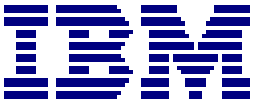

IBM AIX 7 for POWER V7.1 Technology level 7100-00-03 with optional IBM Virtual I/O Server V2.2 Security Target with BSI OSPP Compliance

Version:	**1.8**
Status:	**Release**
Last Update:	**2012-08-15**

Trademarks

IBM and the IBM logo are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Legal Notice

This document is provided AS IS with no express or implied warranties. Use the information in this document at your own risk.

This document may be reproduced or distributed in any form without prior permission provided the copyright notice is retained on all copies. Modified versions of this document may be freely distributed provided that they are clearly identified as such, and this copyright is included intact.

Revision History

Revision	Date	Author(s)	Changes to Previous Revision
1.8	2012-08-15	Scott Chapman, Andreas Siegert	Updated from AIX 6.1 to AIX 7.1 and switched to BSI OSPP.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Security Target Identification

Title:	IBM AIX 7 for POWER V7.1 Technology level 7100-00-03 with optional IBM Virtual I/O Server V2.2 Security Target with BSI OSPP Compliance
Version:	1.8
Status:	Release
Date:	2012-08-15
Sponsor:	IBM Corporation
Developer:	IBM Corporation
Certification ID:	BSI-DSZ-CC-0711
Keywords:	AIX, AIX 7.1, general-purpose operating system, POSIX, UNIX, access control, discretionary access control, information protection, labels, labled security, mandatory access control, MLS, security, Trusted AIX, trusted operating system, LPAR, VIOS, OSPP

1.2 TOE Identification

The TOE is IBM AIX 7 for POWER V7.1 with optional VIOS V2.2.

1.3 TOE Type

The TOE type is an operating system and a virtualization layer.

1.4 TOE Overview

The TOE consists of two major parts: AIX and VIOS. AIX is a highly-configurable UNIX-based operating system that meets the requirements of the BSI Operating System Protection Profile [OSPP] along with several of the [OSPP] Extended Packages.

The AIX portion of the TOE can be installed in two different modes: “BAS mode” or “LAS mode”. In BAS mode (Basic AIX Security mode), AIX offers the capabilities of [OSPP] and all of the [OSPP] Extended Packages defined in section 2 except for the Labeled Security Extended Package of [OSPP-LS]. In LAS mode (Labeled AIX Security mode, a.k.a. Trusted AIX), AIX adds the capabilities of labeled security conforming to the [OSPP-LS] Extended Package. The mode of operation (i.e., BAS mode or LAS mode) is decided at installation time.

Additionally, the IBM Virtual I/O Server (VIOS) is included in the evaluated configuration as an optional component. VIOS exists as a layer between the hardware and operating systems for virtualizing the hardware. VIOS provides logical partitions (LPARs) for running multiple operating systems on the same hardware where each instance of an operating system runs in its own LPAR. VIOS does not claim conformance to any protection profile. VIOS is treated as a separate component of the TOE with separate security problem definitions, objectives, and security functional requirements independent of those for AIX and Trusted AIX.

1.4.1 Required and optional non-TOE software/hardware/firmware

1.4.1.1 Software

There is no required non-TOE software. The following is a list of optional non-TOE software:

- IBM Network Authentication Service (NAS - a.k.a. Kerberos Version 5)
- IBM Tivoli Directory Server (TDS - a.k.a. LDAP)

1.4.1.2 Hardware/Firmware

The following is a list of required non-TOE hardware. The firmware (BootProm) is included with the hardware.

- IBM System p POWER6
- IBM System p POWER7

1.4.2 Intended method of use

AIX is a UNIX-based, multi-user, multi-tasking operating system. After successful login, users have access to a general computing environment, allowing the start-up of user applications, issuing user commands at shell level, and creating and accessing files. AIX provides adequate mechanisms to separate the users and protect their data. Privileged commands are restricted to the system administrator roles.

AIX permits one or more processors and attached peripheral and storage devices to be used by multiple users to perform a variety of functions requiring controlled shared access to the data stored on the system. Such installations are typical of personal, workgroup, or enterprise computing systems accessed by users local to, or with otherwise protected access to, the computer systems.

AIX provides facilities for on-line interaction with users. Networking is covered only to the extent to which the AIX can be considered to be part of a centrally-managed system that meets a common set of security requirements.

Optionally, VIOS can be used as a layer between AIX and the hardware to support multiple LPARs running multiple operating systems.

1.4.3 Major security features

The major AIX security features (in BAS mode) are:

- **Identification & authentication** - Provides identification and authentication of users.
- **Auditing** - Provides audit logs for logging security relevant events.
- **Discretionary access control (DAC)** - Allows object owners to control access to their objects through features like access control lists (ACLs) and the Encrypted File System (EFS).
- **Object reuse** - Provides methods to prevent data contained in deleted objects from being accessed.
- **Security management** - Provides for management of AIX security features.
- **TSF protection** - Provides methods to prevent the modification of TOE Security Functionality.
- **Privileges, authorizations, roles, and superuser emulation** - Provides mechanisms that partition and limit the amount of power a user and executables have. Superuser emulation (BAS mode only) supports the older UNIX-style of a single superuser.

- **TCB protection** - Provides additional protection to objects mark as part of the Trusted Computing Base (TCB).
- **Trusted Execution** - Provides integrity checking of specified resources at access time.
- **Protected remote access** - Provides IPsec protected connections for remote access.
- **IP filtering** - Provides IP filtering for data packets flowing through AIX.
- **Workload Partitions (WPARs)** - Provides virtual AIX environments within AIX.
- **Cryptographic Framework** - Provides a common kernel interface for hardware/software cryptographic functions.

In addition, the following AIX security features are available in LAS mode:

- **Mandatory access control (MAC)** - Provides access control to data based on security level and category labels (a.k.a. labeled security).
- **Mandatory integrity control (MIC)** - Provides access control to data based on data integrity labels.
- **Trusted Network (TN)** - Provides labeled security of network data.

The major VIOS security features are:

- **Identification & authentication** - Provides identification and authentication of users.
- **Discretionary access control** - Provides access control between SCSI device drivers and logical/physical volumes and between Ethernet adapter device drivers and Ethernet device drivers.
- **Role-based access control** - Provides multiple administrative roles for controlled management.
- **Security management** - Provides for management of VIOS security features.

1.5 TOE Description

The target of evaluation (TOE) is the AIX Version 7.1 operating system and the optional IBM Virtual I/O Server (VIOS) Version 2.2.

AIX is a general purpose, multi-user, multi-tasking operating system. It is compliant with all major international standards for UNIX systems, such as the POSIX standards, X/Open XPG 4, Spec 1170, and [FIPS180-3]. It provides a platform for a variety of applications in the governmental and commercial environment. AIX is available on a broad range of computer systems from IBM, ranging from departmental servers to multi-processor enterprise servers, and is capable of running in an LPAR (Logical Partitioning) environment.

Several servers running AIX 7.1 (any combination of BAS mode systems and LAS mode systems can be used) can be connected to form a distributed system, but not all components of such a system are components of the TOE. The communication aspects within AIX 7.1 used for this connection are also part of the evaluation. It is assumed that the communication links themselves are protected against interception and manipulation by measures which are outside the scope of this evaluation.

In LAS mode, the TOE enforces MAC, MIC, DAC, and TCB control policies to implement security goals, such as confidentiality, integrity, and accountability. LAS mode can operate in a network or stand-alone configuration. In a network configuration, LAS mode supports BSO/ESO/CIPSO/RIPSO and provides network filtering on incoming and outgoing packets, based on network interface and host filtering rules.

The AIX evaluation shall consist of a closed network of high-end, mid-range and low-end IBM System p POWER6 and POWER7 servers running the TOE. In addition, each server may optionally run VIOS.

VIOS exists as a layer between the hardware and operating systems for virtualizing the hardware. VIOS provides logical partitions (LPARs) for running multiple operating systems on the same hardware where each instance of an operating system runs in its own LPAR.

The TOE Security Functionality (TSF) consists of those parts of AIX that run in kernel mode plus some trusted processes. These are the functions that enforce the security policy as defined in this Security Target. Tools and commands executed in user mode that are used by the system administrator need also to be trusted to manage the system in a secure way but, as with other operating system evaluations, they are not considered to be part of this TSF. The TSF also consists of the parts that comprise the optional VIOS.

Table 1 and Table 2 provide a guide for what is supported in BAS mode and what is supported in LAS mode. An ‘X' means that the mode supports the description.

BAS Mode	LAS Mode	TOE Description
X	X	The TOE includes installation from CD-ROM and the network.
X	X	The TOE includes the Virtual Input/Output Server (VIOS) which allows for the virtualization of SCSI drives and network adapters.
X	X	System administration tools include the smitty non-graphical system management tool. The WebSM administrative tool is excluded.
X		The TOE includes standard networking applications, such as *ftp*, *rlogin*, *rsh*, and NFS. Port filtering will be used to protect network applications which might otherwise have security exposures.
	X	The TOE includes the following networking applications: telnet and ftp. It also includes NFS as a single level file system.
X		The TOE includes the X-Window graphical interface and many X-Window applications.
	X	The TOE supports BSO/ESO/CIPSO/RIPSO for IPv4 with an AIX specific implementation for IPv6 and provides network filtering on incoming and outgoing packets, based on network interface and host filtering rules.

Table 1: BAS mode vs. LAS mode for TOE

BAS Mode	LAS Mode	Operational Environment Description
X	X	The Operational Environment includes the hardware and the BootProm firmware.
X	X	The Operational Environment includes applications that are not evaluated, but are used as unprivileged tools to access public system services, for example the Mozilla web browser or the Adobe Acrobat Reader to access the supplied online documentation (which is provided in HTML and PDF formats). No HTTP server is included in the evaluated configuration.
X	X	The Operational Environment includes LDAP for maintaining TOE authentication data.

BAS Mode	LAS Mode	Operational Environment Description
X	X	The Operational Environment includes Kerberos for aiding in establishing a trusted channel between NFSv4 clients and servers.

Table 2: BAS mode vs. LAS mode for Operational Environment

1.5.1 Summary of security features

The following sections present a summary of the security features that the TOE offers. These security features are supported by domain separation and reference mediation, which ensure that the features are always invoked and cannot be bypassed.

1.5.1.1 AIX

1.5.1.1.1 Identification and authentication

AIX provides identification and authentication (I&A) based upon user passwords. The quality of the passwords used can be enforced through configuration options controlled by AIX. The evaluated configurations for I&A are:

- The file-based authentication method (the default configuration for authentication), which uses passwords to authenticate users.
- The LDAP authentication method configured for UNIX-type authentication, which uses passwords to authenticate users. (In the UNIX-type configuration, LDAP only stores the data used for I&A. It does not perform I&A for AIX.)
- The NAS (Kerberos Version 5) authentication method, but limited to NFSv4 client-server authentication for establishing trusted channel communications between the NFSv4 client and server.

Other authentication methods (e. g. Kerberos authentication as a general AIX authentication) that are supported by AIX in general are not part of the evaluated configuration. Especially pluggable authentication modules that, for example would allow the use a token based authentication process, are not part of the evaluated configuration.

All individual users are assigned a unique user identifier. This user identifier supports individual accountability. The TOE authenticates the claimed identity of the user before allowing the user to perform any further actions.

IBM Tivoli Directory Server (TDS) 6.1 and 6.2 are used for the LDAP service. The TDS client interface used by AIX uses the IBM Global Services Kit (GSKit) for providing SSL services. The client interface, including GSKit, is part of the Operational Environment.

1.5.1.1.2 Auditing

AIX can collect extensive auditing information about security related actions taken or attempted by users, ensuring that users are accountable for their actions.

For each event record, the audit event logger prefixes an audit header to the event-specific information. This header identifies the user and process for which this event is being audited, as well as the time of the event. The code that detects the event supplies the event type and return

code or status and optionally, additional event-specific information (the event tail). Event-specific information consists of object names (for example, files refused access or tty used in failed login attempts), subroutine parameters, and other modified information.

This audit trail can be analyzed to identify attempts to compromise security and determine the extent of the compromise. The audit tools can also extract audit records of events involving objects and/or subjects having specified security attributes.

1.5.1.1.3 Discretionary access control

Discretionary Access Control (DAC) restricts access to objects, such as files and is based on Access Control Lists (ACLs) and the standard UNIX permissions for user, group and others. Access control mechanisms also protect IPC objects from unauthorized access. BAS mode supports ACLs on sockets for TCP connections. LAS mode supports ACLs on network ports and interfaces.

In addition, AIX supports the Encrypted File System (EFS) which allows for the encryption and decryption of files using the Advanced Encryption Standard (AES). File encryption works as a type of access control mechanism. The user must have DAC access and have access to the file's encryption key in order to decrypt the file's content. AIX uses the IBM CryptoLite for C (CLiC) cryptographic module for EFS encryption and decryption.

1.5.1.1.4 Object reuse

All resources are protected from Object Reuse (scavenging) by one of three techniques: explicit initialization, explicit clearing, or storage management. Four general techniques are used to meet this requirement:

- ***Explicit Initialization:*** The resource's contents are explicitly and completely initialized to a known state before the resource is made accessible to a subject after creation.
- ***Explicit Clearing:*** The resource's contents are explicitly cleared to a known state when the resource is returned for re-use.
- ***Storage Management:*** The storage making up the resource is managed to ensure that uninitialized storage is never accessible.
- ***Erase Disk:*** AIX offers as part of its diagnostic subsystem an Erase Disc service aid that can be invoked by the administrator to overwrite all data currently stored in user-accessible blocks of a disk with predefined bit patterns.

1.5.1.1.5 Security management

The management of the security critical parameters of AIX is performed by administrative users. A set of commands that require system administrator privileges are used for system management. Security parameters are stored in specific files that are protected by the access control mechanisms of the TOE against unauthorized access by users that are not administrative users.

In BAS mode and LAS mode, security management can be split between different roles.

1.5.1.1.6 TSF protection

While in operation, the kernel software and data are protected by the hardware memory protection mechanisms. The memory and process management components of the kernel ensure a user process cannot access kernel storage or storage belonging to other processes.

TSF software and data, files and directories, kernel objects, IPC and networks sockets/packets are protected by TCB, DAC, and process isolation mechanisms. LAS mode provides additional mechanisms of MAC and MIC.

The TOE and the hardware and firmware components are required to be physically protected from unauthorized access. The system kernel mediates all access to the hardware mechanisms themselves, other than program visible CPU instruction functions.

The system administrator has the ability to start a program that checks the hardware for correct operation.

LAS Mode Only: The operational mode of AIX is intended to be the standard operating mode of the machine. The restrictions associated with operational mode cannot be overridden or bypassed by any mechanism. These restrictions are:

- the system security flags (SSFs) cannot be modified
- objects with the file security flags (FSFs) FSF_TLIB and FSF_TLIB_PROC set cannot be created, modified, or deleted

1.5.1.1.7 Privileges, authorizations, roles, and superuser emulation

The TOE implements a privilege mechanism within the kernel that allows users to implement the *least privilege principle*. A privilege is an attribute of a process that allows the process to bypass specific restrictions and limitations of the system. Privileges are associated only with processes, not user accounts. Privileges are used to override security constraints, to permit expanded use of certain system resources such as memory and disk space, and to adjust the performance and priority of the process. Restricting privileges on a process limits the damage that can result if an operation is improperly performed. Untrusted programs must not have any privileges assigned to them.

This ST describes both a “root enabled mode” and a “root disabled mode” available in BAS mode, but **only “root enabled mode” is allowed in the evaluated configuration of BAS mode**. All mention of root enabled mode and root disabled mode refer to a BAS mode system only. (In root enabled mode, the ‘root' user has the typical ‘root' authority found in previous versions of AIX. In root disabled mode, the ‘root' user has its authority reduced to the equivalence of an ordinary user.) Only root disabled mode is supported/allowed in LAS mode.

The TOE least privilege mechanism can take the place of the traditional user ID 0 (superuser/root) mechanism of UNIX. In LAS mode, user ID 0 is treated exactly like any other system user ID unless superuser emulation is in effect for the process. In BAS mode with root enabled mode enabled, user ID 0 supports the traditional superuser mechanism.

Privileges can be associated with executable files and assigned to an executing process, similar to the way the setuid bit on a file modifies the executing process's user ID. A process can also be prevented from acquiring privileges via the exec mechanism. Privileges can be used directly within a user-level program that is responsible for mediating or enforcing security by having the program retrieve its privilege set from the kernel and to make decisions based on the presence or absence of specific privileges. A process can temporarily disable one or more of its privileges if the process needs to perform an action on the system without bypassing the system security policy.

The TOE supports the policy of separation of duties, which provides for the compartmentalization of responsibility reducing the potential damage from a corrupt user or administrator, and places limits on the authority of the user or administrator. Authorizations provide a mechanism to grant rights to users to perform particular actions and run particular programs, such as programs that will run with privileges to bypass MAC, MIC, or DAC limitations. Each authorization has a well-defined

set of functions that can be performed by users who are granted that authorization. There are two types of authorized users: administrative role users and ordinary users. An administrative user is any authorized user that has one or more of the RBAC related authorizations (see the next paragraph for a discussion on RBAC). An ordinary user has no RBAC authorizations.

A role-based access control (RBAC) mechanism is implemented in AIX. Roles are predefined collections of authorizations that can be assigned to users. AIX comes with a set of predefined roles. It also allows system administrators to create new roles for their environment. AIX has two types of RBAC: Legacy RBAC and Enhanced RBAC. The evaluated configuration uses Enhanced RBAC only. All references to RBAC in this document imply Enhanced RBAC unless otherwise specified.

In addition to RBAC functions, combined roles or role based approval can be implemented according to the users needs via the "n-man rule" functionality based on the *authexec* command which will execute other commands only after all required roles have authenticated. Commands needing the n-man rule are listed in the privcmds database and cannot be executed outside of the control of the *authexec* command.

A program has the ability to query the active authorizations associated with the user running the program, and the program can behave differently and use different privileges based on the authorization set of the user running the program. For the evaluated configuration, administrators (or, administrative users) are defined as all users that have any authorization assigned to them. All user IDs below 205 are considered system IDs; they are typically used for daemons and other trusted applications.

Additionally, AIX provides a Privileged Commands (*privcmds*) database for granting privileges and setuid/setgid capabilities to trusted executables at runtime when a user has the proper authorizations. When the kernel invokes a program, it checks the database for the existence of the program. If the program exists and the user has the proper authorizations, the discretionary access control on the program is ignored and the program is invoked with the privileges and/or setuid/setgid specified in the *privcmds* database.

The TOE provides a superuser emulation mechanism that allows the system to operate similar to a standard UNIX system. Superuser emulation can be enabled for specific processes while leaving all other processes running under the standard TOE least privilege and authorization mechanisms. There are several ways in which a process can emulate superuser:

1. A process can be granted all privileges on the system, regardless of its user ID.
2. Using the PV_SU_ROOT privilege, a process can be granted all privileges associated with standard AIX/UNIX superuser regardless of its user ID, such as the privileges to bypass any DAC restrictions and to management the auditing mechanism, but not privileges that are specific to the TOE-provided augmentation of standard AIX/UNIX security functionality, such as the privileges to modify kernel authorization tables, override MAC checks, etc.
3. Alternatively, the PV_SU_EMUL privilege can be set to grant processes all privileges associated with standard AIX/UNIX superuser when their process user ID is 0.
4. A process can be granted all authorizations/roles regardless of its user ID.
5. A process can be granted a “virtual user ID” of 0 so that queries to the kernel for its user ID will return 0 even regardless of the actual user ID associated with the process.

1.5.1.1.8 TCB protection

The TOE provides the concept of a Trusted Computing Base (TCB). Kernel, device drivers, system administration utilities, and other critical software that is used to enforce and administer the security of the system are part of this TCB. In addition, any file system object in the TOE (file, directory, device, etc.) can be marked with a TCB flag: FSF_TLIB. Alternatively, executables can be marked

with the FSF_TLIB_PROC flag. The TCB is subject to several bypass control mechanisms enforced by the TOE, such as additional access control and integrity protection. Changes to objects being flagged as TCB objects can only be made when the system is in configuration mode or when the system security flag (SSF) *trustedlib_enabled* is disabled.

The integrity of objects in the TCB database is verified at every system startup and at the request of an authorized administrator.

1.5.1.1.9 Trusted Execution (TE)

In addition to the TCB, the TOE also supports a more modern form of integrity protection by monitoring files for integrity violations at access. The Trusted Execution function allows the administrator to define system and user resources for which changes to the resource are checked at access time resulting in denied access when the resource has been modified therefore preventing the execution of trojaned programs or libraries as well as the use of configuration files that have been tampered with. The checking is based on verifying SHA-256 checksums. The interface for managing the trusted execution function is the *trustchk* command.

1.5.1.1.10 Networking

1.5.1.1.10.1 Protected remote access

The TOE supports IPsec for protected remote access connections. IPsec provides integrity and confidentiality of the transported data and is able to authenticate the end points.

1.5.1.1.10.2 IP filtering

The TOE supports IP filtering of packets flowing to and through the TOE. IP packet flow can be permitted or denied based on several criteria/rules including presumed source address, destination address, and destination ports. IP packet filtering includes time-based rules where packet flow can be permitted or denied for a limited period of time after which the rules change.

1.5.1.1.11 Workload Partitions (WPARs)

AIX supports virtual environments called Workload Partitions (WPARs) which provide virtual AIX environments within AIX. WPARs provide process isolation so that applications can be installed and tested in a virtual environment. AIX supports two types of WPARs: System WPARs and Application WPARs.

A System WPAR is a virtual AIX system with its own set of users, administrators, hostname, network addresses, process isolation, IPC isolation, and file system isolation. An Application WPAR is similar to a System WPAR except without file system isolation. With the advent of WPARs, the main AIX environment is now called the Global environment. Multiple WPARs can be created and executed within the Global environment by a system administrator.

1.5.1.1.12 Cryptographic Framework

AIX supports the AIX Cryptographic Framework (ACF). This framework is implemented by the AIX kernel and allows applications access to cryptographic hardware and software supported by the kernel while at the same time isolating applications from the cryptographic hardware and software. In the evaluated configuration, IBM's CLiC software is supported by ACF.

1.5.1.1.13 Mandatory access control (LAS mode only)

LAS mode provides full mandatory access control (MAC) for all objects on the system. Every file, directory, IPC object, and process on the system is given a sensitivity label (SL) which cannot be modified by an unprivileged process. Each user account is assigned a range of valid SLs, and the user can only operate on the TOE within that range. A process (or user) can only create objects at its current SL, and can only read and write objects subject to the MAC restrictions imposed by the system. It is not possible for unauthorized users to “downgrade” information or to bypass MAC restrictions by any utility or application on the system. Copies of a file, or portions of a file, created by any possible means, will always be protected at an SL at least as high as the original file.

1.5.1.1.14 Mandatory integrity control (LAS mode only)

LAS mode provides full mandatory integrity control (MIC) for all objects on the system. Every file, directory, IPC object, and process on the system is given an integrity label (TL) which cannot be modified by an unprivileged process. Each user account is assigned a range of valid TLs, and the user can only operate on the TOE within that range. A process (or user) can only create objects at its current TL, and can only read and write objects subject to the MIC restrictions imposed by the system. It is not possible for unauthorized users to "upgrade" integrity levels associated with data or to bypass MIC restrictions by any utility of application on the system. Copies of a file, or portions of a file, created by any possible means, will always be protected at a TL no greater than that of the original file.

1.5.1.1.15 Trusted Network (LAS mode only)

LAS mode provides export and import of labeled data via network interfaces and enforces mandatory access control for network traffic by means of Trusted Network (TN). TN provides two sets of networking rules: network interface and host filtering. Both types of networking rules determine what processing occurs on a packet before its transmission or when it is received. These rules apply sensitivity labels to packets and enforce MAC restrictions on packets according to those labels.

TN network interface rules enforce packet label processing based on the physical network interface of the host. Host rules enforce packet label processing based on the source and destination IP addresses (with network masking allowed) of the packet, the source and destination ports of the request, and the protocol being used. Both types of rules provide several criteria for determining which packets to drop and which to pass.

1.5.1.2 VIOS

1.5.1.2.1 Identification & authentication

VIOS provides identification and authentication (I&A) based upon user passwords. The quality of the passwords used can be enforced through configuration options controlled by VIOS. VIOS uses a file-based database to store user I&A data.

VIOS supports both local and remote login. Remote login is supported through *telnet*.

All individual users are assigned a unique user identifier. This user identifier supports individual accountability. The TOE authenticates the claimed identity of the user before allowing the user to perform any further actions.

1.5.1.2.2 Discretionary access control

VIOS provides DAC between VIOS SCSI device drivers acting on behalf of LPAR partitions as subjects and logical/physical volumes as objects. It also provides DAC between VIOS Ethernet device drivers acting on behalf of groups of LPAR partitions sharing a virtual network and VIOS Ethernet adapter device drivers where one is the subject and the other is the object (the Ethernet packets cannot contain VLAN tags).

1.5.1.2.3 Role-based access control

VIOS includes an RBAC mechanism. VIOS RBAC roles are predefined collections of authorizations that can be assigned to users. The VIOS RBAC mechanism is built on the same mechanism used by AIX RBAC except that the role names and abilities are different. All users of VIOS are considered administrative users. Unlike AIX, there is no legacy VIOS RBAC mechanism.

In this document, the VIOS RBAC mechanism is sometimes referred to as VRBAC in order to make a clear distinction between the VIOS RBAC mechanism and the AIX RBAC mechanism when brevity is necessary.

1.5.1.2.4 Security management

VIOS uses roles to perform system/security management, but defines a separate set of roles for system management than those used by AIX. Each VIOS role has a set of commands available to it. Security parameters are stored in specific files that are protected by the access control mechanisms of the TOE against unauthorized access by users.

1.5.2 Software

The Target of Evaluation is based on the following system software:

- IBM AIX 7 for POWER V7.1 Standard Edition, Program Number 5765-G98, with Recommended Technology Package 7100-00-03.
- The Virtual I/O Server (VIOS) contained in IBM PowerVM Standard Edition Version 2.2.0.0, Program Number 5765-PVS.

The TOE documentation is supplied on CD-ROM.

Table 3 contains a list of LPPs / File Sets that comprise the TOE. For each of these “LPP Names” there may be multiple actual installable components with that prefix. An ‘X' means that the mode supports the LPP.

BAS Mode	LAS Mode	LPP Name	Description
X	X	bos	AIX Base Operating System
X	X	devices	AIX supported devices
X		printers	AIX printer drivers and control files
X	X	sysmgt	System management tools
X		X11	X Windows server, libraries, and applications.
X	X	krb5.client	Kerberos client (optional)
X	X	ldap.client	TDS (LDAP) client (optional)

BAS Mode	LAS Mode	LPP Name	Description
X	X	clic	CLiC cryptographic module

Table 3: List of LPPs / File sets

Table 4 contains the IBM PRPQ ordering information for the evaluated system.

PRPQ	Product ID
P91209	5799-GWG

Table 4: PRPQ table

1.5.3 Configurations

The evaluated configurations are defined as follows:

- Either the BAS installation mode or the LAS installation mode must be selected during installation time.
- If BAS mode is selected, RBAC must also be selected. (LAS mode includes RBAC.)
- AIX 7.1 supports the use of IPv4 and IPv6. IPv6 conforms to [RFC2460]. Claims made by updates and enhancements to [RFC2460] were not considered by this evaluation.
- Only 64 bit architectures are included.
- Web Based Systems Management (WebSM) is not included.
- Both network (NIM, Network Install Manager) and CD installations are supported.
- Only the default mechanism for identification and authentication and the LDAP authentication method configured for “UNIX-type” authentication are included. Support for other authentication options, such as smartcard authentication, is not included in the evaluation configuration.
- If the system console is used, it must be connect directly to the workstation and afforded the same physical protection as the workstation.
- In BAS mode, AIX 7.1 provides both a native and a Sys5 print system. In LAS mode, printing must be disabled in the evaluated configuration.
- LAS Mode Only: System security flags (a.k.a. kernel security flags) need to be configured as identified in section 7.2.2.14.1 "TSF invocation guarantees (TP.1)".
- The system must be configured to disable remote access for an individual user after five consecutively failed login attempts have occurred for this user.
- If in BAS mode and if a windowing environment is used, the CDE file set must be selected at installation time.
- CLiC version 4.7.1 is included in the evaluated configuration. Only the cryptographic operations defined as CLiC operations in the FCS_CKM.*, FCS_COP.*, and FCS_RNG.* security functional requirements in chapter 6.1 were subject to evaluation.
- Dynamic Partitioning (Dynamic LPAR, DLPAR) is not supported in the evaluated configuration (i.e., the dynamic (de-) allocation of resources to a partition during operations is not allowed and must be prevented by organizational means in the Operational Environment).

- If the LDAP authentication method is used by the TOE, the network connection between the TOE and the LDAP server must be protected from modification and disclosure (e.g., by using SSL).

The TOE comprises one of the server machines (and optional peripherals) listed in section 1.5.3.2 "Technical environment for use" running the system software listed in Table 3 (a server running the above listed software is referred to as a “TOE server” below).

If the product is configured with more than one TOE server, they are linked by LANs, which may be joined by bridges/routers or by TOE workstations which act as routers/gateways or they connect using the Virtual Input/Output Server (VIOS).

If other systems are connected to the network they need to be configured and managed by the same authority using an appropriate security policy not conflicting with the security policy of the TOE.

1.5.3.1 File systems

The following file system types are supported:

- the AIX journaled file system (jfs2)
- the High Sierra file system for CD-ROM drives (CDRFS)
- the DVD-ROM file system (UDFS)
- The process file system (PROCFS) (a.k.a. */proc*), provides access to the process image of each process on the machine as if the process was a “file”. Process access decisions are enforced by MAC (LAS mode only), MIC (LAS mode only), and DAC attributes inferred from the underlying process's and user security attributes.
- the Network File System (NFS) V3 and V4
- the Encrypted File System (EFS)
- the Special File System (SPECFS)

LAS Mode Note: CDRFS, UDFS, PROCFS, and (client-side) NFS are single level file systems. For mandatory access control, the labels of their mount point apply to all objects in the mounted file system. Single level file systems are not subject to mandatory integrity control and TCB policies, and their objects cannot be associated with privileges.

1.5.3.2 Technical environment for use

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

1. The TOE is running on the following hardware platforms:
   - The TOE is running in an LPAR on an IBM System p POWER6 server.
   - The TOE is running in an LPAR on an IBM System p POWER7 server.
2. The following peripherals can be run with the TOE preserving the security functionality:
   - all terminals supported by the TOE
   - all storage devices and backup devices supported by the TOE (hard disks, CDROM drives, streamer drives, floppy disk drives)[1]

[1] The system distinguishes between storage and backup devices. Storage devices are hardware devices holding parts of the AIX file system, such as hard disks and CD ROMs. Backup devices are devices used for archiving data like floppy disks and streamer tapes that do not have a file system. Note that the distinction depends on the actual usage.

- all printer devices supported by the TOE (LAS mode must have printing disabled in the evaluated configuration)

3. Network connectors supported by the TOE (e.g., Ethernet) supporting TCP/IP services over the TCP/IP protocol stack.
4. NFSv4 supports the use of the IBM Network Authentication Service (NAS) v1.4, which is based on [RFC4120] (Kerberos Version 5), for aiding in establishing a trusted channel between NFSv4 clients and servers. NAS v1.4 is part of the Operational Environment. NAS v1.4 must be configured to use LDAP for its database.

1.5.3.2.1 LPAR environment

The logical partitioning capable System p POWER6 and POWER7 servers that represent the underlying hardware for the TOE support a logical partitioned environment that enables the System p POWER6 and POWER7 systems to run multiple logical partitions concurrently. In a logical partition, an operating system instance runs with dedicated resources: processors, memory, and I/O slots. These resources are statically assigned to the logical partition. The total amount of assignable resources is limited by the physically installed resources in the system. Because the implementation of logical partitioning is static, one has to shut down every operating system instance in all logical partitions to change the resource assignment of running logical partitions.

From a functional point of view, applications on top of an operating system are running inside partitions in the same way they run on a stand-alone System p machine. There are no issues when moving an application from a stand-alone server to a partition. Operating system software needs to be modified in some areas to call Hypervisor functions instead of native code. The design of partitioning-capable System p POWER6 and POWER7 servers is such that one partition is isolated from software running in the other partitions, including protection against natural software defects and even deliberate software attempts to break the partition barriers.

The logical resources of the underlying hardware that can be assigned to a partition are:

- Processors
- Main memory regions
- I/O slots

The assignment of those resources to the individual logical partitions is stored in non-volatile RAM. This part of the NVRAM is maintained by a “Service Processor” and cannot be read or modified directly by the TOE running in a logical partition. The assignment itself is performed by a System Administrator, who uses a “Hardware Management Console” (HMC) to define those assignments. The HMC communicates with the service processor that accepts the commands from the HMC and sets the values to define the logical partitions in the non-volatile RAM (NVRAM) accordingly. A Run-Time Abstraction Layer (RTAS) provides an abstraction mechanism for platform service calls.

The functions of the underlying LPAR architecture need to be used by different parts of the TOE. The following figure shows the parts of AIX that interact with the functions of the Operational Environment. Adaptations in AIX have been made to enable the TOE to interact in an LPAR specific way with the VMM, virtual TTY console, RTAS and kernel debugger.

Please note that the support of static LPARs does not introduce any additional security functionality for the TOE - the separation between partitions and protection of the TOE from operating systems running in other logical partitions on the same underlying machine is completely enforced by the underlying machine.

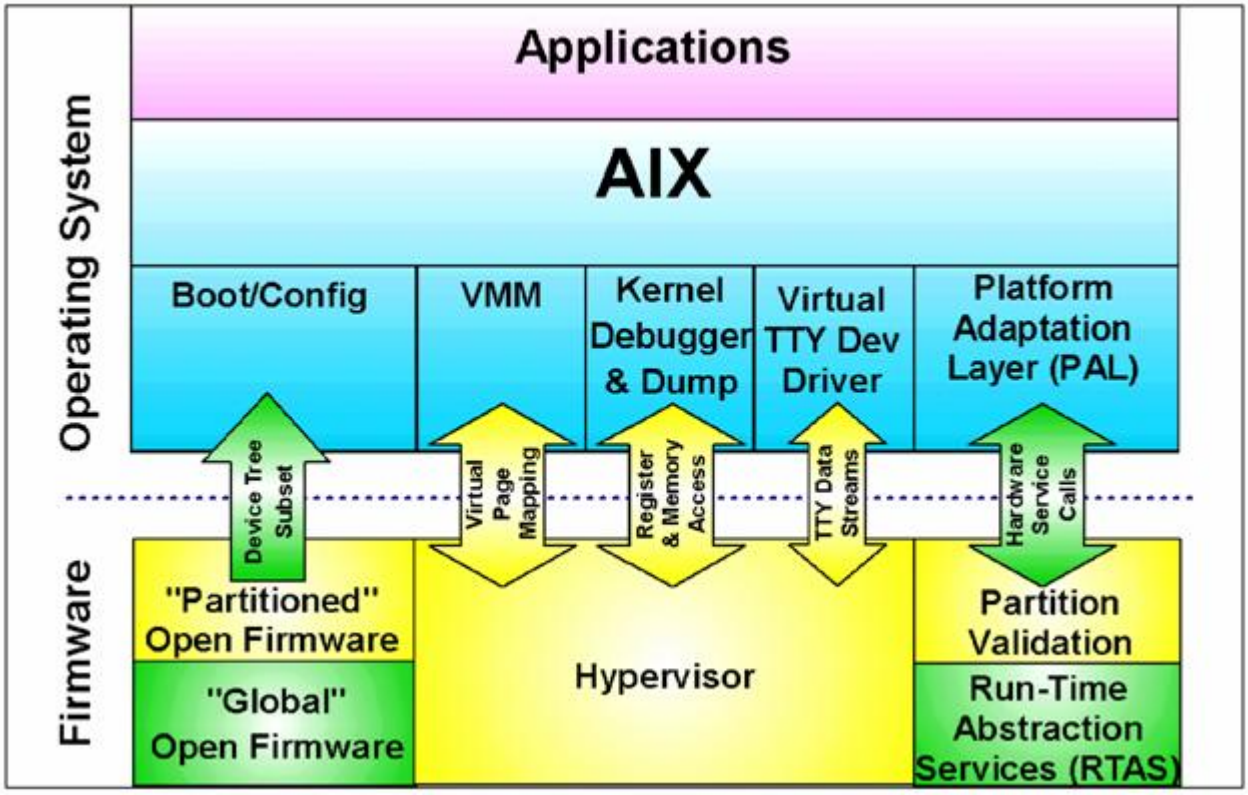

Figure 1: AIX software and firmware structure

2 CC Conformance Claim

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

This ST claims conformance to the following Protection Profiles:

- [OSPP]: BSI Operating System Protection Profile. Version 2.0 as of 2010-06-01; strict conformance.
- [OSPP-AM]: BSI OSPP Extended Package - Advanced Management. Version 2.0 as of 2010-05-28; strict conformance.
- [OSPP-CRYPTO]: BSI OSPP Extended Package - General Purpose Cryptography. Version 2.0 as of 2010-05-28; strict conformance.
- [OSPP-IV]: BSI OSPP Extended Package - Integrity Verification. Version 2.0 as of 2010-05-28; strict conformance.
- [OSPP-LS]: BSI OSPP Extended Package - Labeled Security. Version 2.0 as of 2010-05-28; strict conformance.
- [OSPP-VIRT]: BSI OSPP Extended Package - Virtualization. Version 2.0 as of 2010-05-28; strict conformance.

Common Criteria [CC] version 3.1 revision 3 is the basis for this conformance claim.

2.1 Protection Profile tailoring and additions

2.1.1 BSI Operating System Protection Profile ([OSPP])

[OSPP] applies to AIX and Trusted AIX (i.e., BAS mode and LAS mode).

The following SFR name modifications have been made to [OSPP]:

From	To	Iteration	Hierarchical substitution	Rationale
FAU_GEN.1	FAU_GEN.1(BASE)	X		Iteration is required because the ST defines FAU_GEN.1(LS).
FAU_SEL.1	FAU_SEL.1(BASE)	X		Iteration is required because the ST defines FAU_SEL.1(LS).
FDP_ACC.1(PSO)	FDP_ACC.1(PSO-AIXC), FDP_ACC.1(PSO-NFS)	X		The TOE supports two different PSO policies.
FDP_ACF.1(PSO)	FDP_ACF.1(PSO-AIXC), FDP_ACF.1(PSO-NFS)	X		The TOE supports two different PSO policies.
FDP_ITC.2	FDP_ITC.2(BASE)	X		Iteration is required because [OSPP-VIRT] contains FDP_ITC.2(VIRT).
FIA_SOS.1	FIA_SOS.1(BASE)	X		Iteration is required because VIOS defines FIA_SOS.1(VIOS).

From	To	Iteration	Hierarchical substitution	Rationale
FIA_UID.1	FIA_UID.2(BASE)	X	X	AIX & Trusted AIX support the more restrictive FIA_UID.2. Iteration is required because VIOS defines FIA_UID.2(VIOS).
FMT_SMF.1	FMT_SMF.1(BASE)	X		Iteration is required because VIOS defines FMT_SMF.1(VIOS).
FMT_MSA.1(PSO)	FMT_MSA.1(PSO-AIXC), FMT_MSA.1(PSO-NFS)	X		The TOE supports two different PSO policies.
FMT_MSA.3(PSO)	FMT_MSA.3(PSO-AIXC), FMT_MSA.3(PSO-NFS)	X		The TOE supports two different PSO policies.
FMT_SMR.1	FMT_SMR.2		X	AIX & Trusted AIX support the more restrictive FMT_SMR.2.
FPT_TDC.1	FPT_TDC.1(BASE)	X		Iteration is required because [OSPP-LS] contains FPT_TDC.1(LS).

Table 5: SFR name modifications to [OSPP]

2.1.2 BSI OSPP Extended Package - Advanced Management ([OSPP-AM])

[OSPP-AM] applies to AIX and Trusted AIX (i.e., BAS mode and LAS mode).

2.1.3 BSI OSPP Extended Package - General Purpose Cryptography ([OSPP-CRYPTO])

[OSPP-CRYPTO] applies to AIX and Trusted AIX (i.e., BAS mode and LAS mode).

2.1.4 BSI OSPP Extended Package - Integrity Verification ([OSPP-IV])

[OSPP-IV] applies to AIX and Trusted AIX (i.e., BAS mode and LAS mode).

2.1.5 BSI OSPP Extended Package - Labeled Security ([OSPP-LS])

[OSPP-LS] applies to Trusted AIX (i.e., LAS mode).

2.1.6 BSI OSPP Extended Package - Virtualization ([OSPP-VIRT])

[OSPP-VIRT] applies to AIX and Trusted AIX (i.e., BAS mode and LAS mode).

3 Security Problem Definition

3.1 Threat Environment

All threats and environmental threats refer to AIX (BAS mode) and Trusted AIX (LAS mode) unless otherwise stated. All threats and environmental threats for VIOS are explicitly marked as **VIOS only**. VIOS does not share threats or environmental threats with either AIX or Trusted AIX.

The threat agents and assets are defined by the protection profile and extended packages to which this document conforms and apply to AIX, Trusted AIX, and VIOS.

3.1.1 Threats countered by the TOE

[OSPP]_T.ACCESS.TSFDATA

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

[OSPP]_T.ACCESS.USERDATA

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

[OSPP]_T.ACCESS.TSFFUNC

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

[OSPP]_T.ACCESS.COMM

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

[OSPP]_T.RESTRICT.NETTRAFFIC

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

[OSPP]_T.IA.MASQUERADE

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

[OSPP]_T.IA.USER

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

[OSPP-AM]_T.ROLE.SNOOP

An attacker might obtain the rights granted to a role that was delegated to another user.

[OSPP-AM]_T.ROLE.DELEGATE

An attacker might delegate rights granted to a role that he does not possess or that he is not allowed to delegate.

[OSPP-IV]_T.ALTER.TSF

A threat agent might try to violate the integrity of the TSF code or TSF data in an undetectable way, resulting in a situation where security policies can be bypassed.

The threat that the integrity of the TSF code and TSF data loaded and executed before the integrity verification mechanism is active might be violated cannot be addressed by the TOE, but must be covered by the TOE environment. See A.PROTECT.INTEGRITY.

[OSPP-IV]_T.ALTER.USERDATA

A threat agent might try to violate the integrity of user data in an undetectable way, resulting in a situation where the TOE cannot reliably store user data.

[OSPP-LS]_T.DATA_NOT_SEPARATED

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

[OSPP-VIRT]_T.ACCESS.COMPENV

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

[OSPP-VIRT]_T.INFOFLOW.COMP

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

[OSPP-VIRT]_T.COMM.COMP

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

[ST]_T.ROLE.INCONSISTENT_DB

The RBAC-related databases may become inconsistent, corrupt, or inaccessible either intentionally via a threat agent or unintentionally via a malfunction or administrative error.

[ST]_T.VIOS.ACCESS.TSFDATA

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

[ST]_T.VIOS.ACCESS.TSFFUNC

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

[ST]_T.VIOS.IA.MASQUERADE

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

[ST]_T.VIOS.IA.USER

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

[ST]_T.VIOS.NET.UNPROTECTED

VIOS only: A VIOS Ethernet device driver acting on behalf of a group of LPAR partitions may try to access a VIOS Ethernet adapter device driver intended for a different VIOS Ethernet device driver (or vice versa).

[ST]_T.VIOS.VOL.UNPROTECTED

VIOS only: A VIOS SCSI device driver acting on behalf of an LPAR partition may try to access logical volumes or physical volumes that are not assigned to device driver.

3.1.2 Threats countered by the Operational Environment

[OSPP-IV]_TE.MODIFY_ENVIRONMENT

An external entity might try to violate security policies by manipulating the TOE environment; for example, by (directly or indirectly) installing a device driver that uses hardware functions (e.g., direct memory access) to access or violate the integrity of TSF data or TSF functions.

[ST]_TE.LPAR.ACCESS

A threat agent in a different logical partition might access resources assigned to the TOE's logical partition.

3.2 Assumptions

3.2.1 Environment of use of the TOE

All assumptions refer to AIX (BAS mode) and Trusted AIX (LAS mode) unless otherwise stated. All assumptions for VIOS are explicitly marked as **VIOS only**. VIOS does not share assumptions with either AIX or Trusted AIX.

3.2.1.1 Physical

[OSPP]_A.PHYSICAL

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

[ST]_A.VIOS.PHYSICAL

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

3.2.1.2 Personnel

[OSPP]_A.MANAGE

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

[OSPP]_A.AUTHUSER

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

[OSPP]_A.TRAINEDUSER

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

[ST]_A.VIOS.MANAGE

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

[ST]_A.VIOS.AUTHUSER

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

[ST]_A.VIOS.TRAINEDUSER

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

3.2.1.3 Procedural

[OSPP]_A.DETECT

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

[OSPP]_A.PEER.MGT

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

Application Note 1:

The operational environment must provide the X.509v3 certificates used by the TOE. Those certificates must comply with the cryptographic requirements specified in this ST.

Application Note 2:

Administrators must ensure that NAS (Kerberos Key Distribution Center (KDC)) provides password complexity support and failed login attempt abatement for NAS accounts that meet or exceed the password complexity requirements of the TOE.

Application Note 3:

NAS (Kerberos) must be configured to provide key generation sufficient to support the NFSv4 trusted client/server communications specified in section 7.2.2.14.8.

[OSPP]_A.PEER.FUNC

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

[OSPP-IV]_A.PROTECT.INTEGRITY

It is assumed that the integrity of the following information is ensured:

- All TSF code, including the integrity verification functionality that is loaded and executed before the invocation of the integrity verification mechanism.
- All TSF data, including TSF data to perform integrity verification used by the TSF code loaded and executed before the invocation of the integrity verification mechanism.

[ST]_A.VIOS.DETECT

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

3.2.1.4 Connectivity

[OSPP]_A.CONNECT

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

3.3 Organizational Security Policies

All OSPs refer to AIX (BAS mode) and Trusted AIX (LAS mode) unless otherwise stated. All OSPs for VIOS are explicitly marked as **VIOS only**. VIOS does not share OSPs with either AIX or Trusted AIX.

[OSPP]_P.ACCOUNTABILITY

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

[OSPP]_P.USER

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

[OSPP-AM]_P.APPROVE

Specific rights assigned to users and controlled by the TSF shall only be exercisable if approved by a second user.

[OSPP-LS]_P.CLEARANCE

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

[OSPP-LS]_P.LABELED_OUTPUT

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

[OSPP-LS]_P.RESOURCE_LABELS

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

[OSPP-LS]_P.USER_CLEARANCE

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

[ST]_P.DISK.OVERWRITE

Administrators shall be able to support information compartmentalization by preventing recovery of logically deleted information from physically and logically intact SCSI hard disk drives before they are re-used within the same system. Such hard disk drives will remain within the physical and logical protection domain of the TOE and will reside within the TSC.

[ST]_P.MANDATORY_INTEGRITY

The TOE shall be capable of distinguishing between levels of trustworthiness in terms of integrity, and the TOE shall prevent data from being modified by users operating at a lower level of trust.

[ST]_P.VIOS.USER

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

4 Security Objectives

4.1 Objectives for the TOE

All objectives refer to AIX (BAS mode) and Trusted AIX (LAS mode) unless otherwise stated. All objectives for VIOS are explicitly marked as **VIOS only**. VIOS does not share objectives with either AIX or Trusted AIX.

[OSPP]_O.AUDITING

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

[OSPP]_O.CRYPTO.NET

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

[OSPP]_O.DISCRETIONARY.ACCESS

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

[OSPP]_O.NETWORK.FLOW

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

[OSPP]_O.SUBJECT.COM

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

[OSPP]_O.I&A

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

[OSPP]_O.MANAGE

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

[OSPP]_O.TRUSTED_CHANNEL

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

[OSPP-AM]_O.ROLE.DELEGATE

The TOE must allow roles assigned to users for performing security-relevant management tasks to be delegated to other users in accordance with the security policy.

[OSPP-AM]_O.ROLE.MGMT

The TOE must allow security management actions based on roles to be assigned to different users.

[OSPP-AM]_O.ROLE.APPROVE

The TOE must prevent the execution of user actions allowed by a specific right until a second user with a different right approves this action.

[OSPP-CRYPTO]_O.CRYPTO.BASIC

The TSF must provide the following cryptographic services for general use by authorized entities:

- symmetric and asymmetric ciphers,
- message digest generation,
- symmetric and asymmetric key generation.

[OSPP-IV]_O.INTEGRITY.TSF

The TOE shall be able to verify the integrity of both TSF code and TSF data to ensure that they have not been modified when compared to the integrity information in the integrity database.

[OSPP-IV]_O.INTEGRITY.USERDATA

The TOE shall be able to verify the integrity of user data to ensure that it has not been modified when compared to the integrity information in the integrity database.

[OSPP-IV]_O.INTEGRITY.ACTION

The TOE shall perform pre-defined actions upon detection of a breach of integrity.

[OSPP-IV]_O.INTEGRITY.MANAGE

The TOE shall be able to allow authorized users to update the integrity verification database covering TSF data, the TSF code, and user data.

Also, the TOE shall be able to allow authorized users to configure actions to be performed upon the detection of a breach of integrity.

[OSPP-LS]_O.LS.CONFIDENTIALITY

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

[OSPP-LS]_O.LS.PRINT

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

[OSPP-LS]_O.LS.LABEL

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

[OSPP-VIRT]_O.COMP.INFO_FLOW_CTRL

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

[OSPP-VIRT]_O.COMP.RESOURCE_ACCESS

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

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

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

[OSPP-VIRT]_O.COMP.IDENT

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

[ST]_O.DISK.OVERWRITTEN

The TOE shall offer administrators a mechanism to overwrite user-accessible blocks of SCSI hard disk drives with predefined bit patterns.

[ST]_O.MANDATORY_INTEGRITY

LAS mode only: The TOE shall control access to resources based on the integrity level of the information being accessed and the integrity level of the subject attempting to access that information.

[ST]_O.ROLE

The TOE shall prevent users from gaining access to and performing operations on its resources/objects unless they have been granted access by the resource/object owner or they have been assigned to a role (by an authorized administrator) which permits those operations.

[ST]_O.ROLE.AUTHORIZATIONS

The TOE shall ensure that only authorized users gain access to protected TOE resources and that this access is controlled by authorized administrators.

[ST]_O.ROLE.CONSISTENT_DB

The TOE shall detect inconsistencies, corruption, and inaccessibility in the RBAC-related databases and enforce a fail secure policy.

[ST]_O.ROLE.HIERARCHY

The TOE shall allow hierarchical definitions of roles. Hierarchical definition of roles means the ability to define roles in terms of other roles.

[ST]_O.ROLE.SEP_DUTY

The TOE shall provide the capability of enforcing 'separation of duties', so that no single user has to be granted the right to perform all operations on important information.

[ST]_O.STACK.NO_EXEC

The TOE shall offer a mechanism to prevent the execution of code on the stack of selected processes.

[ST]_O.TCB.ACCESS

The TOE shall control write and/or execute access to resources protected as part of the trusted computing base as specified by an authorized administrator.

[ST]_O.TN.ACCESS

LAS mode only: The TOE shall control access between the TOE and other systems based on host security attributes and the network interface on which packets are sent or received.

