Security Target for z/OS Version 2 Release 5

Version:	**2.17**
Status:	**Released**
Last Update:	**2024-08-27**

Trademarks

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

- ESCON®
- FICON®
- HiperSockets
- IBM®
- IBM® zEnterprise System
- OS/390®
- Processor Resource/Systems Manager
- PR/SM
- ResourceLink®
- RETAIN®
- S/360®
- S/370®
- S/390®
- System z®
- VM/ESA®
- VSE/ESA
- zEnterprise®
- z/Architecture®
- z/OS®
- z/VM®
- zSeries®
- z System
- Linux on z Systems
- IBM LinuxONE(tm)
- LinuxONE
- IBM z16
- z/TPF

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

More details on IBM UNIX hardware, software and solutions may be found at ibm.com/servers/unix/.

InfiniBand is a registered trademark of the InfiniBand Trade Association.

Linux is a registered trademark of Linus Torvalds.

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

IBM, the IBM logo, the e-business logo, LinuxONE, AIX, DB2, DB2 Universal Database, pSeries, RS/6000, SP and WebSphere are registered trademarks or trademarks of the International Business Machines Corporation in the United States and/or other countries.

Other company, product and service names may be trademarks or service marks of others. IBM may not offer the products, programs, services or features discussed herein in other countries, and the information may be subject to change without notice.

Legal Notice

General availability may vary by geography.

IBM hardware products are manufactured from new parts, or new and used parts. Regardless, our warranty terms apply.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Any reliance on these statements is at the relying party's sole risk and will not create any liability or obligation for IBM.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Revision History

Version	Date	Author(s)	Changes to Previous Revision
2.17	2024-08-27	Author: Mark Nelson	Public release.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Security Target Identification

Title:	Security Target for z/OS Version 2 Release 5
Version:	2.17
Status:	Released
Date:	2024-08-27
Author:	Mark Nelson, IBM
Sponsor:	IBM Corporation
Developer:	IBM Corporation
Certification Body:	OCSI
Certification ID:	OCSI/CERT/ATS/05/2023
Keywords:	operating system, access control, identification, authentication, audit, object reuse

1.2 TOE Identification

The TOE is IBM z/OS Version 2 Release 5.

1.3 TOE Type

The TOE type is Operating System.

1.4 TOE Overview

This Security Target (ST) documents the security characteristics of the IBM z/OS Version 2 Release 5 operating system with the additional required licensed programs (see section Software Configuration of this ST) configured in a secure manner as described in z/OS Planning for Multilevel Security and the Common Criteria ([MLSGUIDE]).

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

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

IBM z/OS provides such software technologies as Enterprise Java™ Beans, eXtensible Markup Language (XML), HyperText Markup Language (HTML), Unicode and distributed Internet Protocol (IP) networking. z/OS UNIX System Services allows customers to develop and run UNIX programs on z/OS and exploit the reliability and scalability of the z System processors. z/OS also incorporates cryptographic services, distributed print services, workload management, storage management, parallel sysplex availability,

and automation capabilities. Not all of these functions have been analyzed in this evaluation; see section Software Configuration for the software configuration of z/OS used in this evaluation. The security functions subject to this evaluation are described in chapter 7 of this document.

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

1.5 TOE Description

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

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

- a logical partition provided by a certified version of PR/SM on an IBM z System™ processor (System z16).
- a certified version of z/VM® executing in a logical partition provided by PR/SM on one of the above-listed z System™ processors.

Most of 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.

Cryptographic functions implemented by the Crypto Express 8 coprocessors are used to perform cryptographic operations as well to handle cryptographic keys. It should be noted, that a cryptographic coprocessor is required to operate the TOE in its evaluated configuration.

A user who wants to use cryptographic functions provided by a coprocessor should be aware that although those functions have been tested during the evaluation for functional correctness, no further analysis of the design and implementation of those cryptographic functions implemented on the coprocessors has been performed. Especially no analysis for potentially exploitable side channels of the implementation of the cryptographic functions of the coprocessors has been performed.

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

The individual TOEs can be run alone or within a network as a set of cooperating hosts, operating under and implementing the same set of security policies. They also can be connected to form a loosely-coupled complex of systems called a sysplex

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

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

1.5.1 Intended Method of Use

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

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

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

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

i. Pre-specified identities:
    a. The authorized administrator can specify an identity to be used by server or daemon processes or system address spaces, which may be started either automatically or via system operator commands;

ii. Users are allowed to execute programs that accept network connections on ports the user has access to. In this case the untrusted program has no knowledge about the external "user" and cannot perform authentication. The program executes with the rights of the z/OS user that started it, and any data access occurs using this user's authenticated identity.

All TOE resources are under the control of the TOE. The TOE mediates the access of subjects to TOE-protected objects. Subjects in the TOE are called tasks. Tasks are the active entities that can act on the user's behalf. Data is stored in named objects.

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.

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

The TOE also recognizes the role of an auditor, who uses the auditing system provided by z/OS to monitor the system usage according to the organizational security policies. An additional role of a 'read-only auditor' can be assigned for an auditor that shall not have the capability to manage the audit but only be able to read audit records and audit related configuration options.

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

1.5.2 Summary of Security Features

The primary security features of the product are:

- identification and authentication
- discretionary access control
- auditing
- security management
- cryptographic support
- communications security
- TSF protection
- confidentiality protection of datasets

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

1.5.2.1 Identification and authentication

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

- an alphanumeric RACF user ID and a system-encrypted password or (for applications that support it) password phrase.
- an alphanumeric RACF user ID and a PassTicket, which is a cryptographically-generated password substitute encompassing the user ID, the requested application name, and the current date/time.
- an SSH key that is configured to be trusted by the user and that is presented to the SSH server during the authentication process.

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

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

The password quality can be tailored to the installation's policies using various parameters. When creating users, administrators are required to choose an initial password and optionally a password phrase, that must usually be changed by the user during the initial logon that uses the password/phrase.

Administrators may configure a policy for the lockout of accounts.

1.5.2.2 Discretionary access control

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

RACF makes access control decisions based on the user's identity, security attributes, group authorities, and the access authority specified with respect to the resource profile.

The TOE provides the following DAC mechanisms:

i. The z/OS UNIX DAC mechanism is used for z/OS UNIX objects (files, directories, etc.)

1.5.2.2.1 z/OS UNIX DAC mechanism

z/OS implements POSIX-conformant access control for such named objects in the UNIX realm as UNIX file system objects and UNIX inter-process communication (IPC) objects. Access types for UNIX file system objects are read, write, and execute/search, and read and write for UNIX IPC objects. z/OS file system objects provide either access control based on the permission bits associated with a file, or based on access control lists, which are upward-compatible with the permission bits algorithm and implement the recommendations from Portable Operating System Interface for UNIX (POSIX) 1003.1e draft 17.

1.5.2.3 Auditing

The TOE provides an auditing capability that allows generating audit records for security-critical events. RACF provides a number of logging and reporting functions that allow resource owners and auditors to identify users who attempt to access resources. Audit records are collected by the System Management Facilities (SMF) into an audit trail, which is protected from unauthorized modification or deletion by the DAC mechanisms. This audit trail can reside directly in MVS data sets, or in an MVS log stream (which can be automatically off-loaded into MVS data sets), as configured by the administrator.

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

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

RACF always generates audit records for such events as unauthorized attempts to access the system or changes to the status of the RACF database. The security administrator, auditors, and other users with appropriate authorization can configure which additional optional security events are to be logged. In addition to writing records to the audit trail, messages can be sent to the security console to immediately alert operators of detected policy violations. RACF provides SMF records for all RACF-protected resources (either “traditional” or z/OS UNIX-based) .

For reporting, auditors can unload all or selected parts of the SMF data for further analysis in a human-readable formats and can then upload the data to a query or reporting package, such as DFSORT™ if desired.

1.5.2.4 Security management

z/OS provides a set of commands and options to adequately manage the TOE's security functions. Additionally, the TOE provides the capability of managing users, groups of users as well as general resource profiles.

The TOE recognizes several authorities that are able to perform the different management tasks related to the TOE's security:

- General security options are managed by security administrators.
- Management of users and their security attributes is performed by security administrators. Management of groups (and to some extent users) can be delegated to group security administrators.
- Users can change their own passwords or password phrases, their default groups, and their user names (but not their user IDs).
- Auditors manage the parameters of the audit system (a list of audited events, for example) and can analyze the audit trail.
- Security administrators can define what audit records are captured by the system.

- Discretionary access rights to protected resources are managed by the owners of the applicable profiles (or UNIX objects) or by security administrators.

1.5.2.5 Cryptographic Support

The TOE provides cryptographic functions by the ICSF subsystem. ICSF uses cryptographic hardware provided by the operational environment to provide and support cryptographic functions. The TOE implements TLS Version 1.2 as well SSH version 2 for communication and remote access (see also below).

All key material used for cryptographic functions described in this Security Target when in volatile memory are in memory that is assigned to and is accessible by the TSF only. These keys are finally destructed when the power to the memory is removed. Non-volatile keys are deleted by removing the abstraction that represents this key.

1.5.2.6 Communications Security

z/OS provides means of secure communication between systems sharing the same security policy.

In its evaluated configuration, z/OS supports trusted communication channels for TCP/IP connections. The confidentiality and integrity of network connections are assured by Transport Layer Security (TLS) encrypted communication for TCP/IP connections (Version 1.2 [RFC5246]), which can be used explicitly by applications or applied transparently to their communications (AT-TLS) without changing the applications using it (assuming the applications that do not make use of the TLS capabilities that allow clients to authenticate to the system using a client-supplied X.509 digital certificate. If applications accept client certificates then they do need to have specific TLS-related processing within the applications).

z/OS also supports the SSH v2 protocol and the ssh-daemon provided services of ssh (secure shell), scp (secure copy), and sftp (secure ftp) ([RFC4253])

1.5.2.7 TSF protection

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

- Privileged processor instructions are only available to programs running in the processor's supervisor state
- Semi-privileged instructions are only available to programs running in an execution environment that is established and authorized by the TSF
- While in operation, all address spaces, as well as the data and tasks contained therein, are protected by the memory protection mechanisms of the underlying abstract machine

The TOE's address space management ensures that programs running in problem state cannot access protected memory or resources that belong to other address spaces.

Access to system services - through supervisor call (SVC) or program call (PC) instructions, for example - is controlled by the system, which requires that subjects who want to perform security-relevant tasks be authorized appropriately.

The hardware and firmware components that provide the abstract machine for the TOE are required to be physically protected from unauthorized access. The z/OS Base Control Program mediates all access to the TOE's hardware resources themselves, other than program-visible CPU instruction functions.

Tools are provided in the TOE environment to allow authorized administrators to check the correct operation of the underlying abstract machine.

In addition to the protection mechanism of the underlying abstract machine, the TOE also uses software mechanisms like the authorized program facility (APF) or specific privileges for programs in the UNIX system services environment to protect the TSF.

In addition the TOE provides the following mechanisms to protect its TSF:

- Address space layout randomization (ASLR)
- Stack buffer overflow protection
- Verification of integrity of the IPL process
- Trusted software updates using digital signatures

1.5.2.8 Confidentiality Protection of Data Sets

With z/OS confidentiality protection of data sets, users can encrypt data at rest without requiring application changes. z/OS data set encryption through RACF commands and SMS policies allows the administrator to identify the data sets or groups of data sets that require encryption. The administrator can specify an encryption key label, which refers to an encryption key. Both the key label and encryption key must exist in the ICSF key repository (CKDS). With data set encryption, the administrator is able to protect viewing the data in the clear. This is based on access to the key label that is associated with the data set and used by the access methods to encrypt and decrypt the data.

1.5.3 Configurations

1.5.3.1 Software Configuration

The Target of Evaluation, IBM z/OS Version 2 Release 5, consists of:

- IBM z/OS Version 2 Release 5 (V2R5) Common Criteria Evaluated Base Package:
  - IBM z/OS Version 2 Release 5 (z/OS V2R5, program number 5650-ZOS),
- The following APARs (or their associated PTFs):
  - APAR OA64593 (DOC APAR)
  - APAR OA66552 (DOC APAR)
  - APAR OA66005
  - APAR OA65807

The z/OS V2R5 Common Criteria Evaluated Base package must be installed according to the directions delivered with the media and configured according to the instructions in Chapter 7, “The evaluated configuration for the Common Criteria” in z/OS Planning for Multilevel Security and the Common Criteria ([MLSGUIDE]).

For information in the required and optional software and additional required and mandatory configuration guidance, please refer to [MLSGUIDE], chapter 7.

1.5.3.2 Hardware Configuration

The following assumptions about the technical environment in which the TOE is intended to be used are made: The TOE is running a logical partition provided by PR/SM or a certified version of z/VM on one of the following z System™ processors:

- IBM z16 with CPACF DES/TDES Enablement Feature 3863 active, with Crypto Express8S (CEX8) cards.

The following peripherals can be used with the TOE:

- All terminals that are supported by the TOE.

- Printers:
  - Any printer that is supported by the TOE.
- All storage devices and backup devices supported by the TOE, such as:
  - Direct access storage devices (DASDs), except RVA devices.
  - Tape drives (including encrypting tape drives, though this evaluation has not specifically examined those cryptographic functions).
- All Ethernet and token-ring network adapters that are supported by the TOE.

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

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:

- [GPOSPPv4.3]: Operating System Protection Profile. Version 4.3 as of 2022-09-27; exact conformance.
- [TLSPKGv1.1]: Functional Package for Transport Layer Security (TLS). Version 1.1 as of 2019-03-01; exact conformance.
- [SSHPKGv1.0]: Functional Package for Secure Shell (SSH). Version 1.0 as of 2021-05-13; exact conformance.

Common Criteria [CC] version 3.1 revision 5 is the basis for this conformance claim.

2.1 Protection Profile Tailoring and Additions

2.1.1 Operating System Protection Profile ([GPOSPPv4.3])

Table 1 contains the NIAP Technical Decisions (TDs) for this protection profile at the time of the evaluation and a statement of applicability to the evaluation.

Table 1: NIAP TDs for GPOSPPv4.3

NIAP TD	TD description	Applicable?	Non-applicability rationale
TD0839	Clarification for Local Administration in FTP_TRP.1.3	Yes	
TD0821	Corrections to ECD for PP_OS_V4.3	Yes	
TD0812	Updated CC Conformance Claims in PP_OS_V4.3	Yes	
TD0809	Update to FCS_COP.1/SIGN for CNSA 1.0 compliance with Secure Boot exception	Yes	
TD0789	Correction to TLS Selection in FIA_X509_EXT.2.1	Yes	
TD0773	Updates to FIA_X509_EXT.1 for Exception Processing and Test Conditions	Yes	
TD0713	Functional Package SFR mappings to objectives	Yes	
TD0712	Support for Bluetooth Standard 5.3	No	Not supported by the TOE.
TD0701	Incomplete selection reference in FCS_CKM_EXT.4 TSS activities	Yes	
TD0696	Removal of 160 bit selection from FCS_COP.1/HASH & FCS_COP.1/ KEYHMAC	Yes	
TD0693	Typos in OSPP 4.3	Yes	
TD0691	OSPP 4.3 Conditional authentication testing	Yes	

NIAP TD	TD description	Applicable?	Non-applicability rationale
TD0675	Make FPT_W^X_EXT.1 Optional	Yes	

2.1.2 Functional Package for Transport Layer Security (TLS) ([TLSPKGv1.1])

Table 2 contains the NIAP Technical Decisions (TDs) for this functional package at the time of the evaluation and a statement of applicability to the evaluation.

Table 2: NIAP TDs for TLSPKGv1.1

NIAP TD	TD description	Applicable?	Non-applicability rationale
TD0779	Updated Session Resumption Support in TLS package V1.1	Yes	
TD0770	TLSS.2 connection with no client cert	Yes	
TD0739	PKG_TLS_V1.1 has 2 different publication dates	Yes	
TD0726	Corrections to (D)TLSS SFRs in TLS 1.1 FP	Yes	
TD0513	CA Certificate loading	Yes	
TD0499	Testing with pinned certificates	Yes	
TD0469	Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1	Yes	
TD0442	Updated TLS Ciphersuites for TLS Package	Yes	

2.1.3 Functional Package for Secure Shell (SSH) ([SSHPKGv1.0])

Table 3 contains the NIAP Technical Decisions (TDs) for this PP-Module at the time of the evaluation and a statement of applicability to the evaluation.

Table 3: NIAP TDs for SSHPKGv1.0

NIAP TD	TD description	Applicable?	Non-applicability rationale
TD0777	Clarification to Selections for Auditable Events for FCS_SSH_EXT.1	Yes	
TD0732	FCS_SSHS_EXT.1.3 Test 2 Update.	Yes	
TD0695	Choice of 128 or 256 bit size in AES-CTR in SSH Functional Package.	Yes	
TD0682	Addressing Ambiguity in FCS_SSHS_EXT.1 Tests	Yes	

3 Security Problem Definition

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 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, 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 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.

3.1.3 Threats countered by the TOE

T.NETWORK_ATTACK

An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may engage in communications with applications and services running on or part of the OS with the intent of compromise. Engagement may consist of altering existing legitimate communications.

T.NETWORK_EAVESDROP

An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may monitor and gain access to data exchanged between applications and services that are running on or part of the OS.

T.LOCAL_ATTACK

An attacker may compromise applications running on the OS. The compromised application may provide maliciously formatted input to the OS through a variety of channels including unprivileged system calls and messaging via the file system.

T.LIMITED_PHYSICAL_ACCESS

An attacker may attempt to access data on the OS while having a limited amount of time with the physical device.

3.2 Assumptions

The specific conditions below are assumed to exist in a PP-conformant TOE environment.

3.2.1 Intended usage of the TOE

A.PLATFORM

The OS relies upon a trustworthy computing platform for its execution. This underlying platform is out of scope of this PP.

A.PROPER_USER

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

A.PROPER_ADMIN

The administrator of the OS is not careless, willfully negligent or hostile, and administers the OS within compliance of the applied enterprise security policy.

4 Security Objectives

The following sections describe the security objectives of the Operating System Protection Profile.

4.1 Objectives for the TOE

O.ACCOUNTABILITY

Conformant OSes ensure that information exists that allows administrators to discover unintentional issues with the configuration and operation of the operating system and discover its cause. Gathering event information and immediately transmitting it to another system can also enable incident response in the event of system compromise.

O.INTEGRITY

Conformant OSes ensure the integrity of their update packages. OSes are seldom if ever shipped without errors, and the ability to deploy patches and updates with integrity is critical to enterprise network security. Conformant OSes provide execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems.

O.MANAGEMENT

To facilitate management by users and the enterprise, conformant OSes provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration and application execution control.

O.PROTECTED_STORAGE

To address the issue of loss of confidentiality of credentials in the event of loss of physical control of the storage medium, conformant OSes provide data-at-rest protection for credentials. Conformant OSes also provide access controls which allow users to keep their files private from other users of the same system.

O.PROTECTED_COMMS

To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant OSes provide mechanisms to create trusted channels for CSP and sensitive data. Both CSP and sensitive data should not be exposed outside of the platform.

4.2 Objectives for the Operational Environment

The following objectives are to be met by the operational environment of the TOE.

OE.PLATFORM

The OS relies on being installed on trusted hardware.

OE.PROPER_USER

The user of the OS is not willfully negligent or hostile, and uses the software within compliance of the applied enterprise security policy. Standard user accounts are provisioned in accordance with the least privilege model. Users requiring higher levels of access should have a separate account dedicated for that use.

OE.PROPER_ADMIN

The administrator of the OS is not careless, willfully negligent or hostile, and administers the OS within compliance of the applied enterprise security policy.

4.3 Security Objectives Rationale

4.3.1 Security Objectives Coverage

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

Table 4: Mapping of security objectives to threats and policies

Objective	Threats / OSPs
O.ACCOUNTABILITY	T.NETWORK_ATTACK T.LOCAL_ATTACK
O.INTEGRITY	T.NETWORK_ATTACK T.LOCAL_ATTACK
O.MANAGEMENT	T.NETWORK_ATTACK T.NETWORK_EAVESDROP
O.PROTECTED_STORAGE	T.LIMITED_PHYSICAL_ACCESS
O.PROTECTED_COMMS	T.NETWORK_ATTACK T.NETWORK_EAVESDROP

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.

Table 5: Mapping of security objectives for the Operational Environment to assumptions, threats and policies

Objective	Assumptions / Threats / OSPs
OE.PLATFORM	A.PLATFORM
OE.PROPER_USER	A.PROPER_USER
OE.PROPER_ADMIN	A.PROPER_ADMIN

4.3.2 Security Objectives Sufficiency

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

Table 6: Sufficiency of objectives countering threats

Threat	Rationale for security objectives
T.NETWORK_ATTACK	The threat T.NETWORK_ATTACK is countered by O.PROTECTED_COMMS as this provides for integrity of transmitted data. The threat T.NETWORK_ATTACK is countered by O.INTEGRITY as this provides for integrity of software that is installed onto the system from the network. The threat T.NETWORK_ATTACK is countered by O.MANAGEMENT as this provides for the ability to configure the OS to defend against network attack.

Threat	Rationale for security objectives
	The threat T.NETWORK_ATTACK is countered by O.ACCOUNTABILITY as this provides a mechanism for the OS to report behavior that may indicate a network attack has occurred.
T.NETWORK_EAVESDROP	The threat T.NETWORK_EAVESDROP is countered by O.PROTECTED_COMMS as this provides for confidentiality of transmitted data. The threat T.NETWORK_EAVESDROP is countered by O.MANAGEMENT as this provides for the ability to configure the OS to protect the confidentiality of its transmitted data.
T.LOCAL_ATTACK	The objective O.INTEGRITY protects against the use of mechanisms that weaken the TOE with regard to attack by other software on the platform. The objective O.ACCOUNTABILITY protects against local attacks by providing a mechanism to report behavior that may indicate a local attack is occurring or has occurred.
T.LIMITED_PHYSICAL_ACCESS	The objective O.PROTECTED_STORAGE protects against unauthorized attempts to access physical storage used by the TOE.

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.

Table 7: Sufficiency of objectives holding assumptions

Assumption	Rationale for security objectives
A.PLATFORM	The operational environment objective OE.PLATFORM is realized through A.PLATFORM.
A.PROPER_USER	The operational environment objective OE.PROPER_USER is realized through A.PROPER_USER.
A.PROPER_ADMIN	The operational environment objective OE.PROPER_ADMIN is realized through A.PROPER_ADMIN.

5 Extended Components Definition

Extended component definitions (ECDs) are provided by the PP and FPs to which this ST conforms. Please refer to section 2 for more information on the PP and FPs this ST claims conformance to.

6 Security Requirements

6.1 TOE Security Functional Requirements

All of the following SFRs are derived from the [GPOSPPv4.3] and its related packages [SSHPKGv1.0] and [TLSPKGv1.1].

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**.

Table 8: SFRs for the TOE

Security functional class	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
FAU - Security audit	FAU_GEN.1 Audit Data Generation (Refined)		GPOSPPv4.3	No	No	Yes	Yes
FCS - Cryptographic support	FCS_CKM.1 Cryptographic Key Generation (Refined)		GPOSPPv4.3	No	No	No	Yes
	FCS_CKM.2 Cryptographic Key Establishment (Refined)		GPOSPPv4.3	No	No	No	Yes
	FCS_CKM_EXT.4 Cryptographic Key Destruction		GPOSPPv4.3	No	No	No	Yes
	FCS_COP.1/ENCRYPT Cryptographic Operation - Encryption/Decryption (Refined)	FCS_COP.1	GPOSPPv4.3	Yes	No	No	Yes
	FCS_COP.1/HASH Cryptographic Operation - Hashing (Refined)	FCS_COP.1	GPOSPPv4.3	Yes	No	No	Yes
	FCS_COP.1/SIGN Cryptographic Operation - Signing (Refined)	FCS_COP.1	GPOSPPv4.3	Yes	No	No	Yes
	FCS_COP.1/KEYHMAC Cryptographic Operation - Keyed-Hash Message Authentication (Refined)	FCS_COP.1	GPOSPPv4.3	Yes	No	Yes	Yes
	FCS_RBG_EXT.1 Random Bit Generation		GPOSPPv4.3	No	No	No	Yes
	FCS_STO_EXT.1 Storage of Sensitive Data		GPOSPPv4.3	No	No	No	No
	FCS_TLS_EXT.1 TLS Protocol		TLSPKGv1.1	No	No	No	Yes
	FCS_TLSC_EXT.1 TLS Client Protocol		TLSPKGv1.1	No	No	No	Yes
	FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication		TLSPKGv1.1	No	No	No	No

Security functional class	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FCS_TLSC_EXT.5 TLS Client Support for Supported Groups Extension		TLSPKGv1.1	No	No	No	Yes
	FCS_TLSS_EXT.1 TLS Server Protocol		TLSPKGv1.1	No	No	No	Yes
	FCS_TLSS_EXT.2 Server Support for Mutual Authentication		TLSPKGv1.1	No	No	No	Yes
	FCS_SSH_EXT.1 SSH Protocol		SSHPKGv1.0	No	No	Yes	Yes
	FCS_SSHC_EXT.1 SSH Protocol - Client		SSHPKGv1.0	No	No	No	Yes
	FCS_SSHS_EXT.1 SSH Protocol - Server		SSHPKGv1.0	No	No	No	Yes
FDP - User data protection	FDP_ACF_EXT.1 Access Controls for Protecting User Data		GPOSPPv4.3	No	No	No	No
FIA - Identification and authentication	FIA_AFL.1 Authentication failure handling (Refined)		CC Part 2	No	No	Yes	Yes
	FIA_UAU.5 Multiple Authentication Mechanisms (Refined)		CC Part 2	No	No	Yes	Yes
	FIA_X509_EXT.1 X.509 Certificate Validation		GPOSPPv4.3	No	Yes	Yes	Yes
	FIA_X509_EXT.2 X.509 Certificate Authentication		GPOSPPv4.3	No	No	No	Yes
FMT - Security management	FMT_MOF_EXT.1 Management of security functions behavior		GPOSPPv4.3	No	Yes	No	No
	FMT_SMF_EXT.1 Specification of Management Functions		GPOSPPv4.3	No	No	No	Yes
FPT - Protection of the TSF	FPT_ACF_EXT.1 Access controls		GPOSPPv4.3	No	No	Yes	No
	FPT_ASLR_EXT.1 Address Space Layout Randomization		GPOSPPv4.3	No	No	Yes	Yes
	FPT_SBOP_EXT.1 Stack Buffer Overflow Protection		GPOSPPv4.3	No	No	No	Yes
	FPT_TST_EXT.1 Boot Integrity		GPOSPPv4.3	No	No	Yes	Yes
	FPT_TUD_EXT.1 Trusted Update		GPOSPPv4.3	No	No	No	Yes
	FPT_TUD_EXT.2 Trusted Update for Application Software		GPOSPPv4.3	No	No	No	No
FTA - TOE access	FTA_TAB.1 Default TOE access banners		GPOSPPv4.3	No	No	No	No
FTP - Trusted path/channels	FTP_ITC_EXT.1 Trusted channel communication		GPOSPPv4.3	No	No	Yes	Yes

Security functional class	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FTP_TRP.1 Trusted Path (Refined)		CC Part 2	No	No	No	Yes

6.1.1 Security audit (FAU)

6.1.1.1 FAU_GEN.1 Audit Data Generation (Refined)

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

a) Start-up and shut-down of the audit functions;

b) All auditable events for the not specified level of audit; and

c)
- Authentication events (Success/Failure);
- Use of privileged/special rights events (Successful and unsuccessful security, audit, and configuration changes);
- Privilege or role escalation events (Success/Failure);
- ○ **File and object events (Successful and unsuccessful attempts to create, access, delete, modify, modify permissions)**
  - ○ **User and Group management events (Successful and unsuccessful add, delete, modify, disable, enable, and credential change)**
  - ○ **Audit and log data access events (Success/Failure)**
  - ○ **Administrator or root-level access events (Success/Failure)**
  - ○ **FCS_SSH_EXT.1: None**

.

