3.6 F.AC.6 - Access control enforcement by CP

In addition to the access control checks performed by RACF as outlined above, CP also provides discretionary access control checks. Access to all CP commands and all DIAGNOSE codes is governed by CP.

6.3.6.1 Privilege classes

Each CP command and each DIAGNOSE code is assigned to a privilege class. The TOE provides predefined privilege classes (A to G) and already assigned all CP commands and DIAGNOSE codes to one of them (privilege class H is reserved by IBM for future use, thus having 8 predefined classes on the system: A through H). DIAGNOSE codes and CP commands assigned to the privilege class any are not subject to CP discretionary access control.

Privilege classes can be redefined by the authorized administrator. Also completely new definitions of privilege classes can be configured. The user class restructure feature provides customers with the ability to control access to commands and DIAGNOSE codes more precisely through customer-defined classes. Customers can use this feature to generate up to 24 self-defined privilege classes in addition to the eight pre-defined classes.

Each user is assigned to one or more privilege classes that he can choose from during logon. During runtime of the virtual machine, the user is associated with only one privilege class that cannot be altered, except when logging of and logging in again. Privilege classes can be changed by calling the command “set PRIVCLASS”.

6.3.6.2 Access check

CP performs the access check by matching the privilege class from the user requesting access to one object (CP command or DIAGNOSE code) with the privilege class assigned to this particular object. If both privilege classes are identical, access is granted, otherwise denied.

6.3.6.3 Consistency of access checks between RACF and CP

The access check on those objects is performed sequentially: first the CP check is being performed and RACF authorizes second. In case the CP check denies access, no further RACF check is performed. In contrast, if the CP check accepts the request from the user, RACF performs its access check. Only if both access checks succeed, the request is being allowed to proceed.

6.4 F.I&A: Identification and Authentication

6.4.1 F.I&A.1 - Identification and authentication mechanism

Users can interact with the TOE in one of the following ways:

- As an operator at a console or via Telnet using Control Program commands
- Using software from inside virtual machines executing DIAGNOSE instructions or processor instructions that cause the SIE instruction to terminate and return the processor control to the CP

In all cases users are identified and authenticated by a user ID / password combination.

When authenticating a user, RACF checks:

- If the user is defined to RACF.
- If the user has supplied a valid password, and a valid group name, otherwise a default group name is selected. Also a security label is associated with the user (in LSPP mode).
- If the user ID is associated with the REVOKE attribute, which prevents a RACF-defined user from entering the system at all.
- If a user’s group is associated with the REVOKE attribute, the user cannot enter the system by connecting to that particular group, or access resources as a member of that group.

After it has authenticated the user’s identity, RACF associates the user with its user attributes.

To identify a user means to firmly establish who is using the system to perform a particular act. Every command, DIAGNOSE, and other security-relevant event is directly attributable to a user whose identity has been well-established.

If the connection to RACF is SEVERed (e.g. due to the audit trail being full), CP reverts back to the use of the local System Directory for authenticating users. In the evaluated configuration, only administrative user ID are to be maintained with the System Directory to allow an administrator to log on to the system if RACF is unavailable.

6.4.2 F.I&A.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. This new password will be in an expired state, thus forcing the user to enter a new password on the first logon.

A system administrator can set a variety of rules for forming valid passwords, and this is done via the SETROPS command (for system-wide settings) or with the password command (to affect only one user). Configurable are options such 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; and so on.

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.

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

- Minimum and maximum length of passwords (LENGTH(m1:m2) as part of a RULE suboption)
- Maximum password lifetime (INTERVAL suboption)
- Number of passwords from the user's password history that are not allowed for a new password (HISTORY suboption)
- Maximum number of consecutive failed authentication attempts until the REVOKE attribute is set in the user's profile (REVOKE suboption)
- Type of character for each character position of a password. Possible types are:
  - ALPHA
  - ALPHANUM
  - VOWEL
  - NOVOWEL
  - CONSONANT
  - NUMERIC

6.4.3 F.I&A.3 - Identity Change

During runtime of a virtual machine, the user can switch his identity using the DIAGNOSE 'D4' instruction. The changed user ID applies to all subsequent access control checks (DAC and MAC). Using RACF, the administrator is able to limit the target user IDs a particular user can impersonate. In addition, access to this DIAGNOSE function might be disabled completely for dedicated users.

Using the LOGON BY command, a user can logon using his credentials and assume the identity of another specified user. The administrator must give explicit authority to the user for executing this command.

In LSPP mode: Change of security labels at runtime of a virtual machine is not allowed. For changing the security label, a user has to log off and log on. During the log on process, the user can choose one out of all security labels assigned to this user.

6.5 F.IP: Interference Protection between virtual machines

The TOE provides a strict separation functionality for ensuring confidentiality and integrity between virtual machines to the extend of specifically configured communication channels.

For maintenance of integrity and separation of virtual machines, z/VM exploits the z/Architecture architecture in several other ways:

- The addresses in a virtual machine are virtual addresses. They have no meaning outside the virtual machine in which they are generated and used. Whenever required, these virtual addresses are translated into real addresses by ART (access register translation) and DAT (dynamic address translation), for the address space referenced by the user. Using ART and DAT, the system keeps these address spaces absolutely separate from one another. This means that it is impossible for one user to access an address space of another user unless the owner allows the other user to do so.
- z/VM translates the addresses in all channel programs, except those initiated by DIAGNOSE X'98'. Channel programs are programs built and run by virtual machines that request auxiliary storage devices to perform input and output tasks. z/VM identifies the storage device and performs the I/O operation on behalf of the virtual machine.
- Every z/VM virtual machine runs in interpretive-execution mode which processes most privileged and non-privileged instructions and handles virtual storage address translation without requiring intervention of z/VM (see section 6.8.1.1 for details).
- z/VM uses page protection to prevent read-only saved segments from being modified. A saved segment is a block of data or re-entrant code in virtual, shared storage that many users can share simultaneously. However, if a user has a legitimate reason for wanting to change a read-only saved segment, the user must specifically request an exclusive copy of the saved segment and be authorized to do so in the system user directory. The unmodified code remains shared among the other virtual machines.

Devices with DMA access are accessed by virtual machines by mapping the DMA memory area into the virtual machine's memory. The mapping is enforced by CP upon initialization of the virtual machine during login of a user.

The CP enforces a strict separation of the virtual machines. For doing this, CP ensures:

- Virtual machines can only access assigned memory ranges. CP verifies upon initialization of a virtual machine and during allocation of memory during operation of virtual machines that no memory overlaps are present between virtual machines except specifically configured. A similar check is performed when virtual machine memory is resized during runtime of the virtual machine.
- CP provides only configured processor resources to virtual machines by virtualizing and simulating the number of logical processors configured for each virtual machine. CP also ensures that logical processors are scheduled according their configured processing power on real processors. No virtual machine instruction can block scheduling of logical processors.

Supported by the underlying processor, the TOE restricts results of software failures (such as program checks or virtual machine checks) occurring in a virtual machine to this machine, thus not affecting other virtual machines or the CP.

Memory as well as DASD devices and their derived devices (such as minidisks) can be configured to be shared among virtual machines. The administrator can configure sharing of devices for a subset of virtual machines. Also, the administrator can configure access of virtual machines consoles from other virtual machines using the single console image facility (SCIF) or by allowing the CP SET SECUSER command. The TOE ensures that sharing objects between virtual machines are limited to these objects, hence they are allowed in an evaluated configuration. The administrator has to ensure that shared configurations are in line with the organizational rules.

It is to be noted that specific communication channels can be established between virtual machines that are capable of transporting interference from one virtual machine to another. Interference transmitted through these communication channels are not covered by this security function. However, the TOE ensures that all communication channels can only be used within the boundary of their definition. The following table presents all possible communication channels and defines the boundary the channel is subject to. This table includes communication channels between a virtual machine and the Control Program.

