IBM z/VM Version 6 Release 3 Security Target

Version:	**1.2**
Status:	**Released**
Last Update:	**2014-12-19**

Trademarks

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

- Enterprise Systems Architecture/390
- ESA/390
- IBM
- IBM logo
- HiperSockets
- PR/SM
- Processor Resource/Systems Manager
- RACF
- S/390
- System z
- VM/ESA
- z/Architecture
- z/VM

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

Legal Notice

This document contains information of a confidential nature.

Review and Approval Process: Refer to the inspection process in the System z Software Programming Process.

Required Reviewers

- Brian Hugenbruch

Document Distribution and Change Notification: The document is distributed to the reviewers of this line item. When reissued with changes, the document owner sends a note to the reviewers notifying them of the availability of a new document version.

Archival Requirements: Archival requirement according to the German evaluation and certification scheme is 5 years.

Revision History

Revision	Date	Author(s)	Changes to Previous Revision
0.1	2013-04-04	Stephan Mueller	Initial Draft, rewrite of the version used for z/VM V6.1.
0.2	2013-06-05	Stephan Mueller	Update of cryptographic claims as requested by BSI.
0.3	2013-07-31	Stephan Mueller	Evaluator comments.
0.4	2013-08-29	Stephan Mueller	Update cryptographic SFRs

Revision	Date	Author(s)	Changes to Previous Revision
0.5	2013-10-07	Stephan Mueller	Complete removal of SSLv3/TLSv1.0
0.6	2013-11-05	Stephan Mueller	Clarification of CPU-bound cryptographic mechanisms part of the TOE
0.7	2013-11-14	Stephan Mueller	Synchronization with FIPS 140-2 validation
0.8	2013-12-14	Stephan Mueller	Update CP command reference
0.9	2013-12-18	Stephan Mueller	Addition of APARs
1.0	2014-08-11	Stephan Mueller	Removal of GCM
1.1	2014-12-18	Stephan Mueller	Add RSA key sizes to FCS_CKM.2, remove client operation from FCS_COP.1(NET), add Diffie-Hellman information update RNG SFR
1.2	2014-12-19	Stephan Mueller	update allowed Diffie-Hellman modulus size

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Security Target Identification

Title:	IBM z/VM Version 6 Release 3 Security Target
Version:	1.2
Status:	Released
Date:	2014-12-19
Sponsor:	IBM Corporation
Developer:	IBM Corporation
Certification Body:	BSI
Certification ID:	BSI-DSZ-CC-0903
Keywords:	access control, discretionary access control, general-purpose operating system, information protection, security labels, mandatory access control, security, virtual machine

1.2 TOE Identification

The TOE is IBM z/VM Version 6 Release 3.

1.3 TOE Type

The TOE type is virtual machine operating system implementing a hypervisor.

1.4 TOE Overview

This security target (ST) documents the security characteristics of the IBM z/VM hypervisor product when configured in a secure manner according to the supplied security guide.

z/VM is a highly secure, flexible, robust, scalable virtual machine hypervisor for IBM System z® mainframe servers onto which to deploy mission-critical virtual servers. A single System z server can host one z/VM instance per logical partition (LPAR), and each instance of z/VM can host tens to hundreds of virtual servers. Multiple instances of z/VM can be connected to form a networked system called a "collection". The communication aspects within z/VM used for these connections 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.

z/VM offers multi-system clustering technology allowing between one and four z/VM instances in a single system image (SSI) cluster. New instances of z/VM can be added to the cluster topology at runtime. Support for live guest relocation (LGR) allows the movement of Linux virtual servers without disruption to the operation. The z/VM systems are aware of each other and can take advantage of their combined resources. LGR enables clients to avoid loss of service due to planned outages by relocating guests from a system requiring maintenance to a system that remains active during the maintenance period.

Due to the functionality of performing identification and authentication of users, implementation of DAC and MAC, providing management facilities for all security-related functions and the fact that support functionality is hosted in different virtual machines, z/VM also resembles an operating

system. Therefore, the Operating System Protection Profile ([OSPP]) is used as a basis for this ST. z/VM meets all of the requirements of the Operating System Protection Profile base [OSPP], as well as its extended packages for labeled security [OSPP-LS] and virtualization [OSPP-VIRT].

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

1.5 TOE Description

The Target of Evaluation (TOE) is the z/VM hypervisor product that is part of an SSI cluster formed by one or more z/VM instances with the software components as described in section 1.5.4. z/VM is an operating system designed to host other operating systems, each in its own virtual machine. Multiple virtual machines can run concurrently 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 use of certain options of the processor's DIAGNOSE instruction. In addition to the system directory, the RACF security server is employed to mediate access to resources and privileged functions.

For the purpose of this ST, the TOE is one instance of an z/VM SSI cluster comprising of one through four individual z/VM systems. These individual z/VM systems execute 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 is provided by a logical partition (LPAR) of an IBM System z server.

The LPAR itself is not part of the TOE, but belongs to the TOE environment. Please note 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 within an LPAR. A z/VM instance running within a virtual machine is allowed, but this "second level" z/VM instance is not in an evaluated configuration, as some security functionality is implemented differently, in particular with respect to the usage of the processor's Start Interpretive Execution (SIE) instruction.

The z/VM Single System Image feature (SSI) enables up to four z/VM systems to be configured as members of an SSI cluster, sharing different resources

Members of the SSI cluser can be on the same or separate CECs. SSI enables the members of the cluster to be managed as one system, which allows service to be applied to each member of the cluster, avoiding an outage to the entire cluster. SSI also introduces the concept of live guest relocation (LGR) where a running Linux guest operating system can be relocated from one member in an SSI cluster to another without the need to stop the running Linux guest.

All z/VM member instances of one SSI cluster share the RACF database, but they do not share the RACF audit disks. Each z/VM member instance must execute its own instance of RACF accessing the shared RACF database. The sharing of the RACF database is done by sharing the DASD (direct access storage device) volume keeping the RACF database between the different SSI z/VM member instances. Although sharing of the RACF database between z/VM and z/OS is technically feasible, it is explicitly excluded from this evaluation.

Different instances of the TOE may also share the RACF database. The sharing is implemented similarly to the sharing of the RACF database within the SSI cluster. However depending on the use scenario, such sharing may not be advisable.

The platforms selected for the evaluation consist of IBM products, which are available when the evaluation has been completed and will remain available for some period of time afterwards. Even if withdrawn from general marketing, the product may be obtained by special request to IBM.

The TOE security functions (TSF) are provided by the z/VM operating system kernel (called the Control Program – CP) and by an application called RACF that runs within a specially-privileged virtual machine. In addition to providing user authentication, access control, and audit services to CP, RACF can provide the same services to other authorized virtual machines. 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: a standard mode meeting all requirements of the Operating System Protection Profile base [OSPP] and its extended package for Virtualization [OSPP-VIRT], and a more restrictive mode called Labeled Security Mode, which additionally meets all requirements of the OSPP extended package for Labeled Security [OSPP-LS]. In both modes, the same software elements are used. The two modes have different RACF settings with respect to the use of security labels. All other configuration parameters are identical in the two modes.

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

1.5.1 Structure and concept of z/VM

z/VM provides a mechanism to run a heterogeneous mix of z/Architecture® or Enterprise System Architecture/390 (ESA/390) operating systems with overcommitment of processor and memory resources. 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 CPU, memory, operator controls, and input/output (I/O) resources.

Because each virtual machine provides an environment that conforms to the machine architecture, the virtual machine can host a conformant operating system (called a guest). Multiple instances of Linux, z/OS, z/VSE, z/TPF, and even z/VM itself, can run concurrently on the same z/VM system that is supporting z/VM CMS applications and end users. As a result, application development, testing, and production environments can share a single physical computer.

1.5.1.1 Differences from other operating systems

z/VM is similar to any other operating system in that it implements 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 Labeled Security 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 from a general-purpose operating system 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. That software must be an operating system, regardless whether it supports a single user or multiple users. Applications run within the operating system. A virtual machine differs from application environment of general-purpose operating systems in the following ways:

- 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 to 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 is capable of maintaining a m:n association between the number of logical processors available in the logical partition z/VM is running in, and the number of virtual processors defined for a virtual machine. Given that an arbitrary number of virtual machines may be active at any one time, it is possible to have more virtual processors demanding access to the CPU than are available in the partition. This time-sharing of the available CPU capacity is implemented in CP by having a pre-emptive scheduler for virtual processors. Preemption is implemented by utilizing the SIE instruction’s timing mechanism, which allows CP to specify how long the SIE environment is executed by one processor. After the expiry of the timer, the SIE instruction ends, the virtual processor stops, and control is returned to CP.

1.5.1.2 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 single-user general-purpose operating system that is employed to run the RACF and TCP/IP applications. See section 1.5.2.1 for details on the intended usage of CMS in the evaluated configuration.
- RACF server: provides authentication, authorization, and audit services to CP and other authorized virtual machines that run applications on CMS. It communicates with CP through a tightly-controlled well-defined interface.
- TCP/IP server: provides traditional IP-based communications services. It is not part of CP, but runs within a virtual machine. Embedded within the TCP/IP stack is the Telnet service that enables users to access their virtual machine consoles (“log on”) from the IP network. In particular, this Telnet service receives console traffic from the network, removes the telnet or TN3270 protocol wrappers, and then forwards it to CP using a special form of 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 or TN3270E protocol and sends it back to the client. The TCP/IP server also provides TLS allowing the establishment of a cryptographically secured channel.

1.5.1.3 User’s management of virtual machines using the Control Program

The login facility is provided by the Control Program (CP). When a user connects to the z/VM system, they are presented with a welcome message or logo screen. At that point they are prompted to provide their credentials, which consist of a user ID (the name of the virtual machine) and a password. CP hands the credentials to RACF for validation and, if successful, creates the virtual machine environment and connects the user’s terminal to the virtual machine. This connection is referred to as the *virtual machine operator’s console*, or simply the *virtual console*.

CP provides an interactive shell on the virtual console with which the user can enter CP commands to manipulate the virtual machine. After an operating system is IPLed, the console is treated as an I/O device visible to the guest operating system. If desired, the user may explicitly direct the console to communicate with CP. Thus the console serves two purposes after IPL. There may be multiple terminal devices (e.g. 3270s) available to the virtual machine, but only the virtual machine operator’s console be used to communicate with CP. Any other terminal devices can communicate only with the guest operating system.

The virtual machine definition typically contains a directive to cause the guest operating system to start automatically at logon. If not, the user can manually IPL the guest.

The virtual machine is initialized with an administrator-predefined virtual machine definition. While the virtual machine is running, 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.

If the virtual machine definition contains a CONSOLE statement, the virtual console will be visible to the software running in the virtual machine and can be used to communicate with the guest operating system or applications.

Interfaces to CP for software running in virtual machines are provided using processor instructions (DIAGNOSE, IUCV).

1.5.1.4 Communication between virtual machines and with the Control Program

z/VM offers the following communication facilities:

- A virtual LAN segment or switch that simulates an Ethernet or System z HiperSockets network. They can be used to communicate between virtual machines or, in the case of the virtual switch, with external (physical) LAN segments. This is provided as part of the virtual I/O subsystem for the virtual machine.
- VMCF (Virtual Machine Communication Facility) provides bidirectional communication channels between virtual machines. This is provided by the DIAGNOSE 0x68 instruction.
- IUCV (Inter-User Communication Vehicle) offers bidirectional communication channels between virtual machines or between a virtual machine and CP. This is provided by the IUCV (0xB2F0) instruction.
- CP commands MESSAGE (MSG), SMSG, and WARNING (WNG) provide unidirectional communication channels between virtual machines. Users can send messages to each other, but there is no "reply" mechanism. This is available to the guest by command at the virtual console or by the use of DIAGNOSE 0x08 by the guest operating system.
- Virtual Channel-To-Channel simulates the functions of a point-to-point channel connection, providing a bidirectional communication channel between virtual machines. This is provided as part of the virtual I/O subsystem for the virtual machine.

All listed communication channels are established and maintained by CP. CP protects them against spoofing or eavesdropping ("sniffing").

In addition to the explicitly-defined communication channels above, CP allows the configuration of:

- Shared 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 such as Reserve/Release to avoid data inconsistencies on shared disk space.
- Shared 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 virtual processors belonging to the signaling virtual machine. CP ensures that these signals do not traverse the virtual machine boundary.

1.5.1.5 Single System Image (SSI) Cluster

The TOE is defined as an SSI cluster. That cluster may contain one or more z/VM systems. All members of the SSI cluster are part of the same ISFC collection. Every member of the SSI cluster must have direct ISFC connections to every other member of the SSI cluster. CP validates and manages all resources and data sharing using ISFC messages that flow across channel-to-channel

(CTC) connections between members. It is permissible to configure one or more CTC connections between each pair of members to increase throughput and reduce the relocation and quiesce time needed for the live guest relocation feature.

The different z/VM systems forming the SSI cluster may reside on the same or on different CECs. If they reside on different CECs, the physical systems must be close enough to allow FICON CTC connections, the use of shared DASD as well as common network and disk fabric connections.

Using the Single System Image (SSI) cluster configuration, z/VM systems can be configured to share the following resources:

- Common volumes: Contain the user directory, and system config file
- Release volumes: Contain one set of disks per z/VM release per cluster
- Spool volumes: Require at least one spool volume for each member of the cluster and the spool volumes are owned by that member. A member can only create spool files on volumes that it owns, but all members can view and update files on all spool volumes.
- DASD volumes for other uses
- User minidisks
- Network devices

In addition to shared volumes, the z/VM member system also operate with non-shared volumes. These volumes are private to the z/VM system and are not visible or accessible by other SSI cluster members. Each member has at least the following non-shared volumes:

- Sysres volume
- Paging volume
- Private user volumes

The persistent data record (PDR) is information that must reside on a shared fullpack 3390 volume, usually the common volume. It is used to provide a cross-system serialization point on disk. It contains information about member status and the heartbeat data. It is also used for health-checking and ensures that a stalled or stopped member can be detected.

Each member of the SSI cluster must have identical network connectivity in terms of the virtual switch name, having one or more physical OSA ports connected to the same physical LAN segment. The assignment of MAC addresses to network interface cards (NICs) is coordinated across the SSI cluster. Only with a coordinated network topology a Linux guest relocates across the cluster without any operational disruption. SSI does not allow the members of the SSI cluster to have a MAC address that is already in use by another Linux guest within the cluster.

Every z/VM member system of an SSI cluster uses its own instance of RACF. The RACF database is shared among the SSI cluster members. To support the sharing of the RACF database, the use of fullpack 3390 volumes is required. RACF Volumes other than the RACF database are private to the respective RACF instance. This includes the audit volumes maintained by each RACF instance.

Systems Management is simplified in an SSI cluster. Service can be applied to z/VM from any member within the cluster, but it needs to be put into production on each member. Service updates can be planned over a period of time and can be applied to a single member of the cluster, and then rolled out to other members of the cluster later.

Live guest relocation

The SSI cluster functionality supports live guest relocation for Linux on System z guests. The relocation operation takes place while the Linux system is running and without an interruption in its offered service. During the final phase of the migration, the Linux system is stopped for an administrator-configurable maximum time, the quiesce time, in order to complete the migration. If the final migration phase cannot be completed within the quiesce time window, the Linux instance on the originating z/VM system is continued. Otherwise, the newly migrated Linux instance is continued. Using the live guest relocation, the SSI cluster can be used for:

- planned software or hardware maintenance that can be applied to a single member without affecting the other members of the cluster or disrupting running Linux guest operating systems. The Linux guests can be relocated to another member in the cluster prior to maintenance being applied to the original member that hosted the Linux guest.
- workload balancing, because Linux guests can be manually relocated to members that are under-utilized.

The z/VM control program (CP) attempts to relocate all the memory of the Linux guest in a series of passes, on each pass only moving the memory that changed since the last pass. This process continues until an internal algorithm determines that the guest needs to be quiesced to move the last memory pages. CP quiesces the Linux guest and relocates the final guest state, I/O information, and changed pages.

During the initialization of the SSI cluster or the joining phase, the participating z/VM members negotiate the common virtualization environment that can be offered to the virtual machines. If the different z/VM members execute on hardware with different capabilities, such as a z10 and a z196, the virtual machine support is limited to the common environment of a z10 in this case. Virtual machines started on a z196 will be presented a z10 environment.

To avoid such limitations, relocation domains can be configured for an SSI cluster. A relocation domain limits the allowed movement of a virtual machine to z/VM member systems belonging to the relocation domain the virtual machine was started in. When using relocation domains, the agreement of the supported virtual environments mentioned above is limited to the context of a relocation domain's definition. Continuing the example above: when a cluster has one z10 system and two z196 systems and the administrator configured the two z196 systems to form one relocation domain, all virtual machines started on each of the z196 system are offered a virtual environment resembling a z196. This, however, implies that the virtual machine cannot be relocated to the z10 which is outside of the configured relocation domain.

Single configuration users and multiconfiguration users

Single configuration users are z/VM users in the traditional sense They are defined in the user directory with USER entries and can only log on to one member in the cluster at a time. They have the same definitions and access to the same resources on all members in the SSI cluster. Typically, USERs are guest operating systems or service virtual machines that require only one logon in the cluster.

Multiconfiguration users are IDENTITY entries, which are new to SSI clusters, in the user directory. They can log on to multiple members in the cluster at the same time. Like USERs, they have definitions and resources, which are common to all members. In addition, they have definitions and resources, which are unique to the particular member they are logged on to. These definitions and resources are the SUBCONFIG statements. Typically, IDENTITY directory entries are for system support and service virtual machines.

z/VM SSI cluster operation

A z/VM system that is configured as a member of an SSI cluster joins the cluster when the system is IPLed. A member leaves the cluster when it shuts down.

In the SSI cluster, the members can be in various states. The overall mode of the cluster depends on the combined states of the individual members. The state of each member and the cluster mode determine the degree of communication, resource sharing, and data sharing among the members.

The information about the state of each member of the SSI cluster is held in the persistent data record (PDR), which is located on the shared common disk. This record contains a heartbeat from all the members of the client so that stalled or stopped members can be identified quickly.