[ST]_O.VIOS.I&A

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

[ST]_O.VIOS.MANAGE

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

[ST]_O.VIOS.NET.PROTECTED

VIOS only: The TSF shall control access between VIOS Ethernet adapter device drivers and VIOS Ethernet device drivers acting on behalf of groups of LPAR partitions sharing a virtual network. The TSF shall allow authorized users to specify which VIOS Ethernet adapter device drivers may be accessed by a VIOS Ethernet device driver acting on behalf of a group of LPAR partitions sharing a virtual network.

[ST]_O.VIOS.ROLE

VIOS only: The TOE shall prevent users from gaining access to and performing operations on its resources/objects unless they have been granted access by the resource/object owner or they have been assigned to a role (by an authorized administrator) which permits those operations.

[ST]_O.VIOS.ROLE.HIERARCHY

VIOS only: The TOE shall allow hierarchical definitions of roles. Hierarchical definition of roles means the ability to define roles in terms of other roles.

[ST]_O.VIOS.ROLE.SEP_DUTY

VIOS only: The TOE shall provide the capability of enforcing 'separation of duties', so that no single user has to be granted the right to perform all operations on important information.

[ST]_O.VIOS.VOL.PROTECTED

VIOS only: The TSF shall control access between LPAR partitions and logical/physical volumes and VIOS SCSI device drivers acting on behalf of a group of LPAR partitions. The TSF shall allow authorized users to specify which logical/physical volumes may be accessed by the VIOS SCSI device drivers.

4.2 Objectives for the Operational Environment

All operational environment objectives refer to AIX (BAS mode) and Trusted AIX (LAS mode) unless otherwise stated. All operational environment objectives for VIOS are explicitly marked as **VIOS only**. VIOS does not share operational environment objectives with either AIX or Trusted AIX.

[OSPP]_OE.ADMIN

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

[OSPP]_OE.REMOTE

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

[OSPP]_OE.INFO_PROTECT

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

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

[OSPP]_OE.INSTALL

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

[OSPP]_OE.MAINTENANCE

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

[OSPP]_OE.PHYSICAL

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

[OSPP]_OE.RECOVER

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

[OSPP]_OE.TRUSTED.IT.SYSTEM

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

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

[OSPP-IV]_OE.SECURE_LOAD

The operational environment shall perform checks that ensure the integrity of the TSF code and TSF data loaded and executed before the successful execution of the integrity verification TSF; ensure protection against replay of older versions of that TSF code and TSF data; and ensure that the TSF code and TSF data, when stored, loaded and executed, are protected against reading or loading by unauthorized entities in the TOE environment. If the TOE is started in a different environment or if the TOE is started even when the operational environment has detected a violation of the TOE's integrity, the operational environment shall ensure that the manipulated TOE when started is not able to generate false evidence of its own integrity and the operational environment shall also not falsely generate evidence for the integrity of the TOE when it has not successfully verified the integrity of the TOE.

[OSPP-IV]_OE.SECURE_OPERATION

The operational environment shall ensure that the TSF code and TSF data (when in operation) cannot be manipulated or intercepted by entities not under the control of the TOE.

[ST]_OE.LPAR.NO_ACCESS

The underlying hardware must protect the resources assigned to the TOE's logical partition against access from software running in a different logical partition.

[ST]_OE.VIOS.ADMIN

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

[ST]_OE.VIOS.INFO_PROTECT

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

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

[ST]_OE.VIOS.INSTALL

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

[ST]_OE.VIOS.MAINTENANCE

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

[ST]_OE.VIOS.PHYSICAL

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

[ST]_OE.VIOS.RECOVER

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

4.3 Security Objectives Rationale

4.3.1 Coverage

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

Objective	Threats / OSPs
[OSPP]_O.AUDITING	[OSPP]_P.ACCOUNTABILITY
[OSPP]_O.CRYPTO.NET	[OSPP]_T.ACCESS.TSFDATA [OSPP]_T.ACCESS.USERDATA [OSPP]_T.ACCESS.TSFFUNC
[OSPP]_O.DISCRETIONARY.ACCESS	[OSPP]_T.ACCESS.TSFDATA [OSPP]_T.ACCESS.USERDATA
[OSPP]_O.NETWORK.FLOW	[OSPP]_T.RESTRICT.NETTRAFFIC
[OSPP]_O.SUBJECT.COM	[OSPP]_T.ACCESS.TSFDATA [OSPP]_T.ACCESS.USERDATA
[OSPP]_O.I&A	[OSPP]_T.IA.MASQUERADE [OSPP]_T.IA.USER
[OSPP]_O.MANAGE	[OSPP]_T.ACCESS.TSFFUNC [OSPP]_P.ACCOUNTABILITY [OSPP]_P.USER
[OSPP]_O.TRUSTED_CHANNEL	[OSPP]_T.ACCESS.COMM

Objective	Threats / OSPs
[OSPP-AM]_O.ROLE.DELEGATE	[OSPP-AM]_T.ROLE.SNOOP [OSPP-AM]_T.ROLE.DELEGATE
[OSPP-AM]_O.ROLE.MGMT	[OSPP]_T.ACCESS.TSFFUNC
[OSPP-AM]_O.ROLE.APPROVE	[OSPP-AM]_P.APPROVE
[OSPP-CRYPTO]_O.CRYPTO.BASIC	[OSPP]_T.ACCESS.USERDATA
[OSPP-IV]_O.INTEGRITY.TSF	[OSPP-IV]_T.ALTER.TSF
[OSPP-IV]_O.INTEGRITY.USERDATA	[OSPP-IV]_T.ALTER.USERDATA
[OSPP-IV]_O.INTEGRITY.ACTION	[OSPP-IV]_T.ALTER.TSF [OSPP-IV]_T.ALTER.USERDATA
[OSPP-IV]_O.INTEGRITY.MANAGE	[OSPP-IV]_T.ALTER.TSF [OSPP-IV]_T.ALTER.USERDATA
[OSPP-LS]_O.LS.CONFIDENTIALITY	[OSPP-LS]_T.DATA_NOT_SEPARATED [OSPP-LS]_P.CLEARANCE [OSPP-LS]_P.USER_CLEARANCE
[OSPP-LS]_O.LS.PRINT	[OSPP-LS]_P.LABELED_OUTPUT
[OSPP-LS]_O.LS.LABEL	[OSPP-LS]_P.RESOURCE_LABELS [OSPP-LS]_P.USER_CLEARANCE
[OSPP-VIRT]_O.COMP.INFO_FLOW_CTRL	[OSPP-VIRT]_T.INFOFLOW.COMP
[OSPP-VIRT]_O.COMP.RESOURCE_ACCESS	[OSPP-VIRT]_T.ACCESS.COMPENV [OSPP-VIRT]_T.COMM.COMP
[OSPP-VIRT]_O.COMP.IDENT	[OSPP-VIRT]_T.ACCESS.COMPENV [OSPP-VIRT]_T.INFOFLOW.COMP [OSPP-VIRT]_T.COMM.COMP
[ST]_O.DISK.OVERWRITTEN	[ST]_P.DISK.OVERWRITE
[ST]_O.MANDATORY_INTEGRITY	[ST]_P.MANDATORY_INTEGRITY
[ST]_O.ROLE	[OSPP]_T.ACCESS.TSFDATA [OSPP]_T.ACCESS.USERDATA [OSPP]_T.ACCESS.TSFFUNC
[ST]_O.ROLE.AUTHORIZATIONS	[OSPP]_T.ACCESS.TSFDATA [OSPP]_T.ACCESS.TSFFUNC
[ST]_O.ROLE.CONSISTENT_DB	[ST]_T.ROLE.INCONSISTENT_DB
[ST]_O.ROLE.HIERARCHY	[OSPP]_P.USER
[ST]_O.ROLE.SEP_DUTY	[OSPP]_P.USER

Objective	Threats / OSPs
[ST]_O.STACK.NO_EXEC	[OSPP]_T.ACCESS.TSFDATA [OSPP]_T.ACCESS.USERDATA [OSPP]_T.ACCESS.TSFFUNC [OSPP]_T.IA.MASQUERADE
[ST]_O.TCB.ACCESS	[OSPP-IV]_T.ALTER.TSF
[ST]_O.TN.ACCESS	[OSPP-LS]_P.CLEARANCE
[ST]_O.VIOS.I&A	[ST]_T.VIOS.IA.MASQUERADE [ST]_T.VIOS.IA.USER
[ST]_O.VIOS.MANAGE	[ST]_T.VIOS.ACCESS.TSFFUNC [ST]_P.VIOS.USER
[ST]_O.VIOS.NET.PROTECTED	[ST]_T.VIOS.ACCESS.TSFFUNC [ST]_T.VIOS.NET.UNPROTECTED
[ST]_O.VIOS.ROLE	[ST]_T.VIOS.ACCESS.TSFDATA [ST]_T.VIOS.ACCESS.TSFFUNC
[ST]_O.VIOS.ROLE.HIERARCHY	[ST]_P.VIOS.USER
[ST]_O.VIOS.ROLE.SEP_DUTY	[ST]_P.VIOS.USER
[ST]_O.VIOS.VOL.PROTECTED	[ST]_T.VIOS.ACCESS.TSFFUNC [ST]_T.VIOS.VOL.UNPROTECTED

Table 6: Mapping of security objectives to threats and policies

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

Objective	Assumptions / Threats / OSPs
[OSPP]_OE.ADMIN	[OSPP]_A.MANAGE [OSPP]_A.AUTHUSER [OSPP]_A.TRAINEDUSER
[OSPP]_OE.REMOTE	[OSPP]_A.CONNECT [OSPP]_T.ACCESS.COMM
[OSPP]_OE.INFO_PROTECT	[OSPP]_A.PHYSICAL [OSPP]_A.MANAGE [OSPP]_A.AUTHUSER [OSPP]_A.TRAINEDUSER [OSPP]_P.USER
[OSPP]_OE.INSTALL	[OSPP]_A.MANAGE [OSPP]_A.DETECT
[OSPP]_OE.MAINTENANCE	[OSPP]_A.DETECT

Objective	Assumptions / Threats / OSPs
[OSPP]_OE.PHYSICAL	[OSPP]_A.PHYSICAL
[OSPP]_OE.RECOVER	[OSPP]_A.MANAGE [OSPP]_A.DETECT
[OSPP]_OE.TRUSTED.IT.SYSTEM	[OSPP]_A.PEER.MGT [OSPP]_A.PEER.FUNC [OSPP]_A.CONNECT
[OSPP-IV]_OE.SECURE_LOAD	[OSPP-IV]_A.PROTECT.INTEGRITY
[OSPP-IV]_OE.SECURE_OPERATION	[OSPP-IV]_TE.MODIFY_ENVIRONMENT
[ST]_OE.LPAR.NO_ACCESS	[ST]_TE.LPAR.ACCESS
[ST]_OE.VIOS.ADMIN	[ST]_A.VIOS.MANAGE [ST]_A.VIOS.AUTHUSER [ST]_A.VIOS.TRAINEDUSER
[ST]_OE.VIOS.INFO_PROTECT	[ST]_A.VIOS.PHYSICAL [ST]_A.VIOS.MANAGE [ST]_A.VIOS.AUTHUSER [ST]_A.VIOS.TRAINEDUSER
[ST]_OE.VIOS.INSTALL	[ST]_A.VIOS.MANAGE [ST]_A.VIOS.DETECT
[ST]_OE.VIOS.MAINTENANCE	[ST]_A.VIOS.DETECT
[ST]_OE.VIOS.PHYSICAL	[ST]_A.VIOS.PHYSICAL
[ST]_OE.VIOS.RECOVER	[ST]_A.VIOS.MANAGE [ST]_A.VIOS.DETECT

Table 7: Mapping of security objectives for the Operational Environment to assumptions, threats and policies

4.3.2 Sufficiency

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

Threat	Rationale for security objectives
[OSPP]_T.ACCESS.TSFDATA	The threat of accessing TSF data without proper authorization is removed by: • [OSPP]_O.CRYPTO.NET requiring cryptographically-protected communication channels for data including TSF data controlled by the TOE in transit between trusted IT systems,

Threat	Rationale for security objectives
	• [OSPP]_O.DISCRETIONARY.ACCESS requiring that data, including TSF data stored with the TOE, have discretionary access control protection, • [OSPP]_O.SUBJECT.COM requiring the TSF to mediate communication between subjects. • [ST]_O.ROLE requiring the TOE to prevent users from gaining access to and performing operations on its resources/objects unless they have been granted access by the resource/object owner or they have been assigned to a role which permits those operations. • [ST]_O.ROLE.AUTHORIZATIONS requiring users to be authorized to access protected TOE resources, • [ST]_O.STACK.NO_EXEC disallowing a threat agent, at the discretion of an authorized administrator, from using buffer overflow attacks to place arbitrary code on the stack of a trusted process and forcing the trusted process into executing the arbitrary code.
[OSPP]_T.ACCESS.USERDATA	The threat of accessing user data without proper authorization is removed by: • [OSPP]_O.CRYPTO.NET requiring cryptographically-protected communication channels for data including user data controlled by the TOE in transit between trusted IT systems, • [OSPP]_O.DISCRETIONARY.ACCESS requiring that data including user data stored with the TOE, have discretionary access control protection, • [OSPP]_O.SUBJECT.COM requiring the TSF to mediate communication between subjects. • [OSPP-CRYPTO]_O.CRYPTO.BASIC requiring the TSF to provide cryptographic services for general use by authorized entities, including encryption, decryption, and message digest generation services. • [ST]_O.ROLE requiring the TOE to prevent users from gaining access to and performing operations on its resources/objects unless they have been granted access by the resource/object owner or they have been assigned to a role which permits those operations. • [ST]_O.STACK.NO_EXEC disallowing a threat agent, at the discretion of an authorized administrator, from using buffer overflow attacks to place arbitrary code on the stack of a trusted process and forcing the trusted process into executing the arbitrary code.
[OSPP]_T.ACCESS.TSFFUNC	The threat of accessing TSF functions without proper authorization is removed by: • [OSPP]_O.CRYPTO.NET requiring cryptographically-protected communication channels to limit which TSF functions are accessible to external entities, • [OSPP]_O.MANAGE requiring that only authorized users utilize management TSF functions.

Threat	Rationale for security objectives
	• [OSPP-AM]_O.ROLE.MGMT requiring the TOE to allow security management actions based on roles to be assigned to different users. • [ST]_O.ROLE requiring the TOE to prevent users from gaining access to and performing operations on its resources/objects unless they have been granted access by the resource/object owner or they have been assigned to a role which permits those operations. • [ST]_O.ROLE.AUTHORIZATIONS requiring users to be authorized to access protected TOE resources, • [ST]_O.STACK.NO_EXEC disallowing a threat agent, at the discretion of an authorized administrator, from using buffer overflow attacks to place arbitrary code on the stack of a trusted process and forcing the trusted process into executing the arbitrary code.
[OSPP]_T.ACCESS.COMM	The threat of accessing a communication channel that establishes a trust relationship between the TOE and another remote trusted IT system is removed by: • [OSPP]_O.TRUSTED_CHANNEL requiring that the TOE implements a trusted channel between itself and a remote trusted IT system protecting the user data and TSF data transferred over this channel from disclosure and undetected modification and prevents masquerading of the remote trusted IT system, • [OSPP]_OE.REMOTE requiring that those systems providing the functions required by the TOE are sufficiently protected from any attack that may cause those functions to provide false results.
[OSPP]_T.RESTRICT.NETTRAFFIC	The threat of accessing information or transmitting information to other recipients via network communication channels without authorization for this communication attempt is removed by: • [OSPP]_O.NETWORK.FLOW requiring the TOE to mediate the communication between itself and remote entities in accordance with its security policy.
[OSPP]_T.IA.MASQUERADE	The threat of masquerading as an authorized entity in order to gain unauthorized access to user data, TSF data or TOE resources is removed by: • [OSPP]_O.I&A requiring that each entity interacting with the TOE is properly identified and authenticated before allowing any action the TOE is defined to provide to authenticated users only. • [ST]_O.STACK.NO_EXEC disallowing a threat agent, at the discretion of an authorized administrator, from using buffer overflow attacks to place arbitrary code on the stack of a trusted process and forcing the trusted process into executing the arbitrary code.

Threat	Rationale for security objectives
[OSPP]_T.IA.USER	The threat of accessing user data, TSF data or TOE resources without being identified and authenticated is removed by: • [OSPP]_O.I&A requiring that each entity interacting with the TOE is properly identified and authenticated before allowing any action the TOE has defined to provide to authenticated users only.
[OSPP-AM]_T.ROLE.SNOOP	The threat of an attacker obtaining the rights granted to a role that was delegated to another user is removed by: • [OSPP-AM]_O.ROLE.DELEGATE requiring the TOE to allow delegation of roles to other users in accordance with the security policy.
[OSPP-AM]_T.ROLE.DELEGATE	The threat of an attacker delegating rights granted to a role that he does not possess or that he is not allowed to delegate is removed by: • [OSPP-AM]_O.ROLE.DELEGATE requiring the TOE to allow roles assigned to users for performing security-relevant management tasks to be delegated.
[OSPP-IV]_T.ALTER.TSF	The threat of violating the integrity of the TSF code or TSF data in an undetectable way, resulting in a situation where security policies can be bypassed is removed by: • [OSPP-IV]_O.INTEGRITY.TSF requiring the TOE to verify the integrity of both TSF code and TSF data to ensure that they have not been modified when compared to the integrity information in the integrity database. • [OSPP-IV]_O.INTEGRITY.ACTION requiring the TOE to perform predefined actions upon detection of a breach of integrity. • [OSPP-IV]_O.INTEGRITY.MANAGE requiring the TOE to allow authorized users to update the integrity verification database for TSF data. • [ST]_O.TCB.ACCESS requiring the TOE to control write and execute access to objects marked as part of the TCB.
[OSPP-IV]_T.ALTER.USERDATA	The threat of violating the integrity of the user data in an undetectable way resulting in a situation where the TOE cannot reliably store user data is removed by: • [OSPP-IV]_O.INTEGRITY.USERDATA requiring the TOE to verify the integrity of user data to ensure that it has not been modified when compared to the integrity information in the integrity database. • [OSPP-IV]_O.INTEGRITY.ACTION requiring the TOE to perform predefined actions upon detection of a breach of integrity. • [OSPP-IV]_O.INTEGRITY.MANAGE requiring the TOE to allow authorized users to update the integrity verification database for user data.

Threat	Rationale for security objectives
[OSPP-LS]_T.DATA_NOT_SEPARATED	The threat of not adequately separating data on the basis of its sensitivity label, thereby allowing information to flow illicitly from or to users, is removed by: • [OSPP-LS]_O.LS.CONFIDENTIALITY requiring the TOE to control information flow between entities and resources, based on the sensitivity labels of users and resources.
[OSPP-VIRT]_T.ACCESS.COMPENV	The threat of utilizing or modifying the runtime environment of compartments executing on behalf of other users is removed by: • [OSPP-VIRT]_O.COMP.RESOURCE_ACCESS requiring the TOE to control access of compartments to objects and resources under its control. • [OSPP-VIRT]_O.COMP.IDENT requiring the TOE to identify the compartment requesting to access resources, objects or information for each access request.
[OSPP-VIRT]_T.INFOFLOW.COMP	The threat of accessing information without authorization by the information flow control policy is removed by: • [OSPP-VIRT]_O.COMP.INFO_FLOW_CTRL requiring the TOE to control information flow between compartments under the control of the TOE based on security attributes of these compartments and potentially other TSF data. • [OSPP-VIRT]_O.COMP.IDENT requiring the TOE to identify the compartment requesting to access resources, objects or information for each access request.
[OSPP-VIRT]_T.COMM.COMP	The threat of accessing the data communicated between compartments or between a compartment and an external entity is removed by: • [OSPP-VIRT]_O.COMP.RESOURCE_ACCESS requiring the TOE to control access of compartments to objects and resources under its control, • [OSPP-VIRT]_O.COMP.IDENT requiring the TOE to identify the compartment requesting to access resources, objects or information for each access request.
[ST]_T.ROLE.INCONSISTENT_DB	The threat that the RBAC-related databases may become inconsistent, corrupt, or inaccessible either intentionally via a threat agent or unintentionally via a malfunction or administrative error is countered by: • [ST]_O.ROLE.CONSISTENT_DB requiring the TOE to detect inconsistencies, corruption, and inaccessibilities in the RBAC-related databases and enforce a fail secure policy.
[ST]_T.VIOS.ACCESS.TSFDATA	The threat of accessing TSF data without proper authorization is removed by: • [ST]_O.VIOS.ROLE requiring the TOE to prevent users from gaining access to and performing operations on its resources/objects unless they have been granted access by the resource/object owner or they have been assigned to a role which permits those operations.

Threat	Rationale for security objectives
[ST]_T.VIOS.ACCESS.TSFFUNC	The threat of accessing TSF functions without proper authorization is removed by: • [ST]_O.VIOS.MANAGE requiring that only authorized users utilize management TSF functions. • [ST]_O.VIOS.NET.PROTECTED which provides access control between the VIOS Ethernet adapter device drivers and VIOS Ethernet device drivers. • [ST]_O.VIOS.ROLE requiring the TOE to prevent users from gaining access to and performing operations on its resources/objects unless they have been granted access by the resource/object owner or they have been assigned to a role which permits those operations. • [ST]_O.VIOS.VOL.PROTECTED which provides access control between VIOS SCSI device drivers and logical/physical volumes.
[ST]_T.VIOS.IA.MASQUERADE	The threat of masquerading as an authorized entity in order to gain unauthorized access to user data, TSF data or TOE resources is removed by: • [ST]_O.VIOS.I&A requiring that each entity interacting with the TOE is properly identified and authenticated before allowing any action the TOE is defined to provide to authenticated users only.
[ST]_T.VIOS.IA.USER	The threat of accessing user data, TSF data or TOE resources without being identified and authenticated is removed by: • [ST]_O.VIOS.I&A requiring that each entity interacting with the TOE is properly identified and authenticated before allowing any action the TOE has defined to provide to authenticated users only.
[ST]_T.VIOS.NET.UNPROTECTED	The threat of a VIOS Ethernet device driver acting on behalf of a group of LPAR partitions attempting to access a VIOS Ethernet adapter device driver and vice versa is removed by: • [ST]_O.VIOS.NET.PROTECTED which provides access control between the VIOS Ethernet adapter device drivers and VIOS Ethernet device drivers.
[ST]_T.VIOS.VOL.UNPROTECTED	The threat of a VIOS SCSI device driver acting on behalf of an LPAR partition attempting to access a logical volume that is not assigned to the partition is removed by: • [ST]_O.VIOS.VOL.PROTECTED which provides access control between VIOS SCSI device drivers and logical/physical volumes.
[OSPP-IV]_TE.MODIFY_ENVIRONMENT	The threat of violating security policies by manipulating the TOE environment is removed by: • [OSPP-IV]_OE.SECURE_OPERATION requiring the operational environment to ensure that the TSF code and TSF data (when in operation) cannot be manipulated or intercepted by entities not under the control of the TOE.

Threat	Rationale for security objectives
[ST]_TE.LPAR.ACCESS	The threat that a threat agent in a different logical partition might access resources assigned to the TOE's logical partition is removed by: • [ST]_OE.LPAR.NO_ACCESS requiring the underlying hardware to protect the resources assigned to the TOE's logical partition against access from software running in a different logical partition.

Table 8: Sufficiency of objectives countering threats

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

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

Assumption	Rationale for security objectives
[OSPP]_A.AUTHUSER	The assumption on authorized users to possess the necessary authorization to access at least some of the information managed by the TOE and to act in a cooperating manner in a benign environment is covered by: • [OSPP]_OE.ADMIN ensuring that those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of the information it contains, • [OSPP]_OE.INFO_PROTECT requiring that DAC protections on security-relevant files (such as audit trails and authentication databases) shall always be set up correctly and that users are authorized to access parts of the data maintained by the TOE.
[OSPP]_A.TRAINEDUSER	The assumptions on users to be sufficiently trained and trusted to accomplish some task or group of tasks within a secure IT environment by exercising complete control over their user data is covered by: • [OSPP]_OE.ADMIN requiring competent personnel managing the TOE, • [OSPP]_OE.INFO_PROTECT requiring that those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner and that users are trained to exercise control over their own data.
[ST]_A.VIOS.MANAGE	The assumptions on the TOE security functionality being managed by one or more trustworthy individuals is covered by: • [ST]_OE.VIOS.ADMIN requiring trustworthy personnel managing the TOE, • [ST]_OE.VIOS.INFO_PROTECT requiring personnel to ensure that information is protected in an appropriate manner, • [ST]_OE.VIOS.INSTALL requiring personnel to ensure that components that comprise the system are distributed, installed and configured in a secure manner supporting the security mechanisms provided by the TOE, • [ST]_OE.VIOS.RECOVER requiring personnel to assure that after system failure or other discontinuity, recovery without a protection (security) compromise is achieved.
[ST]_A.VIOS.AUTHUSER	The assumption on authorized users to possess the necessary authorization to access at least some of the information managed by the TOE and to act in a cooperating manner in a benign environment is covered by: • [ST]_OE.VIOS.ADMIN ensuring that those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of the information it contains, • [ST]_OE.VIOS.INFO_PROTECT requiring that DAC protections on security-relevant files (such as authentication databases) shall always be set up correctly and that users are authorized to access parts of the data maintained by the TOE.

Assumption	Rationale for security objectives
[ST]_A.VIOS.TRAINEDUSER	The assumptions on users to be sufficiently trained and trusted to accomplish some task or group of tasks within a secure operational environment by exercising complete control over their user data is covered by: • [ST]_OE.VIOS.ADMIN requiring competent personnel managing the TOE, • [ST]_OE.VIOS.INFO_PROTECT requiring that those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner and that users are trained to exercise control over their own data.
[OSPP]_A.DETECT	The assumption that modification or corruption of security-enforcing or security-relevant files will be detected by an administrative user is covered by: • [OSPP]_OE.INSTALL requiring an administrative user to ensure that the TOE is distributed, installed and configured in a secure manner supporting the security mechanisms provided by the TOE, • [OSPP]_OE.MAINTENANCE requiring an administrative user to ensure that the diagnostics facilities are invoked at every scheduled preventative maintenance period, verifying the correct operation of the TOE, • [OSPP]_OE.RECOVER requiring an administrative user to ensure that procedures and/or mechanisms are provided to assure that after system failure or other discontinuity, recovery without a protection (security) compromise is achieved.
[OSPP]_A.PEER.MGT	The assumption on all remote trusted IT systems to be under the same management control and operate under security policy constraints compatible with those of the TOE is covered by: • [OSPP]_OE.TRUSTED.IT.SYSTEM requiring that these remote trusted IT systems are under the same management domain as the TOE, and are managed based on the same rules and policies applicable to the TOE.
[OSPP]_A.PEER.FUNC	The assumption on all remote trusted IT systems to correctly implement the functionality used by the TSF consistent with the assumptions defined for this functionality is covered by: • [OSPP]_OE.TRUSTED.IT.SYSTEM requiring that the remote trusted IT systems implement the protocols and mechanisms required by the TSF to support the enforcement of the security policy.
[OSPP-IV]_A.PROTECT.INTEGRITY	The assumption on the integrity of the TSF code and TSF data loaded before the integrity verification mechanism enforces its policy is covered by: • [OSPP-IV]_OE.SECURE_LOAD demanding the proper protection of that TSF code and TSF data.

Assumption	Rationale for security objectives
[ST]_A.VIOS.DETECT	The assumption that modification or corruption of security-enforcing or security-relevant files will be detected by an administrative user is covered by: • [ST]_OE.VIOS.INSTALL requiring an administrative user to ensure that the TOE is distributed, installed and configured in a secure manner supporting the security mechanisms provided by the TOE, • [ST]_OE.VIOS.MAINTENANCE requiring an administrative user to ensure that the diagnostics facilities are invoked at every scheduled preventative maintenance period, verifying the correct operation of the TOE, • [ST]_OE.VIOS.RECOVER requiring an administrative user to ensure that procedures and/or mechanisms are provided to assure that after system failure or other discontinuity, recovery without a protection (security) compromise is achieved.
[OSPP]_A.CONNECT	The assumption on all connections to and from remote trusted IT systems and between physically separate parts of the TSF not protected by the TSF itself are physically or logically protected is covered by: • [OSPP]_OE.REMOTE requiring that remote trusted IT systems provide the functions required by the TOE and are sufficiently protected from any attack that may cause those functions to provide false results, • [OSPP]_OE.TRUSTED.IT.SYSTEM demanding the physical and logical protection equivalent to the TOE.

Table 9: Sufficiency of objectives holding assumptions

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

OSP	Rationale for security objectives
[OSPP]_P.ACCOUNTABILITY	The policy to hold users accountable for their security-relevant actions within the TOE is implemented by: • [OSPP]_O.AUDITING providing the TOE with audit functionality, • [OSPP]_O.MANAGE allowing the management of this function.
[OSPP]_P.USER	The policy to match the trust given to a user and the actions the user is given authority to perform is implemented by: • [OSPP]_O.MANAGE allowing appropriately-authorized users to manage the TSF, • [ST]_O.ROLE.HIERARCHY requiring the TOE to support hierarchical definitions of roles. • [ST]_O.ROLE.SEP_DUTY requiring the TOE to provide the capability of enforcing 'separation of duties'.

OSP	Rationale for security objectives
	• [OSPP]_OE.INFO_PROTECT, which requires that users are trusted to use the protection mechanisms of the TOE to protect their data.
[OSPP-AM]_P.APPROVE	The policy that specific rights assigned to users shall only be exercisable when approved by a second user is implemented by: • [OSPP-AM]_O.ROLE.APPROVE requiring the TOE to prevent the execution of user actions allowed by a specific right until a second user with a different right approves this action.
[OSPP-LS]_P.CLEARANCE	The policy to limit information flow between protected resources and authorized users based on whether the user's sensitivity label is appropriate for the labeled information is implemented by: • [OSPP-LS]_O.LS.CONFIDENTIALITY requiring the TOE to control information flow between entities and resources based on the sensitivity labels of users and resources. • [ST]_O.TN.ACCESS requiring the TOE to control access between the TOE and other systems based on host security attributes and the network interface on which packets are sent or received.
[OSPP-LS]_P.LABELED_OUTPUT	The policy to provide the capability to mark printed output with accurate labels based on the sensitivity label of the user causing the output is implemented by: • [OSPP-LS]_O.LS.PRINT providing the capability to mark printed output with accurate labels based on the sensitivity label of the user causing the output.
[OSPP-LS]_P.RESOURCE_LABELS	The policy that resources accessible by subjects and all subjects must have associated labels identifying the sensitivity levels of data contained therein is implemented by: • [OSPP-LS]_O.LS.LABEL providing the capability to label all subjects and all objects accessible by subjects, to restrict the information flow based on the sensitivity labels.
[OSPP-LS]_P.USER_CLEARANCE	The policy that all users must have a clearance level identifying the maximum sensitivity levels of data they may access is implemented by: • [OSPP-LS]_O.LS.CONFIDENTIALITY requiring the TOE to control information flow between entities and resources based on the sensitivity labels of users and resources. • [OSPP-LS]_O.LS.LABEL ensuring that objects and subjects can be labeled such that the TOE can restrict information flow based on those labels.
[ST]_P.DISK.OVERWRITE	The policy that the TOE provide a hard disk drive overwrite function accessible by administrators is implemented by: • [ST]_O.DISK.OVERWRITTEN requiring the TOE to offer overwriting of SCSI hard disk drives by administrators with bit patterns that prevent the recovery of the original information stored on the disk drives.

OSP	Rationale for security objectives
[ST]_P.MANDATORY_INTEGRITY	The policy that the TOE distinguish and enforce levels of integrity is implemented by: • [ST]_O.MANDATORY_INTEGRITY requiring the TOE to implement and enforce mandatory integrity controls (MIC).
[ST]_P.VIOS.USER	The policy to match the trust given to a user and the actions the user is given authority to perform is implemented by: • [ST]_O.VIOS.MANAGE allowing appropriately-authorized users to manage the VIOS TSF. • [ST]_O.VIOS.ROLE.HIERARCHY requiring the TOE to support hierarchical definitions of roles. • [ST]_O.VIOS.ROLE.SEP_DUTY requiring the TOE to provide the capability of enforcing 'separation of duties'.

Table 10: Sufficiency of objectives enforcing Organizational Security Policies

5 Extended Components Definition

5.1 Class FDP: User data protection

The Security Target defines an extended component FDP_RIP.4 as part of the FDP_RIP family in CC Part 2 for use within this ST.

5.1.1 Residual Information protection (RIP)

Component levelling

FDP_RIP.4 is not hierarchical to any other component within the FDP_RIP family and it is not hierarchical to the [OSPP] extended component definition FDP_RIP.3.

FDP_RIP.4 Hard disk drive residual information protection requires that the TSF ensure that any residual information content of a hard disk drive that is being formatted is made unavailable for logical recovery (“erased”) upon administrator-invoked de-allocation, or formatting, of the hard disk drive.

FDP_RIP.4 addresses the problem of residual information existing when a hard disk drive is moved from one system to another or when a drive is repurposed. By overwriting the sectors of a hard disk drive, the information on the drive, including formatting information, is made inaccessible from normal means of reading information from a hard disk drive including the reading of the drive as a "raw" device.

Management: FDP_RIP.4

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

a) The choice of when to erase a hard disk drive, and which hard disk drive, should be made configurable within in the TOE.

b) The choice of bit patterns used to overwrite the blocks of the hard disk drive, and how often to overwrite the blocks, could be made configurable within the TOE.

Audit: FDP_RIP.4

There are no audit events foreseen.

5.1.1.1 FDP_RIP.4 - Hard disk drive residual information protection

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_RIP.4.1 The TSF shall overwrite all data stored in current user-accessible blocks of a hard disk drive with predefined bit patterns upon request of an authorized administrator.

6 Security Requirements

6.1 TOE Security Functional Requirements

6.1.1 Access Control Policies

6.1.1.1 Compartment Access Control Policy (FDP_ACC.2(VIRT), FDP_ACF.1(VIRT))

The TSF shall enforce the Compartment Access Control Policy based on the following subject, objects and information security attributes:

Subjects

Workload Partitions.

Objects

PSO, TSO, devices, processes.

Security Attribute

Compartment ID (CID), identifying the association of a process to a compartment.

If a process has a CID = 0 (i.e. the process is in the Global Environment), then:

- The process can create, modify and delete WPARs;
- The process can assign devices to WPARs;
- The process can see processes in all WPARs via /dev/kmem;
- The process can send signals to a process in any WPAR;
- The process can access all PSO mapped from the Global Environment to the WPARs;
- The process can access all TSO;

subject to the regular DAC and MAC polices of the Global Environment.

If a process has a CID ≠ 0, then:

- The process can only send signals to other processes with the same CID value;
- The process cannot change any parameters for the kernel;
- The process cannot perform operations on system hardware unless the hardware has been assigned to the WPAR by the Global Environment administrator;
- The process can only access file system mapped to the WPAR by the Global Environment administrator.

6.1.1.2 Compartment Information Flow Control Policy (FDP_ETC.2(VIRT), FDP_IFC.2(VIRT),FDP_IFF.1(VIRT), FDP_ITC.2(VIRT))

The TSF shall enforce the Compartment Information Flow Control Policy based on the following subject, objects and information security attributes:

Subjects

Processes, external entities.

Information

Data imported and exported from the WPAR by processes.

Security Attribute

Compartment ID (CID), identifying the association of a process that handles the information to a compartment.

Processes with a CID ≠ 0 can:

1. Only communicate directly with other processes sharing the same CID based on the policies outlined in the other SFRs;
2. Communicate with other entities outside of their own WPAR only through channels mediated by the Global Environment;
3. Communicate with other entities outside of their own WPAR only through channels assigned by the Global Environment;
4. Only bind to and receive packets from the network address(es) associated with its WPAR.

6.1.2 SFR Table

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

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
AIX and Trusted AIX shared security functional requirements	FAU_GEN.1(BASE) Audit data generation [OSPP]	FAU_GEN.1	OSPP	Yes	No	Yes	No
	FAU_GEN.2 User identity association [OSPP]		OSPP	No	No	No	No
	FAU_SAR.1 Audit review [OSPP]		OSPP	No	No	Yes	No
	FAU_SAR.2 Restricted audit review [OSPP]		OSPP	No	No	No	No
	FAU_SAR.3(BASE) Selectable audit review [ST]	FAU_SAR.3	CC Part 2	Yes	No	Yes	No
	FAU_SEL.1(BASE) Selective audit [OSPP]	FAU_SEL.1	OSPP	Yes	No	Yes	No
	FAU_STG.1 Protected audit trail storage [OSPP]		OSPP	No	No	No	Yes
	FAU_STG.3 Action in case of possible audit data loss [OSPP]		OSPP	No	Yes	Yes	No
	FAU_STG.4 Prevention of audit data loss [OSPP]		OSPP	No	No	Yes	Yes

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FCS_CKM.1(SYM) Cryptographic key generation [OSPP]	FCS_CKM.1	OSPP	Yes	Yes	Yes	No
	FCS_CKM.1(RSA) Cryptographic key generation [OSPP]	FCS_CKM.1	OSPP	Yes	Yes	Yes	No
	FCS_CKM.1(DSA) Cryptographic key generation [OSPP]	FCS_CKM.1	OSPP	Yes	Yes	Yes	Yes
	FCS_CKM.2(NET) Cryptographic key distribution [OSPP]	FCS_CKM.2	OSPP	Yes	No	Yes	Yes
	FCS_CKM.4 Cryptographic key destruction [OSPP]		OSPP	No	No	No	Yes
	FCS_COP.1(NET) Cryptographic operation [OSPP]	FCS_COP.1	OSPP	Yes	Yes	Yes	Yes
	FCS_COP.1(CRYPTO-ENC) Cryptographic operation [OSPP-CRYPTO]	FCS_COP.1	OSPP-CRYPTO	Yes	Yes	Yes	No
	FCS_COP.1(CRYPTO-MD) Cryptographic operation [OSPP-CRYPTO]	FCS_COP.1	OSPP-CRYPTO	Yes	Yes	Yes	No
	FCS_COP.1(CRYPTO-SGN) Cryptographic operation [OSPP-CRYPTO]	FCS_COP.1	OSPP-CRYPTO	Yes	Yes	Yes	Yes
	FCS_COP.1(CLIC-ENC) Cryptographic operation [ST]	FCS_COP.1	CC Part 2	Yes	Yes	Yes	No
	FCS_COP.1(CLIC-MD) Cryptographic operation [ST]	FCS_COP.1	CC Part 2	Yes	Yes	Yes	No
	FCS_COP.1(CLIC-SGN) Cryptographic operation [ST]	FCS_COP.1	CC Part 2	Yes	Yes	Yes	No
	FCS_RNG.1(CLIC) Random number generation [OSPP]	FCS_RNG.1	OSPP	No	Yes	Yes	Yes
	FDP_ACC.1(PSO-AIXC) Subset access control [OSPP]	FDP_ACC.1	OSPP	Yes	Yes	Yes	No
	FDP_ACC.1(PSO-NFS) Subset access control [OSPP]	FDP_ACC.1	OSPP	Yes	Yes	Yes	No
	FDP_ACC.1(TSO) Subset access control [OSPP]	FDP_ACC.1	OSPP	Yes	No	Yes	No
	FDP_ACC.1(AUTH) Subset access control [ST]	FDP_ACC.1	CC Part 2	Yes	No	Yes	No

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FDP_ACC.1(RBAC) Subset access control [ST]	FDP_ACC.1	CC Part 2	Yes	No	Yes	No
	FDP_ACC.1(TCB) Subset access control [ST]	FDP_ACC.1	CC Part 2	Yes	No	Yes	No
	FDP_ACC.1(TCP) Subset access control [ST]	FDP_ACC.1	CC Part 2	Yes	Yes	Yes	No
	FDP_ACC.2(VIRT) Complete access control [OSPP-VIRT]	FDP_ACC.2	OSPP-VIRT	Yes	Yes	Yes	No
	FDP_ACF.1(PSO-AIXC) Security attribute based access control [OSPP]	FDP_ACF.1	OSPP	Yes	Yes	Yes	No
	FDP_ACF.1(PSO-NFS) Security attribute based access control [OSPP]	FDP_ACF.1	OSPP	Yes	Yes	Yes	No
	FDP_ACF.1(TSO) Security attribute based access control [OSPP]	FDP_ACF.1	OSPP	Yes	No	Yes	No
	FDP_ACF.1(VIRT) Complete access control [OSPP-VIRT]	FDP_ACF.1	OSPP-VIRT	Yes	No	Yes	No
	FDP_ACF.1(AUTH) Security attribute based access control [ST]	FDP_ACF.1	CC Part 2	Yes	No	Yes	No
	FDP_ACF.1(RBAC) Security attribute based access control [ST]	FDP_ACF.1	CC Part 2	Yes	No	Yes	No
	FDP_ACF.1(TCB) Security attribute based access control [ST]	FDP_ACF.1	CC Part 2	Yes	No	Yes	No
	FDP_ACF.1(TCP) Security attribute based access control [ST]	FDP_ACF.1	CC Part 2	Yes	Yes	Yes	No
	FDP_ETC.2(VIRT) Export of user data with security attributes [OSPP-VIRT]	FDP_ETC.2	OSPP-VIRT	Yes	No	Yes	No
	FDP_IFC.2(NI) Complete information flow control [OSPP]	FDP_IFC.2	OSPP	Yes	No	Yes	No
	FDP_IFC.2(VIRT) Complete information flow control [OSPP-VIRT]	FDP_IFC.2	OSPP-VIRT	Yes	No	Yes	No
	FDP_IFF.1(NI) Simple security attributes [OSPP]	FDP_IFF.1	OSPP	Yes	Yes	Yes	Yes
	FDP_IFF.1(VIRT) Simple security attributes [OSPP-VIRT]	FDP_IFF.1	OSPP-VIRT	Yes	No	Yes	No

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FDP_ITC.2(BASE) Import of user data with security attributes [OSPP]	FDP_ITC.2	OSPP	Yes	Yes	Yes	No
	FDP_ITC.2(VIRT) Import of user data with security attributes [OSPP-VIRT]	FDP_ITC.2	OSPP-VIRT	Yes	No	Yes	No
	FDP_RIP.2 Full residual information protection [OSPP]		OSPP	No	No	No	Yes
	FDP_RIP.3 Full residual information protection of resources [OSPP]		OSPP	No	No	No	Yes
	FDP_RIP.4 Hard disk drive residual information protection [ST]		ECD	No	No	No	No
	FDP_SDI.2(IV) Stored data integrity monitoring and action [OSPP-IV]	FDP_SDI.2	OSPP-IV	Yes	Yes	Yes	Yes
	FIA_AFL.1 Authentication failure handling [OSPP]		OSPP	No	Yes	Yes	Yes
	FIA_ATD.1(HU) User attribute definition [OSPP]	FIA_ATD.1	OSPP	Yes	Yes	Yes	No
	FIA_ATD.1(TU) User attribute definition [OSPP]	FIA_ATD.1	OSPP	Yes	No	Yes	No
	FIA_SOS.1(BASE) Verification of secrets [OSPP]	FIA_SOS.1	OSPP	Yes	No	No	No
	FIA_UAU.1 Timing of authentication [OSPP]		OSPP	No	No	Yes	No
	FIA_UAU.5 Multiple authentication mechanisms [OSPP]		OSPP	No	Yes	Yes	No
	FIA_UAU.7(BASE) Protected authentication feedback [OSPP]	FIA_UAU.7	OSPP	Yes	No	No	No
	FIA_UID.2(BASE) Timing of identification [OSPP]	FIA_UID.2	OSPP	Yes	No	No	No
	FIA_UID.2(VIRT) User identification before any action [OSPP-VIRT]	FIA_UID.2	OSPP-VIRT	Yes	No	No	No
	FIA_USB.2 Enhanced user-subject binding [OSPP]		OSPP	No	No	Yes	No
	FMT_MSA.1(PSO-AIXC) Management of object security attributes [OSPP]	FMT_MSA.1	OSPP	Yes	Yes	Yes	Yes
	FMT_MSA.1(PSO-NFS) Management of object security attributes [OSPP]	FMT_MSA.1	OSPP	Yes	Yes	Yes	Yes

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FMT_MSA.1(TSO) Management of object security attributes [OSPP]	FMT_MSA.1	OSPP	Yes	No	Yes	Yes
	FMT_MSA.1(VIRT-CACP) Management of security attributes [OSPP-VIRT]	FMT_MSA.1	OSPP-VIRT	Yes	No	Yes	Yes
	FMT_MSA.1(VIRT-CIFCP) Management of security attributes [OSPP-VIRT]	FMT_MSA.1	OSPP-VIRT	Yes	No	Yes	Yes
	FMT_MSA.1(AUTH) Management of object security attributes [ST]	FMT_MSA.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.1(RBAC-ADM) Management of object security attributes [ST]	FMT_MSA.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.1(RBAC-AUTH) Management of object security attributes [ST]	FMT_MSA.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.1(RBAC-DFLT) Management of object security attributes [ST]	FMT_MSA.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.1(RBAC-USR) Management of object security attributes [ST]	FMT_MSA.1	CC Part 2	Yes	Yes	Yes	Yes
	FMT_MSA.1(TCB) Management of object security attributes [ST]	FMT_MSA.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.1(TCP) Management of object security attributes [ST]	FMT_MSA.1	CC Part 2	Yes	Yes	Yes	Yes
	FMT_MSA.2(RBAC) Secure security attributes [ST]	FMT_MSA.2	CC Part 2	Yes	No	Yes	No
	FMT_MSA.3(PSO-AIXC) Static attribute initialisation [OSPP]	FMT_MSA.3	OSPP	Yes	Yes	Yes	No
	FMT_MSA.3(PSO-NFS) Static attribute initialisation [OSPP]	FMT_MSA.3	OSPP	Yes	Yes	Yes	No
	FMT_MSA.3(TSO) Static attribute initialisation [OSPP]	FMT_MSA.3	OSPP	Yes	No	Yes	No
	FMT_MSA.3(NI) Static attribute initialisation [OSPP]	FMT_MSA.3	OSPP	Yes	No	Yes	Yes
	FMT_MSA.3(VIRT-CACP) Static attribute initialisation [OSPP-VIRT]	FMT_MSA.3	OSPP-VIRT	Yes	No	Yes	No

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FMT_MSA.3(VIRT-CIFCP) Static attribute initialisation [OSPP-VIRT]	FMT_MSA.3	OSPP-VIRT	Yes	No	Yes	No
	FMT_MSA.3(AUTH) Static attribute initialisation [ST]	FMT_MSA.3	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.3(RBAC) Static attribute initialisation [ST]	FMT_MSA.3	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.3(TCB) Static attribute initialisation [ST]	FMT_MSA.3	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.3(TCP) Static attribute initialisation [ST]	FMT_MSA.3	CC Part 2	Yes	Yes	Yes	Yes
	FMT_MSA.4(PSO) Security attribute value inheritance [OSPP]	FMT_MSA.4	OSPP	Yes	No	Yes	No
	FMT_MTD.1(AE) Management of TSF data [OSPP]	FMT_MTD.1	OSPP	Yes	No	Yes	No
	FMT_MTD.1(AS) Management of TSF data [OSPP]	FMT_MTD.1	OSPP	Yes	No	Yes	Yes
	FMT_MTD.1(AT) Management of TSF data [OSPP]	FMT_MTD.1	OSPP	Yes	No	Yes	Yes
	FMT_MTD.1(AF) Management of TSF data [OSPP]	FMT_MTD.1	OSPP	Yes	No	Yes	Yes
	FMT_MTD.1(NI) Management of TSF data [OSPP]	FMT_MTD.1	OSPP	Yes	No	Yes	Yes
	FMT_MTD.1(IAT) Management of TSF data [OSPP]	FMT_MTD.1	OSPP	Yes	No	Yes	No
	FMT_MTD.1(IAF) Management of TSF data [OSPP]	FMT_MTD.1	OSPP	Yes	No	Yes	No
	FMT_MTD.1(IAU) Management of TSF data [OSPP]	FMT_MTD.1	OSPP	Yes	No	Yes	No
	FMT_MTD.1(AM-AP) Management of TSF data [OSPP-AM]	FMT_MTD.1	OSPP-AM	Yes	No	Yes	Yes
	FMT_MTD.1(AM-MR) Management of TSF data [OSPP-AM]	FMT_MTD.1	OSPP-AM	Yes	No	Yes	Yes
	FMT_MTD.1(AM-MD) Management of TSF data [OSPP-AM]	FMT_MTD.1	OSPP-AM	Yes	No	Yes	Yes
	FMT_MTD.1(AM-MA) Management of TSF data [OSPP-AM]	FMT_MTD.1	OSPP-AM	Yes	Yes	Yes	Yes

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FMT_MTD.1(IV-ACT) Management of TSF data [OSPP-IV]	FMT_MTD.1	OSPP-IV	Yes	Yes	Yes	Yes
	FMT_MTD.1(IV-TSF) Management of TSF data [OSPP-IV]	FMT_MTD.1	OSPP-IV	Yes	No	Yes	Yes
	FMT_MTD.1(IV-USR) Management of TSF data [OSPP-IV]	FMT_MTD.1	OSPP-IV	Yes	No	Yes	Yes
	FMT_MTD.1(VIRT-COMP) Management of TSF data [OSPP-VIRT]	FMT_MTD.1	OSPP-VIRT	Yes	No	Yes	No
	FMT_MTD.1(PRIVS) Management of TSF data [ST]	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MTD.1(RBAC) Management of TSF data [ST]	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MTD.3(RBAC) Secure TSF data [ST]	FMT_MTD.3	CC Part 2	Yes	No	Yes	No
	FMT_REV.1(OBJ) Revocation [OSPP]	FMT_REV.1	OSPP	Yes	No	Yes	No
	FMT_REV.1(USR) Revocation [OSPP]	FMT_REV.1	OSPP	Yes	No	Yes	No
	FMT_SMF.1(BASE) Specification of management functions [OSPP]	FMT_SMF.1	OSPP	Yes	No	Yes	No
	FMT_SMR.2 Security roles [OSPP]		OSPP	No	No	Yes	No
	FPT_FLS.1(RBAC) Failure with preservation of secure state [ST]	FPT_FLS.1	CC Part 2	Yes	No	Yes	No
	FPT_FLS.1(SED) Failure with preservation of secure state [ST]	FPT_FLS.1	CC Part 2	Yes	No	Yes	No
	FPT_RCV.1 Manual recovery [ST]		CC Part 2	No	No	Yes	No
	FPT_RCV.4 Function recovery [ST]		CC Part 2	No	No	Yes	No
	FPT_STM.1 Reliable time stamps [OSPP]		OSPP	No	No	No	No
	FPT_TDC.1(BASE) Inter-TSF basic TSF data consistency [OSPP]	FPT_TDC.1	OSPP	Yes	Yes	Yes	No
	FPT_TDC.1(VIRT) Inter-TSF basic TSF data consistency [OSPP-VIRT]	FPT_TDC.1	OSPP-VIRT	Yes	No	Yes	No
	FPT_TIM.1(IV) TSF integrity monitoring and action [OSPP-IV]	FPT_TIM.1	OSPP-IV	Yes	Yes	Yes	Yes

