IBM z/VM Version 7 Release 2 for VPP Security Target

Version:	**1.0**
Revision:	**1**
Status:	**Released**
Last Update:	**2022-05-16**
Classification:	**Public**

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
- z System
- VM/ESA
- z/Architecture
- z/VM

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

Revision History

Revision	Date	Author(s)	Changes to Previous Revision
1.0	2022-06-02	Author: Brian W. Hugenbruch, Editor: Stephan Mueller	Public release of ST generated on 2022-05-16

Copyright © 2022 IBM

Table of Contents

Copyright © 2022 IBM

Copyright © 2022 IBM

Copyright © 2022 IBM

Copyright © 2022 IBM

Copyright © 2022 IBM

List of Tables

Copyright © 2022 IBM

List of Figures

Copyright © 2022 IBM

1 Introduction

1.1 Security Target Identification

Title:	IBM z/VM Version 7 Release 2 for VPP Security Target
Version:	1.0
Revision:	1
Status:	Released
Date:	2022-05-16
Sponsor:	IBM Corporation
Developer:	IBM Corporation
Keywords:	Security Target, Common Criteria

1.2 TOE Identification

The TOE is IBM z/VM Version 7 Release 2 for VPP.

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 Z® mainframe servers onto which to deploy mission-critical virtual servers. A single z System 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. Note, the functionality of SSI is not covered by security claims in this ST.

Due to the functionality of being a virtual machine monitor, the protection profile for virtualization ([VPP]) is used as a basis for this ST. z/VM meets all of the requirements of the VPP. In addition, z/VM provides extensive administration capabilities and thus meeting the requirement from the protection profile for virtualization extended package for server virtualization ([SVPP]).

Copyright © 2022 IBM

z/VM provides identification and authentication of users using different authentication mechanisms, discretionary 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 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.

Copyright © 2022 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.

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 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

Copyright © 2022 IBM

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.

1.5.1.2 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).

1.5.1.3 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 z System 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. The VMCF facility, and its associated diagnose instruction, are deprecated functionality and available only for legacy use.
- 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.

Copyright © 2022 IBM

- 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.

1.5.2 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.

Copyright © 2022 IBM

- 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 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.

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. 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).

Copyright © 2022 IBM

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
- Separation of virtual machines
- Audit
- Object reuse functionality
- Security management
- TSF protection

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

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

Copyright © 2022 IBM

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
- Access authority specified in the resource profile

1.5.3.3 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.

1.5.3.4 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 mechanism.

Audit records concerning TLS connections are generated by the TLS server and are gathered by privileged users with access to the SSLADMIN command. These audit records are protected from unauthorized modification or deletion by the DAC mechanism.

1.5.3.5 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

Copyright © 2022 IBM

1.5.3.6 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 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.

1.5.3.7 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.

1.5.4 Configurations

1.5.4.1 Software Components

The Target of Evaluation, IBM z/VM Version 7 Release 2, 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

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

- TLS support for the network communication
- SSI feature selected at installation time

The following PTFs must be installed for the TOE:

- RSU1
- APAR PH24751 (FIPS 140-2 compliance)
- APAR VM66540 (direct-to-host service transfer)
- APAR PH28216 (OCSP Support)

Copyright © 2022 IBM

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, and change ownership of system resource.

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 z System processor:

- IBM Z z14 with CPACF 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

Copyright © 2022 IBM

- all network adapters supported by the TOE

1.5.5 Technical decisions

The following table enumerates the applied technical decisions raised by NIAP for the applied protection profiles. In addition, the table outlines how the technical decisions are applied to the ST.

TD	TD Summary	Application in ST
TD0139	Clarification of testing for FDP_RIP_EXT.2	The modified assurance activity will be applied during evaluation.
TD0206	Testing for Non-Existence of Disconnected Virtual Devices	The modified assurance activity will be applied during evaluation.
TD0230	ALC Assurance Activities for Server Virtualization and Base Virtualization PPs	The modified assurance activity will be applied during evaluation.
TD0249	Applicability of FTP_ITC_EXT.1	Changes to the SFR are applied to the ST.
TD0250	Hypercall Controls - FPT_HCL_EXT.1 Clarification	Changes to the SFR are applied to the ST.
TD0264	Clarification of Auditable Events for FPT_RDM_EXT.1	Changes to the SFR are applied to the ST.
TD0360	AD Server configuration in FMT_MOF_EXT.1	Changes to the SFRs are applied to the ST.
TD0363	Access Banner and applicability to programmatic interfaces	Changes to the SFR are applied to the ST.
TD0431	Modification to Cipher Suites for TLS	Changes to the SFR are applied to the ST.
TD0432	Corrections to FIA_AFL_EXT.1	Changes to the SFR are applied to the ST.
TD0433	Assurance activity for FIA_X509_EXT.1.2	The modified assurance activity will be applied during evaluation.
TD0443	FPT_VDP_EXT.1 Clarification for Assurance Activity	The modified assurance activity will be applied during evaluation.
TD0567	Security Objectives Rationale, SFR Rationale, and Implicitly Satisfied SFRs	The modified rationale is applied to this ST.
TD0617	TLSC wildcard testing	The TD does not apply to the ST as the TLS client is not claimed.

Copyright © 2022 IBM

2 CC Conformance Claim

This Security Target is CC Part 2 extended and CC Part 3 extended.

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

- [VPP]: Protection Profile for Virtualization. Version 1.0 as of 2016-11-17; exact conformance.
- [SVPP]: Protection Profile for Virtualization Extended Package Server Virtualization. Version 1.0 as of 2016-11-17; exact conformance.

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

Copyright © 2022 IBM

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

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 [VPP] and extended packages have been copied.

Copyright © 2022 IBM

3.1.3 Threats countered by the TOE

T.DATA_LEAKAGE

It is a fundamental property of VMs that the domains encapsulated by different VMs remain separate unless data sharing is permitted by policy. For this reason, all Virtualization Systems shall support a policy that prohibits information transfer between VMs.

It shall be possible to configure VMs such that data cannot be moved between domains from VM to VM, or through virtual or physical network components under the control of the VS. When VMs are configured as such, it shall not be possible for data to leak between domains, neither by the express efforts of software or users of a VM, nor because of vulnerabilities or errors in the implementation of the VMM or other VS components.

If it is possible for data to leak between domains when prohibited by policy, then an adversary on one domain or network can obtain data from another domain. Such cross-domain data leakage can, for example, cause classified information, corporate proprietary information, or personally identifiable information to be made accessible to unauthorized entities.

T.UNAUTHORIZED_UPDATE

It is common for attackers to target outdated versions of software containing known flaws. This means it is extremely important to update Virtualization System software as soon as possible when updates are available. But the source of the updates and the updates themselves must be trusted. If an attacker can write their own update containing malicious code they can take control of the VS.

T.UNAUTHORIZED_MODIFICATION

System integrity is a core security objective for Virtualization Systems. To achieve system integrity, the integrity of each VMM component must be established and maintained. Malware running on the platform must not be able to undetectably modify Virtualization System components while the system is running or at rest. Likewise, malicious code running within a virtual machine must not be able to modify Virtualization System components.

T.USER_ERROR

If a Virtualization System is capable of simultaneously displaying VMs of different domains to the same user at the same time, there is always the chance that the user will become confused and unintentionally leak information between domains. This is especially likely if VMs belonging to different domains are indistinguishable. Malicious code may also attempt to interfere with the users ability to distinguish between domains. The VS must take measures to minimize the likelihood of such confusion.

T.3P_SOFTWARE

In some VS implementations, critical functions are by necessity performed by software not produced by the virtualization vendor. Such software may include Host Operating Systems and physical device drivers. Vulnerabilities in this software can be exploited by an adversary and result in VMM compromise. Where possible, the VS should mitigate the results of potential vulnerabilities or malicious content in third-party code.

Copyright © 2022 IBM

T.VMM_COMPROMISE

The Virtualization System is designed to provide the appearance of exclusivity to the VMs and is designed to separate or isolate their functions except where specifically shared. Failure of security mechanisms could lead to unauthorized intrusion into or modification of the VMM, or bypass of the VMM altogether. This must be prevented to avoid compromising the Virtualization System.

T.PLATFORM_COMPROMISE

The VS must be capable of protecting the platform from threats that originate within VMs and operational networks connected to the VS. The hosting of untrusted even malicious domains by the VS cannot be permitted to compromise the security and integrity of the platform on which the VS executes. If an attacker can access the underlying platform in a manner not controlled by the VMM, the attacker might be able to modify system firmware or software compromising both the Virtualization System and the underlying platform.

T.UNAUTHORIZED_ACCESS

Functions performed by the management layer include VM configuration, virtualized network configuration, allocation of physical resources, and reporting. Only certain authorized system users (administrators) are allowed to exercise management functions.

Virtualization Systems are often managed remotely over communication networks. Members of these networks can be both geographically and logically separated from each other, and pass through a variety of other systems which may be under the control of an adversary, and offer the opportunity for communications to be compromised. An adversary with access to an open management network could inject commands into the management infrastructure. This would provide an adversary with administrator privilege on the platform, and administrative control over the VMs and virtual network connections. The adversary could also gain access to the management network by hijacking the management network channel.

T.WEAK_CRYPTO

To the extent that VMs appear isolated within the Virtualization System, a threat of weak cryptography may arise if the VMM does not provide good entropy to support security-related features that depend on entropy to implement cryptographic algorithms. For example, a random number generator keeps an estimate of the number of bits of noise in the entropy pool. From this entropy pool random numbers are created. Good random numbers are essential to implementing strong cryptography. Cryptography implemented using poor random numbers can be defeated by a sophisticated adversary.

T.UNPATCHED_SOFTWARE

Vulnerabilities in outdated or unpatched software can be exploited by adversaries to compromise the Virtualization System or platform.

T.MISCONFIGURATION

The Virtualization System may be misconfigured, which could impact its functioning and security. This misconfiguration could be due to an administrative error or the use of faulty configuration data.

T.DENIAL_OF_SERVICE

A VM may block others from system resources (e.g., system memory, persistent storage, and processing time) via a resource exhaustion attack.

Copyright © 2022 IBM

3.2 Assumptions

3.2.1 Intended usage of the TOE

A.PLATFORM_INTEGRITY

The platform has not been compromised prior to installation of the Virtualization System.

A.COVERT_CHANNELS

If the TOE has covert storage or timing channels, then for all VMs executing on that TOE, it is assumed that relative to the IT assets to which they have access, those VMs will have assurance sufficient to outweigh the risk that they will violate the security policy of the TOE by using those covert channels.

3.2.2 Environment of use of the TOE

3.2.2.1 Physical

A.PHYSICAL

Physical security commensurate with the value of the TOE and the data it contains is assumed to be provided by the environment.

3.2.2.2 Personnel

A.TRUSTED_ADMIN

TOE Administrators are trusted to follow and apply all administrator guidance.

A.NON_MALICIOUS_USER

The user of the VS is not willfully negligent or hostile, and uses the VS in compliance with the applied enterprise security policy and guidance. At the same time, malicious applications could act as the user, so requirements which confine malicious applications are still in scope.

3.3 Organizational Security Policies

There are no organizational security policies defined for this ST.

Copyright © 2022 IBM

4 Security Objectives

4.1 Objectives for the TOE

O.VM_ISOLATION

VMs are the fundamental subject of the system. The VMM is responsible for applying the system security policy (SSP) to the VM and all resources. As basic functionality, the VMM must support a security policy that mandates no information transfer between VMs.