Communication channel	**Boundary**
Guest LAN	DAC / MAC enforcement Multidirectional channel between all configured virtual machines
VMCF	MAC enforcement bidirectional channel between two configured virtual machines
IUCV	DAC / MAC enforcement bidirectional channel between two configured virtual machines
CP commands MESSAGE (MSG), SMSG, and WARNING (WNG), MSGNOH	MAC enforcement unidirectional channel between two configured virtual machines
VCTC	DAC / MAC enforcement bidirectional channel between two configured virtual machines
Spool files	DAC / MAC enforcement transferring spool files between virtual machines
AUTOLOG	MAC enforcement Providing of initial console data to virtual machine
XAUTOLOG	DAC / MAC enforcement Providing of initial console data to virtual machine
SET SECUSER	Command is disabled when MAC checking is activated by activating the SECLABEL class. In that case, only the system administrator can set the secondary user by using the CONSOLE statement in the user directory. Enable read and write access to a virtual machine console
SET OBSERVER	Command is disabled when MAC checking is activated by activating the SECLABEL class. In that case, only the system administrator can set the secondary user by using the CONSOLE statement in the user directory. Enable read access to a virtual console
Minidisks	DAC / MAC enforcement configured minidisks are shared
Memory	DAC / MAC enforcement configured memory range is shared

Table 6-4 Communication channel usage

To separate different concurrent Telnet connections, the TCP/IP application (which includes the Telnet server) maintains TCP/IP sessions by exploiting the TCP protocol immanent sequence and acknowledge numbers. The Telnet server uses these maintained sessions to connect each individual Telnet connection with the virtual console where the session-initial identification and authentication of the user was performed. The Diagnose code X'08' is being used by the Telnet application to access the virtual console facility of CP.

6.6 F.OR: Object re-use

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

Subject to object reuse enforced by the TOE are:

- Memory ranges cleared upon reallocation to other virtual machines.
- All registers are reassigned since all virtual machines have the same architected registers. The registers are not cleared, however they cannot, by definition, retain any residual data since all registers are reloaded.
- Temporary disk space is cleared automatically when the FEATURES ENABLE CLEAR_TDISK option is enabled in the system configuration.

Clearing of minidisks, and other DASD volumes must be carried out by the administrator in accordance with organizational policies. Additional software facilities may be used to support this task, but they are not part of this evaluation.

Therefore, subject to object reuse implemented by organizational rules is:

- Clearing of storage space provided minidisks, temporary disk space, and other forms of DASD devices, and auxiliary storage devices, such as tape devices.

6.7 F.SM: Security Management

The TOE allows the management of security functions by trusted users to alter the behavior of security functions and other functions to organizational needs. The following security functions can be managed:

- Management of object security attributes, including discretionary access control and (in LSPP mode) of security labels for mandatory access control
- Management of the audit trail and the events to be audited
- Management of user security attributes, including authentication data and access control

For carrying out security management, the TOE maintains different roles for users. Such user roles depend on the following authorizations:

- Authorization to access and modify objects based on DAC and MAC
- Authorization to access and modify objects based on attributes (such as SPECIAL or RACF AUDITOR)

6.7.1 F.SM.1 - Management of user security attributes

RACF provides the user database holding various security attributes for user. On VM, authorized users can enter RACF commands by preceding the command name with RAC, or by entering a RACF command session on a RACF console. Accessing RACF using the console is similar to a command line. Besides this command line, authorized users can enter RACF commands by using the RACF ISPF panels. These panels provide an interactive, menu driven user interface.

By using the aforementioned interfaces, authorized users can manage users and groups. User management includes:

- Assignment of IDs to usernames
- Assignment of hardware components to users
- Assignment of user profiles to users

- Assignment of attributes (SPECIAL, AUDITOR, OPERATIONS, CLAUTH, REVOKE) to users
- Assignment of a default universal access authority (UACC) of NONE, READ, UPDATE, CONTROL, or ALTER when being connected to a group. RACF uses this default UACC for all new resources a user defines while connected to the specified default group. When a user issues the ADDDIR, ADDFILE, or RDEFINE command to define a new general resource profile and does not specify a value for the UACC operand, RACF uses the default UACC as the UACC for the profile unless a value for UACC is specified in the class descriptor table.
- Assignment of security levels or security labels (a combination of security levels and security categories) (in LSPP mode)

Other user attributes can be set as well.

Group management includes:

- Defining of groups (or group profiles)
- Assignment of the group's superior group (the predefined group SYS1 is the only group having no superior)
- Assignment of the owner of the group

6.7.2 F.SM.2 - Management of object security attributes

Similar to the management of user security attributes, object security attributes can be managed by authorized users through the two available user interfaces (command line and RACF ISPF panels).

Each object can be assigned to a resource profile with RACF.

The following information can be managed for objects:

- Assignment to a general resource classes (such as TERMINAL)
- Assignment to a generic (this profile may cover more than one object) or a discrete (this profile covers only one object) profile name
- Assignment of an universal access authority (UACC – NONE, READ, UPDATE, CONTROL, ALTER) for users who are not otherwise restricted
- Assignment of a user or group as owner of the resource profile
- Assignment of security levels and categories (or assignment of security labels, which cause security levels and categories to be ignored during access check). (LSPP mode)

6.7.3 F.SM.3 - Management of audit

The management of the audit facility can only be performed by users having the AUDITOR attribute, or who belong to a group with the group-AUDITOR attribute. As an exemption, owners of resource profiles can configure RACF to log access attempts to resources protected by the profile (AUDIT operand).

RACF can be configured to audit the following events:

- Changes to any RACF profiles
- All RACF commands that a SPECIAL or group-SPECIAL user issues
- All unauthorized attempts to use RACF commands
- Selected VM events, using the SETEVENT command
- All RACF-related activities of specific users
- All accesses to resources (minidisks and general resources) that RACF allows because the user has the OPERATIONS or group-OPERATIONS attribute

- All accesses to specific minidisks
- All accesses to specific general resources
- All accesses to resources protected by specific profiles in the SECLABEL class (LSPP mode)
- All accesses to a specified class of resources at an access level indicated on the LOGOPTIONS keyword of the SETROPTS command

Similar to the configuration of object and user attributes, the audit facility can be configured either using RACF commands or ISPF panels.

The TOE maintains a reliable clock synchronized with the clock from the underlying abstract machine used to generate time stamps as required for the TOE itself and applications. The audit subsystem requires such a reliable time source for the date and time field in the header of each audit record. The clock uses timers provided by the hardware and interrupt routines that update the value of the clock maintained by the TOE.

The initial value for this clock may be provided by a hardware clock that is part of the underlying abstract machine, or by the system administrator setting the initial value. Only the system administrator is allowed to overwrite the value of the clock maintained by the TOE at IPL time (e. g. to correct the value in case it has drifted over time due to some inaccuracy of the hardware timer used by the TOE).

6.7.4 F.SM.4 - Management of system assurance testing

To perform the system assurance testing, the abstract machine has to be brought into its maintenance mode and the test application has to be started.

The test application is the System Assurance Kernel that tests whether the abstract machine conforms to the z/Architecture Principles of Operation specification.

6.8 F.TP: TOE Self Protection

6.8.1 F.TP.1 - Supporting Mechanisms of the Abstract Machine

The following section provides a short overview of the supporting protection mechanisms of the abstract machine z/VM is executing on. The purpose of this section is to better understand how z/VM uses those mechanisms to protect itself against tampering and bypassing of the security functions of z/VM.

The z/VM control program system integrity is defined as the inability of any program running in a virtual machine not authorized by a z/VM control program mechanism under the customer's control or a guest operating system mechanism under the customer's control to:

- Circumvent or disable the control program real or auxiliary storage protection.
- Access a resource protected by RACF.
- Access a control program password-protected resource.
- Obtain control outside the SIE environment or with privilege class authority or directory capabilities greater than those it was assigned.
- Circumvent the system integrity of any guest operating system that itself has system integrity as the result of an operation by any z/VM control program facility.

Real storage protection refers to the isolation of one virtual machine from another. CP accomplishes this by hardware dynamic address translation, start interpretive-execution (SIE) guest storage extent limitation, access register translation (ART), and dynamic address translation (DAT).

Auxiliary storage protection refers to the disk extent isolation implemented for minidisks/virtual disks through channel program translation.

Password-protected resource refers to a resource protected by CP logon passwords.

Guest operating system refers to a control program that operates under the z/VM control program.

Directory capabilities refer to those directory options that control functions intended to be restricted by specific assignment, such as those that permit system integrity controls to be bypassed or those not intended to be generally granted to users.