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FPT_TST.1 TSF testing [ST]		CC Part 2	No	No	Yes	Yes
	FRU_FLT.2 Limited fault tolerance [ST]		CC Part 2	No	No	Yes	No
	FTA_LSA.1(RBAC) Limitation on scope of selectable attributes [ST]	FTA_LSA.1	CC Part 2	Yes	No	Yes	No
	FTA_SSL.1 TSF-initiated session locking [OSPP]		OSPP	No	No	Yes	No
	FTA_SSL.2 User-initiated locking [OSPP]		OSPP	No	No	Yes	No
	FTA_TSE.1(RBAC) TOE session establishment [ST]	FTA_TSE.1	CC Part 2	Yes	No	Yes	No
	FTP_ITC.1 Inter-TSF trusted channel [OSPP]		OSPP	No	No	Yes	Yes
Additional Trusted AIX security functional requirements (i.e., LAS mode only)	FAU_GEN.1(LS) Audit data generation [ST] (LAS mode only)	FAU_GEN.1	CC Part 2	Yes	Yes	Yes	No
	FAU_SAR.3(LS) Selectable audit review [ST] (LAS mode only)	FAU_SAR.3	CC Part 2	Yes	Yes	Yes	No
	FAU_SEL.1(LS) Selective audit [OSPP] (LAS mode only)	FAU_SEL.1	OSPP	Yes	Yes	Yes	No
	FDP_ETC.2(LS) Export of user data with security attributes [OSPP-LS] (LAS mode only)	FDP_ETC.2	OSPP-LS	Yes	Yes	Yes	No
	FDP_IFC.1(MIC) Subset information flow control [ST] (LAS mode only)	FDP_IFC.1	CC Part 2	Yes	Yes	Yes	No
	FDP_IFC.1(TN) Subset information flow control [ST] (LAS mode only)	FDP_IFC.1	CC Part 2	Yes	Yes	Yes	No
	FDP_IFC.2(LS) Complete information flow control [OSPP-LS] (LAS mode only)	FDP_IFC.2	OSPP-LS	Yes	Yes	Yes	No
	FDP_IFF.2(MIC) Hierarchical security attributes [ST] (LAS mode only)	FDP_IFF.2	CC Part 2	Yes	Yes	Yes	No
	FDP_IFF.2(TN) Hierarchical security attributes [ST] (LAS mode only)	FDP_IFF.2	CC Part 2	Yes	Yes	Yes	No
	FDP_IFF.2(LS) Hierarchical security attributes [OSPP-LS] (LAS mode only)	FDP_IFF.2	OSPP-LS	Yes	Yes	Yes	No

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FDP_ITC.1(LS) Import of user data without security attributes [OSPP-LS] (LAS mode only)	FDP_ITC.1	OSPP-LS	Yes	Yes	Yes	No
	FDP_ITC.2(LS) Import of user data with security attributes [OSPP-LS] (LAS mode only)	FDP_ITC.2	OSPP-LS	Yes	Yes	Yes	No
	FIA_ATD.1(LS) User attribute definition [OSPP-LS] (LAS mode only)	FIA_ATD.1	OSPP-LS	Yes	Yes	No	No
	FIA_ATD.1(LSX) User attribute definition [ST] (LAS mode only)	FIA_ATD.1	CC Part 2	Yes	Yes	Yes	No
	FIA_USB.1(LS) User-subject binding [OSPP-LS] (LAS mode only)	FIA_USB.1	OSPP-LS	Yes	Yes	Yes	No
	FIA_USB.1(LSX) User-subject binding [ST] (LAS mode only)	FIA_USB.1	CC Part 2	Yes	Yes	Yes	No
	FMT_MSA.1(LS) Management of security attributes [OSPP-LS] (LAS mode only)	FMT_MSA.1	OSPP-LS	Yes	Yes	Yes	No
	FMT_MSA.1(MIC) Management of security attributes [ST] (LAS mode only)	FMT_MSA.1	CC Part 2	Yes	Yes	Yes	Yes
	FMT_MSA.1(TN) Management of security attributes [ST] (LAS mode only)	FMT_MSA.1	CC Part 2	Yes	Yes	Yes	Yes
	FMT_MSA.3(LS) Static attribute initialisation [OSPP-LS] (LAS mode only)	FMT_MSA.3	OSPP-LS	Yes	Yes	Yes	No
	FMT_MSA.3(MIC) Static attribute initialisation [ST] (LAS mode only)	FMT_MSA.3	CC Part 2	Yes	Yes	Yes	Yes
	FMT_MSA.3(TN) Static attribute initialisation [ST] (LAS mode only)	FMT_MSA.3	CC Part 2	Yes	Yes	Yes	Yes
	FPT_TDC.1(LS) Inter-TSF basic TSF data consistency [OSPP-LS] (LAS mode only)	FPT_TDC.1	OSPP-LS	Yes	Yes	Yes	No
VIOS security functional requirements	FDP_ACC.1(VIOS) Subset access control [ST] (VIOS only)	FDP_ACC.1	CC Part 2	Yes	Yes	Yes	No
	FDP_ACC.1(VRBAC) Subset access control [ST] (VIOS only)	FDP_ACC.1	CC Part 2	Yes	No	Yes	No

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FDP_ACF.1(VIOS) Security attribute based access control [ST] (VIOS only)	FDP_ACF.1	CC Part 2	Yes	Yes	Yes	No
	FDP_ACF.1(VRBAC) Security attribute based access control [ST] (VIOS only)	FDP_ACF.1	CC Part 2	Yes	No	Yes	No
	FIA_ATD.1(VIOS) User attribute definition [ST] (VIOS only)	FIA_ATD.1	CC Part 2	Yes	Yes	Yes	No
	FIA_SOS.1(VIOS) Verification of secrets [ST] (VIOS only)	FIA_SOS.1	CC Part 2	Yes	Yes	Yes	No
	FIA_UAU.2 User authentication before any action [ST] (VIOS only)		CC Part 2	No	Yes	No	No
	FIA_UAU.7(VIOS) Protected authentication feedback [ST] (VIOS only)	FIA_UAU.7	CC Part 2	Yes	Yes	Yes	No
	FIA_UID.2(VIOS) User identification before any action [ST] (VIOS only)	FIA_UID.2	CC Part 2	Yes	Yes	No	No
	FIA_USB.1(VIOS) User-subject binding [ST] (VIOS only)	FIA_USB.1	CC Part 2	Yes	Yes	Yes	No
	FMT_MSA.1(VIOS) Management of security attributes [ST] (VIOS only)	FMT_MSA.1	CC Part 2	Yes	Yes	Yes	Yes
	FMT_MSA.1(VRBAC-ADM) Management of object security attributes [ST] (VIOS only)	FMT_MSA.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.1(VRBAC-AUTH) Management of object security attributes [ST] (VIOS only)	FMT_MSA.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.1(VRBAC-DFLT) Management of object security attributes [ST] (VIOS only)	FMT_MSA.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.1(VRBAC-USR) Management of object security attributes [ST] (VIOS only)	FMT_MSA.1	CC Part 2	Yes	Yes	Yes	Yes
	FMT_MSA.2(VRBAC) Secure security attributes [ST] (VIOS only)	FMT_MSA.2	CC Part 2	Yes	No	Yes	No
	FMT_MSA.3(VIOS) Static attribute initialisation [ST] (VIOS only)	FMT_MSA.3	CC Part 2	Yes	Yes	Yes	Yes

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FMT_MSA.3(VRBAC) Static attribute initialisation [ST] (VIOS only)	FMT_MSA.3	CC Part 2	Yes	No	Yes	Yes
	FMT_MTD.1(VIOS-ADI) Management of TSF data [ST] (VIOS only)	FMT_MTD.1	CC Part 2	Yes	Yes	Yes	Yes
	FMT_MTD.1(VIOS-ADM) Management of TSF data [ST] (VIOS only)	FMT_MTD.1	CC Part 2	Yes	Yes	Yes	Yes
	FMT_MTD.1(VIOS-NV) Management of TSF data [ST] (VIOS only)	FMT_MTD.1	CC Part 2	Yes	Yes	Yes	Yes
	FMT_MTD.1(VIOS-SA) Management of TSF data [ST] (VIOS only)	FMT_MTD.1	CC Part 2	Yes	Yes	Yes	Yes
	FMT_MTD.1(VRBAC) Management of TSF data [ST] (VIOS only)	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MTD.3(VRBAC) Secure TSF data [ST] (VIOS only)	FMT_MTD.3	CC Part 2	Yes	No	Yes	No
	FMT_REV.1(VIOS) Revocation [ST] (VIOS only)	FMT_REV.1	CC Part 2	Yes	Yes	Yes	Yes
	FMT_SMF.1(VIOS) Specification of management functions [ST] (VIOS only)	FMT_SMF.1	CC Part 2	Yes	Yes	Yes	No
	FMT_SMR.1 Security roles [ST] (VIOS only)		CC Part 2	No	Yes	Yes	No
	FTA_LSA.1(VRBAC) Limitation on scope of selectable attributes [ST] (VIOS only)	FTA_LSA.1	CC Part 2	Yes	No	Yes	No
	FTA_TSE.1(VRBAC) TOE session establishment [ST] (VIOS only)	FTA_TSE.1	CC Part 2	Yes	No	Yes	No

Table 11: Security functional requirements for the TOE

6.1.3 AIX and Trusted AIX shared security functional requirements

This section contains SFRs that are common to both AIX and Trusted AIX (i.e., SFRs supported in both BAS mode and LAS mode) except where otherwise specified.

6.1.3.1 Audit data generation [OSPP] (FAU_GEN.1(BASE))

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

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

b) All auditable events for the basic level of audit; and
c) all modifications to the set of events being audited;
d) all user authentication attempts;
e) all denied accesses to objects for which the access control policy defined in the OSPP base applies;
f) explicit modifications of access rights to objects covered by the access control policies; and
g) **assignment of users, roles, and privileges to roles;**
h) **deletion of users, roles, and privileges from roles;**
i) **creation and deletion of roles.**

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:
a) Date and time of the event, type of event, subject identity (if applicable), and outcome of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST;
   i. User identity (if applicable); and
   ii. **For each invocation of a security function, the RBAC Administrator role that made invocation of that security function possible;**
   iii. **For each access control action on the user data, the role that made possible the invocation of that action.**

6.1.3.2 User identity association [OSPP] (FAU_GEN.2)

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

6.1.3.3 Audit review [OSPP] (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide **the set of authorized RBAC administrators** with the capability to read
a) Date and time of the audit event;
b) The user ID responsible for the event and optionally the role membership which enabled the user to perform the event successfully;
c) The access control operation and the object on which it was performed;
d) The outcome of the event (success or failure);
e) The user session identifier or terminal type
from the audit records.

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

6.1.3.4 Restricted audit review [OSPP] (FAU_SAR.2)

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

6.1.3.5 Selectable audit review [ST] (FAU_SAR.3(BASE))

FAU_SAR.3.1 The TSF shall provide the ability to apply **searches, sorting, and ordering** of audit data based on

a) Date and time of audit event;
b) User identity;
c) Object name and type of access;
d) Role that enabled the access;
e) Audit event;
f) DAC success or failures;
g) Privilege success or failures;
h) FSF success or failure;
i) AUTH success;
j) Event status;
k) Command name;
l) Process ID;
m) Parent process ID;
n) Kernel thread ID.

6.1.3.6 Selective audit [OSPP] (FAU_SEL.1(BASE))

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

a) Type of audit event;
b) Subject or user identity;
c) Outcome (success or failure) of the audit event;
d) Named object identity;
e) **Host identity;**
f) **Users belonging to a specified role;**
g) **Access types on a particular object.**

6.1.3.7 Protected audit trail storage [OSPP] (FAU_STG.1)

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

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

6.1.3.8 Action in case of possible audit data loss [OSPP] (FAU_STG.3)

FAU_STG.3.1 The TSF shall **generate an alarm to the authorized administrator** if the ~~audit trail exceeds~~ *file system holding the audit trail falls below* **an administrator configurable limit of free blocks** or if any of the following **(no additional conditions)** is detected that may result in a loss of audit records.

6.1.3.9 Prevention of audit data loss [OSPP] (FAU_STG.4)

FAU_STG.4.1 The TSF shall **“prevent audited events, except those taken by the authorized administrator"** and **either stop the system in panic mode or count the number of audit records lost** if the audit trail is full.

6.1.3.10 Cryptographic key generation [OSPP] (FCS_CKM.1(SYM))

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

a) 128 bits *(AES)*,

b) 168 bits *(TDES)*,

c) 256 bits *(AES)*,

d) **192 bits (AES)**

that meet the following:

a) **IPsec using CLiC: compliant with [RFC4301] section 4.5,**

b) **EFS using CLiC for vendor-specific key generation,**

c) **ACF using CLiC for application-specific key generation**.

Application Note: *The SFR statement assumes the random bit generator to apply a "pessimistic" estimate of the entropy in its entropy pool. The requirement of the SFR concerning the entropy specifies that even with the pessimistic entropy estimate, the number of bits extracted from the entropy pool is less than the estimated entropy in the pool. See FCS_RNG.1(CLIC) for details on the CLiC random number generator.*

Application Note: *This SFR applies to IPsec, ACF, and EFS which use CLiC. EFS is an IBM proprietary implementation and, therefore, not listed as a standard in this SFR. ACF provides a general purpose CLiC cryptographic API, including random number generation, which can be used for symmetric key generation as per the application's needs.*

6.1.3.11 Cryptographic key generation [OSPP] (FCS_CKM.1(RSA))

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

a) 2048 bits,

b) **1024 bits**

that meet the following:

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

b) **No additional standards**.

Application Note: *This SFR applies to IPSec and EFS which use CLiC.*

6.1.3.12 Cryptographic key generation [OSPP] (FCS_CKM.1(DSA))

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

a) L=1024, N=160 bits;

that meet the following:

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

b) **No additional standards**.

Application Note: *This SFR applies to IPsec which uses CLiC.*

6.1.3.13 Cryptographic key distribution [OSPP] (FCS_CKM.2(NET))

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

a) Diffie-Hellman key agreement method defined for the IKE protocol by RFC2409;

b) Diffie-Hellman key agreement method defined for the IKE protocol by RFC4306;

c) Kerberos Version 5 (NAS) defined by [RFC4120].

Application Note: *This SFR applies to IPsec which uses CLiC. This SFR applies to the NAS (i.e., Kerberos) client when used with NFSv4 which uses the NAS cryptographic library.*

6.1.3.14 Cryptographic key destruction [OSPP] (FCS_CKM.4)

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

6.1.3.15 Cryptographic operation [OSPP] (FCS_COP.1(NET))

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

a) IPSEC with IKE allowing the use of Suite-B-GCM-128, Suite-B-GCM-192, Suite-B-GCM-256, Suite-B-GMAC-128, Suite-B-GMAC-192, and Suite-B-GMAC-256 defined by [RFC4869];

b) Kerberos Version 5 GSS-API allowing the use of TDES in CBC mode with 168 bits key size and SHA-1 defined by [RFC4120] section 8.1 and refined by [RFC3961] section 6.3;

c) Kerberos Version 5 GSS-API allowing the use of AES in CTS mode with 128 bits and 256 bits key sizes and SHA-1 defined by [RFC4120] section 8.1 and refined by [RFC3962].

Application Note: *IPsec uses CLiC for its cryptographic functionality. Kerberos Version 5 GSS-API uses CLiC when communicating between the NFSv4 client and NFSv4 server.*

6.1.3.16 Cryptographic operation [OSPP-CRYPTO] (FCS_COP.1(CRYPTO-ENC))

FCS_COP.1.1 The *AIX Cryptographic Framework* TSF*, when using the default CLiC module,* shall perform encryption and decryption in accordance with the following specified cryptographic algorithms, cryptographic key sizes and applicable standards:

a) TDES with the following block chaining modes:
   i. CBC,
   ii. **CTR**

   with 168 bits key size as defined by NIST Special Publication 800-67;

b) AES *with* the following block chaining modes:
   i. CBC,
   ii. **CCM,**
   iii. **CTR,**
   iv. **CTS (defined in [SP800-38A],**
   v. **GCM (defined in [SP800-38D])**

   and with the following key sizes:
   i. 128 bits,
   ii. 256 bits,
   iii. **192 bits**,

   as defined by FIPS PUB 197;

c) RSA with the following key sizes:
   i. 2048 bits,
   ii. **1024 bits**,

   as defined by PKCS #1 v1.5;

d) **No additional algorithms**.

Application Note: *The CCM block chaining mode applies to generation-encryption and decryption-verification.*

6.1.3.17 Cryptographic operation [OSPP-CRYPTO] (FCS_COP.1(CRYPTO-MD))

FCS_COP.1.1 The *AIX Cryptographic Framework* TSF*, when using the default CLiC module,* shall perform message digest generation in accordance with the following cryptographic algorithms, using no cryptographic keys and in accordance with the following applicable standards:

a) SHA-1 as defined by FIPS PUB 180-3;
b) SHA-256 as defined by FIPS PUB 180-3;
c) SHA-512 as defined by FIPS PUB 180-3;
d) **SHA-224 as defined by FIPS PUB 180-3;**
e) **SHA-384 as defined by FIPS PUB 180-3.**

6.1.3.18 Cryptographic operation [OSPP-CRYPTO] (FCS_COP.1(CRYPTO-SGN))

FCS_COP.1.1 The *AIX Cryptographic Framework* TSF*, when using the default CLiC module,* shall perform signature generation and verification in accordance with the following cryptographic algorithms, cryptographic key sizes and applicable standards:

a) DSA with the following key sizes:

i. L=1024, N=160 bits,

as defined by FIPS PUB 186-3;

b) RSA with the following key sizes:

i. 2048 bits,

ii. **1024 bits**

as defined by FIPS PUB 186-3;

c) **No additional algorithms**.

6.1.3.19 Cryptographic operation [ST] (FCS_COP.1(CLIC-ENC))

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

- **TDES with block chaining modes: CBC and CTR,**
- **AES with block chaining modes: CBC, CCM, CTR, CTS (defined in [SP800-38A]), and GCM (defined in [SP800-38D],**
- **RSA**

and cryptographic key sizes

- **TDES: 168 bits**
- **AES: 128 bits, 192 bits, and 256 bits,**
- **RSA: 1024 bits and 2048 bits**

that meet the following:

- **TDES: [SP800-67],**
- **AES: [FIPS197],**
- **RSA: [PKCS1].**

Application Note: *CLiC is used by EFS, IPsec, and NFSv4 communications.*

6.1.3.20 Cryptographic operation [ST] (FCS_COP.1(CLIC-MD))

FCS_COP.1.1 The *CLiC* TSF shall perform **message digest generation** in accordance with a specified cryptographic algorithm

- **SHA-1,**
- **SHA-224,**
- **SHA-256,**
- **SHA-384,**
- **SHA-512**

and cryptographic key sizes **none** that meet the following: **[FIPS180-3]**.

Application Note: *CLiC is used by FIV/TE, IPsec, and NFSv4 communications.*

6.1.3.21 Cryptographic operation [ST] (FCS_COP.1(CLIC-SGN))

FCS_COP.1.1 The *CLiC* TSF shall perform **signature generation and verification** in accordance with a specified cryptographic algorithm

- **DSA,**
- **RSA**

and cryptographic key sizes

- **DSA: L=1024, N=160 bits,**
- **RSA: 1024 bits and 2048 bits**

that meet the following:

- **DSA: [FIPS186-3],**
- **RSA: [FIPS186-3].**

Application Note: *CLiC is used by EFS, FIV/TE, and IPsec.*

6.1.3.22 Random number generation [OSPP] (FCS_RNG.1(CLIC))

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

a) If initialized with a random seed using /dev/random, an NPTRNG conformant with [BSI-AIS20] class DRG.2, as the random source, the internal state of the RNG shall have a minimum entropy of 48 bits;

b) The RNG provides forward secrecy;

c) The RNG provides backward secrecy.

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

a) The RNG initialized with a random seed holding 160 bits of entropy with string lengths varying from 1 to 160 bits where the number of strings is derived by subtracting the bit lengths of the requested strings from 1024 bits;

b) The [BSI-AIS20] test suite A cannot distinguish the random numbers from output sequences of ideal RNGs.

6.1.3.23 Subset access control [OSPP] (FDP_ACC.1(PSO-AIXC))

FDP_ACC.1.1 The TSF shall enforce the *AIXC* Persistent Storage Object Access Control Policy on

a) **Subjects:**

i. Processes acting on behalf of users;

b) Objects:

i. Persistent Storage Objects of the following type

1. Ordinary files;

2. Directories;

3. Device special files;

4. UNIX Domain socket special files;

5. Named pipes;

ii. **No additional storage objects**;

c) Operations:

The following operations apply to both permission bits and extended permissions. Permission bits are the standard UNIX permission bits for user, group, and world. Extended permissions can be used to allow or deny access to the granularity of a single user or group using access control lists (ACLs).

i. Read;
ii. Write;
iii. Execute (ordinary files only);
iv. Search (directories only).

6.1.3.24 Subset access control [OSPP] (FDP_ACC.1(PSO-NFS))

FDP_ACC.1.1 The TSF shall enforce the *NFS* Persistent Storage Object Access Control Policy on

a) **Subjects:**

i. Processes acting on behalf of users;

b) Objects:

i. Persistent Storage Objects of the following type

1. Ordinary files;
2. Directories;
3. Device special files;
4. UNIX Domain socket special files;
5. Named pipes;

ii. **No additional storage objects**;

c) Operations:

The following operations apply to NFSv4 fine grained permissions. These are the fine grained permissions that apply to the entities: owner, group, and everyone. They can be used to allow or deny access to the granularity of a single entity.

i. Read data (ordinary files, device special files);
ii. List contents (directories);
iii. Write file data (ordinary files, device special files);
iv. Add a file (directories);
v. Append data (ordinary files, device special files);
vi. Add subdirectory (directories);
vii. Read extended attributes;
viii. Write extended attributes;
ix. Execute (ordinary files);
x. Search (directories);
xi. Delete an object within a directory;

xii. Delete the associated object;
xiii. Read core object attributes (size, time, etc.);
xiv. Write core object attributes;
xv. Read ACL contents;
xvi. Write ACL contents;
xvii. Change ownership (user or group);
xviii. Synchronize.

6.1.3.25 Subset access control [OSPP] (FDP_ACC.1(TSO))

FDP_ACC.1.1 The TSF shall enforce the Transient Storage Object Access Control Policy on

a) **Subjects:**
   i. Processes acting on behalf of users;
b) Objects:
   i. Transient Storage Objects of the following type
      1. Message queues;
      2. SysV semaphores;
      3. Shared memory segments;
   ii. **No additional storage objects**;
c) Operations:

The following operations apply to permission bits

i. Read;
ii. Write.

6.1.3.26 Subset access control [ST] (FDP_ACC.1(AUTH))

FDP_ACC.1.1 The TSF shall enforce the **Authorization Policy** on

a) Subjects: Processes acting on the behalf of users;
b) Objects: Functions implemented in executable files;
c) Operations: Attempts of processes to invoke such functions.

6.1.3.27 Subset access control [ST] (FDP_ACC.1(RBAC))

FDP_ACC.1.1 The TSF shall enforce the **Role-based Access Control (RBAC) Policy** on

a) Subjects: Processes acting on the behalf of users;
b) Objects:
   i. Persistent Storage Objects of the following type
      1. Ordinary files;
      2. Directories;
      3. Device special files;
      4. UNIX Domain socket special files;
      5. Named pipes;

ii. Transient Storage Objects of the following type

i. Message queues;

ii. SysV semaphores;

iii. Shared memory segments;

iv. TCP ports;

c) Operations: All operations among subjects and objects covered by this policy.

6.1.3.28 Subset access control [ST] (FDP_ACC.1(TCB))

FDP_ACC.1.1 The TSF shall enforce the **Trusted Computing Base (TCB) Policy** on

a) Subjects: Processes acting on the behalf of users;

b) Objects: jfs2 file system objects (ordinary files, directories, device special files, UNIX Domain socket special files, named pipes);

c) Operations: All operations among subjects and objects covered by the SFP.

6.1.3.29 Subset access control [ST] (FDP_ACC.1(TCP))

FDP_ACC.1.1 *In BAS mode, the* ~~The~~ TSF shall enforce the **TCP Port Access Control Policy** on

a) Subjects:

i. Processes acting on behalf of users;

b) Objects:

i. TCP ports;

c) Operations:

i. Connection establishment.

6.1.3.30 Complete access control [OSPP-VIRT] (FDP_ACC.2(VIRT))

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

a) Subjects: compartments *(a.k.a. Workload Partitions (WPARs))*;

b) Objects:

i. Persistent Storage Objects of **FDP_ACC.1(PSO-AIXC) and FDP_ACC.1(PSO-NFS)**;

ii. Transient Storage Objects of **FDP_ACC.1(TSO)**;

iii. **TCP Ports of FDP_ACC.1(TCP)**

iv. **Processes and devices**

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

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

6.1.3.31 Security attribute based access control [OSPP] (FDP_ACF.1(PSO-AIXC))

FDP_ACF.1.1 The TSF shall enforce the *AIXC* Persistent Storage Object Access Control Policy to objects based on the following:

- **a) Subjects: defined in FDP_ACC.1(PSO-AIXC);**
- **b) Objects: defined in FDP_ACC.1(PSO-AIXC);**
- **c) Security attributes:**
  - **i. Permission bits;**
  - **ii. Extended permissions;**
  - **iii. Privilege sets;**
  - **iv. File encryption attributes.**

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

A subject must have search permission for every element of the pathname and the requested access for the object. A subject has a specific type access to an object if the type of access is within the union of all permission rights (grant entries) defined in the access control list of the object for the subject and is not within the logical union of all restrictions (deny entries) defined in the access control list of the object for the subject. If no entry in the extended permissions either allows or denies access, the access right defined in the permission bits apply. In any other case access is denied.

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

Privilege sets override Persistent Storage Object decisions (a.k.a. discretionary access control (DAC)) in order to allow a process to:

- **a) Change its GID (PV_DAC_GID);**
- **b) Bypass DAC ownership restrictions (PV_DAC_O);**
- **c) Bypass DAC read restrictions (PV_DAC_R);**
- **d) Change its role ID (PV_DAC_RID);**
- **e) Change its UID (PV_DAC_UID);**
- **f) Bypass DAC write restrictions (PV_DAC_W);**
- **g) Bypass DAC execute/search restrictions (PV_DAC_X);**
- **h) Obtain all of the privileges listed in a) through g) (PV_DAC).**

The process has the appropriate privileges to perform the specific access request.

The attributes for a privileged command (an aspect of role-based access control) defined in the Privileged Commands database (privcmds) override the file system DAC attributes.

BAS mode only: A process with a user ID of 0 is known as a root user process. These processes are generally allowed all access permissions. But if a root user process requests execute permission for a program (as a file system object), access is granted only if execute permission is granted to at least one user.

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

a) Write requests to objects that reside on a file system that is mounted read-only;

b) Inability of a subject to decrypt a file encrypted by the EFS file system.

6.1.3.32 Security attribute based access control [OSPP] (FDP_ACF.1(PSO-NFS))

FDP_ACF.1.1 The TSF shall enforce the *NFS* Persistent Storage Object Access Control Policy to objects based on the following:

a) Subjects: defined in FDP_ACC.1(PSO-NFS);

b) Objects: defined in FDP_ACC.1(PSO-NFS);

c) Security attributes:

i. Permission bits;

ii. Fine grained permissions;

iii. Privilege sets;

iv. File encryption attributes.

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

A subject must have search permission for every element of the pathname and the requested access for the object. A subject has the requested type access to an object if all requested access types are specifically allowed before reaching an entry that denies one or more requested types or before reaching the end of the ACL. Otherwise, the requested access is denied.

BAS mode only: A subject with an effective UID other than 0 and with WRITE_OWNER access specified in the ACL can change the object owner to himself, otherwise the request is denied.

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

Privilege sets override Persistent Storage Object decisions (a.k.a. discretionary access control (DAC)) in order to allow a process to:

a) Change its GID (PV_DAC_GID);

b) Bypass DAC ownership restrictions (PV_DAC_O);

c) Bypass DAC read restrictions (PV_DAC_R);

d) Change its role ID (PV_DAC_RID);

e) Change its UID (PV_DAC_UID);

f) Bypass DAC write restrictions (PV_DAC_W);

g) Bypass DAC execute/search restrictions (PV_DAC_X);

h) Obtain all of the privileges listed in a) through g) (PV_DAC).

The process has the appropriate privileges to perform the specific access request. The object owner is always allowed to read/write the ACL contents and read/write the core object attributes.

The attributes for a privileged command (an aspect of role-based access control) defined in the Privileged Commands database (privcmds) override the file system DAC attributes.

BAS mode only: A process with a user ID of 0 is known as a root user process. These processes are generally allowed all access permissions. But if a root user process requests execute permission for a program (as a file system object), access is granted only if execute permission is granted to at least one user. The object owner is always allowed to read/write the ACL contents and read/write the core object attributes.

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

a) Write requests to objects that reside on a file system that is mounted read-only;

b) Inability of a subject to decrypt a file encrypted by the EFS file system.

6.1.3.33 Security attribute based access control [OSPP] (FDP_ACF.1(TSO))

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

a) Subjects: defined in FDP_ACC.1(TSO);

b) Objects: defined in FDP_ACC.1(TSO);

c) Security attributes:

i. Permission bits;

ii. Privilege sets.

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

Access permissions are defined by permission bits of the IPC object. The process creating the object defines the creator, owner and group based on the user ID of the current process. Access of a process to an IPC object is allowed, if:

a) The user ID of the current process is equal to the user ID of the IPC object creator or owner and the "owner" permission bit for the requested type of access is set; or

b) The group ID of the current process is equal to the group ID of the IPC object and the "group" permission bit for the requested type of access is set; or

c) The "world" permission bit for the requested type of access is set.

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

Privilege sets override Transient Storage Object decisions (a.k.a. discretionary access control (DAC)) in order to allow a process to:

a) Change its GID (PV_DAC_GID);
b) Change its role ID (PV_DAC_RID);
c) Change its UID (PV_DAC_UID);
d) Obtain all of the privileges listed in a) through c) (PV_DAC);
e) Bypass DAC ownership restrictions on IPC objects (PV_KER_IPC_O);
f) Bypass DAC read restrictions on IPC objects (PV_KER_IPC_R);
g) Bypass DAC write restrictions on IPC objects (PV_KER_IPC_W).

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

6.1.3.34 Complete access control [OSPP-VIRT] (FDP_ACF.1(VIRT))

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

a) Subject security attributes:
   i. **Corral ID (CID)**;
   ii. **No additional attributes**;
b) Object security attributes: **none**;
c) **No additional attributes**.

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

a) Access of a compartment to an object is allowed when the requested mode of access is allowed for the compartment by the compartment access control permission settings for that object;
b) **The rules for CID ≠ 0 as defined by the Compartment Access Control Policy in section 6.1.1.1**.

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **The rules for CID = 0 as defined by the Compartment Access Control Policy in section 6.1.1.1**.

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

6.1.3.35 Security attribute based access control [ST] (FDP_ACF.1(AUTH))

FDP_ACF.1.1 The TSF shall enforce the **Authorization Policy** to objects based on the following:

a) Subjects: Processes acting on the behalf of users;
   i. Attributes: User identity, authorizations;
b) Objects: Functions implemented in executable files;
   i. Attributes: Privileges associated with a function.

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

a) **If a function within an executable requires a specific authorization, the function will only be executed if the executing user has been assigned the proper authorization.**

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

Privilege sets override authorization decisions in order to allow a process to:

a) **Modify the kernel security table (PV_AZ_ADMIN);**
b) **Bypass all authorization checks (PV_AZ_CHECK);**
c) **Retrieve the entire kernel security table (PV_AZ_READ);**
d) **Pass all authorization checks during exec() (PV_AZ_ROOT).**

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

6.1.3.36 Security attribute based access control [ST] (FDP_ACF.1(RBAC))

FDP_ACF.1.1 The TSF shall enforce the **Role-based Access Control (RBAC) Policy** to objects based on the following:

a) **Subjects: Processes acting on the behalf of users;**
    i. **Attributes:**
        1. **Subject identity;**
        2. **Role(s) which can invoke the subject;**
        3. **Authorized role(s) for the user;**
b) **Objects: As defined in FDP_ACC.1(RBAC);**
    i. **Attributes:**
        1. **Object identity;**
        2. **Operations permitted on the objects for various roles.**

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

a) **The subject invoking the operation on an object is assigned to a role whose privilege set includes the operation on the object.**

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

a) **Allow an access operation by a subject on an object only if the user associated with the subject belongs to a role that permits the access operation on the object.**

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

a) **The user associated with the subject not belonging to any role that permits the requested access operation on the object.**

6.1.3.37 Security attribute based access control [ST] (FDP_ACF.1(TCB))

FDP_ACF.1.1 The TSF shall enforce the **Trusted Computing Base (TCB) Policy** to objects based on the following:

- **a) Subjects: Processes acting on the behalf of users;**
  - **i. Attributes: The PV_TCB privilege;**
- **b) Objects: jfs2 file system objects (ordinary files, directories, device special files, UNIX Domain socket special files, named pipes);**
  - **i. Attributes: The trustedlib_enabled kernel security flag and TCB flag.**

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

- **a) If kernel security flag trustedlib_enabled is set to enabled and the system is in operational mode:**
  - **i. A process may not write to a file with the FSF_TLIB flag set;**
  - **ii. If the FSF_TLIB_PROC flag is set for an executable, a corresponding process may only load shared libraries that have the FSF_TLIB flag set;**
  - **iii. A process may not set or clear the FSF_TLIB or FSF_TLIB_PROC flag on an object;**
- **b) If kernel security flag trustedlib_enabled is set to enabled and the system is in maintenance mode:**
  - **i. A process with the PV_TCB privilege will ignore the file's FSF_TLIB or FSF_TLIB_PROC flag when attempting to access files;**
  - **ii. A process with the PV_TCB privilege may set or clear the FSF_TLIB or FSF_TLIB_PROC flag on an object.**

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **none**.

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

Application Note: *As stated in section 7.2.2.14.1 "TSF invocation guarantees (TP.1)", the evaluated configuration mandates that trustedlib_enabled be enable; therefore, no policy for the TCB being disabled is specified here.*

6.1.3.38 Security attribute based access control [ST] (FDP_ACF.1(TCP))

FDP_ACF.1.1 *In BAS mode, the* ~~The~~ TSF shall enforce the **TCP Port Access Control Policy** to objects based on the following:

- **a) Subjects: defined in FDP_ACC.1(TCP);**
- **b) Objects: defined in FDP_ACC.1(TCP);**
- **c) Security attributes: access control lists with entries of the following form:**
  - **i. user@host;**
  - **ii. user@subnet;**

iii. **group@host;**

iv. **group@subnet.**

FDP_ACF.1.2 *In BAS mode, the* ~~The~~ TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

Establishing a connection from another system to a TCP port on the local system can be regulated by access control lists (ACLs) on the TCP ports of the local system. A connection can only be established to a TCP port on the local system if the connection request is coming from a user and a host where the ACL for the TCP port on the local system has an entry of the form user@host, group@host, user@subnet, or group@subnet that matches the userid or groupid and the host or subnet contained in the connection request. TCP ports listed in */etc/security/services* are exempt from the ACL checks.

Port numbers 0 to 1023 are privileged ports (i.e., they require root privilege to listen). Port numbers 1024 and higher can be individually turned into privileged ports (i. e., a local user that wants to start a server process listening on such a port must have root privileges).

FDP_ACF.1.3 *In BAS mode, the* ~~The~~ TSF shall explicitly authorise access of subjects to objects based on the following additional rules:

Privilege sets override TCP Transient Storage Object decisions (a.k.a. discretionary access control (DAC)) in order to allow a process to:

a) **Change its GID (PV_DAC_GID);**
b) **Bypass DAC ownership restrictions (PV_DAC_O);**
c) **Change its role ID (PV_DAC_RID);**
d) **Change its UID (PV_DAC_UID);**
e) **Obtain all of the privileges listed in a) through d) (PV_DAC).**

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

6.1.3.39 Export of user data with security attributes [OSPP-VIRT] (FDP_ETC.2(VIRT))

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

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

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

FDP_ETC.2.4 The TSF shall enforce the following rules when user data is exported from the TOE: **none**.

Application Note: *The security attribute is the IP address associated with the network interface mapped to the WPAR.*

6.1.3.40 Complete information flow control [OSPP] (FDP_IFC.2(NI))

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

a) Subjects:
   i. unauthenticated external IT entities that send and receive information mediated by the TOE;
   ii. **Processes acting on behalf of users** that send and receive information mediated by the TOE;

b) Information:
   i. Network data routed through the TOE;
   ii. **No additional information**;

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

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

6.1.3.41 Complete information flow control [OSPP-VIRT] (FDP_IFC.2(VIRT))

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

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

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

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

Application Note: *The enforcement of the policy is based on the processes running in compartments which are associated with the compartments by their CID.*

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

6.1.3.42 Simple security attributes [OSPP] (FDP_IFF.1(NI))

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

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

b) TCP/IP information security attributes:

i. Source and destination IP address,

ii. Source and destination TCP port number,

iii. Source and destination UDP port number,

iv. Network protocol of IP, TCP, UDP, ICMP,

v. TCP header flags of SYN, ACK,

vi. IP version;

c) IEEE 802.1Q VLAN tag information security attributes:

i. VLAN tag;

d) No additional attributes.

FDP_IFF.1.2 The TSF shall permit an information flow *(indefinitely or for an authorized administrator specified fixed time period)* between a controlled subject and controlled information via a controlled operation if the following rules hold:

a) The presumed address of the source subject is in the set of source subject addresses;

b) The address of the destination subject is the set of destination subject addresses;

c) The information security attributes match the attributes in an information flow control policy specified by an authorized administrator.

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

a) Information security attribute matching;

b) **Matching based on the state of a TCP connection**;

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

a) Discard the network data **without any further processing**;

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

c) **No additional actions**.

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

FDP_IFF.1.5 The TSF shall explicitly deny an information flow *(indefinitely or for an authorized administrator specified time period)* based on the following rules:

a) Requests for access or services where the presumed source identity of the information received by the TOE is not included in the set of source addresses for the source subject;

b) Requests for access or services where the presumed source address of the information received by the TOE specifies a broadcast address;

c) Requests in which the information received by the TOE contains the route (set of host network addresses) by which information shall flow from the source subject to the TOE.

6.1.3.43 Simple security attributes [OSPP-VIRT] (FDP_IFF.1(VIRT))

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

a) Subject security attributes:
   i. **Corral ID (CID)**;
   ii. **No additional attributes**;

b) Information security attributes:
   i. **No user data security attributes**;
   ii. **The following TSF data security attributes: IP address of the IP interface assigned to the compartment**;
   iii. **No additional attributes**.

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: **the information flow shall be controlled by the Compartment Information Flow Control Policy defined in section 6.1.1.2**.

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

FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: **the information flow from the Global Environment shall be controlled by the Compartment Information Flow Control Policy defined in section 6.1.1.2**.

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

6.1.3.44 Import of user data with security attributes [OSPP] (FDP_ITC.2(BASE))

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

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

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

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

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

a) When importing data from NFSv4, the NFSv4 access control conventions and security attributes will be enforced by the TOE.

6.1.3.45 Import of user data with security attributes [OSPP-VIRT] (FDP_ITC.2(VIRT))

FDP_ITC.2.1 The TSF shall enforce the Compartment Access Control Policy and Compartment Information Flow Control Policy when importing user data, controlled under the SFP, from outside of the TOE.

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

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

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

FDP_ITC.2.5 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: **none**.

6.1.3.46 Full residual information protection [OSPP] (FDP_RIP.2)

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

6.1.3.47 Full residual information protection of resources [OSPP] (FDP_RIP.3)

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

6.1.3.48 Hard disk drive residual information protection [ST] (FDP_RIP.4)

FDP_RIP.4.1 The TSF shall overwrite all data stored in current user-accessible blocks of a hard disk drive with predefined bit patterns upon request of an authorized administrator.

6.1.3.49 Stored data integrity monitoring and action [OSPP-IV] (FDP_SDI.2(IV))

FDP_SDI.2.1 The TSF shall be able to monitor user data stored in containers controlled by the TSF for modification of that data on all objects, based on the following attributes: **cryptographic message digest *(SHA-256)* of the user data**.

FDP_SDI.2.2 Upon detection of a data integrity error, the TSF shall be able to perform one or more of the following actions:

a) Deny the subject's access request to the integrity-violated user data;

b) Inform the subject attempting to access the modified user data upon initial access;

c) **No additional actions**.

Application Note: *AIX directly supports the first option, by stopping to load the file (policy STOP_ON_CHKFAIL). The second option is implicitly supported as the access will fail which the application can report back to the user.*

6.1.3.50 Authentication failure handling [OSPP] (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when an administrator-configurable number of unsuccessful authentication attempts for the authentication ~~method~~ *methods* **file-based I&A and LDAP-based I&A** occur related to **invalid passwords**.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been **surpassed**, the TSF shall: **lock the user's account**.

6.1.3.51 User attribute definition [OSPP] (FIA_ATD.1(HU))

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

a) User identifier;
b) Group memberships;
c) User password;
d) Software token verification data *(i.e., Kerberos ticket granting ticket when NAS is used in conjunction with NFSv4 as defined in this document)*;
e) Security roles *(i.e., Authorizations)*;
f) **Audit classes;**
g) **Password aging data;**
h) **Principle name (Kerberos);**
i) **X.509v3 certificates (EFS)**.

6.1.3.52 User attribute definition [OSPP] (FIA_ATD.1(TU))

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

a) the logical or physical network interface through which the network data entered the TOE;
b) identity of the logical or physical external interface through which the user connected to the TOE;
c) **No additional user security attributes**.

6.1.3.53 Verification of secrets [OSPP] (FIA_SOS.1(BASE))

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

6.1.3.54 Timing of authentication [OSPP] (FIA_UAU.1)

FIA_UAU.1.1 The TSF shall allow

a) the information flow covered by the Network Information Flow Control Policy;
b) **No additional TSF mediated actions**

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

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

6.1.3.55 Multiple authentication mechanisms [OSPP] (FIA_UAU.5)

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

a) Authentication based on username and password;
b) Authentication based on software token verification data *(i.e., Kerberos tickets)*;

c) **No additional authentication mechanisms**

to support user authentication.

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

a) Authentication based on username and password is performed for TOE-originated requests and credentials stored by the TSF *(i.e., file-based authentication)*;

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

c) **Authentication based on username and password is performed for TOE-originated requests and credentials stored by LDAP**.

6.1.3.56 Protected authentication feedback [OSPP] (FIA_UAU.7(BASE))

FIA_UAU.7.1 The TSF shall provide only obscured feedback to the user while the authentication is in progress.

6.1.3.57 Timing of identification [OSPP] (FIA_UID.2(BASE))

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

6.1.3.58 User identification before any action [OSPP-VIRT] (FIA_UID.2(VIRT))

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

6.1.3.59 Enhanced user-subject binding [OSPP] (FIA_USB.2)

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

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

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

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

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

e) Active roles;

f) Active groups;

g) **Audit classes;**

h) **Privilege sets and privilege authorization sets associated with the subject being activated**.

Application Note: *While privilege sets and privilege authorization sets are actually associated with subjects (i.e., executables being activated on the behalf of a user as processes) rather than being user security attributes in the narrower sense, they have been added to this requirement to emphasize the fact that they become part of the subject's security attributes relevant for its execution.*

FIA_USB.2.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users:

a) **Upon successful identification and authentication, the real user identifier, the effective user identifier, and audit user identifier shall be those specified in the user entry for the user that has authenticated successfully;**

b) **Upon successful identification and authentication, the real group identifier, and the effective group identifier shall be those specified via the group membership attribute in the user entry.**

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

a) **The effective user ID of a user can be changed by the use of an executable with the setuid bit set. In this case the program is executed with the effective user ID of the program owner. The effective user ID can also be changed if the executable is listed in the privcmds table and the table includes an effective user ID value. In this case, the program is executed with the effective user ID in the table when the program is executed as a Privileged Command. Access rights are then evaluated using the effective user ID. The login user ID is not changed with this process, so all audit records can be traced to the real user that executes the program.**

b) **The effective user ID of a user can be changed by the su command. In this case the effective user ID of the user is changed to the user specified in the su command (provided authentication is successful). The login user ID remains unchanged, so all audit records can be traced to the real user that executes the program.**

c) **The effective group ID of a user can be changed by the use of an executable with the setguid bit set. In this case the program is executed with the effective group ID of the program owner. The effective group ID can also be changed if the executable is listed in the privcmds table and the table includes an effective group ID value. In this case, the program is executed with the effective group ID in the table when the program is executed as a Privileged Command. Access rights are then evaluated using the effective group ID. The login user ID is not changed with this process, so all audit records can be traced to the real user that executes the program.**

FIA_USB.2.4 The TSF shall enforce the following rules for the assignment of subject security attributes not derived from user security attributes when a subject is created: **none**.

6.1.3.60 Management of object security attributes [OSPP] (FMT_MSA.1(PSO-AIXC))

FMT_MSA.1.1 The TSF shall enforce the *AIXC* Persistent Storage Object Access Control Policy to restrict the ability to modify **(and no additional operations)** the security attributes of the objects covered by the SFP to the owner of the object and **authorized administrators**.

6.1.3.61 Management of object security attributes [OSPP] (FMT_MSA.1(PSO-NFS))

FMT_MSA.1.1 The TSF shall enforce the *NFS* Persistent Storage Object Access Control Policy to restrict the ability to modify **(and no additional operations)** the security attributes of the objects covered by the SFP to the owner of the object and **authorized administrators and other users than have been granted permission within the ACL to modify the access control attributes of the object**.

6.1.3.62 Management of object security attributes [OSPP] (FMT_MSA.1(TSO))

FMT_MSA.1.1 The TSF shall enforce the Transient Storage Object Access Control Policy to restrict the ability to modify **(and no additional operations)** the security attributes of the objects covered by the SFP to the owner of the object and **authorized administrators**.

6.1.3.63 Management of security attributes [OSPP-VIRT] (FMT_MSA.1(VIRT-CACP))

FMT_MSA.1.1 The TSF shall enforce the Compartment Access Control Policy to restrict the ability to change_default, query, modify**, delete, (and no additional operations)** the security attributes of the subjects and objects covered by the SFP, **(and no additional security attributes)** to **authorized administrators in the Global Environment**.

6.1.3.64 Management of security attributes [OSPP-VIRT] (FMT_MSA.1(VIRT-CIFCP))

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

6.1.3.65 Management of object security attributes [ST] (FMT_MSA.1(AUTH))

FMT_MSA.1.1 The TSF shall enforce the **Authorization Policy** to restrict the ability to **modify** the security attributes **authorizations** to **authorized administrators**.

6.1.3.66 Management of object security attributes [ST] (FMT_MSA.1(RBAC-ADM))

FMT_MSA.1.1 The TSF shall enforce the **Role-based Access Control (RBAC) Policy** to restrict the ability to **modify** the security attributes **of objects** to **object owners and the set of RBAC administrative roles**.

6.1.3.67 Management of object security attributes [ST] (FMT_MSA.1(RBAC-AUTH))

FMT_MSA.1.1 The TSF shall enforce the **Role-based Access Control (RBAC) Policy** to restrict the ability to **modify, delete, create instances of** the security attributes **User Role Authorizations** to **a set of RBAC administrative roles**.

6.1.3.68 Management of object security attributes [ST] (FMT_MSA.1(RBAC-DFLT))

FMT_MSA.1.1 The TSF shall enforce the **Role-based Access Control (RBAC) Policy** to restrict the ability to **modify, create** the security attributes **Default Active Role Set** to **a set of RBAC administrative roles**.

6.1.3.69 Management of object security attributes [ST] (FMT_MSA.1(RBAC-USR))

FMT_MSA.1.1 The TSF shall enforce the **Role-based Access Control (RBAC) Policy** to restrict the ability to **modify the composition of** the *session* security attributes **Active Role set for a user** to **the session owner**.

6.1.3.70 Management of object security attributes [ST] (FMT_MSA.1(TCB))

FMT_MSA.1.1 The TSF shall enforce the **Trusted Computing Base (TCB) Policy** to restrict the ability to **change_default** the security attributes **FSF_TLIB and FSF_TLIB_PROC** to **authorized administrators**.

6.1.3.71 Management of object security attributes [ST] (FMT_MSA.1(TCP))

FMT_MSA.1.1 *In BAS mode, the* ~~The~~ TSF shall enforce the **TCP Port Access Control Policy** to restrict the ability to **modify** the security attributes **of the objects covered by the SFP** to **authorized administrators**.

6.1.3.72 Secure security attributes [ST] (FMT_MSA.2(RBAC))

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

6.1.3.73 Static attribute initialisation [OSPP] (FMT_MSA.3(PSO-AIXC))

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

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

6.1.3.74 Static attribute initialisation [OSPP] (FMT_MSA.3(PSO-NFS))

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

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

6.1.3.75 Static attribute initialisation [OSPP] (FMT_MSA.3(TSO))

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

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

6.1.3.76 Static attribute initialisation [OSPP] (FMT_MSA.3(NI))

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

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

6.1.3.77 Static attribute initialisation [OSPP-VIRT] (FMT_MSA.3(VIRT-CACP))

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

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

6.1.3.78 Static attribute initialisation [OSPP-VIRT] (FMT_MSA.3(VIRT-CIFCP))

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

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

6.1.3.79 Static attribute initialisation [ST] (FMT_MSA.3(AUTH))

FMT_MSA.3.1 The TSF shall enforce the **Authorization Policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

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

6.1.3.80 Static attribute initialisation [ST] (FMT_MSA.3(RBAC))

FMT_MSA.3.1 The TSF shall enforce the **Role-based Access Control (RBAC) Policy** to provide **administrative user defined** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the **set of RBAC administrative roles** to specify alternative initial values to override the default values when an object or information is created.

6.1.3.81 Static attribute initialisation [ST] (FMT_MSA.3(TCB))

FMT_MSA.3.1 The TSF shall enforce the **Trusted Computing Base (TCB) Policy** to provide **permissive** default values for security attributes that are used to enforce the SFP.

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

6.1.3.82 Static attribute initialisation [ST] (FMT_MSA.3(TCP))

FMT_MSA.3.1 *In BAS mode, the* ~~The~~ TSF shall enforce the **TCP Port Access Control Policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 *In BAS mode, the* ~~The~~ TSF shall allow the **authorized administrators** to specify alternative initial values to override the default values when an object or information is created.

6.1.3.83 Security attribute value inheritance [OSPP] (FMT_MSA.4(PSO))

FMT_MSA.4.1 The TSF shall use the following rules to set the value of security attributes for Persistent Storage Objects: **the initial permission bits are those specified by the creating process bitwise ANDed with the one's compliment of the process' UMASK value**.

6.1.3.84 Management of TSF data [OSPP] (FMT_MTD.1(AE))

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

Application Note: *This SFR applies to FAU_SEL.1(BASE) and FAU_SEL.1(LS).*

6.1.3.85 Management of TSF data [OSPP] (FMT_MTD.1(AS))

FMT_MTD.1.1 The TSF shall restrict the ability to clear, **configure the storage location, create, delete** the audit storage to **authorized administrators**.

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

6.1.3.86 Management of TSF data [OSPP] (FMT_MTD.1(AT))

FMT_MTD.1.1 The TSF shall restrict the ability to modify, **add, delete** the

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

b) action when the threshold is reached

to **authorized administrators**.

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

6.1.3.87 Management of TSF data [OSPP] (FMT_MTD.1(AF))

FMT_MTD.1.1 The TSF shall restrict the ability to modify, **add, delete** the actions to be taken in case of audit storage failure to **authorized administrators**.