The VMM must support the necessary mechanisms to isolate the resources of all VMs. The VMM partitions a platform's physical resources for use by the supported virtual environments. Depending on the use case, a VM may require a completely isolated environment with exclusive access to system resources, or share some of its resources with other VMs. It must be possible to enforce a security policy that prohibits the transfer of data between VMs through shared devices. When the platform security policy allows the sharing of resources across VM boundaries, the VMM must ensure that all access to those resources is consistent with the policy. The VMM may delegate the responsibility for the mediation of sharing of particular resources to select Service VMs; however in doing so, it remains responsible for mediating access to the Service VMs, and each Service VM must mediate all access to any shared resource that has been delegated to it in accordance with the SSP.

Devices, whether virtual or physical, are resources requiring access control. The VMM must enforce access control in accordance to system security policy. Physical devices are platform devices with access mediated via the VMM per the O.VMM_Integrity objective. Virtual devices may include virtual storage devices and virtual network devices. Some of the access control restrictions must be enforced internal to Service VMs, as may be the case for isolating virtual networks. VMMs may also expose purely virtual interfaces. These are VMM specific, and while they are not analogous to a physical device, they are also subject to access control.

The VMM must support the mechanisms to isolate all resources associated with virtual networks and to limit a VM's access to only those virtual networks for which it has been configured. The VMM must also support the mechanisms to control the configurations of virtual networks according to the SSP.

O.VMM_INTEGRITY

Integrity is a core security objective for Virtualization Systems. To achieve system integrity, the integrity of each VMM component must be established and maintained. This objective concerns only the integrity of the Virtualization Systemnot the integrity of software running inside of Guest VMs or of the physical platform. The overall objective is to ensure the integrity of critical components of a Virtualization System.

Initial integrity of a VS can be established through mechanisms such as a digitally signed installation or update package, or through integrity measurements made at launch. Integrity is maintained in a running system by careful protection of the VMM from untrusted users and software. For example, it must not be possible for software running within a Guest VM to exploit a vulnerability in a device or hypercall interface and gain control of the VMM. The vendor must release patches for vulnerabilities as soon as practicable after discovery.

Copyright © 2022 IBM

O.PLATFORM_INTEGRITY

The integrity of the VMM depends on the integrity of the hardware and software on which the VMM relies. Although the VS does not have complete control over the integrity of the platform, the VS should as much as possible try to ensure that no users or software hosted by the VS is capable of undermining the integrity of the platform.

O.DOMAIN_INTEGRITY

While the VS is not responsible for the contents or correct functioning of software that runs within Guest VMs, it is responsible for ensuring that the correct functioning of the software within a Guest VM is not interfered with by other VMs.

O.MANAGEMENT_ACCESS

VMM management functions include VM configuration, virtualized network configuration, allocation of physical resources, and reporting. Only certain authorized system users (administrators) are allowed to exercise management functions.

Because of the privileges exercised by the VMM management functions, it must not be possible for the VMMs management components to be compromised without administrator notification. This means that unauthorized users cannot be permitted access to the management functions, and the management components must not be interfered with by Guest VMs or unprivileged users on other networksincluding operational networks connected to the TOE.

VMMs include a set of management functions that collectively allow administrators to configure and manage the VMM, as well as configure Guest VMs. These management functions are specific to the virtualization system, distinct from any other management functions that might exist for the internal management of any given Guest VM. These VMM management functions are privileged, with the security of the entire system relying on their proper use. The VMM management functions can be classified into different categories and the policy for their use and the impact to security may vary accordingly.

The management functions might be distributed throughout the VMM (within the VMM and Service VMs). The VMM must support the necessary mechanisms to enable the control of all management functions according to the system security policy. When a management function is distributed among multiple Service VMs, the VMs must be protected using the security mechanisms of the Hypervisor and any Service VMs involved to ensure that the intent of the system security policy is not compromised. Additionally, since hypercalls permit Guest VMs to invoke the Hypervisor, and often allow the passing of data to the Hypervisor, it is important that the hypercall interface is well-guarded and that all parameters be validated.

The VMM maintains configuration data for every VM on the system. This configuration data, whether of Service or Guest VMs, must be protected. The mechanisms used to establish, modify and verify configuration data are part of the VS management functions and must be protected as such. The proper internal configuration of Service VMs that provide critical security functions can also greatly impact VS security. These configurations must also be protected. Internal configuration of Guest VMs should not impact overall VS security. The overall goal is to ensure that the VMM, including the environments internal to Service VMs, is properly configured and that all Guest VM configurations are maintained consistent with the system security policy throughout their lifecycle.

Copyright © 2022 IBM

Virtualization Systems are often managed remotely. For example, an administrator can remotely update virtualization software, start and shut down VMs, and manage virtualized network connections. If a console is required, it could be run on a separate machine or it could itself run in a VM. When performing remote management, an administrator must communicate with a privileged management agent over a network. Communications with the management infrastructure must be protected from Guest VMs and operational networks.

O.PATCHED_SOFTWARE

The Virtualization System must be updated and patched when needed in order to prevent the potential compromise of the VMM, as well as the networks and VMs that it hosts. Identifying and applying needed updates must be a normal part of the operating procedure to ensure that patches are applied in a timely and thorough manner. In order to facilitate this, the VS must support standards and protocols that help enhance the manageability of the VS as an IT product, enabling it to be integrated as part of a manageable network (e.g., reporting current patch level and patchability).

O.VM_ENTROPY

VMs must have access to good entropy sources to support security-related features that implement cryptographic algorithms. For example, in order to function as members of operational networks, VMs must be able to communicate securely with other network entitieswhether virtual or physical. They must therefore have access to sources of good entropy to support that secure communication.

O.AUDIT

The purpose of audit is to capture and protect data about what happens on a system so that it can later be examined to determine what has happened in the past.

O.CORRECTLY_APPLIED_CONFIGURATION

The TOE must not apply configurations that violate the current security policy.

The TOE must correctly apply configurations and policies to newly created Guest VMs, as well as to existing Guest VMs when applicable configuration or policy changes are made. All changes to configuration and to policy must conform to the existing security policy. Similarly, changes made to the configuration of the TOE itself must not violate the existing security policy.

O.RESOURCE_ALLOCATION

The TOE will provide mechanisms that enforce constraints on the allocation of system resources in accordance with existing security policy.

4.2 Objectives for the Operational Environment

OE.CONFIG

TOE administrators will configure the Virtualization System correctly to create the intended security policy.

OE.PHYSICAL

Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.

Copyright © 2022 IBM

OE.TRUSTED_ADMIN

TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

OE.COVERT_CHANNELS

If the TOE has covert storage or timing channels, then for all VMs executing on that TOE, it is assumed that those VMs will have sufficient assurance relative to the IT assets to which they have access, to outweigh the risk that they will violate the security policy of the TOE by using those covert channels.

OE.NON_MALICIOUS_USER

Users are trusted to be not willfully negligent or hostile and use the VS in compliance with the applied enterprise security policy and guidance.

4.3 Security Objectives Rationale

4.3.1 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.VM_ISOLATION	T.DATA_LEAKAGE T.USER_ERROR T.VMM_COMPROMISE
O.VMM_INTEGRITY	T.UNAUTHORIZED_UPDATE T.UNAUTHORIZED_MODIFICATION T.3P_SOFTWARE T.VMM_COMPROMISE
O.PLATFORM_INTEGRITY	T.PLATFORM_COMPROMISE
O.DOMAIN_INTEGRITY	T.DATA_LEAKAGE
O.MANAGEMENT_ACCESS	T.UNAUTHORIZED_ACCESS T.MISCONFIGURATION
O.PATCHED_SOFTWARE	T.UNPATCHED_SOFTWARE
O.VM_ENTROPY	T.WEAK_CRYPTO
O.AUDIT	T.UNAUTHORIZED_MODIFICATION
O.CORRECTLY_APPLIED_CONFIGURATION	T.MISCONFIGURATION
O.RESOURCE_ALLOCATION	T.DENIAL_OF_SERVICE

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.

Copyright © 2022 IBM

Objective	Assumptions / Threats / OSPs
OE.CONFIG	A.TRUSTED_ADMIN A.NON_MALICIOUS_USER
OE.PHYSICAL	A.PHYSICAL
OE.TRUSTED_ADMIN	A.TRUSTED_ADMIN
OE.COVERT_CHANNELS	A.COVERT_CHANNELS
OE.NON_MALICIOUS_USER	A.NON_MALICIOUS_USER

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

4.3.2 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.DATA_LEAKAGE	Logical separation of VMs and enforcement of domain integrity prevent unauthorized transmission of data from one VM to another.
T.UNAUTHORIZED_UPDATE	System integrity prevents the TOE from installing a software patch containing unknown and potentially malicious code.
T.UNAUTHORIZED_MODIFICATION	Enforcement of VMM integrity prevents the bypass of enforcement mechanisms and auditing ensures that abuse of legitimate authority can be detected.
T.USER_ERROR	Isolation of VMs includes clear attribution of those VMs to their respective domains which reduces the likelihood that a user inadvertently inputs or transfers data meant for one VM into another.
T.3P_SOFTWARE	The VMM integrity mechanisms include environment-based vulnerability mitigation and potentially support for introspection and device driver isolation, all of which reduce the likelihood that any vulnerabilities in third-party software can be used to exploit the TOE.
T.VMM_COMPROMISE	Maintaining the integrity of the VMM and ensuring that VMs execute in isolated domains mitigate the risk that the VMM can be compromised or bypassed.
T.PLATFORM_COMPROMISE	Platform integrity mechanisms used by the TOE reduce the risk that an attacker can break out of a VM and affect the platform on which the VS is running.

Copyright © 2022 IBM

Threat	Rationale for security objectives
T.UNAUTHORIZED_ACCESS	Ensuring that TSF management functions cannot be executed without authorization prevents untrusted subjects from modifying the behavior of the TOE in an unanticipated manner.
T.WEAK_CRYPTO	Acquisition of good entropy is necessary to support the TOEs security-related cryptographic algorithms.
T.UNPATCHED_SOFTWARE	The ability to patch the TOE software ensures that protections against vulnerabilities can be applied as they become available.
T.MISCONFIGURATION	Mechanisms to prevent the application of configurations that violate the current security policy help prevent misconfigurations.
T.DENIAL_OF_SERVICE	The ability of the TSF to ensure the proper allocation of resources makes denial of service attacks more difficult.

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.PLATFORM_INTEGRITY	If the underlying platform has not been compromised prior to installation of the TOE, its integrity can be assumed to be intact.
A.COVERT_CHANNELS	It is expected that any data contained within VMs is commensurate with the security provided by the TOE, which includes any vulnerabilities due to the potential presence of covert storage and/or timing channels.
A.PHYSICAL	If the TOE is deployed in a location that has appropriate physical safeguards, it can be assumed to be physically secure.
A.TRUSTED_ADMIN	Providing guidance to administrators and ensuring that individuals are properly trained and vetted before being given administrative responsibilities will ensure that they are trusted.
A.NON_MALICIOUS_USER	If the organization properly vets and trains users, it is expected that they will be non-malicious. If the TOE is administered by a non-malicious and non-negligent user, the expected result is that the TOE will be configured in a correct and secure manner.

Table 4: Sufficiency of objectives holding assumptions

Copyright © 2022 IBM

5 Extended Components Definition

This ST defines the following extended components to match the virtualization SFRs defined in [VPP] (all except FMT_MOF_EXT.1) and [SVPP] (FMT_MOF_EXT.1 only). Most of these SFRs are not from CC Part 2, but were created specifically to address virtualization functionality. The extended components are taken directly from the PP and are not re-iterated here.

5.1 Class ALC: Life Cycle Support

This extended assurance component is derived from [VPP].