6.8.1.1 Processor Features

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

- Start Interpretive-Execution (SIE) instruction of the processor
- Access register translation (ART) and dynamic address translation (DAT) facilities provided by the processor

The SIE instruction provided by the processor is the central facility the TOE manages. It is called by the Control Program (CP) restricting the scope of the processor to a limited memory range to set up a virtual machine environment. If the processor enforcing a SIE environment is instructed to execute predefined privileged instructions, the SIE environment is terminated and control is returned to CP. This SIE instruction is executed with a CP-controlled timer to allow scheduling of logical processors (processors visible from inside a virtual machine) and CP execution time on real processors.

Access register translation (ART) and dynamic address translation (DAT) protect the storage of non-shared segments; i.e. storage that is reserved exclusively for one user. ART and DAT are hardware facilities used by z/VM during the execution of any instruction to translate a virtual address into the corresponding real address. The system uses ART and DAT to provide secure, separate address spaces for each virtual machine in the system. This means that it is impossible for one user to access an address space of another user unless its owner allows the other user to do so.

The zSeries processor execution of code is driven by the “Program Status Word” (PSW). The PSW holds information such as content of processor control registers, and storage keys for pages within real memory. 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. 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. Storage protection is in effect whether the processor is in user 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 / fetch into / 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 (the so-called interrupt vector).

When a virtual machine issues an instruction that breaks the SIE environment, the processor stores the current PSW of the virtual machine (which contains the instruction pointer pointing to the instruction following the current instruction) into a fixed location in the processor individual real storage in the first 8 KB 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.

In addition to the main processor there is a dedicated I/O hardware subsystem, the “Channel” subsystem that allows having I/O operations being performed in parallel to the normal processor operation. Configuring and programming the I/O subsystem is restricted to programs operating in supervisor state.

6.8.1.2 TOE procedures

The TOE's address space management ensures the strict separation of memory assigned to virtual machines and enforced by the SIE environment.

The TOE's scheduling management ensures the operation of multiple logical processors and CP execution time on top of multiple physical processors.

Access to system services (e.g. via a DIAGNOSE instruction) is controlled by the system, which requires subjects who wish to perform security relevant tasks to be appropriately authorized.

6.8.1.3 Abstract Machine Modes of Operation

z/VM executes in the logical partition mode.

In logical partition mode z/VM has full control to all 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/VM 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.

6.8.2 F.TP.2 - Structure of the TOE

The trusted parts of z/VM consist of

- the Control Program kernel, or "nucleus"
- authorized applications

The z/VM nucleus contains the functions invoked either by a CP command, a DIAGNOSE instruction or by terminating the SIE instruction and returning control over the processor back to CP. Those functions start to operate in supervisor state outside the SIE environment with a storage key mask of zero in the PSW. They may change their storage key mask in the PSW (e. g. when checking user operands) but as long as they execute in supervisor state they may set their storage key mask back to zero at any time.

In addition to the control program z/VM has a number of "authorized applications" that need to be trusted since they are granted access to specifically restricted interfaces provided by CP. Using these interfaces, applications may override or modify security policies defined in this Security Target and may implement security functionality. A trusted application establishes a bidirectional communication channel with CP.

There are two authorized applications belonging to the TOE, which run in dedicated virtual machines: RACF and the TCP/IP stack.

For trusting the virtual machines running RACF instances (it is possible to run multiple instances of RACF for one z/VM instance), the Control Program is to be changed. During installation time of the RACF application, the CP is to be recompiled with the following changes:

- A list of all user IDs running an instance of RACF that CP has to trust
- Integration of the RACF interface *RPI into the specifically provided hooks in CP. These hooks are queried for access control by CP; the queries are forwarded through the *RPI interface to RACROUTE, the interface to RACF. *RPI uses IUCVs as transport vehicle.

The TCP/IP does not need specific changes to CP for being executed. However, to run the TCP/IP stack (and optionally the Telnet service), the virtual machine running the TCP/IP stack application must have access to at least one network device. In case Telnet is enabled for accessing virtual consoles, the DIAGNOSE code 0x98 must be enabled for the virtual machine.

6.8.2.1 Protection of Trusted Applications

Trusted applications need to be trusted by CP since they implement part of the security functionality provided by the TOE. Trusted applications therefore must be carefully protected from unauthorized modification and the system must be protected from adding authorized applications other than those allowed in the evaluated configuration. The protection of the trusted application is done by the strict separation of the virtual machines implemented by CP. Each trusted application is running inside a virtual machine on top of the operating system CMS.

Trusted and non-trusted applications are characterized in section 2.4.1.

6.9 Assurance Measures

The following table provides an overview, how the assurance measures of EAL4, and ALC_FLR.2 are met by z/VM.

Assurance Component	Documentation describing how the requirements are met
ACM_AUT.1	Automatic means to version control and generate the TOE are in place.
ACM_CAP.4	z/VM is developed at different sites each using a well defined and highly automated configuration management system. Each site has a detailed description how the configuration management for the z/VM 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/VM is delivered via sales channels controlled by IBM.
ADO_IGS.1	Guidance for installation and system configuration is provided in a number of documents part of the zSeries z/VM Collection.
ADV_FSP.2	The functional specification for z/VM 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 minidisks. 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 parameter of functions available to authorized programs only.
ADV_HLD.2	A high level design of the security functions of z/VM is provided. This document provides an overview of the implementation of the security functions within the subsystems of z/VM and points to other existing documents for further details where appropriate.
ADV_LLD.1	A low level design of the security functions of z/VM is provided. This document provides a detailed description of the implementation of the security functions within the modules of z/VM.
ADV_IMP.1	Portions of the implementation representation are provided for verification of the implementation of security functionality.
ADV_SPM.1	A security policy model describes and analyzes the security policies implemented in z/VM and RACF.
ADV_RCR.1	The correspondence information is provided in the form of a spreadsheet showing the correspondence between the TOE summary specification and the different design aspects.
AGD_ADM.1	A number of documents exist that provide guidance for the system

Assurance Component	Documentation describing how the requirements are met
	administrator. This includes guides for the overall system configuration and management as well as the configuration and management for individual components of z/VM. Especially for the configuration and management of RACF a System Administrator Guide exists, that describes and explains in detail the administration commands and parameter.
AGD_USR.1	User Guidance is provided in a number of documents related to the individual components of z/VM. 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 Endicott) 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.2	The z/VM 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	The life-cycle of the TOE demonstrates the maintenance and development cycle. This description is supported by the fact that z/VM originates in the early 1970s.
ALC_TAT.1	Well defined programming languages are used for the implementation of the TOE.
ATE_COV.2	IBM has detailed test plans to test the functions of z/VM. Those test plan 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 at 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/VM. This team has a long-term experience with potential security problems within z/VM and is also integrated in the design reviews. The developer vulnerability analysis will report the activities and findings of this team.

Table 6-5 Assurance Measures

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 SUCCHANNEL processor instruction in the case of an error within an I/O device. IBM field service has specific utilities that allow locating 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.

7. Protection Profile Claims

7.1 PP 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 PP Tailoring

All operations allowed by security functional requirements from [LSPP] or [CAPP] have been performed and marked with green font in chapter 5.

The following security functional requirements represent a ST specific extension to the requirements defined by [LSPP] and [CAPP]:

- FMT_SMF.1 (added due to updates to the CC not present at the time of writing of LSPP/CAPP)
- FPT_FLS.1 (added to reflect the security functionality of virtual machine separation)
- FPT_SEP_(EXT).1 and FPT_SEP_(EXT).2 as an addition to the security functional requirement FPT_SEP.1 (derived from the manual on “Basic Robustness Environments” to reflect the maintenance of an execution domain for the TOE)
- FRU_FLT.1 (added to reflect the security functionality of virtual machine separation)

Security Functional Requirements have been refined where required by the Protection Profiles.

The following security objectives for the TOE have been added to reflect the objective a strict separation mechanism for virtual machines and the maintenance of an execution domain for the TOE’s own use:

- O.NONINTERFERE
- O.NO_COMM
- O.PARTIAL_SELF_PROTECTION

The following security objective for the TOE environment has 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.

In addition the Security Target has added one security requirement 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 un-trusted software (FPT_SEP_ENV.1).