The cluster has a number of modes depending on the state of the individual members of the cluster. The following modes are the SSI cluster modes:

- Stable: SSI cluster is fully operational.
- Influx: Members are in the process of joining or leaving the cluster. Cross-system functions are temporarily suspended in this mode, and negotiations for shared resources are deferred. Any existing accesses are unaffected.
- Safe: This mode occurs when a remote member's state cannot be determined, or any member is in a suspended state. It results in a failure to access any shared resources. Any existing accesses are unaffected.

1.5.2 Intended Method of Use

z/VM provides a general computing environment that allows users (virtual machines) to gain controlled access to Control Program (CP)-managed resources in different ways:

- Using 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 in the virtual machine definition. The operating system just "sees" those resources assigned to the virtual machine.
- Access to additional authorized resources by use of CP commands issued from the virtual console or via the DIAGNOSE 0x08 instruction. The virtual console may be accessed from
    - A local terminal physically attached to the system,
    - A remote terminal accessing the system using the telnet service provided by the TCP/IP stack application running in a dedicated virtual machine,
    - Another virtual machine that has been authorized to take control of the user's virtual console.
- 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 or other virtual machines from inside the virtual machine using the processor's DIAGNOSE instruction, which is defined by the architecture to be intercepted in all cases and simulated by CP.

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 Labeled Security Mode) a security label.

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

Apart from normal users, z/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.

1.5.2.1 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.

z/VM offers two "flavors" of CMS: the 32 bit version as well as the 64-bit zArchitecture-enabled CMS ("zCMS"), which resides on the MAINT 990. Both flavors of CMS implement the same concepts and form a mediation library between applications executing in virtual machines and CP. Therefore, all statements in this section apply to both flavors.

CMS is a single-user general-purpose operating system delivered with z/VM. It is to be used to run the TCP/IP application in a virtual machine and the RACF security server. Customers can write their own applications to be run on CMS either its native or POSIX-conformant application programming interfaces (APIs).

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 IUCV) 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 the filesystem servers:

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

- Access control and auditing 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. However, it is permitted to boot CMS from the commonly shared read only code segment (Named Saved System, NSS) containing the CMS binary object code.

These restrictions allow CMS to be treated 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 runtime environment for TOE applications.

1.5.3 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 Labeled Security Mode
- Separation of virtual machines
- Audit
- Object reuse functionality
- Security management
- TSF protection
- SSI clustering

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

1.5.3.1 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.

1.5.3.2 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 1.5.3.3)
- Access authority specified in the resource profile

1.5.3.3 Mandatory Access Control and Support for Security Labels

In addition to DAC, z/VM provides Mandatory Access Control (MAC), 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.

1.5.3.4 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.

1.5.3.5 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 Labeled Security Mode) MAC mechanism.

1.5.3.6 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

1.5.3.7 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 Labeled Security 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.

1.5.3.8 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 Interpretive-Execution Facility (SIE instruction). Therefore the hardware and firmware components providing the abstract machine for the TOE are required to be physically protected from unauthorized access.

1.5.3.9 SSI clustering

The SSI clustering mechanism integrates different z/VM systems into one cluster in order to share different resources. The SSI cluster communication ensures serialization of concurrent access to shared resources, if needed.

The main goal of SSI is the support of live guest migration of virtual machines. The CP ensures the transfer of the virtual machine memory and state to another SSI cluster member without the interruption of the service of the virtual machine.

1.5.4 Configurations

1.5.4.1 Software Components

The Target of Evaluation, IBM z/VM Version 6 Release 3, requires the following software components to be installed, enabled, and configured:

- CMS for operating RACF and TCP/IP
- Control Program (CP)

- RACF Security Server feature
- TCP/IP for z/VM
- SSI feature selected at installation time

Apart from these required elements, the following elements may be used in the system:

- TLS support for the network communication

The following APARs must be installed for the TOE:

- APAR VM65473 which applies service 6302RSU
- All PTFs associated with APAR VM65474

1.5.4.2 Software Privileges

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.

1.5.4.3 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.

1.5.4.4 Hardware configurations

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

In this ST, the TOE is seen as an SSI cluster formed by one or more instances of z/VM. z/VM executes on an abstract machine as the sole operating system and exercising full control over this abstract machine. This abstract machine can be provided by one of the following: a logical partition provided by a certified version of PR/SM on an IBM System z processor:

- IBM System z10 Business Class with CP Assist for Cryptographic Functions (CPACF) DES/TDES Enablement Feature 3863 active
- IBM System z10 Enterprise Class with CPACF DES/TDES Enablement Feature 3863 active
- IBM zEnterprise 114 with CPACF DES/TDES Enablement Feature 3863 active
- IBM zEnterprise 196 with CPACF DES/TDES Enablement Feature 3863 active
- IBM zEnterprise EC12 with CPACF DES/TDES Enablement Feature 3863 active

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

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

- all terminals supported by the TOE
- all storage devices supported by the TOE
- all network adapters supported by the TOE

2 CC Conformance Claim

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

This Security Target claims conformance to the following Protection Profiles and PP packages, if any:

- [OSPP]: Operating System Protection Profile (with exception of SFR FCS_RNG.1, which is superseded by FCS_RNG.1 in section 5.1). Version 2.0 as of 2010-06-01; strict conformance.
- [OSPP-LS]: OSPP Extended Package - Labeled Security. Version 2.0 as of 2010-05-28; strict conformance.
- [OSPP-VIRT]: OSPP Extended Package - Virtualization. Version 2.0 as of 2010-05-28; strict conformance.

Common Criteria [CC] version 3.1 revision 4 is the basis for this conformance claim.

3 Security Problem Definition

3.1 Threat Environment

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

3.1.1 Assets

Assets to be protected are:

- Persistent storage objects used to store user data and/or TSF data, where this data needs to be protected from any of the following operations:
  - Unauthorized read access
  - Unauthorized modification
  - Unauthorized deletion of the object
  - Unauthorized creation of new objects
  - Unauthorized management of object attributes
- Transient storage objects holding user data and/or TSF data, including network data
- TSF functions and associated TSF data
- The resources managed by the TSF that are used to store the above-mentioned objects, including the metadata needed to manage these objects
- SSI cluster communication data exchanged between the members of an SSI cluster

3.1.2 Threat agents

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

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

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

The following threats are addressed by the TOE. All threats present in [OSPP] and extended packages have been copied. Additionally, threats targeting the cluster functionality have been added. As in the [OSPP], there are no threats and policies to justify the assurance level.

3.1.3 Threats countered by the TOE

T.ACCESS.TSFDATA

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

T.ACCESS.USERDATA

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

T.ACCESS.TSFFUNC

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

T.ACCESS.COMM

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

T.RESTRICT.NETTRAFFIC

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

T.IA.MASQUERADE

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

T.IA.USER

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

T.DATA_NOT_SEPARATED

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

T.ACCESS.COMPENV

A threat agent might utilize or modify the runtime environment of other compartments in an unauthorized manner.

T.INFOFLOW.COMP

A threat agent might get access to information without authorization by the information flow control policy.

T.COMM.COMP

A threat agent might access the data communicated between compartments or between a compartment and an external entity to read or modify the transferred data.

T.SSI.LGR

A threat agent might read, alter, replay, delete, or newly create the data representing a virtual machine that is communicated between SSI cluster members. The threat agent might initiate, prevent, interrupt, or redirect the virtual machine data transmission between SSI cluster members.

3.2 Assumptions

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

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

3.2.1 Environment of use of the TOE

3.2.1.1 Physical

A.PHYSICAL

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

3.2.1.2 Personnel

A.MANAGE

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

A.AUTHUSER

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

A.TRAINEDUSER

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

3.2.1.3 Procedural

A.DETECT

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

A.PEER.MGT

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

A.PEER.FUNC

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

3.2.1.4 Connectivity

A.CONNECT

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

3.3 Organizational Security Policies

P.ACCOUNTABILITY

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

P.USER

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

P.CLEARANCE

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

P.LABELED_OUTPUT

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

P.RESOURCE_LABELS

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

P.USER_CLEARANCE

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

4 Security Objectives

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

4.1 Objectives for the TOE

O.AUDITING

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

O.CRYPTO.NET

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

O.DISCRETIONARY.ACCESS

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

O.NETWORK.FLOW

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

O.SUBJECT.COM

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

O.I&A

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

O.MANAGE

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

O.TRUSTED_CHANNEL

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

O.LS.CONFIDENTIALITY

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

O.LS.PRINT

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

O.LS.LABEL

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

O.COMP.INFO_FLOW_CTRL

The TOE will control information flow between compartments under the control of the TOE, based on security attributes of these compartments and potentially other TSF data (e.g., security attributes of objects). This information flow control policy must be able to allow the isolation of individual compartments from other compartments controlled by the TOE.

O.COMP.RESOURCE_ACCESS

The TOE will control access of compartments to objects and resources under its control based on:

- security attributes of the objects,
- security attributes of the compartment that attempts to access the object, and
- the type of access attempted.

The rules that determine access may be based on the value of other TSF data. Access must be controlled down to individual compartments and objects.

O.COMP.IDENT

For each access request, the TOE is able to identify the compartment requesting to access resources, objects or information.

O.SSI.VMDATAPROTECTION

The TOE prevents any access to the transmission channels used for the communication of virtual machine data between SSI cluster members by any user.

O.SSI.TSFDATAPROTECTION

The TOE prevents any access to the transmission channels used for the communication of SSI cluster state information between SSI cluster members by any user.

O.SSI.LGRMGT

The TOE restricts the operational control of the SSI cluster communication to authorized administrators.

4.2 Objectives for the Operational Environment

OE.ADMIN

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

OE.REMOTE

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

OE.INFO_PROTECT

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

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

OE.INSTALL

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

OE.MAINTENANCE

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

OE.PHYSICAL

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

OE.RECOVER

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

OE.TRUSTED.IT.SYSTEM

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

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

4.3 Security Objectives Rationale

4.3.1 Security Objectives Coverage

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

Objective	Threats / OSPs
O.AUDITING	P.ACCOUNTABILITY
O.CRYPTO.NET	T.ACCESS.TSFDATA T.ACCESS.USERDATA T.ACCESS.TSFFUNC
O.DISCRETIONARY.ACCESS	T.ACCESS.TSFDATA T.ACCESS.USERDATA
O.NETWORK.FLOW	T.RESTRICT.NETTRAFFIC
O.SUBJECT.COM	T.ACCESS.TSFDATA T.ACCESS.USERDATA
O.I&A	T.IA.MASQUERADE T.IA.USER
O.MANAGE	T.ACCESS.TSFFUNC P.ACCOUNTABILITY P.USER
O.TRUSTED_CHANNEL	T.ACCESS.COMM
O.LS.CONFIDENTIALITY	T.DATA_NOT_SEPARATED P.CLEARANCE P.USER_CLEARANCE
O.LS.PRINT	P.LABELED_OUTPUT
O.LS.LABEL	P.RESOURCE_LABELS P.USER_CLEARANCE

Objective	Threats / OSPs
O.COMP.INFO_FLOW_CTRL	T.INFOFLOW.COMP
O.COMP.RESOURCE_ACCESS	T.ACCESS.COMPENV T.COMM.COMP
O.COMP.IDENT	T.ACCESS.COMPENV T.INFOFLOW.COMP T.COMM.COMP
O.SSI.VMDATAPROTECTION	T.SSI.LGR
O.SSI.TSFDATAPROTECTION	T.SSI.LGR
O.SSI.LGRMGT	T.SSI.LGR

Table 1: Mapping of security objectives to threats and policies

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

Objective	Assumptions / Threats / OSPs
OE.ADMIN	A.MANAGE A.AUTHUSER A.TRAINEDUSER
OE.REMOTE	A.CONNECT T.ACCESS.COMM
OE.INFO_PROTECT	A.PHYSICAL A.MANAGE A.AUTHUSER A.TRAINEDUSER P.USER
OE.INSTALL	A.MANAGE A.DETECT
OE.MAINTENANCE	A.DETECT
OE.PHYSICAL	A.PHYSICAL
OE.RECOVER	A.MANAGE A.DETECT
OE.TRUSTED.IT.SYSTEM	A.PEER.MGT A.PEER.FUNC A.CONNECT

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

4.3.2 Security Objectives Sufficiency

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

Threat	Rationale for security objectives
T.ACCESS.TSFDATA	The threat of accessing TSF data without proper authorization is removed by: ● O.CRYPTO.NET requiring cryptographically-protected communication channels for data including TSF data controlled by the TOE in transit between trusted IT systems, ● O.DISCRETIONARY.ACCESS requiring that data, including TSF data stored with the TOE, have discretionary access control protection, ● O.SUBJECT.COM requiring the TSF to mediate communication between subjects.
T.ACCESS.USERDATA	The threat of accessing user data without proper authorization is removed by: ● O.CRYPTO.NET requiring cryptographically-protected communication channels for data including user data controlled by the TOE in transit between trusted IT systems, ● O.DISCRETIONARY.ACCESS requiring that data including user data stored with the TOE, have discretionary access control protection, ● O.SUBJECT.COM requiring the TSF to mediate communication between subjects.
T.ACCESS.TSFFUNC	The threat of accessing TSF functions without proper authorization is removed by: ● O.CRYPTO.NET requiring cryptographically-protected communication channels to limit which TSF functions are accessible to external entities, ● O.MANAGE requiring that only authorized users utilize management TSF functions.
T.ACCESS.COMM	The threat of accessing a communication channel that establishes a trust relationship between the TOE and another remote trusted IT system is removed by: ● O.TRUSTED_CHANNEL requiring that the TOE implements a trusted channel between itself and a remote trusted IT system protecting the user data and TSF data transferred over this channel from disclosure and undetected modification and prevents masquerading of the remote trusted IT system, ● OE.REMOTE requiring that those systems providing the functions required by the TOE are sufficiently protected from any attack that may cause those functions to provide false results.

Threat	Rationale for security objectives
T.RESTRICT.NETTRAFFIC	The threat of accessing information or transmitting information to other recipients via network communication channels without authorization for this communication attempt is removed by: • O.NETWORK.FLOW requiring the TOE to mediate the communication between itself and remote entities in accordance with its security policy.
T.IA.MASQUERADE	The threat of masquerading as an authorized entity in order to gain unauthorized access to user data, TSF data or TOE resources is removed by: • O.I&A requiring that each entity interacting with the TOE is properly identified and authenticated before allowing any action the TOE is defined to provide to authenticated users only.
T.IA.USER	The threat of accessing user data, TSF data or TOE resources without being identified and authenticated is removed by: • O.I&A requiring that each entity interacting with the TOE is properly identified and authenticated before allowing any action the TOE has defined to provide to authenticated users only.
T.DATA_NOT_SEPARATED	The threat of not adequately separating data on the basis of its sensitivity label, thereby allowing information to flow illicitly from or to users, is removed by: • O.LS.CONFIDENTIALITY requiring the TOE to control information flow between entities and resources, based on the sensitivity labels of users and resources.
T.ACCESS.COMPENV	The threat of utilizing or modifying the runtime environment of compartments executing on behalf of other users is removed by: • O.COMP.RESOURCE_ACCESS requiring the TOE to control access of compartments to objects and resources under its control. • O.COMP.IDENT requiring the TOE to identify the compartment requesting to access resources, objects or information for each access request.
T.INFOFLOW.COMP	The threat of accessing information without authorization by the information flow control policy is removed by: • O.COMP.INFO_FLOW_CTRL requiring the TOE to control information flow between compartments under the control of the TOE based on security attributes of these compartments and potentially other TSF data. • O.COMP.IDENT requiring the TOE to identify the compartment requesting to access resources, objects or information for each access request.
T.COMM.COMP	The threat of accessing the data communicated between compartments or between a compartment and an external entity is removed by: • O.COMP.RESOURCE_ACCESS requiring the TOE to control access of compartments to objects and resources under its control,

Threat	Rationale for security objectives
	• O.COMP.IDENT requiring the TOE to identify the compartment requesting to access resources, objects or information for each access request.
T.SSI.LGR	The threat of reading, altering, replaying, deleting, or newly creating the data representing a virtual machine that is communicated between SSI cluster members as well as the threat of initiating, preventing, interrupting, or redirecting the virtual machine data transmission between SSI cluster members is removed by: • O.SSI.VMDATAPROTECTION requiring the TOE to prevent any access to the transmission channels used for the communication of virtual machine data between SSI cluster members by any user, • O.SSI.TSFDATAPROTECTION requiring the TOE to prevent any access to the transmission channels used for the communication of SSI cluster state information between SSI cluster members by any user, • O.SSI.LGRMGT requiring the TOE to restrict the operational control of the live guest relocation communication to authorized administrators.

Table 3: Sufficiency of objectives countering threats

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

Assumption	Rationale for security objectives
A.PHYSICAL	The assumption on the IT environment to provide the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE is covered by: • OE.INFO_PROTECT requiring the approval of network and peripheral cabling, • OE.PHYSICAL requiring physical protection.
A.MANAGE	The assumptions on the TOE security functionality being managed by one or more trustworthy individuals is covered by: • OE.ADMIN requiring trustworthy personnel managing the TOE, • OE.INFO_PROTECT requiring personnel to ensure that information is protected in an appropriate manner, • OE.INSTALL requiring personnel to ensure that components that comprise the system are distributed, installed and configured in a secure manner supporting the security mechanisms provided by the TOE, • OE.RECOVER requiring personnel to assure that after system failure or other discontinuity, recovery without a protection (security) compromise is achieved.

Assumption	Rationale for security objectives
A.AUTHUSER	The assumption on authorized users to possess the necessary authorization to access at least some of the information managed by the TOE and to act in a cooperating manner in a benign environment is covered by: ● OE.ADMIN ensuring that those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of the information it contains, ● OE.INFO_PROTECT requiring that DAC protections on security-relevant files (such as audit trails and authentication databases) shall always be set up correctly and that users are authorized to access parts of the data maintained by the TOE.
A.TRAINEDUSER	The assumptions on users to be sufficiently trained and trusted to accomplish some task or group of tasks within a secure IT environment by exercising complete control over their user data is covered by: ● OE.ADMIN requiring competent personnel managing the TOE, ● OE.INFO_PROTECT requiring that those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner and that users are trained to exercise control over their own data.
A.DETECT	The assumption that modification or corruption of security-enforcing or security-relevant files will be detected by an administrative user is covered by: ● OE.INSTALL requiring an administrative user to ensure that the TOE is distributed, installed and configured in a secure manner supporting the security mechanisms provided by the TOE, ● OE.MAINTENANCE requiring an administrative user to ensure that the diagnostics facilities are invoked at every scheduled preventative maintenance period, verifying the correct operation of the TOE, ● OE.RECOVER requiring an administrative user to ensure that procedures and/or mechanisms are provided to assure that after system failure or other discontinuity, recovery without a protection (security) compromise is achieved.
A.PEER.MGT	The assumption on all remote trusted IT systems to be under the same management control and operate under security policy constraints compatible with those of the TOE is covered by: ● OE.TRUSTED.IT.SYSTEM requiring that these remote trusted IT systems are under the same management domain as the TOE, and are managed based on the same rules and policies applicable to the TOE.
A.PEER.FUNC	The assumption on all remote trusted IT systems to correctly implement the functionality used by the TSF consistent with the assumptions defined for this functionality is covered by: ● OE.TRUSTED.IT.SYSTEM requiring that the remote trusted IT systems implement the protocols and mechanisms required by the TSF to support the enforcement of the security policy.