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, **user identity, if applicable**

.

6.1.2 Cryptographic support (FCS)

6.1.2.1 FCS_CKM.1 Cryptographic Key Generation (Refined)

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

- **ECC schemes using "NIST curves" P-384 and P-521 that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4**

6.1.2.2 FCS_CKM.2 Cryptographic Key Establishment (Refined)

FCS_CKM.2.1 The OS shall implement functionality to perform cryptographic key establishment in accordance with a specified cryptographic key establishment method:

- **Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 3, “Recommendation for Pair- Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”**

6.1.2.3 FCS_CKM_EXT.4 Cryptographic Key Destruction

FCS_CKM_EXT.4.1 The OS shall destroy cryptographic keys and key material in accordance with a specified cryptographic key destruction method

- **For volatile memory, the destruction shall be executed by a**
  - **removal of power to the memory**
- **For non-volatile memory that consists of the invocation of an interface provided by the underlying platform that**
  - **instructs the underlying platform to destroy the abstraction that represents the key**

.

FCS_CKM_EXT.4.2 The OS shall destroy all keys and key material when no longer needed.

6.1.2.4 FCS_COP.1/ENCRYPT Cryptographic Operation - Encryption/ Decryption (Refined)

FCS_COP.1.1/ENCRYPT The OS shall perform encryption/decryption services for data in accordance with a specified cryptographic algorithm

- **AES-XTS (as defined in NIST SP 800-38E)**
- **AES-CBC (as defined in NIST SP 800-38A)**
- **AES-CTR (as defined in NIST SP 800-38A)**

and

- **AES-GCMP-256 (as defined in NIST SP 800-38D and IEEE 802.11ac-2013)**
- **no other modes**

and cryptographic key sizes 256 bit.

6.1.2.5 FCS_COP.1/HASH Cryptographic Operation - Hashing (Refined)

FCS_COP.1.1/HASH The OS shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm

- **SHA-256**
- **SHA-384**
- **SHA-512**

and message digest sizes

- **256 bits**
- **384 bits**
- **512 bits**

that meet the following: FIPS Pub 180-4.

6.1.2.6 FCS_COP.1/SIGN Cryptographic Operation - Signing (Refined)

FCS_COP.1.1/SIGN The OS shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm

- **RSA schemes using cryptographic key sizes of 3072-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 4**
- **ECDSA schemes using "NIST curves" P-384 and P-521 that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5**

.

Application Note: *The TOE in its evaluated configuration and operation does not generate RSA based signatures.*

Application Note: *The TOE verifies RSA based signatures in the context of trusted update. ECDSA signatures are used for communication security and trusted boot.*

Application Note: *It should be noted, that* [FIPS186-4] *describes RSA based signatures in section 5 and ECDSA based signatures in section 6.*

6.1.2.7 FCS_COP.1/KEYHMAC Cryptographic Operation - Keyed-Hash Message Authentication (Refined)

FCS_COP.1.1/KEYHMAC The OS shall perform keyed-hash message authentication services in accordance with a specified cryptographic algorithm **SHA-256**, **SHA-384**, **SHA-512** with key sizes **between 112 and 2048 bits** and message digest sizes **256 bits**, **384 bits**, **512 bits** that meet the following: FIPS Pub 198-1 The Keyed-Hash Message Authentication Code and FIPS Pub 180-4 Secure Hash Standard.

Application Note: '

The HMAC key is generated by the 'KeyGenerate2' (CSNBKGN2) function of ICSF which allows for key sizes between 112 and 2048 bit. System SSL generates HMACs during TLS processing.

6.1.2.8 FCS_RBG_EXT.1 Random Bit Generation

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

- **Hash_DRBG (any)**

.

FCS_RBG_EXT.1.2 The deterministic RBG used by the OS shall be seeded by an entropy source that accumulates entropy from a

- **platform-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.

6.1.2.9 FCS_STO_EXT.1 Storage of Sensitive Data

FCS_STO_EXT.1.1 The OS shall implement functionality to encrypt sensitive data stored in non-volatile storage and provide interfaces to applications to invoke this functionality.

6.1.2.10 FCS_TLS_EXT.1 TLS Protocol

FCS_TLS_EXT.1.1 The product shall implement

- **TLS as a client**
- **TLS as a server**

.

6.1.2.11 FCS_TLSC_EXT.1 TLS Client Protocol

FCS_TLSC_EXT.1.1 The OS shall implement TLS 1.2 ([RFC5246]) and **no earlier TLS versions** supporting the following cipher suites:

- **TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in [RFC5289]**
- **TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in [RFC5289]**

and also functionality for

- **mutual authentication**

FCS_TLSC_EXT.1.2 The OS shall verify that the presented identifier matches the reference identifier according to [RFC6125].

FCS_TLSC_EXT.1.3 The product shall not establish a trusted channel if the server certificate is invalid

- **with no exceptions**

.

6.1.2.12 FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication

FCS_TLSC_EXT.2.1 The product shall support mutual authentication using X.509v3 certificates.

6.1.2.13 FCS_TLSC_EXT.5 TLS Client Support for Supported Groups Extension

FCS_TLSC_EXT.5.1 The product shall present the Supported Groups Extension in the Client Hello with the following supported groups: **secp384r1**, **secp521r1**.

6.1.2.14 FCS_TLSS_EXT.1 TLS Server Protocol

FCS_TLSS_EXT.1.1 The product shall implement TLS 1.2 ([RFC5246]) and **no earlier TLS versions** as a server that supports the cipher suites

- **TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in [RFC5289] ,**
- **TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in [RFC5289] ,**

and also supports functionality for

- **no session resumption or session tickets,**

and

- **mutual authentication.**

Application Note: *Note that the TOE does not support TLS 1.1 in its evaluated configuration.*

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

FCS_TLSS_EXT.1.3 The product shall perform key establishment for TLS using

- **ECDHE parameters using elliptic curves**
  - **secp384r1**
  - **secp521r1**

  and no other curves ,

.

6.1.2.15 FCS_TLSS_EXT.2 Server Support for Mutual Authentication

FCS_TLSS_EXT.2.1 The product shall support authentication of TLS clients using X.509v3 certificates.

FCS_TLSS_EXT.2.2 The product shall **not establish a trusted channel** if the client certificate is invalid.

FCS_TLSS_EXT.2.3 The product shall not establish a trusted channel if the Distinguished Name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match one of the expected identifiers for the client.

6.1.2.16 FCS_SSH_EXT.1 SSH Protocol

FCS_SSH_EXT.1.1 The TOE shall implement SSH as a **client**, **server** that complies with RFCs 4251, 4252, 4253, 4254 and **4344**, **5647**, **5656**, **no other RFCs** and no other standard.

FCS_SSH_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods:

- **"password" ([RFC4252])**
- **"publickey" ([RFC4252]):**
  - **ecdsa-sha2-nistp384 ([RFC5656]),**
  - **ecdsa-sha2-nistp521 ([RFC5656]),**

and no other methods.

FCS_SSH_EXT.1.3 The TSF shall ensure that, as described in [RFC4253], packets greater than **262144** bytes in an SSH transport connection are dropped.

FCS_SSH_EXT.1.4 The TSF shall protect data in transit from unauthorised disclosure using the following mechanisms:

- **aes256-ctr ([RFC4344])**
- **aes256-cbc ([RFC4253])**
- **AEAD_AES_256_GCM ([RFC5647])**

and no other mechanisms.

FCS_SSH_EXT.1.5 The TSF shall protect data in transit from modification, deletion, and insertion using:

- **hmac-sha2-256 ([RFC6668])**
- **hmac-sha2-512 ([RFC6668])**
- **AEAD_AES_256_GCM ([RFC5647])**
- **implicit**

and no other mechanisms.

FCS_SSH_EXT.1.6 The TSF shall establish a shared secret with its peer using:

- **ecdh-sha2-nistp384 ([RFC5656])**
- **ecdh-sha2-nistp521 ([RFC5656])**

and no other mechanisms.

FCS_SSH_EXT.1.7 The TSF shall use SSH KDF as defined in

- **[RFC5656] (Section 4)**

to derive the following cryptographic keys from a shared secret: session keys.

FCS_SSH_EXT.1.8 The TSF shall ensure that

- **a rekey of the session keys**

occurs when any of the following thresholds are met:

- one hour connection time
- no more than one gigabyte of transmitted data
- or no more than one gigabyte of received data.

6.1.2.17 FCS_SSHC_EXT.1 SSH Protocol - Client

FCS_SSHC_EXT.1.1 The TSF shall authenticate its peer (SSH server) using:

- **using a local database by associating each host name with a public key corresponding to the following list:**
  - **ecdsa-sha2-nistp384 (** [**RFC5656**] **),**
  - **ecdsa-sha2-nistp521 (** [**RFC5656**] **),**

as described in RFC 4251 section 4.1.

6.1.2.18 FCS_SSHS_EXT.1 SSH Protocol - Server

FCS_SSHS_EXT.1.1 The TSF shall authenticate itself to its peer (SSH Client) using:

- **ecdsa-sha2-nistp384 (** [**RFC5656**] **),**
- **ecdsa-sha2-nistp521 (** [**RFC5656**] **),**

.

6.1.3 User data protection (FDP)

6.1.3.1 FDP_ACF_EXT.1 Access Controls for Protecting User Data

FDP_ACF_EXT.1.1 The OS shall implement access controls which can prohibit unprivileged users from accessing files and directories owned by other users.

6.1.4 Identification and authentication (FIA)

6.1.4.1 FIA_AFL.1 Authentication failure handling (Refined)

FIA_AFL.1.1 The OS shall detect when

- **an administrator configurable positive integer within a range of positive integers from 1 to 255**

unsuccessful authentication attempts occur related to events with

- **authentication based on user name and password**

.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts for an account has been met, the OS shall: **Account Disablement** .

6.1.4.2 FIA_UAU.5 Multiple Authentication Mechanisms (Refined)

FIA_UAU.5.1 The OS shall provide the following authentication mechanisms

- **authentication based on user name and password**
- **for use in SSH only, SSH public key-based authentication as specified by the Functional Package for Secure Shell (SSH), version 1.0**

to support user authentication.

FIA_UAU.5.2 The OS shall authenticate any user's claimed identity according to the **following rule:**

- **authentication on (virtual 3270) terminals and the operating system console is based on user name and password or passphrase,**
- **authentication via the SSHv2 protocol using:**
  - **user name and password authentication;**
  - **public key based authentication**

.

6.1.4.3 FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1.1 The OS shall implement functionality to validate certificates in accordance with the following rules:

- [RFC5280] certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The OS 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.
- The TSF shall validate that any CA certificate includes "Certificate Signing" as a purpose the key usage field
- The OS shall validate the revocation status of the certificate using **OCSP as specified in [RFC6960] ,** with **the exception of the TOE's TCP/IP stack being configured to not validate certificates using an OCSP service**.
- The OS shall validate the extendedKeyUsage field according to the following rules:
  - 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.
  - 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.
  - 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 EKU field.
  - ~~S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the EKU field.~~
  - S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the EKU field.
  - 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 EKU field.

- ~~Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field. (conditional)~~

.

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

6.1.4.4 FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1 The OS shall use X.509v3 certificates as defined by [RFC5280] to support authentication **TLS** connections.

6.1.5 Security management (FMT)

6.1.5.1 FMT_MOF_EXT.1 Management of security functions behavior

FMT_MOF_EXT.1.1 The OS shall restrict the ability to perform the function indicated ~~in the "Administrator" column~~ in FMT_SMF_EXT.1 to the administrator.

6.1.5.2 FMT_SMF_EXT.1 Specification of Management Functions

FMT_SMF_EXT.1.1 The OS shall be capable of performing the following management functions:

a) Enable/disable **session timeout**
b) Configure **session** inactivity timeout
c) Configure local audit storage capacity
d) Configure minimum password length
e) Configure minimum number of special characters in password
f) Configure minimum number of numeric characters in password
g) Configure minimum number of uppercase characters in password
h) Configure lockout policy for unsuccessful authentication attempts through **timeouts between attempts**, **limiting number of attempts during a time period**
i) Configure host-based firewall
j) Configure audit rules

.

6.1.6 Protection of the TSF (FPT)

6.1.6.1 FPT_ACF_EXT.1 Access controls

FPT_ACF_EXT.1.1 The OS shall implement access controls which prohibit unprivileged users from modifying:

a) Kernel and its drivers/modules

b) Security audit logs
c) Shared libraries
d) System executables
e) System configuration files
f) **protected MVS data sets and other z/OS objects, protected UNIX file system objects, protected UNIX IPC objects**

.

FPT_ACF_EXT.1.2 The OS shall implement access controls which prohibit unprivileged users from reading:

a) Security audit logs
b) System-wide credential repositories
c) **protected MVS data sets, protected UNIX file system objects**

.

6.1.6.2 FPT_ASLR_EXT.1 Address Space Layout Randomization

FPT_ASLR_EXT.1.1 The OS shall always randomize process address space memory locations with **8** bits of entropy except for **the lower 16 MB of memory** .

6.1.6.3 FPT_SBOP_EXT.1 Stack Buffer Overflow Protection

FPT_SBOP_EXT.1.1 The OS shall **employ stack-based buffer overflow protections**.

6.1.6.4 FPT_TST_EXT.1 Boot Integrity

FPT_TST_EXT.1.1 The OS shall verify the integrity of the bootchain up through the OS kernel and **all executable code stored in mutable media**,

- **IPL Text (IEAIPL00) (validated by firmware (Z Bootloader))**
- **All load modules through LPA creation:**
  - **IRIMs and RIMs (up through LPA creation)**
  - **SYS1.NUCLEUS load modules including z/OS Nucleus (IEANUCxx)**
  - **LPA load modules (LPALST concatenation) and MLPA/FLPA load modules**

, **no other executable code** prior to its execution through the use of **a digital signature using a hardware-protected asymmetric key** .

6.1.6.5 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1 The OS shall provide the ability to check for updates to the OS software itself and shall use a digital signature scheme specified in FCS_COP.1/SIGN to validate the authenticity of the response.

FPT_TUD_EXT.1.2 The OS shall **cryptographically verify** updates to itself using a digital signature prior to installation using schemes specified in FCS_COP.1/SIGN.

6.1.6.6 FPT_TUD_EXT.2 Trusted Update for Application Software

FPT_TUD_EXT.2.1 The OS shall provide the ability to check for updates to application software and shall use a digital signature scheme specified in FCS_COP.1/SIGN to validate the authenticity of the response.

FPT_TUD_EXT.2.2 The OS shall cryptographically verify the integrity of updates to applications using a digital signature specified by FCS_COP.1/SIGN prior to installation.

6.1.7 TOE access (FTA)

6.1.7.1 FTA_TAB.1 Default TOE access banners

FTA_TAB.1.1 Before establishing a user session, the **OS** shall display an advisory warning message regarding unauthorized use of the OS.

6.1.8 Trusted path/channels (FTP)

6.1.8.1 FTP_ITC_EXT.1 Trusted channel communication

FTP_ITC_EXT.1.1 The OS shall use

- **TLS as conforming to the Functional Package for Transport Layer Security (TLS), version 1.1 as a client**, **server**

and

- **SSH as conforming to the Functional Package for Secure Shell (SSH), version 1.0 as a client**, **server**
- **no other protocols**

to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities:

- **remote access**

that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

6.1.8.2 FTP_TRP.1 Trusted Path (Refined)

FTP_TRP.1.1 The OS shall provide a communication path between itself and **remote**, **local** users that is logically distinct from other communication paths and provides assured identification of its endpoints and protection of the communicated data from modification, disclosure.

FTP_TRP.1.2 The OS shall permit **the TSF**, **local users**, **remote users** to initiate communication via the trusted path.

FTP_TRP.1.3 The OS shall require use of the trusted path for **all remote administrative actions**.

6.2 Security Functional Requirements Rationale

All SFRs are reproduced exactly from the PP and the extended packages for exact conformance. Please refer to the claimed PPs and functional packages for a rationale on how the SFRs are suitable to achieve the security objectives:

- [GPOSPPv4.3]: Section 5.1.8
- [TLSPKGv1.1]: N/A
- [SSHPKGv1.0]: N/A

6.3 Security Assurance Requirements

The security assurance requirements (SARs) for the TOE are defined in the GPOSPPV4.3 protection profile.

6.4 Security Assurance Requirements Rationale

The Security Target claims exact compliance to the Protection Profile, including to the Security Assurance Requirements. Please refer to the claimed PPs and functional packages for a rationale why the respective SARs have been chosen.

7 TOE Summary Specification

7.1 TSS Security Assurance Evaluation Activity

7.1.1 Timely security updates (ALC_TSU_EXT.1)

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/OS. Guaranteed response times depend on the selected service level agreement which is outlined in IBM Enterprise Support and Preferred Care options for IBM Z. The security incident and response process allows customers to directly interact with the IBM team via a central contact system documented at the IBM Support Guide 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 IBM Enterprise Support and Preferred Care options for IBM Z. For dedicated or suspected security issues, IBM provides contacts as well as descriptions of processes on the IBM Security Vulnerability Management website.

7.2 Mapping SFR to TSS

The Protection Profile mandates various specific information to be supplied in the TSS to cover aspects of the SFRs. The following table enumerates the SFRs (from the Protection Profile and the extended packages) and adds references into the TSS to document these SFRs.

SFR	Coverage in TSS
FAU_GEN.1	See section Audit (FAU).
FCS_CKM.1	See section General Cryptography.
FCS_CKM.2	See section General Cryptography.
FCS_CKM_EXT.4	See section Cryptographic Key Destruction.
FCS_COP.1/ENCRYPT	See section General Cryptography.
FCS_COP.1/HASH	See section General Cryptography.
FCS_COP.1/SIGN	See section General Cryptography.
FCS_COP.1/KEYHMAC	See section General Cryptography.
FCS_RBG_EXT.1	See section Random Number Generation.
FCS_STO_EXT.1	See section Confidentiality Protection of Data Sets.
FCS_TLSC_EXT.1	See section Communication Security and in particular section System SSL.
FCS_TLSC_EXT.2	See section Communication Security and in particular section System SSL..
FCS_TLSC_EXT.5	See section Communication Security and in particular section System SSL.
FCS_TLSS_EXT.1	See section Communication Security and in particular section System SSL.
FCS_TLSS_EXT.2	See section Communication Security and in particular section System SSL.

SFR	Coverage in TSS
FCS_SSH_EXT.1	See section Communication Security and in particular section OpenSSH..
FCS_SSHC_EXT.1	See section Communication Security and in particular section OpenSSH.
FCS_SSHS_EXT.1	See section Communication Security and in particular section OpenSSH.
FDP_ACF_EXT.1	See section Discretionary Access Control.
FIA_AFL.1	See section Password Quality.
FIA_UAU.5	See section Identification and Authentication (FIA, FTA), Authentication via Public/ Private Key (SSH) and OpenSSH.
FIA_X509_EXT.1	See section Authentication via Client Digital Certificates.
FIA_X509_EXT.2	See section Authentication via Client Digital Certificates.
FMT_MOF_EXT.1	See section Security Management (FMT), RACF configuration and management, RACF Certificate and Key Management, Automatic Logout of Sessions, Management of Communications Server Functions, Password Quality, Protection of the audit trail and Time Management.
FMT_SMF_EXT.1	See section Security Management (FMT), RACF configuration and management, RACF Certificate and Key Management, Automatic Logout of Sessions, Management of Communications Server Functions, Password Quality, Protection of the audit trail and Time Management.
FPT_ACF_EXT.1	See section Discretionary Access Control.
FPT_ASLR_EXT.1	See section Address Space Layout Randomization.
FPT_SBOP_EXT.1	See section Stack Buffer Overflow Protection.
FPT_TST_EXT.1	See section Trusted Boot/IPL.
FPT_TUD_EXT.1	See section Trusted Update Process.
FPT_TUD_EXT.2	See section Trusted Update Process.
FTA_TAB.1	See section Access Banners.
FTP_ITC_EXT.1	See section Communication Security.
FTP_TRP.1	See section Communication Security.

Table 9: Mapping SFRs to TSS Sections

7.3 TOE Security Functionality

7.3.1 z/OS Basic Principles

7.3.1.1 Hardware Platform

The z/OS operating system is designed to operate on a hardware platform that implements the zArchitecture most of which is defined in the IBM document 'Principles of Operation'. The zArchitecture provides two different processor states - user (unprivileged) and supervisor (privileged) - as well as memory protection features that allow to restrict access of programs to such memory to no access, read-only access, or read and write access. Changing those access attributes of memory can only be done by privileged instructions, which can only be executed when the processor is in supervisor mode.

The current status of a processor including the address of the program instruction currently executed is stored in a processor internal register called the 'Program Status Word' (PSW). This PSW also contains a so called 'Access Key Mask' (AKM) which is used by the processor to determine the access to memory the currently executed instruction has. Access to memory is controlled by matching the current AKM in the PSW with the protection bits each page of memory has.

Among the privileged processor instructions of the zArchitecture are also a few instructions related to I/O operations. Unlike most other hardware platforms the IBM zArchitecture has a well-defined interface to an I/O subsystem where individual device are addressed via a channel and device address and where the operations on a device are defined by 'Channel Command Words' (CCW). Several of them can be combined to a 'Channel Program' that is submitted to the I/O subsystem using the privileged instruction 'START SUBCHANNEL' (SSCH). For an overview on how I/O works on a zSeries platform and which other I/O related privileged instructions exist, the reader is referred to chapters 13 to 17 of the Principle of Operations document.

7.3.1.2 System Initialization (IPL)

The system initialization process prepares the system control program and its environment to do work for the installation. The process essentially consists of:

- System and storage initialization, including the creation of system component address spaces.
- Master scheduler initialization and subsystem initialization.

When the system is initialized and the job entry subsystem is active, the installation can submit jobs for processing with the START or MOUNT command.

The initialization process begins when the system operator selects the LOAD function at the system console. MVS locates all of the usable central storage that is online and available to the system, and creates a virtual environment for the building of various system areas. IPL includes the following major initialization functions:

- Loads the DAT-off nucleus into central storage.
- Loads the DAT-on nucleus into virtual storage so that it spans above and below 16 megabytes (except the prefixed storage area (PSA), which IPL loads at virtual zero).
- Builds the nucleus map, NUCMAP, of the DAT-on nucleus. NUCMAP resides in virtual storage above the nucleus.
- Allocates the system's minimum virtual storage for the system queue area (SQA) and the extended SQA.
- Allocates virtual storage for the extended local system queue area (extended LSQA) for the master scheduler address space.

The system continues the initialization process, interpreting and acting on the system parameters that were specified. NIP (Nucleus Initialization Program) carries out the following major initialization functions:

- Expands the SQA and the extended SQA by the amounts specified on the SQA system parameter.
- Creates the pageable link pack area (PLPA) and the extended PLPA for a cold start IPL; resets tables to match an existing PLPA and extended PLPA for a quick start or a warm start IPL.
- Loads modules into the fixed link pack area (FLPA) or the extended FLPA. Note that NIP carries out this function only if the FIX system parameter is specified.
- Loads modules into the modified link pack area (MLPA) and the extended MLPA. Note that NIP carries out this function only if the MLPA system parameter is specified.

- Allocates virtual storage for the common service area (CSA) and the extended CSA. The amount of storage allocated depends on the values specified on the CSA system parameter at IPL.
- Page protects the: NUCMAP, PLPA and extended PLPA, MLPA and extended MLPA, FLPA and extended FLPA, and portions of the nucleus.

Note: An installation can override page protection of the MLPA and FLPA by specifying NOPROT on the MLPA and FIX system parameters.

7.3.1.2.1 Initial System Address Space Creation

In addition to initializing system areas, z/OS establishes system component address spaces. z/OS establishes an address space for the master scheduler (the master scheduler address space (MSTR)) and other system address spaces for various subsystems and system components. Details about the initial address space creation can be found in [MVSTUNE.G], chapter 1.

7.3.1.2.2 Master Scheduler Initialization (MSTR)

Master scheduler initialization routines initialize system services such as the system log and communications task, and start the master scheduler itself. They also cause creation of the system address space for the job entry subsystem (JES2), and then start the job entry subsystem.

7.3.1.2.3 z/OS Subsystem Initialization

z/OS subsystem initialization is the process of readying a subsystem for use in the system. IEFSSNxx members of SYS1.PARMLIB contain the definitions for the primary subsystems, such as JES2, and possibly the secondary subsystems, such as SMS and DB2. For detailed information about the data contained in IEFSSNxx members for secondary systems, please refer to the installation manual for the specific system.

During system initialization, the defined subsystems are initialized. The system administrator should define the primary subsystem (JES2) first, because other subsystems require the services of the primary subsystem in their initialization routines – problems can occur if subsystems that use the primary subsystem's services in their initialization routines are initialized before the primary subsystem.

After the primary subsystem JES2 is initialized, then the subsystems are initialized in the order in which the IEFSSNxx parmlib members are specified by the SSN parameter. For example, for SSN=(aa,bb) parmlib member IEFSSNaa would be processed before IEFSSNbb.

Note: The storage management subsystem (SMS) is the only subsystem that can be defined before the primary subsystem.

Using IEFSSNxx to initialize the subsystems, the system administrator can specify the name of a subsystem initialization routine to be given control during master scheduler initialization, and the system administrator can specify the input parameter to be passed to the subsystem initialization routine.

7.3.1.2.4 Trusted Boot/IPL

The establishment of a trusted boot within the TOE is a two step process, where only the second is actually performed by the TOE:

1. **Controlled Signing**

   On a controlled, separate **non-TOE** system a "secure build" process initiated by the administrator creates signed IPL Text and load modules using administrator defined keys and stores them on disk. Trusted is built upon the validation of the "incoming" code package

deliverables using IBM's vendor public key. The administrator builds/creates all needed load module executables, and signs them with their own private key (including the IPL Text). The signing is happening in a controlled environment.