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

6.1.3.88 Management of TSF data [OSPP] (FMT_MTD.1(NI))

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

a) identification of network data;

b) actions performed on the identified network data

to **authorized administrators**.

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

6.1.3.89 Management of TSF data [OSPP] (FMT_MTD.1(IAT))

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

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

6.1.3.90 Management of TSF data [OSPP] (FMT_MTD.1(IAF))

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

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

6.1.3.91 Management of TSF data [OSPP] (FMT_MTD.1(IAU))

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

Application Note: *This SFR applies to FIA_ATD.1, FIA_UAU.1, and FIA_UID.2(BASE).*

6.1.3.92 Management of TSF data [OSPP-AM] (FMT_MTD.1(AM-AP))

FMT_MTD.1.1 The TSF shall restrict the ability to **execute** the **administrator defined commands tagged with the authroles attribute** to **the roles identified by a command's authroles attribute** only after another user with the role **listed in the authroles attribute and different from the initiating user's role** has approved the action.

Application Note: *This uses the n-man rule of AIX to allow execution of specific commands only if all required roles could authenticate before the execution of the command.*

6.1.3.93 Management of TSF data [OSPP-AM] (FMT_MTD.1(AM-MR))

FMT_MTD.1.1 The TSF shall restrict the ability to modify, **change_default, delete, clear** the assignment of roles to users down to the granularity of single users to **authorized administrators**.

Application Note: *Each user has a potential set of roles associated with their user account. Administrators authorized to create and manage users can manage the set of roles assigned to each user account. A role contains a set of authorizations. Authorizations are used by commands to determine if the user is allowed to perform a specific function within the command. In addition, the privcmds database can provide and control role access to commands. A role can also contain a list of other roles that it includes/implies.*

6.1.3.94 Management of TSF data [OSPP-AM] (FMT_MTD.1(AM-MD))

FMT_MTD.1.1 The TSF shall restrict the ability to **delegate** the **role or roles used to manage a user's authroles attribute** to users granted that role.

Application Note: *By default, only the ISSO role can manage all users' authroles attribute; thereby delegating some or all of the ISSO responsibilities to another user. Additional roles can be created which also allow for the management of all users' authroles attribute.*

6.1.3.95 Management of TSF data [OSPP-AM] (FMT_MTD.1(AM-MA))

FMT_MTD.1.1 The TSF shall restrict the ability to modify, **change_default, query, delete, clear** the **set of administrator defined commands tagged with the authroles attribute as well as the value of each tagged command's authroles attribute** to **authorized administrators**.

6.1.3.96 Management of TSF data [OSPP-IV] (FMT_MTD.1(IV-ACT))

FMT_MTD.1.1 The TSF shall restrict the ability to query, modify, **change_default, delete, clear** the actions to be performed upon detection of an integrity violation to **authorized administrators**.

6.1.3.97 Management of TSF data [OSPP-IV] (FMT_MTD.1(IV-TSF))

FMT_MTD.1.1 The TSF shall restrict the ability to enable, disable, select the TSF code or TSF data for, generate and update the integrity data base of the TSF data for **change_default, query, modify, delete, clear** the integrity verification of the TSF code and TSF data to **authorized administrators**.

6.1.3.98 Management of TSF data [OSPP-IV] (FMT_MTD.1(IV-USR))

FMT_MTD.1.1 The TSF shall restrict the ability to enable, disable, select the user data for, generate and update the integrity data base of the user data for **change_default, query, modify, delete, clear** the integrity verification of user data to **authorized administrators**.

6.1.3.99 Management of TSF data [OSPP-VIRT] (FMT_MTD.1(VIRT-COMP))

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

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

6.1.3.100 Management of TSF data [ST] (FMT_MTD.1(PRIVS))

FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **privileges on applications, devices, and WPARs** to **authorized administrators**.

6.1.3.101 Management of TSF data [ST] (FMT_MTD.1(RBAC))

FMT_MTD.1.1 The TSF shall restrict the ability to **modify, create** the **TSF data**

a) All user passwords;

b) Role definition and role attributes;

c) Role hierarchies (by assigning one or more roles to other roles);

d) Constraints among role relationships;

e) List of auditable events

to **a set of RBAC administrative roles**.

6.1.3.102 Secure TSF data [ST] (FMT_MTD.3(RBAC))

FMT_MTD.3.1 The TSF shall ensure that only secure values are accepted for **role definitions, role hierarchies, and role relationship constraints**.

6.1.3.103 Revocation [OSPP] (FMT_REV.1(OBJ))

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

FMT_REV.1.2 The TSF shall enforce the following rules:

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

b) **No additional revocation rules**.

6.1.3.104 Revocation [OSPP] (FMT_REV.1(USR))

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

FMT_REV.1.2 The TSF shall enforce the following rules:

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

b) **No additional revocation rules**.

6.1.3.105 Specification of management functions [OSPP] (FMT_SMF.1(BASE))

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

a) Management of auditing;

b) Management of cryptographic network protocols;

c) Management of Persistent Storage Object Access Control Policy;

d) Management of Transient Storage Object Access Control Policy;

e) Management of Network Information Flow Control Policy;

f) Management of identification and authentication policy;

g) Management of user security attributes;

h) **Management of the hard disk residual information protection;**

i) **Management of privileges on applications, devices, and WPARS**.

6.1.3.106 Security roles [OSPP] (FMT_SMR.2)

FMT_SMR.2.1 The TSF shall maintain the roles:

a) User role with the following rights:

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

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

iii. **Other rights as assigned by an authorized administrator via the RBAC mechanism**;

b) **The set of RBAC administrative roles**.

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 The TSF shall ensure that the conditions

a) Object owners can modify security attributes for only the objects they own (except for the sensitivity label in LAS mode);

b) The set of RBAC administrative roles can modify security attributes for all objects under the control of TOE (since they automatically inherit the privileges of all objects owners)

are satisfied.

Application Note: *A Global environment administrator is a special name given to an administrator of the Global environment. It is not a special type of administrator.*

6.1.3.107 Failure with preservation of secure state [ST] (FPT_FLS.1(RBAC))

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur:

a) The entire RBAC database containing data on privileges assigned to a role, users authorized for a role, role constraints and relationships or some specific tables containing subsets of these data are off-line, corrupt, or inaccessible.

6.1.3.108 Failure with preservation of secure state [ST] (FPT_FLS.1(SED))

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur:

a) Unauthorized execution of code on a process' stack.

6.1.3.109 Manual recovery [ST] (FPT_RCV.1)

FPT_RCV.1.1 After **a failure or service discontinuity,** the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

6.1.3.110 Function recovery [ST] (FPT_RCV.4)

FPT_RCV.4.1 The TSF shall ensure that

a) The security function that checks whether a specified privilege is assigned to any role but the database containing the privilege data is not on-line or the particular data table is inaccessible;

b) The security function that checks whether a specified role has been assigned to a particular user but the database containing the role membership information is not on-line or the particular data table is inaccessible

have the property that the function either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state.

6.1.3.111 Reliable time stamps [OSPP] (FPT_STM.1)

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

6.1.3.112 Inter-TSF basic TSF data consistency [OSPP] (FPT_TDC.1(BASE))

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret **persistent storage object access control data** when shared between the TSF and ~~another trusted IT product~~ *an NFSv4 server*.

FPT_TDC.1.2 The TSF shall use **the NFSv4 rules** when interpreting the TSF data from ~~another trusted IT product~~ *an NFSv4 server*.

6.1.3.113 Inter-TSF basic TSF data consistency [OSPP-VIRT] (FPT_TDC.1(VIRT))

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret access control and information flow control-related security attributes, **and IP addresses** when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use **a common /etc/hosts file** when interpreting the TSF data from another trusted IT product.

6.1.3.114 TSF integrity monitoring and action [OSPP-IV] (FPT_TIM.1(IV))

FPT_TIM.1.1 The TSF shall be able to monitor TSF code, TSF data for unauthorized modification of the TSF code and TSF data using the following rules:

a) verification of cryptographic message digest *(SHA-256)* of the TSF code;

b) verification of cryptographic message digest *(SHA-256)* of the TSF data;.

FPT_TIM.1.2 Upon detection of a data integrity error, the TSF shall be able to perform one or more of the following actions:

a) Deny the subject's access request to the integrity-violated TSF code or TSF data;

b) Inform the administrator about the integrity violation upon initial access;

c) **No additional actions**.

Application Note: *In FPT_TIM.1.2 above, item a) is supported by stopping to load the file (policy STOP_ON_CHKFAIL). Item b) is supported via an audit record.*

6.1.3.115 TSF testing [ST] (FPT_TST.1)

FPT_TST.1.1 The TSF shall run a suite of self tests **periodically during normal operation, at the request of the authorised user** to demonstrate the correct operation of **the TSF**.

FPT_TST.1.2 The TSF shall provide authorised users with the capability to verify the integrity of **TSF data**.

FPT_TST.1.3 The TSF shall provide authorised users with the capability to verify the integrity of **stored TSF executable code**.

6.1.3.116 Limited fault tolerance [ST] (FRU_FLT.2)

FRU_FLT.2.1 The TSF shall ensure the operation of all the TOE's capabilities when the following failures occur: **when a process with the Stack Execution Disable (SED) flag enabled attempts to execute code on the process' stack**.

6.1.3.117 Limitation on scope of selectable attributes [ST] (FTA_LSA.1(RBAC))

FTA_LSA.1.1 The TSF shall restrict the scope of the session security attributes **active role set for the user**, based on **the set of authorized roles for the user**.

6.1.3.118 TSF-initiated session locking [OSPP] (FTA_SSL.1)

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

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

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

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

a) Successful re-authentication with the credentials of the user owning the session using

i) Authentication based on username and password;

b) **No additional events**.

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

6.1.3.119 User-initiated locking [OSPP] (FTA_SSL.2)

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

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

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

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

a) Successful re-authentication with the credentials of the user owning the session using

i) Authentication based on username and password;

b) **No additional events**.

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

6.1.3.120 TOE session establishment [ST] (FTA_TSE.1(RBAC))

FTA_TSE.1.1 The TSF shall be able to deny session establishment based on **the default active role set for the user being empty**.

6.1.3.121 Inter-TSF trusted channel [OSPP] (FTP_ITC.1)

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

a) Cryptographically-protected communication channel using **Kerberos Version 5 GSS-API for communication with an NFSv4 server**;

b) **No additional mechanisms**.

FTP_ITC.1.2 The TSF shall permit **the TSF** to initiate communication via the trusted channel.

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

Application Note: *In this ST, IPsec is described in the context of protected remote access for user connections, not for trusted IT product connections; therefore, IPsec is excluded from this SFR.*

6.1.4 Additional Trusted AIX security functional requirements (i.e., LAS mode only)

This section contains SFRs that are only supported by Trusted AIX (i.e., LAS mode only). Trusted AIX supports these SFRs in addition to the AIX and Trusted AIX shared SFRs defined in section 6.1.3.

6.1.4.1 Audit data generation [ST] (LAS mode only) (FAU_GEN.1(LS))

FAU_GEN.1.1 *In LAS mode, the* ~~The~~ TSF shall be able to generate an audit record of the following auditable events:

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

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

c) **none**.

FAU_GEN.1.2 *In LAS mode, the* ~~The~~ TSF shall record within each audit record at least the following information:

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

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

i. **The sensitivity labels of subjects, objects, or information involved**.

6.1.4.2 Selectable audit review [ST] (LAS mode only) (FAU_SAR.3(LS))

FAU_SAR.3.1 *In LAS mode, the* ~~The~~ TSF shall provide the ability to apply **searches, sorting, and ordering** of audit data based on

a) Subject sensitivity label;

b) Object sensitivity label;

c) MAC success or failure;

d) MIC success or failure.

Application Note:: *The sensitivity label(s) in an audit record indicate the sensitivity level of the subject and/or object at the time the audit record was generated. They do not indicate the sensitivity level of the audit record. Administrators authorized to review the audit trail can see all audit records regardless of the sensitivity label(s) in the audit record.*

6.1.4.3 Selective audit [OSPP] (LAS mode only) (FAU_SEL.1(LS))

FAU_SEL.1.1 *In LAS mode, the* ~~The~~ TSF shall be able to select the set of events to be audited from the set of all auditable events based on the following attributes:

a) Type of audit event;

b) Subject or user identity;
c) Outcome (success or failure) of the audit event;
d) Named object identity;
e) **Host identity;**
f) **Users belonging to a specified role;**
g) **Access types on a particular object;**
h) **Subject sensitivity label;**
i) **Object sensitivity label.**

Application Note:: *This SFR is an iteration of FAU_SEL.1 from the OSPP base.*

6.1.4.4 Export of user data with security attributes [OSPP-LS] (LAS mode only) (FDP_ETC.2(LS))

FDP_ETC.2.1 *In LAS mode, the* ~~The~~ TSF shall enforce the Multilevel Confidentiality Information Flow Control Policy when exporting user data, controlled under the SFP(s), outside of the TOE.

FDP_ETC.2.2 *In LAS mode, the* ~~The~~ TSF shall export the user data with the user data's associated security attributes.

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

FDP_ETC.2.4 *In LAS mode, the* ~~The~~ TSF shall enforce the following rules when user data is exported from the TOE:

a) When data is exported in hardcopy form, each page shall be marked with a printed representation of the sensitivity label of the subject requesting the export of the page. By default, this marking shall appear on both the top and bottom of each printed page.
b) When the data is exported to a device, the security attributes shall be exported with the data using **the backup command labeling format**.
c) **When data is exported using Trusted Networking (TN), the security attributes shall be exported with the data using the CIPSO and RIPSO formats**.
d) **When data is exported using telnet, rlogin, rsh, rcp, rexec, and FTP, the sensitivity level of the local session and remote session are identical**.

Application Note: *In LAS mode, printing is disabled in the evaluated configuration.*

6.1.4.5 Subset information flow control [ST] (LAS mode only) (FDP_IFC.1(MIC))

Mandatory Integrity Control (MIC) Policy			
Subject	**Object Type**	**Named Object**	**Operations causing Information Flow To/From Subjects over Named Objects**
Processes acting on behalf of a specific user	FSO	device special files – block and character, TCB	Write
		directory – regular	
		file – regular, system, audit	
		symbolic link	
	IPC	message	Write
		semaphore	
		shared memory	
	Miscellaneous	signal vector	Write
		STREAMS message block	

Table 12: MIC subjects, objects, and operations

FDP_IFC.1.1 *In LAS mode, the* ~~The~~ TSF shall enforce the **Mandatory Integrity Control (MIC) Policy** on **the subjects, objects, and operations defined in Table 12**.

6.1.4.6 Subset information flow control [ST] (LAS mode only) (FDP_IFC.1(TN))

FDP_IFC.1.1 *In LAS mode, the* ~~The~~ TSF shall enforce the **Trusted Network (TN) Policy** on

a) Subjects: hosts identified by IP addresses;

b) Objects: data packets to be transferred between hosts;

c) Operations: the transfer of data packets to and from hosts via network connections.

6.1.4.7 Complete information flow control [OSPP-LS] (LAS mode only) (FDP_IFC.2(LS))

Multilevel Confidentiality Information Flow Control (MCIFC) Policy			
Subject	**Object Type**	**Named Object**	**Operations causing Information Flow To/From Subjects over Named Objects**
Processes acting on behalf of a specific user	FSO	device special files – block and character, TCB	Read/Write/Exec
		directory – regular	
		file – regular, system, audit	
		symbolic link	
	IPC	message	Read/Write
		semaphore	
		shared memory	
	Miscellaneous	signal vector	Read/Write
		STREAMS message block	
		pipe - unnamed (FIFO)	

Table 13: MCIFC subjects, objects, and operations

FDP_IFC.2.1 *In LAS mode, the* ~~The~~ TSF shall enforce the Multilevel Confidentiality Information Flow Control Policy on

a) Subjects**: defined in Table 13**;

b) Objects: **defined in Table 13**

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

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

6.1.4.8 Hierarchical security attributes [ST] (LAS mode only) (FDP_IFF.2(MIC))

FDP_IFF.2.1 *In LAS mode, the* ~~The~~ TSF shall enforce the **Mandatory Integrity Control (MIC) Policy** based on the following types of subject and information security attributes:

a) Subject security attributes:

i. Integrity label (TL) consisting of a hierarchical level;

b) Object information security attributes:

i. Integrity label (TL) of objects containing the information and consisting of a hierarchical level.

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

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

FDP_IFF.2.3 *In LAS mode, the* ~~The~~ TSF shall enforce ~~the~~ **no additional rules**.

FDP_IFF.2.4 *In LAS mode, the* ~~The~~ TSF shall explicitly authorise an information flow based on the following rules:

Privilege sets override MIC decisions in order to:

a) Bypass integrity clearance restrictions (PV_MIC_CL);

b) Bypass integrity restrictions (PV_MIC).

FDP_IFF.2.5 *In LAS mode, the* ~~The~~ TSF shall explicitly deny an information flow based on the following rules: **none**.

FDP_IFF.2.6 *In LAS mode, the* ~~The~~ TSF shall enforce the following relationships for any two valid information flow control security attributes:

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

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

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

6.1.4.9 Hierarchical security attributes [ST] (LAS mode only) (FDP_IFF.2(TN))

FDP_IFF.2.1 *In LAS mode, the* ~~The~~ TSF shall enforce the **Trusted Network (TN) Policy** based on the following types of subject and information security attributes:

a) Subject security attributes:

i. IP address;

b) Object information security attributes:

i. IP address packet source and destination;

ii. Protocol;

iii. Port (source and destination);

iv. Network interface;

v. IPSO labels;

vi. IPSO security attributes;

vii. Minimum and maximum SL (only when using CIPSO).

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

a) if the source/destination IP address of the packet is equal to the source/destination IP address specified in the rule;

b) If the IP address of the packet is within the network mask specified for the rule;

c) If the direction of the packet flow corresponds to the direction of the rule (IN/OUT);

d) If the protocol of the packet is equal to the protocol specified in the rule;

e) If the source/destination port is within the source/destination port range specified in the rule;

f) If the network interface of the packet is equal to the network interface specified in the rule;

g) If the IPSO labels are within the range defined by the rule, and rule set to allow IPSO labels;

h) If the packet's SL is within the minimum and maximum SL specified for the rule.

FDP_IFF.2.3 *In LAS mode, the* ~~The~~ TSF shall enforce ~~the~~ **no additional rules**.

FDP_IFF.2.4 *In LAS mode, the* ~~The~~ TSF shall explicitly authorise an information flow based on the following rules:

Privilege sets override TN decisions in order to allow a process to:

a) Perform restricted ioctl calls to drivers (PV_NET_CNTL);

b) Modify network configuration (PV_NET_CONFIG);

c) Open a restricted port (PV_NET_PORT);

d) Access raw sockets (PV_NET_RAWSOCK);

e) Obtain the privileges in a) to c) (PV_NET).

FDP_IFF.2.5 *In LAS mode, the* ~~The~~ TSF shall explicitly deny an information flow based on the following rules: **none**.

FDP_IFF.2.6 *In LAS mode, the* ~~The~~ TSF shall enforce the following relationships for any two valid information flow control security attributes:

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

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

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

6.1.4.10 Hierarchical security attributes [OSPP-LS] (LAS mode only) (FDP_IFF.2(LS))

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

a) Subject security attributes:

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

ii. **No additional attributes**;

b) Object security attributes:

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

ii. **No additional attributes**.

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

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

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

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

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

FDP_IFF.2.3 *In LAS mode, the* ~~The~~ TSF shall enforce ~~the~~ **no additional rules**.

FDP_IFF.2.4 *In LAS mode, the* ~~The~~ TSF shall explicitly authorise an information flow based on the following rules:

a) Privilege sets override MAC decisions in order to bypass:

i. Sensitivity clearance restrictions (PV_MAC_CL);

ii. MAC restrictions when:

1. Files flagged as being exempt from MAC (PV_MAC_OVRRD);

2. Sending a signal (PV_MAC_W_PROC) or when getting information about a process (PV_MAC_R_PROC), provided that the target process's label is within the acting process's sensitivity clearance;

iii. All MAC READ restrictions (PV_MAC_R);

iv. MAC READ restrictions when:

1. The object's label is within the process's sensitivity clearance (PV_MAC_R_CL);

2. Reading a STREAM message block, provided that the message's label is within the process's sensitivity clearance (PV_MAC_R_STR);

v. All MAC WRITE restrictions (PV_MAC_W);

vi. MAC WRITE restrictions when:

1. The process label is greater than or equal to the object's label and the object's label is within the process's sensitivity clearance (PV_MAC_W_DN);

2. The process label is less then or equal to the object's label and the object's label is within the process's sensitivity clearance (PV_MAC_W_UP);

3. The object's label is within the process's sensitivity clearance (PV_MAC_W_CL);

vii. A combination of all other MAC privileges (PV_MAC);

b) SIGCHILD is exempt from the checks in FDP_IFF.2.2(LS);

c) For partitioned directories, signals, and network streams: if the sensitivity label of the object is greater than the sensitivity label of the subject, then the flow of information from the subject to the object is permitted (write operation).

Application Note: *Multilevel Confidentiality Information Flow Control is known as Mandatory Access Control (MAC) in the Trusted AIX guidance.*

FDP_IFF.2.5 *In LAS mode, the* ~~The~~ TSF shall explicitly deny an information flow based on the following rules: **none**.

FDP_IFF.2.6 *In LAS mode, the* ~~The~~ TSF shall enforce the following relationships for any two valid information flow control security attributes:

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

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

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

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

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

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

6.1.4.11 Import of user data without security attributes [OSPP-LS] (LAS mode only) (FDP_ITC.1(LS))

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

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

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

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

b) **Only authorized users can import unlabeled data**.

6.1.4.12 Import of user data with security attributes [OSPP-LS] (LAS mode only) (FDP_ITC.2(LS))

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

FDP_ITC.2.2 *In LAS mode, the* ~~The~~ TSF shall use the label-related security attributes associated with the imported labeled user data.

FDP_ITC.2.3 *In LAS mode, the* ~~The~~ TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

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

FDP_ITC.2.5 *In LAS mode, the* ~~The~~ TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE:

a) **The user data must have a sensitivity label consisting of the following values, otherwise import is denied:**

i. A hierarchical level; and

ii. A set of non-hierarchical categories.

6.1.4.13 User attribute definition [OSPP-LS] (LAS mode only) (FIA_ATD.1(LS))

FIA_ATD.1.1 *In LAS mode, the* ~~The~~ TSF shall maintain the following list of security attributes belonging to individual users:

a) Sensitivity label.

6.1.4.14 User attribute definition [ST] (LAS mode only) (FIA_ATD.1(LSX))

FIA_ATD.1.1 *In LAS mode, the* ~~The~~ TSF shall maintain the following list of security attributes belonging to individual users:

a) Sensitivity clearance range;

b) Default sensitivity label;

c) Integrity label;

d) Integrity clearance range;

e) Default integrity label.

6.1.4.15 User-subject binding [OSPP-LS] (LAS mode only) (FIA_USB.1(LS))

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

a) User sensitivity level that is used to enforce the Multilevel Confidentiality Information Flow Control Policy.

FIA_USB.1.2 *In LAS mode, the* ~~The~~ TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users:

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

b) If the user doesn't specify a sensitivity level when logging in, the default sensitivity label is used.

FIA_USB.1.3 *In LAS mode, the* ~~The~~ TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: **none**.

6.1.4.16 User-subject binding [ST] (LAS mode only) (FIA_USB.1(LSX))

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

a) User integrity level that is used to enforce the Mandatory Integrity Control (MIC) Policy.

FIA_USB.1.2 *In LAS mode, the* ~~The~~ TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users:

a) The integrity label associated with a subject shall be within the integrity clearance range of the user;

b) If the user doesn't specify an integrity level when logging in, the default integrity label is used.

FIA_USB.1.3 *In LAS mode, the* ~~The~~ TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: **none**.

6.1.4.17 Management of security attributes [OSPP-LS] (LAS mode only) (FMT_MSA.1(LS))

FMT_MSA.1.1 *In LAS mode, the* ~~The~~ TSF shall enforce the Multilevel Confidentiality Information Flow Control Policy to restrict the ability to modify the label-related object security attributes to **authorized administrators**.

6.1.4.18 Management of security attributes [ST] (LAS mode only) (FMT_MSA.1(MIC))

FMT_MSA.1.1 *In LAS mode, the* ~~The~~ TSF shall enforce the **Mandatory Integrity Control (MIC) Policy** to restrict the ability to **modify** the security attributes **integrity labels (TL)** to **authorized administrators**.

6.1.4.19 Management of security attributes [ST] (LAS mode only) (FMT_MSA.1(TN))

FMT_MSA.1.1 *In LAS mode, the* ~~The~~ TSF shall enforce the **Trusted Network (TN) Policy** to restrict the ability to **modify** the security attributes **information flow control attributes representing the TN rules** to **authorized administrators**.

6.1.4.20 Static attribute initialisation [OSPP-LS] (LAS mode only) (FMT_MSA.3(LS))

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

FMT_MSA.3.2 *In LAS mode, the* ~~The~~ TSF shall allow the **authorized administrators** to specify alternative initial values to override the default values when an object or information is created.

6.1.4.21 Static attribute initialisation [ST] (LAS mode only) (FMT_MSA.3(MIC))

FMT_MSA.3.1 *In LAS mode, the* ~~The~~ TSF shall enforce the **Mandatory Integrity Control (MIC) Policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 *In LAS mode, the* ~~The~~ TSF shall allow the **authorized administrators** to specify alternative initial values to override the default values when an object or information is created.

6.1.4.22 Static attribute initialisation [ST] (LAS mode only) (FMT_MSA.3(TN))

FMT_MSA.3.1 *In LAS mode, the* ~~The~~ TSF shall enforce the **Trusted Network (TN) Policy** to provide **permissive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 *In LAS mode, the* ~~The~~ TSF shall allow the **authorized administrators** to specify alternative initial values to override the default values when an object or information is created.

6.1.4.23 Inter-TSF basic TSF data consistency [OSPP-LS] (LAS mode only) (FPT_TDC.1(LS))

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

FPT_TDC.1.2 *In LAS mode, the* ~~The~~ TSF shall use **label encoding rules** when interpreting the TSF data from another trusted IT product.

6.1.5 VIOS security functional requirements

This section contains the complete set of SFRs supported by VIOS.

6.1.5.1 Subset access control [ST] (VIOS only) (FDP_ACC.1(VIOS))

FDP_ACC.1.1 *For VIOS, the* ~~The~~ TSF shall enforce the **VIOS Access Control Policy** on

- **a) Network: VIOS Ethernet device drivers acting on behalf of a group of LPAR partitions sharing a virtual network and VIOS Ethernet adapter device drivers (where either one can be the subject and the other the object) and the operations among subjects and objects as covered by the policy;**
- **b) Volumes: VIOS SCSI device drivers acting on behalf of LPAR partitions as subjects with Logical Volumes and Physical Volumes as objects and the operations among subjects and objects as covered by the policy.**

6.1.5.2 Subset access control [ST] (VIOS only) (FDP_ACC.1(VRBAC))

FDP_ACC.1.1 The TSF shall enforce the **VIOS Role-based Access Control (VRBAC) Policy** on

- **a) Subjects: Processes acting on the behalf of users;**
- **b) Objects:**
  - **i. Persistent Storage Objects of the following type**
    1. **Ordinary files;**
    2. **Directories;**
    3. **Device special files;**
    4. **UNIX Domain socket special files;**
    5. **Named pipes;**
  - **ii. Transient Storage Objects of the following type**
    1. **Message queues;**
    2. **SysV semaphores;**
    3. **Shared memory segments;**
    4. **TCP ports;**

c) Operations: All operations among subjects and objects covered by this policy.

6.1.5.3 Security attribute based access control [ST] (VIOS only) (FDP_ACF.1(VIOS))

FDP_ACF.1.1 *For VIOS, the* ~~The~~ TSF shall enforce the **VIOS Access Control Policy** to objects based on the following:

a) Network: A VIOS Ethernet device driver acting on behalf of a group of LPAR partitions sharing a virtual network and a VIOS Ethernet adapter device driver (where either one can be the subject and the other the object) where both have the same security attribute of an inter-LPAR communication channel;

b) Volumes: A logical volume or physical volume (object) can only be mapped to (accessed by) one VIOS SCSI device driver acting on behalf of an LPAR partition (subject) and the security attribute is the mapping table entry that maps the subject and object together.

FDP_ACF.1.2 *For VIOS, the* ~~The~~ TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

a) Network: If a VIOS Ethernet device driver acting on behalf of a group of LPAR partitions sharing a virtual network is mapped via an inter-LPAR communication channel to a VIOS Ethernet adapter device driver, then the device drivers can exchange untagged packets; otherwise, access is denied;

b) Volumes: If the logical volume or physical volume is mapped to a VIOS SCSI device driver acting on behalf of an LPAR partition, then the device driver can access the logical volume or physical volume, respectively; otherwise, access is denied.

FDP_ACF.1.3 *For VIOS, the* ~~The~~ TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **none**.

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

6.1.5.4 Security attribute based access control [ST] (VIOS only) (FDP_ACF.1(VRBAC))

FDP_ACF.1.1 The TSF shall enforce the **VIOS Role-based Access Control (VRBAC) Policy** to objects based on the following:

a) Subjects: Processes acting on the behalf of users;

i. Attributes:

1. Subject identity;

2. Role(s) which can invoke the subject;

b) Authorized users;

i. Attributes:

1. User identity;

2. **Authorized role(s) for the user;**

c) **Objects: As defined in FDP_ACC.1(VRBAC);**

i. **Attributes:**

1. **Object identity;**
2. **Operations permitted on the objects for various roles.**

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

a) **The subject invoking the operation on an object is assigned to a role whose privilege set includes the operation on the object.**

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

a) **Allow an access operation by a subject on an object only if the user associated with the subject belongs to a role that permits the access operation on the object.**

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

a) **The user associated with the subject not belonging to any role that permits the requested access operation on the object.**

6.1.5.5 User attribute definition [ST] (VIOS only) (FIA_ATD.1(VIOS))

FIA_ATD.1.1 *For VIOS, the* ~~The~~ TSF shall maintain the following list of security attributes belonging to individual users:

a) **User identifier;**
b) **Group memberships;**
c) **Security-relevant roles;**
d) **Authentication data.**

6.1.5.6 Verification of secrets [ST] (VIOS only) (FIA_SOS.1(VIOS))

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

6.1.5.7 User authentication before any action [ST] (VIOS only) (FIA_UAU.2)

FIA_UAU.2.1 *For VIOS, the* ~~The~~ TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

6.1.5.8 Protected authentication feedback [ST] (VIOS only) (FIA_UAU.7(VIOS))

FIA_UAU.7.1 *For VIOS, the* ~~The~~ TSF shall provide only **obscured feedback** to the user while the authentication is in progress.

6.1.5.9 User identification before any action [ST] (VIOS only) (FIA_UID.2(VIOS))

FIA_UID.2.1 *For VIOS, the* ~~The~~ TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

6.1.5.10 User-subject binding [ST] (VIOS only) (FIA_USB.1(VIOS))

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

a) **User identity;**

b) **Group memberships;**

c) **Security-relevant roles.**

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

a) **Upon successful identification and authentication, the real user identifier, the effective user identifier and login user identifier shall be those specified in the user entry for the user that has authenticated successfully.**

b) **Upon successful identification and authentication, the real group identifier, and the effective group identifier shall be those specified via the group membership attribute in the user entry.**

FIA_USB.1.3 *For VIOS, the* ~~The~~ TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users:

a) **The effective userID of a user can be changed by the use of an executable with the setuid bit set. In this case the program is executed with the effective userID of the program owner. Access rights are then evaluated using the effective userID of the program owner. The login userID is not changed with this process.**

b) **The effective userID of a user can be changed by the su command. In this case the effective userID of the user is changed to the user specified in the su command (provided authentication is successful). The login userID remains unchanged.**

c) **The effective groupID of a user can be changed by the use of an executable with the setgid bit set. In this case the program is executed with the effective groupID of the program owning group. Access rights are then evaluated using the effective groupID of the program owner. The login userID is not changed with this process.**

6.1.5.11 Management of security attributes [ST] (VIOS only) (FMT_MSA.1(VIOS))

FMT_MSA.1.1 *For VIOS, the* ~~The~~ TSF shall enforce the **VIOS Access Control Policy** to restrict the ability to **modify** the security attributes

a) **For Network: mapping of Ethernet device drivers acting on behalf of a group of LPAR partitions sharing a virtual network to Ethernet adapter device drivers;**

b) For Volumes: mapping SCSI device drivers acting on behalf of LPAR partitions to logical volumes and physical volumes

to **the System Administrator role**.

6.1.5.12 Management of object security attributes [ST] (VIOS only) (FMT_MSA.1(VRBAC-ADM))

FMT_MSA.1.1 The TSF shall enforce the **VIOS Role-based Access Control (VRBAC) Policy** to restrict the ability to **modify** the security attributes **of objects** to **object owners and the set of VRBAC administrative roles**.

6.1.5.13 Management of object security attributes [ST] (VIOS only) (FMT_MSA.1(VRBAC-AUTH))

FMT_MSA.1.1 The TSF shall enforce the **VIOS Role-based Access Control (VRBAC) Policy** to restrict the ability to **modify, delete, create instances of** the security attributes **User Role Authorizations** to **a set of VRBAC administrative roles**.

6.1.5.14 Management of object security attributes [ST] (VIOS only) (FMT_MSA.1(VRBAC-DFLT))

FMT_MSA.1.1 The TSF shall enforce the **VIOS Role-based Access Control (VRBAC) Policy** to restrict the ability to **modify, create** the security attributes **Default Active Role Set** to **a set of VRBAC administrative roles**.

6.1.5.15 Management of object security attributes [ST] (VIOS only) (FMT_MSA.1(VRBAC-USR))

FMT_MSA.1.1 The TSF shall enforce the **VIOS Role-based Access Control (VRBAC) Policy** to restrict the ability to **modify the composition of** the *session* security attributes **Active Role set for a user** to **the session owner**.

6.1.5.16 Secure security attributes [ST] (VIOS only) (FMT_MSA.2(VRBAC))

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

6.1.5.17 Static attribute initialisation [ST] (VIOS only) (FMT_MSA.3(VIOS))

FMT_MSA.3.1 *For VIOS, the* ~~The~~ TSF shall enforce the **VIOS Access Control Policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 *For VIOS, the* ~~The~~ TSF shall allow ~~the~~ **none** to specify alternative initial values to override the default values when an object or information is created.

6.1.5.18 Static attribute initialisation [ST] (VIOS only) (FMT_MSA.3(VRBAC))

FMT_MSA.3.1 The TSF shall enforce the **VIOS Role-based Access Control (VRBAC) Policy** to provide **administrative user defined** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the **set of VRBAC administrative roles** to specify alternative initial values to override the default values when an object or information is created.

6.1.5.19 Management of TSF data [ST] (VIOS only) (FMT_MTD.1(VIOS-ADI))

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

6.1.5.20 Management of TSF data [ST] (VIOS only) (FMT_MTD.1(VIOS-ADM))

FMT_MTD.1.1 *For VIOS, the* ~~The~~ TSF shall restrict the ability to **modify** the **authentication data** to

a) Authorized administrators;

b) All users can modify their own authentication data.

6.1.5.21 Management of TSF data [ST] (VIOS only) (FMT_MTD.1(VIOS-NV))

FMT_MTD.1.1 *For VIOS, the* ~~The~~ TSF shall restrict the ability to **create, modify, and delete** the

a) For Network: mapping of VIOS Ethernet adapter device drivers to VIOS Ethernet device drivers acting on behalf of groups of LPAR partitions sharing virtual networks;

b) For Volumes: mappings of logical volumes and physical volumes to VIOS SCSI device drivers acting on behalf of LPAR partitions

to **authorized administrators**.

6.1.5.22 Management of TSF data [ST] (VIOS only) (FMT_MTD.1(VIOS-SA))

FMT_MTD.1.1 *For VIOS, the* ~~The~~ TSF shall restrict the ability to **initialize, modify** the **user security attributes defined in FIA_ATD.1(VIOS) except for authentication data** to **authorized administrators**.

6.1.5.23 Management of TSF data [ST] (VIOS only) (FMT_MTD.1(VRBAC))

FMT_MTD.1.1 The TSF shall restrict the ability to **modify, create** the **TSF data**

a) All user passwords;

b) Role definition and role attributes;

c) Role hierarchies (by assigning one or more roles to other roles);

d) Constraints among role relationships

to **a set of VRBAC administrative roles**.

6.1.5.24 Secure TSF data [ST] (VIOS only) (FMT_MTD.3(VRBAC))

FMT_MTD.3.1 The TSF shall ensure that only secure values are accepted for **role definitions, role hierarchies, and role relationship constraints**.

6.1.5.25 Revocation [ST] (VIOS only) (FMT_REV.1(VIOS))

FMT_REV.1.1 *For VIOS, the* ~~The~~ TSF shall restrict the ability to revoke

a) Authentication data;

b) Group memberships;

c) Security-relevant roles

associated with the **users** under the control of the TSF to **authorized administrators**.

FMT_REV.1.2 *For VIOS, the* ~~The~~ TSF shall enforce the rules

a) The immediate revocation of security-relevant authorizations.

b) Revocations/modifications made by an administrator to security attributes of a user, such as the user identifier, user name, user group(s), user password, or user login shell, shall be effective the next time the user logs in.

6.1.5.26 Specification of management functions [ST] (VIOS only) (FMT_SMF.1(VIOS))

FMT_SMF.1.1 *For VIOS, the* ~~The~~ TSF shall be capable of performing the following management functions:

a) User attribute management;

b) Authentication data management;

c) VIOS network and volume management.

6.1.5.27 Security roles [ST] (VIOS only) (FMT_SMR.1)

FMT_SMR.1.1 *For VIOS, the* ~~The~~ TSF shall maintain the roles

a) User role with the following rights:

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

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

iii. Other rights as assigned by an authorized administrator via the VRBAC mechanism;

b) The set of VRBAC administrative roles.

FMT_SMR.1.2 *For VIOS, the* ~~The~~ TSF shall be able to associate users with roles.

6.1.5.28 Limitation on scope of selectable attributes [ST] (VIOS only) (FTA_LSA.1(VRBAC))

FTA_LSA.1.1 The TSF shall restrict the scope of the session security attributes **active role set for the user**, based on **the set of authorized roles for the user**.

6.1.5.29 TOE session establishment [ST] (VIOS only) (FTA_TSE.1(VRBAC))

FTA_TSE.1.1 The TSF shall be able to deny session establishment based on **the default active role set for the user being empty**.

6.2 Security Functional Requirements Rationale

6.2.1 Coverage

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

Security Functional Requirements	Objectives
FAU_GEN.1(BASE)	[OSPP]_O.AUDITING
FAU_GEN.2	[OSPP]_O.AUDITING
FAU_SAR.1	[OSPP]_O.AUDITING
FAU_SAR.2	[OSPP]_O.AUDITING
FAU_SAR.3(BASE)	[OSPP]_O.AUDITING
FAU_SEL.1(BASE)	[OSPP]_O.AUDITING
FAU_STG.1	[OSPP]_O.AUDITING
FAU_STG.3	[OSPP]_O.AUDITING
FAU_STG.4	[OSPP]_O.AUDITING
FCS_CKM.1(SYM)	[OSPP-CRYPTO]_O.CRYPTO.BASIC, [OSPP]_O.CRYPTO.NET, [OSPP]_O.DISCRETIONARY.ACCESS
FCS_CKM.1(RSA)	[OSPP-CRYPTO]_O.CRYPTO.BASIC, [OSPP]_O.CRYPTO.NET, [OSPP]_O.DISCRETIONARY.ACCESS
FCS_CKM.1(DSA)	[OSPP-CRYPTO]_O.CRYPTO.BASIC, [OSPP]_O.CRYPTO.NET
FCS_CKM.2(NET)	[OSPP]_O.CRYPTO.NET, [OSPP]_O.TRUSTED_CHANNEL
FCS_CKM.4	[OSPP-CRYPTO]_O.CRYPTO.BASIC, [OSPP]_O.CRYPTO.NET, [OSPP]_O.DISCRETIONARY.ACCESS, [OSPP]_O.TRUSTED_CHANNEL
FCS_COP.1(NET)	[OSPP]_O.CRYPTO.NET, [OSPP]_O.TRUSTED_CHANNEL
FCS_COP.1(CRYPTO-ENC)	[OSPP-CRYPTO]_O.CRYPTO.BASIC
FCS_COP.1(CRYPTO-MD)	[OSPP-CRYPTO]_O.CRYPTO.BASIC
FCS_COP.1(CRYPTO-SGN)	[OSPP-CRYPTO]_O.CRYPTO.BASIC
FCS_COP.1(CLIC-ENC)	[OSPP]_O.CRYPTO.NET, [OSPP]_O.DISCRETIONARY.ACCESS, [OSPP]_O.TRUSTED_CHANNEL

Security Functional Requirements	Objectives
FCS_COP.1(CLIC-MD)	[OSPP]_O.CRYPTO.NET, [OSPP-IV]_O.INTEGRITY.TSF, [OSPP-IV]_O.INTEGRITY.USERDATA, [OSPP]_O.TRUSTED_CHANNEL
FCS_COP.1(CLIC-SGN)	[OSPP]_O.CRYPTO.NET, [OSPP]_O.DISCRETIONARY.ACCESS, [OSPP-IV]_O.INTEGRITY.TSF
FCS_RNG.1(CLIC)	[OSPP]_O.CRYPTO.NET, [OSPP]_O.DISCRETIONARY.ACCESS
FDP_ACC.1(PSO-AIXC)	[OSPP]_O.DISCRETIONARY.ACCESS
FDP_ACC.1(PSO-NFS)	[OSPP]_O.DISCRETIONARY.ACCESS
FDP_ACC.1(TSO)	[OSPP]_O.SUBJECT.COM
FDP_ACC.1(AUTH)	[ST]_O.ROLE.AUTHORIZATIONS
FDP_ACC.1(RBAC)	[ST]_O.ROLE
FDP_ACC.1(TCB)	[ST]_O.TCB.ACCESS
FDP_ACC.1(TCP)	[OSPP]_O.SUBJECT.COM
FDP_ACC.2(VIRT)	[OSPP-VIRT]_O.COMP.RESOURCE_ACCESS
FDP_ACF.1(PSO-AIXC)	[OSPP]_O.DISCRETIONARY.ACCESS
FDP_ACF.1(PSO-NFS)	[OSPP]_O.DISCRETIONARY.ACCESS
FDP_ACF.1(TSO)	[OSPP]_O.SUBJECT.COM
FDP_ACF.1(VIRT)	[OSPP-VIRT]_O.COMP.RESOURCE_ACCESS
FDP_ACF.1(AUTH)	[ST]_O.ROLE.AUTHORIZATIONS
FDP_ACF.1(RBAC)	[ST]_O.ROLE
FDP_ACF.1(TCB)	[ST]_O.TCB.ACCESS
FDP_ACF.1(TCP)	[OSPP]_O.SUBJECT.COM
FDP_ETC.2(VIRT)	[OSPP-VIRT]_O.COMP.INFO_FLOW_CTRL, [OSPP-VIRT]_O.COMP.RESOURCE_ACCESS
FDP_IFC.2(NI)	[OSPP]_O.NETWORK.FLOW
FDP_IFC.2(VIRT)	[OSPP-VIRT]_O.COMP.INFO_FLOW_CTRL
FDP_IFF.1(NI)	[OSPP]_O.NETWORK.FLOW
FDP_IFF.1(VIRT)	[OSPP-VIRT]_O.COMP.INFO_FLOW_CTRL

Security Functional Requirements	Objectives
FDP_ITC.2(BASE)	[OSPP]_O.DISCRETIONARY.ACCESS, [OSPP]_O.NETWORK.FLOW, [OSPP]_O.SUBJECT.COM
FDP_ITC.2(VIRT)	[OSPP-VIRT]_O.COMP.INFO_FLOW_CTRL, [OSPP-VIRT]_O.COMP.RESOURCE_ACCESS
FDP_RIP.2	[OSPP]_O.AUDITING, [OSPP-CRYPTO]_O.CRYPTO.BASIC, [OSPP]_O.CRYPTO.NET, [OSPP]_O.DISCRETIONARY.ACCESS, [OSPP]_O.I&A, [OSPP]_O.NETWORK.FLOW, [OSPP]_O.SUBJECT.COM
FDP_RIP.3	[OSPP]_O.AUDITING, [OSPP-CRYPTO]_O.CRYPTO.BASIC, [OSPP]_O.CRYPTO.NET, [OSPP]_O.DISCRETIONARY.ACCESS, [OSPP]_O.I&A, [OSPP]_O.NETWORK.FLOW, [OSPP]_O.SUBJECT.COM
FDP_RIP.4	[ST]_O.DISK.OVERWRITTEN
FDP_SDI.2(IV)	[OSPP-IV]_O.INTEGRITY.ACTION, [OSPP-IV]_O.INTEGRITY.USERDATA
FIA_AFL.1	[OSPP]_O.I&A
FIA_ATD.1(HU)	[OSPP]_O.I&A, [ST]_O.ROLE
FIA_ATD.1(TU)	[OSPP]_O.NETWORK.FLOW
FIA_SOS.1(BASE)	[OSPP]_O.I&A
FIA_UAU.1	[OSPP]_O.I&A
FIA_UAU.5	[OSPP]_O.I&A
FIA_UAU.7(BASE)	[OSPP]_O.I&A
FIA_UID.2(BASE)	[OSPP]_O.I&A, [OSPP]_O.NETWORK.FLOW
FIA_UID.2(VIRT)	[OSPP-VIRT]_O.COMP.IDENT
FIA_USB.2	[OSPP]_O.I&A, [ST]_O.ROLE
FMT_MSA.1(PSO-AIXC)	[OSPP]_O.MANAGE
FMT_MSA.1(PSO-NFS)	[OSPP]_O.MANAGE
FMT_MSA.1(TSO)	[OSPP]_O.MANAGE

Security Functional Requirements	Objectives
FMT_MSA.1(VIRT-CACP)	[OSPP-VIRT]_O.COMP.RESOURCE_ACCESS
FMT_MSA.1(VIRT-CIFCP)	[OSPP-VIRT]_O.COMP.INFO_FLOW_CTRL
FMT_MSA.1(AUTH)	[ST]_O.ROLE.AUTHORIZATIONS
FMT_MSA.1(RBAC-ADM)	[ST]_O.ROLE
FMT_MSA.1(RBAC-AUTH)	[ST]_O.ROLE
FMT_MSA.1(RBAC-DFLT)	[ST]_O.ROLE
FMT_MSA.1(RBAC-USR)	[ST]_O.ROLE
FMT_MSA.1(TCB)	[ST]_O.TCB.ACCESS
FMT_MSA.1(TCP)	[OSPP]_O.MANAGE
FMT_MSA.2(RBAC)	[ST]_O.ROLE, [ST]_O.ROLE.AUTHORIZATIONS
FMT_MSA.3(PSO-AIXC)	[OSPP]_O.MANAGE
FMT_MSA.3(PSO-NFS)	[OSPP]_O.MANAGE
FMT_MSA.3(TSO)	[OSPP]_O.MANAGE
FMT_MSA.3(NI)	[OSPP]_O.MANAGE
FMT_MSA.3(VIRT-CACP)	[OSPP-VIRT]_O.COMP.RESOURCE_ACCESS
FMT_MSA.3(VIRT-CIFCP)	[OSPP-VIRT]_O.COMP.INFO_FLOW_CTRL
FMT_MSA.3(AUTH)	[ST]_O.ROLE.AUTHORIZATIONS
FMT_MSA.3(RBAC)	[ST]_O.ROLE
FMT_MSA.3(TCB)	[ST]_O.TCB.ACCESS
FMT_MSA.3(TCP)	[OSPP]_O.MANAGE
FMT_MSA.4(PSO)	[OSPP]_O.MANAGE
FMT_MTD.1(AE)	[OSPP]_O.MANAGE
FMT_MTD.1(AS)	[OSPP]_O.MANAGE
FMT_MTD.1(AT)	[OSPP]_O.MANAGE
FMT_MTD.1(AF)	[OSPP]_O.MANAGE
FMT_MTD.1(NI)	[OSPP]_O.MANAGE
FMT_MTD.1(IAT)	[OSPP]_O.MANAGE
FMT_MTD.1(IAF)	[OSPP]_O.MANAGE

Security Functional Requirements	Objectives
FMT_MTD.1(IAU)	[OSPP]_O.MANAGE
FMT_MTD.1(AM-AP)	[OSPP-AM]_O.ROLE.APPROVE
FMT_MTD.1(AM-MR)	[OSPP-AM]_O.ROLE.MGMT
FMT_MTD.1(AM-MD)	[OSPP-AM]_O.ROLE.DELEGATE
FMT_MTD.1(AM-MA)	[OSPP-AM]_O.ROLE.APPROVE
FMT_MTD.1(IV-ACT)	[OSPP-IV]_O.INTEGRITY.MANAGE
FMT_MTD.1(IV-TSF)	[OSPP-IV]_O.INTEGRITY.MANAGE
FMT_MTD.1(IV-USR)	[OSPP-IV]_O.INTEGRITY.MANAGE
FMT_MTD.1(VIRT-COMP)	[OSPP-VIRT]_O.COMP.INFO_FLOW_CTRL, [OSPP-VIRT]_O.COMP.RESOURCE_ACCESS
FMT_MTD.1(PRIVS)	[OSPP]_O.MANAGE
FMT_MTD.1(RBAC)	[ST]_O.ROLE.HIERARCHY
FMT_MTD.3(RBAC)	[ST]_O.ROLE.HIERARCHY
FMT_REV.1(OBJ)	[OSPP]_O.MANAGE, [ST]_O.ROLE
FMT_REV.1(USR)	[OSPP]_O.MANAGE
FMT_SMF.1(BASE)	[OSPP]_O.MANAGE
FMT_SMR.2	[OSPP]_O.MANAGE, [ST]_O.ROLE, [ST]_O.ROLE.SEP_DUTY
FPT_FLS.1(RBAC)	[ST]_O.ROLE.CONSISTENT_DB
FPT_FLS.1(SED)	[ST]_O.STACK.NO_EXEC
FPT_RCV.1	[ST]_O.ROLE.CONSISTENT_DB
FPT_RCV.4	[ST]_O.ROLE.CONSISTENT_DB
FPT_STM.1	[OSPP]_O.AUDITING
FPT_TDC.1(BASE)	[OSPP]_O.DISCRETIONARY.ACCESS, [OSPP]_O.NETWORK.FLOW, [OSPP]_O.SUBJECT.COM
FPT_TDC.1(VIRT)	[OSPP-VIRT]_O.COMP.INFO_FLOW_CTRL, [OSPP-VIRT]_O.COMP.RESOURCE_ACCESS
FPT_TIM.1(IV)	[OSPP-IV]_O.INTEGRITY.ACTION, [OSPP-IV]_O.INTEGRITY.TSF
FPT_TST.1	[ST]_O.ROLE.CONSISTENT_DB