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

Table 4: Sufficiency of objectives holding assumptions

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

OSP	Rationale for security objectives
P.ACCOUNTABILITY	The policy to hold users accountable for their security-relevant actions within the TOE is implemented by: • O.AUDITING providing the TOE with audit functionality, • O.MANAGE allowing the management of this function.
P.USER	The policy to match the trust given to a user and the actions the user is given authority to perform is implemented by: • O.MANAGE allowing appropriately-authorized users to manage the TSF, • OE.INFO_PROTECT, which requires that users are trusted to use the protection mechanisms of the TOE to protect their data.
P.CLEARANCE	The policy to limit information flow between protected resources and authorized users based on whether the user's sensitivity label is appropriate for the labeled information is implemented by: • O.LS.CONFIDENTIALITY requiring the TOE to control information flow between entities and resources based on the sensitivity labels of users and resources.
P.LABELED_OUTPUT	The policy to provide the capability to mark printed output with accurate labels based on the sensitivity label of the user causing the output is implemented by: • O.LS.PRINT providing the capability to mark printed output with accurate labels based on the sensitivity label of the user causing the output.

OSP	Rationale for security objectives
P.RESOURCE_LABELS	The policy that resources accessible by subjects and all subjects must have associated labels identifying the sensitivity levels of data contained therein is implemented by: • O.LS.LABEL providing the capability to label all subjects and all objects accessible by subjects, to restrict the information flow based on the sensitivity labels.
P.USER_CLEARANCE	The policy that all users must have a clearance level identifying the maximum sensitivity levels of data they may access is implemented by: • O.LS.CONFIDENTIALITY requiring the TOE to control information flow between entities and resources based on the sensitivity labels of users and resources. • O.LS.LABEL ensuring that objects and subjects can be labeled such that the TOE can restrict information flow based on those labels.

Table 5: Sufficiency of objectives enforcing Organizational Security Policies

5 Extended Components Definition

The [OSPP] defines following extended components:

- FDP_RIP.3: Full residual information protection of subjects, and
- FIA_USB.2: Enhanced user-subject binding.

This security target does not define any additional extended components.

The definition of FCS_RNG was supplied by BSI. Due to the changes to FCS_RNG.1 requested by BSI, the SFR of FCS_RNG.1 referenced in the OSPP needed to be updated as follows.

5.1 Class FCS: Cryptographic support

5.1.1 Random number generator (RNG)

Family behaviour

This family defines quality requirements for the generation of random numbers that are intended to be used for cryptographic purposes.

Component levelling

FCS_RNG.1 is not hierarchical to any other component within the FCS_RNG family.

Management: FCS_RNG.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_RNG.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: There are no actions defined to be auditable.

b) Basic: There are no actions defined to be auditable.

c) Detailed: There are no actions defined to be auditable.

5.1.1.1 FCS_RNG.1 - Random number generation

Hierarchical to: No other components.

Dependencies: No dependencies.

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

- **DRG.2.1: If initialized with a random seed [selection: *using PTRNG of class PTG.2 as random source, using PTRNG of class PTG.3 as random source, using NPTRNG of class NTG.1 as random source, [assignment: other requirements for seeding]*], the internal state of the RNG shall [selection: *have [assignment: amount of entropy], have [assignment: work factor], require [assignment: guess work]*].**
- **DRG.2.2: The RNG provides forward secrecy.**
- **DRG.2.3: The RNG provides backward secrecy.**

FCS_RNG.1.2 **The TSF shall provide random numbers that meet:**

- **DRG.2.4: The RNG initialized with a random seed [assignment: *requirements for seeding*] generates output for which [assignment: *number of strings*] strings of bit length 128 are mutually different with probability [assignment: *probability*].**
- **DRG.2.5: Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A [assignment: *additional test suites*].**

Rationale

The quality of the random number generator is defined using this SFR. The quality metric required in FCS_RNG.1.2 is detailed in the German Scheme AIS 20 and AIS 31 and amended based on discussions with BSI.

6 Security Requirements

6.1 Security Requirements for the Operational Environment

Although CC Version 3.1 does not mandate the use of security requirements for the IT environment, it allows to define the security objectives for the IT environment to the level of detail useful for the understanding and evaluation of a TOE. In the case of z/VM the security functionality of the TOE defined in the following sections depends on the supporting functionality defined in this section. The authors of this Security Target decided to define this functionality using the structure of Security Functional Requirements.

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

- The instructions provided by the underlying processor (named z/Architecture)
- The “CP Assist for Cryptographic Functions” (CPACF). Although this feature is implemented as instructions of the processor and therefore is part of the z/Architecture, it has been decided by the authors of this Security Target to treat them separate from the other instructions. One reason is that some features of CPACF are available on selected processor types only. This is expressed in the SFRs related to CPACF.

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

6.1.1 General security requirements for the abstract machine

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

FDP_ACC.1.1

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

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

FDP_ACF.1.1

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

FDP_ACF.1.2

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

FDP_ACF.1.3

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

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

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

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

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

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

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

6.2 TOE Security Functional Requirements

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

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
z/VM general purpose computing	FAU_GEN.1 Audit data generation		OSPP	No	No	Yes	No
	FAU_GEN.2 User identity association		OSPP	No	No	No	No
	FAU_SAR.1 Audit review		OSPP	No	No	Yes	No
	FAU_SAR.2 Restricted audit review		OSPP	No	No	No	No

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FAU_SAR.3 Selectable audit review		CC Part 2	No	No	Yes	No
	FAU_SEL.1 Selective audit		OSPP	No	No	Yes	No
	FAU_STG.1 Protected audit trail storage		OSPP	No	No	No	Yes
	FAU_STG.3 Action in case of possible audit data loss		OSPP	No	No	Yes	No
	FAU_STG.4 Prevention of audit data loss		OSPP	No	No	Yes	Yes
	FCS_CKM.1(SYM) Cryptographic key generation	FCS_CKM.1	OSPP	Yes	No	Yes	No
	FCS_CKM.1(RSA) Cryptographic key generation	FCS_CKM.1	OSPP	Yes	No	Yes	No
	FCS_CKM.1(DSA) Cryptographic key generation	FCS_CKM.1	OSPP	Yes	No	Yes	Yes
	FCS_CKM.2(NET) Cryptographic key distribution	FCS_CKM.2	OSPP	No	No	Yes	Yes
	FCS_CKM.4 Cryptographic key destruction		OSPP	No	No	No	Yes
	FCS_COP.1(TDES) Cryptographic operation	FCS_COP.1	CC Part 2	Yes	No	Yes	No
	FCS_COP.1(AES) Cryptographic operation	FCS_COP.1	CC Part 2	Yes	No	Yes	No
	FCS_COP.1(SHA1) Cryptographic operation	FCS_COP.1	CC Part 2	Yes	No	Yes	No
	FCS_COP.1(SHA2) Cryptographic operation	FCS_COP.1	CC Part 2	Yes	No	Yes	No
	FCS_COP.1(NET) Cryptographic operation	FCS_COP.1	OSPP	Yes	No	Yes	Yes
	FCS_RNG.1 Random number generator (Class DRG.2)		ECD	No	Yes	Yes	Yes
	FDP_ACC.2(RACF-PSO) RACF Persistent Storage Object Access Control Policy	FDP_ACC.2	OSPP	Yes	Yes	Yes	No
	FDP_ACC.2(RACF-TSO) RACF Transient Storage Object Access Control Policy	FDP_ACC.2	OSPP	Yes	Yes	Yes	No

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FDP_ACC.2(RACF-SYSTEM) RACF System Object Access Control Policy	FDP_ACC.2	CC Part 2	Yes	No	Yes	No
	FDP_ACC.2(CP) Discretionary Access Control Policy by CP	FDP_ACC.2	CC Part 2	Yes	No	Yes	No
	FDP_ACF.1(RACF) Access Control Functions by RACF	FDP_ACF.1	OSPP	Yes	Yes	Yes	No
	FDP_ACF.1(CP) Discretionary Access Control Functions by CP	FDP_ACF.1	CC Part 2	Yes	No	Yes	No
	FDP_IFC.2(NI) Complete information flow control	FDP_IFC.2	OSPP	Yes	No	Yes	No
	FDP_IFF.1(NI) Simple security attributes	FDP_IFF.1	OSPP	Yes	No	Yes	Yes
	FDP_ITC.2(BA) Import of user data with security attributes	FDP_ITC.2	OSPP	Yes	Yes	Yes	No
	FDP_RIP.2 Full residual information protection		OSPP	No	No	No	Yes
	FDP_RIP.3 Full residual information protection of resources		OSPP	No	No	No	Yes
	FIA_AFL.1 Authentication failure handling		OSPP	No	No	Yes	Yes
	FIA_ATD.1(HU) User attribute definition	FIA_ATD.1	OSPP	Yes	No	Yes	No
	FIA_ATD.1(TU) User attribute definition	FIA_ATD.1	OSPP	Yes	No	Yes	No
	FIA_SOS.1 Verification of secrets		OSPP	No	No	No	No
	FIA_UAU.1 Timing of authentication		OSPP	No	No	Yes	No
	FIA_UAU.5 Multiple authentication mechanisms		OSPP	No	Yes	Yes	No
	FIA_UAU.7 Protected authentication feedback		OSPP	No	No	No	No
	FIA_UID.1 Timing of identification		OSPP	No	No	Yes	No
	FIA_USB.2 Enhanced user-subject binding		OSPP	No	Yes	Yes	No
	FMT_MSA.1(DAC) Management of object security attributes	FMT_MSA.1	OSPP	Yes	Yes	Yes	Yes

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FMT_MSA.3(DAC) Static attribute initialisation	FMT_MSA.3	OSPP	Yes	Yes	Yes	No
	FMT_MSA.3(NI) Static attribute initialisation	FMT_MSA.3	OSPP	Yes	No	Yes	Yes
	FMT_MSA.4(DAC) Security attribute value inheritance	FMT_MSA.4	OSPP	No	No	Yes	No
	FMT_MTD.1(AE) Management of TSF data	FMT_MTD.1	OSPP	Yes	No	Yes	No
	FMT_MTD.1(AS) Management of TSF data	FMT_MTD.1	OSPP	Yes	No	Yes	Yes
	FMT_MTD.1(AT) Management of TSF data	FMT_MTD.1	OSPP	Yes	No	Yes	Yes
	FMT_MTD.1(AF) Management of TSF data	FMT_MTD.1	OSPP	Yes	No	Yes	Yes
	FMT_MTD.1(NI) Management of TSF data	FMT_MTD.1	OSPP	Yes	No	Yes	Yes
	FMT_MTD.1(IAT) Management of TSF data	FMT_MTD.1	OSPP	Yes	No	Yes	No
	FMT_MTD.1(IAF) Management of TSF data	FMT_MTD.1	OSPP	Yes	No	Yes	No
	FMT_MTD.1(IAU) Management of TSF data	FMT_MTD.1	OSPP	Yes	No	Yes	No
	FMT_REV.1(OBJ) Revocation of Object Attributes	FMT_REV.1	OSPP	Yes	No	Yes	No
	FMT_REV.1(USR) Revocation of User Attributes	FMT_REV.1	OSPP	Yes	No	Yes	No
	FMT_SMF.1 Specification of management functions		OSPP	No	Yes	Yes	No
	FMT_SMR.1 Security roles		OSPP	No	No	Yes	No
	FPT_STM.1 Reliable time stamps		OSPP	No	No	No	No
	FPT_TDC.1(BA) Inter-TSF basic TSF data consistency	FPT_TDC.1	OSPP	Yes	No	Yes	No
	FPT_TDC.1(TLS) Inter-TSF basic TSF data consistency	FPT_TDC.1	CC Part 2	Yes	No	Yes	No

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FTA_SSL.1 TSF-initiated session locking		OSPP	No	No	Yes	No
	FTA_SSL.2 User-initiated locking		OSPP	No	No	Yes	No
	FTP_ITC.1 Inter-TSF trusted channel		OSPP	No	Yes	Yes	Yes
Virtual machine related functionality	FDP_ETC.2(VIRT) Export of user data with security attributes	FDP_ETC.2	OSPP-VIRT	Yes	Yes	Yes	No
	FDP_IFC.2(VIRT) Complete information flow control	FDP_IFC.2	OSPP-VIRT	Yes	No	Yes	No
	FDP_IFF.1(VIRT) Simple security attributes	FDP_IFF.1	OSPP-VIRT	Yes	Yes	Yes	No
	FDP_ITC.2(VIRT) Import of user data with security attributes	FDP_ITC.2	OSPP-VIRT	Yes	Yes	Yes	No
	FIA_UID.2(VIRT) User identification before any action	FIA_UID.2	OSPP-VIRT	No	No	No	No
	FMT_MSA.1(VIRT-CIFCP) Management of security attributes	FMT_MSA.1	OSPP-VIRT	Yes	No	Yes	Yes
	FMT_MSA.3(VIRT-CIFCP) Static attribute initialisation	FMT_MSA.3	OSPP-VIRT	Yes	No	Yes	No
	FMT_MTD.1(VIRT-COMP) Management of TSF data	FMT_MTD.1	OSPP-VIRT	Yes	No	Yes	No
	FPT_TDC.1(VIRT) Inter-TSF basic TSF data consistency: virtualization	FPT_TDC.1	OSPP-VIRT	Yes	No	Yes	No
Labeled Security	FDP_ETC.2(LS) (Labeled Security Mode only) Export of user data with security attributes	FDP_ETC.2	OSPP-LS	Yes	Yes	Yes	No
	FDP_IFC.2(LS) (Labeled Security Mode only) Complete information flow control	FDP_IFC.2	OSPP-LS	Yes	Yes	Yes	No
	FDP_IFF.2(LS) (Labeled Security Mode only) Hierarchical security attributes	FDP_IFF.2	OSPP-LS	No	Yes	Yes	No
	FDP_ITC.1(LS) (Labeled Security Mode only) Import of user data without security attributes	FDP_ITC.1	OSPP-LS	No	Yes	Yes	No
	FDP_ITC.2(LS) (Labeled Security Mode only) Import of user data with security attributes: labeled security	FDP_ITC.2	OSPP-LS	Yes	Yes	Yes	No

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FIA_ATD.1(LS) User attribute definition: labeled security	FIA_ATD.1	OSPP-LS	Yes	Yes	No	No
	FIA_USB.1(LS) (Labeled Security Mode only) User-subject binding	FIA_USB.1	OSPP-LS	No	Yes	Yes	No
	FMT_MSA.1(LS) (Labeled Security Mode only) Management of object security attributes: labeled security	FMT_MSA.1	OSPP-LS	Yes	Yes	Yes	No
	FMT_MSA.3(LS) (Labeled Security Mode only) Static attribute initialization: labeled security	FMT_MSA.3	OSPP-LS	Yes	Yes	Yes	No
	FPT_TDC.1(LS) (Labeled Security Mode only) Inter-TSF basic TSF data consistency: labeled security	FPT_TDC.1	OSPP-LS	Yes	No	Yes	No
SSI cluster communication	FPT_ITT.1(SSIVM) Basic internal TSF data transfer protection (VM data)	FPT_ITT.1	CC Part 2	Yes	No	No	Yes
	FPT_ITT.1(SSITSF) Basic internal TSF data transfer protection (TSF data)	FPT_ITT.1	CC Part 2	Yes	No	No	Yes
	FPT_TRC.1(SSIVM) Internal TSF consistency (VM data)	FPT_TRC.1	CC Part 2	Yes	Yes	Yes	No
	FPT_TRC.1(SSITSF) Internal TSF consistency (TSF data)	FPT_TRC.1	CC Part 2	Yes	No	Yes	No
	FMT_MTD.1(LGR) Management of TSF data	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes

Table 6: Security functional requirements for the TOE

6.2.1 z/VM general purpose computing

6.2.1.1 Audit data generation (FAU_GEN.1)

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

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

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

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

d) all user authentication attempts;

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

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

g) **no other events** .

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

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

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

i. User identity (if applicable); and

ii. **(in Labeled Security Mode) The sensitivity labels of subjects, objects, or information involved.**

6.2.1.2 User identity association (FAU_GEN.2)

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

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.*

6.2.1.3 Audit review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide **RACF auditors** 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.

6.2.1.4 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.

6.2.1.5 Selectable audit review (FAU_SAR.3)

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

a) user identity;

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

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

d) object type and object name

6.2.1.6 Selective audit (FAU_SEL.1)

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

a) Type of audit event;

b) Subject or user identity;
c) Outcome (success or failure) of the audit event;
d) Named object identity;
e) **subject sensitivity label (Labeled Security Mode only);**
f) **object sensitivity label (Labeled Security Mode only)**

6.2.1.7 Protected audit trail storage (FAU_STG.1)

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

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

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

FAU_STG.3.1 The TSF shall **generate an alarm to selected RACF auditors** if the audit trail exceeds **the capacity of the SMF disks** or if any of the following **no other** is detected that may result in a loss of audit records.

6.2.1.9 Prevention of audit data loss (FAU_STG.4)

FAU_STG.4.1 The TSF shall **prevent audited events, except those taken by the authorised administrator** and **inform the RACF auditor** if the audit trail is full.

6.2.1.10 Cryptographic key generation (FCS_CKM.1(SYM))

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

a) 128 bits,
b) 168 bits,
c) 256 bits,
d) **no other cryptographic key sizes**

that meet the following: **TLS: generation and exchange of session keys as defined in the [TLSv1.1], and [TLSv1.2] standards with the cipher suites defined in FCS_COP.1(NET)**.