The non-TOE system uses SHA2-512 used for hashing in creating the signature, hashing the data to be signed and an Elliptic Curve ECDSA-P521 key used for signing. The RACF Signing Service (using the existing R_PgmSignVer SAF callable service (IRRSPS00)) is used to sign load modules, and by ICKDSF to sign the IPL Text on an IPL volume

The following modules are subject to signing:

- IPL Text (IEAIPL00) (validated by firmware (Z Bootloader))
- All load modules through LPA creation:
  - IRIMs and RIMs (up through LPA creation)
  - SYS1.NUCLEUS load modules including z/OS Nucleus (IEANUCxx)
  - LPA load modules (LPALST concatenation) and MLPA/FLPA load modules

2. **Trusted IPL**

   At IPL time, platform firmware (Z Bootloader) validates the IPL Text, which contains the validation function code which z/OS will use in subsequent load module validation steps

   - Client's (or other) public keys for validation purposes are provided by the z platform firmware
   - The IPL Text is a signed object, signed by the client and validated by the z platform firmware

   {SP-IPL-V2R5.1}

   As z/OS loads the individual authorized load modules during the IPL, z/OS uses the validation function code to validate their signatures using the administrator's defined public keys. **{SP-IPL-V2R5.2}**

   Any validation failures may result in non-restartable wait state termination of the IPL, or not, depending on IPL validation mode. **{SP-IPL-V2R5.3}**

   The z firmware (SE/HMC and LPAR) provides the trusted validation Certificate Store for use in validation processing done by both the Z Bootloader and z/OS. **{SP-IPL-V2R5.4}**

7.3.1.3 Address Spaces in z/OS

Originally the operating system z/OS evolved from was designed for processing batch jobs where each job was executed in its own address space separated from the address spaces of other jobs. Still today the main task of z/OS is to serve those batch jobs in large production environments.

Address spaces in z/OS have some parts in common, but in the evaluated configuration all those parts can only be written by the operating system itself. Some parts of those common areas of all address spaces contain library programs that can be used by all address spaces, some parts contain data written by the operating system for common use by all address spaces. Some of those common data areas may contain critical data and are therefore also protected from being read by unauthorized user programs.

Address spaces are assigned a user identity which is used for security functions like access control and audit. All subjects within an address space are associated with that user-ID. A user-ID is assigned during address space initialization either as part of the user identification and authentication process. A system administrator of an installation may have defined specific address spaces that are started automatically or on request of a system administrator where the specification of the address space

(in terms of the 'Job Control Language') does not require the specific user-ID to be authenticated. This allows the operating system itself to consists also of dedicated address spaces providing operating system services.

7.3.1.4 System Call Interface

A subject within an address space can request operating system services using either the (legacy) SVC instruction or the (more modern) Program Call (PC) or Program Transfer (PT) instructions. The SVC instruction traps into the operating system kernel which determines the type of service requested, while the PC or PT instruction may call operating system services either provided within the caller's address space or provided by another operating system owned address space. In this case the hardware performs a full address space context switch. This allows z/OS to have operating system services being implemented in a way where a system call never executes with the highest hardware privileges but just with the privileges assigned to the address space (i. e. the 'user' assigned to the address space).

7.3.1.5 Subjects

Subjects in z/OS are programs executing in the context of an address space. The parameters of an address space are stored within an 'Address Space Control Block' (ASCB). Within an address space several programs can execute in parallel as 'tasks'. Each task is assigned to exactly one address space and the parameters of a task are stored in a 'Task Control Block' (TCB). The address of the list of TCBs belonging to an address space is stored in the ASCB. There is also a kind of 'lightweight' subjects controlled by a 'Request Block' (RB), which are also linked to an address space.

7.3.1.6 Security Services

The security services for user identification and authentication, access control, and security related auditing are implemented in a single component of z/OS called the 'Resource Access and Control Facility' (RACF). RACF consists of a set of system services that can be used by properly authorized callers to perform I&A, access control, and security management functions. The authorization required for each service are documented with the service itself.

7.3.1.7 User interaction with z/OS

A user interacts with z/OS either via a JOB, defined using a 'Job Control Language' (JCL) or interactively. For direct interaction with z/OS a user either has to use the 'Time Sharing Option' (TSO) or use a shell of the UNIX System Services (USS) subsystem. In all of those cases the user has provide a valid user-ID and associated authentication credentials.

7.3.1.8 Persistent Storage

To store data permanently z/OS provides storage containers called 'data sets' or – when operating under the USS subsystems – traditional Unix files and directories. Data sets on disks are identified by their data set name and the disk they reside on. To make locating a data set easier, z/OS offers a mechanism called a 'catalog' that can be used to locate data sets without specifying the directly the disk they reside on. Data sets can also be located on tapes.

7.3.1.9 Authorized Programs

In addition to supervisor and PC routines, z/OS has a number of “authorized programs” that need to be trusted because they are not restricted by the security policy defined in this Security Target. An authorized program may call a number of program calls or supervisor calls or use supervisor call

parameters that are reserved for authorized programs. In particular, it is authorized to call the MODESET SVC used to switch into supervisor state. With this function, authorized programs can execute any privileged instruction.

A program is authorized if at least one of the following conditions is true:

- The program is executing in supervisor state **{SP.3::SP.3.1}**.
- The program is executing with a PSW key of 0 to 7 or a PSW key mask value that supports at least one key in the range of 0 to 7 in control register 3 **{SP.3::SP.3.2}** .
- The authorization bit is set in the Job Step Control Block (JSCB) under which the program is executing **{SP.3::SP.3.3}**.

Whenever a supervisor routine reserved for authorized programs is called or when a parameter reserved for authorized programs is used, the routine invoked to service the request checks if one of the above listed conditions is satisfied. Only if this is true, the request is honored **{SP.3::SP.3.4}**. Note that the hardware performs some checks when a supervisor routine is called with a Program Call (PC) instruction. In this case the routine implementing the service only needs to perform its own checks if additional restrictions to those implied by the hardware checks apply. Note also that some supervisor routine may be more restrictive, i. e. only a subset of the three conditions mentioned above is checked and the request is rejected if not one of the conditions in the subset apply. For example the hardware can not check if a program running in problem state with a PSW key of 8 is authorized by the authorization bit in the JSCB.

An authorized program can be started in one of the following ways:

- By starting a program from a dedicated program library (defined in the system configuration data set SYS1.PARMLIB) that has the authorization bit set in the directory entry of the member of the partitioned data set (library) containing the program. This program has to be the one started with the EXEC JCL statement of the job step, as a TSO command, as a UNIX process using exec(), or started as a dedicated task by an authorized program using the ATTACH supervisor call with parameters reserved for authorized programs **{SP.3::SP.3.5}**.

Note: TSO commands might be entered directly by the user at a terminal, executed in a batch job that runs TSO TMP, or entered programmatically using the TSO IKJEFTSR service. They may also be executed by any service that uses either the TMP or IKJEFTSR service such as the REXX 'address tso' function or the Unix shell 'tsocmd' function.

- By starting a started task from an authorized library using the operator START command **{SP.3::SP.3.6}**.
- By starting an authorized program from a zFS file system **{SP.3::SP.3.V1R7.1}**. A program in a zFS file system is authorized when the authorization bit has been set using the extattr –a command for the file containing the program **{SP.3::SP.3.V1R7.2}**. A user needs to have been authorized to the BPX.FILEATTR.APF profile in the FACILITY class to set the authorization bit **{SP.3::SP.3.V1R7.3}**. If a program running in an APF-authorized address space attempts to load a program from zFS that does not have the APF-extended attribute set, the load is rejected **{SP.3::SP.3.V1R7.4}**. Sanction lists can be defined that restrict access of authorized programs in the z/OS Unix System Services environment to files and directories defined in those sanction lists **{SP.3::SP.3.V1R7.5}**. Note that the APF-authorized extended attribute of a file is not honored if the file system containing the file has been mounted as NOSECURITY **{SP.3::SP.3.V2R1.1}**.

For the invocation of MVS programs linkedited AC=1 found in an APF-authorized library invoked via the z/OS UNIX spawn, exec and attach_exec service and for MVS load library programs that are to run as a z/OS UNIX set-user-id or set-group-id program the following rules apply:

- If the z/OS UNIX path name supplied to spawn, exec or attach_exec represents an external link that resolves to a MVS program found in an APF-authorized library and linkedited with the AC=1 attribute, the external link must have a owning UID of 0 and not be found in a file system mounted as NOSECURITY to allow this type of invocation **{SP.3::SP.3.V2R1.1}**.
- If the z/OS UNIX path name supplied to spawn, exec, or attach_exec represents a regular file with the sticky bit attribute that resolves to a MVS program found in an APF-authorized library and linkedited with the AC=1 attribute, the sticky bit file must have an owning UID of 0 or have the APF extended attribute turned on to allow this type of invocation. Additionally, the sticky bit file must not be found in a file system mounted as NOSECURITY to allow this type of invocation **{SP.3::SP.3.V2R1.2}**.
- If the z/OS UNIX path name supplied to spawn, exec or attach_exec represents a symbolic link to a regular file with the sticky bit attribute and the sticky bit file has the set-user-id attribute, the symbolic link must have an owning uid of 0 or an owning uid equal to that of the sticky bit file. If the sticky bit file has the set-group-id attribute, the symbolic link must have an owning uid of 0 or an owning gid equal to that of the sticky bit file. Additionally, the symbolic link must not be found in a file system mounted as NOSECURITY to allow this type of invocation **{SP.3::SP.3.V2R1.3}**.

Libraries that can contain authorized programs need to be protected from unauthorized modifications including the possibility to add new programs to the library. zFS files containing authorized programs also need to be protected from unauthorized modifications. The discretionary access control features of z/OS have to be used to protect those libraries.

The IKJTSOxx member of SYS1.PARMLIB can be used to define the authorized programs and commands that can be executed in the TSO environment **{SP.3::SP.3.V1R7.6}**.

Some trusted subsystems of z/OS are started as part of the standard startup procedure or may be later started by explicit request of a properly authorized user.

7.3.2 Security Functionality

7.3.2.1 Identification and Authentication (FIA, FTA)

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

- As a TSO user
- As an operator at a console
- As a user of the UNIX System Services (USS), including access via the UNIX shell or remote access using OpenSSH.

In all cases users are identified and authenticated by the TOE **{IA.1::IA.1.1}** before being authorized to perform any other security relevant action. In the case of jobs submitted by an already-authenticated user, no additional authentication is required for jobs running with the ID of the user who submitted them. The internal reader accepts (and relies) in this case on the authentication performed when the user has logged on to the system **{IA.1::IA.1.2}**.

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

When authenticating a user, the TOE allows applications to accept:

- A user ID defined to RACF **{IA.1::IA.1.4-R8-RACF-1}** and the RACF password **{IA.1::IA.1.4-R8-RACF-2}** or password phrase **{IA.1::IA.1.4-R10-RACF-4}** or a PassTicket **{IA.1::IA.1.4-R8-RACF-3}**.

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

Some additional considerations:

- For access to UNIX functions, the user must have a valid UID and his default group must have a valid GID **{IA.1::IA.1.6}**.
- If the user is in additional groups they may have GIDs, too, and if so UNIX access checking will make use of those additional GIDs **{IA.1::IA.1.6-R8-USS-3}**.
- If the user ID is in REVOKE status, RACF prevents user from entering the system at all or entering the system with certain groups **{IA.1::IA.1.7}**.
- For a user defined as a system administrator (that is, one who has the system SPECIAL attribute) a message is displayed on the console asking the operator if the user shall be revoked if he exceeds the number of failed login attempts due to incorrect passwords **{IA.1::IA.1.7-R8-RACF-1}** or if he exceeds the system inactivity interval **{IA.1::IA.1.7.R8-RACF-2}**.

7.3.2.1.1 Authentication Functions

7.3.2.1.1.1 RACF Passwords and Password Phrases

In RACF, the user selects his own password/phrase and only the user knows the value chosen. RACF stores the encrypted representation of passwords and passphrases in the RACF database. If the user has forgotten his password/phrase and it needs to be reset, the security administrator will reset the password/phrase **{IA.2::IA.2.1-R10}**. When the system administrator follows the rules for the evaluated configuration, this new password/phrase should be in an expired state, thus forcing the user to enter a new password/phrase on the next logon **{IA.2::IA.2.2-R10}**. When creating a new user ID that is not a protected user ID, the initial password/phrase may be marked as non-expired, allowing it to be used without being changed first. **{IA.2::IA.2.3-V2R5}**.

7.3.2.1.1.1.1 Password Quality

A system administrator can set a variety of system-global rules for forming valid passwords using the SETROPTS command (for system-wide settings) or (to a lesser extent) using the password command to affect only one user. He can change such parameters as the number of days a password is valid for, how long to maintain password history to prevent the user from reusing the same password again, the minimum number of days between password changes, and rules for password content.

When a user changes a password, RACF treats the new, user-supplied password as an encryption key to transform the RACF user ID into an encoded form and is depending on the “ALGORITHM(KDFAES)” setting, is using the DES or AES algorithm to store it in the database. The password is not stored in clear text **{IA.2::IA.2.4-V2R5-RACF-1}**.

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

- Minimum and maximum length of passwords (LENGTH(m1:m2) as part of a RULE suboption) **{IA.2::IA.2.5}**
- Maximum password lifetime (INTERVAL suboption) **{IA.2::IA.2.6}** and minimum password change time (MINCHANGE option) **{IA.2::IA.2.V1R7-1}**
- Number of passwords from the user's password history that are not allowed for a new password (HISTORY suboption) **{IA.2::IA.2.7}**

- Maximum number of consecutive failed authentication attempts until the REVOKE attribute is set in the user's profile (REVOKE suboption) **{IA.2::IA.2.8}**
- Differentiate between upper- and lowercase characters with the PASSWORD(MIXEDCASE) option **{IA.2::IA.2.V1R7-2}**
- allowance for the use of the following special characters in passwords: .<+|&!*-%_>?:=
- Type of character for each character position of a password. Possible types are **{IA.2::IA.2.9}**:
- ALPHA
- ALPHANUM (which includes also the special characters $, # and @)
- VOWEL
- NOVOWEL
- CONSONANT
- NUMERIC
- MIXEDCONSONANT
- MIXEDVOWEL
- MIXEDNUM
- NATIONAL

If the value ALPHANUM is defined for more than one position in the password, at least one alphabetical value and one numeric value are required by RACF.

The SETROPTS command with the SPECIALCHARS option can be used to increase the possible password space by an additional 14 characters (! % & * _ + | : ? > < . - =) **{IA.2::IA.2-V2R2.1}**.

A new MIXEDALL content-keyword is used to force a mixture of the four categories of characters within a password. Note: MIXEDALL considers the existing national characters (@, #, $) as special characters, and not as upper case letters **{IA.2::IA.2-V2R2.2}**.

SMF Type 80 record for the SETROPTS event code is being augmented with new indicators for:

- the NO/SPECIALCHARS keyword specified
- the NO/SPECIALCHARS keyword failed
- the SPECIALCHARS setting in effect after completion of the SETROPTS command.

{IA.2::IA.2-V2R2.3}

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

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

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

7.3.2.1.1.1.2 Password Phrase Quality

Many of the system rules for passwords set by SETROPTS apply to password phrases, too. However, RACF does not provide support for content syntax rules when using password phrases.

When a password phrase is established for a user, RACF treats the new phrase as a sequence of encryption keys to transform the RACF user ID into an encoded form using the DES algorithm with chaining, that it then stores on the database. The password phrase is not stored in clear text **{IA.2::IA.2-R10-RACF-1}**.

The following system-wide options that can be set to enforce a minimum strength of passwords using the PASSWORD option in the SETROPTS command also apply to password phrases:

- Maximum password phrase lifetime (INTERVAL suboption) **{IA.2::IA.2-R10-RACF-2}** and minimum password phrase change time (MINCHANGE option) **{IA.2::IA.2-R10-RACF-3}**
- Number of password phrases from the user's password phrase history that are not allowed for a new password phrase (HISTORY suboption) **{IA.2::IA.2-R10-RACF-4}**
- Maximum number of consecutive failed authentication attempts using a password or password phrase until the REVOKE attribute is set in the user's profile (REVOKE suboption) **{IA.2::IA.2-R10-RACF-5}**
- A password phrase must be changed after the first use, if the EXPIRED parameter is passed to the ADDUSER or ALTUSER command. **{IA.2::IA.2-V2R4-RACF-1}**

Rather than having an administrator specify syntax rules to specify valid password phrase content, RACF enforces the following set of predefined rules:

- maximum length: 100 characters in the absence of exit ICHPWX11 **{IA.2::IA.2-R10-RACF-6}**

  Note: The evaluated configuration of the TOE generally does not allow customers to implement exits to change the system processing. However, RACF supplies a sample ICHPWX11 exit and a sample REXX exec IRRPHREX that the sample ICHPWX11 will invoke. The administrator may install the sample ICHPWX11 unmodified, and may specify tailoring options in IRRPHREX to apply some additional syntax/content rules.
- minimum length:
  - 14 characters in the absence of exit ICHPWX11 and when KDFAES is not the passphrase encryption algorithm **{IA.2::IA.2-V2R2-4}**
  - 9 characters when KDFAES is the passphrase encryption algorithm **{IA.2::IA.2-V2R2-5}**
  - 9 characters if exit ICHPWX11 is present and allows the phrase **{IA.2::IA.2-R10-RACF-8}**
- The phrase may not contain the user ID, in either sequential uppercase or sequential lowercase characters **{IA.2::IA.2-R10-RACF-9}**
- The phrase must contain at least two alphabetic characters (A-Z, a-z) **{IA.2::IA.2-R10-RACF-10}**
- The phrase must contain at least two non-alphabetic characters (numeric, punctuation, special (including blanks)) **{IA.2::IA.2-R10-RACF-11}**
- The phrase may not contain more than two consecutive identical characters **{IA.2::IA.2-R10-RACF-12}**

If the administrator chooses to install the supplied sample exit ICHPWX11, the sample REXX exec IRRPHREX may then apply the following additional checks, if selected by the administrator, and may then accept a shorter phrase or reject a phrase that RACF would have accepted:

- The administrator can set the minimum allowable phrase length to a value between 9 and 100 inclusive by setting variable Phr_minlen **{IA.2::IA.2-R10-RACF-26}**
- The administrator can set the maximum allowable phrase length to a value between 9 and 100 inclusive by setting variable Phr_maxlen **{IA.2::IA.2-R10-RACF-13}**
- The administrator can set a more restrictive set of characters for password phrases by setting the variables numbers, letters, special, and Phr_allowed_chars **{IA.2::IA.2-R10-RACF-14}**
- The administrator can prevent leading or trailing blanks in password phrases by setting the variables Phr_leading_blanks or Phr_trailing_blanks to “no” **{IA.2::IA.2-R10-RACF-15}**
- The administrator can prevent use of password phrases that contain a case-insensitive character string from the user's name by setting the variable Phr_name_allowed to “no” and setting the variable Phr_name_minlen to the longest substring allowed **{IA.2::IA.2-R10-RACF-16}** Example: if the user's name is John Smith the administrator could prevent the user from specifying a phrase containing John or john or jOhn or Smith by appropriate settings of the variables.
- The administrator can enable a triviality check by setting the variable Phr_triviality to “yes”. This will prevent use of a new password phrase that differs from the old one only insertion/deletion of spaces or changing character case. It also will reject a new phrase when the shorter of the old and new phrases is simply a substring of the other. **{IA.2::IA.2-R10-RACF-17}**
- The administrator can prevent use of new phrases that do not differ in a significant number of characters from the old phrase by setting the variable Phr_min_unique to the number of positions that must differ. In addition, if the variable Phr_min_unique_norm has the value “yes” the exec will first normalize the old and new phrases to be checked by converting them to uppercase and removing spaces. **{IA.2::IA.2-R10-RACF-18}**
- The administrator can prevent the user of a new phrase which simply reorders the words of the old phrase by setting the variables Phr_unique_words (number of words that must be unique), Phr_word_minlen (minimum length of the unique words), and Phr_word_unique_upper (if “yes” then the exec will convert the old and new phrases to uppercase for this check **{IA.2::IA.2-R10-RACF-19}**
- The administrator can provide a list of disallowed words by setting the variables Phr_dict.0 to the number of words in a supplied list, and supplying the list in variables Phr_dict.1, Phr_dict.2, etc. **{IA.2::IA.2-R10-RACF-20}**

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

- when entered at a TSO terminal as part of the login process **{IA.2::IA.2-R10-TSO-23}**, or
- when entered into one of the RACF-supplied ISPF panels that allows specification of a password phrase **{IA.2::IA.2-R10-RACF-25}**.

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

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

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

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

- For pthread_security_np() and __passwd(), the application can
  - update the BPXYTHLI control block to indicate that z/OS UNIX should instead use the job name as the APPLID value **{IA.2::IA.2.14-R10-USS-2}**, or
  - update the BPXYTHLI control block to indicate a specific APPLID to use **{IA.2::IA.2.14-R10-USS-3}**.
- By changing to use one of the corresponding new services pthread_security_applid_np(), __login_applid(), and __passwd_applid() the application can specify an APPLID value directly as a parameter on the call **{IA.2::IA.2.14-R10-USS-4}**.

7.3.2.1.1.1.3 Authentication via Client Digital Certificates

In the evaluated configuration, TLS-aware applications, or the Application-Transparent TLS (AT-TLS) functions of the Communications Server, can accept client certificates and map them to RACF user IDs as part of the client authentication process. Such applications must be configured to use RACF to store the keyrings that contain the application private key and the allowed Certificate Authority (CA) certificates that may be used to provide the client certificates that the application will support. The security administrator will use RACDCERT to establish those keyrings, which may reside in RACF profiles in the DIGTRING class or in PKCS#11 tokens maintained in ICSF, and thus to approve of any CAs that will be used. Any CA used in the evaluated configuration must support obtaining revocation information through OCSP responses , and the security administrator must configure the application to use OCSP responses **{IA.2::IA.2.V2R5-SSL-2}**. This configuration may be application-specific, or may be done by establishing LE environment variables that System SSL will use in the absence of specific application-provided OCSP configuration information.

The first step in the client authentication process is for the server or AT-TLS to acquire the client certificate via the standard TLS data flows. As part of that processing, System SSL will validate the client certificate using the process specified by [RFC5280]. **{IA.2::IA.2.15-V2R4-SSL-20}**.

For AT-TLS one can also specify a policy to perform certificate path validation according to [RFC5280] **{IA.2::IA.2-V2R2.7}**.

AT-TLS also allows to specify a security policy that include OCSP for certificate revocation checking **{IA.2::IA.2-V2R5.1}**.

System SSL will perform the following checks against the client certificate and certification chain:

1. [RFC5280], section 6 conforming certificate validation and certificate path validation. **{IA:2::IA.2-V2R5-SSL-CPV-1}**
2. The certificate path must terminate with a trusted CA certificate. **{IA:2::IA.2-V2R4-SSL-CPV-2}**
3. Ensure the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates certificates, and that any path constraints are met. **{IA:2::IA.2-V2R5-SSL-CPV-3}**
4. The TSF shall validate that any CA certificate includes "Certificate Signing" as a purpose the key usage field. **{IA:2::IA.2-V2R5-SSL-CPV-3a}**
5. The revocation status of the certificate is checked using the Online Certificate Status Protocol (OCSP) as specified in [RFC6960]. **{IA:2::IA.2-V2R5-SSL-CPV-4}**
6. The following extendedKeyUsage field rules are applied during validation:
   - 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.
   - 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.

- 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.
- S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the extendedKeyUsage field. It should be noted, that the TOE in its evaluated configuration does not provide applications to process S/MIME encrypted mails and thus S/MIME certificates.
- 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.

{IA:2::IA.2-V2R5-SSL-CPV-5}

7.3.2.1.1.1.4 Authentication via Public/Private Key (SSH)

OpenSSH supports authentication via public/private keys, however for the evaluated configuration OpenSSH on z/OS must be configured to obtain those public/private key pairs from digital certificates associated with RACF key rings. The existing RACDCERT command can be used to generate the keys and a certificate, or the certificates may be generated elsewhere and imported into RACF using RACDCERT. Private keys stored directly in the UNIX file system must not be used.

When a remote user authenticates to the OpenSSH server, the server will use the public key, obtained via a digital certificate which is associated with the user's configured key ring, to perform the authentication. **{IA.2::IA.2-R12-SSH-KEY-1}**

When a z/OS user acts as an SSH client, connecting to an SSH server, the client will obtain the necessary private key via a digital certificate which is associated with the user's configured key ring to perform the authentication. **{IA.2::IA.2-R12-SSH-KEY-2}**

The private key is not needed at the server when a client authenticates. The public keys must be distributed to remote hosts. When stored in a key ring, the certificate must be exported (via RACDCERT) and manually copied (e.g. by transferring them using a secured channel) to the remote host, where it will be imported into that system's key ring.

When configured to use key rings, the OpenSSH server and client code will use existing System SSL interfaces to pull the keys from the RACF key ring, and the server and client will need authority to those key rings to use the RDATALIB service. **{IA.2::IA.2-R12-SSH-KEY-3}**.

When a user registers his public key with the user he wants to access on the server side, a key-based authentication can be performed instead of a password-based authentication.**{CS.5::CS.1-V2R4-SSH-2}**. The key-based authentication is performed as defined by RFC 4252. The public key for the key-based authentication must reside in the home directory of the target user in the file .ssh/authorized_keys. As this file may contain multiple key, each key is tried whether it is appropriate as a public key for the authentication attempt (i.e. whether the public key can decrypt the data sent by the client encrypted with the client's private key). The first key that is found to match the private key indicates a successful authentication. **{CS.5::CS.1-V2R4-SSH-3}**

7.3.2.1.1.1.5 Started procedures

With the concept of a started procedure, the TOE provides a mechanism where a defined task can be started by an operator, but then operates under a defined user ID that is specifically assigned to the started procedure itself **{IA.3::IA.3.1}** .

A started procedure consists of a set of job control language statements that are frequently used together to achieve a certain result. Started procedures usually reside in the system procedure library, SYS1.PROCLIB, which is a partitioned data set. A started procedure is usually started by an operator, but can be associated with a functional subsystem. For example, SMS is treated as a started procedure even though it does not need to be specifically started with a START command.

Only RACF-defined users and groups can be specifically authorized to access RACF-protected resources **{IA.3::IA.3.2}**. Other users can access those resources with the authority allowed in the UACC entry of the RACF profile controlling access to the resource. However, started procedures have system-generated JOB statements that do not contain the USER, GROUP, or PASSWORD parameter.