5.1.1 Timely Security Updates (ALC_TSU_EXT.1)

Objectives

This component requires the TOE developer, in conjunction with any other necessary parties, to provide information as to how the Virtualization System is updated to address security issues in a timely manner. The documentation describes the process of providing updates to the public from the time a security flaw is reported/discovered, to the time an update is released. This description includes the parties involved (e.g., the developer, hardware vendors) and the steps that are performed (e.g., developer testing), including worst case time periods, before an update is made available to the public.

Overview

This extended assurance component is derived from [VPP].

ALC_TSU_EXT.1.1 - Timely Security Updates

Dependencies: No dependencies.

Developer action elements:

ALC_TSU_EXT.1.1D The developer shall provide a description in the TSS of how timely security updates are made to the TOE.

Content and presentation elements:

ALC_TSU_EXT.1.1C The description shall include the process for creating and deploying security updates for the TOE software/firmware.

ALC_TSU_EXT.1.2C The description shall express the time window as the length of time, in days, between public disclosure of a vulnerability and the public availability of security updates to the TOE.

ALC_TSU_EXT.1.3C The description shall include the mechanisms publicly available for reporting security issues pertaining to the TOE.

Evaluator action elements:

ALC_TSU_EXT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Copyright © 2022 IBM

6 Security Requirements

6.1 TOE Security Functional Requirements

The table below summarizes the SFRs for the TOE and the operations performed on the components according to CC part 1. Operations in the SFRs use the following convention:

- Iterations (Iter.) are identified by appending a suffix to the original SFR.
- Refinements (Ref.) added to the text are shown in *italic text*, deletions are shown as ~~strikethrough text~~.
- Assignments (Ass.) are shown in **bold text**.
- Selections (Sel.) are shown in **bold text**.

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
FAU - Security audit	FAU_GEN.1 Audit data generation		VPP	No	Yes	No	Yes
	FAU_SAR.1 Audit review		VPP	No	No	No	No
	FAU_STG.1 Protected audit trail storage		VPP	No	No	No	No
	FAU_STG_EXT.1 Off-Loading of Audit Data		VPP	No	No	Yes	Yes
FCS - Cryptographic support	FCS_CKM.1 Cryptographic Key Generation		VPP	No	No	No	Yes
	FCS_CKM.2 Cryptographic Key Establishment		VPP	No	No	No	Yes
	FCS_CKM_EXT.4 Cryptographic Key Destruction		VPP	No	No	No	No
	FCS_COP.1(1) Cryptographic operation (AES Data Encryption / Decryption)	FCS_COP.1	VPP	Yes	No	No	Yes
	FCS_COP.1(2) Cryptographic operation (Hashing)	FCS_COP.1	VPP	Yes	No	No	Yes
	FCS_COP.1(3) Cryptographic operation (Signature Algorithms)	FCS_COP.1	VPP	Yes	No	No	Yes
	FCS_COP.1(4) Cryptographic operation (Keyed Hash Algorithms)	FCS_COP.1	VPP	Yes	No	No	Yes
	FCS_ENT_EXT.1 Entropy for Virtual Machines		VPP	No	No	No	Yes
	FCS_RBG_EXT.1 Cryptographic Operation (Random Bit Generation)		VPP	No	No	No	Yes

Copyright © 2022 IBM

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FCS_TLSS_EXT.1 TLS Server Protocol		VPP	No	No	No	Yes
FDP - User data protection	FDP_HBI_EXT.1 Hardware-Based Isolation Mechanisms		VPP	No	No	Yes	Yes
	FDP_PPR_EXT.1 Physical Platform Resource Controls		VPP	No	No	Yes	Yes
	FDP_RIP_EXT.1 Residual Information in Memory		VPP	No	No	No	No
	FDP_RIP_EXT.2 Residual Information on Disk		VPP	No	No	No	No
	FDP_VMS_EXT.1 VM Separation		VPP	No	No	Yes	Yes
	FDP_VNC_EXT.1 Virtual Networking Components		VPP	No	No	No	No
FIA - Identification and authentication	FIA_AFL_EXT.1 Authentication Failure Handling		VPP	No	No	Yes	Yes
	FIA_PMG_EXT.1 Password Management		VPP	No	No	No	Yes
	FIA_UAU.5 Multiple authentication mechanisms		VPP	No	No	Yes	Yes
	FIA_UIA_EXT.1 Administrator Identification and Authentication		VPP	No	No	No	No
	FIA_X509_EXT.1 X.509 Certificate Validation		VPP	No	No	No	Yes
	FIA_X509_EXT.2 X.509 Certificate Authentication		VPP	No	No	No	Yes
FMT - Security management	FMT_MOF_EXT.1 Management of Security Functions Behavior		SVPP	No	Yes	No	Yes
	FMT_MSA_EXT.1 Default Data Sharing Configuration		VPP	No	No	Yes	Yes
	FMT_SMO_EXT.1 Separation of Management and Operational Networks		VPP	No	No	No	Yes
FPT - Protection of the TSF	FPT_DVD_EXT.1 Non-Existence of Disconnected Virtual Devices		VPP	No	No	No	No
	FPT_EEM_EXT.1 Execution Environment Mitigations		VPP	No	No	No	Yes

Copyright © 2022 IBM

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FPT_HAS_EXT.1 Hardware Assists		VPP	No	No	Yes	No
	FPT_HCL_EXT.1 Hypercall Controls		VPP	No	No	Yes	No
	FPT_RDM_EXT.1 Removable Devices and Media		VPP	No	No	Yes	Yes
	FPT_TUD_EXT.1 Trusted Updates to the Virtualization System		VPP	No	No	No	Yes
	FPT_VDP_EXT.1 Virtual Device Parameters		VPP	No	No	No	No
	FPT_VIV_EXT.1 VMM Isolation from VMs		VPP	No	No	No	No
FTA - TOE access	FTA_TAB.1 Default TOE access banners		VPP	No	No	No	No
FTP - Trusted path/channels	FTP_ITC_EXT.1 Trusted Channel Communications		VPP	No	No	Yes	Yes
	FTP_UIF_EXT.1 User Interface: I/O Focus		VPP	No	No	No	No
	FTP_UIF_EXT.2 User Interface: Identification of VM		VPP	No	No	No	No

Table 5: SFRs for the TOE

6.1.1 Security audit (FAU)

6.1.1.1 Audit data generation (FAU_GEN.1)

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

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

b) All administrative actions;

c) Specifically defined auditable events in Table ~~1~~ 6;

d) **additional information defined in Table 4 *as stated in the PP***

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

a) Date and time of the event;

b) Type of event;

c) Subject and object identity (if applicable);

d) The outcome (success or failure) of the event;

e) Additional information defined in Table ~~1~~ 6; and

f) **additional information defined in Table 4 *as stated in the PP***

Copyright © 2022 IBM

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_SAR.1	None.	None.
FAU_STG.1	None.	None.
FAU_STG_EXT.1	Failure of audit data capture due to lack of disk space or pre-defined limit. On failure of logging function, capture record of failure and record upon restart of logging function.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM_EXT.4	None.	None.
FCS_COP.1(1)	None.	None.
FCS_COP.1(2)	None.	None.
FCS_COP.1(3)	None.	None.
FCS_COP.1(4)	None.	None.
FCS_ENT_EXT.1	None.	None.
FCS_RBG_EXT.1	Failure of the randomization process.	No additional information.
FCS_TLSS_EXT.1	Failure to establish a TLS Session. Establishment/Termination of a TLS session.	Reason for failure. Non-TOE endpoint of connection (IP address).
FDP_HBI_EXT.1	None.	None.
FDP_PPR_EXT.1	Successful and failed VM connections to physical devices where connection is governed by configurable policy. Security policy violations.	VM and physical device identifiers. Identifier for the security policy that was violated.
FDP_RIP_EXT.1	None.	None.
FDP_RIP_EXT.2	None.	None.
FDP_VMS_EXT.1	None.	None.
FDP_VNC_EXT.1	Successful and failed attempts to connect VMs to virtual and physical networking components. Security policy violations. Administrator configuration of inter-VM communications channels between VMs.	VM and virtual or physical networking component identifiers. Identifier for the security policy that was violated.
FIA_AFL_EXT.1	None.	None.
FIA_PMG_EXT.1	None.	None.
FIA_UAU.5	None.	None.

Copyright © 2022 IBM

Requirement	Auditable Events	Additional Audit Record Contents
FIA_UIA_EXT.1	Administrator session start time and end time	None
FIA_X509_EXT.1	Failure to validate a certificate.	Reason for failure.
FIA_X509_EXT.2	None.	None.
FMT_MOF_EXT.1	None.	None.
FMT_MSA_EXT.1	None.	None.
FMT_SMO_EXT.1	None.	None.
FPT_DVD_EXT.1	None.	None.
FPT_EEM_EXT.1	None.	None.
FPT_HAS_EXT.1	None.	None.
FPT_HCL_EXT.1	Attempts to access disabled hypercall interfaces. Security policy violations.	Interface for which access was attempted. Identifier for the security policy that was violated.
FPT_RDM_EXT.1	Connection/disconnection of removable media or device to/from a VM. Ejection/insertion of removable media or device from/to an already connected VM.	VM Identifier, Removable media/device identifier, event description or identifier (connect/disconnect, ejection/insertion, etc.)
FPT_TUD_EXT.1	None.	None.
FPT_VDP_EXT.1	None.	None.
FPT_VIV_EXT.1	None.	None.
FTA_TAB.1	None.	None.
FTP_ITC_EXT.1	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions.	User ID and remote source (IP Address) if feasible.
FTP_UIF_EXT.1	None.	None.
FTP_UIF_EXT.2	None.	None.

Table 6: Auditable Events

6.1.1.2 Audit review (FAU_SAR.1)

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

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

Application Note: *The "administrator" referenced in the SFR relates to the RACF auditor role.*

Copyright © 2022 IBM

6.1.1.3 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 stored audit records in the audit trail.

6.1.1.4 Off-Loading of Audit Data (FAU_STG_EXT.1)

FAU_STG_EXT.1.1 The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel as specified in FTP_ITC_EXT.1.

FAU_STG_EXT.1.2 The TSF shall **prevent any further actions that may generate audit records** when the local storage space for audit data is full.

6.1.2 Cryptographic support (FCS)

6.1.2.1 Cryptographic Key Generation (FCS_CKM.1)

FCS_CKM.1.1 The TSF shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm

1. **RSA schemes using cryptographic key sizes [2048-bit or greater] that meet the following: [FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3];**
2. **FFC schemes using cryptographic key sizes [2048-bit or greater] that meet the following: [FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.1]].**

6.1.2.2 Cryptographic Key Establishment (FCS_CKM.2)

FCS_CKM.2.1 The TSF shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method:

1. **RSA-based key establishment schemes that meets the following: NIST Special Publication 800-56B, “Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography”;**
2. **Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”].**

6.1.2.3 Cryptographic Key Destruction (FCS_CKM_EXT.4)

FCS_CKM_EXT.4.1 The TSF shall cause disused cryptographic keys in volatile memory to be destroyed or rendered unrecoverable.

FCS_CKM_EXT.4.2 The TSF shall cause disused cryptographic keys in non-volatile storage to be destroyed or rendered unrecoverable.

Copyright © 2022 IBM

6.1.2.4 Cryptographic operation (AES Data Encryption / Decryption) (FCS_COP.1(1))

FCS_COP.1.1 The TSF shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm

1. **AES-CBC (as defined in FIPS PUB 197, and NIST SP 800-38A) mode,**

and cryptographic key sizes **128-bits, 256-bits** .

6.1.2.5 Cryptographic operation (Hashing) (FCS_COP.1(2))

FCS_COP.1.1 The TSF shall perform [cryptographic hashing] in accordance with a specified cryptographic algorithm **SHA-1, SHA-256, SHA-384, SHA-512** and message digest sizes **160, 256, 384, 512 bits** that meet the following: [FIPS PUB 180-4, “Secure Hash Standard”].