6.2.1.11 Cryptographic key generation (FCS_CKM.1(RSA))

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

a) 2048 bits,
b) **3072 bits**
c) **4096 bits**

that meet the following:

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

b) **no other standards**.

6.2.1.12 Cryptographic key generation (FCS_CKM.1(DSA))

FCS_CKM.1.1 The TSF shall generate DSA cryptographic keys in accordance with a specified cryptographic key generation algorithm defined in U.S. NIST FIPS PUB 186-3 appendix B.1 and specified cryptographic key sizes:

a) L=1024, N=160 bits;

that meet the following:

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

b) **no other standards**.

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

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

a) RSA with the size referenced in FCS_CKM.1(RSA)

b) Diffie-Hellman encrypted exchange of pre-master secrets using Diffie-Hellman groups of sizes in multiples of 64 bits between 1024 bits and 2048 bits generated according to [RFC2631]

defined for the TLS protocol by [TLSv1.2], [TLSv1.1].

6.2.1.14 Cryptographic key destruction (FCS_CKM.4)

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

6.2.1.15 Cryptographic operation (FCS_COP.1(TDES))

FCS_COP.1.1 The TSF shall perform **encryption, decryption** in accordance with a specified cryptographic algorithm **Triple DES** and cryptographic key sizes **112 bits or 168 bits** that meet the following: **NIST Special Publication 800-67.**

6.2.1.16 Cryptographic operation (FCS_COP.1(AES))

FCS_COP.1.1 The TSF shall perform **encryption, decryption** in accordance with a specified cryptographic algorithm **AES in CFB, OFB, and CBC-CS modes** and cryptographic key sizes **128 bits, 192 bits or 256 bits** that meet the following: **FIPS 197, November 6, 2001 (AES), NIST Special Publication 800-38A, 2001 Edition (CFB and OFB modes of operation), Addendum to NIST SP 800-38A, October 2010 (CBC-CS mode of operation).**

6.2.1.17 Cryptographic operation (FCS_COP.1(SHA1))

FCS_COP.1.1 The TSF shall perform **hashing** in accordance with a specified cryptographic algorithm **SHA-1** and cryptographic key sizes **not applicable** that meet the following: **FIPS 180-4.**

6.2.1.18 Cryptographic operation (FCS_COP.1(SHA2))

FCS_COP.1.1 The TSF shall perform **hashing** in accordance with a specified cryptographic algorithm **SHA-224, SHA-256, SHA-384, and SHA-512** and cryptographic key sizes **not applicable** that meet the following: **FIPS 180-4.**

6.2.1.19 Cryptographic operation (FCS_COP.1(NET))

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

a) TLS using by the following TLS cipher strings as defined by [TLSv1.1]:

i. TLS_RSA_WITH_3DES_EDE_CBC_SHA
ii. TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA
iii. TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA
iv. TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
v. TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
vi. TLS_RSA_WITH_AES_128_CBC_SHA
vii. TLS_DH_DSS_WITH_AES_128_CBC_SHA
viii. TLS_DH_RSA_WITH_AES_128_CBC_SHA
ix. TLS_DHE_DSS_WITH_AES_128_CBC_SHA
x. TLS_DHE_RSA_WITH_AES_128_CBC_SHA
xi. TLS_RSA_WITH_AES_256_CBC_SHA
xii. TLS_DH_DSS_WITH_AES_256_CBC_SHA
xiii. TLS_DH_RSA_WITH_AES_256_CBC_SHA
xiv. TLS_DHE_DSS_WITH_AES_256_CBC_SHA
xv. TLS_DHE_RSA_WITH_AES_256_CBC_SHA

b) TLS using by the aforementioned and following TLS cipher strings as defined by [TLSv1.2]:

i. TLS_RSA_WITH_AES_128_CBC_SHA256
ii. TLS_RSA_WITH_AES_256_CBC_SHA256
iii. TLS_DH_DSS_WITH_AES_128_CBC_SHA256
iv. TLS_DH_RSA_WITH_AES_128_CBC_SHA256
v. TLS_DHE_DSS_WITH_AES_128_CBC_SHA256
vi. TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
vii. TLS_DH_DSS_WITH_AES_256_CBC_SHA256
viii. TLS_DH_RSA_WITH_AES_256_CBC_SHA256
ix. TLS_DHE_DSS_WITH_AES_256_CBC_SHA256
x. TLS_DHE_RSA_WITH_AES_256_CBC_SHA256

c) RSA peer authentication using the following mechanisms as defined by [TLSv1.2], [TLSv1.1]:

1. Public-key-based authentication of server using RSA-encryption RSAES-PKCS-v1_5 using SHA-1 as defined in [PKCS1_2.1]

2. **Public-key-based authentication of server using RSA-encryption RSAES-PKCS-v1_5 using SHA-256, SHA-384 as defined in [PKCS1_2.1]**
3. **RSA signature verification for server authentication RSASSA-PKCS-v1_5 using SHA-1 as defined in [PKCS1_2.1]**
4. **RSA signature verification for server authentication RSASSA-PKCS-v1_5 using SHA-256, SHA-384 as defined in [PKCS1_2.1]**
5. **DSA signature verification for server authentication using SHA-1**

6.2.1.20 Random number generator (Class DRG.2) (FCS_RNG.1)

FCS_RNG.1.1 The TSF shall provide a deterministic random number generator that implements:

a) DRG.2.1: If initialized with a random seed using **CPU jitter as seed source** , the internal state of the RNG shall **have a minimum entropy of 48 bits**.

b) DRG.2.2: The RNG provides forward secrecy.

c) DRG.2.3: The RNG provides backward secrecy.

FCS_RNG.1.2 The TSF shall provide random numbers that meet

a) DRG.2.4: The RNG ~~initialized~~ *seeded* with a random seed *at initialization time of the SSL server during boot* **holding 96 bits of entropy** generates output for which **2**25** strings of bit length 128 are mutually different with probability **of greater than 1-2**-10** .

b) DRG.2.5: Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A.

Application note: *Quote from the FIPS 140-2 security policy covering the DRNG seeding: "... the DRNG engine seeded with 20 bytes of true random data. This true random number generator extracts entropy from time measurement jitter (minute variations of clock edges). The internal TRNG engine feeds entropy on demand into the DRNG; the TRNG itself maintains a running pool of samples, and provides seed if the pool passes basic entropy content checks.*"

6.2.1.21 RACF Persistent Storage Object Access Control Policy (FDP_ACC.2(RACF-PSO))

FDP_ACC.2.1 The TSF shall enforce the *RACF* Persistent Storage Object Access Control Policy on

a) **Subjects: virtual machines acting on behalf of a human user or technical entity providing a virtual machine environment**;

b) Objects:

i. Persistent Storage Objects of the following type

- **Minidisks**
- **Real DASD volumes**
- **Restricted DCSS**
- **Restricted NSS**
- **Spool files**
- **POSIX information database**

- **RACF database**

ii. **no other storage objects**

and all operations among subjects and objects covered by the SFP.

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

Application Note: *This SFR is hierarchical to the PP SFR of FDP_ACC.1(PSO) which satisfies the strict conformance claim.*

Application Note: *The TOE provides virtual machines for other operating systems as well as other components of the TOE. The objects maintained by the TOE are therefore accessible by virtual machines hosting operating systems as well as virtual machines implementing other aspects of the TOE. Therefore, the ST author decided to merge the SFR FDP_ACC.2(VIRT) defined by the OSPP extended package on virtualization into the access control SFRs specified by the OSPP base. To still satisfy the strict compliance with the PP, the ST author uses the hierarchical SFR of FDP_ACC.2 to implement the OSPP-base access control policies as well as the virtualization access control policy.*
This SFR defines the Persistent Storage Objects covering the SFR of FDP_ACC.2(VIRT) required by the PP. The defined subjects are the compartments defined with the PP SFR. Together with FDP_ACC.2(RACF-PSO) defined below, the PP SFR FDP_ACC.2(VIRT) is also covered, satisfying the strict compliance.

6.2.1.22 RACF Transient Storage Object Access Control Policy (FDP_ACC.2(RACF-TSO))

FDP_ACC.2.1 The TSF shall enforce the *RACF* Transient Storage Object Access Control Policy on

a) **Subjects: virtual machines acting on behalf of a human user or technical entity providing a virtual machine environment**;

b) Objects:

i. Transient Storage Objects of the following type

- **Guest LANs**
- **Virtual Switches**
- **NJE network nodes**
- **Virtual point-to-point communication paths (IUCV, VMCF, APPC, virtual CTC, MSG, WNG, MSGNOH, SMSG)**
- **CP real memory**

ii. **no other storage objects**

and all operations among subjects and objects covered by the SFP.

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

Application Note: *This SFR is hierarchical to the PP SFR of FDP_ACC.1(TSO) which satisfies the strict conformance claim.*

Application Note: *The TOE provides virtual machines for other operating systems as well as other components of the TOE. The objects maintained by the TOE are therefore accessible by virtual machines hosting operating systems as well as virtual machines implementing other aspects of*

the TOE. Therefore, the ST author decided to merge the SFR FDP_ACC.2(VIRT) defined by the OSPP extended package on virtualization into the access control SFRs specified by the OSPP base. To still satisfy the strict compliance with the PP, the ST author uses the hierarchical SFR of FDP_ACC.2 to implement the OSPP-base access control policies as well as the virtualization access control policy.
This SFR defines the Transient Storage Objects covering the SFR of FDP_ACC.2(VIRT) required by the PP. The defined subjects are the compartments defined with the PP SFR. Together with FDP_ACC.2(RACF-TSO) defined below, the PP SFR FDP_ACC.2(VIRT) is also covered, satisfying the strict compliance.

6.2.1.23 RACF System Object Access Control Policy (FDP_ACC.2(RACF-SYSTEM))

FDP_ACC.2.1 The TSF shall enforce the **RACF System Object Access Control Policy** on

- **a) Subjects: virtual machines acting on behalf of a human user or technical entity providing a virtual machine environment;**
- **b) Objects:**
  - **i. User authentication service**
  - **ii. RACROUTE macro**
  - **iii. Alternate (surrogate) user IDs**
  - **iv. Virtual machine console**
  - **v. System access**
  - **vi. Objects accessible through the following interfaces:**
    - **i. CP commands listed in table 5, Appendix A [SCG]**
    - **ii. DIAGNOSE codes listed in table 6, Appendix A [SCG]**
    - **iii. System functions listed in table 7, Appendix A [SCG]**

and all operations among subjects and objects covered by the SFP.

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

6.2.1.24 Discretionary Access Control Policy by CP (FDP_ACC.2(CP))

FDP_ACC.2.1 The TSF shall enforce the **CP Access Control Policy** on

- **a) Subjects: virtual machines acting on behalf of a human user or technical entity providing a virtual machine environment;**
- **b) Objects:**
  - **i. CP commands belonging to one or more privilege classes other than privilege class ANY**
  - **ii. DIAGNOSE code belonging to one or more privilege classes other than privilege class ANY or whose can be access restricted with a system directory statement**
  - **iii. Following processor instruction causing the SIE instruction to terminate:**
    - **i. IUCV processor instruction (0xB2F0)**

and all operations among subjects and objects covered by the SFP.

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

6.2.1.25 Access Control Functions by RACF (FDP_ACF.1(RACF))

FDP_ACF.1.1 The TSF shall enforce the *RACF* Persistent Storage Object Access Control Policy *, RACF Transient Storage Object Access Control Policy, RACF System Object 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:**
  1. **an access control list capable of defining the access rights read, update, execute, alter, control, and none for individual users and groups**
  2. **a default access right (defined by the UACC attribute in the resource profile) for users who are not addressed in the access control list.**

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 compartment or, if a) is not true,**
- **b) if the requested type of access is allowed by an access authority for group the compartment 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 compartment 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 authorise access of subjects to objects based on the following additional rules:

- **a) Assignment of 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)**
- **b) 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 additional rules:

- **a) Assignment of the REVOKE attribute to users**

b) By adding resource profiles to the global access table with a UACC of NONE, this resource is always denied access to.

Application Note: *The OSPP base specifies FDP_ACF.1(PSO) and FDP_ACF.1(TSO). As this TOE implements one rule set for both object types, the ST author choose to define one instance of FDP_ACF.1 covering both. As outlined in this SFR, it applies to both, PSO and TSO. It would always be possible to specify separate FDP_ACF.1 iterations individually for TSO and PSO, but they would specify identical rule sets. Therefore, the ST is still considered to be strictly conformant to the OSPP.*

Application Note: *As outlined for FDP_ACC.2(RACF-PSO) and FDP_ACC.2(RACF-TSO), both policies also cover all subjects and objects required by FDP_ACC.2(VIRT) out of the OSPP extended package of virtualization. Therefore, this SFR also covers FDP_ACF.1(VIRT), satisfying the strict compliance of this ST with the PP.*

Application Note: *FDP_ACC.2(RACF-PSO), FDP_ACC.2(RACF-TSO), and FDP_ACC.2(RACF-SYSTEM) apply to FDP_ACF.1(RACF) which is implemented by the trusted application of RACF.*

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

FDP_ACF.1.1 The TSF shall enforce the **CP 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:

i) 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 authorise access of subjects to objects based on the following additional rules: **none**.

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

Application note: *FDP_ACC.2(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 partially enforced on identical objects: the CP commands and DIAGNOSE codes listed in FDP_ACC.2(RACF-SYSTEM). 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.*

6.2.1.27 Complete information flow control (FDP_IFC.2(NI))

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

a) Subjects:

i. unauthenticated external IT entities that send and receive information mediated by the TOE;

ii. **no other subjects** that send and receive information mediated by the TOE;

b) Information:

i. Network data routed through the TOE;

ii. **no other information**;

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

FDP_IFC.2.2 The TSF shall ensure that all operations that cause any information in the TOE to flow to and from any subject in the TOE are covered by an information flow control SFP.

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

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

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

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

b) IEEE 802.1Q VLAN tag information security attributes:

i. VLAN tag.

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: **If the z/VM TCP/IP stack application configured for IP security allows an IP packet to be sent or to be received by a subject, the packet flow is allowed according to the protocol stack's behavior.**

FDP_IFF.1.3 The TSF shall enforce the following rules:
Identification of network data using one or more of the following concepts:

a) Information security attribute matching;

b) **No other matching concepts;**

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

a) Discard the network data **without any further processing, with sending a notification to the sender**;

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

c) **no other actions**.

FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: **None**.

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: **If the network data is not matched by the rule set, the TSF shall discard the data**.

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

FDP_ITC.2.1 The TSF shall enforce the *RACF* Persistent Storage Access Control Policy, *RACF* Transient Storage Access Control Policy, Network Information Flow Control Policy, **no other access control SFP(s) and/or information flow control SFP(s)** when importing user data, controlled under the SFP, from outside of the TOE.

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

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

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

FDP_ITC.2.5 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: **no additional importation control rules.**

6.2.1.30 Full residual information protection (FDP_RIP.2)

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

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

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

6.2.1.32 Authentication failure handling (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when an administrator-configurable number of unsuccessful authentication attempts for the authentication method **password-based authentication** occur related to **consecutive unsuccessful authentication attempts**.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been **surpassed** , the TSF shall:

a) For all accounts with the SPECIAL attribute, the operator is prompted whether the user status should be set to REVOKE when the limit is reached.

b) For all other accounts, the TSF shall set the user status to REVOKE.

c) For all disabled accounts, any response to an authentication attempt given to the user shall not be based on the result of that authentication attempt.

6.2.1.33 User attribute definition (FIA_ATD.1(HU))

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

a) User identifier;

b) Group memberships;

c) User password;

d) Software token verification data;

e) Security roles;

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.**

6.2.1.34 User attribute definition (FIA_ATD.1(TU))

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

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

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

c) **VLAN tag.**

6.2.1.35 Verification of secrets (FIA_SOS.1)

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet the following quality metric: the probability that a secret can be obtained by an attacker during the lifetime of the secret is less than 2^-20.

Application Note: *The CC guide contains configuration suggestions for the password quality mechanism that covers the above mentioned probability. These configuration suggestions assume the worst-case scenario when attacking these settings.*

6.2.1.36 Timing of authentication (FIA_UAU.1)

FIA_UAU.1.1 The TSF shall allow

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

b) **providing credentials and (in labeled security mode) selection of security label**

c) **use of the LOGON and LOGOFF command**

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

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

6.2.1.37 Multiple authentication mechanisms (FIA_UAU.5)

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

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

b) Authentication based on software token verification data;

c) **no other authentication mechanisms**

to support user authentication.

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

a) Authentication based on username and password */passphrase* is performed for TOE-originated requests and credentials stored by the TSF;

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

c) **no other rules.**

Application Note:

The TLS channel can be configured to perform certificate-based authentication to using bi-directional certificate validation. The SSL server establishes access to the CP console. Therefore, with a TLS certificate authentication the user has to provide his password/passphrase to authenticate with the CP console.

6.2.1.38 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.

6.2.1.39 Timing of identification (FIA_UID.1)

FIA_UID.1.1 The TSF shall allow

a) providing credentials and (in labeled security mode) selection of security label

b) use of the LOGON and LOGOFF command

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

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

6.2.1.40 Enhanced user-subject binding (FIA_USB.2)

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

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

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

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

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

e) Active roles;

f) Active groups;

g) **RACF attributes/roles SPECIAL, group-SPECIAL, AUDITOR, group-AUDITOR, CLAUTH and OPERATIONS associated with the user or any of the user's groups**.

FIA_USB.2.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: **a started virtual machine executes with the user ID of the logged in user it has been defined for**.

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

FIA_USB.2.4 The TSF shall enforce the following rules for the assignment of subject security attributes not derived from user security attributes when a subject is created: **no rules**.