To enable started procedures to access RACF-protected resources with other authorities than those defined in the UACC entry of the profile protecting the resource, started procedures must have RACF user IDs and group names **{IA.3::IA.3.4}**. By assigning them RACF identities, an installation can give started procedures specific authorization to access RACF-protected resources. For example, one can allow JES to access spool data sets.

To associate the names of started procedures with specific RACF group names and user IDs, an administrator can do one of the following:

- Set up the STARTED class (the recommended method)
- Create a started procedures table (ICHRIN03)

Assigning RACF user IDs to started procedures

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

7.3.2.1.1.1.6 Protected user IDs

The user IDs that an administrator assigns to started procedures should have the PROTECTED attribute unless the started procedure is required to have a user ID with a password defined. Protected user IDs are user IDs that have both the NOPASSWORD, NOPHRASE and NOOIDCARD attributes **{IA.3::IA.3.5-V2R5}**. They are defined or modified using the ADDUSER and ALTUSER commands. Protected user IDs can not be authenticated via a password, password phrase, or RACF PassTicket, and are protected from being revoked through incorrect password attempts **{IA.3::IA.3.6-R12-RACFEAL5}**.

7.3.2.1.1.1.7 Authentication Method Summary

The following TOE applications support client authentication via digital certificates when using TLS sessions in the evaluated configuration:

- TN3270, when using a TN3270 emulator that supports TLS certificate based mutual authentication. **{IA.3-V2R5-TN3270-AUTHSSL}**

The following TOE functions support authentication using passwords/phrases in the evaluated configuration:

- TSO/E **{IA.3::IA.3-R10-TSO-AUTHPHRASE}**
- OpenSSH **{IA.3::IA.3-R10-SSH-AUTHPHRASE}**
- The z/OS UNIX shell commands su and passwd **{IA.3::IA.3-R10-USS-AUTHPHRASE-1}**
- The C runtime functions __login(), __passwd(), pthread_security_np() (and the variants that accept an APPL ID), and getpass() **{IA.3::IA.3-R10-LE-AUTHPHRASE}**
- TN3270 Server **{IA.3::IA.3-R13-TN3270-AUTHPHRASE}**
- An operating system console **{IA.3::IA.3-V2R5-CONSOLE-AUTHPHRASE}**

OpenSSH supports authentication via public/private key pairs stored in digital certificates. It can be configured to store the keys and certificates in RACF key rings **{IA.3::IA.3-R12-SSH-AUTHRINGS}**.

7.3.2.1.1.1.8 Handling of Groups During Authentication

During authentication, RACF construct security information that represents the user (subject) for subsequent use during access checking.

- During RACF authentication, RACF determines whether list-of-groups processing is in effect or not. If list-of-groups is not in effect, RACF puts the user's default group into the subject's ACEE, or the group specified by the caller of the RACF interfaces for user authentication. If list-of-groups is in effect, RACF gathers a list of all the groups to which the user is connected, and makes a copy of that list in the subject's ACEE. During access checking (DAC) for MVS resources, RACF can then base its decisions on both the user ID and on the group membership of the user **{IA.3::IA.1.14-R12-RACFEAL5-1}**.
- When a user attempts to use UNIX functions, RACF selects from the group(s) in the subject's ACEE up to the first 300 (alphabetically) which have OMVS segments with GIDs defined. During access checking (DAC) for UNIX resources, RACF can then base its decisions on the user's UID and the selected groups' GIDs **{IA.3::IA.1.14-R10-RACF-2}**.

7.3.2.1.1.1.9 Assertion of User Identity

{IA.5::IA.5-R12-IDPROP-RACF-1} RACF supports specification on initACEE and RACROUTE REQUEST=VERIFY of a distributed identity via a structure called an IDID (containing a user's distinguished name (DN) and a domain/realm name (DC)):

- If an IDID is specified on initACEE but a RACF user ID is not specified, then initACEE will perform a mapping operation using the IDIDMAP class to determine the associated RACF user ID to use during RACROUTE REQUEST=VERIFY processing and will also include the IDID information.
- If both an IDID and a RACF user ID are specified on initACEE, then initACEE will create an ACEE for that user ID as it usually would and not perform mapping. Again, it will include the IDID information on the RACROUTE REQUEST=VERIFY call.
- When an IDID is specified on RACROUTE REQUEST=VERIFY, RACF uses the other parameters to create the ACEE as it normally does, but will anchor the IDID information in the ACEE for later use during auditing.

{IA.5::IA.5-R12-IDPROP-RACF-2} RACF provides a 'RACMAP' command to allow the security administrator to define 'mapping filter rules' to RACF that will support the mapping of distributed user identities, as specified within the IDID data area, into RACF user IDs as required by the customer. This new RACF command is similar to the existing RACDCERT command, which allows the specification of mapping filter rules that RACF uses to map distributed user identities based on the 'subject' and 'issuer' information within Digital Certificates. But instead of being limited to only user identities within Digital Certificates, the new command supports the definition of mapping filter rules within the IDIDMAP class based on an x.500 representation of the user identity and the 'Name-Space' that the user is defined within.

{IA.5::IA.5-R12-IDPROP-RACF-3} The RACF R_cacheserv callable service provides a function (function code 7) that will extract a copy of the ACEE for the currently active user in the form of a RACF environment object (aka RACO), save that RACO in a data space, and return a context reference (ICRX) that will uniquely identify that saved RACO. Subsequently an invoker of RACROUTE REQUEST=VERIFY can provide that ICRX and RACF will recreate the security environment (ACEE) of the original user from the RACO or from the IDID information in the ICRX if necessary. R_cacheserv will also allow deletion of a cached security environment.

{IA.5::IA.5-R12-IDPROP-RACF-5}The RACF R_cacheserv service can also return a pseudo-userID and pseudo-password that RACF authentication functions (initACEE, RACROUTE REQUEST=VERIFY) will subsequently accept and use to create an ACEE for the previously specified RACF user ID with an ICTX data area cached on the earlier R_cacheserv invocation. The pseudo-userID and pseudo-password may be used at most once on a subsequent authentication request.

{IA.5::IA.5-R12-IDPROP-RACF-4} RACF will provide an ENF 71 signal when an administrator has issued an ALTUSER REVOKE or a CONNECT or REMOVE command that changes a user's group connections, allowing applications that have cached ACEEs locally or via R_cacheserv to remove their cache entries and recreate the ACEEs if needed.

{IA.5::IA.5-V2R1-IDPROP-RACF-6} RACF will provide an ENF 71 signal when an administrator has issued a DELUSER or DELGRP command allowing applications that have cached ACEEs local or via R_cacheserv to remove cache entries.

{IA.5::IA.5-V2R1-IDPROP-RACF-7} RACF will provide an ENF 79 signal when an administrator has issued a PERMIT command that changes a user's or group's authorization to resources in a resource class that has been defined in the RACF Class Descriptor Table with the SIGNAL=YES option.

{IA.5::IA.5-R12-IDPROP-USS-1} The UNIX System Services __passwd() (BPX1PWD) and pthread_security_np() (BPX1TLS) function allows appropriately authorized servers to assert a user identity and create a security environment by specification of the pseudo-userID and pseudo-password obtained via a prior authentication and use of R_cacheserv.

7.3.2.1.1.1.10 Access Banners

The TOE displays informative banners before or during the login to users. For SSH based logons, the message can be configured in /etc/profile or in the SSH daemon configuration file with the Banner keyword. For TSO based logons messages can in the logon panel module IKJLPENU. The document [TSO.CUST], chapter 8 describes the details on the customizations of the logon panel **{IA.5::TAB-V2R4-1}**.

7.3.2.2 Access Control (FDP)

7.3.2.2.1 Access Control Overview

7.3.2.2.1.1 Access control principles

z/OS provides the Resource Access Control Facility (RACF) as the component that performs access control between subjects acting on behalf of a user and resources protected by the discretionary access control policies. RACF uses user and resource profiles it stores in the RACF database to decide if a subject has access to a non-UNIX resource. For UNIX resources, the access permissions are carried with the resource itself (permission bits).

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

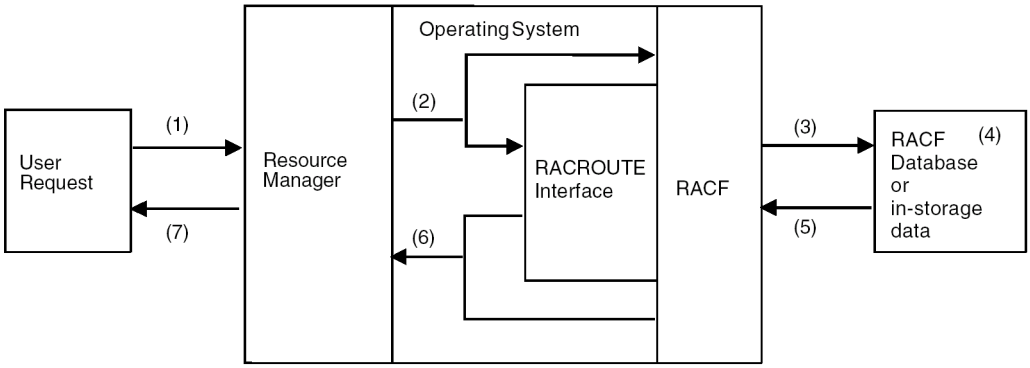

Figure 1: RACF Request Flow

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

The z/OS component responsible for managing the resource calls the RACF component using the internal interface to RACF (mainly the RACROUTE interface) to check the access rights of the user that initiated the user request and passes the name and type of the resource and the requested type of access to RACF **{AC.1::AC.1.1}**. The caller may also pass the ID of the user or an explicit user security context (ACEE), or RACF obtains those values from the security context of the user that has been established during user authentication (2) **{AC.1::AC.1.2}**.

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

If the resource is known to RACF, RACF returns either a “yes” or a “no” decision for the access request **{AC.1::AC.1.3}**. If the resource is not known to RACF, RACF may return a “don't know” return code unless there are specific options set that allow RACF to take a yes or no decision (6) **{AC.1::AC.1.4}**. In the case of a “don't know” result, the resource manager needs to make its own decision whether to allow access or not. Depending on the decision, the resource manager will either perform or reject the access request of the user program (7) **{AC.1::AC.1.5}**.

The protection philosophy of RACF is based on “profiles” that represent protected resources but also users and groups. Profiles are organized in profile classes, where each class represents a type of resource (such as data sets or terminals) or other entity (such as users or groups). A profile stores attributes of the subject or object it represents.

For profiles that represent a protected resource, an access list can be assigned **{AC.1::AC.1.6}**. This access list specifies the type of access subjects may have to the resource represented by the profile.

RACF handles 6 different types of access which are hierarchically ordered. Those access types are (from low to high):

- NONE
- EXECUTE
- READ
- UPDATE
- CONTROL
- ALTER

Hierarchically ordered means that a higher access type implies also the lower access types.

To check access to a z/OS resource, a resource manager will call RACF specifying:

- the resource class
- the name of the resource
- a pointer to the user's ACEE (which represents the user)
- the requested type of access

If RACF knows the resource class and if this resource class is active, RACF will identify the resource profile protecting the resource in that class, extract the access control list for that resource profile and checks if the user has the requested type of access or a higher type of access. The exact details of this access check algorithm are defined later in this section.

The semantics of a specific type of access are defined by the resource manager. This allows RACF to used also for privilege management by defining a specific privilege as a specific type of access to a specific profile in a specific class which represents the privilege. A resource manager then can call RACF

to check if a user has the required type of access to this profile and allow the user to perform a specific privileged function only if the user has that type of access. Quite a number of management activities defined in the Security Management section are implemented that way and the Security Management section describes the classes, profiles in the classes and the semantics z/OS assigns to specific access types in those classes that are used for specific management privileges.

Access control to UNIX file system objects and IPC objects are also handled by RACF, but in the case of these objects, the access rights are stored with the object itself. RACF still performs the access check. For details, see the description of access control for UNIX objects.

7.3.2.2.1.2 Protected resources

The protected resources considered in detail in this Security Target are:

- Data sets
- Programs
- UNIX file system objects
- UNIX IPC objects
- System logger objects

As a general-access control system, RACF is capable of protecting a number of other resources, but those are not considered in detail in this evaluation because it is up to the resource manager that uses them to determine the valid resource names and the semantics of the access control decisions. Instead, we will mention them below and later consider only the rules regarding their administration via the RACF commands.

RACF can also protect installation-defined resource classes, which we will not consider at all in this evaluation.

The reader should note that some other RACF classes are included in this evaluation that do not represent “resources” but represent privileges or restrictions, where assigning “access” to a resource in such a class to a user or a group just determines that the user or group has the privilege or restriction associated with the profile. Those classes and profiles are described in the relevant subsection of the access control section in this Security Target. The reader should also understand that granting privileges that are not described in this document should be done with care, and only for trusted users, as those privileges may allow administrative functions or extraordinary resource accesses.

Resource profiles of RACF are structured into an open set of “resource classes”. IBM provides a set of resources classes used by z/OS (stored in the “static class descriptor table”), but RACF also allows for the definition and activation of additional resource classes using the RDEFINE or RALTER commands addressing the CDT general resource class (those are stored in the “dynamic class descriptor table”). The dynamically defined classes need to be “activated” using the command SETROPTS RACLIST(CDT) REFRESH. Resource classes represent “types” of objects that are access protected by RACF. IBM supplies a default static class descriptor table, which is structured into resources used by different components of z/OS as well as resources used by specific other IBM products like DB2 or CICS.

7.3.2.2.2 Discretionary Access Control

7.3.2.2.2.1 Datasets

7.3.2.2.2.1.1 Standard System Datasets

The following dataset contain the standard z/OS system libraries, as well as the kernel and drivers:

SYS1.PROCLIB
: This library contains JCL procedures distributed with z/OS. In practice, there may be other JCL procedure libraries (supplied with various program products) concatenated with it.

SYS1.PARMLIB
This library contains control parameters for z/OS and for some program products. In practice, there may be other libraries concatenated with it.

SYS1.LINKLIB
This library contains many of the basic execution modules of the system. In practice, it is one of a large number of execution libraries that are concatenated.

SYS1.LPALIB
This library contains system execution modules that are loaded into the link pack area when the system is initialized. There may be several other libraries concatenated with it. Programs stored here are available to other address spaces.

SYS1.NUCLEUS
This library contains the basic supervisor ("kernel") modules of z/OS.

SYS1.SVCLIB
This library contains operating system routines known as supervisor calls (SVCs).

SYS1.SEZAINST
This library contains the configuration data sets and files that are used by the TCP/IP servers and functions.

The data set name table (ICHRDSNT) is an installation-defined load module that describes the data sets in the RACF database to RACF. This table contains entries describing each data set in the RACF database and its backup data set. This database holds, amongst other information the encrypted user password and passphrases.

The installation-defined "Cryptographic Key Data Set (CKDS)", which holds cryptographic keys managed by the TOE.

The standard USS files and libraries are the following:

/etc
System configuration

/usr/bin, /usr/sbin, /bin, /sbin
System binaries

/usr/lib, /lib
System libraries

It should be noted, that the administrator is expected to follow the guidance and verify the standard system datasets are protected from unauthorized access.

7.3.2.2.2.1.2 Standard data set naming conventions

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

If an installation has chosen to define data set profiles under the standard RACF naming conventions, they can create a group for each high-level qualifier that is not a user ID, and permit users to protect any data set that has that high-level qualifier by giving them CREATE authority in that group **{AC.2::AC.2.1}**.

7.3.2.2.2.1.3 Table-driven data set naming conventions

An installation can use the naming convention table to set up and enforce a data set naming convention other than that used by RACF **{AC.2::AC.2.2}**. The table can:

- Supply a qualifier to be used as the high-level qualifier for authorization checking **{AC.2::AC.2.3}**
- Convert data set names to RACF naming convention form for RACF use **{AC.2::AC.2.4}**
- Convert names in RACF form to the installation's format for external display **{AC.2::AC.2.5}**
- Enforce a naming convention by not allowing the definition of data sets that do not conform to an installation's rules **{AC.2::AC.2.6}**
- Reduce RACF overhead by determining whether a data set is a user or group data set

An installation can create a naming convention table (module ICHNCV00), which RACF uses to check and modify (internally to RACF) the data set name in all commands and macros that process data set names **{AC.2::AC.2.7}**. An installation can use the table to selectively rearrange data set names to "fit" the RACF convention without actually changing those names.

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

If some of the data sets in an installation have names that consist of a single qualifier, one can still RACF-protect those data sets **{AC.2::AC.2.8}**. To get RACF protection for single-qualifier names, the SETROPTS command with the PREFIX operand must be issued.

This command defines a high-level qualifier to be used as a prefix for single-qualifier names and activates the facility **{AC.2::AC.2.9}**. Then, when RACF processes requests for the data set, RACF internally modifies single-qualifier names by adding the prefix, making the data set names acceptable to RACF routines **{AC.2::AC.2.10}**. All SMF log records and all messages from RACF contain the RACF-modified version of the data set name **{AC.2::AC.2.11}** unless the SETROPTS REALDSN option is in effect **{AC.2::AC.2-R10-RACF-1}**.

7.3.2.2.2.1.5 Protecting user data sets

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

- In general, all RACF-defined users can protect their own data sets **{AC.2::AC.2.12}**
- A user can RACF-protect a data set for another user under any of the following conditions:
- The user who is protecting the data set has the SPECIAL attribute. A discrete or generic profile can be created **{AC.2::AC.2.13}**.
- The user who is protecting the data set has the group-SPECIAL attribute, and the high-level-qualifier of the data set name is a user within the group-SPECIAL user's scope of authority. A discrete or generic profile can be created **{AC.2::AC.2.14}**.
- The user who is protecting a data set has the OPERATIONS attribute (or the group-OPERATIONS attribute if the data set is within his scope of authority) and is simultaneously creating the data set **{AC.2::AC.2.15}**.

In this case, the user can create a discrete profile:

- Through ADSP **{AC.2::AC.2.16}**
- By specifying the PROTECT operand on the TSO ALLOCATE command that creates the data set **{AC.2::AC.2.17}**
- By specifying the PROTECT=YES OR SECMODEL=profile-name operands on the JCL DD statement that creates the data set **{AC.2::AC.2.18}**

7.3.2.2.2.1.6 Protecting group data sets

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

- The user has JOIN, CONNECT, or CREATE authority in the group **{AC.2::AC.2.19}**;
- The user has the SPECIAL attribute (or the group-SPECIAL attribute for that group) and the request is made using the ADDSD command **{AC.2::AC.2.20}**;
- The user has the OPERATIONS attribute and is not connected to the group **{AC.2::AC.2.21}**.

7.3.2.2.2.1.7 Controlling the creation of new data sets

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

For cataloged data sets, creating, deleting, or renaming the data set involves access not only to the data set profile protecting the data set, but also to the catalog in which the data set is cataloged **{AC.2::AC.2.22}**. In general, users need the following:

- To add entries to the catalog, users need authority to create the data set as specified below and (except for SMS-managed data sets) UPDATE authority to the catalog **{AC.2::AC.2.23}**.
- To delete entries from the catalog, users need ALTER authority to the protecting profile or to the catalog **{AC.2::AC.2.24}**.

The following cases describe how RACF can be used to control the creation of new user and group data sets.

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

- The data set is covered by an existing generic profile and the user does not have ADSP **{AC.2::AC.2.25}**. The creation is allowed if (1) the user has ALTER authority to the data set through a generic profile or global access checking, or (2) the data set is the user's own data set **{AC.2::AC.2.26}**.
- The data set name is not covered by an existing generic profile and the user does not have ADSP and the data set is covered by the Global Access check table granting ALTER. **{AC.2::AC.2.27}**
- The user has ADSP and the data set is the user's own data set. The creation is allowed and RACF creates a discrete profile for the data set **{AC.2::AC.2.28}** .
- The user has the OPERATIONS attribute. If the user has the group-OPERATIONS attribute instead of OPERATIONS, the high-level qualifier of the new data set must be the ID of a user who is within the scope of that group **{AC.2::AC.2.29-R12-RACF}**.

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

- The data set name is protected by an existing generic profile and the user does not have ADSP. The creation is allowed if at least one of the following is true:
- The user has ALTER authority to the data set through the generic profile or global access checking **{AC.2::AC.2.30}**
- The user has CREATE authority in the group **{AC.2::AC.2.31}**
- The data set name is not covered by an existing generic profile and the user does not have ADSP **{AC.2::AC.2.32}**
- The user has ADSP and the data set belongs to a group of which the user is a member. The creation is allowed only if the user has CREATE authority in the group. If the creation is allowed, RACF creates a discrete profile for the data set **{AC.2::AC.2.33}**
- **{AC.2::AC.2.36-R12-RACF}**The user has the OPERATIONS attribute , or the group-OPERATIONS attribute for the group in question (directly or via a superior group), except when both of the following are true: The user is connected to the group with less than CREATE authority **{AC.2::AC.2.34-R12-RACF}**, and the user has less than ALTER access to the data set if it protected by a generic profile **{AC.2::AC.2.35-R12-RACF}**

7.3.2.2.2.1.8 Data set profile ownership

Each data set profile defined to RACF requires a RACF-defined user or group as the owner of the profile. The owner (if a user) has full control over the profile, including the access list **{AC.2::AC.2.37}**.

If the owner of the data set profile is a group, users with group-SPECIAL in that group have full control over the profile **{AC.2::AC.2.38}**.

Ownership of data set profiles is assigned when the profiles are defined to RACF but may be changed later. Note that ownership of a data set profile does not mean that the owner can automatically access that data set. To access a data set, the owner must still be authorized by the DAC rules **{AC.2::AC.2.39-V2R4}**.

7.3.2.2.2.2 Programs

The ability of users to execute programs can be restricted by the RACF program control function. This feature is useful for programs operating with privileges like authorized programs. Program control can for example be used to restrict the ability of a user to start an authorized program from an authorized library in a way such that it executes with APF authorization **{AC.2::AC.2-V1R7-1}**. Users may still have read access to the library and may therefore copy the program into another library and execute it from this library. Although this is possible, the program will then not execute with the privileges it has when executed from the original library **{AC.2::AC.2-V1R7.2}**.

Program control (as described in this section) applies to programs residing in z/OS partitioned data sets or libraries, not to programs stored as part of z/OS UNIX file system. Mechanisms for program control for the z/OS UNIX subsystem are explained in another section of this Security Target.

z/OS allows for three modes for program control: BASIC, ENHANCED and ENHANCED-WARNING. The mode is defined by the strings 'BASIC', 'ENHANCED' or 'ENHANCED-WARNING' in the APPLDATA field of the IRR.PGMSECURITY profile in the FACILITY class **{AC.2::AC.2.V1R7.3}**. An empty value or any other value than 'BASIC' or 'ENHANCED' will result in the ENHANCED-WARNING mode **{AC.2::AC.2.V1R7.4}**. If the IRR.PGMSECURITY profile is not defined, BASIC mode is used **{AC.2::AC.2.V1R7.5}**. In ENHANCED-WARNING mode the access decisions made by the TOE are the same as in BASIC mode but a warning message is issued whenever the access would have been denied in ENHANCED mode **{AC.2::AC.2.V1R7.6}** .

The checks that RACF makes when a user makes a request to load (execute) a program are:

i. If program control has been activated with SETROPTS WHEN(PROGRAM) **{AC.2::AC.2-V1R7.7}**
ii. If program control is active, RACF checks to see whether the program is protected by a profile in the PROGRAM class **{AC.2::AC.2-V1R7.8}**
iii. If the program is not protected, RACF determines whether there are any data sets currently open using PADS or whether there are any execute-controlled programs in storage in the address space:
    - If there are no such data sets or programs, RACF marks the environment dirty (uncontrolled) and allows the user to execute the program **{AC.2::AC.2-V1R7.9}**.
    - If there are data sets currently opened using PADS, or programs to which the user has only EXECUTE authority, RACF fails the request and the system abends the task. RACF issues message ICH423I to document the execute-controlled programs, or message ICH424I to document the PADS data sets that caused the operation to fail. In this way, RACF prevents uncontrolled programs from gaining access to protected data or programs inappropriately **{AC.2::AC.2-V1R7.10}**.
iv. If the program is protected by a profile but the user does not have at least EXECUTE authority to the program, RACF causes the system to abend the task because the user is not authorized to execute the program **{AC.2::AC.2-V1R7.11}**.

v. If the program is protected by a profile and the user has only EXECUTE authority to the PROGRAM profile or to the library that contains the program (when the program is loaded from a JOBLIB, STEPLIB, or tasklib), and if the job step or TSO session is running in ENHANCED program security mode, RACF checks whether an appropriate program established the program environment. RACF determines if the first program executed in the job step had the 'MAIN' attribute, or (if necessary) if the program invoked by TSOEXEC or IKJEFTSR had the 'MAIN' attribute. If the program does not have MAIN, RACF next determines if the first program run in the current task (TCB) or the first program executed in some parent task had the 'BASIC' attribute. If so, RACF allows the Program control request. Otherwise, RACF fails the request and issues message ICH429I to describe the problem and tell you what program established the environment **{AC.2::AC.2-V1R7.12}**.

vi. If the user is still authorized to execute the program and the program was defined with the PADCHK attribute, RACF checks whether any program-accessed data sets are open.

- If no program-accessed data sets are open, RACF allows the user to execute the program **{AC.2::AC.2-V1R7.13}**.
- If program-accessed data sets are open, RACF checks the user or program combination to verify that the combination has at least the same authority to each data set in the list that was required when each data set was opened.
- If the user or program combination has sufficient authority to all of the opened data sets, RACF allows the user to execute the program **{AC.2::AC.2-V1R7.14}**
- If the user or program combination does not have sufficient authority to all of the opened data sets, RACF causes the system to end the task (with abend code 306 or 806) **{AC.2::AC.2-V1R7.15}**.

With program control enabled, z/OS provides the ability to allow users to access data sets which they are not allowed to access directly by using program controlled programs **{AC.2::AC.2.V1R7.16}**.

The following algorithm is used to determine if a user has access to a data set via a controlled program:

Whenever the user has the requested access to the data set as determined by normal RACF access checking, access is granted **{AC.2::AC.2.V1R7.17}**.