6.1.2.6 Cryptographic operation (Signature Algorithms) (FCS_COP.1(3))

FCS_COP.1.1 The TSF shall perform [cryptographic signature services (generation and verification)] in accordance with a specified cryptographic algorithm

1. **RSA schemes using cryptographic key sizes [2048-bit or greater] that meet the following: [FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 4]**

6.1.2.7 Cryptographic operation (Keyed Hash Algorithms) (FCS_COP.1(4))

FCS_COP.1.1 The TSF shall perform [keyed-hash message authentication] in accordance with a specified cryptographic algorithm **HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512** and cryptographic key sizes **160, 256, 384, 512 bits** that meet the following: [FIPS PUB 198-1, “The Keyed-Hash Message Authentication Code”, FIPS PUB 180-4, “Secure Hash Standard”].

6.1.2.8 Entropy for Virtual Machines (FCS_ENT_EXT.1)

FCS_ENT_EXT.1.1 The TSF shall provide a mechanism to make available to VMs entropy that meets FCS_RBG_EXT.1 through **passthrough access to hardware entropy source**.

FCS_ENT_EXT.1.2 The TSF shall provide independent entropy across multiple VMs.

Application Note: *The CPACF facility of the CPU provides the noise source.*

6.1.2.9 Cryptographic Operation (Random Bit Generation) (FCS_RBG_EXT.1)

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with NIST Special Publication 800-90A using **Hash_DRBG (any)**.

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by an entropy source that accumulates entropy from **a software-based noise source** with a minimum of **256 bits** of entropy at least equal to the greatest security strength according to NIST SP 800-57, of the keys and hashes that it will generate.

Copyright © 2022 IBM

6.1.2.10 TLS Server Protocol (FCS_TLSS_EXT.1)

FCS_TLSS_EXT.1.1 The TSF shall implement **TLS 1.2 (RFC 5246)** supporting the following ciphersuites:

1. **TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268**
2. **TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268**
3. **TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246**
4. **TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246**
5. **TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246**
6. **TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246**

.

FCS_TLSS_EXT.1.2 The TSF shall deny connections from clients requesting SSL 1.0, SSL 2.0, SSL 3.0, TLS 1.0, and **TLS 1.1**

FCS_TLSS_EXT.1.3 The TSF shall **perform RSA key establishment with key size 2048 bits, generate DiffieHellman parameters of size 2048 bits**

6.1.3 User data protection (FDP)

6.1.3.1 Hardware-Based Isolation Mechanisms (FDP_HBI_EXT.1)

FDP_HBI_EXT.1.1 The TSF shall use

1. **SIE instruction**
2. **virtualization support in the I/O controller**

to constrain a Guest VM's direct access to the following physical devices:

1. **CPU**
2. **RAM**
3. **I/O devices**

.

6.1.3.2 Physical Platform Resource Controls (FDP_PPR_EXT.1)

FDP_PPR_EXT.1.1 The TSF shall allow an authorized administrator to control Guest VM access to the following physical platform resources:

a) I/O devices

.

Application Note:

Please see the TSS for a description of the TSF regarding the memory management.

FDP_PPR_EXT.1.2 The TSF shall explicitly deny all Guest VMs access to the following physical platform resources: **no physical platform resources** .

FDP_PPR_EXT.1.3 The TSF shall explicitly allow all Guest VMs access to the following physical platform resources: **no physical platform resources** .

Copyright © 2022 IBM

6.1.3.3 Residual Information in Memory (FDP_RIP_EXT.1)

FDP_RIP_EXT.1.1 The TSF shall ensure that any previous information content of physical memory is cleared prior to allocation to a Guest VM.

6.1.3.4 Residual Information on Disk (FDP_RIP_EXT.2)

FDP_RIP_EXT.2.1 The TSF shall ensure that any previous information content of physical disk storage is cleared prior to allocation to a Guest VM.

6.1.3.5 VM Separation (FDP_VMS_EXT.1)

FDP_VMS_EXT.1.1 The VS shall provide the following mechanisms for transferring data between Guest VMs: **virtual networking , Virtual point-to-point communication paths (IUCV, APPC, virtual CTC)**.

FDP_VMS_EXT.1.2 The TSF shall allow Administrators to configure these mechanisms to **enable , disable** the transfer of data between Guest VMs.

FDP_VMS_EXT.1.3 The VS shall ensure that no Guest VM is able to read or transfer data to or from another Guest VM except through the mechanisms listed in FDP_VMS_EXT.1.1.

6.1.3.6 Virtual Networking Components (FDP_VNC_EXT.1)

FDP_VNC_EXT.1.1 The TSF shall allow Administrators to configure virtual networking components to connect VMs to each other, and to physical networks.

FDP_VNC_EXT.1.2 The TSF shall ensure that network traffic visible to a Guest VM on a virtual network—or virtual segment of a physical network—is visible only to Guest VMs configured to be on that virtual network or segment.

6.1.4 Identification and authentication (FIA)

6.1.4.1 Authentication Failure Handling (FIA_AFL_EXT.1)

FIA_AFL_EXT.1.1 The TSF shall detect when **an administrator configurable positive integer within a positive integer space** unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a **password**.

FIA_AFL_EXT.1.2 When the defined number of unsuccessful authentication attempts has been met, the TSF shall: **prevent the offending Administrator from successfully establishing remote session using any authentication method that involves a password or PIN until the account has been unlocked by the administrator managing users is taken by an Administrator** .

6.1.4.2 Password Management (FIA_PMG_EXT.1)

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for administrative passwords:

a) Passwords shall be able to be composed of any combination of upper and lower case characters, digits, and the following special characters: **!, @, #, $, %, ^, &, *, (,)**

b) Minimum password length shall be configurable;

Copyright © 2022 IBM

c) Passwords of at least 15 characters in length shall be supported.

6.1.4.3 Multiple authentication mechanisms (FIA_UAU.5)

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

a) local, directory-based authentication based on username and password

to support Administrator authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the **following rule: password-based authentication is used with the authentication backend configured by the administrator**.

6.1.4.4 Administrator Identification and Authentication (FIA_UIA_EXT.1)

FIA_UIA_EXT.1.1 The TSF shall require Administrators to be successfully identified and authenticated using one of the methods in FIA_UAU.5 before allowing any TSF-mediated management function to be performed by that Administrator.

6.1.4.5 X.509 Certificate Validation (FIA_X509_EXT.1)

FIA_X509_EXT.1.1 The TSF shall validate certificates in accordance with the following rules:

a) RFC 5280 certificate validation and certificate path validation.

b) The certificate path must terminate with a trusted certificate.

c) The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.

d) The TSF shall validate the revocation status of the certificate using **the Online Certificate Status Protocol (OCSP) as specified in RFC 2560, a Certificate Revocation List (CRL) as specified in RFC 5759**.

e) The TSF shall validate the extendedKeyUsage field according to the following rules:

1. Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
2. Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
3. Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
4. OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

FIA_X509_EXT.1.2 The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

Copyright © 2022 IBM

6.1.4.6 X.509 Certificate Authentication (FIA_X509_EXT.2)

FIA_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for **TLS**, and **no additional uses**.

FIA_X509_EXT.2.2 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall **not accept the certificate**.

6.1.5 Security management (FMT)

6.1.5.1 Management of Security Functions Behavior (FMT_MOF_EXT.1)

FMT_MOF_EXT.1.1 The TSF shall be capable of supporting **local, remote** administration.

FMT_MOF_EXT.1.2 The TSF shall be capable of performing the following management functions, controlled by an Administrator or User as shown in Table ~~1~~ 7, based on the following key:
X = Mandatory (TOE must provide that function to that role)
O = Optional (TOE may or may not provide that function to that role)
N = Not Permitted (TOE must not provide that function to that role)
S = Selection-Based (TOE must provide that function to that role if the TOE claims a particular selection-based SFR)

Number	Function	Administrator	User	Notes
1	Ability to update the Virtualization System	X	N	See FPT_TUD_EXT.1
2	Ability to configure Administrator password policy as defined in FIA_PMG_EXT.1	X	N	Must be selected if ST includes FIA_PMG_EXT.1.
3	Ability to create, configure and delete VMs	X	N	
4	Ability to set default initial VM configurations	X	N	
5	Ability to configure virtual networks including VM	X	N	See FDP_VNC_EXT.1
6	Ability to configure and manage the audit system and audit data	X	N	
7	Ability to configure VM access to physical devices	X	N	
8	Ability to configure inter-VM data sharing	X	N	See FDP_VMS_EXT.1 and FMT_MSA_EXT.1
9	Ability to enable/disable VM access to Hypercall functions	X	N	See FPT_HCL_EXT.1
10	Ability to configure removable media policy	X	N	See FPT_RDM_EXT.1

Copyright © 2022 IBM

Number	Function	Administrator	User	Notes
11	Ability to configure the cryptographic functionality	X	N	
12	Ability to change default authorization factors	X	N	
13	Ability to enable/disable screen lock	N	N	
14	Ability to configure screen lock inactivity timeout	N	N	
15	Ability to configure remote connection inactivity timeout	X	N	
16	Ability to configure lockout policy for unsuccessful authentication attempts through limiting number of attempts during a time period	X	N	See FIA_AFL_EXT.1
17	Ability to configure name/address of directory server to bind with	X	N	
18	Ability to configure name/address of audit/logging server to which to send audit/logging records	X	N	
19	Ability to configure name/address of network time server	X	N	
20	Ability to configure banner	X	N	See FTA_TAB.1
21	Ability to connect/disconnect removable devices to/from a VM	X	N	
22	Ability to start a VM	X	N	
23	Ability to stop/halt a VM	X	N	
24	Ability to checkpoint a VM	N	N	
25	Ability to suspend a VM	N	N	
26	Ability to resume a VM	N	N	

Table 7: Server Virtualization Management Functions

Copyright © 2022 IBM

6.1.5.2 Default Data Sharing Configuration (FMT_MSA_EXT.1)

FMT_MSA_EXT.1.1 The TSF shall by default enforce a policy prohibiting sharing of data between Guest VMs using **virtual networking, Virtual point-to-point communication paths (IUCV, APPC, virtual CTC)**.

FMT_MSA_EXT.1.2 The TSF shall allow Administrators to specify alternative initial configuration values to override the default values when a Guest VM is created.

6.1.5.3 Separation of Management and Operational Networks (FMT_SMO_EXT.1)

FMT_SMO_EXT.1.1 The TSF shall support the configuration of separate management and operational networks through **physical means, logical means**.

6.1.6 Protection of the TSF (FPT)

6.1.6.1 Non-Existence of Disconnected Virtual Devices (FPT_DVD_EXT.1)

FPT_DVD_EXT.1.1 The TSF shall limit a Guest VM's access to virtual devices to those that are present in the VM's current virtual hardware configuration.

6.1.6.2 Execution Environment Mitigations (FPT_EEM_EXT.1)

FPT_EEM_EXT.1.1 The TSF shall take advantage of execution environment-based vulnerability mitigation mechanisms supported by the Platform such as:

1. **No mechanisms**

6.1.6.3 Hardware Assists (FPT_HAS_EXT.1)

FPT_HAS_EXT.1.1 The VMM shall use **SIE instruction, I/O controller virtualization** to reduce or eliminate the need for binary translation.

FPT_HAS_EXT.1.2 The VMM shall use **ART (access register translation), DAT (dynamic address translation)** to reduce or eliminate the need for shadow page tables.

6.1.6.4 Hypercall Controls (FPT_HCL_EXT.1)

FPT_HCL_EXT.1.1 The TSF shall provide a Hypercall interface for Guest VMs to use to invoke functionality provided by the VMM.