6.2.1.41 Management of object security attributes (FMT_MSA.1(DAC))

FMT_MSA.1.1 The TSF shall enforce the *RACF* Persistent Storage Object Access Control Policy *, RACF Transient Storage Object Access Control Policy, RACF System Object Access Control Policy, CP Access Control Policy* to restrict the ability to modify **change_default, query** the security attributes ~~of the~~*associated with the named* objects covered by the SFP to the owner of the *resource profile of the named* object and **users with**

- **the SPECIAL attribute or the appropriate group-SPECIAL attribute,**
- **the CLAUTH attribute for the class the resource is assigned to,**
- **and users who have ALTER authority to the object.**

Application note: *Since the DAC policies contain persistent as well as transient and other objects and thus cover both FDP_ACC.2(PSO) and FDP_ACC.2(TSO) claimed in the* [OSPP]*, and in addition both DAC mechanism implemented in RACF and CP are enforced on identical objects: the CP commands and DIAGNOSE codes listed in FDP_ACC.2(CP); there is only one management SFR for DAC required.*

Application Note: *This SFR also covers FMT_MSA.1(VIRT-CACP) specified by the OSPP extended package on virtualization.*

6.2.1.42 Static attribute initialisation (FMT_MSA.3(DAC))

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

FMT_MSA.3.2 The TSF shall allow the **authorized administrators, or the owner (non-Labeled Security 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: *Since the DAC policies contain persistent as well as transient and other objects and thus cover both FDP_ACC.2(PSO) and FDP_ACC.2(TSO) claimed in the* [OSPP]*, and in addition both DAC mechanism implemented in RACF and CP are enforced on identical objects: the CP commands and DIAGNOSE codes listed in FDP_ACC.2(CP); there is only one management SFR for DAC required.*

Application Note: *This SFR also covers FMT_MSA.3(VIRT-CACP) specified by the OSPP extended package on virtualization.*

6.2.1.43 Static attribute initialisation (FMT_MSA.3(NI))

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

FMT_MSA.3.2 The TSF shall allow the **authorized administrator** to specify alternative initial values to override the default values when an object or information is created.

6.2.1.44 Security attribute value inheritance (FMT_MSA.4(DAC))

FMT_MSA.4.1 The TSF shall use the following rules to set the value of security attributes for Persistent Storage Objects: **no rules for setting the values of security attributes**.

Application Note: *This SFR is applicable when a subject can create new objects. However, this TOE does not allow subjects to create new objects.*

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

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

a) users with the AUDITOR attribute or the appropriate group-AUDITOR attribute

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

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

6.2.1.46 Management of TSF data (FMT_MTD.1(AS))

FMT_MTD.1.1 The TSF shall restrict the ability to clear, **create, delete** the audit storage to **users with the AUDITOR attribute or the appropriate group-AUDITOR attribute**.

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

6.2.1.47 Management of TSF data (FMT_MTD.1(AT))

FMT_MTD.1.1 The TSF shall restrict the ability to modify the

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

b) action when the threshold is reached

to **users with the AUDITOR attribute or the appropriate group-AUDITOR attribute**.

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

Application Note: *As the threshold of the audit trail is always set to a value allowed within the TOE, it can only be modified. Therefore, the selection in FMT_MTD.1.1 is not applicable.*

6.2.1.48 Management of TSF data (FMT_MTD.1(AF))

FMT_MTD.1.1 The TSF shall restrict the ability to modify the actions to be taken in case of audit storage failure to **the RACFVM user ID** .

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

Application Note: *As the list of actions is always defined within the TOE, it can only be modified. Therefore, the selection in FMT_MTD.1.1 is not applicable.*

Application Note: *The RACFVM user ID is the virtual machine owner hosting the RACF instance. This ID is privileged in the sense as CP is configured to trust this ID to provide the authentication backend.*

6.2.1.49 Management of TSF data (FMT_MTD.1(NI))

FMT_MTD.1.1 The TSF shall restrict the ability to query, modify, delete, **no other operations** the security attributes for the rules governing the

a) identification of network data;

b) actions performed on the identified network data

to **authorized administrators** .

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

6.2.1.50 Management of TSF data (FMT_MTD.1(IAT))

FMT_MTD.1.1 The TSF shall restrict the ability to modify the threshold for unsuccessful authentication attempts to **authorized administrators** .

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

6.2.1.51 Management of TSF data (FMT_MTD.1(IAF))

FMT_MTD.1.1 The TSF shall restrict the ability to re-enable the authentication to the account subject to authentication failure to **authorized administrators** .

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

6.2.1.52 Management of TSF data (FMT_MTD.1(IAU))

FMT_MTD.1.1 The TSF shall restrict the ability to initialize, modify, delete the user security attributes to

a) the authorized administrator

b) users authorized to modify their own authentication data.

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

6.2.1.53 Revocation of Object Attributes (FMT_REV.1(OBJ))

FMT_REV.1.1 The TSF shall restrict the ability to revoke object security attributes defined by SFPs associated with the corresponding object under the control of the TSF to

a) DAC permissions: owner of the resource profile of the named object and authorized administrators;

b) Other security attributes: authorized administrators.

FMT_REV.1.2 The TSF shall enforce the following rules:

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

b) **Labeled Security Mode only: the rules of the Mandatory Access Control Policy are enforced on all future operations.**

6.2.1.54 Revocation of User Attributes (FMT_REV.1(USR))

FMT_REV.1.1 The TSF shall restrict the ability to revoke user security attributes defined by the SFP associated with the corresponding user under the control of the TSF to **the authorized administrators.**

FMT_REV.1.2 The TSF shall enforce the following rules:

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

b) **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** .

6.2.1.55 Specification of management functions (FMT_SMF.1)

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

a) Management of auditing;

b) Management of cryptographic network protocols;

c) Management of *RACF* Persistent Storage Object Access Control Policy;

d) Management of *RACF* Transient Storage Object Access Control Policy;

e) Management of Network Information Flow Control Policy;

f) Management of identification and authentication policy;

g) Management of user security attributes;

h) **Management of RACF System Object Access Control Policy;**

i) **Management of CP Access Control Policy**.

6.2.1.56 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles:

a) User role with the following rights:

i. Users are authorized to modify their own user password;

ii. Users are authorized to modify the access control permissions for the named objects they own;

iii. **no other rights**;

b) **users authorized by the RACF Persistent Storage Access Control Policy or RACF Transient Storage Access Control Policy to modify object security attributes;**

c) **in Labeled Security Mode: users authorized by the Mandatory Access Control Policy to modify object security attributes;**

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

e) **authorized administrators (users with the SPECIAL or group-SPECIAL attribute in their profile);**

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

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.

6.2.1.57 Reliable time stamps (FPT_STM.1)

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

6.2.1.58 Inter-TSF basic TSF data consistency (FPT_TDC.1(BA))

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret **the following data types:**

a) Packet filter: VLAN tag out of the IP header;

when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use **the following interpretation rules:**

a) Packet filter: VLAN tag specification provided in IEEE 802.1Q

when interpreting the TSF data from another trusted IT product.

6.2.1.59 Inter-TSF basic TSF data consistency (FPT_TDC.1(TLS))

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret **the following TSF data types:**

- **X.509 certificate.**

when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use **the following interpretation rules:**

a) X.509 certificate: The validity of the X.509-certificate must be verified along the certificate chain up to the trusted CA certificate

when interpreting the TSF data from another trusted IT product.

6.2.1.60 TSF-initiated session locking (FTA_SSL.1)

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

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

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

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

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

b) **no other events.**

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

6.2.1.61 User-initiated locking (FTA_SSL.2)

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

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

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

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

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

b) **no other events.**

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

6.2.1.62 Inter-TSF trusted channel (FTP_ITC.1)

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

a) Cryptographically-protected communication channel using **the the network protocols specified in FCS_COP.1(NET).**

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

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

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

6.2.2 Virtual machine related functionality

6.2.2.1 Export of user data with security attributes (FDP_ETC.2(VIRT))

FDP_ETC.2.1 The TSF shall enforce the ~~Compartment Access Control Policy~~*RACF Persistent Storage Object Access Control Policy, RACF Transient Storage Object Access Control Policy,* and Compartment Information Flow Control Policy when exporting user data, controlled under the SFP(s), outside of the TOE.

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

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

FDP_ETC.2.4 The TSF shall enforce the following rules when user data is exported from the TOE: **The host system ensures that the source IP-address is equal to the IP-address assigned to the virtual machine initiating the data export using the network**.

6.2.2.2 Complete information flow control (FDP_IFC.2(VIRT))

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

a) Subjects:
   i. Compartments;
   ii. External entities;
   iii. **No other entities**;

b) Information:
   i. User data belonging to compartments;
   ii. User data belonging to subjects outside of compartments;
   iii. TSF data;
   iv. **No additional information**

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

Application Note: *Compartments are the virtual machines mentioned in other SFRs.*

FDP_IFC.2.2 The TSF shall ensure that all operations that cause any information in the TOE to flow to and from any subject in the TOE are covered by an information flow control SFP.

6.2.2.3 Simple security attributes (FDP_IFF.1(VIRT))

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

a) Subject security attributes:
   i. **virtual machine ID**;
   ii. **no additional subject security attributes**;

b) Information security attributes:

i. **virtual machine ID**;
ii. **No TSF data security attributes**;
iii. **No additional information security attributes**.

Application Note: *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.*

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: **No operations is allowed other than via the TSF-provided inter-VM communication channels specified by FDP_ACC.2(RACF-PSO) and FDP_ACC.2(RACF-TSO)**.

FDP_IFF.1.3 The TSF shall enforce ~~the~~**no additional information flow control SFP rules**.

FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: **None**.

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: **None**.

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

FDP_ITC.2.1 The TSF shall enforce the ~~Compartment Access Control Policy~~*RACF Persistent Storage Object Access Control Policy, RACF Transient Storage Object Access Control Policy,* and Compartment Information Flow Control Policy when importing user data, controlled under the SFP, from outside of the TOE.

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

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

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

FDP_ITC.2.5 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: **no additional importation control rules**.

6.2.2.5 User identification before any action (FIA_UID.2(VIRT))

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

6.2.2.6 Management of security attributes (FMT_MSA.1(VIRT-CIFCP))

FMT_MSA.1.1 The TSF shall enforce the Compartment Information Flow Control Policy to restrict the ability to change_default, query, modify **, delete** the security attributes of the subjects and information covered by the SFP, **no additional security attributes** to **the authorized administrator**.

Application Note: *This SFR covers the definition of a virtual machine and all its resources not covered by the persistent or transient storage object access control policies.*

6.2.2.7 Static attribute initialisation (FMT_MSA.3(VIRT-CIFCP))

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

FMT_MSA.3.2 The TSF shall allow the **authorized administrator** to specify alternative initial values to override the default values when an object or information is created.

6.2.2.8 Management of TSF data (FMT_MTD.1(VIRT-COMP))

FMT_MTD.1.1 The TSF shall restrict the ability to initialize, modify, delete the compartment security attributes to **authorized administrators**.

Application Note: *This SFR applies to FIA_UID.2(VIRT).*

6.2.2.9 Inter-TSF basic TSF data consistency: virtualization (FPT_TDC.1(VIRT))

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret access control and information flow control-related security attributes, **and no additional TSF data types** when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use **the IP addresses part of the network packet transmitted by the TSF as specified in RFC 791** when interpreting the TSF data from another trusted IT product.

6.2.3 Labeled Security

The following SFRs apply only when the TOE is installed and operated with labeled security enabled.

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

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

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

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

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

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

b) When the data is exported to a *printer* device the security attributes shall be exported with the data using **the association of a printer to a single sensitivity label that can only be changed by the authorized administrator**.

c) **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.**

d) **Devices used to export data with security attributes shall completely and unambiguously associate the security attributes with the corresponding data.**

6.2.3.2 Complete information flow control (FDP_IFC.2(LS) (Labeled Security Mode only))

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

a) Subjects: **virtual machines acting on behalf of a human user or technical entity providing a virtual machine environment**;

b) 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 (surrogate) user IDs**
- **RACF database**
- **Virtual machine console**
- **System access**
- **Objects accessible through the following interfaces:**
  - **CP commands listed in table 5, Appendix A [SCG]**
  - **DIAGNOSE codes listed in table 6, Appendix A [SCG]**
  - **System functions listed in table 7, Appendix A [SCG]**

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

FDP_IFC.2.2 The TSF shall ensure that all operations that cause any information in the TOE to flow among untrusted subjects and named objects in the TOE are covered by the *Mandatory Access Control Policy*~~Multilevel Confidentiality Information Flow Control Policy~~.

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

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

a) Subject security attributes:

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

ii. **none**;

b) Object security attributes:

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

ii. **none**.

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

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

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

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

Application Note: *If the label of the object is greater than the label of the subject, this is a blind append (i.e., write does not imply a read).*

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 explicitly authorise an information flow based on the following rules: **none.**

FDP_IFF.2.5 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.2.6 The TSF shall enforce the following relationships for any two valid information flow control security attributes:

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

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

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

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

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

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

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

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

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

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

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

b) **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**.

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

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

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

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

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

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

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

b) **Security labels consist of the following:**

i. a hierarchical level; and

ii. a set of non-hierarchical categories.

6.2.3.6 User attribute definition: labeled security (FIA_ATD.1(LS))

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

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

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

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

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

a) User sensitivity label that is used to enforce the *Mandatory Access Control Policy*~~Multilevel Confidentiality Information Flow Control Policy~~*which consists of the following:*

i A hierarchical level; and

ii A set of non-hierarchical categories.

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users:

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

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

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

FMT_MSA.1.1 The TSF shall enforce the *Mandatory Access Control Policy*~~Multilevel Confidentiality Information Flow Control Policy~~ to restrict the ability to modify the label-related object security attributes to **users that satisfy the following rules: users with the SPECIAL attribute or the appropriate group-SPECIAL attribute**.

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

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

FMT_MSA.3.2 The TSF shall allow the **authorized administrator** to specify alternative initial values to override the default values when an object or information is created.

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

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

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

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

6.2.4 SSI cluster communication

6.2.4.1 Basic internal TSF data transfer protection (VM data) (FPT_ITT.1(SSIVM))

FPT_ITT.1.1 The TSF shall protect TSF data from **disclosure, modification** when it is transmitted between separate parts of the TOE.

6.2.4.2 Basic internal TSF data transfer protection (TSF data) (FPT_ITT.1(SSITSF))

FPT_ITT.1.1 The TSF shall protect TSF data from **disclosure, modification** when it is transmitted between separate parts of the TOE.

6.2.4.3 Internal TSF consistency (VM data) (FPT_TRC.1(SSIVM))

FPT_TRC.1.1 The TSF shall ensure that TSF data *encapsulating virtual machine data for live guest migration* is consistent when replicated between parts of the TOE.

FPT_TRC.1.2 When parts of the TOE containing replicated TSF data are disconnected, the TSF shall ensure the consistency of the replicated TSF data upon reconnection before processing any requests for **all operations performed on the virtual machine that is a target for the live guest migration.**

Application note: *The author notes that the virtual machine data subject to transferral to the target virtual machine is user data. However, that user data becomes TSF data when the live guest migration functionality transfers this data to the target virtual machine as the virtual machine user cannot operate with or on this data.*

6.2.4.4 Internal TSF consistency (TSF data) (FPT_TRC.1(SSITSF))

FPT_TRC.1.1 The TSF shall ensure that TSF data is consistent when replicated between parts of the TOE.

FPT_TRC.1.2 When parts of the TOE containing replicated TSF data are disconnected, the TSF shall ensure the consistency of the replicated TSF data upon reconnection before processing any requests for **cross-cluster functions and negotiations for shared resources except existing accesses to shared resources.**

Application note: *This SFR applies to the shared SSI cluster system configuration as well as the PDR.*

6.2.4.5 Management of TSF data (FMT_MTD.1(LGR))

FMT_MTD.1.1 The TSF shall restrict the ability to **initiate, test, cancel, modify the quiesce time of, modify the relocation time of** the **live guest relocation operation of virtual machines** to **authorized administrators** .

6.3 Security Functional Requirements Rationale

6.3.1 Security Requirements Coverage

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

Security functional requirements	Objectives
FAU_GEN.1	O.AUDITING
FAU_GEN.2	O.AUDITING
FAU_SAR.1	O.AUDITING
FAU_SAR.2	O.AUDITING
FAU_SAR.3	O.AUDITING
FAU_SEL.1	O.AUDITING
FAU_STG.1	O.AUDITING
FAU_STG.3	O.AUDITING
FAU_STG.4	O.AUDITING
FCS_CKM.1(SYM)	O.CRYPTO.NET
FCS_CKM.1(RSA)	O.CRYPTO.NET
FCS_CKM.1(DSA)	O.CRYPTO.NET
FCS_CKM.2(NET)	O.CRYPTO.NET
FCS_CKM.4	O.CRYPTO.NET
FCS_COP.1(TDES)	O.CRYPTO.NET
FCS_COP.1(AES)	O.CRYPTO.NET
FCS_COP.1(SHA1)	O.CRYPTO.NET

Security functional requirements	Objectives
FCS_COP.1(SHA2)	O.CRYPTO.NET
FCS_COP.1(NET)	O.CRYPTO.NET
FCS_RNG.1	O.CRYPTO.NET
FDP_ACC.2(RACF-PSO)	O.COMP.RESOURCE_ACCESS, O.DISCRETIONARY.ACCESS
FDP_ACC.2(RACF-TSO)	O.COMP.RESOURCE_ACCESS, O.DISCRETIONARY.ACCESS, O.SUBJECT.COM
FDP_ACC.2(RACF-SYSTEM)	O.DISCRETIONARY.ACCESS
FDP_ACC.2(CP)	O.DISCRETIONARY.ACCESS
FDP_ACF.1(RACF)	O.COMP.RESOURCE_ACCESS, O.DISCRETIONARY.ACCESS, O.SUBJECT.COM
FDP_ACF.1(CP)	O.DISCRETIONARY.ACCESS
FDP_IFC.2(NI)	O.NETWORK.FLOW
FDP_IFF.1(NI)	O.NETWORK.FLOW
FDP_ITC.2(BA)	O.DISCRETIONARY.ACCESS, O.NETWORK.FLOW, O.SUBJECT.COM
FDP_RIP.2	O.AUDITING, O.CRYPTO.NET, O.DISCRETIONARY.ACCESS, O.I&A, O.NETWORK.FLOW, O.SUBJECT.COM
FDP_RIP.3	O.AUDITING, O.CRYPTO.NET, O.DISCRETIONARY.ACCESS, O.I&A, O.NETWORK.FLOW, O.SUBJECT.COM
FIA_AFL.1	O.I&A
FIA_ATD.1(HU)	O.I&A
FIA_ATD.1(TU)	O.NETWORK.FLOW
FIA_SOS.1	O.I&A
FIA_UAU.1	O.I&A
FIA_UAU.5	O.I&A