Security Functional Requirements	Objectives
FRU_FLT.2	[ST]_O.STACK.NO_EXEC
FTA_LSA.1(RBAC)	[ST]_O.ROLE
FTA_SSL.1	[OSPP]_O.I&A
FTA_SSL.2	[OSPP]_O.I&A
FTA_TSE.1(RBAC)	[ST]_O.ROLE
FTP_ITC.1	[OSPP]_O.TRUSTED_CHANNEL
FAU_GEN.1(LS)	[OSPP]_O.AUDITING
FAU_SAR.3(LS)	[OSPP]_O.AUDITING
FAU_SEL.1(LS)	[OSPP]_O.AUDITING
FDP_ETC.2(LS)	[OSPP-LS]_O.LS.CONFIDENTIALITY, [OSPP-LS]_O.LS.PRINT, [ST]_O.TN.ACCESS
FDP_IFC.1(MIC)	[ST]_O.MANDATORY_INTEGRITY
FDP_IFC.1(TN)	[ST]_O.TN.ACCESS
FDP_IFC.2(LS)	[OSPP-LS]_O.LS.CONFIDENTIALITY
FDP_IFF.2(MIC)	[ST]_O.MANDATORY_INTEGRITY
FDP_IFF.2(TN)	[ST]_O.TN.ACCESS
FDP_IFF.2(LS)	[OSPP-LS]_O.LS.CONFIDENTIALITY
FDP_ITC.1(LS)	[OSPP-LS]_O.LS.CONFIDENTIALITY, [OSPP-LS]_O.LS.LABEL, [ST]_O.TN.ACCESS
FDP_ITC.2(LS)	[OSPP-LS]_O.LS.CONFIDENTIALITY, [OSPP-LS]_O.LS.LABEL, [ST]_O.TN.ACCESS
FIA_ATD.1(LS)	[OSPP-LS]_O.LS.LABEL, [ST]_O.TN.ACCESS
FIA_ATD.1(LSX)	[ST]_O.MANDATORY_INTEGRITY, [ST]_O.TN.ACCESS
FIA_USB.1(LS)	[OSPP-LS]_O.LS.LABEL, [ST]_O.TN.ACCESS
FIA_USB.1(LSX)	[ST]_O.MANDATORY_INTEGRITY, [ST]_O.TN.ACCESS
FMT_MSA.1(LS)	[OSPP-LS]_O.LS.LABEL
FMT_MSA.1(MIC)	[ST]_O.MANDATORY_INTEGRITY

Security Functional Requirements	Objectives
FMT_MSA.1(TN)	[ST]_O.TN.ACCESS
FMT_MSA.3(LS)	[OSPP-LS]_O.LS.LABEL
FMT_MSA.3(MIC)	[ST]_O.MANDATORY_INTEGRITY
FMT_MSA.3(TN)	[ST]_O.TN.ACCESS
FPT_TDC.1(LS)	[OSPP-LS]_O.LS.CONFIDENTIALITY, [OSPP-LS]_O.LS.LABEL, [ST]_O.TN.ACCESS
FDP_ACC.1(VIOS)	[ST]_O.VIOS.NET.PROTECTED, [ST]_O.VIOS.VOL.PROTECTED
FDP_ACC.1(VRBAC)	[ST]_O.VIOS.ROLE
FDP_ACF.1(VIOS)	[ST]_O.VIOS.NET.PROTECTED, [ST]_O.VIOS.VOL.PROTECTED
FDP_ACF.1(VRBAC)	[ST]_O.VIOS.ROLE
FIA_ATD.1(VIOS)	[ST]_O.VIOS.I&A, [ST]_O.VIOS.ROLE
FIA_SOS.1(VIOS)	[ST]_O.VIOS.I&A
FIA_UAU.2	[ST]_O.VIOS.I&A
FIA_UAU.7(VIOS)	[ST]_O.VIOS.I&A
FIA_UID.2(VIOS)	[ST]_O.VIOS.I&A
FIA_USB.1(VIOS)	[ST]_O.VIOS.I&A, [ST]_O.VIOS.ROLE
FMT_MSA.1(VIOS)	[ST]_O.VIOS.MANAGE, [ST]_O.VIOS.NET.PROTECTED, [ST]_O.VIOS.VOL.PROTECTED
FMT_MSA.1(VRBAC-ADM)	[ST]_O.VIOS.ROLE
FMT_MSA.1(VRBAC-AUTH)	[ST]_O.VIOS.ROLE
FMT_MSA.1(VRBAC-DFLT)	[ST]_O.VIOS.ROLE
FMT_MSA.1(VRBAC-USR)	[ST]_O.VIOS.ROLE
FMT_MSA.2(VRBAC)	[ST]_O.VIOS.ROLE
FMT_MSA.3(VIOS)	[ST]_O.VIOS.MANAGE, [ST]_O.VIOS.NET.PROTECTED, [ST]_O.VIOS.VOL.PROTECTED
FMT_MSA.3(VRBAC)	[ST]_O.VIOS.ROLE
FMT_MTD.1(VIOS-ADI)	[ST]_O.VIOS.MANAGE

Security Functional Requirements	Objectives
FMT_MTD.1(VIOS-ADM)	[ST]_O.VIOS.MANAGE
FMT_MTD.1(VIOS-NV)	[ST]_O.VIOS.MANAGE
FMT_MTD.1(VIOS-SA)	[ST]_O.VIOS.MANAGE
FMT_MTD.1(VRBAC)	[ST]_O.VIOS.ROLE.HIERARCHY
FMT_MTD.3(VRBAC)	[ST]_O.VIOS.ROLE.HIERARCHY
FMT_REV.1(VIOS)	[ST]_O.VIOS.MANAGE
FMT_SMF.1(VIOS)	[ST]_O.VIOS.MANAGE
FMT_SMR.1	[ST]_O.VIOS.MANAGE, [ST]_O.VIOS.NET.PROTECTED, [ST]_O.VIOS.ROLE, [ST]_O.VIOS.ROLE.SEP_DUTY, [ST]_O.VIOS.VOL.PROTECTED
FTA_LSA.1(VRBAC)	[ST]_O.VIOS.ROLE
FTA_TSE.1(VRBAC)	[ST]_O.VIOS.ROLE

Table 14: Mapping of security functional requirements to security objectives

6.2.2 Sufficiency

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

Security objectives	Rationale
[OSPP]_O.AUDITING	The events to be audited are defined in [FAU_GEN.1(BASE), FAU_GEN.1(LS)] and are associated with the identity of the user that caused the event [FAU_GEN.2]. Authorized users are provided the capability to read the audit records [FAU_SAR.1], while all other users are denied access to the audit records [FAU_SAR.2]. The TOE provides the capability to search, sort, and order audit data [FAU_SAR.3(BASE), FAU_SAR.3(LS)]. The authorized user must have the capability to specify which audit records are generated [FAU_SEL.1(BASE), FAU_SEL.1(LS)]. The TOE prevents the audit log from being modified or deleted [FAU_STG.1] and ensures that the audit log is not lost due to resource shortage [FAU_STG.3, FAU_STG.4]. To support auditing, the TOE is able to maintain proper time stamps [FPT_STM.1]. The protection of reused resources ensures that no data leaks from other protected sources [FDP_RIP.2, FDP_RIP.3].
[OSPP]_O.CRYPTO.NET	The cryptographically-protected network protocol (i.e., IPsec) [FCS_COP.1(NET)] is supported by the generation of symmetric keys [FCS_CKM.1(SYM)], as well as asymmetric keys [FCS_CKM.1(RSA), FCS_CKM.1(DSA)]. As part of the cryptographic network protocol (i.e.,

Security objectives	Rationale
	IPsec), the TOE securely exchanges the symmetric key with a remote trusted IT system [FCS_CKM.2(NET)]. The TOE ensures that all keys are zeroized upon de-allocation [FCS_CKM.4]. The cryptographic algorithms used by IPsec are specified by [FCS_COP.1(CLIC-ENC), FCS_COP.1(CLIC-MD), FCS_COP.1(CLIC-SGN)]. The random number generator is specified by [FCS_RNG.1(CLIC)]. The protection of reused resources ensures that no data leaks from other protected sources [FDP_RIP.2, FDP_RIP.3].
[OSPP]_O.DISCRETIONARY.ACCESS	The TSF must control access to resources based on the identity of users that are allowed to specify which resources they want to access for storing their data. The access control policy must have a defined scope of control [FDP_ACC.1(PSO-AIXC), FDP_ACC.1(PSO-NFS)]. The rules for the access control policy are defined [FDP_ACF.1(PSO-AIXC), FDP_ACF.1(PSO-NFS)]. When import of user data is allowed, the TOE must ensure that user data security attributes required by the access control policy are correctly interpreted [FDP_ITC.2(BASE), FPT_TDC.1(BASE)]. EFS generates per-file AES keys [FCS_CKM.1(SYM)] used to encrypt the files and per-account RSA keys [FCS_CKM.1(RSA)] used to encrypt the AES keys using the CLiC random number generator [FCS_RNG.1(CLIC)]. EFS uses AES and RSA encryption and decryption [FCS_COP.1(CLIC-ENC), FCS_COP.1(CLIC-SGN)] to control access to filesystem objects. EFS includes key destruction [FCS_CKM.4]. The protection of reused resources ensures that no data leaks from other protected sources [FDP_RIP.2, FDP_RIP.3].
[OSPP]_O.NETWORK.FLOW	The network information flow control mechanism controls the information flowing between different entities [FDP_IFC.2(NI)]. The TOE implements a rule-set governing the information flow [FDP_IFF.1(NI)]. To facilitate the information flow control, the information must be identified [FIA_UID.2(BASE)] based on "individual technical user" security attributes the TOE can maintain [FIA_ATD.1(TU)]. The TOE must ensure that security attributes of the network data required by the information flow control policy are correctly interpreted [FDP_ITC.2(BASE), FPT_TDC.1(BASE)]. The protection of reused resources ensures that no data leaks from other protected sources [FDP_RIP.2, FDP_RIP.3].
[OSPP]_O.SUBJECT.COM	The TSF must control the exchange of data using transient storage objects between subjects based on the identity of users. The access control policy must have a defined scope of control [FDP_ACC.1(TSO), FDP_ACC.1(TCP)]. The rules for the access control policy are defined [FDP_ACF.1(TSO), FDP_ACF.1(TCP]. When import of user data is allowed, the TOE must ensure that user data security attributes required by the access control policy are correctly interpreted [FDP_ITC.2(BASE), FPT_TDC.1(BASE)]. The protection of reused resources ensures that no data leaks from other protected sources [FDP_RIP.2, FDP_RIP.3].

Security objectives	Rationale
[OSPP]_O.I&A	The TSF must ensure that only authorized users gain access to the TOE and its resources. Users authorized to access the TOE must use an identification and authentication process [FIA_UID.2(BASE), FIA_UAU.1]. Multiple I&A mechanisms are allowed as specified in [FIA_UAU.5]. To ensure authorized access to the TOE, authentication data is protected [FIA_ATD.1(HU), FIA_UAU.7(BASE)]. Proper authorization for subjects acting on behalf of users is also ensured [FIA_USB.2]. The appropriate strength of the authentication mechanism is ensured [FIA_SOS.1(BASE)]. To support the strength of authentication methods, the TOE is capable of identifying and reacting to unsuccessful authentication attempts [FIA_AFL.1]. In addition, user-initiated and TSF-initiated session locking [FTA_SSL.1, FTA_SSL.2] protect the authenticated user's session. The protection of reused resources ensures that no data leaks from other protected sources [FDP_RIP.2, FDP_RIP.3] are present.
[OSPP]_O.MANAGE	The TOE provides management interfaces globally defined in [FMT_SMF.1(BASE)] for: • the access control policies [FMT_MSA.1(PSO-AIXC), FMT_MSA.1(PSO-NFS), FMT_MSA.1(TSO), FMT_MSA.1(TCP), FMT_MSA.3(PSO-AIXC), FMT_MSA.3(PSO-NFS), FMT_MSA.3(TSO), FMT_MSA.3(TCP)]; • the information flow control policy [FMT_MSA.3(NI), FMT_MTD.1(NI)]; • the auditing aspects [FMT_MTD.1(AE), FMT_MTD.1(AS), FMT_MTD.1(AT), FMT_MTD.1(AF)]; • the identification and authentication aspects [FMT_MTD.1(IAT), FMT_MTD.1(IAF), FMT_MTD.1(IAU)]; • the application, device, and WPAR privilege aspects [FMT_MTD.1(PRIVS)]. Persistently stored user data is stored either in hierarchical or relational fashion, which implies an inheritance of security attributes from parent object [FMT_MSA.4(PSO)]. The rights management for the different management aspects is defined with [FMT_SMR.2]. The management interfaces for the revocation of user and object attributes is provided with [FMT_REV.1(OBJ) and FMT_REV.1(USR)].
[OSPP]_O.TRUSTED_CHANNEL	The TOE provides a trusted channel protecting communication between a remote trusted IT system and itself [FTP_ITC.1]. Network communication protection between the TOE and NFSv4 uses a combination of NAS and CLiC. This communication is specified by [FCS_CKM.2(NET), FCS_CKM.4, FCS_COP.1(NET), FCS_COP.1(CLIC-ENC), FCS_COP.1(CLIC-MD)].
[OSPP-AM]_O.ROLE.DELEGATE	The delegation of roles is defined and specified in [FMT_MTD.1(AM-MD)].
[OSPP-AM]_O.ROLE.MGMT	The definition and management of rights based on roles is defined in [FMT_MTD.1(AM-MR)].

Security objectives	Rationale
[OSPP-AM]_O.ROLE.APPROVE	The approval mechanism for roles is defined with [FMT_MTD.1(AM-AP)], supported by management of the approval mechanism, i.e., specification of which roles can approve which operations [FMT_MTD.1(AM-MA)].
[OSPP-CRYPTO]_O.CRYPTO.BASIC	The cryptographic mechanisms are defined as follows: • Symmetric [FCS_CKM.1(SYM)] and asymmetric [FCS_CKM.1(RSA), FCS_CKM.1(DSA)] key generation – the SFRs are defined in the OSPP base and are supportive to [OSPP-CRYPTO]_O.CRYPTO.BASIC • Key destruction is provided with [FCS_CKM.4] defined in the OSPP base – this SFR is supportive to [OSPP-CRYPTO]_O.CRYPTO.BASIC • Symmetric and asymmetric encryption and decryption [FCS_COP.1(CRYPTO-ENC)] • Signature generation and verification [FCS_COP.1(CRYPTO-SGN)] • Message digest generation [FCS_COP.1(CRYPTO-MD)]
[OSPP-IV]_O.INTEGRITY.TSF	The integrity verification mechanism for validating TSF data and TSF code is defined with [FPT_TIM.1(IV)]. The message digests used by the integrity verification mechanism for validating TSF data and TSF code are defined in [FCS_COP.1(CLIC-MD)]. The message digest values generated during the creation of the AIX installation image are signed by IBM and can be verified by each AIX instance using [FCS_COP.1(CLIC-SGN)].
[OSPP-IV]_O.INTEGRITY.USERDATA	The integrity verification mechanism for validating user data is defined with [FDP_SDI.2(IV)]. The message digests used by the integrity verification mechanism for validating user data are defined in [FCS_COP.1(CLIC-MD)].
[OSPP-IV]_O.INTEGRITY.ACTION	The TOE shall perform pre-defined actions upon detection of a breach of integrity; actions are defined by [FDP_SDI.2(IV)] for user data and [FPT_TIM.1(IV)] for TSF data and TSF code.
[OSPP-IV]_O.INTEGRITY.MANAGE	The management aspect for the integrity verification mechanism is covered by: • [FMT_MTD.1(IV-ACT)] covering management of actions performed by the TOE upon detection of integrity violation. • [FMT_MTD.1(IV-TSF)] covering update of the integrity verification database of the TSF data and selection of data subject to verification for TSF data. • [FMT_MTD.1(IV-USR)] covering update of the integrity verification database and selection of data subject to verification for user data.
[OSPP-LS]_O.LS.CONFIDENTIALITY	The information flow control policy is defined by specifying the subjects, objects, security attributes and rules in [FDP_IFC.2(LS), FDP_IFF.2(LS)]. Supportive to the enforcement of the policy are the automated label assignment when exporting data [FDP_ETC.2(LS)] and during the import

Security objectives	Rationale
	of data [FDP_ITC.1(LS), FDP_ITC.2(LS)]. For assigning labels to imported data, the label information transmitted with the data must be interpretable by the TOE [FPT_TDC.1(LS)].
[OSPP-LS]_O.LS.PRINT	The addition of label information on exported data during printing is governed by [FDP_ETC.2(LS)].
[OSPP-LS]_O.LS.LABEL	The assignment of labels to users is performed during user-subject binding [FIA_USB.1(LS)] with security attributes maintained by the TOE [FIA_ATD.1(LS)]. Object labels are assigned to objects when importing them into the TOE [FDP_ITC.1(LS), FDP_ITC.2(LS), FPT_TDC.1(LS)]. The management of labels is allowed for the TOE with [FMT_MSA.1(LS), FMT_MSA.3(LS)].
[OSPP-VIRT]_O.COMP.INFO_FLOW_CTRL	The information flow control policy covering the runtime of the compartments is specified with [FDP_IFC.2(VIRT)], and [FDP_IFF.1(VIRT)]. As the TOE shall allow export of data belonging to compartments, the TOE assigns the security attributes for enforcing the information flow control policy to the communicated data as specified with [FDP_ETC.2(VIRT)], [FDP_ITC.2(VIRT)], and [FPT_TDC.1(VIRT)]. Management of the security attributes for the information flow control policy is specified with [FMT_MSA.1(VIRT-CIFCP)], and [FMT_MSA.3(VIRT-CIFCP)] as well as FMT_MTD.1(VIRT-COMP).
[OSPP-VIRT]_O.COMP.RESOURCE_ACCESS	The access control policy for the resources belonging to the different compartments is defined with [FDP_ACC.2(VIRT)], and [FDP_ACF.1(VIRT)]. As the TOE shall allow export of data belonging to compartments, the TOE assigns the security attributes for enforcing the access control policy to the communicated data as specified with [FDP_ETC.2(VIRT)], [FDP_ITC.2(VIRT)], and [FPT_TDC.1(VIRT)]. Management of the security attributes for the access control policy is specified with [FMT_MSA.1(VIRT-CACP)], and [FMT_MSA.3(VIRT-CACP)] as well as FMT_MTD.1(VIRT-COMP).
[OSPP-VIRT]_O.COMP.IDENT	The identification of compartments to support the information flow control and access control policies is established with [FIA_UID.2(VIRT)].
[ST]_O.DISK.OVERWRITTEN	Overwriting SCSI hard drives with predefined bit patterns by administrators for residual information protection purposes is specified with [FDP_RIP.4].
[ST]_O.MANDATORY_INTEGRITY	Mandatory integrity is implemented by the SFRs defining the mandatory integrity control policy in [FDP_IFC.1(MIC), FDP_IFF.2(MIC)], user-subject binding in [FIA_USB.1(LSX)] and security attributes spelled out in [FIA_ATD.1(LSX)], as well as management functionality in [FMT_MSA.1(MIC), FMT_MSA.3(MIC)].
[ST]_O.ROLE	The TOE must prevent users from gaining access to and performing operations on its resources/objects unless they have been granted access by the resource/object owner or they have been assigned to a role (by

Security objectives	Rationale
	an authorized administrator) which permits those operations [FMT_SMR.2]. RBAC policy is defined in [FDP_ACC.1(RBAC), FDP_ACF.1(RBAC)]. The security attributes of subjects used to enforce the RBAC policy must be defined [FIA_ATD.1(HU), FIA_USB.2] and only the defined (henceforth, secure) values used [FMT_MSA.2(RBAC)]. Authorized users must be able to control who has access to objects [FMT_MSA.1(RBAC-ADM), FMT_MSA.1(RBAC-AUTH), FMT_MSA.1(RBAC-DFLT), FMT_MSA.1(RBAC-USR)] and be able to revoke that access [FMT_REV.1(OBJ)]. The RBAC protection of named objects must be continuous, starting from object creation [FMT_MSA.3(RBAC)]. A user's active role set shall be limited to their set of authorized roles [FTA_LSA.1(RBAC)]. User's with no roles (explicit or implied) shall be denied the ability to establish a session [FTA_TSE.1(RBAC)].
[ST]_O.ROLE.AUTHORIZATIONS	The TOE uses authorizations to control access to TOE protected resources. The authorization access control policy is specified by [FDP_ACC.1(AUTH), FDP_ACF.1(AUTH)]. The management of this policy is specified by [FMT_MSA.1(AUTH), FMT_MSA.3(AUTH)]. Since RBAC uses authorizations as security attributes, the authorizations must be defined and only the defined values used [FMT_MSA.2(RBAC)].
[ST]_O.ROLE.CONSISTENT_DB	The TOE detects inconsistencies in the RBAC-related databases using tests which validate the RBAC-related databases that can be executed by an authorized administrator [FPT_TST.1]. The TOE provides fail secure and recovery scenarios as defined in [FPT_FLS.1(RBAC), FPT_RCV.1, FPT_RCV.4] when inconsistencies, corruption, and inaccessibilities are detected in the RBAC-related databases.
[ST]_O.ROLE.HIERARCHY	The TOE must provide the capability of defining hierarchical roles as specified by [FMT_MTD.1(RBAC)]. The values used to define the roles and role hierarchies must accept only defined values [FMT_MTD.3(RBAC)].
[ST]_O.ROLE.SEP_DUTY	The TOE must provide the capability of enforcing ‘separation of duties’. The enforcement of role separation by [FMT_SMR.2] supports this objective.
[ST]_O.STACK.NO_EXEC	Allowing or denying execution of code on a process' stack is specified by [FPT_FLS.1(SED), FRU_FLT.2].
[ST]_O.TCB.ACCESS	The TOE provides access control to TCB objects as specified by [FDP_ACC.1(TCB), FDP_ACF.1(TCB)]. The management of the TCB objects and object attributes is specified by [FMT_MSA.1(TCB), FMT_MSA.3(TCB)].
[ST]_O.TN.ACCESS	The TSF is implemented by the TOE's Trusted Network policy, which has been defined in [FDP_IFC.1(TN), FDP_IFF.2(TN)], with the sensitivity clearance label and integrity clearance label security attributes defined in [FIA_ATD.1(LS), FIA_ATD.1(LSX)], user-subject binding in [FIA_USB.1(LS), FIA_USB.1(LSX)], and management functionality in [FMT_MSA.1(TN), FMT_MSA.3(TN)]. The requirements for importing user data are defined by [FDP_ITC.1(LS), FDP_ITC.2(LS)]. The requirement for exporting user data is defined by [FDP_ETC.2(LS)]. The requirement for consistent interpretation of security labels in networked sessions is spelled out in [FPT_TDC.1(LS)].

Security objectives	Rationale
[ST]_O.VIOS.I&A	The TSF must ensure that only authorized users gain access to the TOE and its resources. Users authorized to access the TOE must use an identification and authentication process [FIA_UID.2(VIOS), FIA_UAU.2]. To ensure authorized access to the TOE, authentication data is protected [FIA_ATD.1(VIOS), FIA_UAU.7(VIOS)]. Proper authorization for subjects acting on behalf of users is also ensured [FIA_USB.1(VIOS)]. The appropriate strength of the authentication mechanism is ensured [FIA_SOS.1(VIOS)].
[ST]_O.VIOS.MANAGE	VIOS provides management interfaces globally defined in [FMT_SMF.1(VIOS)] for: • the network and volume access control policies [FMT_MSA.1(VIOS), FMT_MSA.3(VIOS)]; • the authentication data aspects [FMT_MTD.1(VIOS-ADI), FMT_MTD.1(VIOS-ADM)] • the network and volume mapping aspects [FMT_MTD.1(VIOS-NV)] • the user security attributes aspects except for authentication data [FMT_MTD.1(VIOS-SA)] • the revocation of user attributes is provided with [FMT_REV.1(VIOS)]. The roles used for management are defined with [FMT_SMR.1].
[ST]_O.VIOS.NET.PROTECTED	Access control between VIOS Ethernet adapter device drivers and VIOS Ethernet device drivers acting on behalf of groups of LPAR partitions sharing a virtual network is specified by [FDP_ACC.1(VIOS), FDP_ACF.1(VIOS)] with management functionality specified by [FMT_MSA.1(VIOS), FMT_MSA.3(VIOS)]. The roles are specified by [FMT_SMR.1].
[ST]_O.VIOS.ROLE	The TOE must prevent users from gaining access to and performing operations on its resources/objects unless they have been granted access by the resource/object owner or they have been assigned to a role (by an authorized administrator) which permits those operations [FMT_SMR.1]. VIOS RBAC policy is defined in [FDP_ACC.1(VRBAC), FDP_ACF.1(VRBAC)]. The security attributes of subjects used to enforce the VRBAC policy must be defined [FIA_ATD.1(VIOS), FIA_USB.1(VIOS)] and only the defined (henceforth, secure) values used [FMT_MSA.2(VRBAC)]. Authorized users must be able to control who has access to objects [FMT_MSA.1(VRBAC-ADM), FMT_MSA.1(VRBAC-AUTH), FMT_MSA.1(VRBAC-DFLT), FMT_MSA.1(VRBAC-USR)]. The VIOS RBAC protection of named objects must be continuous, starting from object creation [FMT_MSA.3(VRBAC)]. A user's active role set shall be limited to their set of authorized roles [FTA_LSA.1(VRBAC)]. User's with no roles (explicit or implied) shall be denied the ability to establish a session [FTA_TSE.1(VRBAC)].
[ST]_O.VIOS.ROLE.HIERARCHY	The TOE must provide the capability of defining hierarchical roles as specified by [FMT_MTD.1(VRBAC)]. The values used to define the roles and role hierarchies must accept only defined values [FMT_MTD.3(VRBAC)].

Security objectives	Rationale
[ST]_O.VIOS.ROLE.SEP_DUTY	The TOE must provide the capability of enforcing ‘separation of duties’. The enforcement of role separation by [FMT_SMR.1] supports this objective.
[ST]_O.VIOS.VOL.PROTECTED	Access control between LPAR partitions and logical/physical volumes and VIOS SCSI device drivers acting on behalf of a group of LPAR partitions is specified by [FDP_ACC.1(VIOS), FDP_ACF.1(VIOS)] with management functionality specified by [FMT_MSA.1(VIOS), FMT_MSA.3(VIOS)]. The roles are specified by [FMT_SMR.1].

Table 15: Security objectives for the TOE rationale

6.2.3 Security requirements dependency analysis

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

Security Functional Requirement	Dependencies	Resolution
FAU_GEN.1(BASE)	FPT_STM.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1	FAU_GEN.1(BASE) FAU_GEN.1(LS)
	FIA_UID.1	FIA_UID.2(BASE)
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1(BASE) FAU_GEN.1(LS)
FAU_SAR.2	FAU_SAR.1	FAU_SAR.1
FAU_SAR.3(BASE)	FAU_SAR.1	FAU_SAR.1
FAU_SEL.1(BASE)	FAU_GEN.1	FAU_GEN.1(BASE)
	FMT_MTD.1	FMT_MTD.1(AE)
FAU_STG.1	FAU_GEN.1	FAU_GEN.1(BASE) FAU_GEN.1(LS)
FAU_STG.3	FAU_STG.1	FAU_STG.1
FAU_STG.4	FAU_STG.1	FAU_STG.1
FCS_CKM.1(SYM)	[FCS_CKM.2 or FCS_COP.1]	FCS_COP.1(NET)
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.1(RSA)	[FCS_CKM.2 or FCS_COP.1]	FCS_COP.1(NET)
	FCS_CKM.4	FCS_CKM.4

Security Functional Requirement	Dependencies	Resolution
FCS_CKM.1(DSA)	[FCS_CKM.2 or FCS_COP.1]	FCS_COP.1(NET)
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.2(NET)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(SYM) FCS_CKM.1(RSA) FCS_CKM.1(DSA)
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(SYM)
FCS_COP.1(NET)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(SYM) FCS_CKM.1(RSA) FCS_CKM.1(DSA)
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1(CRYPTO-ENC)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(SYM) FCS_CKM.1(RSA)
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1(CRYPTO-MD)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	Not satisfied. See [OSPP-CRYPTO].
	FCS_CKM.4	Not satisfied. See [OSPP-CRYPTO].
FCS_COP.1(CRYPTO-SGN)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(RSA) FCS_CKM.1(DSA)
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1(CLIC-ENC)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(SYM) FCS_CKM.1(RSA)
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1(CLIC-MD)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	Not specified. The specified message digests do not require keys; therefore, key generation is not required.
	FCS_CKM.4	Not specified. The specified message digests do not require keys; therefore, key destruction is not required.
FCS_COP.1(CLIC-SGN)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(RSA)
	FCS_CKM.4	FCS_CKM.4
FCS_RNG.1(CLIC)	No dependencies.	
FDP_ACC.1(PSO-AIXC)	FDP_ACF.1	FDP_ACF.1(PSO-AIXC)

Security Functional Requirement	Dependencies	Resolution
FDP_ACC.1(PSO-NFS)	FDP_ACF.1	FDP_ACF.1(PSO-NFS)
FDP_ACC.1(TSO)	FDP_ACF.1	FDP_ACF.1(TSO)
FDP_ACC.1(AUTH)	FDP_ACF.1	FDP_ACF.1(AUTH)
FDP_ACC.1(RBAC)	FDP_ACF.1	FDP_ACF.1(RBAC)
FDP_ACC.1(TCB)	FDP_ACF.1	FDP_ACF.1(TCB)
FDP_ACC.1(TCP)	FDP_ACF.1	FDP_ACF.1(TCP)
FDP_ACC.2(VIRT)	FDP_ACF.1	FDP_ACF.1(VIRT)
FDP_ACF.1(PSO-AIXC)	FDP_ACC.1	FDP_ACC.1(PSO-AIXC)
	FMT_MSA.3	FMT_MSA.3(PSO-AIXC)
FDP_ACF.1(PSO-NFS)	FDP_ACC.1	FDP_ACC.1(PSO-NFS)
	FMT_MSA.3	FMT_MSA.3(PSO-NFS)
FDP_ACF.1(TSO)	FDP_ACC.1	FDP_ACC.1(TSO)
	FMT_MSA.3	FMT_MSA.3(TSO)
FDP_ACF.1(VIRT)	FDP_ACC.1	FDP_ACC.2(VIRT)
	FMT_MSA.3	FMT_MSA.3(VIRT-CACP)
FDP_ACF.1(AUTH)	FDP_ACC.1	FDP_ACC.1(AUTH)
	FMT_MSA.3	FMT_MSA.3(AUTH)
FDP_ACF.1(RBAC)	FDP_ACC.1	FDP_ACC.1(RBAC)
	FMT_MSA.3	FMT_MSA.3(RBAC)
FDP_ACF.1(TCB)	FDP_ACC.1	FDP_ACC.1(TCB)
	FMT_MSA.3	FMT_MSA.3(TCB)
FDP_ACF.1(TCP)	FDP_ACC.1	FDP_ACC.1(TCP)
	FMT_MSA.3	FMT_MSA.3(TCP)
FDP_ETC.2(VIRT)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.2(VIRT) FDP_IFC.2(VIRT)
FDP_IFC.2(NI)	FDP_IFF.1	FDP_IFF.1(NI)
FDP_IFC.2(VIRT)	FDP_IFF.1	FDP_IFF.1(VIRT)

Security Functional Requirement	Dependencies	Resolution
FDP_IFF.1(NI)	FDP_IFC.1	FDP_IFC.2(NI)
	FMT_MSA.3	FMT_MSA.3(NI)
FDP_IFF.1(VIRT)	FDP_IFC.1	FDP_IFC.2(VIRT)
	FMT_MSA.3	FMT_MSA.3(VIRT-CIFCP)
FDP_ITC.2(BASE)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(PSO-AIXC) FDP_ACC.1(PSO-NFS) FDP_ACC.1(TSO) FDP_ACC.1(TCP) FDP_IFC.2(NI)
	[FTP_ITC.1 or FTP_TRP.1]	FTP_ITC.1
	FPT_TDC.1	FPT_TDC.1(BASE)
FDP_ITC.2(VIRT)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.2(VIRT) FDP_IFC.2(VIRT)
	[FTP_ITC.1 or FTP_TRP.1]	FTP_ITC.1
	FPT_TDC.1	FPT_TDC.1(VIRT)
FDP_RIP.2	No dependencies.	
FDP_RIP.3	No dependencies.	
FDP_RIP.4	No dependencies.	
FDP_SDI.2(IV)	No dependencies.	
FIA_AFL.1	FIA_UAU.1	FIA_UAU.1
FIA_ATD.1(HU)	No dependencies.	
FIA_ATD.1(TU)	No dependencies.	
FIA_SOS.1(BASE)	No dependencies.	
FIA_UAU.1	FIA_UID.1	FIA_UID.2(BASE)
FIA_UAU.5	No dependencies.	
FIA_UAU.7(BASE)	FIA_UAU.1	FIA_UAU.1
FIA_UID.2(BASE)	No dependencies.	
FIA_UID.2(VIRT)	No dependencies.	
FIA_USB.2	FIA_ATD.1	FIA_ATD.1(HU)

Security Functional Requirement	Dependencies	Resolution
FMT_MSA.1(PSO-AIXC)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(PSO-AIXC)
	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MSA.1(PSO-NFS)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(PSO-NFS)
	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MSA.1(TSO)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(TSO)
	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MSA.1(VIRT-CACP)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.2(VIRT)
	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MSA.1(VIRT-CIFCP)	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.2(VIRT)
	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MSA.1(AUTH)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(AUTH)
	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MSA.1(RBAC-ADM)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(RBAC)
	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MSA.1(RBAC-AUTH)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(RBAC)
	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MSA.1(RBAC-DFLT)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(RBAC)
	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)

Security Functional Requirement	Dependencies	Resolution
FMT_MSA.1(RBAC-USR)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(RBAC)
	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MSA.1(TCB)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(TCB)
	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MSA.1(TCP)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(TCP)
	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MSA.2(RBAC)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(RBAC)
	FMT_MSA.1	FMT_MSA.1(RBAC-ADM) FMT_MSA.1(RBAC-AUTH) FMT_MSA.1(RBAC-DFLT) FMT_MSA.1(RBAC-USR)
	FMT_SMR.1	FMT_SMR.2
FMT_MSA.3(PSO-AIXC)	FMT_MSA.1	FMT_MSA.1(PSO-AIXC)
	FMT_SMR.1	FMT_SMR.2
FMT_MSA.3(PSO-NFS)	FMT_MSA.1	FMT_MSA.1(PSO-NFS)
	FMT_SMR.1	FMT_SMR.2
FMT_MSA.3(TSO)	FMT_MSA.1	FMT_MSA.1(TSO)
	FMT_SMR.1	FMT_SMR.2
FMT_MSA.3(NI)	FMT_MSA.1	Satisfied with FMT_MTD.1(NI) as per [OSPP].
	FMT_SMR.1	FMT_SMR.2
FMT_MSA.3(VIRT-CACP)	FMT_MSA.1	FMT_MSA.1(VIRT-CACP)
	FMT_SMR.1	FMT_SMR.2
FMT_MSA.3(VIRT-CIFCP)	FMT_MSA.1	FMT_MSA.1(VIRT-CIFCP)
	FMT_SMR.1	FMT_SMR.2
FMT_MSA.3(AUTH)	FMT_MSA.1	FMT_MSA.1(AUTH)
	FMT_SMR.1	FMT_SMR.2

Security Functional Requirement	Dependencies	Resolution
FMT_MSA.3(RBAC)	FMT_MSA.1	FMT_MSA.1(RBAC-ADM) FMT_MSA.1(RBAC-AUTH) FMT_MSA.1(RBAC-DFLT) FMT_MSA.1(RBAC-USR)
	FMT_SMR.1	FMT_SMR.2
FMT_MSA.3(TCB)	FMT_MSA.1	FMT_MSA.1(TCB)
	FMT_SMR.1	FMT_SMR.2
FMT_MSA.3(TCP)	FMT_MSA.1	FMT_MSA.1(TCP)
	FMT_SMR.1	FMT_SMR.2
FMT_MSA.4(PSO)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(PSO-AIXC) FDP_ACC.1(PSO-NFS)
FMT_MTD.1(AE)	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MTD.1(AS)	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MTD.1(AT)	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MTD.1(AF)	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MTD.1(NI)	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MTD.1(IAT)	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MTD.1(IAF)	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MTD.1(IAU)	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MTD.1(AM-AP)	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)

Security Functional Requirement	Dependencies	Resolution
FMT_MTD.1(AM-MR)	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MTD.1(AM-MD)	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MTD.1(AM-MA)	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MTD.1(IV-ACT)	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MTD.1(IV-TSF)	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MTD.1(IV-USR)	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MTD.1(VIRT-COMP)	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MTD.1(PRIVS)	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MTD.1(RBAC)	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MTD.3(RBAC)	FMT_MTD.1	FMT_MTD.1(RBAC)
FMT_REV.1(OBJ)	FMT_SMR.1	FMT_SMR.2
FMT_REV.1(USR)	FMT_SMR.1	FMT_SMR.2
FMT_SMF.1(BASE)	No dependencies.	
FMT_SMR.2	FIA_UID.1	FIA_UID.2(BASE)
FPT_FLS.1(RBAC)	No dependencies.	
FPT_FLS.1(SED)	No dependencies.	
FPT_RCV.1	AGD_OPE.1	AGD_OPE.1
FPT_RCV.4	No dependencies.	
FPT_STM.1	No dependencies.	

Security Functional Requirement	Dependencies	Resolution
FPT_TDC.1(BASE)	No dependencies.	
FPT_TDC.1(VIRT)	No dependencies.	
FPT_TIM.1(IV)	No dependencies.	
FPT_TST.1	No dependencies.	
FRU_FLT.2	FPT_FLS.1	FPT_FLS.1(SED)
FTA_LSA.1(RBAC)	No dependencies.	
FTA_SSL.1	FIA_UAU.1	FIA_UAU.1
FTA_SSL.2	FIA_UAU.1	FIA_UAU.1
FTA_TSE.1(RBAC)	No dependencies.	
FTP_ITC.1	No dependencies.	
FAU_GEN.1(LS)	FPT_STM.1	FPT_STM.1
FAU_SAR.3(LS)	FAU_SAR.1	FAU_SAR.1
FAU_SEL.1(LS)	FAU_GEN.1	FAU_GEN.1(LS)
	FMT_MTD.1	FMT_MTD.1(AE)
FDP_ETC.2(LS)	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.2(LS)
FDP_IFC.1(MIC)	FDP_IFF.1	FDP_IFF.2(MIC)
FDP_IFC.1(TN)	FDP_IFF.1	FDP_IFF.2(TN)
FDP_IFC.2(LS)	FDP_IFF.1	FDP_IFF.2(LS)
FDP_IFF.2(MIC)	FDP_IFC.1	FDP_IFC.1(MIC)
	FMT_MSA.3	FMT_MSA.3(MIC)
FDP_IFF.2(TN)	FDP_IFC.1	FDP_IFC.1(TN)
	FMT_MSA.3	FMT_MSA.3(TN)
FDP_IFF.2(LS)	FDP_IFC.1	FDP_IFC.2(LS)
	FMT_MSA.3	FMT_MSA.3(LS)
FDP_ITC.1(LS)	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.2(LS)
	FMT_MSA.3	FMT_MSA.3(LS)

Security Functional Requirement	Dependencies	Resolution
FDP_ITC.2(LS)	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.2(LS)
	[FTP_ITC.1 or FTP_TRP.1]	FTP_ITC.1
	FPT_TDC.1	FPT_TDC.1(LS)
FIA_ATD.1(LS)	No dependencies.	
FIA_ATD.1(LSX)	No dependencies.	
FIA_USB.1(LS)	FIA_ATD.1	FIA_ATD.1(LS) FIA_ATD.1(LSX)
FIA_USB.1(LSX)	FIA_ATD.1	FIA_ATD.1(LSX)
FMT_MSA.1(LS)	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.2(LS)
	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MSA.1(MIC)	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.1(MIC)
	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MSA.1(TN)	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.1(TN)
	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1(BASE)
FMT_MSA.3(LS)	FMT_MSA.1	FMT_MSA.1(LS)
	FMT_SMR.1	FMT_SMR.2
FMT_MSA.3(MIC)	FMT_MSA.1	FMT_MSA.1(MIC)
	FMT_SMR.1	FMT_SMR.2
FMT_MSA.3(TN)	FMT_MSA.1	FMT_MSA.1(TN)
	FMT_SMR.1	FMT_SMR.2
FPT_TDC.1(LS)	No dependencies.	
FDP_ACC.1(VIOS)	FDP_ACF.1	FDP_ACF.1(VIOS)
FDP_ACC.1(VRBAC)	FDP_ACF.1	FDP_ACF.1(VRBAC)
FDP_ACF.1(VIOS)	FDP_ACC.1	FDP_ACC.1(VIOS)
	FMT_MSA.3	FMT_MSA.3(VIOS)

Security Functional Requirement	Dependencies	Resolution
FDP_ACF.1(VRBAC)	FDP_ACC.1	FDP_ACC.1(VRBAC)
	FMT_MSA.3	FMT_MSA.3(VRBAC)
FIA_ATD.1(VIOS)	No dependencies.	
FIA_SOS.1(VIOS)	No dependencies.	
FIA_UAU.2	FIA_UID.1	FIA_UID.2(VIOS)
FIA_UAU.7(VIOS)	FIA_UAU.1	FIA_UAU.2
FIA_UID.2(VIOS)	No dependencies.	
FIA_USB.1(VIOS)	FIA_ATD.1	FIA_ATD.1(VIOS)
FMT_MSA.1(VIOS)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(VIOS)
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1(VIOS)
FMT_MSA.1(VRBAC-ADM)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(VRBAC)
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1(VIOS)
FMT_MSA.1(VRBAC-AUTH)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(VRBAC)
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1(VIOS)
FMT_MSA.1(VRBAC-DFLT)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(VRBAC)
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1(VIOS)
FMT_MSA.1(VRBAC-USR)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(VRBAC)
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1(VIOS)
FMT_MSA.2(VRBAC)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(VRBAC)
	FMT_MSA.1	FMT_MSA.1(VRBAC-ADM) FMT_MSA.1(VRBAC-AUTH) FMT_MSA.1(VRBAC-DFLT) FMT_MSA.1(VRBAC-USR)
	FMT_SMR.1	FMT_SMR.1

Security Functional Requirement	Dependencies	Resolution
FMT_MSA.3(VIOS)	FMT_MSA.1	FMT_MSA.1(VIOS)
	FMT_SMR.1	FMT_SMR.1
FMT_MSA.3(VRBAC)	FMT_MSA.1	FMT_MSA.1(VRBAC-ADM) FMT_MSA.1(VRBAC-AUTH) FMT_MSA.1(VRBAC-DFLT) FMT_MSA.1(VRBAC-USR)
	FMT_SMR.1	FMT_SMR.1
FMT_MTD.1(VIOS-ADI)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1(VIOS)
FMT_MTD.1(VIOS-ADM)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1(VIOS)
FMT_MTD.1(VIOS-NV)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1(VIOS)
FMT_MTD.1(VIOS-SA)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1(VIOS)
FMT_MTD.1(VRBAC)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1(VIOS)
FMT_MTD.3(VRBAC)	FMT_MTD.1	FMT_MTD.1(VRBAC)
FMT_REV.1(VIOS)	FMT_SMR.1	FMT_SMR.1
FMT_SMF.1(VIOS)	No dependencies.	
FMT_SMR.1	FIA_UID.1	FIA_UID.2(VIOS)
FTA_LSA.1(VRBAC)	No dependencies.	
FTA_TSE.1(VRBAC)	No dependencies.	

Table 16: TOE SFR dependency analysis

6.2.4 Internal consistency and mutual support of SFRs

6.3 Security Assurance Requirements

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

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

Security assurance class	Security assurance requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
ASE Security Target evaluation	ASE_CCL.1 Conformance claims	OSPP	No	No	No	No
	ASE_INT.1 ST introduction	CC Part 3	No	No	No	No
	ASE_SPD.1 Security problem definition	CC Part 3	No	No	No	No
	ASE_OBJ.2 Security objectives	CC Part 3	No	No	No	No
	ASE_ECD.1 Extended components definition	CC Part 3	No	No	No	No
	ASE_REQ.2 Derived security requirements	CC Part 3	No	No	No	No
	ASE_TSS.1 TOE summary specification	CC Part 3	No	No	No	No
ADV Development	ADV_ARC.1 Security architecture description	CC Part 3	No	No	No	No
	ADV_FSP.4 Complete functional specification	CC Part 3	No	No	No	No
	ADV_IMP.1 Implementation representation of the TSF	CC Part 3	No	No	No	No
	ADV_TDS.3 Basic modular design	CC Part 3	No	No	No	No
AGD Guidance documents	AGD_OPE.1 Operational user guidance	CC Part 3	No	No	No	No
	AGD_PRE.1 Preparative procedures	CC Part 3	No	No	No	No
ALC Life-cycle support	ALC_CMC.4 Production support, acceptance procedures and automation	CC Part 3	No	No	No	No
	ALC_CMS.4 Problem tracking CM coverage	CC Part 3	No	No	No	No
	ALC_DEL.1 Delivery procedures	CC Part 3	No	No	No	No
	ALC_DVS.1 Identification of security measures	CC Part 3	No	No	No	No
	ALC_FLR.3 Systematic flaw remediation	CC Part 3	No	No	No	No
	ALC_LCD.1 Developer defined life-cycle model	CC Part 3	No	No	No	No
	ALC_TAT.1 Well-defined development tools	CC Part 3	No	No	No	No
ATE Tests	ATE_COV.2 Analysis of coverage	CC Part 3	No	No	No	No
	ATE_DPT.1 Testing: basic design	CC Part 3	No	No	No	No
	ATE_FUN.1 Functional testing	CC Part 3	No	No	No	No
	ATE_IND.2 Independent testing - sample	CC Part 3	No	No	No	No

Security assurance class	Security assurance requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
AVA Vulnerability assessment	AVA_VAN.3 Focused vulnerability analysis	CC Part 3	No	No	No	No

Table 17: Security assurance requirements

6.3.1 Security Target evaluation (ASE)

6.3.1.1 Conformance claims (ASE_CCL.1)

Content and presentation elements:

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

6.4 Security Assurance Requirements Rationale

The evaluation assurance level has been chosen to match an Enhanced-Basic attack potential commensurate with the threat environment that is experienced by typical consumers of the TOE. This evaluation assurance level is also required by [OSPP].

The security assurance requirement ASE_CCL.1.10C has been refined by [OSPP]. The [OSPP] version of ASE_CCL.1.10C is provided in section 6.3.1.1 without any additional modifications.

The evaluation assurance level has been augmented with ALC_FLR.3 which is required by [OSPP] and is commensurate with the flaw remediation capabilities offered by the developer.

7 TOE Summary Specification

7.1 Security Enforcing Components Overview

7.1.1 Introduction

AIX provides a multi-user, multitasking, and multi-virtualized environment, where users interact with the operating system through commands issued to a command interpreter. The command interpreter invokes command programs, which in turn function by making system calls to the operating system kernel. The TSF is comprised of the kernel and trusted processes (trusted programs that are not part of the kernel). All operations performed by users are mediated by the TOE Security Functionality (TSF) in accordance with the Security Requirements defined in chapter 6.

Within AIX, a user can LOGIN to the console of any host computer, request local services at that computer, as well as request network services from any other host in the system.

Processes perform all activity. A process may be started by a user issuing a command, may be created automatically to service a network request, or may be part of the running system created at system initialization. Each process is running a program. A process may begin running a new program (via the exec system call), or create a copy of itself (via the fork system call).

Some activities, such as responding to network requests, are performed directly by the kernel.

The following sections discuss services provided by the kernel, by non-kernel trusted software and the network services. Network services are discussed separately because their implementation is split between kernel and non-kernel components.

As long as those functions just provide a user interface (e. g., System Management tools) they are not considered to be part of the TSF. But if they directly implement part of a security function (e. g. the trusted processes that reads identification and authentication data) they are considered to be part of the TSF.

7.1.2 Kernel services

The AIX kernel includes the base kernel and kernel extensions. The base kernel includes support for system initialization, memory management, file and I/O management, process control, audit services and Inter-Process Communications (IPC) services. Kernel extensions and device drivers are separate kernel software modules that perform specific functions within the operating system.

Device drivers are implemented as kernel extensions.

The base kernel has the following key characteristics:

- Can be paged out: Portions of the kernel code and data can be paged out, permitting the kernel to run using less memory than would be required for the whole kernel.
- Pinned: Part of the kernel is always resident or “pinned” into memory and cannot be paged. Pinned code cannot call kernel services that may result in a page fault.
- Can be preempted: The AIX kernel can be preempted. Higher priority threads may interrupt kernel threads, providing support for time critical functions.
- Dynamic and extensible: In standard AIX, kernel extensions can be loaded and unloaded while the system is running to allow a dynamic, extensible kernel without requiring a rebuild and reboot. In the evaluated configuration, dynamic changes to the kernel are prohibited through warnings described in [SecGuide]. At system start up, only the kernel extensions that are part of the evaluated product may be loaded. As an example, the administrator can

add pieces of hardware (as long as they are part of the hardware configuration listed in this Security Target) to a specific configuration and reboot the system. This will cause the kernel extensions that support the needed device drivers for the new hardware to be loaded. The ability to load/unload kernel extensions is restricted to the processes having the PV_DEV_LOAD privilege.

The AIX kernel implements a virtual memory manager (VMM) that allocates a large, contiguous address space to each process running on the system. This address space is spread across physical memory and paging space on a secondary storage device. The VMM manages the paging space used by the AIX file system and provides memory buffers for use within the kernel. The file system and VMM are tightly coupled. Disk pages, whether for file I/O or paging space, are faulted into free pages in memory. The VMM does not maintain a separate pool of pages solely for file system I/O.

The process management component includes the software that is responsible for creating, scheduling, and terminating processes and process threads. Process management allows multiple processes to exist simultaneously on a computer and to share usage of the computer's processor(s). A process is defined as a program in execution, that is, it consists of the program and the execution state of the program.

Process management also provides services such as inter-process communications (IPC) and event notification. The base kernel implements

- named pipes
- unnamed pipes
- signals
- semaphores
- shared memory
- message queues
- Internet domain sockets
- UNIX domain sockets
- Audit event generation

The file and I/O software provides access to files and devices. The AIX Logical File System (LFS) provides a consistent view of multiple physical file system implementations. The following file system types are included in the evaluated configuration:

- Journaled File System 2 (JFS2)
- Encrypted File System (EFS)
- CDROM File System (CDRFS)
- Universal Disk Format file system (UDFS)
- Network File System (NFS)
- Special File System (SPECFS)

JFS2, EFS, CDRFS and UDFS work off of a physical medium (disk, CDROM, DVD) and NFS works across the network. SPECFS is a file system used internally by the kernel to support disk and other physical and virtual device I/O. The process file system, PROCFS, provides access to the process image of each process on the machine as if the process were a “file”. Process access decisions are enforced by DAC, MAC (LAS mode only), MIC (LAS mode only), and TCB attributes inferred from the underlying process's security attributes.

LAS Mode Only: CDRFS, UDFS, PROCFS and (client-side) NFS are single level file systems: For mandatory access control, the labels of their mount point apply to all objects in the mounted file system. Single level file systems are not subject to mandatory integrity control, TCB and file security flag policies, and their objects cannot be associated with privileges. This is to be taken into account when reading the following sections of the TOE Summary Specification (TSS).