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

- The conditional access list contains the name of the currently running program, the name of the first program currently running in the current task (TCB), or the name of the first program currently running in a parent task, with the requested level of access or higher **{AC.2::AC.2.V1R7.18}**.
- The user's group or user ID is associated with the program name in the conditional access list **{AC.2::AC.2.V1R7.19}**.
- The current program environment (job step, or task established under TSO/E using TSOEXEC or IKJEFTSR) is controlled. In other words, it has not loaded an uncontrolled program. If either of these conditions are not met, the environment is considered uncontrolled. The user's attempt to open the program-accessed data set fails and the task ends with abend code 913. RACF issues message ICH417I, specifying what caused the environment to become uncontrolled **{AC.2::AC.2.V1R7.20}**.
- If the job step or TSO session is running in ENHANCED program security mode, one of the following is true:

- The current environment (job step or task created by TSOEXEC or IKJEFTSR) first ran a program defined with the 'MAIN' attribute.
- The current program running in the current task, or the first program run in the current task or a parent task, has the BASIC attribute. If neither of these conditions is met, the user's attempt to open the program-accessed data set fails and the task ends with abend code 913. RACF issues message ICH426I, specifying the non-MAIN program that established the current environment **{AC.2::AC.2.V1R7.21}** .
- If there is more than one controlled program running in the current environment (job step or task created by TSOEXEC or IKJEFTSR), all of those programs defined with the PADCHK attribute have conditional access list entries allowing them to access the data set. If one or more programs in the environment are not authorized, the attempt fails and the task terminates with abend code 913. RACF issues message ICH418I specifying one or more programs that were missing from the conditional access list **{AC.2::AC.2.V1R7.22}**.
- If all the conditions for program access to data set are met and the requested type of access is granted to the program by the profile protecting the data set, access is granted **{AC.2::AC.2.V1R7.23}**.

7.3.2.2.2.3 DAC for MVS Resources

RACF controls the types of access to all MVS (non-UNIX) resources. The access types are ordered hierarchically, an access type listed higher in the list implies all the access types lower in this list (except for NONE access). The full semantics of each access type are defined by the resource manager. The semantics for MVS data sets are:

- ALTER

  ALTER allows users to read, update, delete, rename, move, or scratch the data set.

  When specified in a discrete profile, ALTER allows users to read, alter, and delete the profile itself including the access list **{AC.4::AC.4.1}**.

  ALTER does not allow users to change the owner of the profile using the ALTDSD command **{AC.4::AC.4.2}**. However, if a user with ALTER access authority to a discrete data set profile renames the data set, changing the high-level qualifier to his or her own user ID, both the data set and the profile are renamed, and the OWNER of the profile is changed to the new user ID **{AC.4::AC.4.3}**.

  When specified in a generic profile, ALTER gives users no authority over the profile itself **{AC.4::AC.4.4}**.
- CONTROL

  For VSAM data sets, CONTROL is equivalent to the VSAM CONTROL password; that is, it allows users to perform improved control interval processing. This is control-interval access (access to individual VSAM data blocks), and the ability to retrieve, update, insert, or delete records in the specified data set **{AC.4::AC.4.5}**.

  For non-VSAM data sets, CONTROL is equivalent to UPDATE **{AC.4::AC.4.6}**.
- UPDATE

  Allows users to read from, copy from, or write to the data set **{AC.4::AC.4.7}**. UPDATE does not, however, authorize a user to delete, rename, move, or scratch the data set **{AC.4::AC.4.8}**.
- READ

  Allows users to access the data set for reading only **{AC.4::AC.4.9}**. (Note that users who can read the data set can copy or print it.)

- EXECUTE

  For a private load library, EXECUTE allows users to load and execute, but not to read or copy programs (load modules) in the library **{AC.4::AC.4.10}**.
- NONE

  The specified user or group is not permitted to access the resource or list the profile **{AC.4::AC.4.11}**.

These access types can be defined per user, group or for all users not addressed specifically by a user or group access entry (“universal access”) **{AC.4::AC.4.12}**. It is also possible to specify ID(*) in an ACL, which then applies to all RACF defined users, while the value for UACC applies to users not defined in RACF **{AC.4::AC.4.13}**. To modify those entries (as well as other parts of the resource profile) a user must be the owner of the profile, have ALTER access to the discrete profile of the resource or must have the SPECIAL attribute in his user profile **{AC.4::AC.4.14}**.

The access lists defined in a profile can be either a standard access lists, allowing access in general or a conditional access lists allowing access under defined conditions. Possible conditions are:

- the user must be logged on using a defined terminal that the user has been granted access to **{AC.4::AC.4.15}**
- the user must be logged on to a defined console **{AC.4::AC.4.16}**
- the batch job requesting access must have been submitted from a defined JES input device **{AC.4::AC.4.17}**
- the user must have entered the system from a defined network port **{AC.4::AC.4.18}**
- the resource manager has asserted a criteria, such as the name of an SQL role (SQLROLE), which applies to this check, on the authorization request (note: this applies only to a FASTAUTH type of authorization check) **{AC.4::AC.4-R8-RACF-1}** .

Access to resources can be controlled by discrete resource profiles or generic profiles for a set of resources of the same type. Discrete profiles protect one single resource (e. g. one data set) while generic profiles can be used to define a whole set of resources and protect them using a single profile based on patterns in the resource name. Whenever a discrete profile exists for a resource it has precedence over a generic profile that also would apply for the resource **{AC.4::AC.4.19}**. If more than one generic profiles would apply, z/OS always chooses the most specific profile applicable based on a matching algorithm **{AC.4::AC.4.20}**.

The access types above also apply to MVS resources other than data sets (called general resources). However while the usages remain hierarchical in definition (ALTER includes UPDATE, UPDATE includes READ, etc.) the interpretation and usage of the access types is the responsibility of each resource manager. For most resource managers and resources, the meaningful access types are NONE (the user/group has no access) or READ (the user/group does have access). For most cases access levels higher than READ convey no added authority (except that ALTER allows administration of a discrete profile). In specific cases the resource manager may treat UPDATE, CONTROL, and ALTER as granting additional authority. This security target and evaluation will not address all of those cases.

7.3.2.2.2.4 Algorithm to check DAC Access to MVS Resources

To perform authorization checking for RACF-protected resources, RACF makes the following checks. RACF stops processing when the request is granted or denied. This algorithm takes the mandatory SETROPTS settings for RACF in the evaluated configuration into account.

Note: *Statements with a grey background are not relevant for the evaluated configuration.*

Mandatory SETROPTS options are:
ADSP, CATDSNS(FAILURE), NOCOMPATMODE, CLASSACT(TEMPDSN), ERASE(ALL), GENERIC(*) , ENHANCEDGENERICOWNER, GLOBAL(*), JES(BATCHALLRACF), PROTECTALL(FAILURES)

1. For general resource classes, if the class of the resource is not active, RACF returns the "not protected" return code. **{AC.4::AC.4-V2R4-RACF-1}**
2. If the RACF class must be RACLISTed, as specified in the class descriptor table (CDT), but is not currently RACLISTed, RACF returns the "not protected" return code. **{AC.4::AC.4-V2R4-RACF-2}**
3. If the user requesting access is "trusted" or "privileged", RACF grants the request. See the following:
   - If the user has the trusted attribute, RACF grants the request (unless the CSA or PRIVATE operand was specified on the authorization request). Such requests can be audited only by using the LOGOPTIONS operand on the SETROPTS command (which audits access requests issued by all users) or the UAUDIT operand on the ALTUSER command (which audits all access requests by a particular user).
   - If the user has the privileged attribute, RACF grants the request (unless the CSA or PRIVATE operand was specified on the authorization request). Such requests cannot be audited.

   {AC.4::AC.4-V2R4-RACF-3}
4. RACF invokes the naming convention table if:
   - The naming convention routine exists
   - The resource being checked is a CLASS data set

   The naming convention table can continue REQUEST=AUTH processing or deny the request. **{AC.4::AC.4-V2R4-RACF-4}**
5. If global access checking is active for the class, RACF searches the global access table (unless the CSA or PRIVATE operand was specified on the authorization request). If RACF finds a matching entry that allows access to the resource, RACF grants the request for all users, except those with the RESTRICTED attribute. **{AC.4::AC.4-V2R4-RACF-5}**
6. RACF looks for a profile in storage or in the RACF database. If no profile is found that protects the resource, RACF returns the default return code of the class. Specifically, no profile is found in the following cases:
   - Profiles for the class exist in the user's storage or in a data space, but no profile matches the resource name.
   - Profiles for the class do not exist in the user's storage, in a data space, or in the RACF database.

   {AC.4::AC.4-V2R4-RACF-6}
7. If users attempt to access their own resources, RACF grants the request. For example:
   - For tape and DASD data sets, if the user ID of the requesting user is the high-level qualifier of the data set name, RACF grants the request.
   - For spool data sets, if the JESSPOOL class is active, RACF compares the user ID and node of the requester with the user ID and node of the creator of the spool data set (using the security token). If the user IDs match, RACF grants the request.

   {AC.4::AC.4-V2R4-RACF-7}
8. RACF checks the user's access authority in the standard access list. If the user is in the list and if the specified access authority is sufficient to allow access, RACF grants the request. If the user is in the list and if the specified access authority is less than the requested access, RACF continues processing at Step **(III)** (conditional access list checking). This prevents access based on ID(*), UACC, or the OPERATIONS attribute. **{AC.4::AC.4-V2R4-RACF-8}**

9. RACF determines whether the user has access to the resource because the user is a member of a group and the group is on the standard access list. Which group is used depends on whether list-of-groups processing is in effect. (List-of-groups processing is in effect if the SETROPTS command has been issued with the GRPLIST operand.) RACF determines which group to use according to the following rules:
    - If list-of-groups processing is not in effect, RACF uses only the user's current connect group.
    - If list-of-groups processing is in effect, RACF finds all of the groups to which the user is connected that are also in the access list. Of these groups, RACF uses the group that has the highest access authority to the resource. (For example, assume that a user is a member of groups A, B, and C. If group A has NONE access authority, group B has READ access authority, and group C has UPDATE access authority, RACF uses group C to determine the user's access.)

    If the highest access authority is sufficient to allow the requested access, RACF grants the request. If the highest group that was found in the list does not have the requested authority, RACF continues processing at Step **(I)** (conditional access list checking). This prevents access based on ID(*), UACC, or the OPERATIONS attribute. **{AC.4::AC.4-V2R4-RACF-9}**
10. If a user ID of * is found on the standard access list, the current user is defined to RACF without the RESTRICTED attribute, and the access authority granted to * is:
    - Sufficient to allow the requested access, RACF grants the request.
    - Not sufficient to allow the requested access, RACF continues processing at Step **(II)** (OPERATIONS attribute checking).

    {AC.4::AC.4-V2R4-RACF-10}
11. If the universal access authority (UACC) for the resource provides sufficient access authority and the requesting user is not defined with the RESTRICTED attribute, RACF grants the request. **{AC.4::AC.4-V2R4-RACF-11}**
12. **(II)** If the requesting user has the OPERATIONS attribute (or group-OPERATIONS if the resource is within the scope of that group) and OPERATIONS access is allowed for the class, RACF grants the request. **{AC.4::AC.4-V2R4-RACF-12}**
13. **(I)** RACF checks the user's access authority in the conditional access list specified with WHEN(TERMINAL), WHEN(CONSOLE), WHEN(APPCPORT), WHEN(JESINPUT) or WHEN(SERVAUTH). If the user is in the list, if the user meets the specified condition (such as logged on at the specified terminal), and if the specified access authority is sufficient to allow access, RACF grants the request. If the user is in the list with insufficient access authority, RACF authorization processing continues at Step **(III)**. **{AC.4::AC.4-V2R4-RACF-13}**
14. RACF determines whether the user has access to the resource because the user is a member of a group that meets a condition specified on the conditional access list specified with WHEN(TERMINAL), WHEN(CONSOLE), WHEN(APPCPORT), WHEN(JESINPUT) or WHEN(SERVAUTH). Which group is used depends on whether list-of-groups processing is in effect. (List-of-groups processing is in effect if the SETROPTS command has been issued with the GRPLIST operand). RACF determines which group to use according to the following rules:
    - If list-of-groups processing is not in effect, RACF uses only the user's current connect group.

- If list-of-groups processing is in effect, RACF finds all of the groups to which the user is connected that are also in the access list. Of these groups, RACF uses the group that has the highest access authority to the resource. (For example, assume that a user is a member of groups A and B. If A has READ access authority and B has UPDATE access authority, RACF uses group B to determine the user's access.)

If the group to be used according to the preceding rules has sufficient access authority to allow the requested access, RACF authorization processing continues at Step 19. If none of the user's groups has sufficient authority, RACF does not grant the request, and continues with the next step. **{AC.4::AC.4-V2R4-RACF-14}**

15. If a user ID of * is found on the conditional access list specified with WHEN(TERMINAL), WHEN(CONSOLE), WHEN(APPCPORT), WHEN(JESINPUT) or WHEN(SERVAUTH), and if the current user is defined to RACF without the RESTRICTED attribute, and if the current user meets the specified condition (such as logged on at the specified terminal), and the access authority granted to * is sufficient to allow the requested access, RACF grants the request. **{AC.4::AC.4-V2R4-RACF-15}**
16. **(III)** RACF checks the user's access authority in the conditional access list specified with WHEN(PROGRAM). If the user is in the list, if the user meets the specified condition (such as running the specified program), and if the specified access authority is sufficient to allow access, RACF grants the request.

    Note: For DASD data sets, if program control is active and a controlled program is executing, RACF performs authorization checking for program access to data sets. If the user/program combination is in the conditional access list with sufficient authority to allow access to the data sets, RACF grants the request. If the user/program combination is in the conditional access list with insufficient authority, RACF denies the request.

    RACF determines whether the user has access to the resource because the user is a member of a group that meets a condition specified on the conditional access list (such as running a specified program). Which group is used depends on whether list-of-groups processing is in effect. (List-of-groups processing is in effect if the SETROPTS command has been issued with the GRPLIST operand.) RACF determines which group to use according to the following rules:

    - If list-of-groups processing is not in effect, RACF uses only the user's current connect group.
    - If list-of-groups processing is in effect, RACF finds all of the groups to which the user is connected that are also in the access list. Of these groups, RACF uses the group that has the highest access authority to the resource. (For example, assume that a user is a member of groups A and B. If A has READ access authority and B has UPDATE access authority, RACF uses group B to determine the user's access.)

    If the group to be used according to the preceding rules has sufficient access authority to allow the requested access, RACF grants the request. If the group is specified in the list with insufficient access authority, RACF denies the request.

    {AC.4::AC.4-V2R4-RACF-16}
17. If a user ID of * is found on the conditional access list specified with WHEN(PROGRAM), and if the current user is defined to RACF without the RESTRICTED attribute, and if the current user meets the specified condition (such as logged on at the specified terminal or running the specified program), and the access authority granted to * is sufficient to allow the requested access, RACF grants the request. **{AC.4::AC.4-V2R4-RACF-17}**
18. If the WARNING flag is ON in the profile (set using the WARNING operand on the ADDSD, ALTDSD, RDEFINE, or RALTER command), RACF grants the request. **{AC.4::AC.4-V2R4-RACF-18}**

19. Since SETROPTS CATDSNS(FAILURES) is in effect in the evaluated configuration, RACF denies the request unless at least one of the following is true:
    - The data set is newly created in this job, or is a system temporary data set.
    - The data set is on tape, and the request is for UPDATE access.
    - The data set is protected by a discrete profile.
    - The data set is cataloged in the master catalog.
    - The user has access to FACILITY resource ICHUNCAT.data set-name (truncated to 39 characters total, if needed);
    - The user has the SPECIAL attribute.

    Note: If the user gains access through having the SPECIAL attribute and none of the prior conditions were true, RACF issues a warning message and creates an SMF record as though CATDSNS(WARNING) were in effect. **{AC.4::AC.4-V2R5-RACF-19}**
20. For the DATASET class, if no profile is found, RACF denies the request since in the evaluated configuration the SETROPTS PROTECTALL(FAILURES) option is in effect. **{AC.4::AC.4-V2R4-RACF-20}**

When the SECLABEL class is active on your system, and a user or job requests access to a resource, RACF compares the security label of the user with the security label of the resource using the algorithm below before checking for discretionary access. Note that using SETROPTS NOMLS is not allowed in the evaluated configuration when the SECLABEL class is activated.

1. If the user requesting access does not have a security label and the resource does have a security label, RACF fails the request.
2. **(IV)** If the SETROPTS MLACTIVE(FAILURES) option is in effect and the resource does not have a security label associated with it, and the resource class is DATASET or another class that requires security labels as defined in the class descriptor table (CDT), RACF fails the request.
3. If the SETROPTS MLACTIVE(WARNING) option is in effect, RACF makes the same checks as in Step **(IV)** If the access check fails because the resource does not have a security label, RACF issues a warning message and grants the request.
4. **(V)** If the SETROPTS MLS(FAILURES) option is in effect, RACF checks for read-only request if the user's security label dominates the security label of the resource and fails the request if this is not the case. For a write request (which implies read), RACF tests if the user's security label and the security label of the subject are equivalent and denies the request if they are not.
5. If the SETROPTS MLS(WARNING) option is in effect for this resource class, RACF makes the same checks as in Step **(V)**. If any test fails the request, RACF issues a warning message and grants the request.
6. If the resource is a JES spool data set, RACF uses the security label in the token associated with the data set (specified on the RTOKEN parameter of the RACROUTE REQUEST=AUTH macro). Otherwise, RACF uses the security label kept in the resource profile protecting the resource, in the FSP for files, or in the ISP for IPC objects.

7.3.2.2.2.5 Access Control to System Logger Objects

Please refer to section Using a System Log Stream for SMF.

7.3.2.2.2.6 z/OS UNIX file system objects

UNIX file system objects in the zFS file system have their access control defined by:

- UNIX permission bits
- Access control list entries

All of those access-control-related attributes of file system objects are stored with the object. Access control lists are stored and managed as extended attributes of the file system object and are not stored in the RACF database **{AC.2::AC.2.65-V2R4}**. RACF is still involved when an access decision is made to a UNIX file system object **{AC.2::AC.2.66}**. The UNIX System Services subsystem of the TOE extracts the permission bits, access control list entries from the file system object as well as the effective user ID the user that performed the request and passes this information to RACF using the ck_access RACF callable service. RACF then evaluates this information, extracts other information relevant for the access decision from the RACF database, performs the auditing in accordance with the audit policy defined by the system administrator and returns the access decision to the calling UNIX System Services subsystem of the TOE **{AC.2::AC.2.67-V2R4}**.

Besides the access control lists, additional privileges and restrictions may be defined to allow a finer granularity. Those privileges and restrictions are defined as profiles in the UNIXPRIV class and users can be granted those privileges or restrictions by giving them authority to those profiles. The ones that are considered in this Security Target are:

- SUPERUSER.FILESYS.ACL.ACLOVERRIDE

  When this profile is defined and active in RACF, a user who has been given authority to this profile is able to override the access control defined by the access control lists for z/OS UNIX file system objects.

  In z/OS, a UNIX superuser can access all z/OS UNIX files, but is still bound by his rights defined in RACF with respect to z/OS data sets and other resources **{AC.2::AC.2.68-V2R4}**.
- SUPERUSER.FILESYS.DIRSRCH

  Users with at least READ access to this profile are allowed to search directories regardless of the general access control settings **{AC.2::AC.2-V2R4-6}**

7.3.2.2.2.7 z/OS UNIX IPC objects

z/OS UNIX IPC objects are subject to discretionary access control. The permission bits associated with the IPC object define the discretionary access to those objects. The permission bits are determined by the creator of the IPC object and are saved in-memory by the UNIX Kernel. UNIX System Services will collect the permission bits from the IPC object and call RACF using the ck_IPC_access RACF callable service. RACF will then determine if the user can be granted the requested type of access and returns the decision to UNIX System Services. For security claims see DAC for UNIX objects.

7.3.2.2.2.8 DAC for UNIX objects

DAC controls for UNIX objects involve the user's effective UID and effective GID (which may be different from the user's real UID and real GID) **{AC.4::AC.4-R8-USS-1}** and the user's supplemental GIDs. If the user is connected to 5 groups, and 3 of them have GIDs, then he would have one real GID and 2 supplemental GIDs **{AC.4::AC.4-R8-USS-2}**.

DAC checking for UNIX file objects (files, directories) involves permission bits that specify the permissions (read, write, execute/search) separately for the object's owner, the owning group, and everyone else (the world), and optional access list entries (ACLs) with similar permission settings.

DAC checking for UNIX IPC objects (semaphores, shared memory) involves only permission bits.

7.3.2.2.2.9 Algorithm to check DAC access to UNIX file system objects

To perform authorization checking for z/OS UNIX files and directories, RACF makes the following checks. RACF stops processing when the request is granted or denied.

Notes:

The effective UID and effective GID of the process is used in determining access decisions. The only exception is that when CREDFUNCTION is AFC_ACCESS, the real UID and real GID are used. In other words, if file access is being tested, rather than requested, the real UID and GID are used instead of the effective UID and GID. The real and effective IDs are generally the same for a process, but if a set-uid or set-gid program is executed, they can be different.

If the requesting user is represented by an unauthenticated client ACEE, then the access check algorithm defined below is performed first for the client, and then, if successful, for the server. Both client and server must have access to the file in order for the request to succeed.

Statements with a grey background are not relevant for the evaluated configuration.

1. If the system (kernel) is the caller, then access is failed if either of the following conditions occurs:
   - The request includes execute authority for a file and execute authority cannot be granted. In this condition, none of the permissions bits grant execute access, and, if an ACL is present and the FSSEC class is active, no ACL entry grants execute access.
   - Security label authorization checking fails. In this condition, the SECLABEL class is active, the object being accessed is a directory, the directory's SECLABEL is not SYSMULTI, and the CRED contains a SECLABEL.

   Otherwise, access is granted. **{AC.4::AC.4-V2R4-UNIX-1}**
2. RACF checks for a profile in the FSACCESS class that covers the file system name when all of the following conditions are met:
   - The user does not have the AUDITOR or ROAUDIT attribute.
   - A file system name was specified in the CRED.
   - The FSACCESS class is active and RACLISTed.

   If a matching profile is found and the user does not have at least UPDATE authority, access is denied. Otherwise, access checking continues. **{AC.4::AC.4-V2R4-UNIX-2}**
3. RACF checks for a profile in the FSEXEC class that covers the file system name when all of the following conditions are met:
   - The request is for file execution access.
   - A file system name was specified in the CRED.
   - The FSEXEC class is active and RACLISTed.

   If a matching profile is found and the user does not have at least UPDATE authority, access is denied. Otherwise, access checking continues. **{AC.4::AC.4-V2R4-UNIX-3}**
4. If the SECLABEL class is not active, then go to Step **(I)**. **{AC.4::AC.4-V2R4-UNIX-4}**
5. If the user has the TRUSTED or PRIVILEGED attribute, then access is granted automatically unless the user is executing a file. If the user is executing a file, access is denied only if none of the permissions bits grant execute access, and, if an ACL is present and the FSSEC class is active, no ACL entry grants execute access. Otherwise, access is granted. **{AC.4::AC.4-V2R4-UNIX-5}**
6. If the user has the RACF AUDITOR or ROAUDIT attribute, and has read or search access for a directory is requested, access is granted. **{AC.4::AC.4-V2R4-UNIX-6}**
7. If SETROPTS MLFSOBJ is active and the file does not have a security label, the request is failed.
8. If SETROPTS MLS is active (either in WARNING or FAILURES mode) and all of the following conditions occur, the request is failed.
   - The user has a security label.
   - The file has no security label.

- The user explicitly requested write access but is not in writedown mode.

Note: The SETROPTS MLS(WARNING) option is not supported for UNIX files and directories, and it is treated the same as MLS(FAILURES).

9. If the file has a security label but the user does not, then the request is failed.
10. If the user's security label is equivalent to the security label of the file (this condition is also satisfied if either security label is SYSMULTI), then continue at Step **(II)**.
11. If ANY access is requested, then two security label dominance checks (RACROUTE REQUEST=DIRAUTH) are performed: one for READ and one for WRITE. If either succeeds, then continue at Step **(II)**. Otherwise, the request is failed.
12. If the user is requesting write access along with read or search/execute access, then a READWRITE dominance check is performed. If it succeeds, then continue at Step **(II)**. Otherwise, the request is failed.
13. If the user is requesting only read or search/execute access, then a READ dominance check is performed. If it succeeds, then continue at Step **(III)**. Otherwise, the request is failed.
14. If the user is requesting only write access, then a WRITE dominance check is performed. If it succeeds, then continue at Step **(II)**. Otherwise, the request is failed.
15. **(I)** If the user has the RACF AUDITOR or ROAUDIT attribute, and read or search access for a directory is requested, access is granted. **{AC.4::AC.4-V2R4-UNIX-7}**
16. **(II)** If the user has UID(0), then access is granted automatically unless the user is executing a file. If the user is executing a file, access is denied only if none of the permissions bits grant execute access, and, if an ACL is present and the FSSEC class is active, no ACL entry grants execute access. Otherwise, access is granted. **{AC.4::AC.4-V2R4-UNIX-8}**
17. If the UID matches the file owner UID, the file's "owner" permission bits are checked. If the "owner" bits allow the requested access, then access is granted. Otherwise, go to Step **(IV)**. **{AC.4::AC.4-V2R4-UNIX-9}**
18. If the FSSEC class is active, and an ACL exists, and there is an ACL entry for the requesting UID, then the permission bits of that ACL entry are checked. If the ACL entry allows the requested access, then access is granted. Otherwise, go to Step **(V)**. **{AC.4::AC.4-V2R4-UNIX-10}**
19. If the GID matches the file owner GID, the file's "group" permission bits are checked. If the "group" bits allow the requested access, then access is granted. **{AC.4::AC.4-V2R4-UNIX-11}**
20. **(III)** If the FSSEC class is active, and an ACL exists, and there is an ACL entry for the requesting GID, then the permission bits of that ACL entry are checked. If the ACL entry allows the requested access, then access is granted. If not, then the next ACL entry is checked until there are no more entries. **{AC.4::AC.4-V2R4-UNIX-12}**
21. If any of the user's supplemental GIDs match the file owner GID, the file's "group" permission bits are checked. If the "group" bits allow the requested access, then access is granted. **{AC.4::AC.4-V2R4-UNIX-13}**
22. If the FSSEC class is active, and an ACL exists, and there is an ACL entry for any of the user's supplemental GIDs, then the permission bits of that ACL entry are checked. If the ACL entry allows the requested access, then access is granted. If not, then the next ACL entry is checked until there are no more entries. **{AC.4::AC.4-V2R4-UNIX-14}**
23. If at least one matching ACL entry was found for the GID, or any of the supplemental GIDs, then processing continues with Step **(V)**. If the GID, or any of the supplemental GIDs, matched the file owner GID, then processing continues with Step **(IV)**. Otherwise (neither the GID nor any of the supplemental GIDs matched either the file owner GID or an ACL entry), processing continues with the next step. **{AC.4::AC.4-V2R4-UNIX-15}**

24. If the requesting user has the RESTRICTED attribute, and the UNIXPRIV class is active and RACLISTed, and the RESTRICTED.FILESYS.ACCESS resource is protected by a profile in the UNIXPRIV class, and the user does not have at least READ access, then go to Step **(IV)**. **{AC.4::AC.4-V2R4-UNIX-16}**
25. The file's "other" permission bits are checked. If the "other" bits allow the requested access, then access is granted. Otherwise, go to Step **(IV)**. **{AC.4::AC.4-V2R4-UNIX-17}**
26. **(V)** If the UNIXPRIV class is active and RACLISTed, and if the SUPERUSER.FILESYS.ACLOVERRIDE resource is protected by a profile in the UNIXPRIV class, then the user must have the correct access level as documented for the ck_access (IRRSKA00) callable service in z/OS Security Server RACF Callable Services. If the profile exists, it determines whether file access is granted or denied. **{AC.4::AC.4-V2R4-UNIX-18}**
27. **(IV)** If the request is for directory read or search, the UNIXPRIV class is active and RACLISTed, and the user has at least read permission to the SUPERUSER.FILESYS.DIRSRCH resource, then access is granted. Otherwise, if the UNIXPRIV class is active and RACLISTed, and if the SUPERUSER.FILESYS resource is protected by a profile in the UNIXPRIV class, then the user must have the correct access level as documented for the ck_access (IRRSKA00) callable service in z/OS Security Server RACF Callable Services. If the profile exists, it determines whether file access is granted or denied. **{AC.4::AC.4-V2R4-UNIX-19}**
28. If this step is reached, access is denied. **{AC.4::AC.4.42}**

7.3.2.2.2.10 Algorithm to check DAC access to UNIX IPC objects

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

- if the user has an effective user ID of zero **{AC.4::AC.2.70}**
- if the user is the owner or creator of the IPC object and the requested type of access is allowed by the owner related permission bits **{AC.4::AC.2.71}**
- if the user is neither the owner or creator of the IPC object but is a member of the IPC object's creating group or owning group and the requested type of access is allowed by the group related permission bits **{AC.4::AC.2.72}**
- if the user is neither owner nor creator of the IPC object and also is not a member of the IPC object's creating group or owning group and the access is allowed by the other related permission bits **{AC.4::AC.2.73}**

If none of the above mentioned conditions is satisfied, permission is denied by the discretionary access control rules for IPC objects **{AC.4::AC.2.74}**.