Security functional requirements	Objectives
FIA_UAU.7	O.I&A
FIA_UID.1	O.I&A, O.NETWORK.FLOW
FIA_USB.2	O.I&A
FMT_MSA.1(DAC)	O.COMP.RESOURCE_ACCESS, O.MANAGE
FMT_MSA.3(DAC)	O.COMP.RESOURCE_ACCESS, O.MANAGE
FMT_MSA.3(NI)	O.MANAGE
FMT_MSA.4(DAC)	O.MANAGE
FMT_MTD.1(AE)	O.MANAGE
FMT_MTD.1(AS)	O.MANAGE
FMT_MTD.1(AT)	O.MANAGE
FMT_MTD.1(AF)	O.MANAGE
FMT_MTD.1(NI)	O.MANAGE
FMT_MTD.1(IAT)	O.MANAGE
FMT_MTD.1(IAF)	O.MANAGE
FMT_MTD.1(IAU)	O.MANAGE
FMT_REV.1(OBJ)	O.MANAGE
FMT_REV.1(USR)	O.MANAGE
FMT_SMF.1	O.MANAGE
FMT_SMR.1	O.MANAGE
FPT_STM.1	O.AUDITING
FPT_TDC.1(BA)	O.DISCRETIONARY.ACCESS, O.NETWORK.FLOW, O.SUBJECT.COM
FPT_TDC.1(TLS)	O.I&A
FTA_SSL.1	O.I&A
FTA_SSL.2	O.I&A
FTP_ITC.1	O.TRUSTED_CHANNEL
FDP_ETC.2(VIRT)	O.COMP.INFO_FLOW_CTRL, O.COMP.RESOURCE_ACCESS

Security functional requirements	Objectives
FDP_IFC.2(VIRT)	O.COMP.INFO_FLOW_CTRL
FDP_IFF.1(VIRT)	O.COMP.INFO_FLOW_CTRL
FDP_ITC.2(VIRT)	O.COMP.INFO_FLOW_CTRL, O.COMP.RESOURCE_ACCESS
FIA_UID.2(VIRT)	O.COMP.IDENT
FMT_MSA.1(VIRT-CIFCP)	O.COMP.INFO_FLOW_CTRL
FMT_MSA.3(VIRT-CIFCP)	O.COMP.INFO_FLOW_CTRL
FMT_MTD.1(VIRT-COMP)	O.COMP.INFO_FLOW_CTRL, O.COMP.RESOURCE_ACCESS
FPT_TDC.1(VIRT)	O.COMP.INFO_FLOW_CTRL, O.COMP.RESOURCE_ACCESS
FDP_ETC.2(LS) (Labeled Security Mode only)	O.LS.CONFIDENTIALITY, O.LS.PRINT
FDP_IFC.2(LS) (Labeled Security Mode only)	O.LS.CONFIDENTIALITY
FDP_IFF.2(LS) (Labeled Security Mode only)	O.LS.CONFIDENTIALITY
FDP_ITC.1(LS) (Labeled Security Mode only)	O.LS.CONFIDENTIALITY, O.LS.LABEL
FDP_ITC.2(LS) (Labeled Security Mode only)	O.LS.CONFIDENTIALITY, O.LS.LABEL
FIA_ATD.1(LS)	O.LS.LABEL
FIA_USB.1(LS) (Labeled Security Mode only)	O.LS.LABEL
FMT_MSA.1(LS) (Labeled Security Mode only)	O.LS.LABEL
FMT_MSA.3(LS) (Labeled Security Mode only)	O.LS.LABEL
FPT_TDC.1(LS) (Labeled Security Mode only)	O.LS.CONFIDENTIALITY, O.LS.LABEL
FPT_ITT.1(SSIVM)	O.SSI.VMDATAPROTECTION
FPT_ITT.1(SSITSF)	O.SSI.TSFDATAPROTECTION
FPT_TRC.1(SSIVM)	O.SSI.VMDATAPROTECTION
FPT_TRC.1(SSITSF)	O.SSI.TSFDATAPROTECTION
FMT_MTD.1(LGR)	O.SSI.LGRMGT

Table 7: Mapping of security functional requirements to security objectives

6.3.2 Security Requirements Sufficiency

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

Security objectives	Rationale
O.AUDITING	The events to be audited are defined in FAU_GEN.1 and are associated with the identity of the user that caused the event (FAU_GEN.2). Authorized users are provided the capability to read the audit records (FAU_SAR.1), while all other users are denied access to the audit records (FAU_SAR.2). The TOE provides a review facility which allows searching of audit trails (FAU_SAR.3). The authorized user must have the capability to specify which audit records are generated (FAU_SEL.1). The TOE prevents the audit log from being modified or deleted (FAU_STG.1) and ensures that the audit log is not lost due to resource shortage (FAU_STG.3, FAU_STG.4). To support auditing, the TOE is able to maintain proper time stamps (FPT_STM.1). The protection of reused resources ensures that no data leaks from other protected sources FDP_RIP.2, FDP_RIP.3.
O.CRYPTO.NET	The cryptographically-protected network protocol (FCS_COP.1(NET)) is supported by the generation of symmetric keys (FCS_CKM.1(SYM)), as well as asymmetric keys (FCS_CKM.1(RSA), FCS_CKM.1(DSA)). Key generation is supported by the provision of good-quality random numbers (FCS_RNG.1). As part of the cryptographic network protocol, the TOE securely exchanges the symmetric key with a remote trusted IT system (FCS_CKM.2(NET)). The TOE ensures that all keys are zeroized upon de-allocation (FCS_CKM.4). The cipher primitives required for the TLS protocol are specified by FCS_COP.1(TDES), FCS_COP.1(AES), FCS_COP.1(SHA1), and FCS_COP.1(SHA2). The protection of reused resources ensures that no data leaks from other protected sources (FDP_RIP.2, FDP_RIP.3).
O.DISCRETIONARY.ACCESS	The TSF must control access to resources based on the identity of users that are allowed to specify which resources they want to access for storing their data. The access control policy must have a defined scope of control (FDP_ACC.2(RACF-PSO), FDP_ACC.2(RACF-TSO), FDP_ACC.2(RACF-SYSTEM), as well as FDP_ACC.2(CP)). The rules for the access control policy are defined (FDP_ACF.1(RACF), FDP_ACF.1(CP)). When import of user data is allowed, the TOE must ensure that user data security attributes required by the access control policy are correctly interpreted (FDP_ITC.2(BA), FPT_TDC.1(BA)). The protection of reused resources ensures that no data leaks from other protected sources (FDP_RIP.2, FDP_RIP.3).
O.NETWORK.FLOW	The network information flow control mechanism controls the information flowing between different entities (FDP_IFC.2(NI)). The TOE implements a rule-set governing the information flow (FDP_IFF.1(NI)). To facilitate the information flow control, the information must be identified (FIA_UID.1) based on security attributes the TOE can maintain

Security objectives	Rationale
	(FIA_ATD.1(TU)). The TOE must ensure that security attributes of the network data required by the information flow control policy are correctly interpreted (FDP_ITC.2(BA), FPT_TDC.1(BA)). The protection of reused resources ensures that no data leaks from other protected sources (FDP_RIP.2, FDP_RIP.3).
O.SUBJECT.COM	The TSF must control the exchange of data using transient storage objects between subjects based on the identity of users. The access control policy must have a defined scope of control (FDP_ACC.2(RACF-TSO)). The rules for the access control policy are defined (FDP_ACF.1(RACF)). When import of user data is allowed, the TOE must ensure that user data security attributes required by the access control policy are correctly interpreted (FDP_ITC.2(BA), FPT_TDC.1(BA)). The protection of reused resources ensures that no data leaks from other protected sources (FDP_RIP.2, FDP_RIP.3).
O.I&A	The TSF must ensure that only authorized users gain access to the TOE and its resources. Human users authorized to access the TOE must use an identification and authentication process (FIA_UID.1, FIA_UAU.1). Multiple I&A mechanisms are allowed as specified in FIA_UAU.5 and supported by FPT_TDC.1(TLS) covering the X.509 certificate authentication. To ensure authorized access to the TOE, authentication data is protected (FIA_ATD.1(HU), FIA_UAU.7). Proper authorization for subjects acting on behalf of users is also ensured (FIA_USB.2). The appropriate strength of the authentication mechanism is ensured (FIA_SOS.1). To support the strength of authentication methods, the TOE is capable of identifying and reacting to unsuccessful authentication attempts (FIA_AFL.1). In addition, user-initiated and TSF-initiated session locking (FTA_SSL.1, FTA_SSL.2) protect the authenticated user's session. The protection of reused resources ensures that no data leaks from other protected sources (FDP_RIP.2, FDP_RIP.3).
O.MANAGE	The TOE provides management interfaces globally defined in FMT_SMF.1 for: • the access control policies FMT_MSA.1(DAC), FMT_MSA.3(DAC); • the information flow control policy FMT_MSA.3(NI), FMT_MTD.1(NI); • the auditing aspects FMT_MTD.1(AE), FMT_MTD.1(AS), FMT_MTD.1(AT)]; • the identification and authentication aspects FMT_MTD.1(IAT), FMT_MTD.1(IAF), FMT_MTD.1(IAU). Persistently stored user data is stored either in hierarchical or relational fashion, which implies an inheritance of security attributes from parent objects (FMT_MSA.4(DAC)). The rights management for the different management aspects is defined with FMT_SMR.1. The management interfaces for the revocation of user and object attributes is provided with FMT_REV.1(OBJ) and FMT_REV.1(USR).

Security objectives	Rationale
O.TRUSTED_CHANNEL	The TOE provides a trusted channel protecting communication between a remote trusted IT system and itself (FTP_ITC.1).
O.LS.CONFIDENTIALITY	The information flow control policy is defined by specifying the subjects, objects, security attributes and rules in FDP_IFC.2(LS) (Labeled Security Mode only), FDP_IFF.2(LS) (Labeled Security Mode only). Supportive to the enforcement of the policy are the automated label assignment when exporting data (FDP_ETC.2(LS) (Labeled Security Mode only)) and during the import of data (FDP_ITC.1(LS) (Labeled Security Mode only), FDP_ITC.2(LS) (Labeled Security Mode only)). For assigning labels to imported data, the label information transmitted with the data must be interpretable by the TOE (FPT_TDC.1(LS) (Labeled Security Mode only)).
O.LS.PRINT	The addition of label information on exported data during printing is governed by FDP_ETC.2(LS) (Labeled Security Mode only).
O.LS.LABEL	The assignment of labels to users is performed during user-subject binding (FIA_USB.1(LS) (Labeled Security Mode only)) with security attributes maintained by the TOE (FIA_ATD.1(LS)). Object labels are assigned to objects when importing them into the TOE (FDP_ITC.1(LS) (Labeled Security Mode only), FDP_ITC.2(LS) (Labeled Security Mode only), FPT_TDC.1(LS) (Labeled Security Mode only)). The management of labels is allowed for the TOE with FMT_MSA.1(LS) (Labeled Security Mode only), FMT_MSA.3(LS) (Labeled Security Mode only).
O.COMP.INFO_FLOW_CTRL	The information flow control policy covering the runtime of the compartments is specified with FDP_IFC.2(VIRT) and FDP_IFF.1(VIRT). As the TOE shall allow export of data belonging to compartments, the TOE assigns the security attributes for enforcing the information flow control policy to the communicated data as specified with FDP_ETC.2(VIRT), FDP_ITC.2(VIRT), and FPT_TDC.1(VIRT). Management of the security attributes for the information flow control policy is specified with FMT_MSA.1(VIRT-CIFCP), and FMT_MSA.3(VIRT-CIFCP) as well as FMT_MTD.1(VIRT-COMP).
O.COMP.RESOURCE_ACCESS	The access control policy for the resources belonging to the different compartments is defined with FDP_ACC.2(RACF-PSO), FDP_ACC.2(RACF-TSO) and FDP_ACF.1(RACF). As the TOE shall allow export of data belonging to compartments, the TOE assigns the security attributes for enforcing the access control policy to the communicated data as specified with FDP_ETC.2(VIRT), FDP_ITC.2(VIRT), and FPT_TDC.1(VIRT). Management of the security attributes for the access control policy is specified with FMT_MSA.1(DAC), and FMT_MSA.3(DAC) as well as FMT_MTD.1(VIRT-COMP).
O.COMP.IDENT	The identification of compartments to support the information flow control and access control policies is established with FIA_UID.2(VIRT).

Security objectives	Rationale
O.SSI.VMDATAPROTECTION	The implementation of the transmission channel for the virtual machine data is established with FPT_TRC.1(SSIVM). The TOE ensures that the channel used for this virtual machine data communication is protected with FPT_ITT.1(SSIVM).
O.SSI.TSFDATAPROTECTION	The implementation of the transmission channel for the SSI cluster state is established with FPT_TRC.1(SSITSF). The TOE ensures that the channel used for this cluster communication is protected with FPT_ITT.1(SSITSF).
O.SSI.LGRMGT	The restriction of the operational access to the SSI cluster is established with FMT_MTD.1(LGR).

Table 8: Security objectives for the TOE rationale

6.3.3 Security Requirements Dependency Analysis

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

Security functional requirement	Dependencies	Resolution
FAU_GEN.1	FPT_STM.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1	FAU_GEN.1
	FIA_UID.1	FIA_UID.1
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1
FAU_SAR.2	FAU_SAR.1	FAU_SAR.1
FAU_SAR.3	FAU_SAR.1	FAU_SAR.1
FAU_SEL.1	FAU_GEN.1	FAU_GEN.1
	FMT_MTD.1	FMT_MTD.1(AE)
FAU_STG.1	FAU_GEN.1	FAU_GEN.1
FAU_STG.3	FAU_STG.1	FAU_STG.1
FAU_STG.4	FAU_STG.1	FAU_STG.1
FCS_CKM.1(SYM)	[FCS_CKM.2 or FCS_COP.1]	FCS_COP.1(NET) FCS_COP.1(TDES) FCS_COP.1(AES)
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.1(RSA)	[FCS_CKM.2 or FCS_COP.1]	FCS_COP.1(NET)
	FCS_CKM.4	FCS_CKM.4

Security functional requirement	Dependencies	Resolution
FCS_CKM.1(DSA)	[FCS_CKM.2 or FCS_COP.1]	FCS_COP.1(NET)
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.2(NET)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(SYM) FCS_CKM.1(RSA) FCS_CKM.1(DSA)
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(SYM)
FCS_COP.1(TDES)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(SYM)
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1(AES)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(SYM)
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1(SHA1)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	SHA1 does not require cryptographic keys.
	FCS_CKM.4	SHA1 does not require cryptographic keys.
FCS_COP.1(SHA2)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	SHA2 does not require cryptographic keys.
	FCS_CKM.4	SHA2 does not require cryptographic keys.
FCS_COP.1(NET)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(SYM) FCS_CKM.1(RSA) FCS_CKM.1(DSA)
	FCS_CKM.4	FCS_CKM.4
FCS_RNG.1	No dependencies.	
FDP_ACC.2(RACF-PSO)	FDP_ACF.1	FDP_ACF.1(RACF)
FDP_ACC.2(RACF-TSO)	FDP_ACF.1	FDP_ACF.1(RACF)
FDP_ACC.2(RACF-SYSTEM)	FDP_ACF.1	FDP_ACF.1(RACF)
FDP_ACC.2(CP)	FDP_ACF.1	FDP_ACF.1(CP)
FDP_ACF.1(RACF)	FDP_ACC.1	FDP_ACC.2(RACF-PSO) FDP_ACC.2(RACF-TSO) FDP_ACC.2(RACF-SYSTEM)
	FMT_MSA.3	FMT_MSA.3(DAC)
FDP_ACF.1(CP)	FDP_ACC.1	FDP_ACC.2(CP)
	FMT_MSA.3	FMT_MSA.3(DAC)

Security functional requirement	Dependencies	Resolution
FDP_IFC.2(NI)	FDP_IFF.1	FDP_IFF.1(NI)
FDP_IFF.1(NI)	FDP_IFC.1	FDP_IFC.2(NI)
	FMT_MSA.3	FMT_MSA.3(NI)
FDP_ITC.2(BA)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.2(RACF-PSO) FDP_ACC.2(RACF-TSO) FDP_ACC.2(RACF-SYSTEM) FDP_ACC.2(CP) FDP_IFC.2(NI)
	[FTP_ITC.1 or FTP_TRP.1]	FTP_ITC.1
	FPT_TDC.1	FPT_TDC.1(BA)
FDP_RIP.2	No dependencies.	
FDP_RIP.3	No dependencies.	
FIA_AFL.1	FIA_UAU.1	FIA_UAU.1
FIA_ATD.1(HU)	No dependencies.	
FIA_ATD.1(TU)	No dependencies.	
FIA_SOS.1	No dependencies.	
FIA_UAU.1	FIA_UID.1	FIA_UID.1
FIA_UAU.5	No dependencies.	
FIA_UAU.7	FIA_UAU.1	FIA_UAU.1
FIA_UID.1	No dependencies.	
FIA_USB.2	FIA_ATD.1	FIA_ATD.1(HU)
FMT_MSA.1(DAC)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.2(RACF-PSO) FDP_ACC.2(RACF-TSO) FDP_ACC.2(RACF-SYSTEM)
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.3(DAC)	FMT_MSA.1	FMT_MSA.1(DAC)
	FMT_SMR.1	FMT_SMR.1
FMT_MSA.3(NI)	FMT_MSA.1	Satisfied with FMT_MTD.1(NI) as per [OSPP].
	FMT_SMR.1	FMT_SMR.1
FMT_MSA.4(DAC)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.2(RACF-PSO)

Security functional requirement	Dependencies	Resolution
FMT_MTD.1(AE)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(AS)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(AT)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(AF)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(NI)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(IAT)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(IAF)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(IAU)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_REV.1(OBJ)	FMT_SMR.1	FMT_SMR.1
FMT_REV.1(USR)	FMT_SMR.1	FMT_SMR.1
FMT_SMF.1	No dependencies.	
FMT_SMR.1	FIA_UID.1	FIA_UID.1
FPT_STM.1	No dependencies.	
FPT_TDC.1(BA)	No dependencies.	
FPT_TDC.1(TLS)	No dependencies.	
FTA_SSL.1	FIA_UAU.1	FIA_UAU.1
FTA_SSL.2	FIA_UAU.1	FIA_UAU.1
FTP_ITC.1	No dependencies.	
FDP_ETC.2(VIRT)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.2(RACF-PSO) FDP_ACC.2(RACF-TSO) FDP_IFC.2(VIRT)