FPT_HCL_EXT.1.2 The TSF shall allow administrators to configure any VM’s Hypercall interface to disable access to individual functions, all functions, or groups of functions.

FPT_HCL_EXT.1.3 The TSF shall permit exceptions to the configuration of the following Hypercall interface functions: **none**.

FPT_HCL_EXT.1.4 The TSF shall validate the parameters passed to the hypercall interface prior to execution of the VMM functionality exposed by that interface.

Copyright © 2022 IBM

6.1.6.5 Removable Devices and Media (FPT_RDM_EXT.1)

FPT_RDM_EXT.1.1 The TSF shall implement controls for handling the transfer of virtual and physical removable media and virtual and physical removable media devices between information domains.

FPT_RDM_EXT.1.2 The TSF shall enforce the following rules when **tape** are switched between information domains, then

1. **the Administrator has granted explicit access for the media or device to be connected to the receiving domain,**

.

6.1.6.6 Trusted Updates to the Virtualization System (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1 The TSF shall provide administrators the ability to query the currently executed version of the TOE firmware/software as well as the most recently installed version of the TOE firmware/software.

FPT_TUD_EXT.1.2 The TSF shall provide administrators the ability to manually initiate updates to TOE firmware/software and **no other update mechanism**.

FPT_TUD_EXT.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a **published hash** prior to installing those updates.

6.1.6.7 Virtual Device Parameters (FPT_VDP_EXT.1)

FPT_VDP_EXT.1.1 The TSF shall provide interfaces for virtual devices implemented by the VMM as part of the virtual hardware abstraction.

FPT_VDP_EXT.1.2 The TSF shall validate the parameters passed to the virtual device interface prior to execution of the VMM functionality exposed by those interfaces.

6.1.6.8 VMM Isolation from VMs (FPT_VIV_EXT.1)

FPT_VIV_EXT.1.1 The TSF must ensure that software running in a VM is not able to degrade or disrupt the functioning of other VMs, the VMM, or the Platform.

FPT_VIV_EXT.1.2 The TSF must ensure that a Guest VM is unable to invoke platform code that runs at a privilege level equal to or exceeding that of the VMM without involvement of the VMM.

6.1.7 TOE access (FTA)

6.1.7.1 Default TOE access banners (FTA_TAB.1)

FTA_TAB.1.1 Before establishing an administrative user session the TSF shall display a Security Administrator-specified advisory notice and consent warning message regarding use of the TOE.

Copyright © 2022 IBM

6.1.8 Trusted path/channels (FTP)

6.1.8.1 Trusted Channel Communications (FTP_ITC_EXT.1)

FTP_ITC_EXT.1.1 The TSF shall use

1. TLS as conforming to FCS_TLSS_EXT.1

to provide a trusted communication channel between itself and:

a) audit servers (as required by FAU_STG_EXT.1), and

b) **i. remote administrators,**

ii. separation of management and operational networks (if selected in FMT_SMO_EXT.1),

that is logically distinct from other communication paths and provides assured identification of its endpoints and protection of the communicated data from disclosure and detection of modification of the communicated data.

6.1.8.2 User Interface: I/O Focus (FTP_UIF_EXT.1)

FTP_UIF_EXT.1.1 The TSF shall indicate to users which VM, if any, has the current input focus.

6.1.8.3 User Interface: Identification of VM (FTP_UIF_EXT.2)

FTP_UIF_EXT.2.1 The TSF shall support the unique identification of a VM's output display to users.

6.2 Security Functional Requirements Rationale

6.2.1 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.AUDIT
FAU_SAR.1	O.MANAGEMENT_ACCESS
FAU_STG.1	O.AUDIT
FAU_STG_EXT.1	O.AUDIT
FCS_CKM.1	O.MANAGEMENT_ACCESS
FCS_CKM.2	O.MANAGEMENT_ACCESS
FCS_CKM_EXT.4	O.MANAGEMENT_ACCESS
FCS_COP.1(1)	O.MANAGEMENT_ACCESS
FCS_COP.1(2)	O.MANAGEMENT_ACCESS
FCS_COP.1(3)	O.MANAGEMENT_ACCESS

Copyright © 2022 IBM

Security functional requirements	Objectives
FCS_COP.1(4)	O.MANAGEMENT_ACCESS
FCS_ENT_EXT.1	O.VM_ENTROPY
FCS_RBG_EXT.1	O.MANAGEMENT_ACCESS, O.VM_ENTROPY
FCS_TLSS_EXT.1	O.MANAGEMENT_ACCESS
FDP_HBI_EXT.1	O.VM_ISOLATION
FDP_PPR_EXT.1	O.CORRECTLY_APPLIED_CONFIGURATION, O.PLATFORM_INTEGRITY, O.VM_ISOLATION
FDP_RIP_EXT.1	O.RESOURCE_ALLOCATION
FDP_RIP_EXT.2	O.RESOURCE_ALLOCATION
FDP_VMS_EXT.1	O.VM_ISOLATION
FDP_VNC_EXT.1	O.CORRECTLY_APPLIED_CONFIGURATION, O.VM_ISOLATION
FIA_AFL_EXT.1	O.MANAGEMENT_ACCESS
FIA_PMG_EXT.1	O.MANAGEMENT_ACCESS
FIA_UAU.5	O.MANAGEMENT_ACCESS
FIA_UIA_EXT.1	O.MANAGEMENT_ACCESS
FIA_X509_EXT.1	O.AUDIT, O.MANAGEMENT_ACCESS, O.PATCHED_SOFTWARE
FIA_X509_EXT.2	O.AUDIT, O.MANAGEMENT_ACCESS, O.PATCHED_SOFTWARE
FMT_MOF_EXT.1	O.MANAGEMENT_ACCESS
FMT_MSA_EXT.1	O.CORRECTLY_APPLIED_CONFIGURATION, O.VM_ISOLATION
FMT_SMO_EXT.1	O.VMM_INTEGRITY
FPT_DVD_EXT.1	O.PLATFORM_INTEGRITY
FPT_EEM_EXT.1	O.VMM_INTEGRITY
FPT_HAS_EXT.1	O.VMM_INTEGRITY
FPT_HCL_EXT.1	O.CORRECTLY_APPLIED_CONFIGURATION, O.DOMAIN_INTEGRITY, O.VM_ISOLATION, O.VMM_INTEGRITY

Copyright © 2022 IBM

Security functional requirements	Objectives
FPT_RDM_EXT.1	O.VM_ISOLATION
FPT_TUD_EXT.1	O.PATCHED_SOFTWARE
FPT_VDP_EXT.1	O.VMM_INTEGRITY
FPT_VIV_EXT.1	O.DOMAIN_INTEGRITY, O.PLATFORM_INTEGRITY, O.VM_ISOLATION, O.VMM_INTEGRITY
FTA_TAB.1	O.MANAGEMENT_ACCESS
FTP_ITC_EXT.1	O.AUDIT, O.DOMAIN_INTEGRITY, O.MANAGEMENT_ACCESS
FTP_UIF_EXT.1	O.DOMAIN_INTEGRITY
FTP_UIF_EXT.2	O.DOMAIN_INTEGRITY

Table 8: Mapping of security functional requirements to security objectives

6.2.2 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.VM_ISOLATION	FDP_HBI_EXT.1: This SFR supports the objective by requiring the TSF to enforce VM isolation through limiting access to hardware resources. FDP_PPR_EXT.1: This SFR supports the objective by requiring the TSF to enforce VM isolation through limiting access to hardware resources. FDP_VMS_EXT.1: This SFR supports the objective by limiting the methods that can be used to transfer data between Guest VMs. FDP_VNC_EXT.1: This SFR supports the objective by isolating virtual networks from one another. FMT_MSA_EXT.1: This SFR supports the objective by defining the default security posture of data isolation between Guest VMs. FPT_HCL_EXT.1: This SFR supports the objective by controlling the extent to which Guest VMs can interact indirectly with each other via hypercalls. FPT_RDM_EXT.1: This SFR supports the objective by ensuring that removable media cannot be accessed simultaneously by multiple Guest VMs without authorization. FPT_VIV_EXT.1: This SFR supports the objective by ensuring that a Guest VM cannot disrupt the functionality of another Guest VM.

Copyright © 2022 IBM

Security objectives	Rationale
O.VMM_INTEGRITY	FMT_ML_EXT.1: Although TD0567 contains a mapping for this SFR, the protection profile does not contain this SFR. Hence, it is assumed the TD0567 contains a typo and this ST discards the offending mapping. FMT_SMO_EXT.1: This SFR supports the objective by isolating management traffic bound for the VMM from operational traffic transmitted to and from Guest VMs. FPT_EEM_EXT.1: This SFR supports the objective by ensuring that platform-based security functions can be used to protect the integrity of the VMM. FPT_HAS_EXT.1: This SFR supports the objective by allowing the VMM to support hardware-based assistance mechanisms to reduce its own attack surface. FPT_HCL_EXT.1: This SFR supports the objective by controlling the extent to which Guest VMs can interact with the VMM via hypercalls. FPT_VDP_EXT.1: This SFR supports the objective by ensuring that malformed data from a Guest VM cannot be used to compromise the VMM. FPT_VIV_EXT.1: This SFR supports the objective by ensuring that a Guest VM cannot disrupt the functionality of the VMM.
O.PLATFORM_INTEGRITY	FDP_PPR_EXT.1: This SFR supports the objective by limiting the extent to which Guest VMs can interface with the physical platform. FPT_DVD_EXT.1: This SFR supports the objective by limiting the extent to which a Guest VM can interface with the underlying platform. FPT_VIV_EXT.1: This SFR supports the objective by ensuring that a Guest VM cannot disrupt the functionality of the underlying platform
O.DOMAIN_INTEGRITY	FPT_HCL_EXT.1: This SFR supports the objective by controlling the extent to which Guest VMs can interact with the VMM via hypercalls. FPT_VIV_EXT.1: This SFR supports the objective by ensuring that a Guest VM cannot disrupt the functionality of another Guest VM. FTP_ITC_EXT.1: This SFR supports the objective by reducing the likelihood that user actions are inadvertently performed against the wrong Guest VM. FTP_ITC_EXT.2: Although TD0567 contains a mapping for this SFR, the protection profile does not contain this SFR. Hence, it is assumed the TD0567 contains a typo and this ST discards the offending mapping. FTP_UIF_EXT.1: This SFR supports the objective by ensuring the user knows which virtual machine has the active focus. (this mapping is not in TD0567) FTP_UIF_EXT.2: This SFR supports the objective by ensuring the user is able to identify the virtual machine. (this mapping is not in TD0567)
O.MANAGEMENT_ACCESS	FAU_SAR.1: This SFR supports the objective by ensuring that audit data cannot be read by unauthorized subjects.