7.3.2.3 Audit (FAU)

7.3.2.3.1 Generation of audit records

The TOE provides a general facility to collect data required for auditing and accounting services. This function, the System Management Facilities (SMF), collects and records system and job-related information that an installation can use for such tasks as the following:

- Billing users
- Reporting reliability
- Analyzing the configuration
- Scheduling jobs
- Summarizing direct access volume activity
- Evaluating data set activity

- Profiling system resource use
- Maintaining system security

This component is used by the TOE to collect security-related auditing information as required by FAU_GEN.1.

Each SMF record consists of a standard header which contains (among other information) the type of the record and the time the record was produced **{AU.1::AU.1.1}**. SMF supports up to 256 different record types. SMF records can only be generated by authorized processes or processes specifically authorized to generate specific types of SMF records under the mediation of the TOE **{AU.1::AU.1.2}**.

One record type is usually reserved for a whole class of events where the individual events are identified by the record subtype or event code in the header of the SMF record.

RACF as the central access control function has three SMF record types reserved for its use (80, 81, 83), with record type number 80 being the most important one. The information recorded in this record type contains (among other non security related information):

- The record type
- Time stamp (time and date)
- System identification
- Event code and qualifier
- User identification
- Group name
- Authorities used to successfully execute commands or access resources
- Reasons for logging
- Command processing error flag
- Foreground user terminal ID or other port-of-entry information
- Job log number (job name, entry time, and date)
- RACF version, release, and modification number

Each record contains further data specific to the event code and qualifier **{AU.1::AU.1.3}**.

The administrator can configure RACF and other elements of the TOE to generate audit records for all events listed in the table in FAU_GEN.1 **{AU.1::AU.1-R9-MULTI-1}** .

7.3.2.3.2 Protection of the audit trail

SMF writes audit records into either

1. Dedicated SMF data sets that have been defined during system configuration. At least two SMF data sets must be defined by the administrator for compliance with the evaluated configuration. Those data sets need to be protected against unauthorized access by appropriate RACF access control lists. The administrator guidance documentation provides specific guidelines for the protection of the audit trail using RACF.

Or

2. A system log stream, which may reside solely in DASD data sets, or in a combination of data sets and a coupling facility structure for better performance, as specified by the administrator. The administrator configures profiles in the LOGSTRM class to control who can access the data while it exists in the managed log stream, and profiles in the DATASET class to control access to any data extracted from the log stream.

7.3.2.3.2.1 Using MVS Data Sets for SMF

When the system is started SMF searches for the first non-full data set in the list of SMF data sets defined. This data set becomes the active SMF data set used to store audit records. Once this data set is full, SMF marks the data set to be processed by the SMF Dump program and takes the next empty data set as the active, searching the list of SMF data sets in a wraparound way **{AU.2::AU.2.2}**. The operator is also alerted to switch the data set.

SMF data sets that are full need to be processed by the SMF Dump program, IFASMFDP. This program copies the content of a full SMF data set to another data set (the “dump data set”) defined by the installation and marks the SMF data set as empty **{AU.2::AU.2.3}**. The SMF Dump program itself creates two SMF records (Dump Header and Dump Trailer) that are stored in the beginning and at the end of the dump data set **{AU.2::AU.2.4}**. Dump data sets must be protected by RACF access control lists.

If no non-full data set is found, SMF stores the records in its buffers until a data set is made available **{AU.2::AU.2.5}**. If the TOE is configured according to the administrative guidance, the system will halt if no buffer space is left **{AU.2::AU.2.6}**.

7.3.2.3.2.2 Using a System Log Stream for SMF

In contrast to using MVS data sets directly, when using a log stream for the SMF data only one logical stream exists. Although this stream may reside in multiple MVS data sets as determined by system logger processing, the administrator will view the stream as one logical entity, starting with the earliest available data and ending with the current data, rather than dealing with the individual data sets.

Operators do not need to switch SMF data sets, nor dump them to archive storage, nor clear them. Rather, the data can simply reside in the logger-managed data sets.

z/OS provides the IFASMFDL utility program that can extract an administrator-specified set of SMF data from the log stream, based on time/date, system ID, and/or SMF record type and write that extracted data to a standard MVS data set for later processing **{AU.2::AU-R9-SMF-1}**.

IFASMFDL can invoke exit routines, just as IFASMFDP can, and so the RACF SMF Unload routine will work with IFASMFDL just as with IFASMFDP, providing an interpreted flat-file of RACF-relevant security records for subsequent analysis **{AU.2::AU-R9-RACF-1}**.

7.3.2.4 Cryptographic Functions (FCS)

7.3.2.4.1 General Cryptography

Several components of the TOE use cryptographic functions as part of their security functions. With the inclusion of the Integrated Cryptographic Services Facility (ICSF) the cryptographic functions are provided by hardware coprocessors attached to the TOE. ICSF checks for the availability of hardware support for individual cryptographic functions and uses this in the TOE in its evaluated configuration. The TOE will use the support provided by CPACF and ICSF for AES, and the SHA-2 family of hash functions in its evaluated configuration. For the RACDCERT command, ICSF is used.

This support function is used by RACDCERT for certificate public/private key pair generation using the new secure PKCS#11 support. See the description for the RACDCERT command.

7.3.2.4.1.1 Cryptographic Functions supported by the TOE

The TOE also provides various cryptographic functions via ICSF that are available for use by applications running on the system.

The TOE suports the following cryptographic primitives:

Encryption

The TOE has the capability to perform encryption and decryption operations with the following algorithms and key sizes

- AES with CBC or CTR block chaining modes and 256-bit keys as defined in NIST SP 800-38A.
- AES in XTS mode with 256-bit keys as defined in NIST SP 800-38E.
- AES in GCM mode with 256 bit keys NIST SP 800-38D. And in the context of TLS processing, according to [RFC5289].

CR.2::CR-V2R5-COP-1

Hashing

The TOE has the capability to perform message digest generation in accordance with the following cryptographic algorithms:

- SHA-256
- SHA-384
- SHA-512

with the implementation following FIPS Pub 180-4. **CR.2::CR-V2R5-COP-2**

Digital Signatures

The TOE has the capability to generate and verify signatures using the following algorithms and key sizes:

- RSA with 3072 to 4096-bit keys using the RSASSA-PSS scheme (PKCS#1) FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 4.

  It should be noted that the TOE in its evaluated configuration does not generate RSA based digital signatures.
- ECDSA with:
  - NIST curve=384
  - NIST curve=521

  as defined in FIPS 186-4 and specified in ANSI X9.62-1998, Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA).

CR.2::CR-V2R5-COP-3

HMAC

The TOE has the capability to perform keyed-hash message authentication with the following cryptographic algorithms:

- SHA-256
- SHA-384
- SHA-512

The used key sizes are between 112 and 2048 bits, and the corresponding message sizes ares 256-, 384 and 512-bits respectively. The implementation follows FIPS Pub 198-1 The Keyed-Hash Message Authentication Code and FIPS Pub 180-4 Secure Hash Standard. **CR.2::CR-V2R5-COP-4**

7.3.2.4.2 Cryptographic Key Destruction

z/OS provides two principal methods protecting cryptographic keys. Both of them are not subject to this evaluation as they are provided be the operational environment and are mentioned here for information to only (please refer to [ICSF.OVW], chapter 4 for additional information):

- Secure key, and
- Protected key

Secure keys are keys that are generated and used in the CryptoExpress cryptographic coprocessor cards only. Operations with those keys are performed on those coprocessors only. Those keys never leave the coprocessor in unencrypted form.

Protected keys are keys generated and established by an ICSF administrator and are immediately after generation wrapped by a wrapping key inside the processor firmware. Instead of providing the cryptographic operations implemented in the processor by CPACF with a key in clear, one may also provide those operations with the wrapped key. The processor firmware checks the integrity of the wrapped key provided and, if the integrity check passed successfully, unwraps the key in the firmware, performs the requested cryptographic operation, and returns the result to the caller of the instruction. This allows protecting keys even when they need to be used by applications that have no need to get access to the key in clear. For those wrapped keys there is no need for explicit key destruction.

In addition, there are clear keys, where the owning application is able to obtain the clear value of the key.

Keys used by the TSF are either clear session keys that are generated and held in volatile storage only or long-living keys (public or private keys used for asymmetric encryption or long-living keys for symmetric encryption) which are stored in non-volatile storage, which are data sets managed and controlled by the TSF component ICSF.

Clear keys in volatile storage are usually either overwritten with new key values or - when in dynamically allocated memory - are overwritten with zeros as part of the operating system functionality when the memory is released. It has to be noted that all key material used for cryptographic functions described in this Security Target when in volatile memory are in memory that is assigned to and is accessible by the TSF only. Within the TOE however, keys are finally destructed when the power to the memory is removed.

This applies to SSH and AT-TLS. They all perform their cryptographic functions within their own address space which is part of the TSF and protected from access by functions in other address spaces from the operating system.

Long-living keys used by the TSF are stored in data sets managed by ICSF, which are encrypted using secure key hat is held and managed solely by the cryptographic co-processor. Symmetric keys are are encrypted using a symmetric master keys, where asymmetric keys are encrypted using an asymmetric master key. As indicated that master key never leaves the cryptographic co-processor in the clear. The managed keys never exist in the TOE's volatile memory in the clear.

ICSF also provides functions where users can use keys without having any access that would allow them to access keys directly. Instead users are provided with key handles (if they are allowed by RACF to use that key), which they can use to request ICSF to use the key in cryptographic functions and ICSF will just provide the result of those functions to the user. In all those cases the keys themselves are never stored in a user's address space and therefore users cannot access the keys.

Persistent ICSF keys are stored in VSAM data sets which are overwritten with spaces before the space is eligible for allocation to another user. **{CR.1::CR-V2R5-CKM4-1}** As all keys managed by ICSF are stored encrypted using a master key that is managed by the cryptographic co-processor, they are deleted once their master key is deleted. **{CR.1::CR-V2R5-CKM4-2}**

The following table summarizes the keys used by the TOE, their storage as well as how they are destructed:

Table 10: Cryptographic Keys used by the TOE

Key	Generator	Storage	Destruction
TLS Session Keys	ICSF	Volatile	Removal of power to memory
SSH Session Keys	ICSF	Volatile	Removal of power to memory
Dataset encryption keys	ICSF	Persistent	Destruction of the abstraction that represents the key
System SSL private keys	ICSF	Persistent	Destruction of the abstraction that represents the key
SSH private keys	ICSF	Persistent	Destruction of the abstraction that represents the key

7.3.2.4.3 Random Number Generation

The ICSF PKCS#11 implementation generates random bytes in compliance with NIST SP 800-90A "Hash_DRBG". The Hash_DRBG is hardware (CPACF) based and is a SHA-512 Hash DRBG with a security strength of 256 bits. The DRBG is seeded by a hardware based (CPACF) TRNG. **{CR.1::CR-V2R4-DRBG-1}**

The CEX8C card is designed to meet NIST SP800-90B using a NIST SP 800-90A compliant 256-bit strength SHA-512 Hash_DRBG. **{CR.1::CR-V2R4-DRBG-2}**

7.3.2.5 Security Management (FMT)

7.3.2.5.1 RACF User and Group Management

7.3.2.5.1.1 Definition of users and groups

z/OS users and groups are defined in RACF using user and group profiles.

To create a z/OS user, a user profile for the new user has to be created in RACF. Each user profile consists of a base segment and optional segments for the use of specific subsystems. In the evaluated configuration, the base segment and the OMVS segment for the specification of attributes for z/OS UNIX System Services contain the information required by the security functions defined in this Security Target. Other segments of the user profile may exist but the effects of any values in those segments do not influence the security policy defined in this Security Target. RACF also supports a special user profile segment, CSDATA, for which the security administrator can specify the format and content of the data fields using other profiles in the CFIELD class, as well as specifying access rules in the FIELD class to determine which users can view or update data in the segment **{SM.1::SM.1-R10-RACF-19}**.

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

- the SPECIAL role as a general system administrator **{SM.1::SM.1.1}**
- the UPDATE authority to the fields in a non-base segment of the profile he wants to modify through field-level access checking **{SM.1::SM.1.2}**
- the CLAUTH attribute for the USER class while one of the following is true when creating a user:

- the user is the owner of the default group specified in this command;
- the user has JOIN authority in the new user's default group;
- the user's default group is within the scope of a group in which the user has the group-SPECIAL attribute.

{SM.1::SM.1-V2R5-1}

- The user must have the SPECIAL attribute to give the new user the OPERATIONS, SPECIAL, AUDITOR, or ROAUDIT attribute. The user does not need the SPECIAL attribute to specify the OWNER operand. **{SM.1::SM.1-V2R4-1}**
- to modify the attribute of a user: the CLAUTH attribute for the user class **{SM.1::SM.1.4}**. Note that only the CLAUTH and NOCLAUTH attribute can be changed **{SM.1::SM.1.5}**.

RACF allows groups of users to be defined, making the management of users and user attributes and roles easier. To create a new group, a group profile must be defined in RACF. A group profile (as a user profile) consists of a base segment and (optional) other segments. As with the user profiles all group attributes related to the Security Policy as defined in this Security Target are contained in the base segment and the OMVS segment of the group profile. Each group defined in RACF must be owned by a RACF-defined user or by its superior group. Ownership of a group is assigned with the ADDGROUP command when a new group profile is created and can be changed with the ALTGROUP command used to change an existing group profile **{SM.1::SM.1.6}**.

RACF also supports a special group profile segment, CSDATA, for which the security administrator can specify the format and content of the data fields using other profiles in the CFIELD class, as well as specifying access rules in the FIELD class to determine which users can view or update data in the segment **{SM.1::SM.1-R10-RACF-20}**.

The owner of a group or a user connected to a group that has the group-SPECIAL role can:

- Define new users to RACF (provided he also has the CLAUTH attribute for the USER class) **{SM.1::SM.1.7}**.
- Connect and remove users from the group **{SM.1::SM.1.8}**.
- Delegate and change group authorities and set the default UACC for all new resources belonging to members of the group **{SM.1::SM.1.9}**.
- Modify, list, and delete the group profile **{SM.1::SM.1.10}**.
- Define, delete, and list the names of the subgroups under the group **{SM.1::SM.1.11}**.
- Specify the group terminal option **{SM.1::SM.1.12}**.

Users can be connected to a number of groups and have the group-related authorities of all the groups they are connected to **{SM.1::SM.1.13}**.

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

Management of z/OS user and group profiles occurs primarily via the RACF commands described later (ADDUSER, ALTUSER, DELUSER, LISTUSER , ADDGROUP, ALTGROUP, DELGROUP, LISTGRP). Administrators enter these commands while running in a TSO session.

7.3.2.5.1.2 User roles and attributes

User roles and attributes are extraordinary capabilities, restrictions, or environments that can be assigned to a user, either all of the time or when the user is connected to a specific group or groups. User attributes are stored and managed within the RACF database.

When a role or attribute is to apply only to a specific group or groups, it is specified at the group level and is called a group-related user attribute. For example, user attributes that are specified in an ADDUSER or ALTUSER command are stored in the user's profile and are in effect regardless of the group to which the user is connected **{SM.1::SM.1.14}**.

RACF maintains the roles and attributes specified in this section in fields in the user profile. The distinction between roles and attributes in this Security Target is artificial and reflects the definition in Chapter 5 for roles and user attributed. RACF does not make this distinction and the IBM guidance describes all of the following as user attributes.

Apart from the explicitly mentioned roles and attributes described below, users are assigned certain roles implicitly:

- Users implicitly are in the “user” role which allows them to change their own authentication data
- Users can be assigned the operator role by authorizing them to issue an operator command in the command's own profile.
- Ownership of objects entitles users to change the object's security attributes. Ownership for non-UNIX objects is identical to ownership of the profile protecting the object.

7.3.2.5.1.3 RACF Roles

7.3.2.5.1.3.1 SPECIAL and group-SPECIAL

A user who has the SPECIAL attribute at the system level can issue all RACF commands (but not all operands. There are AUDITOR-only operands related to the configuration of the audit function that only a user with the AUDITOR attribute is allowed to use) **{SM.1::SM.1.15}**. The SPECIAL attribute gives the user full control over all of the RACF profiles in the RACF database. The SPECIAL attribute can also be assigned at the group level. Such a user with the group-SPECIAL attribute has full control over all of the profiles within the scope of the group.

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

- add, delete, list and modify user, group, DATASET and other profiles **{SM.1::SM.1.16}**
- list and define RACF general options (except options related to auditing) **{SM.1::SM.1.17}**

A system administrator can delegate administrative activities to users such that they can administer profiles belonging to a defined group. He does this by assigning such users the group-SPECIAL attribute. Those users then have administrative capabilities within the group they were assigned the group SPECIAL attribute **{SM.1::SM.1.18}**. Users with the attribute group-SPECIAL can not use general RACF options of the SETROPTS command (except for the REFRESH GENERIC and LIST operands) **{SM.1::SM.1.19}**.

7.3.2.5.1.3.2 AUDITOR and group-AUDITOR

The AUDITOR attribute is given only to users who are responsible for auditing RACF security controls and functions. To provide a check and balance on RACF security measures, the AUDITOR attribute should be given to security or group administrators other than those who have the SPECIAL attribute. The AUDITOR attribute can also be assigned at the group level. Such a user with the group-AUDITOR attribute can control the audit configuration within the scope of the group where the attribute was assigned **{SM.1::SM.1.20}**.

A user with the AUDITOR attribute can define and modify the audit related options in user and the auditor related options for resource profiles **{SM.1::SM.1.21}**. This allows him to define which activities are to be recorded in the audit trail. He can also list the content of any profile and set the system wide audit related options using the SETROPTS command. Those options are:

- AUDIT or NOAUDIT (for each profile class) **{SM.1::SM.1.22}**
- CMDVIOL or NOCMDVIOL **{SM.1::SM.1.23}**
- LOGOPTIONS (for each profile class) **{SM.1::SM.1.24}**

- OPERAUDIT or NOOPERAUDIT **{SM.1::SM.1.25}**
- SAUDIT or NOSAUDIT **{SM.1::SM.1.26}**

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

A user with the group-Auditor attribute can define and modify the audit related options in user, and resource profiles associated with his group **{SM.1::SM.1.28}**. He can not modify or set audit related attributes that operate system-wide **{SM.1::SM.1.29}**. Note that a user with SPECIAL controls the activation/deactivation of the OMVS audit related classes (DIRACC, DIRSRCH, FSOBJ, FSSEC, IPOBJ, PROCACT and PROCESS)

7.3.2.5.1.3.3 ROAUDIT

A user who has the ROAUDIT attribute can monitor audit information and view RACF profiles to verify that appropriate audit controls have been defined, but can not alter the auditing controls.

7.3.2.5.1.4 z/OS UNIX superuser

A user operating with an effective UID of zero or a user that has been authorized to the BPX.SUPERUSER profile in the FACILITY class is defined to have the role of a z/OS UNIX superuser.

7.3.2.5.1.5 Pseudo user

A user defined with the NOPASSWORD, NOPHRASE, and NOOIDCARD parameter in his user profile is defined as having the role of a "pseudo-user". The TOE prohibits that a user with those attributes can log into the TOE. Those IDs can be used by SURROGAT-submitted batch jobs or by started procedures defined in the STARTED class or the started procedures table.

7.3.2.5.1.6 RACF Attributes

CLAUTH

If a user has the CLAUTH attribute in a class, RACF allows the user to define profiles in that class **{SM.1::SM.1.32}**.

Users receive the CLAUTH attribute on a class-by-class basis. The CLAUTH attribute can be assigned at the user or group level **{SM.1::SM.1.33}**.

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

- SPECIAL or NOSPECIAL **{SM.1::SM.1.34}**
- AUDITOR or NOAUDITOR **{SM.1::SM.1.35}**
- OPERATIONS or NOOPERATIONS **{SM.1::SM.1.36}**
- ROAUDIT or NOROAUDIT **{SM.1::SM.1-V2R2-RACF-70}**

7.3.2.5.1.7 REVOKE

A user can be prevented from entering the system by assigning the REVOKE attribute **{SM.1::SM.1.37}**. This attribute is useful when a user needs to be prevented from entering the system, but cannot be deleted using the DELUSER command because the user still owns RACF resource profiles. It is also useful when a user must be temporarily prevented from using the system for some reason.

User accounts can be revoked automatically after a period of inactivity **{SM.1::SM.1.38}**. This applies also to accounts that have never been active **{SM.1::SM.1.39}**.

7.3.2.5.1.8 User Revocation

User revocation can take two forms in the TOE:

- Revocation of the RACF user ID associated with a user: As all user authentication occurs via RACF, and all users have a RACF identity, the administrator can revoke a user by using the ALTUSER command with the REVOKE operand **{SM.1::SM.1-R8-REV-1}**. Note that this will not cover immediate revocation, but it will prevent the user from entering the system in the future.
- Revocation of a user's digital certificate: For certificates registered in RACF via the RACDCERT command, the administrator can delete the certificate using RACDCERT **{SM.1::SM.1-R8-REV-2}**. This will prevent the system from recognizing that certificate in the future and associating it with the user's RACF identity.

For immediate revocation of a user in extreme situations a simple ALTUSER or certificate revocation may not suffice. In that case the administrator may determine which applications the user has access to (e.g., TSO/E, z/OS UNIX System Services). The administrator can then issue appropriate system or application commands to determine if the user is active in the system, and if so issue the appropriate system or application commands to terminate the user's sessions.

For example, for a TSO/E user the administrator could issue the CANCEL U=user-ID command. For a batch job the administer could issue CANCEL jobname.

As a final resort the administrator could stop related servers if the administrator is not sure how to locate the user's sessions on the system, as well as stopping all UNIX processing, TSO/E processing, and batch processing.

7.3.2.5.2 RACF configuration and management

7.3.2.5.2.1 Configuring RACF with the SETROPTS command

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

- Specify logging of certain RACF commands and events (requires AUDITOR). **{SM.3::SM.3.3}**
- Enable or disable list-of-groups access checking (requires SPECIAL). **{SM.3::SM.3.4}**
- Enable generic profile checking for all active classes (requires SPECIAL). **{SM.3::SM.3.6}**
- Establish password syntax rules (requires SPECIAL). **{SM.3::SM.3.7}**
- Activate password processing for checking previous passwords, limit invalid password attempts, and warn of password expiration (requires SPECIAL). **{SM.3::SM.3.8}**
- Control global access checking for selected individual resources or generic names with selected generalized access rules (requires SPECIAL). **{SM.3::SM.3.9}**
- Activate auditing of access attempts to RACF-protected resources based on installation-defined security levels (requires AUDITOR). **{SM.3::SM.3.13}**
- Activate enhanced generic naming (requires SPECIAL). **{SM.3::SM.3.14}**
- Activate protection for data sets with single-level names (requires SPECIAL). **{SM.3::SM.3.16}**
- Control logging of real data set names (requires AUDITOR). **{SM.3::SM.3.17}**
- Enable the erasure of scratched DASD data sets (requires SPECIAL). **{SM.3::SM.3.21}**
- Activate program control (requires SPECIAL). **{SM.3::SM.3.22}**

Some administration activities can be delegated to user with other roles. See the definition of those roles for the administrative options that can be set or defined by those roles.

To operate in correspondence with the requirements in this Security Target, the system administrator needs to configure RACF (using the SETROPTS command) with the following options: CATDSNS(FAILURES), NOCOMPATMODE, ERASE(ALL), GENERIC(*), PROTECTALL(FAILURES), CLASSACT (TEMPDSN), JES(BATCHALLRACF). Additional parameter for the PASSWORD operand need to be set to define the password policy. See RACF Passwords and Password Phrases for more information.

7.3.2.5.3 RACF Certificate and Key Management

RACF provides the RACDCERT command which can be used to

- generate public/private key pairs and certificates (DIGTCERT class) **{SM.1::SM.1-R8-RACF-RACDCERT-2}**
- export a certificate or certificate packages to a data set, optionally with the private key **{SM.1::SM.1-R8-RACF-RACDCERT-3}**
- install certificates into the RACF database and register them as belonging to a user or to a certifying authority **{SM.1::SM.1-R8-RACF-RACDCERT-4}**. The __certificate() and initACEE() services can also register/deregister certificates **{SM.1::SM.1-R8-RACF-RACDCERT-5}**, and administrators an allow users to register their own certificates by granting them READ access to FACILITY resource IRR.DIGTCERT.ADD **{SM.1::SM.1-R8-RACF-RACDCERT-6}**.
- delete or list certificates in the RACF database **{SM.1::SM.1-R8-RACF-RACDCERT-7}**
- maintain (create, list, delete) key rings containing certificates (DIGTRING class) **{SM.1::SM.1-R8-RACF-RACDCERT-8}**
- add certificates to or delete them from key rings **{SM.1::SM.1-R8-RACF-RACDCERT-9}**
- create mapping rules (certificate name filters) that can map client certificates that are not installed/registered in the database to specified user IDs based on subject or issuer information (DIGTNMAP class) **{SM.1::SM.1-R8-RACF-RACDCERT-10}**. This can allow a many-to-one mapping for applications that do not need to have each user run under his own ID. In this case, accountability can be maintained for auditing purposes by having the application provide the subject's distinguished name via the X500Name parameter when creating the security environment (ACEE) for the user **{SM.1::SM.1-R8-RACF-RACDCERT-11}**. The mapping process can also make use of mapping criteria specified by the DIGTCRIT class when it is necessary to map a client certificate into different IDs depending on characteristics of the user's session (such as the application name, or system name where the application is running) **{SM.1::SM.1-R8-RACF-RACDCERT-12}**.
- create and manage the contents of PKCS#11 cryptographic tokens contained in the ICSF TKDS **{SM.1::SM-1.R9-RACF-RACDCERT-13}**.
- use ICSF to create and manage private cryptographic keys **{SM.1::SM-1-V2R5-RACF-RACDCERT-1}**.