7.1.3 Non-kernel TSF services

The non-kernel TSF services are:

- Identification and authentication services
- Auditing journaling and post-processing services
- Network application layer services
- System integrity checking

Those services support the security functions implemented within the kernel and use the kernel interface for this purpose, but they are not running themselves in kernel mode. Those functions are included in the TSF as far as they are required for the security services of the TOE (Identification and Authentication services), while other services that are implemented as tools or commands for the use of the system administrator and where the kernel prohibits the use/misuse of those tools or commands since they use kernel functions restricted to the system administrator and attempted use by normal users is prohibited by the kernel.

7.1.4 Network services

Each computer is capable of providing the following types of services:

- Local services to the user currently logged in to the local computer console.
- Local services to previous users via deferred jobs.
- Local services to users who have accessed the local host via the network using protocols such as telnet.
- Network services to clients on either the local host or on remote hosts.

Network services are provided to clients via a client-server architecture. This client-server architecture refers to the division of the software that provides a service into a client portion, which makes requests, and a server portion, which carries out client requests (usually on a different computer). A service protocol acts as the interface between the client and server.

The primary low-level protocols are Internet Protocol (IP), Transmission Control Protocol (TCP), and User Datagram Protocol (UDP). IP is not user visible, but non-TSF processes may communicate with other hosts using a reliable byte stream or unreliable datagrams, TCP and UDP respectively.

The higher-level network services are built on TCP or UDP. The TOE supports the TCP application protocols listed below:

- Internet remote login and file transfer services (*telnet* and *ftp*) are supported within the evaluated product, as are similar BSD interfaces, including remote command execution (*rlogin, rcp, rsh, rexec*).
- The Hyper-Text Transfer Protocol (HTTP) is used by the WebInfo document display system (docsearch) for the presentation of public data. The HTTP server is not security relevant and therefore not part of the TSF.

- The Network File System (NFS) protocol is supported for remote file access. This includes some subsidiary protocols, such as the Remote Procedure Call (RPC), portmap protocols, and the mountd protocol for file system import and export.

BAS Mode Only: AIX includes multiple X Windows clients in addition to an X Windows server on each host. Each server accepts connections from local clients using UNIX domain sockets.

LAS Mode Only: In addition to the base connectivity provided, all network connections can be configured to support labeling as specified in the BSO/ESO/CIPSO/RIPSO protocols.

7.1.5 Workload Partitions

AIX supports virtual environments called Workload Partitions (WPARs) which provide virtual AIX environments within AIX. It provides the following distinct environments:

- **Global environment** – This is the main or top-level environment which provides the standard, full functionality of AIX. From this level, the other types of WPARs can be created, controlled, and destroyed. Only one Global environment exists within an LPAR.
- **System WPAR** – This is a virtual AIX environment created from the Global environment. It contains most of the functionality and programs that the Global environment contains and, in short, has the look and feel of a separate AIX system. The Global environment can create multiple, concurrent System WPARs.
- **Application WPAR** – This is a very limited virtual AIX environment created from a Global environment. This WPAR contains only the applications specified during the creation of this WPAR. It allows for the running of one or more programs with process-space isolation from the Global environment. In short, the processes in an Application WPAR only know about the other processes in the same Application WPAR. The Global environment can create multiple, concurrent Application WPARs.

The install type (i.e., LAS mode or BAS mode) of the Global environment determines the type of System WPARs and Application WPARs. Thus, if the Global environment is in LAS mode, then all WPARs created by the Global environment will be in LAS mode. Similarly, if the Global environment is in BAS mode, then all WPARs created by the Global environment will be in BAS mode.

7.1.6 Security policy overview

The TOE is distributed across multiple host computers, each running a semiautonomous instance of the AIX operating system optionally using the NFSv4 distributed file system. The policy is described as follows:

- There is not a single kernel; rather, there is an AIX kernel running on each host computer in the system.
- The system does not have a common memory space; rather, each host in the system has its own memory space. Memory management, segmentation and paging are all managed locally, without respect to other hosts.
- The systems are maintained using a consistent user management policy across all systems.
- Identification and authentication (I&A) is performed locally by each host computer, but can use a common database. Each user is required to LOGIN with a valid password and user identifier combination at the local workstation and also at any remote computer where the user can enter commands to a shell program (e.g., remote login and telnet sessions).
- Neither the process ID, nor the associated thread IDs, are unique within the system; rather, a PID, and its associated TIDs, are unique on each host within the system. Process and thread management is performed locally, without respect to other hosts.

- The names of objects may not be unique within the system; rather, object names are unique on each host. For example, each host maintains its own local file system, but may mount NFS exported file systems at various locations in the local directory tree.
- Discretionary access control (DAC) is performed locally by each of the host computers and is based on user identity and group membership. Each process has an identity (the user on whose behalf it is operating) and belongs to one or more groups. All named objects have an owning user, an owning group and a DAC attribute, which is a set of permission bits. In addition, file system objects optionally have an extended permission list also known as an AIXC Access Control List (ACL) or, in lieu of enforced permission bits, an NFSv4 ACL. Both the extended permissions mechanism and NFSv4 ACL are significant enhancements beyond traditional UNIX systems, and permits control of access based on lists of users and/or groups to whom specific permissions may be individually granted or denied. For TCP based services in BAS mode, an additional type of ACL is provided that can be used to restrict user access to specific services.
- LAS Mode Only: The system supports mandatory access control (MAC) based on sensitivity labels. From a defined set of hierarchical sensitivity levels (SLs), each named object is assigned a dedicated SL, and each user is assigned a range of SLs that he is allowed to access. Users (or, processes acting on behalf of users) can create new objects only with the SL they are currently operating under, cannot read objects that have a higher SL than the user and not write objects that have another SL than themselves. Directories and devices can optionally be assigned a label range – in this case, users can write to an object if their SL is within the SL range of the object.
- LAS Mode Only: The system supports mandatory integrity control (MIC) based on hierarchical integrity labels (TLs). Every named object is assigned a dedicated TL, and each user is assigned a range of valid TLs. Users (or, processes acting on behalf of users) can create objects with the TL that they are currently operating under and cannot write objects that have a higher TL than themselves.
- The system protects trusted computing base (TCB) objects from modification during normal multi-user operation. Objects tagged with the FSF_TLIB flag can only be modified when the system is in configuration mode and the user (or, process acting on behalf of a user) has the appropriate privileges to apply changes to the TCB.
- The system protects users and administrators from using code or data that has been tampered with via the Trusted Execution (TE) function. Objects monitored under the TE policy cannot be accessed when the TE function detects unauthorized modifications.
- The system uses authorizations and privileges to implement administrative roles (RBAC) and to allow the controlled by-passing of the security policies enforced by the TOE.
- Object reuse is performed locally, without respect to other hosts.
- Audit is performed locally by each host computer. The audit facility generates audit records for activities performed directly by untrusted processes (e.g., the system calls that perform file I/O) as well as trusted process activities (e.g., requests for batch jobs). Audit tools are available to merge audit files from the various hosts.
- Interrupt handling is performed locally, without respect to other hosts.
- Root Enabled Mode Only: Privilege is based on the root identity. All privileged processes (setuid root programs and programs run under the root identity) start as processes with all privileges enabled. Unprivileged processes, which include setgid trusted processes, start and end with no privileges enabled.

- VIOS discretionary access control is performed by VIOS to provide access control between VIOS SCSI device drivers acting on behalf of LPAR partitions and logical/physical volumes. It also provides access control between VIOS Ethernet device drivers acting on behalf of groups of LPAR partitions sharing virtual networks and VIOS Ethernet adapter device drivers.

7.1.7 TSF structure

The TSF is the portion of the system that is responsible for enforcing the system's security policy. The TSFs of AIX are distributed across each System p POWER6 and POWER7 host computer and consist of three major components: kernel software, kernel extension software, and trusted processes. All these components must operate correctly for the system to be trusted. Those functions are supported by the mechanisms of the underlying hardware which are used to protect the TSF from tampering by untrusted processes.

The hardware components support two execution states where kernel mode or supervisor state, software runs with hardware privilege and user mode or problem state software runs without hardware privilege. AIX also provides two types of memory protection: segmentation and page protection. The memory protection features isolate critical parts of the kernel from user processes and ensure that segments in use by one process are not available to other processes. The two-state architecture and the memory protections form the basis of the argument for process isolation and protection of the TSF.

The trusted processes include programs such as AIX administrative programs, shells, and standard UNIX utilities that run with administrative privilege, as a consequence of being invoked by a user with the root identity, if in the root enabled mode, or proper authorization, if in LAS mode or in root disabled mode, and, in the case of LAS mode, with the appropriate MAC privileges. Non-kernel TSF software also includes daemons that provide system services, such as networking and managing audit data, as well as setuid, setgid, and privileged programs that can be executed by untrusted users and privileged commands which are defined in the *privcmds* table.

7.1.8 TSF interfaces

Each sub-section here summarizes a class of interfaces in the AIX distributed system, and characterizes them in terms of the TSF boundary. The TSF boundary includes some interfaces, such as commands implemented by privileged processes, which are similar in style to other interfaces that are not part of the TSF boundary and thus not trusted. Some interfaces are part of the TSF boundary only when used in a privileged environment, such as an administrator's process, but not when used in a non-privileged environment, such as a normal user process. All interface classes are described in further detail in the next chapter, and the mechanisms in subsequent chapters. As this is only an introduction, no explicit forward references are provided.

7.1.8.1 User interfaces

The typical interface presented to a user is the command interpreter, or shell. The user types commands to the interpreter, and in turn, the interpreter invokes programs. The programs execute hardware instructions and invoke the kernel to perform services, such as file access or I/O to the user's terminal. A program may also invoke other programs, or request services using an IPC mechanism. Before using the command interpreter, a user must log in.

The command interpreter or shell as well as other programs operating on behalf of a user have the following interfaces:

- CPU instructions, which a process uses to perform computations within the processor's registers and a process's memory areas. CPU instructions are interpreted by the hardware, which is part of the Operational Environment; CPU instructions are therefore not a TSF interface.
- System calls (e.g., *open*, *fork*), through which a process requests services from the kernel, and are invoked using a special CPU instruction; System calls are the primary way for a program operating on behalf of a user to request services of the TOE including the security services. System calls related to security functions are therefore part of the TSF interface.
- Directly-invoked trusted processes (e.g., *passwd*) which perform higher-level services, and are invoked with an exec system call that names an appropriate program which is part of the TSF, and replaces the current process's content with it; a limited number of those processes exist that perform security functions and are therefore part of the TSF interface.
- Daemons, which accept requests stored in files or communicated via other IPC mechanisms, generally created through use of directly invoked processes (some trusted, some untrusted). A few daemons perform security functions and therefore present part of the TSF interface.
- Distributed Services (e.g., *telnet*) - The distributed services interface operates at many different levels of abstraction. At the highest level, it provides a means for users on one host within the system to request a virtual terminal connection on another host within the system. At a lower level, it allows a system to request a specific service from another system on behalf of a user. Examples of requested services include, executing a command line (e.g., *rsh*) or transferring whole files (e.g., FTP). At the lowest level, it allows a subject on one host in the system to request a connection (e.g., TCP), or deliver data (e.g., UDP) to a listening subject. Distributed services usually consist of a client on the requestor's side and a server (usually a daemon) running on the server's side. Authentication (if required by the service) and access control use dedicated interfaces to the functions on the server side which are therefore part of the TSF interface.

7.1.8.2 Operation and administrator interface

The primary administrative interfaces to AIX are the same as the interfaces for ordinary users. In an LAS mode system or in a root disabled mode system, authorized administrators log into the system using their user ID and password, and perform administrative tasks they have been authorized to perform. Additionally, in a root enabled mode system, an administrator can log into the system with a standard, untrusted identity and password and, after assuming the root identity, uses standard AIX commands to perform administrative tasks.

The system is composed of one or more System p POWER6 and/or POWER7 computer systems. Each of these host computers may be in one of the following states: shut down, initialization, single-user mode, or multi-user secure state. Administration entails the configuration of multiple computers and the interactions of those computers, as well as the administration of users, groups, files, printers, and other resources within the system. It also includes administering WPARs within a host.

AIX provides a general purpose, menu-based utility for system administration: *smitty*. Other programs (e.g., */usr/bin/acledit*, */usr/bin/chuser*, */usr/bin/rm*) and scripts are used for system administration, but *smitty* is significant because it provides comprehensive system administration capabilities.

smitty is required for the administration of the AIX distributed system, but the decision as to which administrative utility to use depends upon whether or not the system is in a secure state:

- *smitty* (a cursor-based ASCII version of the System Management Interface Tool (SMIT)) is a text menu interface and dispatcher for a collection of administrative programs.

- *smitty* is used to administer the local host, i.e., the computer where it is run.

There are other tools for system administration (e. g., *msmit*) that provide a graphical user interface for system administration. Those tools are not part of the evaluated configuration.

The part of the administrative database that is used to configure and manage TSF is seen as part of the TSF interface. The administrative database is protected by the access control mechanisms of the TOE. It is therefore very important to set the access rights to the files of the administrative database such that non-administrative users are prohibited from modifying those files and have read access on a need to know basis only.

7.1.9 Secure and Non-Secure States

The secure state for the AIX distributed system is defined as a host's entry into multi-user mode with auditing fully operational. At this point, the host accepts user logins and services network requests. If these facilities are not available, the host is considered to be in a non-secure state. Although it may be operational in a limited sense and available for an administrative user to perform system repair, maintenance, and diagnostic activity, the TSF are not in full operation and is not necessarily protecting all system resources according to the security policy. The non-secure state is also a specific configuration state of the system where system security flags (SSFs) and the trusted library path can be modified and FSF_TLIB or FSF_TLIB_PROC tagged objects can be created, modified, or deleted. This functionality is not available in the operational / multiuser mode.

With respect to auditing, this Security Target does not define a minimum level of events that need to be audited. But it is required that the system administrator is able to configure all the events mentioned in this Security Target to be included in the audit trail. A system administrator may then - according to his requirements - define the events that are audited. The administrator is able to change those events using the audit configuration functions during system operation.

7.2 TOE Security Functions

7.2.1 Introduction

This chapter describes how the Security Enforcing components of the TOE provide the Security Requirements identified in chapter 6.

A high level description is provided for each group of security enforcing functions (SEFs) providing a common feature or service, and stating how the functionality specified by the Security Enforcing Function group is provided by the security enforcing components identified in this chapter.

The security enforcing function groups identified in this chapter follow the description given in chapter 2.

The TOE Security Functionality (TSF) is described with sufficient detail to provide a general understanding of those functions and how they work. A more detailed description of those functions and a mapping of the TSF to TOE subsystems are provided in the high level design documentation.

References to components given in *italics* can be traced to manual pages or TOE sources for further information. Note also that some commands initiate trusted processes or are a local front end to a trusted process (e.g., *ftp* and the *ftpd* daemon, *telnet* and the *telnetd* daemon). In these instances, a generic reference to the command is made.

7.2.2 AIX & Trusted AIX

7.2.2.1 Identification and authentication (IA)

User identification and authentication in the AIX system includes all forms of interactive login (e.g., using the Telnet or FTP protocols) as well as identity changes through the *su* command. These all rely on explicit authentication information provided interactively by a user.

Identification and authentication of users is performed either from a terminal where no user is logged on or when a user that is logged on starts a service that requires additional authentication. All those services use a common mechanism for authentication described as function IA.2 in this section. They all use the administrative databases described in function IA.1 in this section. Function IA.3 describes the authentication process for those network services that require authentication. Function IA.4 describes the change of the user's identity using the *su* command. Function IA.5 describes the login process when a user logs in at a terminal. Function IA.6 describes the logoff process.

7.2.2.1.1 User identification and authentication data management (IA.1)

The TOE supports the following identification and authentication (I&A) administrative database types for both the Global environment and System WPAR:

- File-based
- LDAP-based
- NAS (Kerberos Version 5), limited to NFSv4 client-server authentication

Only remote logins and logins using the *clogin* command are possible for System WPARs.

This section and its subsections map to the following SFR(s):

- FIA_ATD.1(HU)
- FIA_SOS.1(BASE)
- FIA_UAU.5
- FMT_MSA.2(RBAC)

7.2.2.1.1.1 File-based I&A

AIX by default maintains a local administrative database. This database is used to manage identification and authentication data used by the operating system.

Administrators, through the SMIT (*smitty*) administrative interface or via command line tools, perform changes to the files that constitute the administrative database.

Users are allowed to change their passwords by using the *passwd* command, which is a privileged program. This configuration allows a process running the *passwd* program to read the contents of */etc/security/user* and to modify the */etc/security/passwd* file for the user's password entry, both which would ordinarily be inaccessible to a non-privileged user process. Users are also forced to change their passwords at login time, if the password has expired.

The */etc/passwd* file contains the user's name, the ID of the user, an indicator, if the password of the user is valid, the principal group ID of the user, the clearance label (MAC) of the user (in LAS mode only), and a few other, not security relevant information. The encrypted password of the user itself is not stored in this file but in the */etc/security/passwd* file which is protected against read access for ordinary users. This prohibits dictionary attacks on passwords in the passwd file as, for example, described in [PwdSecHist].

The */etc/security/passwd* file contains the encrypted password, the time the password was last changed and some other information that are not subject to the security functions as defined in this Security Target.

For a complete list of user attributes see the description of the function SM.4.

The system administrator defines restrictions on authentication data like minimum and maximum size, the minimum number of alphabetic characters, the minimum number of characters that are different from the old password, the minimum number of non-alphabetic characters as well as the maximum life time of a password, the number of unsuccessful login attempts allowed before the account is locked and the times and days the user is allowed to log into the system. Those restrictions can be defined on a per user basis and are stored in the */etc/security/user* file. The system administrator can use those parameters to define a password policy.

The */etc/security/lastlog* file contains the time since the last successful login, the time of the last unsuccessful login and the number of unsuccessful login attempts since the last successful login.

7.2.2.1.1.2 LDAP-based I&A

The TOE includes LDAP-based I&A where the LDAP-base I&A is configured in the “UNIX-type” authentication mode. (The LDAP server is part of the Operational Environment, not the TOE.) In this mode, the administrative data (including user names, IDs, passwords, and, in the case LAS mode, clearance labels (MAC)) are stored in LDAP where access to the data is limited to the LDAP administrator. When a user logs into the TOE, the TOE binds to the LDAP server using the LDAP administrator account, retrieves the necessary data for the user (including the password) from LDAP, and then performs authentication using the data retrieved from LDAP.

The system maintains an administrative database on an LDAP server. The remaining hosts import the administrative data from the same LDAP server through the same mechanism described in the previous paragraph.

The system maintains a consistent administrative database by making all administrative changes on the designated LDAP server. A user ID on any computer refers to the same individual on all other computers. In addition, the password configuration, name-to-UID mappings, and other data are identical on all hosts in the distributed system.

Administrators, through the SMIT administrative interface, perform changes to the LDAP data that constitute the administrative database.

Users are allowed to change their passwords by using the *passwd* command, which is a privileged program. This configuration allows a process running the *passwd* command to communicate to the local trusted process LDAP authentication daemon (which is also a privileged program) over a privileged socket, and request the LDAP authentication daemon to retrieve and modify the user's password entry. Users are also forced to change their passwords at login time, if the password has expired.

LDAP contains the user's name, the id of the user, the encrypted password, the time the password was last changed, an indicator, if the password of the user is valid, the principal group id of the user, and a few other not security relevant attributes.

For a complete list of user attributes see the description of the function SM.4.

The system administrator defines restrictions on authentication data like minimum and maximum size, the minimum number of alphabetic characters, the minimum number of characters that are different from the old password, the minimum number of non-alphabetic characters as well as the maximum life time of a password, the number of unsuccessful login attempts allowed before the account is locked and the times and days the user is allowed to log into the system. Those restrictions

can be defined on a per user basis and are stored in LDAP. The system administrator can use those parameters to define a password policy such that the passwords satisfy the requirements defined in FIA_SOS.1(BASE).

LDAP also contains the time since the last successful login, the time of the last unsuccessful login and the number of unsuccessful login attempts since the last successful login.

7.2.2.1.1.3 NAS-based I&A for NFSv4

When using the NFSv4 clients and servers, the clients authenticate to the servers using NAS, which uses the Kerberos ticketing mechanisms. (The NAS (Kerberos Version 5) servers are part of the Operational Environment, not the TOE.) To perform this authentication, the TOE user (through the use of the TOE's NAS client and NAS cryptographic library) contacts the NAS Authentication Server located in the Operational Environment to obtain a Kerberos ticket granting ticket (TGT). The TGT returned by the Authentication Server contains a key used to identify the TOE user. The NFSv4 client then sends its TGT to the NAS Ticket Granting Server (TGS) located in the Operational Environment to obtain an NFSv4 server ticket used to establish a connection between the NFSv4 client and the NFSv4 server. The TGS returns an NFSv4 server ticket to the NFSv4 client to use when contacting the NFSv4 server. The NFSv4 client then sends the user's NFSv4 server ticket to the NFSv4 server to authenticate itself and to establish a trusted channel connection with the NFSv4 server. The NFSv4 server sends the client's ticket to the TGS for validation. If the validation is successful, the user is considered authenticated and a trusted channel is established between the NFSv4 client and NFSv4 server. Both the NFSv4 client and the NFSv4 server use CliC for cryptography, not the NAS cryptographic library. For more information on this trusted channel, see section 7.2.2.14.8 "Protected communication (TP.8)".

7.2.2.1.2 Common authentication mechanism (IA.2)

AIX includes a common authentication mechanism which is a subroutine used for all activities that create a user session, including all the interactive login activities, batch jobs, and authentication for the *su* command.

The common mechanism includes the following checks and operations:

- Check password authentication
- Check password expiration
- Check whether access should be denied due to too many consecutive authentication failures
- Get user security characteristics (e.g., user, groups, clearances (LAS mode only), authorizations)

The common I&A mechanism identifies the user based on the supplied user name, gets that user's security attributes, and performs authentication against the user's password. A result of success indicated by a 1, or a failure indicated by a 0, is returned to the Terminal State Manager (TSM) program which continues the login process.

LAS Mode Only: When accessing the TOE via its local system console, the ISSO and SO users are exempt from checks pertaining to consecutive authentication failures.

This section maps to the following SFR(s):

- FIA_AFL.1
- FIA_ATD.1(HU)
- FIA_UAU.5
- FIA_UID.2(BASE)

7.2.2.1.3 Interactive login and related mechanisms (IA.3)

There are multiple mechanisms for interactive login and similar activities:

- the standard *login* program for interactive login sessions on the console of a user's local host
- the *telnet* protocol and the *rlogin* protocol for ordinary interactive login sessions on any host in the system
- the *rsh*, *rcp*, and the *rexec* protocols for remote shell, copy, and single command executions
- the FTP protocol for interactive file transfer
- BAS Mode Only: the *xlock* program that is used to lock active X window sessions
- the *swrole* command for changing roles within a login session
- the *su* command for changing user identity during a session

All of these mechanisms use the common authentication mechanism described above, but only those that create normal interactive sessions use the standard *login* program; others implement special-purpose types of sessions.

All those mechanisms will not display a password that is entered via a keyboard for authentication but provide obscured feedback.

The TOE supports both user-initiated session locking and TSF-initiated session locking. The TSF-initiated session locking allows the administrator to configure the inactivity timeout period. In both cases, the information on the screen is obscured from the viewer. Also, in both cases, the user must enter their password in order to reactivate the session.

Note: *xlock* is not a full login mechanism but uses the same authentication mechanism to re-authenticate a user who has locked an X window session.

This section and its subsections map to the following SFR(s):

- FIA_UAU.1
- FIA_UAU.7(BASE)
- FIA_UID.2(BASE)
- FTA_SSL.1
- FTA_SSL.2

7.2.2.1.3.1 The login program

The *login* program establishes interactive user sessions. The *login* program is part of the Terminal State Manager (TSM) program. This program prompts for a user identity and authentication (e.g., password), and validates them using the common authentication mechanism described above.

In LAS mode only, during login, the user can append the option " –e" to his user name, which allows him to specify a label within his clearance to be used during this session (rather than the default label defined for the user).

Authentication prompting may also be suppressed when appropriate (e.g., *rsh*). If the validation fails, the prompts are repeated until the limits on successive authentication failures are exceeded. Each failure is considered an event that may be audited.

Login establishes a user session as follows:

1. Assigns a session identifier
2. Sets exclusive access for the controlling terminal to the process logging in

3. Calls the common authentication mechanism to check validity of the password provided for the account being accessed, and gains the session security attributes
4. Sets up the user environment
5. Checks for password expiration and if so, prompts for password change
6. The process's user and group identities are changed to those of the user
7. The process's authorizations and privileges are set to those of the user
8. In LAS mode only, the user's sensitivity clearance is set according to the users entry in the sensitivity clearance database
9. In LAS mode only, the process's sensitivity label is set to what was specified by the user, provided that the SL is within the user's sensitivity clearance, or to the user's default SL if no SL was specified
10. In LAS mode only, the process's integrity label is set to the user's default TL
11. In LAS mode only, the user's integrity clearance is set according to the users entry in the integrity clearance database
12. User is changed to his or her home directory
13. Invokes the user's default shell

The *login* program is always invoked with open file descriptors for the controlling terminal, used when prompting for identity and authentication information, and passes control to the user's shell when the session is created. At this point, the user session is established, the user environment is set up, and the program replaces itself, using the *exec* system call, with the user's shell).

7.2.2.1.3.2 Network login

After an initial login on the console or a terminal, access to other hosts within the same security domain may occur through the following network protocols: telnet, rlogin, rsh, rcp, rexec, and FTP (refer to section 7.2.2.1.3.2.6 "File transfer using FTP" for FTP). In LAS mode, connections are restricted to work only on the same sensitivity clearance.

7.2.2.1.3.2.1 Login with telnet

The telnet protocol always requests user identity and authentication by invoking the *login* program, which uses the common authentication mechanism. A user can change identity across a telnet connection if the password for another account is known.

7.2.2.1.3.2.2 Login with rlogin

The rlogin protocol includes user identity as part of the protocol information passed from host to host. User is not permitted to switch identity between hosts using -l option. See the description of *rsh* command execution below for details on the enforcement mechanism.

7.2.2.1.3.2.3 Command execution using rsh

The rsh protocol includes user identity as part of the protocol information passed from host to host. User is not permitted to switch identity between hosts using -l option. The *rshd* program checks to see that the remote and local user names are the same. Remember that remote user ID information flows with the rsh connection request as part of the protocol. The requirement that a privileged/reserved port be used as part of the setup ensures that the information in the protocol flow was created by a trusted process.

7.2.2.1.3.2.4 Command execution using rcp

The RCP protocol includes user identity as part of the protocol information passed from host to host. User is not permitted to switch identity between hosts using -l option. See the description of command execution using *rsh* for details of the enforcement mechanism.

7.2.2.1.3.2.5 Command execution using rexec

The rexec protocol always requires the user to enter a valid user identity and password. The authentication is performed by invoking the common authentication mechanism directly rather than by invoking the *login* program. A user can change their identity if the password is known.

7.2.2.1.3.2.6 File transfer using FTP

The FTP protocol is used to create a special type of interactive session that only permits file transfer activities. An FTP session is validated and created directly by the FTP server, which then executes all the user requests directly, as opposed to invoking a user-specified program.

The FTP server invokes the authenticate() subroutine that uses the common authentication mechanism to validate the user identity and password supplied through FTP protocol transactions. A user can change their identity if the password is known.

7.2.2.1.4 User identity changing (IA.4)

Users can change identity (i.e., switch to another identity) using the *su* command. When switching identities, the login UID is not changed, so all actions are ultimately traceable in the audit trail to the originating user. The primary use of the *su* command within AIX is to allow appropriately authorized individuals the ability to assume the root or other administrative identities. In both LAS mode and root disabled mode systems, the capability to login as the root identity and to su to the root user have been eliminated. In BAS mode in the */etc/security/user* file, login to root is set to false for all users and su is set to true for administrators. This allows an administrative user to login under his/her real identity, then su to the root or other administrative identities.

1. The *su* command invokes the common authentication mechanism to validate the supplied authentication.
2. When using the *su* command to change the ID, the authorizations associated with the ID are also changed.

This section maps to the following SFR(s):

- FAU_GEN.2
- FIA_USB.2

7.2.2.1.5 Login processing (IA.5)

Permissions on the device special files control access to exclusively used public devices. When a user successfully logs in at the local direct attached console, the TSM program changes the ownership of */dev/lft0, /dev/kbd0, /dev/rcm0*, and, in system using X windows, */dev/mouse0* to the login UID of the user and sets the permissions on these devices to be readable and writable by this user. */dev/lft0* is a logical device that provides the users interface to the keyboard and graphics adapter. At system initialization, */dev/lft0* grabs the keyboard and graphics adapter devices. In case of a serially attached ASCII terminal, the tty device associated with the terminal changes ownership to the user that is logged in (for example */dev/tty0*)

The */dev/kbd0* device contains two channels for communication between the keyboard and the device driver. Only one channel is active at any given time. The */dev/lft0* device registers for the first channel when the system boots. The second channel is reserved for the X server (which is not supported by all configurations of the TOE). The permissions on the */dev/kbd0* device restrict that only the user who is logged in on the console can access this device. The logged in user could open the second channel, because he/she has permissions. This would redirect the users own keyboard device. This would pose no threat to the operation of the system. The worst thing that would happen is that the login process would not be able to regain access to the */dev/kbd0* device and no other users would be able to login on the console device until the host was rebooted.

The */dev/mouse0* device contains only one channel, which is grabbed by the */dev/lft0* device on system startup. Attempts to open additional instances of the */dev/mouse0* device will result in an error message.

The login process executes a revoke to invalidate any open file descriptors for */dev/lft0* or the appropriate */dev/ttyN* device held by a previous user. The revoke call modifies the file descriptors entry in the system open file table, causing further attempts to access the device special file based on that file descriptor to return “bad file descriptor”. This ensures that the new login session is isolated from any previous login sessions.

For LAS mode only, users are assigned a default login SL and TL which is the effective SL and effective TL of the user's process after a successful login. If the user does not want to log in at his/her default login SL, the user may choose to supply a different SL at login time by using the -e option of the login command. The SL supplied by the user must be dominated by the user's clearance and contained in the system accreditation range. The TL cannot be specified by the user at login time. The default login SL and TL are defined in the file /etc/security/clear along with the username and clearance for each user.

This section maps to the following SFR(s):

- FIA_USB.2

7.2.2.1.6 Logoff processing (IA.6)

When a user logs off, all files that were opened by the login shell are closed. Files and devices that were opened by background tasks remain open. However, a background job that had access to the console loses that access prior to the next user's login as stated above.

The ownership of */dev/ttyN, /dev/lft0, /dev/kbd0, /dev/mouse0* (on evaluated configurations supporting X windows), and */dev/rcm0* is returned to root when the logoff occurs.

This section maps to the following SFR(s):

- FIA_USB.2

7.2.2.2 Auditing (AU)

This section discusses the implementation of auditing in the evaluated configuration. The data structures and formats are discussed first, followed by how audit is controlled, a description of bin mode auditing, the programs used to post process the audit data, the programs used to review audit data, audit file protection, the potential for audit data loss, and finally the audit privileges.

7.2.2.2.1 Audit record format (AU.1)

The audit record consists of a header that contains information identifying the user and process who generated the record, the status of the event (success or failure), and the CPU ID for the system. The CPU ID field allows the administrator to differentiate between individual machines when merging the contents of multiple audit trails. An optional variable length tail contains extra information about the event, as defined in the */etc/security/audit/events* file. The audit records in the Global environment are also tagged with the WPAR CID so that the trail can also distinguish between WPARs.

The audit record is a fixed length record that contains information about the user who caused the event and whether the event was created due to a success or failure. The audit record is defined in */usr/include/sys/audit.h*.

The audit record format is:

- Magic number for audit record.
- The length of the tail portion of the audit record.
- The name of the event and a null terminator.
- An indication of whether the event describes a successful or failed operation.
- The effective user ID.
- The effective group ID.
- The real user ID; that is, the ID number of the user who created the process that wrote this record.
- The login ID of the user who created the process that wrote this record.
- The program name of the process, along with a null terminator.
- The process ID of the process that wrote this record.
- The process ID of the parent of this process.
- The thread ID.
- The time in seconds at which this audit record was written.
- The nanoseconds offset from time (used during bin recovery and trail merging to ensure proper record ordering).
- CPU identifier.
- Active role IDs.
- WPAR corral ID (CID).
- Effective privileges.
- Sensitivity label.
- Integrity label.

This section maps to the following SFR(s):

- FAU_GEN.1(BASE)
- FAU_GEN.1(LS)
- FAU_GEN.2

7.2.2.2.2 Audit record generation (AU.2)

Audit record generation begins with the detection of an event, and follows the record as it advances to storage.

Event detection is distributed throughout the TSF, both in kernel and user mode. Programs and kernel modules that detect events that may be audited are responsible for reporting these events to the system audit logger. The system audit logger is part of the kernel, and can be accessed via a system call for trusted program auditing, or via a kernel procedure call for supervisor state auditing.

The audit logger is responsible for constructing the complete audit record, including the identity and state information and the event specific information. The audit logger appends the record to the active bin. A bin is a file that is used to store raw audit records before they are processed and stored in the audit trail.

This section maps to the following SFR(s):

- FAU_GEN.1(BASE)
- FAU_GEN.1(LS)
- FAU_GEN.2

7.2.2.2.3 Audit record processing (AU.3)

Audit record processing includes a description of bin mode auditing and the backend processors that are utilized by the audit subsystem.

This section maps to the following SFR(s):

- FAU_SAR.1
- FAU_SAR.3(BASE)
- FAU_SAR.3(LS)
- FAU_SEL.1(BASE)
- FAU_SEL.1(LS)

7.2.2.2.3.1 Bin mode auditing

When bin mode auditing starts, two separate bin files are allocated to store raw audit records by the *auditbin* daemon. When one bin file fills (reaches an administrator configurable threshold), the daemon switches to the other bin file and invokes the processing command specified in the */etc/security/audit/bincmds* file to empty the full cache file.

When that operation is complete, *auditbin* notifies the kernel that it is permitted to reuse the cache file. This mechanism of switching and emptying audit bins continues so long as auditing is enabled. The size a bin file may reach before being considered full is defined in */etc/security/audit/config*.

A bin file begins with a header. The tail is written when the audit bin is switched or when auditing is shut down.

7.2.2.2.3.2 Backend audit processors

There are two backend utilities available for use: *auditcat* and *auditselect*. The backend processor writes the raw audit records to the system audit trail or to a specified file after manipulating them.

Bin mode auditing makes use of *auditcat* and *auditselect*. The result of *auditcat* or *auditselect* can be directed to a file for permanent audit storage.

7.2.2.2.3.2.1 The auditcat command

The *auditcat* command reads audit records from standard input or from a file, and processes the records and sends them to standard output or to the system audit trail.

7.2.2.2.3.2.2 The auditselect command

The *auditselect* command allows to selectively extract individual audit records. The command can be used as both a preprocessing and post-processing tool. As a preprocessing tool, the *auditselect* command serves the same purpose as *auditcat*, but adds the ability to specify conditions that an audit record must meet. This allows a system to be configured to save audit records that relate to login in one file, and audit records that relate to file access in a separate file.

The *auditselect* command utilizes an expression to apply against the current audit record. The expression consists of one or more terms joined by the logical operators && (and), || (or) and ! (not). Each term in the expression describes a field, a relational operator and a value.

The following is an example expression to select all the FILE_Open events:

```
event==FILE_Open
```

The event field identifies that *auditselect* should query based on the name of the event. The operator is equal and the name of the event is FILE_Open.

Event Field Value	Definition
event	Name of the audit event
command	Name of the command that generated the audit event
result	Status of the audit event. The value of the result field must be one of the following: OK, FAIL, FAIL_PRIV, FAIL_AUTH, FAIL_ACCESS, FAIL_DAC. FAIL matches all other error codes.
login	ID of the login user of the process that generated the audit event.
real	ID of the real user of the process that generated the audit event.
pid	ID of the process that generated the audit event.
ppid	ID of the parent of the process that generated the audit event.
tid	ID of the kernel thread that generated the event.
time	Time of day the audit event was generated.
date	Date the audit event was generated.
priv	Privilege name
sl	Sensitivity label name
tl	Integrity label name
role	Role name

Table 18: auditselect event field values

7.2.2.2.4 Audit review (AU.4)

Three different commands exist for the review of audit records in the system:

- *auditselect*

- *auditpr*
- *auditmerge*

The *auditselect* command is described in section 7.2.2.2.3.2.2 "The auditselect command".

The *auditpr* command formats audit records to a display device or to a printer for review. The *auditpr* command also allows the administrator to select which of the fields to include in the output as well as the order to display them. The fields available for inclusion with the output of the *auditpr* command are:

- Audit event
- User's login name
- Event status
- Time the records was written
- Command name
- Real user name
- Process ID
- Parent process ID
- Kernel thread ID
- Role names or IDs of the audited process
- Effective privilege
- Effective sensitivity label (SL)
- Effective integrity label (TL)
- WPAR name

The default values are the audit event, the user's login name, the audit status, the kernel thread ID and the command name *auditselect* allows the administrator to build an expression that will be applied to the stored audit records. The details of the *auditselect* command are listed in section 7.2.2.2.3 "Audit record processing (AU.3)".

The *auditmerge* command provides a method of combining multiple audit trail files into a single audit trail file. These multiple files can come from different hosts, providing a centralized audit analysis function. As the two files are processed, the record with the oldest time stamp that still remains is written into the audit trail. This process continues until there are no more audit records to process. [SecGuide] directs the system administrator to transfer the audit files to be merged to the same host.

The *auditpr* and *auditmerge* commands allow an authorized administrator to read the audit records and convert them into human readable formats. The audit records are either a stream of data or a flat file. In LAS mode, if an administrator is authorized to read the audit records, that administrator can see all audit records regardless of the sensitivity label listed in the audit record. The commands listed in this section allow the authorized administrator to filter audit records based on the sensitivity label values contained in an audit record. They do not prevent authorized administrators from seeing audit records that have certain sensitivity labels. (The sensitivity label(s) in an audit record indicate the sensitivity level of the subject and/or object at the time the audit record was generated. They do not indicate the sensitivity level of the audit record.)

This section maps to the following SFR(s):

- FAU_SAR.1
- FAU_SAR.3(BASE)

- FAU_SAR.3(LS)

7.2.2.2.5 Audit file protection (AU.5)

The audit trail files, configuration files, bin files, and the */audit* directory are protected on each system using normal file system permissions. Each audit file grants read access to the root user and the audit group, and write access to only the root user. The AUDIT authorization is needed for the audit programs that are used to read these files. [SecGuide] instructs the administrator that if the cached and permanent audit trails are kept somewhere other than in the */audit* directory, then the alternate directory must be protected from access by non-root users.

This section maps to the following SFR(s):

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

7.2.2.2.6 Audit record loss prevention (AU.6)

Bin mode auditing is susceptible to the exhaustion of disk space available to the */audit* directory or to a system crash. In the case of a system crash, all data in physical memory is lost, including any audit records that had not yet been flushed to disk. The audit subsystem enforces a 32K byte limit on the size of an individual audit record, and only one audit record can be in transit between a thread and the kernel at any given time. When the system is no longer able to write audit records to the audit bins either the system will stop in “panic” mode or a counter will show the number of audit records lost. This counter is written in an audit record the next time the system is able to produce audit records again. If the TOE stops in case it is unable to write audit records or if the TOE just counts the number of audit record lost is a configuration parameter that can be set by the System Administrator.

[SecGuide] includes instructions to the administrator to backup all files, including audit data, on a regular basis to avoid the loss of data due to hard disk failures.

This section and its subsections map to the following SFR(s):

- FAU_STG.3
- FAU_STG.4
- FMT_MTD.1(AT)

7.2.2.2.6.1 Audit record loss prevention for bin mode auditing

AIX allows an administrator to define a threshold value for the amount of free space in the file system holding the audit files. When the amount of free space in this file system is below this defined threshold value this fact will be reported to an administrator. This allows the administrator to take the appropriate actions to prevent the system to enter the panic mode due to the inability to write events to the audit trail.

AIX provides a panic mode for use with bin mode auditing. The panic mode option halts the host when the current audit bin stops accepting additional records, preventing the unnecessary loss of audit records. This only occurs with the exhaustion of disk space. If a host halts because it cannot collect audit records, the other hosts in the distributed system are not affected, unless the host is acting as the administrative master. [SecGuide] contains instructions for enabling panic mode, as panic mode is not enabled by default.

The result of halting the system because panic mode was invoked would be the loss of any audit data presently in the host's memory that had not been written to disk. In addition, audit records could be lost for operations that were underway but had not yet completed generating audit records. This minimizes the damage caused by the lack of disk space, because only the audit records that are currently in memory are lost.

A recovery process for audit bins exists in the evaluated configuration. If either of the bin files is not empty when audit is started, the *auditbin* daemon executes the bin mode post-processing command to process the bins.

The amount of audit data that can be lost in bin mode is minimized by the use of the **binsize** and **bytethreshold** parameters in the */etc/security/audit/config* file. The **binsize** parameter sets the maximum size a bin may reach before the *auditbin* daemon switches to the other bin, and executes the bin mode post-processing command. The **bytethreshold** parameter sets the amount of data in bytes that is written to a bin before a synchronous update is performed. [SecGuide] states that the **binsize** and **bytethreshold** parameters should be set to 64K bytes each to minimize audit data loss. The amount of audit data that could be lost due to a failure in bin mode is the combination of these two files, or 128K bytes.

7.2.2.2.7 Audit system privileges (AU.7)

The enforcement of the TOE's auditing policies is supported, in addition to the general access control policies (DAC, MAC (LAS mode only), MIC (LAS mode only)), by the following privileges allowing a process to:

1. record/add audit records (PV_AU_ADD)
2. turn on/off auditing or change audit system configuration (PV_AU_ADMIN)
3. query the status of the audit system or the audit mask of a process (PV_AU_PROC)
4. read a file marked as an audit file (PV_AU_READ)
5. write or delete a file marked as an audit file, or mark a file as an audit file (PV_AU_WRITE)
6. obtain all privileges listed in 1 through 5 (PV_AU)

See section 7.2.2.14.7 "File security flags (TP.7) (LAS mode only)" for a description of file security flags that influence the behavior of the audit subsystem.

This section maps to the following SFR(s):

- FAU_SAR.2

7.2.2.3 Discretionary access control (DA)

This section outlines the general DAC policy in AIX as implemented for resources. A subset of these resources are file system objects where access is controlled by one of two policies (i.e., a file system object can only have one policy associated with it at a time):

- AIXC policy – the AIX classic access control policy including the Encrypted File System (EFS)
- NFSv4 policy – the Network File System version 4 (NFSv4) access control policy

The AIXC policy uses permission bits and, optionally, extended permissions and encryption. The extended permissions are in the form of an access control list (ACL) where each entry in the ACL can define the permissions of a specific user or group. The encryption is in the form AES encrypted files using the EFS file system (an extended version of the JFS2 file system). This is described in more detail in the following sections.

The NFSv4 policy uses fine grained permissions. The fine grained permissions are in the form of an ACL where each entry in the ACL can enable a number of fine grained permissions for a user, group, or for everyone. This is described in more detail in the following sections.

Permission bits are the standard UNIX DAC mechanism and are used on all AIX file system named objects. Individual bits are used to indicate permission for read, write, and execute access for the object's owner, the object's group, and all other users (i.e., world). The extended permission and fine grained permission mechanisms are supported only for file system objects and provide a finer level of granularity than do permission bits.

The policies for all resources are based on user identity (and in some cases on group membership associated with the user identity). To allow for enforcement of the DAC policy, all users must be identified and their identities authenticated.

Details of the specific DAC policy applied to each type of resource are covered in the section 7.2.2.3.3 "Discretionary access control: File system objects (DA.3)" and the section 7.2.2.3.4 "Discretionary access control: IPC objects (DA.4)".

The general policy enforced is that subjects (i.e., processes) are allowed only the accesses specified by the class-specific policies. Further, the ability to propagate access permissions is limited to those subjects who have that permission, as determined by the class-specific policies.

The privilege PV_DAC will also grant full access regardless of the setting of permission bits or ACLs. A subset of PV_DAC can be used for explicit overrides, for example PV_DAC_W to override DAC restrictions on any file.

Finally, in a root enabled mode system, a subject with an effective UID of 0 is exempt from all restrictions and can perform any action desired, including the execution of files for which at least one exec DAC bit is set.

DAC provides the mechanism that allows users to specify and control access to objects that they own. DAC attributes are assigned to objects at creation time and remain in effect until the object is destroyed or the object attributes are changed. DAC attributes exist for, and are particular to, each type of object on AIX.

AIXC permission-bit and extended permission policy

A subject whose effective UID matches the file owner ID can change the file attributes, the base permissions, and the extended permissions. Changes to the file group are restricted to the owner.

For new files, the group identifier must either be the current effective group identifier or one of the group identifiers in the concurrent group set. In addition when using a root enabled mode system, a subject whose effective UID is 0 can make any desired changes to the file attributes, the base permissions, the extended permissions, and owning user of the file.

Encryption

The Encrypted File System (EFS) provides for the ability to encrypt and decrypt files using AES encryption. A subject who doesn't know or have access to the key to decrypt the file cannot view the contents of the file, including administrative users who might be able to override the permission bits of a file.

NFSv4 policy

A subject whose effective UID matches the file owner ID can change the file attributes, the base permissions, and the fine grained permissions. (If an object has an NFSv4 ACL, the base permissions (excluding the setuid, setgid,and save text bits) are ignored when making access decisions, but they are set to approximate the value of the ACL.) Additional rules regarding who can manage NFSv4 ACLs and object attributes are provided in section 7.2.2.3.3.1.2.2 "NFSv4 contents policy".

For new files, the group identifier must either be the current effective group identifier or one of the group identifiers in the concurrent group set. In a root enabled mode system, if the subject's effective UID is 0, the group identifier can be any chosen value.

The NFSv4 policy allows for file system objects to inherit ACL entries from the parent directory's ACL. Subdirectories can inherit different entries than other file system objects. The ability to propagate the ACL entries to subdirectories can be limited to just the subdirectories within the parent directory.

7.2.2.3.1 Permission bits (DA.1)

AIX uses standard UNIX permission bits to provide one form of DAC for file system named objects. There are three sets of three bits that define access for three categories of users: the owning user, users in the owning group, and other users. The three bits in each set indicate the access permissions granted to each user category: one bit for read (r), one for write (w) and one for execute (x). Each subject's access to an object is defined by some combination of these bits:

- `rwx` symbolizing read/write/execute
- `r-x` symbolizing read/execute
- `r--` symbolizing read
- `---` symbolizing null

When a process attempts to reference an object protected only by permission bits, the access is determined as follows:

- Effective UID = Object's owning UID and the owning user permission bits allow the type of access requested. Access is granted with no further checks.
- Effective GID, or any supplementary groups of the process = Object's owning GID, and the owning group permission bits allow the type of access requested. Access is granted with no further checks.
- If the process is neither the owner nor a member of an appropriate group and the permission bits for world allow the type of access requested, then the subject is permitted access.
- In root enabled mode, if the process is the root identity, then the subject is permitted read and write access.
- In root enabled mode, if the process is the root identity, and the attempted access is an execution of the object, the access is granted only if at least one of the execution bits is set.
- If none of the conditions above are satisfied and the process does not possess the needed PV_DAC privilege or the appropriate subset of PV_DAC then the access attempt is denied.

As a special case that has been modeled as part of the DAC Policy in this Security Target, a read-only bit for file systems can be set upon mount time, yielding the denial of every write request for the file system.

This section maps to the following SFR(s):

- FDP_ACC.1(PSO-AIXC)

- FDP_ACF.1(PSO-AIXC)
- FDP_ITC.2(BASE)

7.2.2.3.2 Extended permissions (DA.2)

This section and its subsections map to the following SFR(s):

- FDP_ACC.1(PSO-AIXC)
- FDP_ACF.1(PSO-AIXC)
- FDP_ACC.1(PSO-NFS)
- FDP_ACF.1(PSO-NFS)
- FDP_ITC.2(BASE)
- FPT_TDC.1(BASE)
- FPT_TDC.1(VIRT)

7.2.2.3.2.1 AIXC extended permissions

The extended permissions consist of an essentially unlimited number of additional permissions and restrictions for specific users and groups. Each entry in the extended permissions list consists of three parts: an entry type, a set of permissions, and an identity list.

- The entry type is the value permit, deny, or specify (indicating that the entry indicates a set of permissions to be allowed as supplemental to the listed identity(-ies), denied to the listed identity(-ies), or that the permissions permitted and the complementary set denied to the listed identity(-ies) respectively).
- The permission set is zero or more of the permissions read, write, and execute.
- The identity list is one or more values specifying users and/or groups. The entry is applied if the process' effective UID, effective GID, and supplemental groups match all values in the list. The term “match” means that for each value in the identity list, if the value is for a UID, that the specified UID is the same as the process' effective UID, and if the value is for a GID, that the specified GID is either the same as the process' effective GID or the specified GID is included in the process' list of supplemental GIDs.

There is no explicit ordering of entries within the extended permissions. To determine access rights, the kernel takes into account all entries that match the UID or GID of the process. For each entry, the permit and specify bits are added to a permissions list and the deny and bitwise negation of the specify are added to a restrictions list. The restrictions are bitwise removed from the permissions and the resulting list is used in the access determination.

The maximum size for the extended permissions is one memory page (4096 bytes). The entries are variable length. Each entry takes a minimum of 12 bytes (two for the length of the entry, two for the permission type and permissions allowed, two for the number of identity entries, two for the type of identity entry, and four for each UID/GID). As a result, there can be over 300 entries in an extended permissions list, which is in practice unlimited.

Collectively, the file attributes, base permissions, extended permissions, and extended attributes are known as the file AIXC Access Control List (ACL). AIXC ACLs have a textual representation (used with commands such as aclget) and binary representations (for storage in the file system).

When a process attempts to reference an object protected by an ACL, it does so through a system call (e.g., open, exec). If the object has been assigned an ACL, access is determined according to the following algorithm:

A subject must have search permission for every element of the pathname and the requested access for the object. A subject has a specific type access to an object if the type of access is within the union of all permission rights (grant entries) defined in the access control list of the object for the subject and is not within the logical union of all restrictions (deny entries) defined in the access control list of the object for the subject. If no entry in the extended permissions either allows or denies access, the access right defined in the permission bits apply. In any other case access is denied.

7.2.2.3.2.2 NFSv4 fine grained permissions

An NFSv4 ACL consists of a list of entries with the following fields:

- Type Field – This field contains one of the following values:
  - ALLOW – Grants the subject, specified in the Who field, the permission(s) specified in the Mask field.
  - DENY – Denies the subject, specified in the Who field, the permission(s) specified in the Mask field.
- Mask Field – This field contains one or more of the following fine grained permission values:
  - READ_DATA / LIST_DIRECTORY – Read the data from a non-directory object or list the objects in a directory.
  - WRITE_DATA / ADD_FILE – Write data into a non-directory object or add a non-directory object to a directory.
  - APPEND_DATA / ADD_SUBDIRECTORY – Append data into a non-directory object or add a subdirectory to a directory.
  - READ_NAMED_ATTRS – Read the named attributes of an object. (There are no named attributes.)
  - WRITE_NAMED_ATTRS – Write the named attributes of an object. (There are no named attributes.)
  - EXECUTE – Execute a file or traverse/search a directory.
  - DELETE_CHILD – Delete a file or directory within a directory. (Applies to directories.)
  - READ_ATTRIBUTES – Read the basic (non-ACL) attributes of a file.
  - WRITE_ATTRIBUTES – Change the times associated with a file or directory.
  - DELETE – Delete a file or directory.
  - READ_ACL – Read the ACL.
  - WRITE_ACL – Write the ACL.
  - WRITE_OWNER – Change the owner and group.
  - SYNCHRONIZE – Synchronize access. (Exists for compatibility with other NFSv4 clients, but has no implemented function.)
- Flags Field – This field defines the inheritance capabilities of directory ACLs and indicates whether the Who field contains a group or not. The field contains zero or more of the following flags:
  - FILE_INHERIT – Specifies that, in this directory, newly created non-directory objects will inherit this entry.
  - DIRECTORY_INHERIT – Specifies that, in this directory, newly created subdirectories will inherit this entry.