Copyright © 2022 IBM

Security objectives	Rationale
	FCS_CKM.1: This SFR supports the objective by implementing cryptographic functions that are used to secure administrative interactions with the TSF. FCS_CKM.2: This SFR supports the objective by implementing cryptographic functions that are used to secure administrative interactions with the TSF. FCS_CKM_EXT.4: This SFR supports the objective by implementing cryptographic functions that are used to secure administrative interactions with the TSF. FCS_COP.1(1): This SFR supports the objective by implementing cryptographic functions that are used to secure administrative interactions with the TSF. FCS_COP.1(2): This SFR supports the objective by implementing cryptographic functions that are used to secure administrative interactions with the TSF. FCS_COP.1(3): This SFR supports the objective by implementing cryptographic functions that are used to secure administrative interactions with the TSF. FCS_COP.1(4): This SFR supports the objective by implementing cryptographic functions that are used to secure administrative interactions with the TSF. FCS_RBG_EXT.1: This SFR supports the objective by giving the TOE access to a strong entropy source that can be used to generate strong keys for administrative sessions. FIA_AFL_EXT.1: This SFR supports the objective by protecting against unauthorized access to administrative accounts. FIA_PMG_EXT.1: This SFR supports the objective by defining a password policy that reduces the likelihood of brute force password guessing. FIA_UAU.5: This SFR supports the objective by defining the mechanisms the TSF uses to authenticate administrators. FIA_UIA_EXT.1: This SFR supports the objective by ensuring that administrators must be identified and authenticated before access to the TSF is granted. FIA_X509_EXT.1: This SFR supports the objective by defining how the TSF validates X.509 certificates that may be presented to it as part of establishing a trusted channel. FIA_X509_EXT.2: This SFR supports the objective by defining how the TSF validates X.509 certificates that may be presented to it as part of establishing a trusted channel. FMT_MOF_EXT.1: This SFR supports the objective by defining the management functions to be provided by the TOE. (this mapping is not in TD0567) FTA_TAB.1: This SFR supports the objective by ensuring that administrators are presented with a warning banner that imputes actionable consequences for misuse of the TOE.

Copyright © 2022 IBM

Security objectives	Rationale
	FTP_ITC_EXT.1: This SFR supports the objective by defining any trusted protocols used for remote administration.
O.PATCHED_SOFTWARE	FPT_TUD_EXT.1: This SFR supports the objective by defining a mechanism used to securely update the VMM. FIA_X509_EXT.1: This SFR supports the objective by defining how the TSF validates X.509 certificates that may be presented to it as an attestation of the authenticity and integrity of a software update. FIA_X509_EXT.2: This SFR supports the objective by optionally using X.509 certificates as the method of validating software updates.
O.VM_ENTROPY	FCS_ENT_EXT.1: This SFR supports the objective by providing a mechanism for Guest VMs to have entropy data available for use. FCS_RBG_EXT.1: This SFR supports the objective by giving the TOE access to a strong entropy source that can be used by Guest VMs.
O.AUDIT	FAU_GEN.1: This SFR supports the objective by ensuring that audit records are generated for security-relevant events. FAU_STG.1: This SFR supports the objective by ensuring that audit data cannot be deleted without authorization or modified by any subject. FAU_STG_EXT.1: This SFR supports the objective by requiring redundant storage of audit data. FIA_X509_EXT.1: This SFR supports the objective by defining how the TSF validates X.509 certificates that may be presented to it as part of establishing a trusted channel. FIA_X509_EXT.2: This SFR supports the objective by defining how the TSF validates X.509 certificates that may be presented to it as part of establishing a trusted channel. FTP_ITC_EXT.1: This SFR supports the objective by defining the trusted protocols used for remote audit data transfer.
O.CORRECTLY_APPLIED_CONFIGURATION	FDP_PPR_EXT.1: This SFR supports the objective by defining the security policy used to govern Guest VM access to physical resources. FDP_VNC_EXT.1: This SFR supports the objective by defining the security policy used to govern Guest VM access to network resources. FMT_MSA_EXT.1: This SFR supports the objective by defining the default security policy for data sharing between VMs. FPT_HCL_EXT.1: This SFR supports the objective by defining the security policy used to govern Guest VM access to Hypercall functions.
O.RESOURCE_ALLOCATION	FDP_RIP_EXT.1: This SFR supports the objective by ensuring that physical memory cannot be allocated to multiple Guest VMs.

Copyright © 2022 IBM

Security objectives	Rationale
	FDP_RIP_EXT.2: This SFR supports the objective by ensuring that disk storage cannot be allocated to multiple Guest VMs.

Table 9: Security objectives for the TOE rationale

6.2.3 Security Requirements Dependency Analysis

The security functional requirements in this Security Target do not introduce dependencies on any security assurance requirement; neither do the security assurance requirements in this Security Target introduce dependencies on any security functional requirement.

The following table demonstrates the dependencies of the SFRs modeled in [VPP] and [SVPP], and how the SFRs for the TOE resolve those dependencies.

Security functional requirement	Dependencies	Resolution
FAU_GEN.1	FPT_STM.1	The PP explicitly excludes the SFR of FPT_STM.1. Due to claiming exact conformance, this SFR is excluded from the ST as well.
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1
FAU_STG.1	FAU_GEN.1	FAU_GEN.1
FAU_STG_EXT.1	No dependencies	
FCS_CKM.1	FCS_COP.1	FCS_COP.1(3)
	FCS_CKM.2	FCS_CKM.2
	FCS_CKM.4	The SFR of FCS_CKM_EXT.4 is the replacement for FCS_CKM.4 defined by the PP. This SFR enhances FCS_CKM.4
FCS_CKM.2	FCS_CKM.1	FCS_CKM.1
	FCS_CKM.4	The SFR of FCS_CKM_EXT.4 is the replacement for FCS_CKM.4 defined by the PP. This SFR enhances FCS_CKM.4
FCS_CKM_EXT.4	FCS_CKM.1	FCS_CKM.1

Copyright © 2022 IBM

Security functional requirement	Dependencies	Resolution
FCS_COP.1(1)	FCS_CKM.1	FCS_CKM.2 The symmetric cipher operation is employed as part of the trusted channel as specified by FTP_ITC_EXT.1. The trusted channel uses a key agreement schema defined by FCS_CKM.2 to derive the symmetric session keys. Hence, the dependency for key generation is met by the SFR claiming the key agreement mechanism.
	FCS_CKM.4	The SFR of FCS_CKM_EXT.4 is the replacement for FCS_CKM.4 defined by the PP. This SFR enhances FCS_CKM.4
FCS_COP.1(2)	FCS_CKM.1	The hashing operation does not require any key material.
	FCS_CKM.4	The SFR of FCS_CKM_EXT.4 is the replacement for FCS_CKM.4 defined by the PP. This SFR enhances FCS_CKM.4
FCS_COP.1(3)	FCS_CKM.1	FCS_CKM.1
	FCS_CKM.4	The SFR of FCS_CKM_EXT.4 is the replacement for FCS_CKM.4 defined by the PP. This SFR enhances FCS_CKM.4
FCS_COP.1(4)	FCS_CKM.1	The keyed hash cipher operation is employed as part of the trusted channel as specified by FTP_ITC_EXT.1. The trusted channel uses a key agreement schema defined by FCS_CKM.2 to derive the keyed hash keys. Hence, the dependency for key generation is met by the SFR claiming the key agreement mechanism.
	FCS_CKM.4	The SFR of FCS_CKM_EXT.4 is the replacement for FCS_CKM.4 defined by the PP. This SFR enhances FCS_CKM.4
FCS_ENT_EXT.1	No dependencies	
FCS_RBG_EXT.1	No dependencies	
FCS_TLSS_EXT.1	No dependencies	

Copyright © 2022 IBM

Security functional requirement	Dependencies	Resolution
FDP_HBI_EXT.1	No dependencies	
FDP_PPR_EXT.1	No dependencies	
FDP_RIP_EXT.1	No dependencies	
FDP_RIP_EXT.2	No dependencies	
FDP_VMS_EXT.1	No dependencies	
FDP_VNC_EXT.1	No dependencies	
FIA_AFL_EXT.1	No dependencies	
FIA_PMG_EXT.1	No dependencies	
FIA_UAU.5	No dependencies	
FIA_UIA_EXT.1	No dependencies	
FIA_X509_EXT.1	FPT_STM.1	The PP explicitly excludes the SFR of FPT_STM.1. Due to claiming exact conformance, this SFR is excluded from the ST as well.
FIA_X509_EXT.2	No dependencies	
FMT_MOF_EXT.1	No dependencies	
FMT_MSA_EXT.1	No dependencies	
FMT_SMO_EXT.1	No dependencies	
FPT_DVD_EXT.1	No dependencies	
FPT_EEM_EXT.1	No dependencies	
FPT_HAS_EXT.1	No dependencies	
FPT_HCL_EXT.1	No dependencies	
FPT_RDM_EXT.1	No dependencies	
FPT_TUD_EXT.1	No dependencies	
FPT_VDP_EXT.1	No dependencies	
FPT_VIV_EXT.1	No dependencies	
FTA_TAB.1	No dependencies	
FTP_ITC_EXT.1	No dependencies	
FTP_UIF_EXT.1	No dependencies	

Copyright © 2022 IBM

Security functional requirement	Dependencies	Resolution
FTP_UIF_EXT.2	No dependencies	

Table 10: TOE SFR dependency analysis

6.3 Security Assurance Requirements

The security assurance requirements (SARs) for the TOE are defined in assurance packages.

The following table shows the SARs, 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 Conformance claims		No	No	No	No
	ASE_ECD.1 Extended components definition		No	No	No	No
	ASE_INT.1 ST introduction		No	No	No	No
	ASE_OBJ.2 Security objectives		No	No	No	No
	ASE_REQ.1 Stated security requirements		No	No	No	No
	ASE_SPD.1 Security problem definition		No	No	No	No
	ASE_TSS.1 TOE summary specification		No	No	No	No
ADV Development	ADV_FSP.1 Basic functional specification		No	No	No	No
AGD Guidance documents	AGD_OPE.1 Operational user guidance		No	No	No	No
	AGD_PRE.1 Preparative procedures		No	No	No	No
ALC Life-cycle support	ALC_CMC.1 Labelling of the TOE		No	No	No	No
	ALC_CMS.1 TOE CM coverage		No	No	No	No
	ALC_TSU_EXT.1		No	No	No	No
ATE Tests	ATE_IND.1 Independent testing - conformance		No	No	No	No
AVA Vulnerability assessment	AVA_VAN.1 Vulnerability survey		No	No	No	No

Table 11: SARs

6.4 Security Assurance Requirements Rationale

There are no security assurance components defined in this ST apart from the ones taken from CC Part 3 and as defined in the PP.

Copyright © 2022 IBM

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 and support the security requirements claimed in section 6.1 are described here.

7.1.1 Overview of the TOE architecture

z/VM is an operating system operating on IBM 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. Cryptographic support including the implementation of a trusted channel (F.CS)
3. Discretionary access control (F.AC)
4. Identification & authentication (F.I&A)
5. Interference Protection between virtual machines (F.IP)
6. Object re-use (F.OR)
7. Security management functions to administer audit, discretionary access control as well as users and groups with their related attributes (F.SM)
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 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.

Copyright © 2022 IBM

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.

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

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

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.

Copyright © 2022 IBM

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.

The TOE offers a read-only auditor role which allows reviewing the audit trail without affecting it or changing the audit configuration.

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.

Copyright © 2022 IBM

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.

If configured by the administrator, RACF offloads audit data to a remote audit server. The remote audit server is accessed via a TLS channel that is initiated by the RACF using System SSL.

This supports FAU_SAR.1.

7.1.2.4 F.AU.4 - Generation of Audit Records by TLS Server

Encrypted network connections are audited by the enablement of tracing inside the TLS Server. This worker machine handles the encryption and decryption of all traffic flowing into and out of the TOE. By default, it only captures base information about TLS configuration at time of start-up. However, auditing on a per-connection basis may be enabled by use of the SSLADMIN command. This allows a network security administrator to view:

- Successful and/or failing connections
- IP addresses, domain names, client certificate Distinguished Names
- Timestamps of connection (open-to-fail, open-to-close)

Auditing in the TLS server can be tailored to cover a subset of IP addresses and/or ports.

Information is gathered from the TLS Server by issuance of the SSLADMIN LOG command. This command (like all SSLADMIN commands) is privileged and requires DTCPARMS access on a per-TLS-server basis. Issuance of the SSLADMIN LOG returns all data from the TLS worker machine’s spool area as a reader file sent to a pertinent virtual machine; this can then be saved onto local disk by the network administrator and stored as part of the audit trail, or transmitted off-platform for archival purposes as relevant.