The assurance requirements of the Protection Profile are those defined in the Evaluation Assurance Level EAL3 of the Common Criteria. This Security Target specifies an Evaluation Assurance Level EAL4 augmented by ALC_FLR.2. Since the Evaluation Assurance Levels in the Common Criteria define a hierarchy, all assurance requirements of the Protection Profile are included in this Security Target plus all additional SARs which are added by the higher EAL. ALC_FLR.2 which has been added to the assurance requirements defined in the 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.

8. Rationale

This chapter provides the rationale for the selection, creation, and use of 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 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 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 8.2 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 8.2. Section 8.3 provides the rationale for the TOE summary specification and the consistency analysis between the TOE security functions and security functional requirements. The assessment of the PP consistency is given 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 fort the existence of each 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.

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.NONINTERFERE O.NO_COMM O.PARTIAL_SELF_PROTECTION OE.HW_SEP
P.ACCOUNTABILITY	O.AUDITING O.MANAGE O.ENFORCEMENT OE.HW_SEP
P.CLASSIFICATION (LSPP mode	O.MANDATORY_ACCESS

Organizational Security Policy	Objective
only)	O.RESIDUAL_INFORMATION O.MANAGE O.ENFORCEMENT OE.HW_SEP OE.CLASSIFICATION

Table 8-1 Mapping from OSP to objectives

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.

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 which 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 re-used. O.MANAGE ensures that permissions can be managed properly, and O.ENFORCEMENT ensures that the access control functions are invoked and operate correctly. The interference protection of virtual machines has to facets: the O.NONINTERFERE objective implements the functionality for ensuring no software actions in one virtual machine affects the operation of any other virtual machine. In addition O.NO_COMM ensures that no communication channels exist between virtual machines that can be used to obtain any information from other virtual machines (specifically configured communication channels, such as shared memory or disks are allowed – the TOE ensures that the restrictions of the specifically configured communication channel are enforced). Finally, O.PARTIAL_SELF_PROTECTION ensures that the TOE maintains an execution domain for its own use protecting itself and its resources against interference and tampering by untrusted subjects.

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 and O.ENFORCEMENT ensures that functions provided for O.AUDITING are invoked and operate correctly.

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, and O.ENFORCEMENT ensures that the mandatory access control functions are invoked and operate correctly. 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 OSP:

Objective	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.NONINTERFERE	P.NEED_TO_KNOW
O.NO_COMM	P.NEED_TO_KNOW
O.PARTIAL_SELF_PROTECTION	P.NEED_TO_KNOW

Table 8-2 Mapping from objectives to OSP

8.1.2 Complete Coverage – Environmental Assumptions

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

Non-IT Security Objectives	Environmental Assumptions and Policies
OE.INSTALL	A.MANAGE A.NO_EVIL_ADM A.PEER
OE.PHYSICAL	A.LOCATE A.PROTECT A.CONNECT
OE.CREDEN	A.COOP

Non-IT Security Objectives	Environmental Assumptions and Policies
OE.HW_SEP	P.AUTHORIZED_USERS P.NEED_TO_KNOW P.CLASSIFICATION P.ACCOUNTABILITY
OE.CLASSIFICATION (LSPP mode only)	A.CLEARANCE A.SENSITIVITY P.CLASSIFICATION

Table 8-3 Mapping Non-IT Security Objectives to Environmental Assumptions and Policies

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 which maintains IT security objectives.*

The assumption on competent administrators A.MANAGE is covered by the objective, requiring the TOE to be delivered, installed, managed, and operated in a manner maintaining security. A.NO_EVIL_ADM requires the administrator to be trustworthy, which is also covered by the objective, since the administrator has to maintain security of the TOE. The requirements of secure delivery, installation and management cover the assumption on the same management control and security policy constraints for systems with which the TOE communicates (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 assumption on physical protection of the processing resources of the TOE (A.LOCATE) is covered by the objective requiring physical protection. The assumption on physical protection of hard and software critical to the TOE's security (A.PROTECT) is covered by the objective. The assumption on controlled access to peripheral devices and protected internal communication paths is covered by the objective for ensuring physical protection. Finally, the assumption on securing all connections to peripheral devices is covered by A.CONNECT.

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 which maintains IT security objectives.*

The assumption on authorized users to act in a cooperating manner (A.COOP) is covered by the objective requiring the safe storage and non-disclosure of authentication credentials.

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.*

In order for the TOE to provide all functions implementing all policies (P.AUTHORIZED_USERS, P.NEED_TO_KNOW, P.CLASSIFICATION, and P.ACCOUNTABILITY), the underlying hardware must support the self-protection of the TOE as required by this objective.

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). The assumptions about assigning users a clearance level and objects a sensitivity level are covered by A.CLEARANCE and A.SENSITIVITY.

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:

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

Table 8-4 Non-IT Security Objectives to Environmental Assumptions

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. Functional component FMT_SMF.1 (Specification of Management Functions) has been added in accordance with CC version 2.3. The use of component refinement was accomplished in accordance with CC guidelines.

An additional component was included by the [LSPP] Protection Profile to clarify the relationship of objects and security attributes. Additionally [LSPP] extended one component.

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 exists in a TOE.

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

Audit

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 (since 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 that always all possible audit records are 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.

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

Access Control

FDP_ACC.1 requires the existence of a Discretionary Access Control Policy for named objects in z/VM. The rules of this policy are described in FDP_ACF.1. 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 and FMT_REV.1. To be effective, a discretionary access control mechanism requires users to be properly identified and authenticated (as required by FIA_UID.1 and FIA_UAU.1), and the subjects acting on their behalf being bound to them (as required by FIA_USB.1). Reference mediation (as required by FPT_RVM.1) and domain separation (as required by FPT_SEP.1) assure that the DAC mechanisms are always invoked and cannot be tampered with. The policy is also supported by the requirement for residual information protection (FDP_RIP.2), which prevents that users can access information they are not authorized to via residual information remaining in objects that they allocate. To ensure unauthorized users have no access to resources, the DAC mechanisms have a restrictive default setting as required by FMT_MSA.3.

FDP_IFC.1 requires the existence of a Mandatory Access Control Policy for named objects in z/VM. The rules of this policy are described in FDP_IFF.1. 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 and FMT_REV.1. To be effective, a mandatory access control mechanism requires users to be properly identified and authenticated (as required by FIA_UID.1 and FIA_UAU.1), and the subjects acting on their behalf being bound to them (as required by FIA_USB.1). Reference mediation (as required by FPT_RVM.1) and domain separation (as required by FPT_SEP.1) assure that the MAC mechanism is always invoked and cannot be tampered with. The policy is also supported by the requirement for residual information protection (FDP_RIP.2), which prevents that users can access information they are not authorized to via residual information remaining in objects that they allocate. To ensure unauthorized users have no access to resources, the DAC mechanisms have a restrictive default setting as required by FMT_MSA.3. The export of unlabelled data is required to be possible within the constraints of the MAC mechanisms (FDP_ETC.1). Also, export of labeled data bearing the user's security attributes is required by FDP_ETC.2. For importing data, the TOE is required to ignore any user security data during the import of unlabelled data (FDP_ITC.1). When importing labeled data, the TOE is required to import and use (thus enforce) the user's security data associated with the labeled data (FDP_ITC.2).

Identification and Authentication

Identification and Authentication is required for discretionary and mandatory access control, which are based on the identity of individual users. FIA_UAU.1 and FIA_UID.1 require that users be authenticated before they can perform any action on the TOE. 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/VM virtual machines) is properly bound to the user for whom it runs. This association provides also the user attributes (defined by FIA_ATD.1) necessary to take policy decisions.

Object Reuse

Object reuse (as required by FDP_RIP.2 and Note 1) is a supporting function that prevents unauthorized access to information via residuals left in objects when they are re-allocated to another subject or object. As this the function supports the intention of the discretionary and mandatory access control policies.

Security Management

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

Management of access rights and labels attached to objects is necessary to configure the DAC and MAC mechanisms; it is defined by FMT_MSA.1 and FMT_REV.1 "Revocation of Object Attributes". In addition new objects require having default access rights and security labels which are required by FMT_MSA.3.

Management of users and groups is defined in FMT_MTD.1 "Management of User Attributes" and FMT_REV.1 "Revocation of User Attributes". Since passwords are used for authentication, the management of authentication data is also required in FMT_MTD.1 "Management of Authentication Data".