7.3.2.5.4 Audit configuration and management

Within the system configuration it needs to be decided, which SMF records shall be generated by z/OS. Three record types (type 80, 81, and 83) are dedicated to RACF and are the most important ones for security. Which events are actually recorded with those records can be configured by a user with the AUDITOR attribute in his RACF user profile **{AU.3::AU.3.1}**. In addition record type 30 is generated for a number of security related events.

Because a set of mandatory events is always audited, not all audit records (such as unauthorized attempts to access the system or changes to the status of the RACF database) can be configured.

In addition, resource profiles can define which events related to this resource are audited **{AU.3::AU.3.2}**. The owner of a resource profile as well as a user in the AUDITOR role are able to change the entries related to auditing within the resource profile **{AU.3::AU.3.3}**.

The system can be configured to send certain audit messages to the security console to immediately alert operators of detected policy violations **{AU.3::AU.3.4}**.

Users that have the ROAUDIT attribute can read the audit trail and obtain audit related information form RACF profiles, but can not manage audit policy related aspects **{AU.3::AU-V2R2-1}**.

7.3.2.6 Self Protection

7.3.2.6.1 Time Management

The TOE supports using the following reliable time sources:

Local Clock

An operator with appropriate authorization can use the SET command to set the local time (using the CLOCK parameter) and the date (using the DATE parameter). **{SP.1::TIME-V2R4-1}**

It should be noted that this method of managing the time is only available if the TOE is not using the STP protocol to use a synchronized network time source.

Synchronized Network Time

Synchronized network time is provided by the TOE's System z platform, which uses synchronized platform time sources or by a network time server using the STP protocol to provide correct time stamps to the TOE. Using the CLOCKxx PARMLIB member, the TOE is instructed to use the specified time source to synchronize its clock. **{SP.1::TIME-V2R4-2}**

7.3.2.6.2 Automatic Logout of Sessions

The TOE provides, for both TSO/E and USS a means of automatically logging out inactive sessions. For TSO/E this is achieved by setting the timeout value JWT parameter in the SMFPRMxx member in PARMLIB. Also the TIME parameter (in minutes) in the user's login procedure needs to be set accordingly. For USS this is achieved by setting PWT(SMF) in the BPXPRMxx member in PARMLIB as well as setting the environment variable _BPXK_TIMEOUT to the value SMF **{SP.1::FTA-V2R5-1}**.

7.3.2.6.3 Address Space Layout Randomization

With the ASLR feature is enabled, the start of the 24 and 31 bit low private areas will be increased by a random number of 4k units ranging from 0-63 and 0-255 pages respectively. Similarly, the start 64 bit private will be increased by a random multiple of megabytes. By changing the start of these storage ranges, subsequent storage allocation requests will not be satisfied by the addresses that they normally would be, making it more difficult for an attacker to guess the starting address of an executable or some other storage area.

The system is using 6 bits of random data for 24bit address spaces and 8 bits of random data for all other address spaces to randomize the layout of the address space as described **{SP.1::ASLR-V2R4-1}**.

7.3.2.6.4 Stack Buffer Overflow Protection

The TOE protects TOE binaries from stack-based buffer overflow attacks by using code that is inserted into compiled modules when the STACKPROTECT C/C++ compiler option is being used. That inserted code provides protection against malicious code or programming errors that overwrite or corrupt the stack.

The following list provides the binaries shipped with the TOE that support stack-based buffer overflow protection:

Binary	Binary	Binary
/bin/IBM/IEWULD	/bin/IBM/FDBXDBD6	/bin/IBM/FDBXDBXX

Binary	Binary	Binary
/bin/IBM/FDBXDBX3	/bin/IBM/FDBXDBX6	/bin/IBM/FOMFTSO
/bin/IBM/FOMFUISH	/bin/IBM/FSUMSBSN	/bin/IBM/FSUMSCAT
/bin/IBM/FSUMVCHA	/bin/IBM/FSUMSCHG	/bin/IBM/FSUMSCHL
/bin/IBM/FSUMSCHM	/bin/IBM/FSUMSCHT	/bin/IBM/FSUMSCHW
/bin/IBM/FSUMSCKS	/bin/IBM/FSUMSCMM	/bin/IBM/FSUMSCMP
/bin/IBM/FSUMSCOS	/bin/IBM/FSUMSDU	/bin/IBM/FSUMSECH
/bin/IBM/FSUMSFLS	/bin/IBM/FSUMSFND	/bin/IBM/FSUMSGRP
/bin/IBM/FSUMSICV	/bin/IBM/FSUMSLS	/bin/IBM/FSUMSMKD
/bin/IBM/FSUMSNWG	/bin/IBM/FSUMSPCK	/bin/IBM/FSUMSPCT
/bin/IBM/FSUMSPRN	/bin/IBM/FSUMSPWD	/bin/IBM/FSUMSRM
/bin/IBM/FSUMSSD0	/bin/IBM/FSUMSSHB	/bin/IBM/FSUMSTIM
/bin/IBM/FSUMSTRU	/bin/IBM/FSUMSTST	/bin/IBM/FSUMSUNE
/bin/IBM/FSUMSUNP	/bin/IBM/FSUMSWC	/bin/IBM/FSUMSYMN
/bin/IBM/FSUMSAR	/bin/IBM/FSUMSAT	/bin/IBM/FSUMSCNC
/bin/IBM/FSUMSCOL	/bin/IBM/FSUMSCRT	/bin/IBM/FSUMSDAT
/bin/IBM/FSUMSDMS	/bin/IBM/FSUMSDPC	/bin/IBM/FSUMSENV
/bin/IBM/FSUMSEXD	/bin/IBM/FSUMSEXP	/bin/IBM/FSUMSGEN
/bin/IBM/FSUMSID	/bin/IBM/FSUMSIDN	/bin/IBM/FSUMSJON
/bin/IBM/FSUMSLP	/bin/IBM/FSUMSLPS	/bin/IBM/FSUMSMKC
/bin/IBM/FSUMSMKF	/bin/IBM/FSUMSOD	/bin/IBM/FSUMSPEN
/bin/IBM/FSUMSPR	/bin/IBM/FSUMSPSW	/bin/IBM/FSUMSRMD
/bin/IBM/FSUMSSFL	/bin/IBM/FSUMSSLP	/bin/IBM/FSUMSSPL
/bin/IBM/FSUMSSRT	/bin/IBM/FSUMSSTP	/bin/IBM/FSUMSSTS
/bin/IBM/FSUMSSU	/bin/IBM/FSUMSUNX	/bin/IBM/FSUMSASA
/bin/IBM/FSUMSBC	/bin/IBM/FSUMSCSP	/bin/IBM/FSUMSCUT
/bin/IBM/FSUMSDD	/bin/IBM/FSUMSDRN	/bin/IBM/FSUMSFLD
/bin/IBM/FSUMSFU	/bin/IBM/FSUMSGFL	/bin/IBM/FSUMSGTC
/bin/IBM/FSUMSHED	/bin/IBM/FSUMSLGN	/bin/IBM/FSUMSLIN
/bin/IBM/FSUMSMEG	/bin/IBM/FSUMSNHP	/bin/IBM/FSUMSNIC
/bin/IBM/FSUMSNL	/bin/IBM/FSUMLOGN	/bin/IBM/FSUMSPST
/bin/IBM/FSUMSPTH	/bin/IBM/FSUMSRNC	/bin/IBM/FSUMSSCR
/bin/IBM/FSUMSSP0	/bin/IBM/FSUMSSVR	/bin/IBM/FSUMSTBS
/bin/IBM/FSUMSTCH	/bin/IBM/FSUMSTEE	/bin/IBM/FSUMSTLK
/bin/IBM/FSUMSTOT	/bin/IBM/FSUMSTPT	/bin/IBM/FSUMSTR
/bin/IBM/FSUMSTTY	/bin/IBM/FSUMSUDC	/bin/IBM/FSUMSUNC
/bin/IBM/FSUMSUNQ	/bin/IBM/FSUMSUPT	/bin/IBM/FSUMSWHO
/bin/IBM/FSUMSWLL	/bin/IBM/FSUMSWMI	/bin/IBM/FSUMSWRT
/bin/IBM/FSUMSXRG	/bin/IBM/FOMIPCRM	/bin/IBM/FOMIPCS
/bin/IBM/FSUMSAWK	/bin/IBM/FSUMSCAL	/bin/IBM/FSUMSCLN
/bin/IBM/FSUMSCTG	/bin/IBM/FSUMSCU	/bin/IBM/FSUMSDFF
/bin/IBM/FSUMSED0	/bin/IBM/FSUMSFIL	/bin/IBM/FSUMSLCL
/bin/IBM/FSUMSLEX	/bin/IBM/FSUMSLGG	/bin/IBM/FSUMSMAK
/bin/IBM/FSUMSML1	/bin/IBM/FSUMSMNP	/bin/IBM/FSUMSMOR

Binary	Binary	Binary
/bin/IBM/FSUMSPTC	/bin/IBM/FSUMSRML	/bin/IBM/FSUMSTAL
/bin/IBM/FSUMSUCP	/bin/IBM/FSUMSUNM	/bin/IBM/FSUMSUST
/bin/IBM/FSUMSUUE	/bin/IBM/FSUMSUUX	/bin/IBM/FSUMSVI
/bin/IBM/FSUMSPS	/bin/IBM/FSUMSCPI	/bin/IBM/FSUMSPAX
/bin/IBM/FSUMSTAR	/bin/IBM/FSUMSSTT	/bin/IBM/FSUMSNM
/bin/IBM/FSUMSCS	/bin/IBM/FSUMSCP	/bin/IBM/FSUMSLK
/bin/IBM/FSUMSLN	/bin/IBM/FSUMSMV	/bin/IBM/FSUMSSH
/bin/IBM/FSUMSPG	/bin/IBM/FSUMSDF	/bin/IBM/FOTSXADD
/bin/IBM/FOTSXAGT	/bin/IBM/FOTSXFTP	/bin/IBM/FOTSXKGN
/bin/IBM/FOTSXKSC	/bin/IBM/FOTSXPYC	/bin/IBM/FOTSXSSH
/bin/IBM/FOTSXSCP	/bin/IBM/CDAHE003	/bin/IBM/CDAHEAS0
/bin/ld	/bin/dbxd64	/bin/dbx
/bin/dbx31	/bin/dbx64	/bin/tso
/bin/fomfuish	/bin/basename	/bin/cat
/bin/chaudit	/bin/chgrp	/bin/chlabel
/bin/chmod	/bin/chtag	/bin/chown
/bin/cksum	/bin/sum	/bin/comm
/bin/cmp	/bin/compress	/bin/du
/bin/echo	/bin/false	/bin/find
/bin/grep	/bin/fgrep	/bin/egrep
/bin/iconv	/bin/ls	/bin/mkdir
/bin/newgrp	/bin/pack	/bin/pcat
/bin/printf	/bin/pwd	/bin/rm
/bin/unlink	/bin/sed	/bin/getopts
/bin/command	/bin/kill	/bin/read
/bin/wait	/bin/cd	/bin/umask
/bin/alias	/bin/bg	/bin/fc
/bin/fg	/bin/jobs	/bin/unalias
/bin/ulimit	/bin/hash	/bin/type
/bin/time	/bin/true	/bin/test
/bin/uncompress	/bin/zcat	/bin/unpack
/bin/wc	/bin/yacc	/bin/ar
/bin/at	/bin/batch	/bin/cancel
/bin/col	/bin/crontab	/bin/date
/bin/dspmsg	/bin/dspcat	/bin/env
/bin/expand	/bin/expr	/bin/gencat
/bin/id	/bin/iden	/bin/join
/bin/lp	/bin/lpstat	/bin/mkcatdefs
/bin/mkfifo	/bin/od	/bin/printenv
/bin/pr	/bin/passwd	/bin/rmdir
/bin/setfacl	/bin/sleep	/bin/split
/bin/sort	/bin/strip	/bin/strings
/bin/su	/bin/unexpand	/bin/asa

Binary	Binary	Binary
/bin/bc	/bin/csplit	/bin/cut
/bin/dd	/bin/dirname	/bin/fold
/bin/fuser	/bin/getfacl	/bin/getconf
/bin/head	/bin/logname	/bin/line
/bin/mesg	/bin/nohup	/bin/nice
/bin/nl	/bin/login	/bin/paste
/bin/pathchk	/bin/renice	/bin/script
/bin/spell	/bin/sysvar	/bin/tabs
/bin/touch	/bin/tee	/bin/talk
/bin/tsort	/bin/tput	/bin/clear
/bin/tr	/bin/tty	/bin/uudecode
/bin/uuencode	/bin/uniq	/bin/uptime
/bin/who	/bin/wall	/bin/whoami
/bin/write	/bin/xargs	/bin/ipcrm
/bin/ipcs	/bin/awk	/bin/cal
/bin/calendar	/bin/ctags	/bin/cu
/bin/diff	/bin/dircmp	/bin/ed
/bin/file	/bin/locale	/bin/lex
/bin/logger	/bin/make	/bin/mailx
/bin/mail	/bin/man	/bin/more
/bin/patch	/bin/rmail	/bin/tail
/bin/uucp	/bin/uname	/bin/uustat
/bin/uuname	/bin/uux	/bin/vi
/bin/ex	/bin/ps	/bin/cpio
/bin/pax	/bin/tar	/bin/stty
/bin/nm	/bin/md5	/bin/rmd160
/bin/sha1	/bin/sha224	/bin/sha256
/bin/sha384	/bin/sha512	/bin/cp
/bin/link	/bin/ln	/bin/mv
/bin/sh	/bin/pg	/bin/df
/bin/ssh-add	/bin/ssh-agent	/bin/sftp
/bin/ssh-keygen	/bin/ssh-keyscan	/bin/ssh-proxyc
/bin/ssh	/bin/scp	/bin/dbgld
/bin/as	/usr/sbin/IBM/FOMRLOG2	/usr/sbin/IBM/FSUMSCHR
/usr/sbin/IBM/FSUMSMKN	/usr/sbin/IBM/FSUMSUCD	/usr/sbin/rlogind2
/usr/sbin/chroot	/usr/sbin/mknod	/usr/sbin/uucpd
/usr/lib/hwicmuss.dll	/usr/lib/iewbndd.so	/usr/lib/iewbndd6.so
/usr/lib/iewbnddx.so	/usr/lib/dbx31.dll	/usr/lib/dbx64.dll
/usr/lib/libigwznsqd64.so	/usr/lib/libigwznsqd31.so	/usr/lpp/zosmf/installableApps/izurestconsoles.ear/liblzuConsoleJni64.so
/usr/lpp/zosmf/lib/liblzuCommandJni.so	/usr/lpp/zosmf/lib/liblzuJesVsam64.so	/usr/lpp/zosmf/lib/liblzuJesStatus64.so
/usr/lpp/zosmf/lib/liblzugJni64.so	/usr/lpp/zosmf/lib/liblzuTsoSrvJni64.so	/usr/lpp/zosmf/lib/liblzuRestJobsJni64.so
/usr/lpp/zosmf/lib/liblzuMessageQueueJni64.so	/usr/lpp/zosmf/lib/liblzuCeaTsoJni64.so	/usr/lpp/zosmf/lib/liblzuSgJni64.so
/usr/lpp/zosmf/lib/liblzuspJni64.so	/usr/lpp/zosmf/lib/liblzudDmJni.so	/usr/lpp/zosmf/lib/liblzuComplianceJni64.so

Binary	Binary	Binary
/usr/lpp/IBM/aie/zdnn/lib/libzdnn.so	/usr/lpp/IBM/aie/zaio/lib/libzaio.so	/usr/lpp/IBM/aie/zade/lib/libzade.so
/usr/lpp/IBM/zdg/REST/smf/v1/SmfBridgeImpl.so	/usr/lpp/pkcs11/lib/csnpcapi.so	/usr/lpp/pkcs11/lib/csnpca64.so
/usr/lpp/pkcs11/lib/csnpca3x.so	/usr/lpp/dfsms/bin/libarcudll.so	/usr/lpp/tcpip/lib/ libcmpiOSBase_EthernetPortProvider.so
/usr/lpp/tcpip/lib/ libcmpiOSBase_IPProtocolEndpointProvider.so	/usr/lpp/tcpip/lib/ libcmpiOSBase_NetworkPortImplementsIPEndpointProvider.so	/usr/lpp/tcpip/lib/ libcmpiOSBase_CSNetworkPortProvider.so
/usr/lpp/tcpip/lib/libEZAFTP.so	/usr/lpp/tcpip/lib/libEZBCPP.so	/usr/lpp/tcpip/lib/libEZBCPP64.so
/usr/lpp/tcpip/lib/libEZBTrustedPartner.so	/usr/lpp/tcpip/lib/libEZBTrustedPartner64.so	/usr/lpp/tcpip/lib/papi.dll
/usr/lpp/tcpip/lib/rapi.dll	/usr/lpp/tcpip/X11R6/lib/X11.dll	/usr/lpp/tcpip/X11R6/lib/ICE.dll
/usr/lpp/tcpip/X11R6/lib/PEX5.dll	/usr/lpp/tcpip/X11R6/lib/SM.dll	/usr/lpp/tcpip/X11R6/lib/Xaw.dll
/usr/lpp/tcpip/X11R6/lib/Xm.dll	/usr/lpp/tcpip/X11R6/lib/Mrm.dll	/usr/lpp/tcpip/X11R6/lib/Uil.dll
/usr/lpp/ihsa_zos/.31bit/lib/apachecore.dll	/usr/lpp/ihsa_zos/.31bit/lib/libapr-1.so	/usr/lpp/ihsa_zos/.31bit/lib/libaprutil-1.so
/usr/lpp/ihsa_zos/.31bit/lib/liblua.so	/usr/lpp/ihsa_zos/.31bit/lib/libpcre.so	/usr/lpp/ihsa_zos/.31bit/lib/libpcreposix.so
/usr/lpp/ihsa_zos/.31bit/modules/debug/ mod_mpmstats.so	/usr/lpp/ihsa_zos/.31bit/modules/debug/ mod_net_trace.so	/usr/lpp/ihsa_zos/.31bit/modules/debug/ mod_whatkilledus.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_a2e.so	/usr/lpp/ihsa_zos/.31bit/modules/ mod_access_compat.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_actions.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_alias.so	/usr/lpp/ihsa_zos/.31bit/modules/ mod_allowmethods.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_asis.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_auth_basic.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_authn_anon.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_authn_cert.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_authn_core.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_authn_dbm.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_authn_file.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_authnz_ldap.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_authnz_saf.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_authz_core.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_authz_dbm.so	/usr/lpp/ihsa_zos/.31bit/modules/ mod_authz_groupfile.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_authz_host.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_authz_user.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_autoindex.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_cache.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_cache_disk.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_cgi.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_charset_lite.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_deflate.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_dir.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_env.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_expires.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_ext_filter.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_fastcgi.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_filter.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_headers.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_ibm_ssl.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_imagemap.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_include.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_info.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_ldap.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_log_config.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_logio.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_lua.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_macro.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_mem_cache.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_mime.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_mvsds.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_negotiation.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_proxy.so	/usr/lpp/ihsa_zos/.31bit/modules/ mod_proxy_connect.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_proxy_fcgi.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_proxy_ftp.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_proxy_http.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_remoteip.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_reqtimeout.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_request.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_rewrite.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_setenvif.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_smf.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_speling.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_status.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_substitute.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_unique_id.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_unixd.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_userdir.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_usertrack.so
/usr/lpp/ihsa_zos/.31bit/modules/mod_vhost_alias.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_wlm.so	/usr/lpp/ihsa_zos/.31bit/modules/mod_zos_cmds.so
/usr/lpp/ihsa_zos/lib/libapr-1.so	/usr/lpp/ihsa_zos/lib/libaprutil-1.so	/usr/lpp/ihsa_zos/lib/liblua.so
/usr/lpp/ihsa_zos/lib/libpcre.so	/usr/lpp/ihsa_zos/lib/libpcreposix.so	/usr/lpp/ihsa_zos/lib/apachecore.dll
/usr/lpp/ihsa_zos/modules/debug/mod_backtrace.so	/usr/lpp/ihsa_zos/modules/debug/mod_mpmstats.so	/usr/lpp/ihsa_zos/modules/debug/mod_net_trace.so
/usr/lpp/ihsa_zos/modules/debug/mod_whatkilledus.so	/usr/lpp/ihsa_zos/modules/mod_a2e.so	/usr/lpp/ihsa_zos/modules/mod_access_compat.so
/usr/lpp/ihsa_zos/modules/mod_actions.so	/usr/lpp/ihsa_zos/modules/mod_alias.so	/usr/lpp/ihsa_zos/modules/mod_allowmethods.so
/usr/lpp/ihsa_zos/modules/mod_asis.so	/usr/lpp/ihsa_zos/modules/mod_auth_basic.so	/usr/lpp/ihsa_zos/modules/mod_authn_anon.so

Binary	Binary	Binary
/usr/lpp/ihsa_zos/modules/mod_authn_cert.so	/usr/lpp/ihsa_zos/modules/mod_authn_core.so	/usr/lpp/ihsa_zos/modules/mod_authn_dbm.so
/usr/lpp/ihsa_zos/modules/mod_authn_file.so	/usr/lpp/ihsa_zos/modules/mod_authnz_ldap.so	/usr/lpp/ihsa_zos/modules/mod_authnz_saf.so
/usr/lpp/ihsa_zos/modules/mod_authz_core.so	/usr/lpp/ihsa_zos/modules/mod_authz_dbm.so	/usr/lpp/ihsa_zos/modules/mod_authz_groupfile.so
/usr/lpp/ihsa_zos/modules/mod_authz_host.so	/usr/lpp/ihsa_zos/modules/mod_authz_user.so	/usr/lpp/ihsa_zos/modules/mod_autoindex.so
/usr/lpp/ihsa_zos/modules/mod_cache.so	/usr/lpp/ihsa_zos/modules/mod_cache_disk.so	/usr/lpp/ihsa_zos/modules/mod_cgi.so
/usr/lpp/ihsa_zos/modules/mod_charset_lite.so	/usr/lpp/ihsa_zos/modules/mod_deflate.so	/usr/lpp/ihsa_zos/modules/mod_dir.so
/usr/lpp/ihsa_zos/modules/mod_env.so	/usr/lpp/ihsa_zos/modules/mod_expires.so	/usr/lpp/ihsa_zos/modules/mod_ext_filter.so
/usr/lpp/ihsa_zos/modules/mod_fastcgi.so	/usr/lpp/ihsa_zos/modules/mod_filter.so	/usr/lpp/ihsa_zos/modules/mod_headers.so
/usr/lpp/ihsa_zos/modules/mod_ibm_ssl.so	/usr/lpp/ihsa_zos/modules/mod_imagemap.so	/usr/lpp/ihsa_zos/modules/mod_include.so
/usr/lpp/ihsa_zos/modules/mod_info.so	/usr/lpp/ihsa_zos/modules/mod_ldap.so	/usr/lpp/ihsa_zos/modules/mod_log_config.so
/usr/lpp/ihsa_zos/modules/mod_logio.so	/usr/lpp/ihsa_zos/modules/mod_lua.so	/usr/lpp/ihsa_zos/modules/mod_macro.so
/usr/lpp/ihsa_zos/modules/mod_mem_cache.so	/usr/lpp/ihsa_zos/modules/mod_mime.so	/usr/lpp/ihsa_zos/modules/mod_mvsds.so
/usr/lpp/ihsa_zos/modules/mod_negotiation.so	/usr/lpp/ihsa_zos/modules/mod_proxy_connect.so	/usr/lpp/ihsa_zos/modules/mod_proxy_fcgi.so
/usr/lpp/ihsa_zos/modules/mod_proxy_ftp.so	/usr/lpp/ihsa_zos/modules/mod_remoteip.so	/usr/lpp/ihsa_zos/modules/mod_reqtimeout.so
/usr/lpp/ihsa_zos/modules/mod_request.so	/usr/lpp/ihsa_zos/modules/mod_setenvif.so	/usr/lpp/ihsa_zos/modules/mod_smf.so
/usr/lpp/ihsa_zos/modules/mod_speling.so	/usr/lpp/ihsa_zos/modules/mod_status.so	/usr/lpp/ihsa_zos/modules/mod_substitute.so
/usr/lpp/ihsa_zos/modules/mod_unique_id.so	/usr/lpp/ihsa_zos/modules/mod_unixd.so	/usr/lpp/ihsa_zos/modules/mod_userdir.so
/usr/lpp/ihsa_zos/modules/mod_usertrack.so	/usr/lpp/ihsa_zos/modules/mod_vhost_alias.so	/usr/lpp/ihsa_zos/modules/mod_wlm.so
/usr/lpp/ihsa_zos/modules/mod_zos_cmds.so	/usr/lpp/ihsa_zos/modules/mod_proxy.so	/usr/lpp/ihsa_zos/modules/mod_proxy_http.so
/usr/lpp/ihsa_zos/modules/mod_rewrite.so	/usr/lpp/pkiserv/lib/ObjStore.dll	/usr/lpp/pkiserv/lib/Serial.dll
/usr/lpp/pkiserv/lib/asn1.dll	/usr/lpp/pkiserv/lib/confobj.dll	/usr/lpp/pkiserv/lib/crypto.dll
/usr/lpp/pkiserv/lib/kspkix.dll	/usr/lpp/pkiserv/lib/misc.dll	/usr/lpp/pkiserv/lib/mvsutils.dll
/usr/lpp/pkiserv/lib/ossrv.dll	/usr/lpp/pkiserv/lib/pkiapi.dll	/usr/lpp/pkiserv/lib/pkiconf.dll
/usr/lpp/pkiserv/lib/pkimsg.dll	/usr/lpp/pkiserv/lib/policy.dll	/usr/lpp/pkiserv/lib/safring.dll
/usr/lpp/pkiserv/lib/pkilc.dll	/usr/lpp/pkiserv/lib/librpkisJNI.so	/usr/lpp/pkiserv/lib/cmpasn1.dll
/usr/lpp/pkiserv/lib64/librpkisJNI64.so	/usr/lpp/Printsrv/lib/libaopjnixp.so	/usr/lpp/Printsrv/lib/aopapi.dll
/usr/lpp/Printsrv/lib/aopapi2.dll	/usr/lpp/Printsrv/lib/aoprform.dll	/usr/lpp/Printsrv/lib/aopeapi.dll
/usr/lpp/Printsrv/lib/aoprxf.so	/usr/lpp/Printsrv/lib/aop.so	/usr/lpp/Printsrv/lib/aopdb.so
/usr/lpp/Printsrv/lib/aopipc2.so	/usr/lpp/wbem/lib/libCIMxmlIndicationHandler.so	/usr/lpp/wbem/lib/libIBMJobCompletedHandler.so
/usr/lpp/wbem/lib/libcmpiCppImpl.so	/usr/lpp/wbem/lib/libCMPIProviderManager.so	/usr/lpp/wbem/lib/libCMPIRProxyProvider.so
/usr/lpp/wbem/lib/libCMPIRTCPComm.so	/usr/lpp/wbem/lib/libpegclient.so	/usr/lpp/wbem/lib/libpegcliutils.so
/usr/lpp/wbem/lib/libpegcommon.so	/usr/lpp/wbem/lib/libpegcompiler.so	/usr/lpp/wbem/lib/libpegconfig.so
/usr/lpp/wbem/lib/libpegexportclient.so	/usr/lpp/wbem/lib/libpeggetoopt.so	/usr/lpp/wbem/lib/libpeglistener.so
/usr/lpp/wbem/lib/libpegmanagedclient.so	/usr/lpp/wbem/lib/libpegprovider.so	/usr/lpp/wbem/lib/libpegslp.so
/usr/lpp/wbem/lib/libpegslp_client.so	/usr/lpp/wbem/lib/libSLPProvider.so	/usr/lpp/wbem/lib/libcfzsys.so
/usr/lpp/wbem/lib/libpegclient64.so	/usr/lpp/wbem/lib/libpegcommon64.so	/usr/lpp/wbem/lib/libcfzsys64.so
/usr/lpp/cpo/lib/libcpoprovider.so	/usr/lpp/cpo/lib/libcposocket.so	/usr/lpp/cpo/lib/libcpoarm.so
/usr/lpp/cpo/lib/libcpoconsole.so	/usr/lpp/cpo/lib/libcpostream.so	/usr/lpp/cpo/lib/libcpoii.so
/usr/lpp/sdsf/java/lib_64/libisfbuffer.so	/usr/lpp/sdsf/java/lib_64/libisfjcall.so	/usr/lpp/gpm/bin/libgpmccli.so

Table 11: List of Executables with Stack-buffer Overflow Protection

Any other load module or executable the TOE ships does **not** have support for stack based buffer overflow protection enabled as they are not compiled using the C/C++ compiler framework that supports named protection or the protection mechanisms could not be applied for technical reasons.