    - NO_PROPAGATE_INHERIT – Specifies that, in this directory, newly created subdirectories will inherit this entry, but these subdirectories will not pass this entry to their newly created subdirectories.
    - INHERIT_ONLY – Specifies that this entry does not apply to this directory, only to the newly created objects that inherit this entry.
    - IDENTIFIER_GROUP – Specifies that the Who field represents a group; otherwise, the Who field represents a user or a special Who value.
- Who Field – This field contains one of the following values:
    - User – Specifies the user that this entry applies to.
    - Group – Specifies the group that this entry applies to.
    - Special – This attribute can be one of the following values:
        - OWNER@ - Specifies that this entry applies to the owner of the object
        - GROUP@ - Specifies that this entry applies to the owning group of the object.
        - EVERYONE@ - Specifies that this entry applies to all users of the system including the owner and group.

If the ACL is empty, access requires the PV_DAC_R privilege or, in the case of root enabled mode, an effective UID of 0.

The owner of an object implicitly has the following mask values regardless of what the ACL may or may not contain:

- READ_ACL
- WRITE_ACL
- READ_ATTRIBUTES
- WRITE_ATTRIBUTES

APPEND_DATA is implemented as WRITE_DATA. Effectively, there's no functional distinction between WRITE_DATA and APPEND_DATA. Both values must be set or unset in unison which is enforced by the TOE.

Object ownership can be modified through the use of WRITE_OWNER. Section 7.2.2.3.3.1.2.2 "NFSv4 contents policy" details how WRITE_OWNER works. When the owner is changed, the setuid bit is turned off. When the group is changed, the setgid bit is turned off.

The inheritance flags only have meaning in a directory's ACL and only apply to objects that are created in the directory after the inheritance flags have been set (i.e., existing objects are not affected by inheritance changes to the parent directory's ACL).

The entries in an NFSv4 ACL are order dependent. To determine if the requested access is allowed, each entry is processed in order. Only entries which have a Who field that matches the effective UID, if a user is specified in the entry, or effective GID, if a group is specified in the entry, of the subject are considered. Each entry is processed until all of the bits of the requester's access have been ALLOWED. Once an access type has been ALLOWED by an entry, it is no longer considered in the processing of later entries. If a DENY entry is encountered where the requester's access for that mask value is necessary and undetermined, the request is denied. If the evaluation reaches the end of the ACL, the request is denied.

The maximum supported ACL size is 64KB. Each entry in an ACL is of variable length and 64KB is the only limit on an entry.

7.2.2.3.3 Discretionary access control: File system objects (DA.3)

The Discretionary Access Control (DAC) policy is described above. This section describes the details of DAC policies as they apply to file system objects.

This section and its subsections map to the following SFR(s):

- FDP_ACC.1(PSO-AIXC)
- FDP_ACC.1(PSO-NFS)
- FDP_ACF.1(PSO-AIXC)
- FDP_ACF.1(PSO-NFS)
- FDP_ITC.2(BASE)
- FMT_MSA.1(PSO-AIXC)
- FMT_MSA.1(PSO-NFS)
- FMT_MSA.4(PSO)
- FMT_REV.1(OBJ)

7.2.2.3.3.1 Common file system access control

This section describes the common DAC policy applied to file system objects, including policies for object contents and attributes.

7.2.2.3.3.1.1 DAC contents policy

7.2.2.3.3.1.1.1 AIXC permission-bit and extended permissions contents policy

The permission-bit and ACL DAC policy determines the effective access that a process may have to the contents of a file system object: some combination of read(r), write (w), and execute (x). In general, read access permits the object's contents to be read by a process, and write permits them to be written; execute is interpreted differently for different object types. Some object types (unnamed pipes, symbolic links) do not use the permission bits at all.

7.2.2.3.3.1.1.2 NFSv4 contents policy

The NFSv4 policy determines the effective access that a process may have to the contents of a file system object. How this policy works is described in DA.2. Some object types (unnamed pipes, symbolic links) do not use the NFSv4 policy at all. The permission bits (excluding the setuid, setgid, and save text bits), specifically the user/group/other bits, are ignored when making access control decisions if an NFSv4 ACL exists on the object.

7.2.2.3.3.1.2 DAC attributes policy

7.2.2.3.3.1.2.1 AIXC permission-bit and extended permissions contents policy

In general, a process must be the object's owner, or have privilege, to change the objects attributes, and there are no DAC restrictions on viewing the attributes, so any process may view them. However, the following are exceptions to the rule:

- The permission bits and ACL (permission bits, extended permissions and attributes) of an object may be changed by an owner or by a subject having the PV_DAC privilege or the appropriate subset of PV_DAC or, in root enabled mode, by the root identity.

- The owning group ID of an object may be changed by an owner, but only to a group of which the process is currently a member, unless it has PV_DAC or the appropriate subset of PV_DAC or, in root enabled mode, by the root identity.
- The owning user ID of an object may only be changed by an administrator with PV_DAC or the appropriate subset of PV_DAC or, in root enabled mode, by the root identity.

7.2.2.3.3.1.2.2 NFSv4 contents policy

The NFSv4 policy provides control over who can read and write the attributes of an object. A subject with the PV_DAC privilege or, in root enabled mode, effective UID 0 can always override the NFSv4 policy. The object owner can allow others to read and write the attributes of an object using the READ_ATTRIBUTES, WRITE_ATTRIBUTES, READ_NAMED_ATTRS, and WRITE_NAME_ATTRS attributes of the ACL mask. The owner can control who can read and write the ACL using the READ_ACL and WRITE_ACL attributes of the ACL mask. The object owner always has READ_ATTRIBUTES, WRITE_ATTRIBUTES, READ_ACL, and WRITE_ACL access. The owner can also allow others to change the owner and group of the object using the WRITE_OWNER attribute. An object owner cannot change the owner or group of the object by default, but the owner can add a WRITE_OWNER entry to the ACL specifying themselves, or the object could inherit an ACL entry which specifies a WRITE_OWNER entry with a Who value of OWNER@.

There are some exceptions to the rules in root enabled mode:

- If the object is owned by UID 0, only UID 0 can change the owner, but the group can still be changed by a subject with WRITE_OWNER.
- If the object has a non-UID 0 owner, a non-UID 0 user with WRITE_OWNER can only change the owner to himself.
- The group can be changed to any group in the subject's concurrent group set with the exception that it can never be changed to GID 0 or GID 7 even if these two groups are in the concurrent group set of the subject.

7.2.2.3.3.1.3 DAC defaults

7.2.2.3.3.1.3.1 AIXC permission-bit and extended permissions defaults

The default access control on newly created FSOs is determined by the permissions associated with the directory where the FSO was created, the effective user ID, group ID, and umask value of the process that created the FSO, and the specific permissions requested by the program creating the FSO.

- The owning user of a newly created FSO will be the effective UID of the creating process.
- If the setgid bit is set on the containing directory, then the owning group of a newly created FSO will be the owning group of the containing directory. If the setgid bit is not set on the containing directory, then the owning group of the newly created FSO will be the effective GID of the creating process.
- The initial access permissions on the FSO are those specified by the creating process bitwise ANDed with the one's complement of the umask value. For example, if a program specified initial permissions of 0664 (read/write for owner, read/write for group, and read for world) but the umask value were set to 0027 (prevent write for group or world, prevent all permissions for world), then the initial file permissions would be set to 0640 (or 0664 bit-and 0750).
- There are initially no extended permissions associated with an FSO. Extended permissions can be set by applications or by users using AIX commands.

Base and extended access permissions can be changed by any process with an effective UID equal to the owning UID of the FSO, providing that the effective UID has at least the execute permission to the containing directory. Note that since a file may have multiple hard links, the process can use any of the containing directories (e.g., if there is any directory containing a link to the file, then that path could be used as a means to get to the file and change its permissions).

7.2.2.3.3.1.3.2 NFSv4 defaults

If the parent directory does not have any NFSv4 inheritance entries applicable to the FSO being created, then the FSO will be created using the AIXC defaults mentioned above. Otherwise, the parent directory's inheritance entries will be copied into and become the ACL of the newly created FSO as per the rules of NFSv4 inheritance. NFSv4 inheritance is described in section 7.2.2.3.2 "Extended permissions (DA.2)".

- The owning user of a newly created FSO will be the effective UID of the creating process.
- If the setgid bit is set on the containing directory, then the owning group of a newly created FSO will be the owning group of the containing directory. If the setgid bit is not set on the containing directory, then the owning group of the newly created FSO will be the effective GID of the creating process.

The permission bits are set on the object to approximate the values contained in the ACL.

7.2.2.3.3.1.4 DAC revocation on file system objects

With the exception of NFS objects, file system objects (FSOs) access checks are performed when the FSO is initially opened, and are not checked on each subsequent access. Changes to access controls (e.g., revocation) are effective with the next attempt to open the FSO.

For NFS objects, access is checked for each operation. A change to the access rights for an NFS FSO take effect as of the next NFS request.

In cases where the administrator determines that immediate revocation of access to an FSO is required, the administrator can reboot the computer, resulting in a close on the FSO and forcing an open of the FSO on system reboot. This method is described in [SecGuide].

Applications that want to revoke tty access of other processes can use the *revoke()* and *frevoke()* system calls to revoke all current access of other processes, forcing them to reopen the file and to undergo the associated access checks again.

7.2.2.3.3.2 DAC: Ordinary file

Ordinary files support the concept of execution. Execute access is required to execute the file as a program or script. When an executable file has the set-user-ID or set-group-ID flags set, and the file owner or file group is not the same as the process's current effective user-ID or group-ID the executing program changes the process's security attributes. Otherwise the attributes remain unchanged. AIX doesn't support set-UID or set-GID scripts with this mechanism.

Note that the *privcmds* table can override the set-user-ID and/or set-group-ID flags. See section 7.2.2.6.2 "Privileged commands (PV.2)" for more details.

7.2.2.3.3.3 DAC: Directory

The execute access for directories governs the ability to traverse the directory as part of a pathname. A process must have execute access in order to traverse the directory during pathname resolution.

Directories may not be written directly, but only by creating, renaming, and removing (unlinking) objects within them. These operations are considered writes for the purpose of the DAC policy.

7.2.2.3.3.4 DAC: UNIX domain socket special file

UNIX domain socket files are treated as files in the AIX file system from the perspective of access control, with the exception that using the bind or connect system calls requires that the calling process must have both read and write access to the socket file.

UNIX domain sockets exist in the file system name space. The socket files are supported by both the AIXC and NFSv4 policies.

UNIX domain sockets consist of a socket special file (managed by the File System) and a corresponding socket structure (managed by IPC). The VFS controls access to the socket based upon the caller's rights to the socket special file.

7.2.2.3.3.5 DAC: Named pipes

Named pipes are treated identically to any other file in the AIX file system from the perspective of access control. Therefore both AIXC and NFSv4 policies are supported by named pipes. For this reason named pipes are listed as file system objects (although they are used for interprocess communication). Note that named pipes follow the rules for IPC objects, if no ACLs are used (which probably is the normal case they are used).

7.2.2.3.3.6 DAC: Device special file

The access control scheme described for FSOs is used for protection of character and block device special files. The DAC settings on most device special files are configured to allow read and write access by the root user, and read access by privileged groups. With the exception of terminal and pseudo-terminal devices and a few special cases (e.g., */dev/null* and */dev/tty*), devices are configured to be not accessible to normal users.

7.2.2.3.3.7 DAC: Special cases for NFS file systems

An NFS file system may contain any of the supported object types of the underlying file system. That includes device special files. Non-regular files or directories which occur on NFS file systems are treated similar to objects defined on the local file system -- a device special file on an NFS mounted file system will reference the underlying device on the local system. It would not reference the device on the remote system.

DAC checks by the NFS server for file contents permit a subject with the same effective owning user ID as the file to have access to the contents regardless of the DAC attributes. This is used to support the standard UNIX semantics for access to open files, because such access is not re-validated when a file's DAC attributes change. This special case relies on the property that, ordinarily, only a file's owner changes its DAC while the file is open, and it is thus sufficient to handle the owner specially.

DAC changes do have immediate effect for users other than the owner, unlike local files: if an NFS-accessed file's DAC is changed to deny access, any subsequent read or write operation to an open file will fail if the operation would no longer be permitted by the new DAC attributes.

However, this can never grant additional access, because the client would have checked the access when the file was opened and not permitted more access than the DAC attributes allowed at open time.

The file system maintains a “handle” on the credentials which were used at the time an NFS file was opened. It is those credentials which are used to reference files via NFS, not the current process credentials which might be modified by *setuid()*.

7.2.2.3.4 Discretionary access control: IPC objects (DA.4)

In general, WPARs limit the scope of IPC objects to the scope of the WPAR. Thus, a WPAR can only see the IPC objects within itself.

This section maps to the following SFR(s):

- FDP_ACC.1(TSO)
- FDP_ACF.1(TSO)
- FMT_MSA.1(TSO)
- FMT_MSA.3(TSO)
- FMT_REV.1(OBJ)

7.2.2.3.4.1 DAC: Shared memory

For shared memory segment objects (henceforth SMSs), access checks are performed when the SMS is initially attached, and are not checked on each subsequent access. Changes to access controls (e.g., revocation) are effective with the next attempt to attach to the SMS.

In cases where the administrator determines that immediate revocation of access to a SMS is required, the administrator can reboot the computer, thus destroying the SMS and all access to it.

This method is described in [SecGuide]. Since a SMS exists only within a single host in the distributed system, rebooting the particular host where the SMS is present is sufficient to revoke all access to that SMS.

If a process requests deletion of a SMS, it is not deleted until the last process that is attached to the SMS detaches itself (or equivalently, the last process attached to the SMS terminates).

However, once a SMS has been marked as deleted, additional processes cannot attach to the SMS and it cannot be undeleted.

The default access control on newly created SMSs is determined by the effective user ID and group ID of the process that created the SMS and the specific permissions requested by the process creating the SMS.

- The owning user and creating user of a newly created SMS will be the effective UID of the creating process.
- The owning group and creating group of a newly created SMS will be the effective GID of the creating process.
- The creating process must specify the initial access permissions on the SMS, or they are set to null and the object is inaccessible until the owner sets them.
- SMSs do not have extended permissions.
- SMSs do not support NFSv4 ACLs.

Access permissions can be changed by any process with an effective UID equal to the owning UID or creating UID of the SMS. In root enabled mode, access permissions can also be changed by any process with an effective UID of 0, also known as running with the root identity.

7.2.2.3.4.2 DAC: Message queues

For message queues, access checks are performed for each access request (e.g., to send or receive a message in the queue). Changes to access controls (e.g., revocation) are effective upon the next request for access. That is, the change affects all future send and receive operations, except if a process has already made a request for the message queue and is waiting for its availability (e.g., a process is waiting to receive a message), in which case the access change is not effective for that process until the next request.

If a process requests deletion of a message queue, it is not deleted until the last process that is waiting for the message queue receives its message (or equivalently, the last process waiting for a message in the queue terminates). However, once a message queue has been marked as deleted, additional processes cannot perform messaging operations and it cannot be undeleted.

The default access control on newly created message queues is determined by the effective user ID and group ID of the process that created the message queue and the specific permissions requested by the process creating the message queue.

- The owning user and creating user of a newly created message queue will be the effective UID of the creating process.
- The owning group and creating group of a newly created message queue will be the effective GID of the creating process.
- The initial access permissions on the message queue must be specified by the creating process, or they are set to null and the object is inaccessible until the owner sets them.
- Message queues do not have extended permissions.
- Message queues do not support NFSv4 ACLs.

Access permissions can be changed by any process with an effective UID equal to the owning UID or creating UID of the message queue. Access permissions can also be changed by any process having the appropriate privilege or, in root enabled mode, with an effective UID of 0.

7.2.2.3.4.3 DAC: Semaphores

For UNIX System V (SysV) semaphores, access checks are performed for each access request (e.g., to lock or unlock the semaphore). Changes to access controls (e.g., revocation) are effective upon the next request for access. That is, the change affects all future SysV semaphore operations, except if a process has already made a request for the SysV semaphore and is waiting for its availability, in which case the access change is not effective for that process until the next request.

In cases where the administrator determines that immediate revocation of access to a SysV semaphore is required, the administrator can reboot the computer, thus destroying the semaphore and any processes waiting for it. This method is described in [SecGuide]. Since a SysV semaphore exists only within a single host in the distributed system, rebooting the particular host where the semaphore is present is sufficient to revoke all access to that semaphore.

If a process requests deletion of a SysV semaphore, it is not deleted until the last process that is waiting for the semaphore obtains its lock (or equivalently, the last process waiting for the semaphore terminates). However, once a SysV semaphore has been marked as deleted, additional processes cannot perform semaphore operations and it cannot be undeleted.

The default access control on newly created SysV semaphores is determined by the effective user ID and group ID of the process that created the semaphore and the specific permissions requested by the process creating the semaphore.

- The owning user and creating user of a newly created SysV semaphore will be the effective UID of the creating process.
- The owning group and creating group of a newly created SysV semaphore will be the effective GID of the creating process.
- The initial access permissions on the SysV semaphore must be specified by the creating process, or they are set to null and the object is inaccessible until the owner sets them.
- SysV semaphores do not have extended permissions.
- SysV semaphores do not support NFSv4 ACLs.

Access permissions can be changed by any process with an effective UID equal to the owning UID or creating UID of the semaphore, and can be overridden by privileges or, in root enabled mode, by any process with an effective UID of 0.

No security claims are made for non-SysV semaphores.

In addition to the regular IPC semaphores, AIX also supports memory mapped semaphores that are accessible within the memory mapped address space of processes that can share the mapped address space. There is no explicit access control on these semaphores (handled by the *msem_init(), msem_lock(), msem_unlock(), msem_remove(), msleep()*, and *mwakeup()* subroutines). The mmap routines map files into shared memory and all access control is performed via the DAC protection mechanisms of the mapped files or memory.

7.2.2.3.5 Discretionary access control: TCP connections (DA.5) (BAS mode only)

This section applies to BAS mode only.

TCP based services can be protected with ACLs as well. By specifying TCP port, host/network, user/group combinations, ports can be restricted to specific hosts and/or users. For example, specifying port 6000 (X server port), machine *colorado*, and user *joe*, only user *joe* coming from machine *colorado* will be able to connect to the X server port. The remote hosts use TCP to send the information about the user together with the connection request. AIX checks the user information against the ACLs and either allows or denies the connection.

AIX stores the ACLs in the */etc/security/acl* file. TCP ports listed in the */etc/security/services* file are exempt from the ACL checks.

With the DACinet feature of AIX, the concept of privileged ports (ports that can only be opened by the superuser, typically all ports below 1024) is extended so that any port can be a privileged port. A bitmap of privileged ports is defined to hold information on whether a port is privileged. A system administrator can modify this the set of privileged ports using the *dacinet* command. The information used to build the bitmap is stored in the */etc/security/priv* file. The bitmap is loaded into the kernel when the system starts.

This section maps to the following SFR(s):

- FDP_ACC.1(TCP)
- FDP_ACF.1(TCP)
- FMT_MSA.1(TCP)
- FMT_MSA.3(TCP)

7.2.2.3.6 Discretionary access control: Encrypted files (DA.6)

The JFS2 file system supports the Encrypted File System (EFS). EFS allows individual files to be selectively encrypted and it allows for the files in a directory to be encrypted by default. EFS can be enabled on an existing file system, but files on that system will not be automatically encrypted by the enablement.

For each encrypted file, a random symmetric file encryption key (called the file key) is generated to encrypt the file. The key size of the file key matches the algorithm used (e.g., a random 256-bit key is generated for AES 256-bit). The symmetric encryption mechanisms supported by the evaluated configuration are:

- AES 128-bit CBC (default)
- AES 192-bit CBC
- AES 256-bit CBC

The file key is than wrapped with the public key of the user creating the file and stored as part of the file's meta data in the file system. For other users and groups to have access to the file's content, the file key must be wrapped with their public keys too and stored in the file's meta data.

When EFS is enabled, users and groups must have asymmetric (public/private) keys in the form of X.509v3 certificates. These keys are stored in individual PKCS #12 keystores for both users and groups. The user and group keystores are restricted to root read/write access only using file system DAC. The private keys are protected within the keystore using AES 256-bit encryption and an encryption key. For user keystores, the keystore encryption key is generated from a user supplied password (e.g., the user's login password). For group keystores, the password is generated by AIX.

When the user's keystore is opened using the correct password (typically at login time), the keys are loaded and stored in the kernel to improve the performance of EFS. The kernel protects the private keys from unauthorized access while the keys reside in the kernel. The keys are scrambled (i.e., not stored in the clear) while in the kernel.

The certificates can be generated using one of the following algorithms:

- RSA 1024-bit (default)
- RSA 2048-bit

The CLiC module is used to implement the cryptographic functions of EFS. (The CLiC version is specified in section 1.5.3.)

This section maps to the following SFR(s):

- FCS_CKM.1(SYM)
- FCS_CKM.1(RSA)
- FCS_CKM.4
- FCS_COP.1(CLIC-ENC)
- FCS_COP.1(CLIC-SGN)
- FCS_RNG.1(CLIC)
- FDP_ACC.1(PSO-AIXC)
- FDP_ACC.1(PSO-NFS)
- FDP_ACF.1(PSO-AIXC)
- FDP_ACF.1(PSO-NFS)
- FIA_ATD.1(HU)

7.2.2.4 Workload Partitions (WP)

7.2.2.4.1 WPAR information control (WP.1)

A System WPAR works like an independent AIX installation. The characteristics of a System WPAR are:

- Administration: Internally, each System WPAR has its own administrator(s) which can add and delete users, groups, modify RBAC settings, modify MAC and MIC labels (LAS mode only), etc. of the System WPAR. These changes are contained within the System WPAR and, thus, are independent of the Global environment.
- Authentication: The authentication subsystem for a System WPAR can be different than the Global environment. Users log into the System WPAR using remote login commands like *rlogin* and *telnet* and by using the *clogin* command.
- Process space: Processes in a System WPAR can only see other processes within the same System WPAR. Thus, a System WPAR's processes can only create IPC objects and send signals to processes within its System WPAR. Processes in the Global environment can signal processes in a System WPAR. The */dev/kmem* device does not exist in a System WPAR.
- Network: Each System WPAR has its own IP address(es). This allows for network communications between WPARs. Raw socket access is not allowed without the WPAR being assigned the privilege to access raw sockets.
- File system space: System WPAR file systems can be isolated from other System WPARs. This is accomplished through the use of the *chroot()* system call and the *mount()* system call. The Global environment administrator controls the amount of file system isolation. When the System WPAR is started, the Global environment effectively chroot's and/or mount's the System WPAR to its file system space. Like any AIX system, file systems can be shared among WPARs.
- Kernel: Only one kernel image exists which performs the virtualization and, thus, is shared by all WPARs and the Global environment in an LPAR.
- Libraries & Kernel extensions: In general, only one instance of a library and kernel extension exists and is shared by all WPARs and the Global environment. It is possible for unique libraries to be loaded by a System WPAR and, under controlled conditions, for a kernel extension to be loaded by a System WPAR.
- Auditing: Each System WPAR has its own audit trail and has control over its own audit trail (e.g. enable/disable auditing within the System WPAR). The Global environment can independently audit the System WPAR (regardless of the state of the System WPAR audit subsystem) placing the audit records into the Global environment's audit trail.
- Authority: The actual power that a System WPAR can exert on a system is controlled by the Global environment administrator that creates and/or modifies the System WPAR.
- Other System WPARs: Each System WPAR looks like a separate AIX system to other WPARs.

Each WPAR has a unique ID called a CID (corral ID) which the kernel uses to support separation between the WPARs and to uniquely identify/track a WPAR in subsystems like the Global environment's auditing subsystem.

Privileges can be assigned to a WPAR allowing the WPAR creator to control the capabilities of a WPAR.

This section maps to the following SFR(s):

- FDP_ACC.2(VIRT)

- FDP_ACF.1(VIRT)
- FDP_ETC.2(VIRT)
- FDP_IFC.2(VIRT)
- FDP_IFF.1(VIRT)
- FDP_ITC.2(VIRT)
- FIA_UID.2(VIRT)
- FMT_MSA.1(VIRT-CACP)
- FMT_MSA.1(VIRT-CIFCP)
- FMT_MTD.1(VIRT-COMP)
- FPT_TDC.1(VIRT)

7.2.2.5 Role-based access (RA)

7.2.2.5.1 Role-based access control (RA.1)

AIX provides a role-based access control (RBAC) mechanism. By default, RBAC contains the following administrative roles:

- Information System Security Officer (ISSO)
- System Administrator (SA)
- System Operator (SO)

It contains the following non-administrative role:

- Normal user (This is the default role when a user has no explicit role assigned to them or no explicit role active.)

RBAC provides the ability to define other site-based roles. The TOE associates authorization sets with a role and ensures that the authorizations required to assume a role are satisfied before allowing operations associated with the role to be performed. Through the use of authorizations, the scope of a role can be reduced or increased.

By default, only users with the ISSO role can manage roles of users. Thus, users with the ISSO role can delegate roles to other users including the sub-delegation of the ISSO role.

AIX supports up to 8 roles per user session. A user can switch between assigned roles by using the *swrole* (switch role) command. This command creates a new shell process and assigns the requested roles to the new shell. This allows a user to add or delete roles from their active set of roles within their login session. Users can only *swrole* to roles that have been assigned to them.

The roles and authorizations tables for RBAC are maintained in user space, compiled, and loaded into the kernel. When the tables are compiled, they are checked for consistency. The kernel loads the compiled tables during the boot process (if compiled tables do not exist, the boot process will attempt to create compiled tables). Through this methodology, the TOE preserves a secure state and can recover to a consistent, secure state.

System WPAR administrators can create and modify roles independently of the Global environment. The modifications affect only the modified System WPAR.

This section maps to the following SFR(s):

- FDP_ACC.1(RBAC), FDP_ACF.1(RBAC)
- FMT_MSA.1(RBAC-ADM)

- FMT_MSA.1(RBAC-AUTH)
- FMT_MSA.1(RBAC-DFLT)
- FMT_MSA.1(RBAC-USR)
- FMT_MSA.3(RBAC)
- FMT_MTD.1(AM-MD)
- FMT_SMR.2
- FPT_FLS.1(RBAC)
- FPT_RCV.1
- FPT_RCV.4
- FTA_LSA.1(RBAC)
- FTA_TSE.1(RBAC)

7.2.2.5.2 n-man access control (RA.2)

Four eyes principle or n-man rule based access control is implemented on top of RBAC. Commands that need authorization by more than one role are listed in the *privcmds* database with the *authroles* attribute which specifies which roles need to authenticate before the command can be executed. This command is no longer directly executable (which is enforced by the kernel), but instead needs to be run under the control of the *authexec* command which performs the authentication before running the command. See section 7.2.2.6.2 for more information on the *privcmds* database.

This section maps to the following SFR(s):

- FMT_MTD.1(AM-AP)
- FMT_MTD.1(AM-MA)

7.2.2.6 Privileges (PV)

A privilege is an attribute of a process that allows the process to bypass specific restrictions and limitations of the system (DAC, MAC, MIC, TCB). Privileges are used to override security constraints, to permit expanded use of certain system resources such as memory and disk space, and to adjust the performance and priority of the process. In addition, privileges can be used directly within a user-level program that is responsible for mediating or enforcing security.

The privileges for AIX allow for the administration of the system without the use of the all powerful root administrator ID by granting explicit privileges to users (via authorizations) or commands or WPARs.

Privilege sets can be attached to both subjects and (indirectly via *privcmds* to) objects, but object privilege sets are used only on executable files and only for modifying the process privilege sets when the file is executed and by WPARs for controlling the capabilities of a WPAR.

The privileges enforced by the TOE's reference monitor are as follows:

- Audit privileges (identified in section 7.2.2.2.7 "Audit system privileges (AU.7)")
- Authorization privileges (identified in FDP_ACF.1(AUTH))
- DAC privileges (identified in FDP_ACF.1(PSO-AIXC), FDP_ACF.1(PSO-NFS), FDP_ACF.1(TSO), FDP_ACF.1(TCP))
- MAC privileges (identified in FDP_IFF.2(LS))
- MIC privileges (identified in FDP_IFF.2(MIC))
- Network/Driver/STREAMS privileges (identified in FDP_IFF.2(TN))

- TCB privilege (identified in FDP_ACF.1(TCB)

7.2.2.6.1 Process privilege sets (PV.1)

There are three types of process privilege sets:

- EPS - effective privilege sets
- MPS - maximum privilege sets
- LPS - limiting privilege sets

The EPS is used to actually override system restrictions. A process can add or remove privileges from its own EPS subject to the limitations imposed by the MPS.

The MPS is the set of privileges over which a process has control. The MPS is always a superset of the process's EPS. A process can always remove a privilege from its MPS. A process's MPS can only be increased if the process has the appropriate privilege, and even then it is restricted by the LPS of the process. The MPS of a process can also be modified when the process runs an executable file, but this too is limited by the process's LPS.

The LPS represents the maximum possible privilege set that the process can have. The LPS is always a superset of the MPS. Any process can remove a privilege from its LPS, but there is no override mechanism to add a privilege to the LPS. A process cannot acquire privileges from an executable file if the privileges are not in the process's LPS.

Privileges are inherited by child processes just like the IDs associated with the process.

This section maps to the following SFR(s):

- FAU_SAR.1, FAU_SAR.2
- FDP_ACC.1(PSO-AIXC), FDP_ACF.1(PSO-AIXC)
- FDP_ACC.1(PSO-NFS), FDP_ACF.1(PSO-NFS)
- FDP_ACC.1(TSO), FDP_ACF.1(TSO)
- FDP_ACC.1(TCP), FDP_ACF.1(TCP)
- FDP_ACC.1(AUTH), FDP_ACF.1(AUTH)
- FDP_ACC.1(TCB), FDP_ACF.1(TCB)
- FDP_IFC.1(MIC), FDP_IFF.2(MIC)
- FDP_IFC.1(TN), FDP_IFF.2(TN)
- FDP_IFC.2(LS), FDP_IFF.2(LS)
- FMT_MTD.1(PRIVS)

7.2.2.6.2 Privileged commands (PV.2)

AIX provides a Privileged Commands (*privcmds*) mechanism that assigns privileges to commands (both binaries and shell scripts) based on the role(s) the user has currently active. The mechanism uses a kernel table that contains the list of privileged commands along with the authorizations, privileges, effective uid, and effective gid for each command. For all commands executed, the command is first located in the *privcmds* table prior to execution by the kernel. If the command is found in the table, the kernel deduces the user's authorization from the user's active set of roles. If the user has the proper authorizations to execute the command (as defined by the **accessauths** attribute in the table), the command will be executed as a privileged command where the kernel creates a new process, assigns the privileges from the table's **innateprivs** attribute, conditionally

assigns the privileges from the table's **authprivs** attribute, assigns the effective uid/gid from the table's **euid** and **egid** attributes, and executes the command, overriding (ignoring) the DAC execution permissions and file system setuid/setgid bits existing on the command.

Each command defined in the table can include the following attributes:

- **accessauths** – Specifies the access authorizations where the user's current session must have at least one of the specified authorizations in order to execute it as a privileged command.
- **authprivs** – Specifies additional privileges to be assigned to the program based on the authorizations of the user's current session if it's executed as a privileged command.
- **authroles** – Specifies the list of roles required by the command to support the n-man rule functionality. Commands containing this attribute can only be executed using the *authexec* command.
- **inheritprivs** - Specifies the privileges to be passed to child processes.
- **innateprivs** – Specifies the privileges to be assigned to the program if it's executed as a privileged command.
- **euid** – Specifies the effective user ID to be assigned to the program if it's executed as a privileged command.
- **egid** – Specifies the effective group ID to be assigned to the program if it's executed as a privileged command.
- **ruid** - Specifies the real user ID to be assigned to the program if it's executed as a privileged command.
- **secflags** - Specifies the security flags. Currently **FSF_EPS** is the only defined flag. **FSF_EPS** loads the process maximum privilege set inot the effective privilege set upon execution.

Both the **accessauths** attribute and **authprivs** attribute can contain the following special values:

- ALLOW_OWNER – Allows the user that owns the command to execute the command as a privileged command.
- ALLOW_GROUP – Allows the group that owns the command to execute the command as a privileged command.
- ALLOW_ALL – Allows everyone to execute the command as a privileged command.

In the evaluated configuration, shell scripts defined in the *privcmds* table should not have the ALLOW_ALL value assigned to them in the **accessauths** attribute and/or the **authprivs** attribute.

The *privcmds* table is contained in the */etc/security/privcmds* file and protected from unauthorized access (specific authorizations are required in order to modify this file).

This section maps to the following SFR(s):

- FAU_SAR.1, FAU_SAR.2
- FDP_ACC.1(PSO-AIXC), FDP_ACF.1(PSO-AIXC)
- FDP_ACC.1(PSO-NFS), FDP_ACF.1(PSO-NFS)
- FDP_ACC.1(TSO), FDP_ACF.1(TSO)
- FDP_ACC.1(TCP), FDP_ACF.1(TCP)
- FDP_ACC.1(AUTH), FDP_ACF.1(AUTH)
- FDP_ACC.1(TCB), FDP_ACF.1(TCB)
- FDP_IFC.1(MIC), FDP_IFF.2(MIC)
- FDP_IFC.1(TN), FDP_IFF.2(TN)

- FDP_IFC.2(LS), FDP_IFF.2(LS)
- FMT_MTD.1(AM-AP)
- FMT_MTD.1(AM-MA)
- FMT_MTD.1(PRIVS)

7.2.2.6.3 Device privilege sets (PV.3)

AIX supports the use of privileges for accessing a device. Each device can have two sets of privileges:

- readprivs – Privileges used to control reading from the device.
- writeprivs – Privileges used to control writing to the device.

Each privilege set supports up to 8 privileges. Only one privilege is needed from the privilege set to allow the type of access defined by the privilege set. When no privileges are assigned to the access type, then access is governed by DAC for the access type. By default, AIX does not define any privileges for any devices.

This section maps to the following SFR(s):

- FDP_ACC.1(PSO-AIXC), FDP_ACF.1(PSO-AIXC)
- FDP_ACC.1(PSO-NFS), FDP_ACF.1(PSO-NFS)
- FDP_ACC.1(TSO), FDP_ACF.1(TSO)
- FDP_ACC.1(TCP), FDP_ACF.1(TCP)
- FDP_ACC.1(AUTH), FDP_ACF.1(AUTH)
- FDP_ACC.1(TCB), FDP_ACF.1(TCB)
- FDP_IFC.1(MIC), FDP_IFF.2(MIC)
- FDP_IFC.1(TN), FDP_IFF.2(TN)
- FDP_IFC.2(LS), FDP_IFF.2(LS)
- FMT_MTD.1(PRIVS)

7.2.2.6.4 WPAR privilege set (PV.4)

The WPAR Privilege Set (WPS) defines the maximum privilege set that a System WPAR and Application WPAR can have. Each System WPAR and Application WPAR has its own WPS assigned to it by the Global WPAR administrator. System WPARs and Application WPARs cannot modify their WPS.

This section maps to the following SFR(s):

- FDP_ACC.1(PSO-AIXC), FDP_ACF.1(PSO-AIXC)
- FDP_ACC.1(PSO-NFS), FDP_ACF.1(PSO-NFS)
- FDP_ACC.1(TSO), FDP_ACF.1(TSO)
- FDP_ACC.1(TCP), FDP_ACF.1(TCP)
- FDP_ACC.1(AUTH), FDP_ACF.1(AUTH)
- FDP_ACC.1(TCB), FDP_ACF.1(TCB)
- FDP_IFC.2(VIRT), FDP_IFF.1(VIRT)
- FDP_IFC.1(MIC), FDP_IFF.2(MIC)
- FDP_IFC.1(TN), FDP_IFF.2(TN)
- FDP_IFC.2(LS), FDP_IFF.2(LS)

- FMT_MTD.1(PRIVS)

7.2.2.7 Authorizations (AZ)

Authorizations are the mechanism used to mark certain user accounts as being associated with special administrative roles, such as the information system security officer (ISSO), the system administrator (SA), or the system operator (SO). Roles are simply collections of authorizations. Authorizations can be used to enforce the two-man rule, such as adding new users to the system, where one user can add the account and another can set up the account security information.

7.2.2.7.1 User authorizations (AZ.1)

Authorizations are used to limit access to privileged programs, such as the *mount* command and the *shutdown* command. This allows a distinction to be made between administrators and regular users and to divide administrative duties among different classes of administrators.

Authorizations are also used by trusted programs to provide different levels of functionality for different classes of users. For example, the *chmod* program will allow a user with ISSO role to change the permission bits of any file, but allow a regular user to change the permission bits only for his own files.

Authorizations are enforced even when superuser emulation mode (SEM) is enabled. Thus, some commands cannot be executed by user ID 0 even when SEM is enabled.

Authorizations may be used for the following:

1. to determine if a user (or a process running on the user's behalf) can run a program
2. to determine what privileges a process can have or use while running a program
3. to determine what functionality is available to a user while running a specified program

Cases 1 and 2 can be handled without any modification to the source or binary file. Case 3 requires that the source code of the program be modified to specifically check for authorization and to behave differently based on the authorization check.

There are two categories of authorizations:

- system-defined authorizations – authorizations defined by AIX and hardcoded into the system
- user-defined authorizations – authorizations modifiable by an authorized system administrator

The system-defined authorizations are hardcoded into the system and used by the Global environment and all WPARs, but the user-defined authorizations are located in the */etc/security/authorizations* file. The Global environment and each System WPAR can have its own independent set of user-defined authorizations. The system-defined authorizations and the user-defined authorizations are combined to create the complete set of authorizations for the specific environment.

Section 7.2.2.6.2 "Privileged commands (PV.2)" defines how authorizations play a role in the Privileged Commands mechanism.

An authorization consists of the following components:

- a unique name – dot separated notation string of up to 63 printable characters
- a unique numeric identifier - in the range from 1 to 10000 inclusive for system-defined authorizations and greater than 10000 for user-defined authorizations
- a default description message – textual description of the authorization
- message catalog information for a description message

Users have roles assigned to them and can control which assigned roles are active. Roles are a collection of authorizations. The kernel maintains the set of assigned user roles for a user, the set of active roles of the user, and the authorizations associated with all the defined roles in the environment (i.e., Global environment or WPAR) in tables within the kernel. Thus, the kernel can determine for each user the set of active authorizations. The kernel uses these user authorizations to determine (using the *privcmds* table) if the user can execute a privileged command and the set of privileges the kernel should assign to the process.

Additionally, processes can obtain the invoker's list of active authorizations from the kernel and make access decisions based on the invoker's active authorization set.

Because the authorization set associated with a process is a function of the process's RUID, the authorizations of a process may change dynamically, such as after a *setuid* system call.

This section maps to the following SFRs:

- FDP_ACC.1(AUTH)
- FDP_ACF.1(AUTH)

7.2.2.8 Mandatory access control (MAC) (LAS mode only)

This section applies to LAS mode only.

The TOE provides a MAC policy that enforces access to named objects listed in Table 19 based on sensitivity labels. A sensitivity label is an access control structure that enumerates the MAC access control properties for the object. Sensitivity labels exist for all subjects and named objects on the system. A sensitivity label contains a single classification and a set of zero or more categories. The TOE supports up to 32,001 hierarchical classifications and up to 1,024 categories.

Object Type	Named Object	MAC Operations on Named Object
FSO	device special files – block and character, TCB	Read/Write/Exec
	directory – regular	
	file – regular, system, audit	
	symbolic link	
IPC	message	Read/Write/Exec
	semaphore	
	shared memory	
Miscellaneous	signal vector	Read/Write
	STREAMS message block	
	pipe – unnamed (FIFO)	

Table 19: MAC objects and operations

The MAC policy enforced by the TOE can be described using the concept of sensitivity label dominance. A sensitivity label dominates another if the classification of the first equals or exceeds the classification of the second, and if every category in the second is included in the first. For a subject to read an object, the subject's sensitivity label must dominate the object's sensitivity label. The TOE uses a label encodings file to designate TOE labels and their dominance relationships that are used for MAC enforcement. For a subject to write an object, the object's sensitivity label must be equal to the subject's sensitivity label.

The TOE can be configured with an accreditation range. The accreditation range is defined by a System High and System Low sensitivity label. The System High label must dominate all data processed on the system. All data on the system must dominate the System Low sensitivity label.

Processes can use privileges to override MAC restrictions as defined in FDP_IFF.2(LS).

The import and export of data is implemented through a trusted version of the *backup/restore* utilities. These utilities have been extended to handle labeling. The extensions are transparent to the user and, aside from the labeling extensions, the command functions in the standard fashion. A combination of privilege and authorization mechanisms protects the import/export system.

Import and Export of labeled data describes the system's ability to maintain security attributes when objects are imported and exported to and from the system using predefined security enforcing interfaces. Import and Export of unlabeled data describes the system's ability to disregard security attributes, and the restrictions that are in place to maintain the system's integrity. The TOE enforces mandatory access control when exporting labeled and unlabeled data. Mandatory access control decisions are based on the files sensitivity label and the configured device sensitivity label or label range. The subject exporting the data on behalf of a user must have appropriate privileges, or the user must have appropriate authorizations.

Labeled functionality is included on the TOE. Data can be sent to both hardware devices and to files. Labels are automatically included in backup headers. An unauthorized user cannot override the placing of labels with the data. Only an ISSO authorized administrator can export data without security labels. The TOE provides the tar command that can be used to export data, and does not preserve security information. The TOE requires that a manual change in the device state be performed before using a device to export data without security attributes. This action is auditable.

When writing to disks or tapes, *backup* sensitivity labels (SLs) are included with the data. A user must have the ISSO role as an active authorization to use backup/restore to import or export unlabeled data from tapes or disks. When writing unlabeled data, the data receives the SL of the writing process. When importing unlabeled data, the data is assigned the SL of the importing process. However, if unlabeled data were written with a high-level process, the reading process must have an equal or higher level to read the data.

The evaluated TOE generates printer output using the Postscript Version 2.0 standard and PCL Version 5. The ISSO has the ability to specify the printable label assigned to the sensitivity label of a print job that is sent through the SystemV printing subsystem of AIX. (In LAS mode, printing must be disabled in the evaluated configuration.)

Mounted file systems are the only devices that are available to users for importing/exporting **labeled** data. The TOE places labels on the data being imported or exported via these devices. If an administrator changes the device such that it does not handle labeled data, the system is outside the evaluated configuration.

For remote access via NFS, as well as the other single level file systems CDRFS, UDFS, and PROCFS, a single level policy is implemented: On the client side of NFS, as well as for the other single level file systems, the labels of the mount point are applied to the entire file system (precisely, the max SL and min SL for each file system object located within the single level file system is equal to the max SL and min SL of the mount point).

The following devices are available to users for importing/exporting **unlabeled** data only:

- floppy drive
- raw hard disk drive
- CD-ROM writable
- serial port
- other devices

The TOE does not place labels on data being imported or exported via these devices. If an administrator changes the device such that it handles labeled data, the system is outside the evaluated configuration. Although, administrators can make these devices unavailable, this action does not change the device state from unlabeled to labeled.

This section maps to the following SFRs:

- FDP_ETC.2(LS)
- FDP_IFC.2(LS), FDP_IFF.2(LS)
- FIA_ATD.1(LS)
- FIA_ATD.1(LSX)
- FIA_USB.1(LS)
- FIA_USB.1(LSX)

7.2.2.9 Networking (NET)

7.2.2.9.1 Protected remote access (NET.1)

In both BAS and LAS mode, the TOE supports IPsec for protected remote access connections. IPsec provides integrity and confidentiality of the transported data and is able to authenticate the end points. It supports both IKEv1 and IKEv2 standards and uses Diffie-Hellman key agreement. IPsec performs cryptographic key destruction using zeroization. The TOE uses the CLiC cryptographic library for all IPsec cryptographic functions including random number generation. (The CLiC version is specified in section 1.5.3.)

IPsec supports the following cryptographic algorithms:

- AES-128, AES-192, and AES-256, CBC, GCM
- SHA-256 and SHA-384
- RSA 1024 and 2048 bits
- DSA 1024 bits
- Suite-B-GCM-128, Suite-B-GCM-192, Suite-B-GCM-256, Suite-B-GMAC-128, Suite-B-GMAC-192, and Suite-B-GMAC-256 defined in [RFC4869]

This section maps to the objective [OSPP]_O.CRYPTO.NET and the following SFR(s):

- FCS_CKM.1(SYM)
- FCS_CKM.1(RSA)

- FCS_CKM.1(DSA)
- FCS_CKM.2(NET)
- FCS_CKM.4
- FCS_COP.1(NET) - IPsec with IKE
- FCS_COP.1(CLIC-ENC)
- FCS_COP.1(CLIC-MD)
- FCS_COP.1(CLIC-SGN)
- FCS_RNG.1(CLIC)
- FDP_RIP.2
- FDP_RIP.3

7.2.2.9.2 IP filtering (NET.2)

The TOE supports IP filtering of both IPv4 and IPv6 network packets. Authorized administrators can specify IP filtering rules used for IP filtering. The rules can either permit or deny packet flow. The rules can either always permit/deny the flow or they can permit/deny the flow for a fixed period of time. The rules can be based on IP version (4 or 6), protocols (UDP, ICMP, ICMPV6, TCP, ESP, AH), presumed source addresses, destination addresses, destination ports, and source routings. The rules can be applied to different interface types (e.g., tr0, en0, lo0, pp0). Initially, packet flows are unrestricted until rules are applied.

This section maps to the following SFR(s):

- FDP_ETC.2(VIRT)
- FDP_IFC.2(NI)
- FDP_IFF.1(NI)
- FIA_ATD.1(TU)
- FMT_MSA.3(NI)
- FMT_MTD.1(NI)

7.2.2.10 Trusted Networking (TN) (LAS mode only)

This section applies to LAS mode only.

In LAS mode, the following devices can be configured for both labeled and unlabeled data:

- network

7.2.2.10.1 Network and interface rules (TN.1)

Labeling of data is handled by TN, which reads settings from rules loaded into the kernel from in configuration files. On initial installation, the rules are set to import and export labeled data. The rules can be changed with the *netrule* command. Such changes are not saved on power down. The *tninit* command saves changes between sessions.

The configuration files for TN are */etc/security/rules.host* and */etc/security/rules.int*. These are in binary format. If */etc/security/rules.int* does not exist, the system will create a rules file from the ASCII text file */etc/tn/scripts/iniRules.txt* by running *tninit* during the boot sequence.

The “change in device state” from labeled to unlabeled import and export of data is determined by the TN rules. Therefore, using *netrule* or *tninit* to change the TN rules constitutes the “manual change in device state” required for the evaluation.

The *netrule* and *tninit* commands generate audit events when rules are changed. This action satisfies the Security Target requirement for auditing the “change in device state”.

Exported data, even if labeled, may be read by a non-LAS mode machine. For example, if an administrator uses *backup* to archive data onto a tape, then the tape can be read by an unmodified machine using *restore*, and all labels will be ignored, granting access to all data on the tape. Administrators should be aware that this is not a supported option.

TN provides two sets of networking rules: network interface and host filtering. Both types of networking rules determine what processing occurs on a packet before its transmission or when it is received. These rules are an extension of the MAC policy because 1) they apply sensitivity labels to packets, and 2) they enforce MAC restrictions on packets according to those labels. The TOE enforces MAC between subjects and TN named objects.

Network interface rules enforce packet label processing based on the physical network interface of the host. Host rules enforce packet label processing based on the source and destination IP addresses (with network masking allowed) of the packet, the source and destination ports of the request, and the protocol being used. Both types of rules provide several criteria for determining which packets to drop and which to pass.

Information flow is permitted only if the following sequential ordering relationships between security attributes hold:

1. if the IP address of the packet is equal to the IP address specified in the rule
2. if the IP address of the packet is within the network mask specified for the rule
3. if the direction of the packet flow corresponds to the direction of the rule (IN/OUT)
4. if the protocol of the packet is equal to the protocol specified in the rule
5. if the port is within the range specified in the rule
6. if the network interface of the packet is equal to the network interface specified in the rule
7. if the IPSO labels are within the range defined by the rule, and the rule set to allow IPSO labels
8. if the packet's SL is within the minimum and maximum SL specified for the rule

This section maps to the following SFR(s):

- FDP_ETC.2(LS)
- FDP_IFC.1(TN), FDP_IFF.2(TN)
- FDP_IFC.2(LS), FDP_IFF.2(LS)
- FDP_ITC.1(LS)
- FDP_ITC.2(LS)
- FPT_TDC.1(LS)

7.2.2.10.2 Internet Protocol Security Option (IPSO) (TN.2)

The TOE supports the Revised Internet Protocol Security Option (RIPSO) specified in [RFC1108] and the Common Internet Protocol Security Option (CIPSO) as specified in [FIPS188], which allows the transmission of labeled data over IP networks. It also supports the DoD Basic Security Option (BSO) and Extended Security Option (ESO) specified in [RFC1108]. The TOE uses a standards compliant implementation for IPv4. Since the standards do not address IPv6, the TOE's implementation for IPv6 is specific to the TOE.

In order to preserve the labels for the transmitted data, IP datagrams are extended with IP options that provide for classification of the transmitted data. To translate system-specific sensitivity labels defined in */etc/security/enc/LabelEncodings* into the [RFC1108]-specified RIPSO labels, a translation table has to be created in (for example) */etc/security/rfc1108* and its location to be supplied to *tninit*. CIPSO uses a security tag to indicate the rule used to construct the security data in the IPSO. TN supports the tag types 1, 2 and 5 as specified in [FIPS188].

The *netrule* command (which requires authorizations in order to execute) is used to define TN rules that specify the usage of CIPSO or RIPSO for network connections (see above).

This section maps to the following SFR(s):

- FDP_ETC.2(LS)
- FDP_ITC.1(LS)
- FPT_TDC.1(LS)

7.2.2.11 Mandatory Integrity Control (MIC) (LAS mode only)

This section applies to LAS mode only.

Mandatory integrity control is a system-enforced means of restricting access to and modification of objects based on the integrity of the object and the clearance of the user. While MAC is concerned with the sensitivity of an object, MIC is concerned with its trustworthiness.

The TOE enforces MIC using a system of labels. All named objects have integrity labels (TLs) to identify the integrity level of the object. Processes also have TLs. Process TLs indicate what level of information integrity the processes are allowed to access. The higher the TL, the more trustworthy the object or process. A process must be at least as trustworthy as an object in order to modify it. This means a process must have a TL equal to or exceeding that of the object. Thus TLs can be used to make files accessible for read only. For creating new objects in a file system, a process must be at least as trustworthy as the directory the object is to be created in.

MIC for read access is not enforced in the evaluated configuration.