To validate entropy used by the TLS Server, special tracing must be enabled in the cryptographic library used by the TLS Server. This setting is enabled by adding a GSKTRACE operand in the DTCPARMS configuration file. The GSKTRACE setting is static and cannot be enabled once the TLS Server is running. Once the TLS server is running, gather gsktrace data from a Byte File System (BFS) record after the TLS Server has been shut down. This record can only be processed from GSKADMIN, a privileged virtual machine which is authorized to managed certificates and certificate databases. Once generated, this can be saved onto local disk by the security administrator and stored as part of the audit trail, or transmitted off-platform for archival purposes as relevant.

7.1.3 F.CS: Cryptographic Support

The TOE 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.

Copyright © 2022 IBM

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 keys. In addition, the SSL server implements the cryptographic primitives needed for the TLS protocol compliant to FIPS 186-4.

The random numbers required for the the cryptographic operations of the TLS protocol is delivered with an SP800-90A Hash_DRBG using SHA-512 as core that is seeded with 256 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.

Sensitive data such as keys held in volatile memory are immediately overwritten with zeros when discarded. When such data is stored in non-volatile store, the data is overwritten with zeros when the memory is reclaimed.

This supports FCS_COP.1(1), FCS_COP.1(3), FCS_RBG_EXT.1.

7.1.3.1 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
- PRNO Perform Random Number Operation

Copyright © 2022 IBM

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
- Random Number Generation including providing of an entropy source used to seed the deterministic random number generator

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.

CPACF offers "Protected keys in CPACF", which is available on all supported platforms. 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.

7.1.3.2 Trusted Update

With the forthcoming PTF for z/VM 7.2 to enable direct-to-host service download, z/VM will use a local web service to connect to ShopZ and pull administrator-specific packages from an order list down to an authorized z/VM userid. These order packages must correspond to the links provided via ShopZ email. Digital hashes will be provided in the ordering email for visual integrity verification against hash values posted on ShopZ; direct-to-host code will validate the hash correspondence at the time of package download. The download will only completed successfully if the hash values match the GIMPAF value listed on ShopZ.

7.1.3.3 X.509 Certificate Validation and Authentication

X.509 certificates are checked during TCP/IP processing when establishing a connection to the hypervisor layer. This processing occurs in the TLS worker machines (SSL00001 and similar), which are exclusively associated with a given TCP/IP stack.

The TLS worker machine has a FIPS 140-2 compliant instantiation of the System SSL cryptographic library, which is used for software cryptographic operations related to connectivity. As of the PFT for z/VM APAR PH28216, Online Certificate Status Protocol (OSCP) is supported for certificate validation during the handshake process. OCSP or connectivity to a CRL provider is configured as part of z/VM TCP/IP and SSL set-up.

Copyright © 2022 IBM

When the TLS server cannot reach OCSP / CRL, the connection is rejected, as security policy cannot be validated.

The remote peer's IP address or full qualified domain name is matched with the SAN of the X.509 certificate first. If no SAN is present, the matching against the CN entry in the X.509 certificate is performed. If no matching with the CN entry is detected, the remote peer is not authenticated with the current certificate. In this case, another certificate is attempted, if one is available.

7.1.4 F.AC: Access Control

7.1.4.1 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) 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 7.1.4.6.

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.

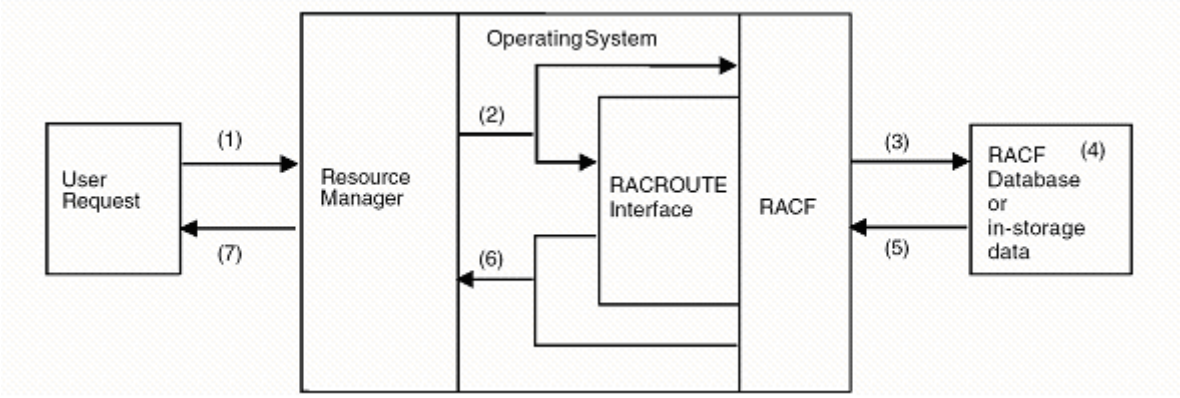

Figure 1: RACF and its relationship to the operating system

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 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.

Copyright © 2022 IBM

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".

7.1.4.2 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 or optionally AES 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

Copyright © 2022 IBM

Name	Description
OPERATIONS	Gives the user the system-wide OPERATIONS attribute

Table 12: 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.

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 13: RACF group profile

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
ACLs	Access control information (see definition below on the content of an individual ACL)

Table 14: RACF resource profile

Copyright © 2022 IBM

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.

7.1.4.3 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
- The access authority specified in the resource profile

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.

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).

Copyright © 2022 IBM

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 (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

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.

ROAUDIT Attribute

A user with the ROAUDIT attribute allows that user to review the audit trail without affecting either the audit trail nor the audit configuration.

Copyright © 2022 IBM

Group-ROAUDIT

A user with the group-ROAUDIT attribute can review audit trail.

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 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.

Copyright © 2022 IBM

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].

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.

7.1.4.4 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

Copyright © 2022 IBM

- 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.

7.1.4.5 F.AC.5 – Protected Resources

On z/VM, RACF can be used to control access to all objects with discretionary 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).
- SURROGAT
  If surrogate login or access is allowed, and if allowed, which user IDs can act as surrogates.
- TERMINAL
  Terminals.

7.1.4.6 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.

Copyright © 2022 IBM

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.

7.1.5 F.I&A: Identification and Authentication

7.1.5.1 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. 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.

Copyright © 2022 IBM

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.

The administrator is able to configure an access banner which is displayed to a user before authenticating to the system. This banner may contain arbitrary information such as legal statements.

This supports FIA_UAU.5.

7.1.5.2 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 and minimum 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. Also the user can set this interval to specify a minimum time, the password needs to be remain unchanged.

With the SPECIALCHARS option, users are allowed to use special characters in their 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.

Copyright © 2022 IBM

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. Optionally the AES encryption algorithm can be used for that purpose as well. 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)
- Minimum password lifetime (MININTERVAL 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)
- Allowing special characters in the password (SPECIALCHARS suboption)
- Type of character for each character position of a password. Possible types are:
    - ALPHA
    - ALPHANUM
    - VOWEL
    - NOVOWEL
    - CONSONANT
    - NUMERIC
    - MIXEDALL

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. The administrator is able to configure the threshold of consecutive failed authentication attempts that triggers the REVOKE setting.

This supports FIA_AFL_EXT.1, FIA_PMG_EXT.1.

7.1.5.3 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 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.

7.1.6 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.

Copyright © 2022 IBM

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.
- 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

Copyright © 2022 IBM

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	Multidirectional channel between all configured virtual machines
VMCF	bidirectional channel between two configured virtual machines - this communication channel is deprecated
IUCV	bidirectional channel between two configured virtual machines
VCTC	bidirectional channel between two configured virtual machines
Spool files	transferring spool files between virtual machines
AUTOLOG	Providing of initial console data to virtual machine
XAUTOLOG	Providing of initial console data to virtual machine
SET SECUSER	Enable read and write access to a virtual machine console
SET OBSERVER	Enable read access to a virtual console
Minidisks	configured minidisks are shared
Memory	configured memory range is shared

Table 15: Communication channel usage

This supports FMT_MSA_EXT.1.

7.1.6.1 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

Copyright © 2022 IBM

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.

7.1.6.2 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.

7.1.7 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.

7.1.8 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
- Management of the audit trail and the events to be audited
- Management of user security attributes, including authentication data and access control

Copyright © 2022 IBM

- Management of VLAN to virtual machine mappings
- Management of virtual machine resource assignments

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

This supports FMT_MSA_EXT.1.

7.1.8.1 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, ROAUDIT, 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.

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

This supports FPT_DVD_EXT.1.

7.1.8.2 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)

Copyright © 2022 IBM

- 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

7.1.8.3 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 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).

7.1.8.4 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.

Copyright © 2022 IBM

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.

This supports FPT_DVD_EXT.1.

Processor Features

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

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

The SIE instruction provided by the processor is the central facility the TOE manages. It is called by the Control Program (CP) restricting the scope of the processor to a limited memory range to set up a virtual machine environment. If the processor enforcing a SIE environment is instructed to execute predefined privileged instructions, the SIE environment is terminated and control is returned to CP. This SIE instruction is executed with a CP-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,

Copyright © 2022 IBM

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 z System 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.

Copyright © 2022 IBM

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 add 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.

7.1.10 Audit Data Generation

See section 7.1.2 for a full description of the audit functionality.

This supports FAU_GEN.1.

7.1.11 Protected Audit Trail Storage

See section 7.1.2.2 for a full description of the audit trail storage.

This supports FAU_STG.1.

7.1.12 Off-Loading of Audit Data

Section 7.1.2.3 explains the possibilities of events that may occur when the audit trail becomes full.

This supports FAU_STG_EXT.1.

7.1.13 Cryptographic Key Generation

Section 7.1.3 explains the general use case for cryptography employed by the TOE.

The following RSA key sizes are supported by the TOE: 2048 and 4096.

The following Diffie-Hellman key sizes are supported by the TOE: 2048.

This supports FCS_CKM.1.

Copyright © 2022 IBM

7.1.14 Cryptographic Key Establishment

Section 7.1.3 explains the general use case for cryptography employed by the TOE.

This supports FCS_CKM.2.

7.1.15 Cryptographic Key Destruction

Section 7.1.3 explains the general use case for cryptography employed by the TOE.

The following key types are maintained by the TOE:

- Ephemeral in memory: symmetric keys, keyed message digest keys, asymmetric keys during processing
- Non-volatile keys stored on disk: X.509 certificates and keys used for X.509 related operations.

This supports FCS_CKM_EXT.4.

7.1.16 Cryptographic Operation (Hashing)

Section 7.1.3 explains the general use case for cryptography employed by the TOE.

The TOE uses hashing operations for the following purposes:

- TLS KDF and keyed message digests: all hash types as defined by the chosen TLS cipher suite.
- Signature operations: all SHA-2 hashes as defined by the X.509 certificate meta data.

This supports FCS_COP.1(2).

7.1.17 Cryptographic Operation (Keyed Hash Algorithms)

Section 7.1.3 explains the general use case for cryptography employed by the TOE.

The TOE uses the keyed hash algorithms with the following cipher specification:

- HMAC-SHA-1: key length is equal to block size, block size is 512 bits, output MAC length is 160 bits
- HMAC-SHA-256: key length is equal to block size, block size is 512 bits, output MAC length is 256 bits
- HMAC-SHA-384: key length is equal to block size, block size is 1024 bits, output MAC length is 384 bits
- HMAC-SHA-512: key length is equal to block size, block size is 1024 bits, output MAC length is 512 bits

This supports FCS_COP.1(4).