7.3.2.6.5 Trusted Update Process

Updates are queried for, obtained, applied using the SMP/E subsystem. SMP/E is also used to verify the signatures of the updates as well as the signatures of the responses of the update server. The signature metadata is provided by IBM's updates servers by means of an additional file that is supplied as part of the update or the query for updates. This file, named GIMPAFM2.XML contains the SHA-256 hash sums for the individual files of the packages comprising the updates or the query. GIMPAFM2.XML contains the SHA256withRSA signature, which is protecting the contents metadata from undetected modification. This signature is verified upon acceptance of the package by SMP/E, where the GIMUNZIP program is used for verification **{SP.1::TUD-V2R5-1}**.

The TOE does not differentiate between updates to applications and updates to the operating system. The same mechanisms are used for both of them.

Signature, using the RSASSA-PSS scheme, verification is done using the cryptographic hardware: IBMJCECCA is the backing methods for the cryptographic operations. The IBMJCECCA provider extends the JCE capability with the use of hardware cryptography via the IBM Common Cryptographic Architecture (CCA) APIs (which drive ICSF and the cryptographic hardware CEX8) **{SP.1::TUD-V2R5-2}**.

7.3.2.7 Communication Security

7.3.2.7.1 Methods of remote Administration

An Administrator can access the TOE using the TLS protected TN3270 (see section System SSL) as well as the SSH protocol (see section OpenSSH) to carry out administrative actions.

7.3.2.7.2 Communications Server

z/OS provides basic networking functions with the Communications Server component. This subsystem provides support for network communication using the IBM SNA protocols as well as the TCP/IP protocol suite. APIs for both protocol stacks are provided. For IP, both IPv4 and IPv6 are supported.

The Communications Server uses RACF to protect access of users to the following resources:

- the TCP/IP stack in general **{CS.1::CS.1.1}**
- TCP and UDP ports **{CS.1::CS.1.2}**
- IP addresses **{CS.1::CS.1.3}**

z/OS provides the following security functions as part of the Communications Server:

- TLS layer to set up a trusted channel to another trusted IT product, in a way transparent to the application (called Application Transparent TLS, or AT-TLS). The selectable algorithms can be limited by configuring a subset of allowable algorithms at the server. The TLS protocol can be used to set up a trusted channel to another system through a potentially insecure network. TLS protects the data against disclosure and attacks related to integrity like undetectable modifications or replay. Servers can support encryption using AES with 256-bit key length. Application Transparent Transport Layer Security (AT-TLS) supports the use of all cipher suites supported by System SSL **{CS.1::CS.1.4-V2R4}**. The TN3270 or e.g. the FTP protocols can be enabled to use AT-TLS and can subsequently be tunneled through TLS to establish a trusted channel to another trusted IT product that also implements this protocol **{CS.1::CS.1.5}**. Applications that AT-TLS has been configured to support, can be tunneled through SSL/TLS to establish a trusted channel to another trusted IT product that also implements this protocol **{CS.1::CS.1-V1R7.1}**.

  AT-TLS is configured through the PROFILE.TCPIP configuration file and the Policy Agent. .

- Packet filtering functionality that can control information flow into or out of the system based on security characteristics of the packets or of the network interface they use, as follows:
  - **{CS.1::CS.1-R12-PF-1}** Filter rules can apply to a packet based on information within the packet or information external to the packet.
    - Internal information: source address, destination address, protocol, source port, or destination port, ICMP type and code, OSPF type, and mobility header type.
    - External information: the direction of packet flow routing attribute, security class (determined by the network interface used by the packet).
  - **{CS.1::CS.1-R11-PF-3}** A z/OS TCP/IP stack configured for IP security implements a default “deny” policy in the absence of any configured filter rules.

7.3.2.7.3 System SSL

z/OS provides TLS functions via the System SSL component for applications wishing to use TLS directly (without taking advantage of the AT-TLS functions of the Communications Server). The selectable algorithms can be limited by configuring a subset of allowable algorithms at the server **{CS.2::CS.1-R8-SSL-2}**. The TLS (Version 1.2 [RFC5246], **{CS.2::CS.1-V2R4-SSL-1}**), protocol can be used to set up a trusted channel to another system through a potentially insecure network. TLS protects the data against disclosure and attacks related to integrity like undetectable modifications or replay. Servers can support encryption using AES with either 256-bit key length utilizing session keys generated with information provided through an EC Diffie-Hellman key exchange method.

Note that in the evaluated configuration, SSLv2, SSLv3, TLSv1.0 and TLSv1.1 are **not** supported.

The following cipher suites are supported:

- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 **{CS.2::CS.1-V2R1-SSL-16}** (number C024)
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 **{CS.2::CS.1-V2R1-SSL-14}** (number C02C)

The TOE supports the following groups in the "*Elliptic Curve (Supported Groups) Extension*" in the "*Client Hello*":

- secp384r1 (number 0024)
- secp521r1 (number 0025)

{CS.2::CS.1-V2R5-SSL-24}.

The TOE does not support certificate pinning. The TOE does not support session resumption.

The TOE supports certificate validation according to [RFC6125] using the DN as well as the DNS name reference identifier **{CS.2::CS.1-V2R5-SSL-1}**. The TOE supports and validates wildcards as part of the server certificates SAN extension (also according to [RFC6125]) **{CS.2::CS.1-V2R5-SSL-2}**. The TOE supports certificate based client authentication by passing the client certificate to RACF for validation. **{CS.2::CS.1-V2R5-ATTLS-1}**

When an X.509 certificate is presented as part of the TLS connection establishment, the TOE verifies the certificate path, and certification validation process by verifying the rules described in Authentication via Client Digital Certificates.

7.3.2.7.4 OpenSSH

The TOE provides the Secure Shell Protocol Version 2 (SSH v2.0) to allow users from a remote host to establish a secure connection and perform a logon to the TOE.

The TOE supports the generation of ECDSA key pairs. These key pairs are used by OpenSSH for the host keys as well as for the per-user keys.

OpenSSH suports the following cryptographic algorithms (please refer to General Cryptography for additional information):

Encryption

aes256-ctr, aes256-cbc, AEAD_AES_256_GCM

Public Key Algorithms

ecdsa-sha2-nistp384 and ecdsa-sha2-nistp521

MAC algorithms

hmac-sha2-256, hmac-sha2-512 and AEAD_AES_256_GCM

Key Exchange Methods

ecdh-sha2-nistp384, ecdh-sha2-nistp521

Key Derivation Functions

Key derivation functions used by the TOE are according to RFC 5656 (Section 4).

{CS.5::CS.1-V2R5-SSH-1}.

z/OS provides OpenSSH functionality, with an sshd daemon that supports the SSHv2 protocol **{CS.5::CS.1-R8-SSH-1}** and these commands to allow remote users to perform work on the z/OS system:

- ssh, to establish a UNIX shell environment **{CS.5::CS.1-R8-SSH-2}**
- scp to perform remote file copying operations **{CS.5::CS.1-R8-SSH-3}**
- sftp to perform file transfer operations **{CS.5::CS.1-R8-SSH-4}**
- ssh-keygen to generate the host key files and the (EC)DSA key pairs **{CS.5::CS.1-R8-SSH-7}**

Please refer to Authentication via Public/Private Key (SSH) for more information regarding user authentication to the SSH server.

The SSH protocol can be used to set up a trusted channel to another system through a potentially insecure network. SSH protects the data against disclosure and attacks related to integrity like undetectable modifications or replay. SSH supports encryption using AES with 256-bit key length **{CS.5::CS.1-R8-SSH-6}** , **{CS.5::CS.1-R9-OpenSSH-1}**.

Both OpenSSH client and server discard packets larger than 2^{18} **{CS.5::CS.1-V2R4-SSH-4}**.

The following conditions affect OpenSSH's rekeying:

- processing at most 2^30 bytes covering both sent and received data or;
- the last re-key is more than 1 hour ago.

If any of these conditions is true, the TOE initiates a re-keying. This is controlled by the option RekeyLimit which needs to be set appropriately in the OpenSSH configuration file.**{CS.5::CS.1-V2R4-SSH-5}**.

With regard to the authentication functions supported by OpenSSH, please refer to Authentication Method Summary as well as Authentication via Public/Private Key (SSH).

7.3.2.7.5 Management of Communications Server Functions

z/OS provides some basic configuration data sets and files for TCP/IP and TCP/IP based protocols. Those configuration data sets that are also related to security are:

- PROFILE.TCPIP

  Provides TCP/IP initialization parameters and specifications for network interfaces and routing.

- TCPIP.DATA

  Provides parameters for TCP/IP based client and server programs.
- Additional Communications Server configuration information (e.g., AT-TLS or firewall functions) exists in policy files accessed via the Communications Server Policy Agent.

Configuration statements in those data sets define the properties (including security properties) of the TCP/IP protocol itself as well as the main protocol server.

These policy files and data sets are protected by the system's access control mechanisms and are available for changing to authorized users only.

7.3.2.8 Confidentiality Protection of Data Sets

With z/OS confidentiality protection of data sets, users can encrypt data at rest without requiring application changes. z/OS data set encryption through RACF commands and SMS policies allows the administrator to identify the data sets or groups of data sets that require encryption. The administrator can specify an encryption key label, which refers to an encryption key. Both the key label and encryption key must exist in the ICSF key repository (CKDS). With data set encryption, the administrator is able to protect viewing the data in the clear. This is based on access to the key label that is associated with the data set and used by the access methods to encrypt and decrypt the data.

z/OS data set encryption provides the ability to encrypt the following types of data sets:

- Sequential extended format data sets, accessed through BSAM and QSAM,
- VSAM extended format data sets (KSDS, ESDS, RRDS, VRRDS, LDS), accessed through base VSAM and VSAM RLS,
- PDSE data sets which do not contain program objects.

Encrypted data sets must be in SMS-managed extended format. They also can be in compressed format. To create an encrypted data set, a key label must be supplied on new data set allocation. The key label must point to an AES-256 bit encryption key within the ICSF key repository (CKDS) to be used to encrypt or decrypt the data. For each encrypted data set, its key label is stored in the catalog. The key label is not sensitive information; it identifies the encryption key. Dataset encryption makes uses the XTS mode of operation for AES (as defined by IEEE P1619/D16) as well as CPACF for the actual cryptographic operations.

{SM.4::CDP-V2R3-1} RACF controls which applications can use specific keys to ensure that keys are used only by authorized users and jobs. To do so, the administrator can generate a RACF general resource profiles in the CSFKEYS class. The CSFKEYS class controls access to cryptographic keys with the key label. The user requires READ authority to the key label in the CSFKEYS class to access or create the encrypted data set. Since the system requires cryptographic support from ICSF to process encrypted data sets, users of encrypted data sets must be authorized to READ the CSNBKRR2 resource in the CSFSERV class, either explicitly or through a generic resource profile. **{SM.4::CDP-V2R3-9}** Conditional access to the keys can be granted in the context of accessing encrypted datasets by means of the WHEN(CRITERIA(SMS(DSENCRYPTION))) parameter for PERMIT.

{SM.4::CDP-V2R3-2} To create an encrypted data set, the administrator must assign a key label to the data set when it is newly allocated (data set create). A key label can be specified through any of the following methods:

- RACF data set profile
- JCL, dynamic allocation, TSO
- SMS data class
- IDCAMS DEFINE

{SM.4::CDP-V2R3-3} To specify a key label using the DFP segment in the RACF data set profile, use keyword DATAKEY(Key-Label). The system use this key label for extended format data sets that are created after DATAKEY is added to the data set profile. Use new keyword NODATAKEY to remove a key label, if defined, from the RACF DFP segment. The key label is ignored for a data set that is not a DASD data set.

{SM.4::CDP-V2R3-4} To specify a key label using JCL, dynamic allocation, and TSO allocate, use JCL keyword DSKEYLBL='key-label', dynamic allocation text unit DALDKYL, or TSO allocate DSKEYLBL(label-name). DSKEYLBL is effective only if the new data set is on DASD. The key label is ignored for a data set that is not a DASD data set.

{SM.4::CDP-V2R3-5} To specify a key label using SMS data class, use the Data Set Key Label field on the ISMF DEFINE/ALTER panel. The system will use this key label for extended format data sets that are created after the data set key label is added to the data class. The key label is ignored for a data set that is not a DASD data set.

{SM.4::CDP-V2R3-6} To specify a key label using the IDCAMS DEFINE command for a VSAM CLUSTER, use the KEYLABEL parameter; for example, KEYLABEL(MYLABEL). Any alternate index associated with the CLUSTER will also be encrypted and use the same key label as specified for the CLUSTER. The key label is ignored for a data set that is not a DASD data set.

{SM.4::CDP-V2R3-7} When a key label is specified on more than one source, the key label is derived from one of the above sources only on the first data set allocation (on data set create). The key label is derived in the following order of precedence:

1. From RACF DFP segment DATASET profile.
2. Explicitly specified on the DD statement, dynamic allocation text unit, TSO ALLOCATE command, or IDCAMS DEFINE control statement.
3. From the data class that applies to the current DD statement.

{SM.4::CDP-V2R5-1} The TOE supports the encryption of the RACF database in a VSAM dataset.

7.3.2.8.1 Enabling data set encryption

{SM.4::CDP-V2R3-8} An enablement action is required to allow the creation of encrypted data sets when the key label is specified through a method outside of the DFP segment in the RACF data set profile.

To allow the system to create encrypted sequential extended format or VSAM data sets using a key label specified through a method other than through the DFP segment in the RACF data set profile, the user must have at least READ authority to the following resource in the FACILITY class: STGADMIN.SMS.ALLOW.DATASET.ENCRYPT

To allow the system to create PDSE data sets using a key label specified through a method other than through the DFP segment in the RACF data set profile, the user must have at least READ authority to the following resource in the FACILITY class: STGADMIN.SMS.ALLOW.PDSE.ENCRYPT.

8 Abbreviations, Terminology, and References

8.1 Abbreviations

BCPii
Base Control Program internal interface

CC
Common Criteria

CFCC
Coupling Facility Control Code

CHPID
Channel Path Identifier

CP
Central Processor

CPC
Central Processing Complex

CSS
Channel Subsystem

HMC
Hardware Management Console

ICF
Internal Coupling Facility

IFL
Integrated Facility for Linux

IOCDS
I/O Configuration Data Set

LIC
Licensed Internal Code

MCM
Multichip Module

PR/SM
Processor Resource/Systems Manager™

PU
Processor Unit

SAP
Assist Processor

SAR
Security Assurance Requirement

SE
Support Element

SFR
Security Functional Requirement

ST
Security Target

STP
Server Time Protocol

SVMM
Separation Virtual Machine Monitor

TKE
Trusted Key Entry

TOE
Target of Evaluation

zIIP
IBM zEnterprise Integrated Information Processor

APF
Authorized Program Facility

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.

Administrator
An administrator is responsible for management activities, including setting policies that are applied by the enterprise on the operating system. This administrator could be acting remotely through a management server, from which the system receives configuration policies. An administrator can enforce settings on the system which cannot be overridden by non-administrator users.

APF
The Authorized Program Facility is used to control access and use of authorized programs.

API
A specification of routines, data structures, object classes, and variables that allows an application to make use of services provided by another software component, such as a library. APIs are often provided for a set of libraries included with the platform.

app
Software that runs on a platform and performs tasks on behalf of the user or owner of the platform, as well as its supporting documentation.

ASLR
An anti-exploitation feature which loads memory mappings into unpredictable locations. ASLR makes it more difficult for an attacker to redirect control to code that they have introduced into the address space of a process.

Assets
Information or resources to be protected by the countermeasures of a TOE.

Attack potential
The perceived potential for success of an attack, should an attack be launched, expressed in terms of an attacker's expertise, resources and motivation.

audit log
See security log

audit record
An entry in the audit log.

Authentication data
Information used to verify the claimed identity of a user.

authorized
A user be authorized to perform certain tasks with security implications. The possible authorizations are:
- supervisor state of the CPU
- APF-authorized
- having access to memory keys 0 through 7
- running with USS UID 0
- authority to FACILITY resources BPX.DAEMON, BPX.SERVER, or BPX.SUPERUSER
- authority to UNIXPRIV resources

Authorized user
A user who may, in accordance with the TSP, perform an operation.

BCPii
IBM provides Base Control Program internal interface (BCPii) support within z/OS that allows authorized applications to query, change, and perform operational procedures against the installed System z hardware base through a set of application program interfaces. These applications can access the System z hardware that the application is running on and extend their reach to other System z processors within the attached process control (Hardware Management Console) network.

CC
Common Criteria for Information Technology Security Evaluation.

CEM
Common Evaluation Methodology for Information Technology Security Evaluation.

check-stopped
This state indicates that a physical or logical processor has been subject to an unrecoverable failure.

component
The smallest selectable set of elements that may be included in a PP, an ST, or a package.

Credential
Data that establishes the identity of a user, e.g. a cryptographic key or password.

CSP
Information that is either user or system defined and is used to operate a cryptographic module in processing encryption functions including cryptographic keys and authentication data, such as passwords, the disclosure or modification of which can compromise the security of a cryptographic module or the security of the information protected by the module.

DAR Protection
Countermeasures that prevent attackers, even those with physical access, from extracting data from non-volatile storage. Common techniques include data encryption and wiping.

DEP
An anti-exploitation feature of modern operating systems executing on modern computer hardware, which enforces a non-execute permission on pages of memory. DEP prevents pages of memory from containing both data and instructions, which makes it more difficult for an attacker to introduce and execute code.

Developer
An entity that writes OS software. For the purposes of this document, vendors and developers are the same.

EP
An implementation-independent set of security requirements for a specific subset of products described.

General Purpose Operating System
A class of OSes designed to support a wide-variety of workloads consisting of many concurrent applications or services. Typical characteristics for OSes in this class include support for third-party applications, support for multiple users, and security separation between users and their respective resources. General Purpose Operating Systems also lack the real-time constraint that defines Real Time Operating Systems (RTOS). RTOSes typically power routers, switches, and embedded devices.

Host-based Firewall
A software-based firewall implementation running on the OS for filtering inbound and outbound network traffic to and from processes running on the OS.

human user
Any person who interacts with the TOE.

identity
A representation (e.g. a string) uniquely identifying an authorized user, which can either be the full or abbreviated name of that user or a pseudonym

internal TOE transfer object
Communicating data between separated parts of the TOE. An entity within the TSC that contains or receives information and upon which subjects perform operations.

object
An object is a passive entity in a computing system. Objects are subject to access control. In this document the term "object" can be used synonymously to "resource".

OS
Software that manages physical and logical resources and provides services for applications. The terms *TOE* and *OS* are interchangeable in this document.

PII
Any information about an individual maintained by an agency, including, but not limited to, education, financial transactions, medical history, and criminal or employment history and information which can be used to distinguish or trace an individual's identity, such as their name, social security number, date and place of birth, mother's maiden name, biometric records, etc., including any other personal information which is linked or linkable to an individual.

PP
An implementation-independent set of security requirements for a category of products.

processor unit
This is the generic term for the z/Architecture processor on the Multichip Module (MCM) that can be characterized as a:
- Central Processor (CP) to be used by an operating system
- Internal Coupling Facility (ICF) to be used by the Coupling Facility Control Code (CFCC)
- Integrated Facility for Linux (IFL)
- Additional Assist Processors (SAPs) to be used by the Channel Subsystem (CSS)
- IBM zEnterprise Integrated Information Processor (zIIP)

resource
An object that can be allocated to a logical partition, i.e. channel path, control unit, I/O device, storage, physical processor, logical processor.

role
A predefined set of rules establishing the allowed interactions between a user and the TOE

SAR
A requirement to assure the security of the TOE.

Sensitive Data
Sensitive data may include all user or enterprise data or may be specific application data such as PII, emails, messaging, documents, calendar items, and contacts. Sensitive data must minimally include credentials and keys. Sensitive data shall be identified in the OS's TSS by the ST author.

SFR
A requirement for security enforcement by the TOE.

ST
A set of implementation-dependent security requirements for a specific product.

subject
A subject is an active entity in a computing system. Subjects can access objects. Subjects act on behalf of users.

TOE
The product under evaluation. In this case, the Operating System as described in the TOE description and the TOEs supporting documentation.

TSF
The security functionality of the product under evaluation.

TSS
A description of how a TOE satisfies the SFRs in a ST.

User
A user is subject to configuration policies applied to the operating system by administrators. On some systems under certain configurations, a normal user can temporarily elevate privileges to that of an administrator. At that time, such a user should be considered an administrator.

8.3 References

CC — **Common Criteria for Information Technology Security Evaluation**
Version 3.1R5
Date April 2017
Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R5.pdf
Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R5.pdf
Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R5.pdf

FIPS186-4 — **Digital Signature Standard (DSS)**
Date 2013-07-19
Location https://csrc.nist.gov/pubs/fips/186-4/final

GPOSPPv4.3 — **Protection Profile for General Purpose Operating Systems Version 4.3**
Version 4.3
Date 2022-07-29
Location https://www.niap-ccevs.org/MMO/PP/PP_OS_4.3.pdf

ICSF.OVW — **z/OS 2.5 Cryptographic Services: Integrated Cryptographic Service Facility Overview**

Version SC14-7505-10
Date April 2023

MLSGUIDE **z/OS 2.5 Planning for Multilevel Security and the Common Criteria**
Version GA32-0891-50
Date August 2023

MVSTUNE.G **z/OS 2.5 MVS Initialization and Tuning Guide**
Version SA23-1379-50
Date March 2023

RFC4252 **The Secure Shell (SSH) Authentication Protocol**
Author(s) T. Ylonen, C. Lonvick
Date 2006-01-01
Location http://www.ietf.org/rfc/rfc4252.txt

RFC4253 **The Secure Shell (SSH) Transport Layer Protocol**
Author(s) T. Ylonen, C. Lonvick
Date 2006-01-01
Location http://www.ietf.org/rfc/rfc4253.txt

RFC4344 **The Secure Shell (SSH) Transport Layer Encryption Modes**
Author(s) M. Bellare, T. Kohno, C. Namprempre
Date 2006-01-01
Location http://www.ietf.org/rfc/rfc4344.txt

RFC5246 **The Transport Layer Security (TLS) Protocol Version 1.2**
Author(s) T. Dierks, E. Rescorla
Date 2008-08-01
Location http://www.ietf.org/rfc/rfc5246.txt

RFC5280 **Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile**
Author(s) D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk
Date 2008-05-01
Location http://www.ietf.org/rfc/rfc5280.txt

RFC5289 **TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)**
Author(s) E. Rescorla
Date 2008-08-01
Location http://www.ietf.org/rfc/rfc5289.txt

RFC5647 **AES Galois Counter Mode for the Secure Shell Transport Layer Protocol**
Author(s) K. Igoe, J. Solinas
Date 2009-08-01
Location http://www.ietf.org/rfc/rfc5647.txt

RFC5656 **Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer**
Author(s) D. Stebila, J. Green
Date 2009-12-01

	Location http://www.ietf.org/rfc/rfc5656.txt
RFC6125	**Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)** Author(s) P. Saint-Andre, J. Hodges Date 2011-03-01 Location http://www.ietf.org/rfc/rfc6125.txt
RFC6668	**SHA-2 Data Integrity Verification for the Secure Shell (SSH) Transport Layer Protocol** Author(s) D. Bider, M. Baushke Date 2012-07-01 Location http://www.ietf.org/rfc/rfc6668.txt
RFC6960	**X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP** Author(s) S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams Date 2013-06-01 Location http://www.ietf.org/rfc/rfc6960.txt
SSHPKGv1.0	**Functional Package for Secure Shell (SSH)** Version 1.0 Date 2021-05-13 Location https://www.niap-ccevs.org/MMO/PP/pkg_ssh_v1.0.pdf
TLSPKGv1.1	**Functional Package for Transport Layer Security (TLS)** Version 1.1 Date 2019-03-01 Location https://www.niap-ccevs.org/MMO/PP/PKG_TLS_V1.1.pdf
TSO.CUST	**z/OS 2.5 TSO/E Customization** Version SA32-0976-50 Date September 2022