Security functional requirement	Dependencies	Resolution
FDP_IFC.2(VIRT)	FDP_IFF.1	FDP_IFF.1(VIRT)
FDP_IFF.1(VIRT)	FDP_IFC.1	FDP_IFC.2(VIRT)
	FMT_MSA.3	FMT_MSA.3(VIRT-CIFCP)
FDP_ITC.2(VIRT)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.2(RACF-PSO) FDP_ACC.2(RACF-TSO) FDP_IFC.2(VIRT)
	[FTP_ITC.1 or FTP_TRP.1]	FTP_ITC.1
	FPT_TDC.1	FPT_TDC.1(VIRT)
FIA_UID.2(VIRT)	No dependencies.	
FMT_MSA.1(VIRT-CIFCP)	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.2(VIRT)
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.3(VIRT-CIFCP)	FMT_MSA.1	FMT_MSA.1(VIRT-CIFCP)
	FMT_SMR.1	FMT_SMR.1
FMT_MTD.1(VIRT-COMP)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FPT_TDC.1(VIRT)	No dependencies.	
FDP_ETC.2(LS) (Labeled Security Mode only)	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.2(LS) (Labeled Security Mode only)
FDP_IFC.2(LS) (Labeled Security Mode only)	FDP_IFF.1	FDP_IFF.2(LS) (Labeled Security Mode only)
FDP_IFF.2(LS) (Labeled Security Mode only)	FDP_IFC.1	FDP_IFC.2(LS) (Labeled Security Mode only)
	FMT_MSA.3	FMT_MSA.3(LS) (Labeled Security Mode only)
FDP_ITC.1(LS) (Labeled Security Mode only)	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.2(LS) (Labeled Security Mode only)
	FMT_MSA.3	FMT_MSA.3(LS) (Labeled Security Mode only)
FDP_ITC.2(LS) (Labeled Security Mode only)	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.2(LS) (Labeled Security Mode only)
	[FTP_ITC.1 or FTP_TRP.1]	FTP_ITC.1
	FPT_TDC.1	FPT_TDC.1(LS) (Labeled Security Mode only)

Security functional requirement	Dependencies	Resolution
FIA_ATD.1(LS)	No dependencies.	
FIA_USB.1(LS) (Labeled Security Mode only)	FIA_ATD.1	FIA_ATD.1(LS)
FMT_MSA.1(LS) (Labeled Security Mode only)	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.2(LS) (Labeled Security Mode only)
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.3(LS) (Labeled Security Mode only)	FMT_MSA.1	FMT_MSA.1(LS) (Labeled Security Mode only)
	FMT_SMR.1	FMT_SMR.1
FPT_TDC.1(LS) (Labeled Security Mode only)	No dependencies.	
FPT_ITT.1(SSIVM)	No dependencies.	
FPT_ITT.1(SSITSF)	No dependencies.	
FPT_TRC.1(SSIVM)	FPT_ITT.1	FPT_ITT.1(SSIVM)
FPT_TRC.1(SSITSF)	FPT_ITT.1	FPT_ITT.1(SSITSF)
FMT_MTD.1(LGR)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1

Table 9: TOE SFR dependency analysis

6.3.4 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

6.4 Security Assurance Requirements

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

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

The following table shows the Security assurance requirements, and the operations performed on the components according to CC part 3: iteration (Iter.), refinement (Ref.), assignment (Ass.) and selection (Sel.).

Security assurance class	Security assurance requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
ASE Security Target evaluation	ASE_CCL.1(CCL) Conformance claims	CC Part 3	No	Yes	No	No
	ASE_INT.1 ST introduction	CC Part 3	No	No	No	No
	ASE_SPD.1 Security problem definition	CC Part 3	No	No	No	No
	ASE_OBJ.2 Security objectives	CC Part 3	No	No	No	No
	ASE_ECD.1 Extended components definition	CC Part 3	No	No	No	No
	ASE_REQ.2 Derived security requirements	CC Part 3	No	No	No	No
	ASE_TSS.1 TOE summary specification	CC Part 3	No	No	No	No
ADV Development	ADV_ARC.1 Security architecture description	CC Part 3	No	No	No	No
	ADV_FSP.4 Complete functional specification	CC Part 3	No	No	No	No

Security assurance class	Security assurance requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
	ADV_IMP.1 Implementation representation of the TSF	CC Part 3	No	No	No	No
	ADV_TDS.3 Basic modular design	CC Part 3	No	No	No	No
AGD Guidance documents	AGD_OPE.1 Operational user guidance	CC Part 3	No	No	No	No
	AGD_PRE.1 Preparative procedures	CC Part 3	No	No	No	No
ALC Life-cycle support	ALC_CMC.4 Production support, acceptance procedures and automation	CC Part 3	No	No	No	No
	ALC_CMS.4 Problem tracking CM coverage	CC Part 3	No	No	No	No
	ALC_DEL.1 Delivery procedures	CC Part 3	No	No	No	No
	ALC_DVS.1 Identification of security measures	CC Part 3	No	No	No	No
	ALC_FLR.3 Systematic flaw remediation	CC Part 3	No	No	No	No
	ALC_LCD.1 Developer defined life-cycle model	CC Part 3	No	No	No	No
	ALC_TAT.1 Well-defined development tools	CC Part 3	No	No	No	No
ATE Tests	ATE_COV.2 Analysis of coverage	CC Part 3	No	No	No	No
	ATE_DPT.1 Testing: basic design	CC Part 3	No	No	No	No
	ATE_FUN.1 Functional testing	CC Part 3	No	No	No	No
	ATE_IND.2 Independent testing - sample	CC Part 3	No	No	No	No
AVA Vulnerability assessment	AVA_VAN.3 Focused vulnerability analysis	CC Part 3	No	No	No	No

Table 10: Security assurance requirements

6.4.1 Security Target evaluation (ASE)

6.4.1.1 Conformance claims (ASE_CCL.1(CCL))

Content and presentation elements:

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

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

6.5 Security Assurance Requirements Rationale

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

The refinement of ASE_CCL.1.10C is considered to include certain requirements of the [OSPP] with which the ST author must comply. These requirements specify conditional requirements that only apply when the TOE shows special properties or mechanisms. The [CC] does not define such conditional statements, which are therefore introduced by the [OSPP].

7 TOE Summary Specification

7.1 TOE Security Functionality

This chapter provides a summary of the security functions of z/VM that are subject to the evaluation. z/VM has more security functions than described in this chapter; only those that implement the security requirements claimed in chapter 1.5 are described here.

7.1.1 Overview of the TOE architecture

z/VM is an operating system operating on IBM System z 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 Labeled Security 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 Labeled Security Mode) mandatory access control as well as users and groups with their related attributes (F.SM)
7. Single System Image cluster operation supporting live guest relocation (F.SSI)
8. 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.

7.1.2 F.AU: Auditing

7.1.2.1 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

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

This section maps to the following SFRs:

- FAU_GEN.1
- FAU_GEN.2
- FPT_STM.1

7.1.2.2 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.

This section maps to the following SFRs:

- FAU_SAR.1
- FAU_SAR.2
- FAU_STG.1 and FMT_MTD.1(AS)

7.1.2.3 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.

This section maps to the following SFRs:

- FAU_SAR.3
- FAU_SEL.1 and FMT_MTD.1(AE)
- FAU_STG.3 and FMT_MTD.1(AT)
- FAU_STG.4 and FMT_MTD.1(AF)

7.1.3 F.AC: Access Control

7.1.3.1 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 Object (Discretionary) and (in Labeled Security 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 7.1.3.6.

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.

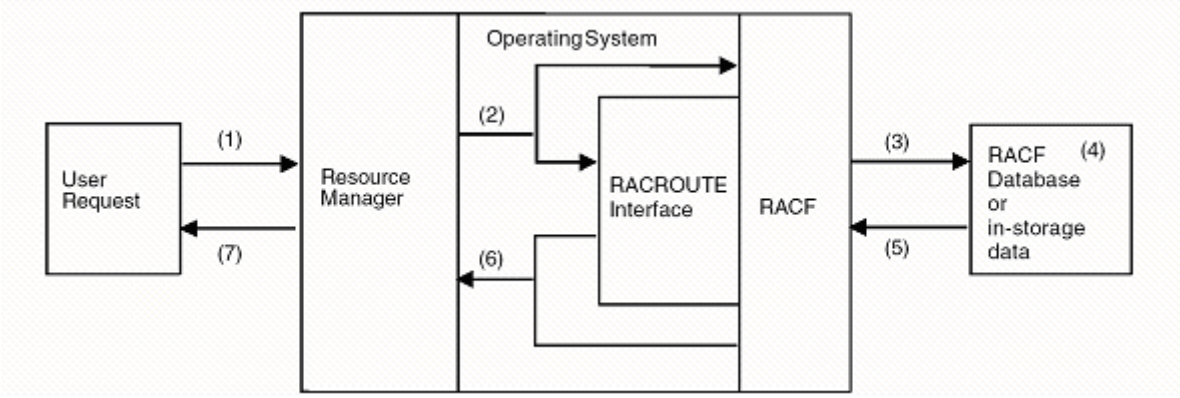

Figure 1: RACF and its relationship to the operating system

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 Labeled Security 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). In the evaluated configuration, CP interprets the "don't know" return code as "no".

7.1.3.2 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

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
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 11: 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.

This section maps to the following SFRs:

- FDP_ACC.2(RACF-PSO), FDP_ACC.2(RACF-TSO), FDP_ACC.2(RACF-SYSTEM), and FDP_ACC.2(CP)
- FDP_IFC.2(NI), and FDP_IFC.2(VIRT)
- FIA_ATD.1(HU), and FIA_ATD.1(LS)
- FMT_MSA.1(DAC), FMT_MSA.1(LS), and FMT_MSA.1(VIRT-CIFCP)

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 12: RACF group profile

This section maps to the following SFRs:

- FDP_ACC.2(RACF-PSO), FDP_ACC.2(RACF-TSO), FDP_ACC.2(RACF-SYSTEM), and FDP_ACC.2(CP)
- FDP_IFC.2(NI), and FDP_IFC.2(VIRT)
- FIA_ATD.1(HU), and FIA_ATD.1(LS)
- FMT_MSA.1(DAC), FMT_MSA.1(LS), and FMT_MSA.1(VIRT-CIFCP)

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
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 13: 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.

This section maps to the following SFRs:

- FAU_SAR.2
- FAU_STG.1
- FDP_ACC.2(RACF-PSO), FDP_ACC.2(RACF-TSO), FDP_ACC.2(RACF-SYSTEM), and FDP_ACC.2(CP)
- FDP_IFC.2(LS), and FDP_IFC.2(VIRT)
- FIA_ATD.1(HU), and FIA_ATD.1(LS)
- FMT_MSA.1(DAC), FMT_MSA.1(LS), and FMT_MSA.1(VIRT-CIFCP)

7.1.3.3 F.AC.3 – Access control enforcement

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 Labeled Security Mode)
- The access authority specified in the resource profile

For printing support, the TOE marks the print output with the label of the user started the print job on the banner and trailer page as well as on the top and bottom of each page.

Data archival and restore allows storing of the meta data for the user data, including associated labels. When restoring data, the archived label is enforced on the restored data set.

This section maps to the following SFRs:

- FAU_SAR.2
- FAU_STG.1
- FDP_ACF.1(RACF) and FDP_ACF.1(CP)
- FDP_ETC.2(LS), FDP_ITC.2(LS) for printing and data archival, FPT_TDC.1(LS) for data archival and restore
- FDP_IFF.1(VIRT)
- FDP_IFF.2(LS)

- FDP_ITC.1(LS)
- FMT_MSA.4 (implicitly as no object generation facility is provided by the TOE)

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).

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 labeled security mode)
- SECLEVELAUDIT or NOSECLEVELAUDIT (in labeled security 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.

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.

Security classification (Labeled security 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.

In order for a user to acquire the access rights defined by the security label, the user must be explicitly authorized to access the label. The user’s default label is assigned by the security administrator.

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 a non-default security label by using the "LOGON *userid* SECLABEL *seclabel*" command.

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 in a class other than VMMDISK, 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 or in a discrete profile in the VMMDISK class, 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).

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.

7.1.3.4 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

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.

This section maps to the following SFRs:

- FMT_MSA.1(DAC), FMT_MSA.1(LS)
- FMT_MSA.3(DAC), FMT_MSA.3(LS)
- FMT_MTD.1(IAT), FMT_MTD.1(IAF), FMT_MTD.1(IAU)
- FMT_REV.1(OBJ)
- FMT_SMF.1
- FMT_SMR.1

7.1.3.5 F.AC.5 – Protected Resources

On z/VM, RACF can be used to control access to all objects with discretionary and 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 the SYSLOW security label.
- 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 Labeled security mode)
  If security labels are used, and, if so, their definitions.
- SURROGAT

If surrogate login or access is allowed, and if allowed, which user IDs can act as surrogates.

- TERMINAL
  Terminals.

7.1.3.6 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.

Privilege classes

Each CP command and 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.

When a virtual machine is defined, the system administrator assigns one of more privilege classes to the virtual machine. When the virtual machine logs on, its *active* set of privileges will be the same as the *defined* set of privileges.

If the SET PRIVCLASS command is enabled by the system administrator, a user can remove privileges from his or her *active* set of privileges. The user may restore the removed privileges to their active set of privileges at any time by again using the SET PRIVCLASS command. Privilege classes that are not in the *defined* set of privileges are not permitted to be added to the *active* set.

Command and DIAGNOSE access check

When a user enters a CP command or executes a DIAGNOSE instruction, CP intercepts the operation and examines the privilege class assigned to the command or the specific code specified on the DIAGNOSE instruction. If that privilege class is currently in the issuing user's *active* set of privilege classes, the command or DIAGNOSE is potentially allowed, subject to any additional protections imposed by RACF (such as is defined for the CP STORE HOST command).

Consistency of access checks between RACF and CP

The access check on those CP commands and DIAGNOSE codes is performed sequentially. First the CP check is performed and then, if the command or DIAGNOSE is defined to have additional RACF checks, RACF is consulted. 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, is the request allowed to proceed.

7.1.4 F.I&A: Identification and Authentication

7.1.4.1 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 must be defined to RACF and are identified and authenticated by a user ID / password combination.

When authenticating a user, RACF will verify:

- If the user is defined in the RACF database. If the user ID is not defined to RACF, the virtual machine cannot be started.
- If the user has supplied a valid password or phrase, and a valid, authorized group name. Otherwise a default group name is selected. Also a security label is associated with the user (in Labeled Security mode). If a user does not have a password defined, then local or telnet logon to the virtual machine console is not permitted.
- If the user ID has the REVOKE attribute, the virtual machine cannot be started.
- If the user's group has the REVOKE attribute, the user cannot enter the system as a member of 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 and permits CP to create the user's virtual machine.

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 previously well-established.

If the connection between CP and RACF is severed for any reason, no security-sensitive activities (LOGON, LINK, MESSAGE, etc.) will be permitted, except as described below. The connection can be severed as a result of the RACF server being forcibly removed from the system (CP FORCE), abnormal termination of the RACF service, or due to explicit action by the RACF server itself.

In the evaluated configuration, the RACF server is configured to sever its connection to CP in the event the both audit logs are full. In the event RACF services are unavailable, only select administrative user IDs can login to the system to repair the situation. These user IDs are authenticated using the password maintained in the System Directory (USER DIRECT).

The z/VM Telnet server uses the SCANINTERVAL, INACTIVE and TIMEMARK parameters (as part of the INTERNALCLIENTPARMS statement) to establish a means through which the Telnet server will disconnect a session which is inactive for a configurable number of seconds. This mechanism provides the automated session protection.

The CP DISCONNECT command allows a user to disconnect from the virtual machine terminal. A user would need to reauthenticate before access to the session data is restored.

The SSL server allows the configuration of a bi-directional certificate verification. This mechanism therefore provides the token-based authentication. As the SSL server establishes the communication channel between a remote entity and the CP console, a user still needs to provide his password/passphrase to authenticate with the CP console.

This functionality maps to the following SFRs:

- FTA_SSL.1
- FTA_SSL.2
- FIA_UAU.5
- FPT_TDC.1(TLS)

7.1.4.2 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. So that self-service security management software can be implemented, RACF also includes the ability to set an *unexpired* password. This is intended for use only by automation software operating on behalf of the end user.

Using the SETROPTS PASSWORD RULES command, a system administrator can define the rules for forming valid passwords. Additional suboptions are provided to enable the administrator to control the maximum lifetime of a password (the change interval) and the number of password changes required before a password may be reused.

If desired, a user can use the PASSWORD command set their password change interval to any value *less* than the interval set by SETROPTS PASSWORD.

All password change and history policies defined by SETROPTS PASSWORD also apply to password phrases. The syntax requirements for password phrases are contained within the installation-controlled exit routine ICHPWX11.

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. Neither the clear-text password nor its encrypted form are stored in the RACF database.

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

When the user provides wrong passwords in consecutive authentication attempts, the account status is set to REVOKE by the TSF until the administrator re-enables the account. For accounts with the SPECIAL attribute, the system operator is prompted whether the account status of the offending user shall be set to REVOKE when the limit for consecutive failed authentication attempts is surpassed.

This functionality maps to the following SFRs:

- FIA_AFL.1
- FIA_SOS.1
- FIA_UAU.1

- FIA_UAU.7
- FIA_UID.1
- FIA_USB.2

As the identifier for virtual machines is identical to the user identifier, FIA_UID.2(VIRT) and FIA_USB.1(LS) is also covered by this functionality.

7.1.4.3 F.I&A.3 – Identity Change

During runtime of a virtual machine, an authorized user can switch his identity using the DIAGNOSE 0xD4 instruction. The changed user ID applies to all subsequent access control checks (DAC and MAC) for the LINK command, IUCV connections, and spool file transmission. Using RACF, the administrator is able to limit the target user IDs a particular user can impersonate. This is a privileged (class B) function that is not available to general (class G) users. It is used by trusted virtual machines to do work on behalf of other users.