7.1.18 Entropy for Virtual Machines

Section 7.1.3 explains the general use case for cryptography employed by the TOE.

The TOE allows virtual machines to access the CPU TRNG acting as a noise source.

This supports FCS_ENT_EXT.1.

Copyright © 2022 IBM

7.1.19 TLS Server Protocol

See section 7.1.3 for a description of the TLS support.

The TOE in the evaluated configuration is configured to only allow TLS 1.2. The list of cipher suites supported by the TOE is provided in FCS_TLSS_EXT.1. The supported key agreement mechanisms are listed in FCS_TLSS_EXT.1.

This supports FCS_TLSS_EXT.1.

7.1.20 Hardware-Based Isolation Mechanisms

See section 7.1.9.1 for a list of used hardware mechanisms.

This supports FDP_HBI_EXT.1.

7.1.21 Physical Platform Resource Controls

See section 7.1.9 for the description how a virtual machine is maintained. Specifically, this section outlines that each virtual machine is maintained using the CPU’s SIE instruction allowing for interception and proper handling of processor instructions related to physical I/O-devices.

z/VM, as a virtualization platform, tends not to have direct physical access to "real" resources. While it is technically possible, it happens most often around disk and PCIe devices (such as network or cryptographic accelerators).

z/VM does not see either "real" CPU or "real" RAM, however. Because z/VM runs inside of an IBM Z Logical Partition, real CPU devices are abstracted as "logical CPUs" as viewed by the z/VM host (or privileged administrator), and as "virtual CPUs" when viewed by an unprivileged guest or automated workload. The correlation of real to logical to virtual is flexible, depending upon operational requirements, but it is recommended that z/VM "overcommits" to allow for maximum utility of resource by means of virtualization. Overcommitment implies that more resources are assigned to virtual machines than actually present assuming that all virtual machines do not utilize all their assigned resources at the same time.

Similarly, z/VM sees a "logical" version of the physical memory, in that there's a dynamic translation of addresses that takes place. When z/VM partition A references memory location x'00010000' and partition B references memory location x'0001000', they refer to different locations in the physical RAM/memory on the underlying IBM mainframe hardware (CPC). PR/SM uses "zones" to tell the HW where a z/VM partition's logical memory is located in the physical memory of the CPC. Additionally, PR/SM has the ability to dynamically move a z/VM partition's logical memory to a new location in the physical RAM/memory on the CPC (for instance) to better align the topological location of the z/VM partition's memory and CPUs.

This supports FDP_PPR_EXT.1.

7.1.22 Residual Information in Memory

The CP kernel manages the logical memory. Every time a process is given new memory, the kernel allocates the requested amount of memory pages which are all zeroized before the process can access them as outlined in section 7.1.7.

This supports FDP_RIP_EXT.1.

Copyright © 2022 IBM

7.1.23 Residual Information on Disk

Section 7.1.7 outlines how disk space is cleared before reassignment to another virtual machine.

This supports FDP_RIP_EXT.2.

7.1.24 VM Separation

See section 7.1.6 for the separation of virtual machines including a configuration of inter-VM-communication. The IUCV provides bidirectional communication channels between virtual machines.

This supports FDP_VMS_EXT.1.

7.1.25 Virtual Networking Components

See section 7.1.6 for the description on virtual machine networking. The TOE allows configuration of virtual switches and network interfaces for establishing a communication between virtual machines within a host. In addition, the TOE can be configured to link a virtual switch or a virtual interface with a physical interface to allow a virtual machine to communicate outside of the host.

This supports FDP_VNC_EXT.1.

7.1.26 Administrator Identification and Authentication

See section 7.1.5 describing the TOE-provided identification and authentication.

This supports FIA_UIA_EXT.1 as well as all other SFRs from the FIA family.

7.1.27 X.509 Certificate Validation

Section 7.1.3.3 outlines the X.509 certificate handling, processing and validation during TLS connection establishment.

This supports FIA_X509_EXT.1.

7.1.28 X.509 Certificate Authentication

Section 7.1.3.3 outlines the X.509 certificate handling, processing and validation during TLS connection establishment.

This supports FIA_X509_EXT.2.

7.1.29 Management of Security Functions Behavior

See section 7.1.8 for details about the security management of the TOE.

This supports FMT_MOF_EXT.1.

7.1.30 Separation of Management and Operational Networks

Section 7.1.6 outlines the networking configuration support offered to virtual machines.

This supports FMT_SMO_EXT.1.

Copyright © 2022 IBM

7.1.31 Execution Environment Mitigations

The TOE does not implement specific mitigation techniques.

This supports FPT_EEM_EXT.1.

7.1.32 Hardware Assists

See section 7.1.9 for the description how a virtual machine is maintained. Specifically, this section outlines that each virtual machine is maintained using the CPU's SIE instruction.

This supports FPT_HAS_EXT.1.

7.1.33 Hypercall Controls

See section 7.1.4.6 for the description how access to the DIAGNOSE codes (the z/VM equivalent of hypercalls, including MSG, WNG, MSGNOH, SMSG) are governed.

This supports FPT_HCL_EXT.1.

7.1.34 Removable Devices and Media

z/VM allows the administrator to assign the removable media of tapes to individual virtual machines. For details. please see section 7.1.6.

This supports FPT_RDM_EXT.1.

7.1.35 Trusted Updates to the Virtualization System

See section 7.1.3.2 for the description of the trusted update process.

This supports FPT_TUD_EXT.1.

7.1.36 TOE Access Banner

See section 7.1.5.1 for the description of access banner support.

This supports FTA_TAB.1.

7.1.37 Virtual Device Parameters

See sections 7.1.4 as well as 7.1.6 for the description the management of virtual devices.

The TOE supports the following types of interfaces:

- Hypercalls: A guest can issue hypercalls (i.e. DIAGNOSE codes) to access para-virtualized host services implemented in the kernel.
- Exceptions: A guest can trigger exceptions that must be caught by the host kernel.
- Networking: The TOE allows access to a guest's console via telnet. That VNC communication channel may be tunneled through cryptographic channels if configured by the administrator.

The TOE provides virtualization of the following types of devices:

- Disk storage devices
- Network devices

Copyright © 2022 IBM

Access to a virtual machine's console is given via the Control Program interface. This allows user to log on, if configured by the administrator. After logging on, the console of the corresponding virtual machine can be accessed. The TOE uses this approach as a replacement for VNC support.

This supports FPT_VDP_EXT.1.

7.1.38 Trusted Channel Communications

See section 7.1.3 for the description of the SSL server and how it provides a cryptographic communication channel for the TOE.

This supports FTP_ITC_EXT.1

7.1.39 User Interface: I/O Focus

See section 7.1.6.1 for the description of how the TOE allows users to access virtual machines.

This supports FTP_UIF_EXT.1.

7.1.40 User Interface: Identification of VM

See section 7.1.6.1 for the description of how the TOE allows users to access virtual machines.

Access to virtual machines is only possible via network access: either network access of the guest operating system or via telnet offered by the TOE for the guest console. Each virtual machine has one or more unique IP addresses and ports for network access allowing an unambiguous identification. Using the IP address and the port the client is able to unambiguously identify the the VM. A graphical rendering of the identification of the VM to a human user is to be performed by the client.

This supports for FTP_UIF_EXT.2.

7.1.41 Vendor Attestation

The following statements are made by the TOE vendor to satisfy specific Security Functional Requirements.

7.1.41.1 FDP_VMS_EXT.1.1

A Guest VM cannot access the data of another Guest VM, or transfer data to another Guest VM other than through the mechanisms described in FDP_VMS_EXT.1.1 when expressly enabled by an authorized Administrator. There are no design or implementation flaws that permit the above mechanisms to be bypassed or defeated, or for data to be transferred through undocumented mechanisms. This claim does not apply to covert channels or architectural side-channels.

7.1.41.2 FDP_VNC_EXT.1.2

Traffic traversing a virtual network is visible only to Guest VMs that are configured by an Administrator to be members of that virtual network. There are no design or implementation flaws that permit the virtual networking configuration to be bypassed or defeated, or for data to be transferred through undocumented mechanisms. This claim does not apply to covert channels or architectural side-channels.

Copyright © 2022 IBM

7.1.41.3 FPT_VDP_EXT.1.2

Parameters passed from Guest VMs to virtual device interfaces are thoroughly validated and all illegal values (as specified in the TSS) are rejected. Additionally, parameters passed from Guest VMs to virtual device interfaces are not able to degrade or disrupt the functioning of other VMs, the VMM, or the Platform. Thorough testing and architectural design reviews have been conducted to ensure the accuracy of these claims, and there are no design or implementation flaws that bypass or defeat the security of the virtual device interfaces.

7.1.41.4 FPT_VIV_EXT.1.2

Software running in a VM is not able to degrade or disrupt the functioning of other VMs, the VMM, or the Platform. There are no design or implementation flaws that bypass or defeat VM isolation.

7.1.42 Timely Updates

The entire TOE is subject to an extensive update process. The update process starts when IBM is informed about defects. Depending on the severity (security incidents are considered to be severe), fixes are developed, tested and released with PTFs.

The entire update process is handled by IBM and covers all components shipped as part of z/VM.

Guaranteed response times depend on the selected service level agreement which is outlined in https://www.ibm.com/support/pages/node/738891. The security incident and response process allows customers to directly interact with the IBM team via a central contact system documented at https://www.ibm.com/support/pages/node/733923 to report issues. Based on a timely triage and root cause analysis a responsible resolution of the incident report is ensured which may result in the release of an update of the affected software binaries. Such updates are made available via the automated update channels to all customers.

Identified issues can be relayed to IBM either via the support channels defined by the service level agreement or via the communication specified in https://www.ibm.com/support/pages/node/733923.

This supports ALC_TSU_EXT.1.

7.2 TOE Assurance Measures

The assurance measures provided by the developer to meet the security assurance requirements for the TOE are based on the developer action elements and the requirements on content and presentation of evidence elements defined for the individual assurance requirements in CC Part 3:

SAR	Assurance Measures
ADV_FSP.1	The Functional Specification for all externally visible interfaces of the TOE and the security functions implemented by the TOE will be provided.
AGD_OPE.1	Administrator guidance is provided to allow the secure operation of the TOE in its intended environment in compliance with the evaluated configuration.
AGD_PRE.1	To the extent that the TOE is available to end-users, the relevant security functionality and assumptions on user behavior is documented for end-users.
ALC_CMC.1	The developer uses configuration management systems to provide version and access control for configuration items.

Copyright © 2022 IBM

SAR	Assurance Measures
ALC_CMS.1	Configuration management is provided for the implementation representation of the TOE, the design documentation, test documentation, and all other evidence relevant for the TOE evaluation.
ATE_IND.1	Independent testing of the security functionality is conducted compliant to the assurance activities defined by the PP.
AVA_VAN.1	A search for publicly known vulnerabilities is conducted as part of the evaluation.

Table 16: Assurance measures meeting the TOE security assurance requirements

Copyright © 2022 IBM

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

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*.

Copyright © 2022 IBM

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.

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*.

Mediation
When access control policy rules 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.

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

Copyright © 2022 IBM

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
RACFSAG	**Resource Access Control Facility Security Administrator's Guide**	
	Version	SC24-6308-01
	Date	2019-06-20
	Location	https://www-01.ibm.com/servers/resourcelink/svc0302a.nsf/pages/zVMV7R1sc246308/$file/icha1_v7r1.pdf
SVPP	**Protection Profile for Virtualization Extended Package Server Virtualization**	
	Version	1.0
	Date	2016-11-17
VPP	**Protection Profile for Virtualization**	
	Version	1.0
	Date	2016-11-17

Copyright © 2022 IBM