Management of the audit system is covered by the requirements for the management of the audit trail (FMT_MTD.1 "Management of the Audit Trail") and the management of the audit events (FMT_MTD.1 "Management of the Audit Events"). Audit trail management is supported by the requirements for the audit review (FAU_SAR.1, FAU_SAR.2 and FAU_SAR.3) as well as the requirements for the protection of the audit trail (FAU_STG.1, 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 also comprises the management of a reliable time stamps. Such time stamps are essential for correct time information within audit records. Times stamps are addressed by FPT_STM.1.

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.

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).

Interference Protection

The TOE ensures that any action in one virtual machine (specifically software errors or hardware errors of hardware dedicated to this virtual machine) is contained in this virtual machine (FPT_FLS.1.1). This also implies a fault tolerance against errors occurring in one virtual machine (FRU_FLT.1). Reference mediation (FPT_RVM.1) and domain separation (FPT_SEP.1) ensure that no communication between virtual machines is allowed except specifically configured.

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.

Security Objective	Security Functional Requirement
O.AUTHORIZATION	5.1.3.1 User Attribute Definition (FIA_ATD.1) 5.1.3.2 Strength of Authentication Data (FIA_SOS.1) 5.1.3.3 Authentication (FIA_UAU.1) 5.1.3.4 Protected Authentication Feedback (FIA_UAU.7) 5.1.3.5 Identification (FIA_UID.1) 5.1.4.6 Management of *Authentication Data* (FMT_MTD.1)
O.DISCRETIONARY_ACCESS	5.1.2.1 Discretionary Access Control Policy by RACF (FDP_ACC.1) 5.1.2.2 Discretionary Access Control Policy by CP (FDP_ACC.1) 5.1.2.3 Discretionary Access Control Functions by RACF (FDP_ACF.1) 5.1.2.4 Discretionary Access Control Functions by CP (FDP_ACF.1) 5.1.3.1 User Attribute Definition (FIA_ATD.1) 5.1.3.6 User-Subject Binding (FIA_USB.1) 5.1.4.1 Management of *Object Security* Attributes (FMT_MSA.1) 5.1.4.2 Static Attribute Initialization (FMT_MSA.3) 5.1.4.8 Revocation *of Object Attributes* (FMT_REV.1)
O.MANDATORY_ACCESS	5.1.2.5 Export of Unlabeled User Data (FDP_ETC.1) 5.1.2.6 Export of Labeled User Data (FDP_ETC.2) 5.1.2.7 Mandatory Access Control Policy (FDP_IFC.1) 5.1.2.8 Mandatory Access Control Functions (FDP_IFF.2) 5.1.2.9 Import of Unlabeled User Data (FDP_ITC.1) 5.1.2.10 Import of Labeled User Data (FDP_ITC.2) 5.1.3.1 User Attribute Definition (FIA_ATD.1) 5.1.3.6 User-Subject Binding (FIA_USB.1) 5.1.4.1 Management of Object Security Attributes (FMT_MSA.1) 5.1.4.2 Static Attribute Initialization (FMT_MSA.3) 5.1.4.8 Revocation *of Object Attributes* (FMT_REV.1)
O.AUDITING	5.1.1.1 Audit Data Generation (FAU_GEN.1) 5.1.1.2 User Identity Association (FAU_GEN.2) 5.1.1.3 Audit Review (FAU_SAR.1) 5.1.1.4 Restricted Audit Review (FAU_SAR.2) 5.1.1.5 Selectable Audit Review (FAU_SAR.3) 5.1.1.6 Selective Audit (FAU_SEL.1)

Security Objective	Security Functional Requirement
	5.1.1.7 Guarantees of Audit Data Availability (FAU_STG.1) 5.1.1.8 Action in Case of Possible Audit Data Loss (FAU_STG.3) 5.1.1.9 Prevention of Audit Data Loss (FAU_STG.4) 5.1.3.6 User-Subject Binding (FIA_USB.1) 5.1.4.3 Management of *the Audit Trail* (FMT_MTD.1) 5.1.4.4 Management of *Audited Events* (FMT_MTD.1) 5.1.5.4 Reliable Time Stamps (FPT_STM.1)
O.RESIDUAL_INFORMATION	5.1.2.11 Object Residual Information Protection (FDP_RIP.2) 5.1.2.12 Subject Residual Information Protection (Note 1)
O.MANAGE	5.1.1.3 Audit Review (FAU_SAR.1) 5.1.1.5 Selectable Audit Review (FAU_SAR.3) 5.1.1.6 Selective Audit (FAU_SEL.1) 5.1.1.8 Action in Case of Possible Audit Data Loss (FAU_STG.3) 5.1.1.9 Prevention of Audit Data Loss (FAU_STG.4) 5.1.4.3 Management of the Audit Trail (FMT_MTD.1) 5.1.4.4 Management of Audited Events (FMT_MTD.1) 5.1.4.5 Management of User Attributes (FMT_MTD.1) 5.1.4.6 Management of Authentication Data (FMT_MTD.1) 5.1.4.7 Revocation of User Attributes (FMT_REV.1) 5.1.4.9 Specification of Management Functions (FMT_SMF.1) 5.1.4.10 Security Management Roles (FMT_SMR.1)
O.ENFORCEMENT	5.1.5.2 Reference Mediation (FPT_RVM.1) 5.1.5.3 Domain Separation (FPT_SEP.1)
O.NONINTERFERE	5.1.5.1 Failure with preservation of secure state (FPT_FLS.1) 5.1.6.1 Degraded fault tolerance (FRU_FLT.1)
O.NO_COMM	5.1.5.2 Reference Mediation (FPT_RVM.1) 5.1.5.3 Domain Separation (FPT_SEP.1)
O.PARTIAL_SELF_PROTECTION	5.1.5.3 Domain Separation (FPT_SEP.1)

Table 8-5 Mapping Security Objectives to Security Functional Requirements

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 is protected [FIA_ATD.1, FIA_UAU.7, FIA_MTD.1"Management of Authentication Data”]. The strength of the authentication mechanism must be sufficient to ensure unauthorized users to easily pose as authorized users [FIA_SOS.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 (RACF) and FDP_ACC.1 (CP)]. The rules of the DAC policy must be defined [FDP_ACF.1 (RACF) and FDP_ACF.1 (CP)]. 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] and be able to revoke that access [FMT_REV.1 "Revocation of

Object Attributes"]. Protection of named objects must be continuous, starting from object creation [FMT_MSA.3].

O.MANDATORY_ACCESS

> *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]. 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, FMT_MSA.1 “Management of Object Security Attributes”, 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 “Management of the Audit Trail”, FMT_MTD.1 “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 [Note1].

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 [FMT_MTD.1 “Management of the Audit Trail”, FMT_MTD.1 “Management of the Audit Events”, FAU_SEL.1] and review it [FAU_SAR.1, FAU_SAR.3], to manage user accounts [FMT_MTD.1 "Management of User Attributes", FMT_MTD.1 "Management of Authentication Data", FMT_REV.1 "Revocation of User Attributes"]. In order to manage the audit trail effectively, [FAU_STG.3, FAU_STG.4] provides the notification to the administrator.

O.ENFORCEMENT

> *The TSF must be designed and implemented in a manner which 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]. 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.NONINTERFERE

The preservation of a secure state in case of a failure in a virtual machine [FPT_FLS.1] and the degraded fault tolerance [FRU_FLT.1] implement the functionality of a strict separation of virtual machines.

O.NO_COMM

Reference mediation [FPT_RVM.1] ensures that all TSF are invoked prior to execution of any functionality, hence no unauthorized communication between virtual machines can take place. This is supported by the domain separation [FPT_SEP.1], ensuring that TSF cannot be altered by unauthorized subjects or objects in order to bypass them.

O.PARTIAL_SELF_PROTECTION

The domain separation [FPT_SEP.1], ensures that the TOE maintains an execution domain for its own use, protecting it from interference and tampering by untrusted subjects. In addition, the TOE enforces separation between the domains of subjects under control of the TOE.

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.

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.2 the TOE restricts the accessibility of the audit trail to authorized users to prevent the disclosure of confidential information.

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_SEL.1 the administrator can include or exclude auditable events to ensure that only events of interest are audited.

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.