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

In Labeled Security 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.

This functionality maps to the following SFRs:

- FIA_USB.1(LS)
- FIA_USB.2

7.1.5 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 peripheral devices to perform input and output tasks. For unassisted I/O operations, z/VM performs the I/O on behalf of the virtual machine. If the I/O is assisted by the PR/SM firmware, the I/O is handled by the firmware without interception by CP.
- 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 1.5.1.1 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.

CP enforces a strict separation of the virtual machines. To accomplish this, CP ensures:

- Virtual machines can only access memory that is mapped to page frames in real memory. The mapping is controlled by CP and is not accessible by the virtual machine. Memory references by the virtual machine to pages not contained of the virtual machine's memory configuration (as defined in the System Directory) will result in an addressing exception program interrupt. References to pages that are defined, but which do not exist or that have been swapped out will be resolved by CP and the operation retried. 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 those explicitly 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 Virtual Switch	DAC / MAC enforcement Multidirectional channel between all configured virtual machines

Communication channel	Boundary
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 14: Communication channel usage

This section maps to the following SFRs:

- FDP_IFC.2(VIRT) and FDP_IFF.1(VIRT)
- FIA_UID.2(VIRT) as each virtual machine is assigned with the unique user identifier that is maintained by CP

7.1.5.1 Access to virtual machines

The TOE provides access to the virtual machine's consoles as well as to the virtual machine's CP command line after the user has identified and authenticated. Remote access to the console and the CP command line is provided with the TCP/IP server application executing in its own virtual machine. That TCP/IP server implements a Telnet server that provides the connection between the virtual machine console or virtual machine CP command line and the remote entity.

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.

This section maps to the following SFRs:

- FDP_ETC.2(VIRT) and FDP_ITC.2(VIRT)

The TOE also provides an SSL server operating again in a separate virtual machine that implements the TLS protocol, including the initial handshake, as well as the encryption and decryption of data.

The TCP/IP server utilizes the SSL server when it detects TLS traffic. When a TLS handshake request is detected, the TCP/IP server forwards the request to the SSL server to process the handshake and to generate the replies. The TCP/IP server submits the handshake replies to establish a TLS tunnel.

Subsequently, any TLS traffic received by the TCP/IP server is forwarded to the SSL server for decryption. When data is supposed to be send to the remote peer, the TCP/IP server hands the data to the SSL server for encryption, obtains the encrypted data and forwards the data to the remote entity.

To facilitate the TLS protocol, the SSL server is able to generate random numbers to generate the TLS pre-master secret as well as the RSA and DSA keys. In addition, the SSL server implements the cryptographic primitives needed for the TLS protocol.

The random numbers required for the the cryptographic operations of the TLS protocol is delivered with a FIPS 186-3 compliant RNG that is seeded with 160 bits of entropy.

SystemSSL is the main provider of basic cryptographic services within z/VM and for the functions specified in the SFRs is used for the basic cryptographic services for certificate/key generation for certificates used for user authentication as well as certificates used in the establishment of trusted channels. Also the basic cryptographic functions used for the TLS protocol are provided by SystemSSL, unless the basic functions are already provided by CPACF, such as AES, TDES, SHA-1 and SHA-2.

Although SystemSSL exports interfaces for cryptographic functions that can be used by unauthorized user programs, those functions are not related to SFRs defined in this Security Target except that those functions are used within the TSF by the TLS Server component to provide the cryptographic services for setting up a trusted channel.

This section maps to the following SFRs:

- FCS_CKM.1(SYM), FCS_CKM.1(RSA), FCS_CKM.1(DSA)
- FCS_CKM.2(NET)
- FCS_CKM.4
- FCS_COP.1(NET)

- FCS_RNG.1
- FTP_ITC.1

CPACF

CP Assist Cryptographic Function (CPACF) is a microcode assist function plus hardware component implemented in a coprocessor (shared with a data compress function) accessed by a pair of PUs within the PU chip in an MCM. CPACF provides high performance encryption and decryption support. It is a hardware-synchronous implementation; that is, holding processor processing of the instruction flow until the operation completes. Several instructions are able to invoke the CPACF, when executed by the PU:

- KMAC Compute Message Authentic Code
- KM Cipher Message
- KMC Cipher Message with Chaining
- KMF Cipher Message with CFB
- KMCTR Cipher Message with Counter
- KMO Cipher Message with OFB
- KIMD Compute Intermediate Message Digest
- KLMD Compute Last Message Digest
- PCKMO Provide Cryptographic Key Management Operation

CPACF offers a set of symmetric (meaning that encryption and decryption processes use the same key) cryptographic functions that enhance the encryption and decryption performance of clear key operations of TLS, VPN, and data storing applications that do not require FIPS 140-2 level 4 security. Clear key means that the key used is located in central storage.

The following algorithms are available:

- Data encryption and decryption algorithms: DES, Triple-DES, AES 128 bit, AES 192 bit, AES 256 bit
- Hashing algorithms: SHA-1, SHA-256, SHA-384, and SHA-512
- Message authentication code (MAC): single-key MAC, double-key MAC
- Pseudo Random Number Generation (PRNG) -- this PRNG is not used for the key generation mechanisms defined in this ST

These functions are directly available to application programs, thereby diminishing programming overhead of going through SystemSSL. The CPACF complements, but does not execute, public key (PKA) functions. Note that keys, when needed, are to be provided in clear form only.

Traditionally CPACF was only able to execute clear-key cryptographic algorithms. This has changed with "Protected keys in CPACF", which is available for z196, z114 and newer processor models. The key management instructions for this feature can only be executed in supervisor state.

When the wrapping key is unique to each LPAR, a protected key cannot be shared with another LPAR. CPACF, using key wrapping, ensures that key material is not visible to applications or operating systems during encryption operations.

CPACF code generates the wrapping key and stores it in the protected area of hardware system area (HSA). The wrapping key is accessible only by firmware. It cannot be accessed by operating systems or applications. DES/TDES and AES algorithms were implemented in CPACF code with support of hardware assist functions. Two variations of wrapping key are generated, one for DES/TDES keys and another for AES keys.

Wrapping keys are generated during the clear reset each time an LPAR is activated or reset. There is no customizable option available at SE or HMC that permits or avoids the wrapping key generation.

This section maps to the following SFRs:

- FCS_COP.1(TDES), FCS_COP.1(AES), FCS_COP.1(SHA1), FCS_COP.1(SHA2)

7.1.5.2 Virtual machine networking

CP allows virtual machines to communicate as part of virtual networks maintained by CP as well as to communicate with external entities. CP assigns each virtual machine an IP address that can be used by external entities to communicate with the virtual machine.

In addition, CP can restrict network communication based on VLAN tags. Each guest can be associated with a VLAN tag where the TOE maintains the VLAN to virtual machine mapping configuration. Only if the VLAN tag present in the IP packet matches the VLAN tag of a virtual machine, the packet is forwarded.

For communication originating by virtual machines, CP ensures that the proper VLAN tag is added to each outgoing packet.

If no VLAN tags or communication restrictions based on VLAN tags are configured for a virtual machine, that virtual machine is not subject to any communication restriction.

This section maps to the following SFRs:

- FIA_ATD.1(TU)
- FDP_IFC.2(NI) and FDP_IFF.1(NI)
- FDP_ITC.2(BA) (to ensure that data intended for one user is sent to that user's virtual machine) and FPT_TDC.1(BA)
- FMT_MSA.3(NI)
- FPT_TDC.1(VIRT)

7.1.6 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 each time a virtual CPU is dispatched.
- Temporary disk space is cleared automatically when the FEATURES ENABLE CLEAR_TDISK option is specified 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 disk storage space used to contain minidisks prior to allocation to a virtual machine,
- Clearing of temporary disk space prior to re-allocation to another virtual machine,

- Erasure of reusable removable media such as tapes prior to re-assignment to another user.

This section maps to the following SFRs:

- FDP_RIP.2
- FDP_RIP.3

7.1.7 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 Labeled Security Mode) of security labels for mandatory access control
- Management of the audit trail and the events to be audited (FAU_SEL.1 and FMT_MTD.1(AE), FMT_MTD.1(AS), FMT_MTD.1(AT), FMT_MTD.1(AF))
- Management of user security attributes, including authentication data and access control (FMT_MTD.1(IAT), FMT_MTD.1(IAF), FMT_MTD.1(IAU), FMT_REV.1(USR))
- Management of VLAN to virtual machine mappings (FMT_MTD.1(NI))
- Management of virtual machine resource assignments (FMT_MSA.3(VIRT-CIFCP) and FMT_MTD.1(VIRT-COMP))

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)

This and the following sections explain the authorizations needed for performing administrative tasks and cover therefore FMT_SMF.1, FMT_SMR.1.

7.1.7.1 F.SM.1 – Management of user security attributes

RACF manages the database holding various security attributes assigned to a user. On z/VM, authorized users can enter RACF commands by preceding the command name with RAC, by entering a RACF command session, or by use of RACF ISPF panels. The ISPF 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 Labeled Security 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

7.1.7.2 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). (Labeled Security Mode)

7.1.7.3 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 z/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 (Labeled Security 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).

7.1.7.4 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.

7.1.8 F.SSI: Single System Image

z/VM offers a multi-system clustering technology allowing up to four z/VM instances in a single system image (SSI) cluster. This technology is important, because it offers clients an attractive alternative to vertical growth by adding new z/VM systems. In the past, this capability required duplicate efforts to install, maintain, and manage each system. With SSI, these duplicate efforts are reduced or eliminated.

Support for live guest relocation (LGR) allows the administrator to move Linux virtual servers without disruption to the business, helping you to avoid planned outages. The z/VM systems are aware of each other and can take advantage of their combined resources. LGR enables clients to avoid loss of service due to planned outages by relocating guests from a system requiring maintenance to a system that remains active during the maintenance period.

The SSI cluster synchronization is established via shared volumes which host the PDR as well as the system configuration. Using the PDR, the individual z/VM member systems of the SSI cluster inform the other members about their state. The aggregation of the individual state information forms the SSI cluster state. Based on this SSI cluster state, the cluster members decide whether operations affecting the cluster or shared resources are allowed.

The z/VM member systems of the SSI cluster establish CTC communication links between each pair of member systems. These links are used to communicate the virtual machine data when a live guest relocation is performed. These CTC communication links are exclusively maintained for the purpose of the cluster communication and are not used for any other operations. The CP of each z/VM member system has exclusive access to these links, preventing virtual machines from accessing these communication links.

Privileged commands are provided by CP of each z/VM member system allow administrators to initiate, interrupt and configure the live guest relocation facility.

This section maps to the following SFRs:

- FPT_ITT.1(SSITSF) and FPT_TRC.1(SSITSF)

- FPT_ITT.1(SSIVM) and FPT_TRC.1(SSIVM)
- FMT_MTD.1(LGR)

7.1.9 F.TP: TOE Self Protection

7.1.9.1 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's memory protection mechanisms,
- Circumvent or disable minidisk protection mechanisms,
- Access a resource protected by RACF to which the virtual machine is not authorized,
- Access a virtual machine using a CP-managed passwords (except when the system is being operated in recovery mode)
- Obtain control outside the SIE environment or with privilege class authority or directory capabilities greater than those it was assigned. This refers 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 unprivileged or untrusted users.
- 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.

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-managed timer to allow scheduling of virtual processors (processors visible from inside a virtual machine) and CP execution time on logical processors.

The primary input to the SIE instruction is the SIE Descriptor. It contains a variety of architectural information about the virtual processor, including the Program Status Word and the location of the address translation tables used by SIE.

Access register translation (ART) and dynamic address translation (DAT) protect resident memory objects, whether that memory is shared or exclusive to a single user. ART and DAT are hardware facilities used by the machine during the execution of any instruction to translate a virtual address into the corresponding real address. The system depends on 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, or the Control Program, unless its owner allows the other user to do so.

In the System z processor, execution of code is driven by the *Program Status Word* (PSW). The PSW holds, among other things, the results of the most recently executed instruction (the *condition code*) and information about the next instruction to be run.

When a virtual machine issues an instruction that exits the SIE environment, the processor stores the current PSW and other information about the status of the virtual machine into the SIE descriptor and the processor executes the instruction immediately following the SIE instruction. CP examines the reason for the exit from SIE and responds appropriately.

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.

Abstract Machine Modes of Operation

z/VM executes within a logical partition. The Control Program (CP) full control to all the resources allocated to the partition when it has been set up on the hardware management console (HMC). 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 peripheral devices 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.

7.1.9.2 F.TP.2 – Structure of the TOE

The trusted parts of z/VM consist of

- the Control Program kernel
- authorized applications

The z/VM kernel 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: the RACF security server and the TCP/IP stack.

In order to trust the virtual machine(s) running an instance of RACF (it is possible to run multiple instances of RACF for one z/VM instance), the Control Program must be modified. When RACF is enabled, the CP kernel is rebuilt by the system service tools to include:

- A list of all user IDs that are planned to run an instance of RACF that CP will use and trust.
- RACF modifications to enable the CP Access Control Interface (ACI). The ACI is the mechanism used by CP to detect the presence of a security product (RACF) and to communicate with it. This includes enablement of the *RPI IUCV system service used by the RACF server to establish a bidirectional connection with CP. Using this connection, CP sends requests to RACF and receives responses.

The TCP/IP does not modifications to CP. 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.

Protection of Trusted Applications

Certain 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 1.5.4.1.

8 Abbreviations, Terminology and References

8.1 Abbreviations

CC
Common Criteria

CEC
Central Electronic Complex

CP
Control Program

DAC
Discretionary Access Control

IPL
Initial Program Load

LGR
Live Guest Relocation

MAC
Mandatory Access Control

PSW
Program Status Word

PR/SM
Processor Resource/Systems Manager™

RACF
Resource Access Control Facility

SSI
Single System Image

TOE
Target of Evaluation

TSP
TOE Security Policy

8.2 Terminology

This section contains definitions of technical terms that are used with a meaning specific to this document. Terms defined in the [CC] are not reiterated here, unless stated otherwise.

Access
If an authorized user (virtual machine) is granted a request to operate on an object, the user is said to have *access* to that object. Access rights determine whether the user can update or only read the object, and whether the user has shared or exclusive access to the object.

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

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

Authorized Administrator
An *authorized user* who has been granted the authority to manage the TOE. Authorized administrators are expected to use this authority only in the manner prescribed by the guidance that is given to them.

Authorized User
A *user* who has been properly defined, identified, and authenticated to the Control Program (CP) and RACF. Authorized users are considered to be legitimate *users* of the TOE.

Category
See *security category*.

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

Cluster (SSI)
The z/VM Single System Image feature (VMSSI) enables up to four z/VM systems to be configured as members of an SSI cluster, sharing the following resources: User directory, DASD volumes, User minidisks, Spool files, Network devices..

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 based on individual user identity or membership in a group (PROJECTA, for example).

Live Guest Relocation (LGR)
The concept of live guest relocation (LGR) allows a running Linux guest operating system to be relocated from one member in an SSI cluster to another without the need to stop the running Linux guest.

Logical Processor (IBM)
A logical processor is a share of a *real processor* that is used by a logical partition (LPAR). Logical processors have the same behavior as *real processors*, but may "float" among the available *real processors*. The point-in-time mapping of a local processor to a *real processor* is managed by the PR/SM LPAR hypervisor which can overcommit the available CPU capacity, making LPARs wait for access to the CPU. This means that the total number of logical processors can exceed the number of *real processors*.

Mandatory Access Control (MAC)
An *access control policy* that determines *access* based on the sensitivity (SECRET, for example) or *category* (PERSONNEL or MEDICAL, for example) 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 a processor that is physically installed in the server and configured to be usable by a logical partition.

SECLABEL
See *security label*.

SECLEVEL
See *security level (IBM)*.

Security Category (IBM)
When assigned to an object, it identifies the type of information that may be held by the object. When assigned to a user, it identifies the types of information the subject is authorized to access.

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

Security Level (IBM)
A numerical value that represents the relative sensitivity of the information an object contains or that a user is permitted to access. A higher number represents a higher level of sensitivity. Security levels are sometimes referred to as *SECLEVEL*s. The equivalent MLS term is *classification*.

Security Level (MLS policy in the Bell-LaPadula model)
The combination of a hierarchical classification (called *security level* in z/VM) and a set of non-hierarchical categories that represents the sensitivity of information is known as the security level.
The equivalent term in other IBM documentation is *security label*.

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

User
A named virtual machine (virtual server) attempting to access or invoke a service offered by the TOE.

8.3 References

CC	**Common Criteria for Information Technology Security Evaluation**	
	Version	3.1R4
	Date	September 2012
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R4.pdf
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R4.pdf
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R4.pdf

OSPP **Operating System Protection Profile (with exception of SFR FCS_RNG.1, which is superseded by FCS_RNG.1 in section 5.1)**
Version 2.0
Date 2010-06-01
Location https://www.bsi.bund.de/cae/servlet/contentblob/1098082/publicationFile/88584/pp0067b_pdf.pdf

OSPP-LS **OSPP Extended Package - Labeled Security**
Version 2.0
Date 2010-05-28
Location https://www.bsi.bund.de/cae/servlet/contentblob/1098148/publicationFile/88582/pp0067_EP_zip.zip

OSPP-VIRT **OSPP Extended Package - Virtualization**
Version 2.0
Date 2010-05-28
Location https://www.bsi.bund.de/cae/servlet/contentblob/1098148/publicationFile/88582/pp0067_EP_zip.zip

PKCS1_2.1 **Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1**
Author(s) RSA Laboratories
Version 2.1
Date February 2003

RACFSAG **Resource Access Control Facility Security Administrator's Guide**
Version SA22-7683-14
Date July 2010
Location http://publibz.boulder.ibm.com/epubs/pdf/ichza7b0.pdf

RFC2631 **Diffie-Hellman Key Agreement Method**
Date June 1999

SCG **Secure Configuration Guide**
Version SC24-6230-02
Date March 2012

TLSv1.1 **The Transport Layer Security (TLS) Protocol Version 1.1**
Author(s) Network Working Group, T. Dierks, E. Rescorla (RTFM, Inc.)
Version 1.1
Date April 2006

TLSv1.2 **The Transport Layer Security (TLS) Protocol Version 1.2**
Author(s) T. Dierks, E. Rescorla (RTFM, Inc.)
Version 1.2
Date August 2008