In FAU_STG.4 the prevention of audit loss is stated. This ensures that no security critical event can occur without being audited.

In FDP_ACC.1 (RACF) the different objects that RACF controls with a discretionary access control function are listed.

In FDP_ACC.1 (CP) the different objects that CP controls with a discretionary access control function are listed.

In FDP_ACF.1 (RACF) the discretionary access control function enforced by RACF is described with their rules. This ensures the proper enforcement of access control to objects based on subjects.

In FDP_ACF.1 (CP) the discretionary access control function enforced by CP is described with their rules. This ensures the proper enforcement of access control to objects based on subjects.

In FDP_ETC.1, the export rules for Mandatory Access Control on unlabeled user data are specified. In particular, no security attributes of users must be exported when exporting unlabeled user data.

In contrast, FDP_ETC.2 defines the rules for exporting labeled user data governed under Mandatory Access Control. For exporting labeled user data, the TOE must also export the user's security attributed associated with the exported object.

FDP_IFC.1 governs enforcement of Mandatory Access Control on objects. Based on this requirement, the TOE implements MAC mediating the access to these objects.

In FDP_IFF.2, the basic rules for Mandatory Access Controls are described. This ensures the correct enforcement of MAC upon subjects and objects.

In FDP_ITC.1, the import of unlabeled user data is defined. Based on this requirement, the TOE must ignore any security attributes associated with this user data.

In FDP_ITC.2, the import of labeled user data is defined. In particular, the user data's associated security labels have to be used and interpreted correctly as intended by the source of the data.

In FDP_RIP.2, the TOE implements the object reuse functionality for objects. This prevents any attacker from gaining information from newly allocated resources.

In Note 1, the TOE implements the object reuse functionality for subjects. This prevents any attacker from gaining information from newly allocated resources.

In FIA_ATD.1 additional security attributes of users within the evaluated configuration of z/VM have been added.

In FIA_SOS.1, the password mechanism's quality has been added. This is the lowest boundary for the quality of passwords that can be used in the TOE.

In FIA_UAU.1, the TOE allows dedicated functions to be performed by users prior to authentication. In addition, all other functionality must only be available after successful authentication.

In FIA_UAU.7, the TOE implements an obscured feedback for the authentication process to provide as little information to attackers about the successfulness of his attack as possible.

In FIA_UID.1, the TOE allows dedicated functions to be performed by users prior to identification. In addition, all other functionality must only be available after successful identification.

In FIA_USB.1, the way how z/VM associates real users with tasks is expressed. Based on these attributes, access control checks (based on DAC and MAC rules) are enforced.

In FMT_MSA.1 the ability of the authorized administrator and the object owner to modify access rights for objects is expressed. Both, management of discretionary and mandatory access control is specified.

In order to reduce the effect of uncontrolled access, default values for mandatory and discretionary access control is specified in FMT_MSA.3.

In FMT_MTD.1, the restrictions for audit trail management, the management of audited events, the restrictions for the user attribute management, and the restriction of authentication data management are specified.

In FMT_REV.1 "Revocation of User Attributes" the delayed revocation method has been added, since this is the standard way z/VM 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/VM 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 FMT_MSA.1 and the four instantiations of FMT_MTD.1 for

- Audit trail management
- Audit event management

- User attribute management
- Authentication data management.

Those aspects are listed in this security functional requirement.

FMT_SMR.1 defines roles the TOE maintains.

In FPT_FLS.1, the TOE implements the functionality of secure state preservation regardless the behaviour of software in one virtual machine.

FPT_RVM.1 ensures that prior to the execution of security functions, the TSP enforcement must succeed. This ensures that the security policies are really enforced as specified.

The TOE has to maintain a domain for its own operation to prevent any un-trusted subject or object from tampering with TOE functions or data as specified in FPT_SEP.1. In addition, separation of domains from subjects is enforced.

To maintain the audit trail, the TOE has to provide a reliable time stamp as required in FPT_STM.1.

To prevent any failures occurring in one virtual machine, the TOE implements a degraded fault tolerance functionality, specified in FRU_FLT.1.

8.2.4 Explicit stated security requirements

The explicitly stated SFR ‘Note1’ is drawn from LSPP. A rationale for it is given there.

The ST specifies two explicitly stated SFRs applicable for the TOE and not derived from any PP:

- FPT_SEP_(EXP).1
- FPT_SEP_(EXP).2

Both SFRs support the domain separation for the TOE and for the domain separation of subject domains. They have been added to appropriately reflect all necessary requirements for separating all domains maintained by the TOE.

The following explicit stated SFR applicable for the IT environment is present in the ST:

- FPT_SEP_ENV.1

This SFR reflects the necessary presence of a virtual memory management unit and two execution rings to provide support for domain separation. The two execution rings are implemented with the SIE processor instruction.

These three explicitly stated SFRs where derived from the CC Part 2 SFR FPT_SEP.1. However, this SFR does not provide any selection or assignment operation to model the functionality provided by the TOE and its environment. These new SFRs cannot be regarded as an refinement of FPT_SEP.1, because they do not specify this requirement more precisely, but they present different but similar requirements to FPT_SEP.1.

All security assurance requirements as stated in chapter 6.9 also apply to the three above mentioned explicit stated SFRs, because they extend to the domain separation SFRs already present in the CC with requirements of similar intentions to the already present SFRs.

The three explicit stated SFRs have no dependency on other security functional requirements, similar to the already present SFRs for domain separation.

It is to be noted that all three explicit stated functional requirements are derived from the manual on "Basic Robustness Environments".

8.2.5 Security Requirements Coverage

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

Section	CC Identifier	Security Objective
5.1.1.1	FAU_GEN.1	O.AUDITING
5.1.2.3	FAU_GEN.2	O.AUDITING
5.1.1.3	FAU_SAR.1	O.AUDITING, O.MANAGE
5.1.1.4	FAU_SAR.2	O.AUDITING
5.1.1.5	FAU_SAR.3	O.AUDITING, O.MANAGE
5.1.1.6	FAU_SEL.1	O.AUDITING, O.MANAGE
5.1.1.7	FAU_STG.1	O.AUDITING
5.1.1.8	FAU_STG.3	O.AUDITING, O.MANAGE
5.1.1.9	FAU_STG.4	O.AUDITING, O.MANAGE
5.1.2.1	FDP_ACC.1	O.DISCRETIONARY_ACCESS
5.1.2.2	FDP_ACC.1	O.DISCRETIONARY_ACCESS
5.1.2.3	FDP_ACF.1	O.DISCRETIONARY_ACCESS
5.1.2.4	FDP_ACF.1	O.DISCRETIONARY_ACCESS
5.1.2.5	FDP_ETC.1	O.MANDATORY_ACCESS
5.1.2.6	FDP_ETC.2	O.MANDATORY_ACCESS
5.1.2.7	FDP_IFC.1	O.MANDATORY_ACCESS
5.1.2.8	FDP_IFF.2	O.MANDATORY_ACCESS
5.1.2.9	FDP_ITC.1	O.MANDATORY_ACCESS
5.1.2.10	FDP_ITC.2	O.MANDATORY_ACCESS
5.1.2.11	FDP_RIP.2	O.RESIDUAL_INFORMATION
5.1.2.12	Note 1	O.RESIDUAL_INFORMATION
5.1.3.1	FIA_ATD.1	O.AUTHORIZATION, O.DISCRETIONARY_ACCESS, O.MANDATORY_ACCESS
5.1.3.2	FIA_SOS.1	O.AUTHORIZATION
5.1.3.3	FIA_UAU.1	O.AUTHORIZATION
5.1.3.4	FIA_UAU.7	O.AUTHORIZATION
5.1.3.5	FIA_UID.1	O.AUTHORIZATION
5.1.3.6	FIA_USB.1	O.DISCRETIONARY_ACCESS, O.MANDATORY_ACCESS, O.AUDITING
5.1.4.1	FMT_MSA.1	O.DISCRETIONARY_ACCESS, O.MANDATORY_ACCESS
5.1.4.2	FMT_MSA.3	O.DISCRETIONARY_ACCESS, O.MANDATORY_ACCESS
5.1.4.3	FMT_MTD.1	O.AUDITING, O.MANAGE
5.1.4.4	FMT_MTD.1	O.AUDITING, O.MANAGE
5.1.4.5	FMT_MTD.1	O.MANAGE
5.1.4.6	FMT_MTD.1	O.AUTHORIZATION, O.MANAGE
5.1.4.7	FMT_REV.1	O.MANAGE

Section	CC Identifier	Security Objective
5.1.4.8	FMT_REV.1	O.DISCRETIONARY_ACCESS, O.MANDATORY_ACCESS. O.MANAGE
5.1.4.9	FMT_SMF.1	O.MANAGE
5.1.4.10	FMT_SMR.1	O.MANAGE
5.1.5.1	FPT_FLS.1	O.NONINTERFERE
5.1.5.2	FPT_RVM.1	O.ENFORCEMENT, O.NO_COMM
5.1.5.3	FPT_SEP.1	O.ENFORCEMENT, O.NO_COMM, O.PARTIAL_SELF_PROTECTION
5.1.5.4	FPT_STM.1	O.AUDITING
5.1.6.1	FRU_FLT.1	O.NONINTERFERE

Table 8-6 Mapping Security Functional Requirements to Objectives

8.2.6 Rationale for Security Requirements for the IT Environment

That requirement defines the need for a virtual memory management facility and two execution rings implemented in the underlying abstract machine that allows to reserve the access and manipulation of critical processor and memory resources to specially software (instructions) operating outside the environment specified by the SIE instruction of the processor. The TSF have to ensure that no un-trusted software will ever execute outside the SIE environment. 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. This is addressed by the explicitly stated SFR for the IT environment FPT_SEP_ENV.1, which has no dependencies on other security functional requirements.

The security requirement for the IT environment address the security objective OE.HW_SEP since the virtual memory management facility and the providing of two execution rings allows the TOE to protect the TSF and the TSF data from unauthorized access by un-trusted software. The TOE has to use the virtual memory management facility and the two execution rings to allow memory access by un-trusted software just to those memory areas that belong to the un-trusted software itself. Access to special processor instructions will be managed by the TSF such that this access will always be reserved to trusted software by setting up the SIE instruction appropriately. 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: 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:

SFR	Objective
FPT_SEP_ENV.1	OE.HW_SEP
FPT_AMT.1	OE.HW_SEP

Table 8-7 Mapping Security Functional Requirements for the IT Environment to Objectives

8.2.7 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 – in it indicates that the dependency has not been satisfied.

Section	CC Identifier	ADV_SPM.1	FAU_GEN.1	FAU_SAR.1	FAU_STG.1	FDP_ACC.1	FDP_ACF.1	FDP_IFC.1	FDP_IFF.1	FIA_ATD.1	FIA_UAU.1	FIA_UID.1	FMT_MSA.1	FMT_MSA.3	FMT_MTD.1	FMT_SMF.1	FMT_SMR.1	FPT_FLS.1	FPT_STM.1	FPT_TDC.1	FTP_ITC.1	FTP_TRP.1
5.1.1.1	FAU_GEN.1																		X			
5.1.2.3	FAU_GEN.2		X									X										
5.1.1.3	FAU_SAR.1		X																			
5.1.1.4	FAU_SAR.2			X																		
5.1.1.5	FAU_SAR.3			X																		
5.1.1.6	FAU_SEL.1		X												X							
5.1.1.7	FAU_STG.1		X																			
5.1.1.8	FAU_STG.3				X																	
5.1.1.9	FAU_STG.4				X																	
5.1.2.1	FDP_ACC.1						X															
5.1.2.2	FDP_ACC.1						X															
5.1.2.3	FDP_ACF.1					X								X								
5.1.2.4	FDP_ACF.1					X								X								
5.1.2.5	FDP_ETC.1					O		O														
5.1.2.6	FDP_ETC.2					O		O														
5.1.2.8	FDP_IFC.1								X													
5.1.2.8	FDP_IFF.1							X														
5.1.2.9	FDP_ITC.1					O		O														
5.1.2.10	FDP_ITC.2					O		O												–	–	–
5.1.2.11	FDP_RIP.2																					
5.1.2.12	Note 1																					
5.1.3.1	FIA_ATD.1																					
5.1.3.2	FIA_SOS.1																					
5.1.3.3	FIA_UAU.1											X										
5.1.3.4	FIA_UAU.7										X											
5.1.3.5	FIA_UID.1																					
5.1.3.6	FIA_USB.1									X												
5.1.4.1	FMT_MSA.1					O		O								X	X					
5.1.4.2	FMT_MSA.3												X				X					
5.1.4.3	FMT_MTD.1															X	X					
5.1.4.4	FMT_MTD.1															X	X					
5.1.4.5	FMT_MTD.1															X	X					

Section	CC Identifier	ADV_SPM.1	FAU_GEN.1	FAU_SAR.1	FAU_STG.1	FDP_ACC.1	FDP_ACF.1	FDP_IFC.1	FDP_IFF.1	FIA_ATD.1	FIA_UAU.1	FIA_UID.1	FMT_MSA.1	FMT_MSA.3	FMT_MTD.1	FMT_SMF.1	FMT_SMR.1	FPT_FLS.1	FPT_STM.1	FPT_TDC.1	FTP_ITC.1	FTP_TRP.1
5.1.4.6	FMT_MTD.1															X	X					
5.1.4.7	FMT_REV.1																X					
5.1.4.8	FMT_REV.1																X					
5.1.4.9	FMT_SMF.1																					
5.1.4.10	FMT_SMR.1											X										
5.1.5.1	FPT_FLS.1	X																				
5.1.5.2	FPT_RVM.1																					
5.1.5.3	FPT_SEP.1																					
5.1.5.4	FPT_STM.1																					
5.1.6.1	FRU_FLT.1																	X				
5.3.1	FPT_SEP_ENV.1																					
5.3.2	FPT_AMT.1																					

Table 8-8 IT Security Functional Requirements dependency analysis

Remarks

The dependencies of FMT_MSA.1 and FMT_MTD.1 on FMT_SMF.1 were introduced by CC version 2.3 and have been considered here.

The multiple instantiations of FMT_MTD.1 and FMT_REV.1 have been included in this table, since 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, since they all rely on the same simple role model of the TOE.

This table shows that the following unresolved dependencies exist between security functional requirements of the TOE:

- FDP_ITC.2 depends on FPT_TDC.1 and on either FTP_ITC.1 or FTP_TRP.1. These dependencies are not solved, because they are only required by FDP_ITC.2. However, FDP_ITC.2 applies only to mulit-leveled devices, which do not exist in the TOE, causing the SFR FDP_ITC.2 being not applicable to the TOE. Hence, these dependencies are also not applicable to the TOE.

There are no unresolved dependencies between security assurance requirements. This is because the evaluation assurance level EAL4 with the augmentation ALC_FLR.2 which has no dependencies and therefore there are no unresolved dependencies for assurance components.

The explicitly stated SFRs FPT_SEP_(EXP).1, FPT_SEP_(EXP).2, and FPT_SEP_ENV.1 do not have any dependency on other security functional requirements.

8.2.8 Strength of Function

This Security Target claims a SOF rating 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 prohibits that well-skilled, hostile attackers get physical access to the TOE.

It is to be noted that the password storage using the DES encryption algorithm is not subject to a strength of function claim due to its cryptographic nature.

8.2.9 Evaluation Assurance Level

This security target claims EAL4 augmented with ALC_FLR.2, which is seen appropriate for a well-controlled, non-hostile environment. The flaw remediation assurance component of ALC_FLR.2 has been chosen based on the manual on “Basic Robustness Environments”.

8.3 TOE Summary Specification Rationale

8.3.1 Security Functions Justification

The following table shows that the IT security functions as specified in the TOE summary specification, meet all the security functional requirements for the TOE and work together to satisfy the TOE security functional requirements.

Section	SFR	Security Functions from the TOE Summary Specification
5.1.1.1	FAU_GEN.1	The requirement to generate audit records is met by F.AU, providing for the generation of audit data for the auditable events listed in FAU_GEN.1.
5.1.2.3	FAU_GEN.2	User identity association is achieved by storing the user ID in audit records generated as part of F.AU, and supported by identification of users provided by F.I&A.
5.1.1.3	FAU_SAR.1	F.AU offers functionality to review audit records to authorized users.
5.1.1.4	FAU_SAR.2	Access to audit records is granted to users with the AUDITOR attribute or group-AUDITOR attribute as defined by F.AC.
5.1.1.5	FAU_SAR.3	F.AU offers search functionality for audit records.
5.1.1.6	FAU_SEL.1	F.AU offers search functionality for audit records, based on selectable criteria as listed in FAU_SEL.1. The configuration of the audit functionality is implemented by F.SM.
5.1.1.7	FAU_STG.1	F.AC protects the access to audit records from unauthorized users based on the assignment of resources to the virtual machine running RACF. This is supported by F.I&A that identifies and authenticates users.
5.1.1.8	FAU_STG.3	F.AU ensures the notification of the administrator in case no additional audit records can be stored due to insufficient space.
5.1.1.9	FAU_STG.4	F.AU prohibits any access control check in case no additional audit records can be stored due to insufficient space.
5.1.2.1	FDP_ACC.1	The discretionary access control policy is enforced by the access control function F.AC.
5.1.2.2	FDP_ACC.1	The discretionary access control policy is enforced by the access control function F.AC.
5.1.2.3	FDP_ACF.1	The discretionary access control policy provides the access check mechanism as listed in FDP_ACF.1 and is enforced by F.AC.

Section	SFR	Security Functions from the TOE Summary Specification
5.1.2.3	FDP_ACF.1	The discretionary access control policy provides the access check mechanism as listed in FDP_ACF.1 and is enforced by F.AC.
5.1.2.5	FDP_ETC.1	The export of unlabelled user data is governed by the access control functionality F.AC.
5.1.2.6	FDP_ETC.2	The export of labeled user data is governed by the access control functionality F.AC.
5.1.2.7	FDP_IFC.1	The mandatory access control policy is enforced by the access control function F.AC.
5.1.2.8	FDP_IFF.2	The mandatory access control policy part of F.AC provides the access check mechanism as listed in FDP_IFF.2.
5.1.2.9	FDP_ITC.1	The import of unlabelled user data is governed by the access control functionality F.AC.
5.1.2.10	FDP_ITC.2	The import of labeled user data is governed by the access control functionality F.AC.
5.1.2.11	FDP_RIP.2	The object reuse function F.OR ensures the clearing of resources during allocation to objects.
5.1.2.12	Note 1	The object reuse function F.OR ensures the clearing of resources during allocation to subjects.
5.1.3.1	FIA_ATD.1	User attributes are maintained in order to implement the security functions F.AU, F.AC, F.I&A, F.IP, and F.SM.
5.1.3.2	FIA_SOS.1	The password policy is implemented by F.I&A.
5.1.3.3	FIA_UAU.1	User authentication is implemented by F.I&A.
5.1.3.4	FIA_UAU.7	The identification and authentication function F.I&A returns in case of an authentication failure no other information than the fact of the failed logon.
5.1.3.5	FIA_UID.1	User identification is implemented by F.I&A.
5.1.3.6	FIA_USB.1	User-subject binding is provided by F.I&A by mapping authenticated credentials to users stored in the user database.
5.1.4.1	FMT_MSA.1	F.AC prohibits the modification of access control attributes and sensitivity labels of objects for unauthorized users. This is supported by identification of users provided by F.I&A.
5.1.4.2	FMT_MSA.3	F.AC provides restrictive default values for discretionary and mandatory access control.
5.1.4.3	FMT_MTD.1	The management of the audit trail is provided by F.SM. This is supported by F.AC and F.I&A to protect the configuration of the audit trail.
5.1.4.4	FMT_MTD.1	The management of the audited events is provided by F.SM. This is supported by F.AC and F.I&A to protect the configuration of the audited events.
5.1.4.5	FMT_MTD.1	The management of user attributes is provided by F.SM. This is supported by F.AC and F.I&A to protect the configuration of user attributes.
5.1.4.6	FMT_MTD.1	The management of authentication data is provided by F.SM. This is supported by F.AC and F.I&A to protect the configuration of authentication data.
5.1.4.7	FMT_REV.1	The revocation of user attributes is provided by F.SM. This is supported by

Section	SFR	Security Functions from the TOE Summary Specification
		F.AC and F.I&A to protect the configuration of user attributes and enforce the revocation.
5.1.4.8	FMT_REV.1	The revocation of object attributes is provided by F.SM. This is supported by F.AC and F.I&A to protect the configuration of object attributes and enforce the revocation.
5.1.4.9	FMT_SMF.1	Management of security functions listed in FMT_SMF.1 is provided by F.SM. These management functions are protected by F.AC and F.I&A against unauthorized access.
5.1.4.10	FMT_SMR.1	Maintenance of user roles is primarily implemented by F.I&A. The roles are privileged in the authorization mechanisms implemented by F.AC. The roles are managed by F.SM.
5.1.5.1	FPT_FLS.1	The protection of interference between virtual machines is provided with F.IP. In addition, the TOE protects its own domain with F.TP.
5.1.5.2	FPT_RVM.1	F.I&A ensures that all users are authenticated prior to further security relevant actions, which in turn is subject to access control by F.AC.
5.1.5.3	FPT_SEP.1	F.TP protects the TOE against interference and tampering by un-trusted subjects. F.IP enforces the separation of domains of subjects.
5.1.5.4	FPT_STM.1	For providing accurate audit data, F.AU ensures the generation of a correct time stamp. The management interface of this time stamp is provided by F.SM.
5.1.6.1	FRU_FLT.1	The keeping of a secure state in case of software or hardware failures in virtual machines is provided with F.IP. In addition, the TOE protects its own domain with F.TP.

Table 8-9 Mapping from SFR to Security Funtions

8.3.2 Mutual Support of the Security Functions

The TOE's main purpose is the providing of virtual machines for each logged in user and to serve as a general-purpose operating system that can execute arbitrary software.

In order to control and supervise the correct and secure operation of the TOE, the audit trail stores information about the activity of subjects. The audit facility is provided by F.AU. Audit records are generated and can be reviewed by authorized users. Thus, accountability (as a result of prior authentication) and misuse detection is provided.

In order to allow users (including those in different special roles), identification and authentication of users is provided by F.I&A.

F.AC enforces access control decisions based on administrator-defined access control information for discretionary access control. In addition, administrator-defined sensitivity labels, security categories, and security labels are enforced by F.AC. Administrators themselves are not subject to any access restrictions.

To manage user data, including access control and sensitivity/security attributes for subjects and objects, F.SM provides the necessary interfaces. Also the management of the audit function is provided by F.SM.

For serving the main purpose of providing virtual machines that are strictly separated, F.IP provides the facility to maintain such virtual machines. In addition, F.TP protects the TOE against tampering by and disclosure of confidential information to un-trusted subjects.

Since the TOE dynamically reallocates resources from one subject to another (such as memory or processors), F.OR ensures that these resources are cleared prior to reallocation. This function ensures that no residual information can be transmitted between objects and subjects.

As a result

- no security relevant transactions can be requested by users without being authenticated
- all transactions requested by users are subject to access control
- accountability for transactions is provided
- the management of user data, as well as access control data and the audit facility is controlled and restricted to authorized users
- no interference between virtual machines and between one virtual machine and the TOE can take place, which is not specifically allowed by the virtual machine configurations

8.3.3 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 F.I&A, 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 says 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.

Therefore, a strength of SOF-medium is consistent with the description of the TOE environment. As stated in section 8.2.8, the password storage functionality is not subject to a SOF claim.

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 CAPP, as referenced in [CAPP].

Additional security objectives for the TOE (O.NONINTERFERE, O.NO_COMM, and O.PARTIAL_SELF_PROTECTION) have been defined to reflect the ability of the TOE to prevent any interference including communication between virtual machines (except through explicitly defined communication channels) due to software failure occurring in one virtual machine. Based on this enhancement, two additional security functional requirements are added to implement these objectives (FPT_FLS.1 and FRU_FLT.1). Also the maintenance of a security domain for the TOE's own use and the separation of the domains for subjects are reflected by the explicit stated security functional requirements FPT_SEP_(EXT).1 and FPT_SEP_(EXT).2. They are taken from the manual on "Basic Robustness Environments".

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.

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 green and underlined letters.

In addition 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 has 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).

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.

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 is already satisfied by the underlying hardware as part of the IT environment.

End of document