7.2.2.11.1 MIC labels (MIC.1)

All system objects, such as files, processes, etc. have TLs. TLs are automatically placed on objects at the time of creation. All core dumps are considered objects on the system and are automatically labeled by the system.

Only processes with proper privileges and authorizations are able to change the TL of a file or process.

Unlike MAC, directories only have a single label for MIC. There is a special TL that can be put on a file or process, called NOTL. When NOTL is on an object or process, no MIC checks are performed on it. Only privileged users can set a TL to NOTL, or change a TL if it is currently NOTL.

This section maps to the following SFR(s):

- FDP_IFC.1(MIC)
- FDP_IFF.2(MIC)
- FIA_USB.1(LSX)

7.2.2.12 Object reuse (OR)

Object Reuse is the mechanism that protects against scavenging, or being able to read information that is left over from a previous subject's actions. The following general techniques are applied to meet this requirement in the AIX distributed system:

- explicit initialization of resources on initial allocation or creation
- explicit clearing of resources on release or deallocation
- management of storage for resources that grow dynamically
- administrator-initiated wiping of hard disk drives

Explicit initialization is appropriate for most TSF-managed abstractions, where the resource is implemented by some TSF internal data structure whose contents are not visible outside the TSF: queues, datagrams, pipes, and devices. These resources are completely initialized when created, and have no information contents remaining.

Explicit clearing is used in AIX only for directory entries, because they are accessible in two ways: through TSF interfaces both for managing directories and for reading files. Because this exposes the internal structure of the resource, it must be explicitly cleared on release to prevent the internal state from remaining visible.

Storage management is used in conjunction with explicit initialization for object reuse on files, and processes. This technique keeps track of how storage is used, and whether it can safely be made available to a subject.

Hard disk wiping provides means to an administrator to overwrite information on hard disks with bit patterns in order to render previously stored information on the disks unrecoverable. Rather than being a system property, this is a function that can be invoked by the administrator.

The following sections describe in detail how object reuse is handled for the different types of objects and data areas.

7.2.2.12.1 Object reuse: File system objects (OR.1)

All file system objects (FSOs) available to general users are accessed by a common mechanism for allocating disk storage and a common mechanism for paging data to and from disk. This includes the Journaled File System (JFS2) and Network File System (which exists physically as a JFS2 volume on a server host). This includes both normal and large JFS2 file systems.

Object reuse is irrelevant for the CD-ROM File System (CDRFS) and the Universal Data Standard File System (UDFS) because they are read-only file systems, making it impossible for a user to read residual data left by a previous user. File systems on other media (tapes, diskettes) are irrelevant because of warnings in [SecGuide] not to mount file systems on these devices.

For this analysis, the term FSO refers not only to named file system objects (files, directories, device special files, named pipes, and UNIX domain sockets) but also to unnamed abstractions that use file system storage (symbolic links and unnamed pipes). All of these, except unnamed pipes, device special files, and UNIX domain sockets, have a directory entry that contains the pathname and an inode that controls access rights and points to the disk blocks used by the FSO.

In general, file system objects are created with no contents, directories and symbolic links are exceptions, and their contents are fully specified at creation time.

This section and its subsections map to the following SFR(s):

- FDP_RIP.2

- FDP_RIP.3

7.2.2.12.1.1 Object reuse: Files

Storage for files is allocated automatically in pages as a file grows. These pages are cleared before they become accessible, within the file. However, when a file is deleted the space holding the data from the file, both in memory and on disk, is not cleared. This data will persist until the space is assigned to another file, when it will be cleared. These internal fragments of deleted files are protected by the kernel to prevent accessing of deleted data.

If data is read before it is written, it will read only as zeroes. Reads terminate when the end-of-file (EOF) is detected. It is possible to seek past the EOF, but any reads will return zeroes. File writes may cause the file to grow, thus overwriting any residual data and moving the EOF. If the file pointer is advanced past the EOF and then written, this leaves a hole in the file that will subsequently be read as zeroes.

7.2.2.12.1.2 Object reuse: Directories and directory entries

In part, object reuse for directories is handled as for ordinary files: pages allocated are always cleared before being incorporated into the directory. When a directory is first created, it is explicitly initialized to have the entries “.” and “..”, but the remainder of the directory's storage is cleared.

Individual directory entries are manipulated as distinct resources, such as when referencing file system objects, and as part of the directory, such as when reading the entire directory itself. When a directory entry is removed or renamed the space occupied by that directory entry is either combined with the previous entry as free space or else the inode number of the entry is set to zero when the entry occurs on a 512 byte boundary.

When a directory entry does not occur on a 512-byte boundary, the size of the preceding directory entry is incremented by the size of the directory entry which has been removed. The space in a directory entry in excess of that which is needed to store the necessary information may be allocated when a directory entry is to be created. The fields of the directory entry remain unchanged.

When a directory entry occurs on a 512-byte boundary, the inode number is set to zero to indicate that this entry is now available for re-use. All other fields of the directory entry remain unchanged.

The directory entry is no longer visible to interfaces which perform file name operations and may only be seen when the entire directory is examined and the process has read access to the directory.

7.2.2.12.1.3 Object reuse: Symbolic links

Symbolic links have their contents (the link pathname) fully specified at creation time, and the readlink operation returns only the string specified at creation time, not the entire contents of the block it occupies.

7.2.2.12.1.4 Object reuse: Device special files

All device special files are initialized to a known state on first open and never grow. Device special files refer to actual hardware or else to virtualized objects. There are no file system blocks, unless the device references a file system (in which case the mechanism for object reuse of file system objects apply). Nor is there memory, unless the device is associated with memory (in which case the object reuse mechanisms for memory objects apply).

7.2.2.12.1.5 Object reuse: Named pipes (FIFOs)

FIFOs (First In First Out) are created empty. Buffers are allocated to contain data written to a pipe, but the read and write pointers are managed to ensure that only data that was written to the pipe can ever be read from it.

7.2.2.12.1.6 Object reuse: Unnamed pipes

Unnamed pipes are created empty. Buffers are allocated to contain data written to a pipe, but the read and write pointers are managed to ensure that only data that was written to the pipe can ever be read from it.

7.2.2.12.1.7 Object reuse: Socket special file (UNIX domain)

UNIX domain sockets have no contents; they are fully initialized at creation time.

7.2.2.12.2 Object reuse: IPC objects (OR.2)

AIX shared memory, message queues, and semaphores are initialized to all zeroes at creation. These objects are of a finite size (shared memory segment is from one byte to 256 MBytes, semaphore is one bit), and so there is no way to grow the object beyond its initial size.

No processing is performed when the objects are accessed or when the objects are released back to the pool.

This section maps to the following SFR(s):

- FDP_RIP.2
- FDP_RIP.3

7.2.2.12.3 Object reuse: Queuing system objects (OR.3)

7.2.2.12.3.1 Object reuse: Batch queue entries

The *cron* daemon and *at*daemon jobs are defined in batch files, which are subject to the object reuse protections specified for files as described previously.

This section maps to the following SFR(s):

- FDP_RIP.2
- FDP_RIP.3

7.2.2.12.4 Object reuse: Miscellaneous objects (OR.4)

7.2.2.12.4.1 Object reuse: Process

A new process's context is completely initialized from the process's parent when the fork system call is issued. All program visible aspects of the process context are fully initialized. All kernel data structures associated with the new process are copied from the parent process, then modified to describe the new process, and are fully initialized.

The AIX kernel zeroes each memory page before allocating it to a process. This pertains to memory in the program's data segment and memory in shared memory segments. When a process requests more memory, the memory is explicitly cleared before the process can gain access to it.

When the kernel performs a context switch from one thread to another, it saves the previous thread's General Purpose Registers (GPRs) and restores the new thread's GPRs, completely overwriting any residual data left in the previous thread's registers. Floating Point Registers (FPRs) are saved only if a process has used them. The act of accessing an FPR causes the kernel to subsequently save and restore all the FPRs for the process, thus overwriting any residual data in those registers.

Processes are created with all attributes taken from the parent. The process inherits its memory (text and data segments), registers, and file descriptors from its parent. When a process execs a new program, the text segment is replaced entirely.

This section maps to the following SFR(s):

- FDP_RIP.2
- FDP_RIP.3

7.2.2.12.5 Object reuse: Hard disk drives (OR.5)

For SCSI disks that do not participate as “pdisks” in RAID arrays, the diagnostic subsystem (see section 7.2.2.14.9 "Diagnosis (TP.9)") offers a “Hard File Erase Disk” functionality to administrators of the TOE. SCSI disks which are part of a pdisk must be detached from the pdisk for erasure.

This option can be used to overwrite (remove) all data stored in currently user-accessible blocks of the disk. The Erase Disk option writes one or more user-specifiable patterns to the disk.

The administrator can specify the number (0-3) of patterns to be written to the hard disk. The patterns are written serially; that is, the first pattern is written to all blocks. Then the next pattern is written to all blocks, overlaying the previous pattern. Also, a random pattern can be written.

Please note that there are two abstraction layers in the underlying environment involved that restrict the TOE to the deletion of user-accessible blocks on those hard disk drives only: Firmware of SCSI hard disk drives and firmware of SCSI disk controllers may remap “bad blocks” containing user or TSF data to healthy blocks on the physical hard disk drive and maintain a pool of unallocated blocks for this purpose. The TOE is not able to (and does not claim to) overwrite such blocks, since it is using the generic SCSI interfaces to access the hard disk drive. Since the hard disk drive stays within the TSC it is ensured that users of the TOE, accessing the drive via the TOE-provided interfaces, won't be able to recover any residual information on it.

This section maps to the following SFR(s):

- FDP_RIP.4
- FMT_SMF.1(BASE)

7.2.2.13 Security Management (SM)

This section describes the functions for the management of security attributes that exist within AIX.

7.2.2.13.1 Roles (SM.1)

AIX provides a role-based access control (RBAC) mechanism. See section 7.2.2.5 "Role-based access (RA)" for more details.

In LAS mode, the administrator is not root, but one of the ISSO, SA and SO roles. Root itself will not be used as a user ID that a user can directly log in to or for administration.

In BAS mode, the root user can be allowed to log in (root enabled mode) or root login can be disabled (root disabled mode). Note that **root disabled mode is not supported in the evaluated configuration in BAS mode**.

Users that have the appropriate authorizations associated with their account may run administrative programs associated with those authorizations/roles.

This section maps to the following SFR(s):

- FMT_MTD.1(AM-AP)
- FMT_MTD.1(AM-MR)
- FMT_MTD.1(AM-MD)
- FMT_MTD.1(AM-MA)
- FMT_MTD.1(RBAC)
- FMT_SMF.1(BASE)
- FMT_SMR.2

7.2.2.13.1.1 Normal users

In AIX, normal users cannot perform actions that require administrator privileges. They can only execute those setuid root programs and privileged programs they have access to (either via DAC or authorizations or the *privcmds* table). In the evaluated configuration this is restricted to those programs they need, like the passwd program that allows a user to change his/her own password.

7.2.2.13.1.2 n-man rule

The RBAC mechanism supports execution of commands that require multiple authentications. The roles required to execute a command can be configured by the authorized administrator in the privcmds file.

7.2.2.13.2 Audit configuration and management (SM.2)

Audit control consists of the files in Table 20 that are used to maintain the configuration of the audit subsystem and a description of the *audit* command and its associated parameters.

Audit Control File	Description
/etc/security/audit/config	Defines whether bin mode auditing is enabled, the names of the files used to store audit data and the names of the available classes. Also defines the audit classes, i.e., for each audit class the audit events belonging to the class are defined.
/etc/security/audit/events	Defines audit events to be used on the system. An event needs to be defined in this file to be formatted correctly.
/etc/security/audit/objects	Contains a list of the objects whose access will be audited.
/etc/security/audit/bincmds	Contains the post-processing command or commands for bin mode auditing.
/etc/security/user	Contains a record for each user which specifies which classes will apply to the user account.

Table 20: Audit control files

There are two different types of audit event selection: per-user and per-object. Per-user auditing allows the administrator to specify specific classes of audit events that will be recorded for that user. Each process stores a copy of the audit classes that apply to that user as part of the process table. An audit class is a subset of the total number of audit events available and is defined in the file */etc/security/audit/config*.

Per-object auditing allows the administrator to specify file system objects that will be audited. This is defined in the file */etc/security/audit/objects*. There, for individual objects, the audit event for the access modes that one wants to be audited is defined.

These objects can be audited based on accesses of a specified mode (read/write/execute) and record the result of the access attempt (success/failure).

The *audit* command is used to start and stop the auditing subsystem, to temporarily switch the auditing subsystem on or off, and to query the audit subsystem for the current audit parameters.

The *audit* command is started from the host's rc initialization script, as stated in [SecGuide].

The on and off parameters of the *audit* command enable and disable audit, without modifying the current configuration that is stored in the kernel. The on parameter can have an additional parameter, panic, which causes the system to shut down if bin data collection is enabled and records cannot be written to one of the bin files. The bin mode panic option can also be specified in */etc/security/audit/config*.

When the *audit* command is issued with the **shutdown** parameter, the collection of audit records is halted, and all audit configuration information is flushed from the kernel's tables. All audit records are flushed from the kernel's buffers and processed. The collection of audit data is halted until the next audit start command is entered.

When the *audit* command is issued with the start parameter, the following events occur:

- the */etc/security/audit/config* file is read
- the */etc/security/audit/objects* files is read and the objects that will be audited based on access are written into kernel tables
- the audit class definitions are written into kernel tables from */etc/security/audit/config*
- the *auditbin* daemon is started, depending on the options in */etc/security/audit/config*
- auditing is enabled for users specified in the user's stanza of the */etc/security/audit/config* file
- auditing is turned on, with panic mode enabled or turned off, depending on what mode is specified in */etc/security/audit/config*

To access the audit functions, the appropriate PV_AU privileges are required. The AUDIT authorization is needed for reading audit records.

The events that are audited can be selected on a per user basis, per event basis and per object basis using the configuration files described above.

A description of the structure of those files and the syntax of the entries can be found in *AIX 7.1 Files Reference* document.

A System WPAR contains its own audit trial which functions independently from the Global environment's audit subsystem. A Global environment determines what it wants to audit of the System WPAR and those events are added to the Global environment's audit trial regardless of the state of the auditing subsystem in the System WPAR. Similarly, a System WPAR configures and controls its own audit trail. An Application WPAR does not have a separate auditing subsystem, so it is under the audit control of the Global environment.

This section maps to the following SFR(s):

- FAU_SEL.1(BASE)
- FAU_SEL.1(LS)
- FMT_MTD.1(AE)
- FMT_MTD.1(AS)
- FMT_MTD.1(AT)
- FMT_MTD.1(AF)
- FMT_MTD.1(RBAC)
- FMT_SMF.1(BASE)

7.2.2.13.3 Access control configuration and management (SM.3)

Discretionary access control to objects is defined by the permission bits and by the Access Control Lists (for those objects that have access control lists associated with them) or by NFSv4 ACLs. Default access permission bits are defined in the system configuration files that define the value of the access control bits for objects being created without explicit definition of the permission bits. The system administrator can define and modify those default values.

Permissions can be changed by the object owner and the system administrator. When an object is created the creator is the object owner. Object ownership can be transferred except for TCP ports, where the owner always remains the system administrator. In the case of IPC objects, the creator will always have the same right as the owner, even when the ownership has been transferred.

LAS Mode Only: For MAC, TN, and MIC, sensitivity and integrity labels are assigned to all objects and users, whereas users are assigned a specific range of levels (clearance) within which they can operate.

IP Filtering rules are managed by authorized administrators. Authorized administrators can create, modify, delete, activate, de-active, and query rules.

TCB files are identified by a specific file security flag that can be set by authorized administrators. Authorizations can be granted to users by authorized administrators.

NFSv4 ACLs provide a mechanism which allows the object owner to give others the ability to modify the entries within the ACL. Directory NFSv4 ACLs can include entries that are inherited by child objects.

EFS provides a mechanism (including the commands to enable/disable and manage it) to encrypt/decrypt files. Encryption prevents the disclosure of the data in the files except by those who have the proper key to decrypt the data.

File Integrity Verification (FIV) provides a mechanism for the system administrator to determine if system critical objects have the appropriate security attributes (user, group, permission bits, etc.) set on the objects.

System WPARs are created, managed, and deleted by Global environment administrators within the Global environment. The Global environment administrator can control the file systems available to a WPAR, the mode of the file systems available to a WPAR, the devices available to a WPAR, and the privileges available to the processes of a WPAR.

This section maps to the following SFR(s):

- FMT_MSA.1(AUTH), FMT_MSA.3(AUTH)
- FMT_MSA.1(RBAC-AUTH), FMT_MSA.3(RBAC)

- FMT_MSA.1(TCB), FMT_MSA.3(TCB)
- FMT_MSA.1(PSO-AIXC), FMT_MSA.3(PSO-AIXC)
- FMT_MSA.1(PSO-NFS), FMT_MSA.3(PSO-NFS)
- FMT_MSA.1(TSO), FMT_MSA.3(TSO)
- FMT_MSA.1(TCP), FMT_MSA.3(TCP)
- FMT_MTD.1(NI), FMT_MSA.3(NI)
- FMT_MSA.1(VIRT-CACP), FMT_MSA.3(VIRT-CACP)
- FMT_MSA.1(VIRT-CIFCP), FMT_MSA.3(VIRT-CIFCP)
- FMT_REV.1(USR)
- FMT_SMF.1(BASE)
- FMT_MSA.1(LS), FMT_MSA.3(LS)
- FMT_MSA.1(MIC), FMT_MSA.3(MIC)
- FMT_MSA.1(TN), FMT_MSA.3(TN)

7.2.2.13.4 Management of user, group, and authentication data (SM.4)

Each System WPAR can create and manage its own set of users, groups, and authentication data independently from the Global environment and from other System WPARs.

This section and its subsections map to the following SFR(s):

- FIA_ATD.1(HU)
- FIA_SOS.1(BASE)
- FIA_USB.2
- FMT_MSA.1(RBAC-DFLT)
- FMT_MSA.1(RBAC-USR)
- FMT_MSA.2(RBAC)
- FMT_MTD.1(IAT)
- FMT_MTD.1(IAF)
- FMT_MTD.1(IAU)
- FMT_MTD.1(RBAC)
- FMT_MTD.3(RBAC)
- FMT_REV.1(USR)
- FMT_SMF.1(BASE)
- FMT_SMR.2

7.2.2.13.4.1 Creating new users

An administrator (role SA) can create a new user and can assign a unique user ID to this user. The initial password and other security attributes have to be defined by the ISSO using various utilities (e.g., the *passwd* command). The new user will be disabled until the initial password is set.

Attributes that can be set for each user are among others (a complete list can be found in the description of the chuser command and the description of the content of the file */etc/security/user*):

- Lock attribute (i.e., temporarily locking a user account)
- Administrative status of the user

- List of audit classes for the user
- List of groups the user belongs to
- Home directory for this user
- Number of consecutive unsuccessful login attempts allowed before the user account is locked
- Password parameter including the maximum and minimum age of a password, minimum length, difference to the old password, etc.

Those attributes are stored in the following files:

- */etc/group*
- */etc/passwd*
- */etc/security/passwd*
- */etc/security/user*
- */etc/security/audit/config*

7.2.2.13.4.2 Modification of user attributes

User attributes can be modified by the system administrator (ISSO). Modifications of user attributes require the modification of the administration database that contains the user attributes (mainly */etc/security/user*).

7.2.2.13.4.3 Management of authentication data

The system administrator (ISSO) has the capability to define rules and restrictions for passwords used to authenticate users. Table 21 contains the AIX password parameters.

AIX Password Parameter	Description
minage	Minimum number of weeks that must pass before a password can be changed.
maxage	Maximum number of weeks that can pass before a password must be changed.
maxexpired	Maximum number of weeks beyond maxage that a password can be changed before administrative action is required to change the password. (Root is exempt.)
minalpha	Minimum number of alphabetic characters the new password must contain.
minother	Minimum number of non-alphabetic characters the new password must contain. (Other characters are any ASCII printable characters that are non-alphabetic and are not national language code points).
minlen	Minimum number of characters the new password must contain.
maxrepeats	Maximum number of times a character can be used in the new password.
mindiff	Minimum number of characters in the new password that must be different from the characters in the old password.
histexpire	Number of weeks that a user is unable to reuse a password.
histsize	Number of previous passwords that cannot be reused.

AIX Password Parameter	Description
dictionlist	List of dictionary files checked when a password is changed. Dictionary files contain passwords that are not allowable.

Table 21: AIX password parameters

Users are also allowed to change their own password using the *passwd* command. The password restrictions defined by the system administrator are enforced by the *passwd* command.

7.2.2.13.5 Time management (SM.5)

AIX provides the standard UNIX functions to manage the system clock. The time can be set or modified by an administrator (ISSO). Modifications to the system time are audited (if configured) allowing a system administrator to extract the differences between the “old” and “new” value of the system clock. The value of the system clock cannot be manipulated by normal users.

This section maps to the following SFR(s):

- FPT_STM.1

7.2.2.14 TSF protection (TP)

While in operation, the kernel software and data are protected by the hardware memory protection mechanisms described in the high level design and the hardware reference manuals of AIX. The memory and process management components of the kernel ensure a user process cannot access kernel storage or storage belonging to other processes.

Non-kernel TSF software and data are protected by DAC, MAC (LAS mode only), MIC (LAS mode only), and TCB controls and process isolation mechanisms. In general, files and directories containing internal TSF data (e.g., audit files, batch job queues) are also protected from reading by access control permissions.

The TSF and the hardware and firmware components are required to be physically protected from unauthorized access. The system kernel mediates all access to the hardware mechanisms themselves, other than program visible CPU instruction functions.

The boot image for each host in the distributed system is adequately protected.

7.2.2.14.1 TSF invocation guarantees (TP.1)

All system protected resources are managed by the TSF. Because all TSF data structures are protected, these resources can be directly manipulated only by the TSF, through defined TSF interfaces. This satisfies the condition that the TSF must be “always invoked” to manipulate protected resources.

Resources managed by the kernel software can only be manipulated while running in kernel mode.

Processes run in user mode and can call functions of the kernel only as the result of an exception or interrupt. The hardware and the kernel software handling these events and ensure that the kernel is entered only at predetermined locations, and within predetermined parameters. All kernel managed resources are protected such that only the kernel software is able to manipulate them.

Trusted processes implement resources managed outside the kernel. The trusted processes and the data defining the resources are protected as described above depending on the type of interface. For directly invoked trusted processes the program invocation mechanism ensures that the trusted process always starts in a protected environment at a predetermined point. Other trusted process interfaces are started during system initialization and use well defined protocol or file system mechanisms to receive requests.

Some system calls or parameters of system calls are reserved for trusted processes. When called, the kernel checks that the calling process runs with the appropriate privileges.

The TOE contains a Stack Execution Disable (SED) facility. When enabled, SED ensures that code residing on the stack of selected processes cannot be executed by the processes. This facility is configured by an administrator with the *sedmgr* command. If a process is configured to deny execution of code on its stack and the process attempts to execute code on its stack, the system will generate an exception and terminate the process. This helps prevent buffer overflow attacks by not allowing attackers to execute arbitrary code on the stack of an executable.

The TOE implements the TSP through a reference monitor. The kernel reference monitor (KRM) is a single C-language function that is called any time the security policy may need to be enforced. The KRM accepts as arguments all of the security attributes of the subject and object associated with the security check, along with an indication of what check or checks and auditing need to be performed. The KRM itself consists of many code modules that are called to handle the appropriate check or checks that are required. The KRM is used to enforce the security policy in the following instances:

- Systems calls
- File system creation, deletion, and access events
- Interprocess communication, including signals
- Network packet checks

The KRM handles all of the following kernel security mediation events:

- Discretionary access control (DAC) checks
- Mandatory access control (MAC) checks
- Trusted Network (TN) rules
- Mandatory integrity control (MIC) checks
- Trusted computing base (TCB) checks
- Trusted Execution (TE) checks
- System security flag (SSF) checks
- File security flag (FSF) checks
- Privilege (PV) checks
- Authorization (AZ) checks
- Auditing (AUD) for all of the above

The KRM accepts as arguments the following:

- subject security attributes
- action to be performed by the reference monitor
- security attributes and object type for up to 4 objects
- flag indicating if auditing should be done
- pointer to the system-wide security settings

Although the KRM can check MAC, MIC, DAC, TCB, FSF, SSF, PV, AZ, and AUD components, not all actions require all checks. When multiple checks are required, they are performed in the following order:

- MAC (with SSF, FSF, PV and AUD as needed)
- MIC (with SSF, FSF, PV and AUD as needed)
- FSF (with PV and AUD as needed)
- TCB (with SSF, PV and AUD as needed)
- DAC (with PV and AUD as needed)
- PV (with SSF, FSF, and AUD as needed)
- AZ (with SSF, FSF, PV, and AUD as needed)

System security flags (SSFs), also known as kernel security flags, support the enforcement of the TSP.

BAS Mode Only: The evaluated configuration mandates the settings in Table 22 for SSFs in order to ensure that the TSP be enforced accurately:

BAS Mode System Security Flag	Operational Mode
Root (ROOT)	ENABLED

Table 22: System security flags (SSFs) (BAS mode only)

LAS Mode Only: The evaluated configuration mandates the settings in Table 23 for SSFs in order to ensure that the TSP be enforced accurately.

LAS Mode System Security Flag	Operational Mode
Sl_enforced (MAC)	ENABLED
trustedlib_enabled (TLIB)	ENABLED
tl_read_enforced (MIC)	DISABLED
tl_write_enforced (MIC)	ENABLED
tnet_enabled (TN)	ENABLED
Root (ROOT)	DISABLED

Table 23: System security flags (SSFs) (LAS mode only)

This section maps to the following SFR(s):

- FPT_FLS.1(SED)
- FRU_FLT.2

7.2.2.14.2 Kernel (TP.2)

The AIX software consists of a privileged kernel and a variety of non-kernel components (trusted processes). The kernel operates on behalf of all processes (subjects).

The kernel runs in the CPU's privileged mode and has access to all system memory. All kernel software, including kernel extensions and kernel processes (*kprocs*), execute with kernel privileges but only defined subsystems within the kernel are part of the TSF. The kernel is entered by some event that causes a context switch such as a system call, I/O interrupt, or a program exception condition.

Upon entry the kernel determines the function to be performed, performs it, and, when finished, performs another context switch to return to user processing (eventually on behalf of a different subject).

The kernel is shared by all processes, and manages system wide shared resources. It presents the primary programming interface for AIX in the form of system calls.

Because the kernel is shared among all processes, any process running “in the kernel” (that is, running in privileged hardware state as the result of a context switch) is able to directly reference the data structures that implement shared resources.

The major components of the kernel are memory management, process management, the file system, the low-level I/O system, WPAR management, and the kernel extensions like implementing for example network protocols (IP, TCP, UDP, and NFS).

This section maps to the following SFR(s):

- FDP_IFC.2(VIRT)
- FDP_IFF.1(VIRT)

7.2.2.14.3 Kernel extensions (TP.3)

Kernel extensions are dynamically loaded code modules that add function to the kernel. They include device drivers, virtual file systems (e.g., CDRFS, NFS), inter-process communication methods (e.g., named pipes), networking protocols, and other supporting services. Kernel extensions can be loaded only at system boot in the evaluated configuration.

Kernel extensions run with kernel privilege, similarly to kernel processes (*kprocs*). However, extensions differ from *kprocs* in that the kernel does not schedule them. Instead, kernel extensions are invoked from user processes by system calls, or internal calls within the kernel, or started to handle external events such as interrupts.

Kernel extensions run entirely within the kernel protection domain. An extension may export system calls in addition to those exported by the base AIX kernel. User-domain code can only access these extensions through the exported system calls, or indirectly via the system calls exported by the base kernel.

Device drivers are kernel extensions that manage specific peripheral devices used by the operating system. Device drivers shield the operating system from device-specific details and provide a common I/O model for user programs to access the associated devices. For example, a user process calls read to read data, write to write data, and ioctl to perform I/O control functions.

7.2.2.14.4 Trusted processes (TP.4)

Trusted processes in AIX are processes running in user mode but with privileges. Some high-level TSF functions are performed by trusted processes particularly those providing distributed services.

A trusted process is distinguished from other user processes by the ability to affect the security policy. Some trusted processes implement security policies directly (e.g., identification and authentication) but many are trusted simply because they operate in an environment that confers the ability to access TSF data (e.g., programs run by administrators or during system initialization).

Trusted processes have all the kernel interfaces for which they have the appropriate privilege available for their use, but are limited to kernel-provided mechanisms for communication and data sharing, such as files for data storage and pipes, sockets and signals for communication.

The major functions implemented with trusted processes include user login, identification and authentication, batch processing, audit data management and reduction, some network operations, system initialization, and system administration.

The kernel will check for each system call that requires privileges if the process that issued the call has those privileges. If not, the kernel will refuse to perform the system call. The kernel will also, for each access to an object protected by any of DAC, MAC (LAS mode only), and MIC (LAS mode only) mechanisms, check if the process has the required access rights for the attempted type of access. Note that commands listed in the *privcmds* database can override the DAC rules if the user as the proper roles.

Any program executed with DAC override privileges has the ability to perform the actions of a trusted process. It is therefore important that a site operating AIX strictly controls those programs and prohibits that those programs are modified and prohibits that programs from untrusted sources are executed with root privileges.

Trusted processes are not part of the kernel and (except for those processes that perform system initialization and identification and authentication) not part of the TSF itself.

Trusted processes provide a contribution to security management and identification and authentication.

Note: Trusted processes may use system management commands or system calls as mentioned in the section on supporting functions that are not part of the TSF. But in any case the kernel will verify that the process has the right to perform the system call with the parameter specified by the caller and has the right to access all files with the intended access mode.

This section maps to the following SFR(s):

- FDP_ACC.1(TCB)
- FDP_ACF.1(TCB)

7.2.2.14.5 TSF databases (TP.5)

Table 24 identifies the primary TSF databases used in AIX and their purpose. These are listed both as individual files (by pathname) or collections of files.

With the exception of databases listed with the User attribute (which indicates that a user can read, but not write, the file), all of these databases shall only be accessible to administrators. None of these databases shall be modifiable by a user other than a system administrator with the appropriate authorization.

Those databases are part of the file system and therefore the file system protection mechanisms of the TOE have to be used to protect those databases from unauthorized access. It is the task of the persons responsible for setting up and administrating the system to ensure that the access control features of the TOE are used throughout the lifetime of the system to protect those databases.

For LDAP-based I&A, just the user and group information is stored in LDAP. Therefore, LDAP replaces the following files:

- */etc/group*
- */etc/passwd*
- */etc/security/lastlog*
- */etc/security/passwd*
- */etc/security/user*
- */etc/security/user.roles*

When the NFSv4 client and server map user/group string names to UIDs/GIDs, they use the local OS authentication mechanism to make the mapping; thus, they will use the file-based I&A mechanism.

Administrative Database	**Purpose**
/etc/ftpusers	Limits access to FTP.
/etc/group	Stores group names, supplemental GIDs, and group members for all system groups.
/etc/hosts	Contains hostnames and their address for hosts in the network. This file is used to resolve a hostname into an Internet address in the absence of a domain name server.
/etc/inetd.conf	Configures start of network daemons.
/etc/inittab	Controls the system startup by running the appropriate command scripts and SRC invocations
/etc/krb5/krb5.conf	Specifies the default NAS (Kerberos) client configuration data.
/etc/nfs/security_default	Specifies the default NFSv4 client authentication data.
/etc/passwd	Stores user names, UIDs, primary GID, home directories for all system users.
/etc/security/acl	Specification of TCP port, host (or subnet), and user/group at that host or subnet allowed access to the port.
/etc/security/audit/bincmds	Specifies the pipeline of commands to be performed by the *auditbin* daemon.
/etc/security/audit/config	Specifies who and what is going to be audited, where the bin audit data will reside, and how auditing will be performed.
/etc/security/audit/events	Defines all of the audit events that are recognized by the system and the form of their tail data.
/etc/security/audit/objects	Specifies file system objects whose access is to be audited along with for what access modes it will be done.

Administrative Database	Purpose
/etc/security/authorizations	Contains the list of valid user-defined authorizations.
/etc/security/enc/LabelEncodings	Label mappings
/etc/security/lastlog	Stores time/date of last successful and unsuccessful login attempts for each user. Stores the number of unsuccessful login attempts since the last successful one.
/etc/security/ldap/ldap.cfg	Defines configuration attributes enforced by the LDAP client when LDAP-based I&A is used.
/etc/security/login.cfg	Defines attributes enforced when logging in or changing passwords.
/etc/security/mkuser.default	Defaults for user account creation.
/etc/security/passwd	Defines user passwords in one-way encrypted form, plus additional characteristics including previous passwords, password quality parameters.
/etc/security/portlog	Records ports locked as a result of login failures
/etc/security/priv	Defines the privileged ports as part of the access control on TCP ports
/etc/security/privcmds	Contains the security attributes for privileged commands.
/etc/security/privdevs	Contains the security attributes for privileged devices.
/etc/security/privfiles	Contains the security attributes for privileged files.
/etc/security/roles	Contains the RBAC role definitions for user-defined roles.
/etc/security/rules.host and /etc/security/rules.int	Configuration files for network port protection and labels. (LAS mode only)
/etc/security/services	Specification of service names to be used by DACinet (in the style of /etc/services). Ports listed here are exempt from DACinet ACL checks.
/etc/security/tsd/tsd.dat	Integrity checking database (Trusted Signature Database)
/etc/security/user	Defines supplementary data about users, including audit status, required password characteristics, access to su command.
/etc/security/user.roles	Contains the mapping of users to roles.
/etc/wpar/devexports	Contains the list of exported devices to System WPARs.

Administrative Database	Purpose
/usr/ldap/clientkey.kdb	Contains the LDAP client's Kerberos key data. (The actual pathname of this file is specified by the administrator.)
/var/efs/*	Contains the key storage for EFS.
/var/krb5/security/creds/krb5cc_<pid>	Specifies the default NAS (Kerberos) client ticket file.
ODM attributes: mls_config and mls_operation	Specification of the system security flags (SSFs) for configuration and operational mode.
ODM attributes: ipsec_filter	Specifies the IP Filter rules used by the system.

Table 24: Administrative databases

These tables are not functions, but they are part of the management of the TSF.

The databases in Table 25 are converted into binary format and loaded into the kernel.

Kernel Database	Description
/etc/security/authorizations	KAT (Kernel Authorization Table)
/etc/security/privcmds	KCT (Kernel Privileged Command Table)
/etc/security/devices	KDT (Kernel Privileged Device Table)
/etc/security/roles	KRT (Kernel Role Table)

Table 25: Kernel databases

This section maps to the following SFR(s):

- FMT_MSA.2(RBAC)
- FMT_MTD.1(AM-AP)
- FMT_MTD.1(AM-MR)
- FMT_MTD.1(AM-MD)
- FMT_MTD.1(AM-MA)
- FMT_MTD.1(IV-ACT)
- FMT_MTD.1(IV-TSF)
- FMT_MTD.1(IV-USR)

7.2.2.14.6 File Integrity Verification, Trusted Execution, & integrity checks (TP.6)

AIX provides a File Integrity Verification (FIV) mechanism that allows an administrator to check the integrity of the trusted files on the system. The database, called the Trusted Signature Database (TSD), is created during the build process of AIX. For each object listed in TSD, TSD contains the object owner, group, permission bits, hash value, and other security related information. The system administrator can use the *trustchk* command to verify the integrity of the objects listed in TSD. An administrator can also add, delete, and modify entries in TSD. The database is located at:

- */etc/security/tsd/tsd.dat*

The *trustchk* command verifies the attributes of the trusted file system objects by comparing them to the stored values in TSD. If the attributes of the object do not match what is stored in TSD, the command reports the error.

The AIX build process uses the SHA-256 message digest when creating a hash value (message digest) for each object. It then signs the hash values using a private key from a signing certificate that's unique to that version of AIX. These signatures are stored in TSD with each object. The public key is provided with the AIX installation image so that the signatures can be verified on the installed system.

Trusted Execution (TE) uses TSD to check the integrity of executables and data. The mechanism can be used to monitor any file on the system. TE also supports policy configuration allowing administrators to tailor TE to their environment. In the evaluated configuration, the TE policy must contain the following configuration:

- TE=ON
- STOP_ON_CHKFAIL=ON

The TE configuration value enables/disables Trusted Execution. When the STOP_ON_CHKFAIL configuration value is set to ON, TE stops the loading of files whose integrity check fails.

If the kernel detects a modified object, it can deny access (STOP_ON_CHKFAIL) to the object which will cause the application attempting to execute a program or attempting to access the file to generate an error whereby informing the user of the access denial. In addition, TE will generate an audit record of the failure.

The CLiC module is used to implement the cryptographic functions of FIV, TE, and TSD. (The CLiC version is specified in section 1.5.3.) TSD supports the following message digests:

- SHA-1
- SHA-256 (default)
- SHA-512

This section maps to the following SFR(s):

- FCS_COP.1(CLIC-MD)
- FCS_COP.1(CLIC-SGN)
- FDP_SDI.2(IV)
- FPT_TIM.1(IV)

7.2.2.14.7 File security flags (TP.7) (LAS mode only)

This section applies to LAS mode only.

File security flags (FSFs) are used to mark files with various types of information which are then evaluated as part of the checks implemented in the reference monitor. The file security flags supported in the evaluated configuration are identified and explained in the following table:

File Security Flag	Semantics of the flag being enabled/set
FSF_APPEND	If this FSF is set, a file can only be appended to and not altered otherwise in operational mode. In configuration mode, this can be overridden by the PV_TCB privilege.

File Security Flag	Semantics of the flag being enabled/set
FSF_AUDIT	Marks a file as being part of the audit subsystem and allows reading only if a process has the PV_AU_READ privilege and writing only with the PV_AU_WRITE privilege. (See also AU.7)
FSF_MAC_EXMPT	A process with the PV_MAC_OVRRD privilege will ignore MAC restrictions when attempting to access the file system object.
FSF_PDIR	Identifies a partitioned directory.
FSF_PSDIR	Identifies a partitioned subdirectory.
FSF_PSSDIR	Identifies a partitioned sub-subdirectory.
FSF_TLIB	The object is marked as part of the Trusted Library. In the evaluated configuration, this flag can only be changed when the system is in configuration mode or when the system security flag trustedlib_enabled is disabled.
FSF_TLIB_PROC	For executables, a process marked as a TLIB process will only be able to link to shared libraries (*.so) that have the FSF_TLIB flag set. In the evaluated configuration, this flag can only be changed when the system is in configuration mode or when the system security flag trustedlib_enabled is disabled.

Table 26: File security flags (FSFs)

This section maps to the following SFR(s):

- FDP_ACC.1(TCB), FDP_ACF.1(TCB)
- FDP_IFC.2(LS), FDP_IFF.2(LS)

7.2.2.14.8 Protected communication (TP.8)

NFSv4 provides an optional inter-TSF trusted channel between NFSv4 clients and servers. To establish this trusted channel, the TOE user (through the use of the TOE's NAS client and NAS cryptographic library) contacts the NAS (Kerberos Version 5) Authentication Server located in the Operational Environment to obtain a Kerberos ticket granting ticket (TGT). The TGT returned by the Authentication Server contains a key (i.e., Kerberos key distribution) used to identify the TOE user. The TOE then sends its TGT (i.e., Kerberos key distribution) to the NAS Ticket Granting Server (TGS) located in the Operational Environment to obtain an NFSv4 server ticket used to establish a connection between the TOE user and the NFSv4 server. The TGS returns an NFSv4 server ticket (i.e., Kerberos key distribution) to the TOE user to use when contacting the NFSv4 server. The TOE's NFSv4 client then sends the user's NFSv4 server ticket (i.e., Kerberos key distribution) to the NFSv4 server to establish a trusted channel connection with the NFSv4 server using Kerberos Version 5 GSS-API. This trusted channel uses the following cryptographic algorithms:

- AES in CTS mode with 128 bits and 256 bits key sizes and SHA-1
- TDES in CBC mode with 168 bits key size and SHA-1

This trusted channel uses the CLiC module on both the NFSv4 client and server side, but it uses the keys obtained from NAS. (The CLiC version is specified in section 1.5.3.) The TOE's NAS client uses the NAS cryptographic library.

Note: The Kerberos encryption library has not been FIPS certified nor has it been analyzed or tested to conform to cryptographic standards during this evaluation. All cryptography has only been asserted as tested by the vendor.

This section maps to the following SFR(s):

- FCS_CKM.2(NET) - NAS
- FCS_CKM.4 - NAS
- FCS_COP.1(NET) - NAS
- FCS_COP.1(CLIC-ENC)
- FCS_COP.1(CLIC-MD)
- FTP_ITC.1

7.2.2.14.9 Diagnosis (TP.9)

AIX provides a diagnosis program that can be used to check the correct operation of the underlying hardware of the system. This program can be executed by administrators or by hardware maintenance personnel. Results of the diagnosis program are stored in the diagnostic error log file, which can be protected by the discretionary access control functions of AIX against access by normal users.

This section maps to the following SFR(s):

- FPT_TST.1

7.2.2.15 AIX Cryptographic Framework (CRYPTO.1)

AIX supports the AIX Cryptographic Framework (ACF). This framework is implemented by the AIX kernel and allows applications access to cryptographic hardware and software supported by the kernel while at the same time isolating applications from the cryptographic hardware and/or software. In the evaluated configuration, IBM's CLiC software is supported by ACF. (The CLiC version is specified in section 1.5.3.)

ACF provides a [PKCS11] API that is accessible by all applications. The cryptographic algorithms supported in the evaluated configuration are:

- AES 128 bits, 192 bits, 256 bits with CBC, CCM, CTR, CTS, and GCM block chaining modes
- TDES 168 bits with CBC and CTR block chaining modes
- DSA 1024 bits
- RSA 1024 bits and 2048 bits
- SHA-1
- SHA-224
- SHA-256
- SHA-384
- SHA-512

This section maps to the following SFR(s):

- FCS_CKM.1(SYM)
- FCS_CKM.1(RSA)
- FCS_CKM.1(DSA)
- FCS_CKM.4

- FCS_COP.1(CRYPTO-ENC)
- FCS_COP.1(CRYPTO-MD)
- FCS_COP.1(CRYPTO-SGN)
- FDP_RIP.2
- FDP_RIP.3

7.2.3 VIOS

7.2.3.1 Identification and authentication (VIOS.IA)

User identification and authentication (I&A) in the VIOS component of the TOE includes multiple forms of interactive login (e.g., using the Telnet) as well as identity changes through the *su* command. These all rely on explicit authentication information provided interactively by a user.

Identification and authentication of users is performed from a terminal where no user is logged on, via *telnet*, or via the *su* command. All those interfaces use a common mechanism for authentication described in section 7.2.3.1.2. They all use the administrative database described in section 7.2.3.1.1. The administrative database is managed by an authorized administrator but other authorized users are allowed to modify their own password using the *passwd* command. Section 7.2.3.1.3 describes the authentication process. Section 7.2.3.1.4 describes the login processing. Section 7.2.3.1.5 describes the logoff processing.

7.2.3.1.1 User identification and authentication data management (VIOS.IA.1)

VIOS supports a file-based authentication database only. This database is used to manage identification and authentication data used by VIOS.

Administrators, through the built-in commands of the VIOS *ioscli* command line interface, perform changes to the files that constitute the administrative database.

Users are allowed to change their passwords by using the *ioscli* built-in *passwd* command. This command reads the contents of the */etc/security/user* file and modifies the */etc/security/passwd* file for the user's password entry, both of which would ordinarily be inaccessible to a non-privileged user process. Users are also forced to change their passwords at login time, if the password has expired.

The */etc/passwd* file contains the user's name, the ID of the user, an indicator if the password of the user is valid, the principal group ID of the user, and a few other, non-security relevant information. The encrypted password of the user itself is not stored in this file but in the */etc/security/passwd* file which is protected against read access for ordinary users.

The */etc/security/passwd* file contains the encrypted password, the time the password was last changed, and some other information that are not subject to the security functions as defined in this Security Target.

For a complete list of user attributes see section 7.2.3.4.3.

The system administrator defines restrictions on authentication data (i.e., passwords) like the minimum size, the minimum number of non-alphabetic characters, the maximum life time of a password, and the number of unsuccessful login attempts allowed before the account is locked. Those restrictions can be defined on a per user basis and are stored in the */etc/security/user* file. The system administrator can use those parameters to define a password policy.

The */etc/security/lastlog* file contains the time since the last successful login, the time of the last unsuccessful login and the number of unsuccessful login attempts since the last successful login.

This section maps to the following SFR(s):

- FIA_ATD.1(VIOS)
- FIA_SOS.1(VIOS)
- FMT_SMF.1(VIOS)

7.2.3.1.2 Common authentication mechanism (VIOS.IA.2)

VIOS includes a common authentication mechanism which is a subroutine used for all activities that create a user session, including all the interactive login activities and authentication for the *su* command.

The common mechanism includes the following checks and operations:

- Check password authentication
- Check password expiration
- Check whether access should be denied due to too many consecutive authentication failures
- Get user security characteristics (e.g., user, groups, roles)

The common I&A mechanism identifies the user based on the supplied user name, gets that user's security attributes, and performs authentication against the user's password. A result of success indicated by a 1, or a failure indicated by a 0, is returned to the Terminal State Manager (TSM) program which continues the login process.

This section maps to the following SFR(s):

- FIA_UAU.2
- FIA_UID.2(VIOS)

7.2.3.1.3 Interactive login and related mechanisms (VIOS.IA.3)

There are multiple mechanisms for interactive login into VIOS and similar activities:

- the standard *login* program for interactive login sessions on the console of a user's local host
- the *telnet* protocol for ordinary interactive login sessions into VIOS
- the *su* command for changing user identity during a session

All of these mechanisms use the common authentication mechanism described above, but only those mechanisms that create normal interactive sessions use the standard *login* program; other mechanisms implement special-purpose types of sessions.

None of the mechanisms display a password that is entered via a keyboard for authentication. Instead, they provide obscured feedback.

This section and its subsections map to the following SFR(s):

- FIA_UAU.2
- FIA_UAU.7(VIOS)
- FIA_UID.2(VIOS)
- FIA_USB.1(VIOS)

7.2.3.1.3.1 The login program

The *login* program establishes interactive user sessions. *login* is part of the Terminal State Manager (TSM) program. This program prompts for a user identity and authentication (e.g., password), and validates them using the common authentication mechanism described above.

Authentication prompting may also be suppressed when appropriate. If the validation fails, the prompts are repeated until the limits on successive authentication failures are exceeded.

Logging in establishes a user session as follows:

1. Assigns a session identifier
2. Sets exclusive access for the controlling terminal to the process logging in
3. Calls the common authentication mechanism to check validity of the password provided for the account being accessed, and gains the session security attributes
4. Sets up the user environment
5. Checks for password expiration and, if necessary, prompts for password change
6. The process' user and group identities are changed to those of the user
7. The process' authorizations and privileges are set to those of the user
8. User is changed to his or her home directory
9. Invokes the user's default shell

The *login* program is always invoked with open file descriptors for the controlling terminal, used when prompting for identity and authentication information, and passes control to the user's shell when the session is created. At this point, the user session is established, the user environment is set up, and the program replaces itself, using the *exec* system call, with the user's shell).

7.2.3.1.3.2 Login with telnet

The telnet protocol always requests user identity and authentication by invoking the *login* program, which uses the common authentication mechanism. A user can change identity across a telnet connection if the password for another account is known.

7.2.3.1.3.3 User identity changing

VIOS contains the *su* command, but it doesn't allow users to directly execute it. Instead, the command-line interface will execute a subset of the commands available to a role by using the *su* command under the covers. Commands that would normally allow a user to escape to a shell (i.e., the *vi* command) have been modified to disable the shell escape feature. Thus, users cannot directly change their identities during a session.

7.2.3.1.4 Login processing (VIOS.IA.4)

Permissions on the device special files control access to exclusively used public devices. When a user successfully logs in at the local direct attached console, the TSM program changes the ownership of */dev/lft0, /dev/kbd0, /dev/rcm0* to the login UID of the user and sets the permissions on these devices to be readable and writable by this user. */dev/lft0* is a logical device that provides the user's interface to the keyboard and graphics adapter. At system initialization, */dev/lft0* grabs the keyboard and graphics adapter devices. In case of a serially attached ASCII terminal, the tty device associated with the terminal changes ownership to the user that is logged in (for example */dev/tty0*)

The */dev/kbd0* device contains two channels for communication between the keyboard and the device driver. Only one channel is active at any given time. The */dev/lft0* device registers for the first channel when the system boots. The second channel is reserved for the X server (which is not

supported by VIOS). The permissions on the */dev/kbd0* device restrict that only the user who is logged in on the console can access this device. The logged in user could open the second channel, because he/she has permissions. This would redirect the users own keyboard device. This would pose no threat to the operation of the system. The worst thing that would happen is that the login process would not be able to regain access to the */dev/kbd0* device and no other users would be able to login on the console device until the host was rebooted.

The login process executes a *revoke* call to invalidate any open file descriptors for */dev/lft0* or the appropriate */dev/ttyN* device held by a previous user. The *revoke* call modifies the file descriptors entry in the system's open file table, causing further attempts to access the device special file based on that file descriptor to return "bad file descriptor". This ensures that the new login session is isolated from any previous login sessions.

This section maps to the following SFR(s):

- FIA_USB.1(VIOS)

7.2.3.1.5 Logoff processing (VIOS.IA.5)

When a user logs off, all files that were opened by the login shell are closed. Files and devices that were opened by background tasks remain open. However, a background job that had access to the console loses that access prior to the next user's login as stated above.

The ownership of */dev/ttyN, /dev/lft0, /dev/kbd0*, and */dev/rcm0* is returned to root when the logoff occurs.

This section maps to the following SFR(s):

- FIA_USB.1(VIOS)

7.2.3.2 Discretionary access control (VIOS.DA.1)

VIOS resides in a separate LPAR partition and provides basic discretionary access control between VIOS SCSI device drivers acting on behalf of LPAR partitions and SCSI-based logical volumes and physical volumes through mappings. An LPAR partition (via a VIOS SCSI device driver) may be mapped to 0 or more logical and physical volumes, but a volume can only be mapped to at most one LPAR partition. This mapping limits an LPAR partition to only the volumes assigned to it.

VIOS also controls the mapping of VIOS Ethernet adapter device drivers to VIOS Ethernet device drivers acting on behalf of groups of LPAR partitions sharing a virtual network. In the evaluated configuration, only a one-to-one mapping of an Ethernet adapter device driver to an Ethernet device driver acting on behalf of a group of LPAR partitions is allowed. The one-to-one mapping is configured by the administrator and enforced by the device drivers. Also, the Ethernet packets must not be tagged with a VLAN tag in the evaluated configuration. This mechanism can be used to limit which LPAR partitions see certain Ethernet packets.

VIOS is restricted to administrator access only. VIOS allows all administrative roles except the Service Representative roles to manage the access control mechanisms previously mentioned.

This section maps to the following SFR(s):

- FDP_ACC.1(VIOS)
- FDP_ACF.1(VIOS)
- FMT_MSA.1(VIOS)
- FMT_MSA.3(VIOS)
- FMT_MTD.1(VIOS-NV)

- FMT_SMF.1(VIOS)

7.2.3.3 Role-based access control (VIOS.RA)

Like AIX, VIOS provides a role-based access control (RBAC) mechanism. For the purposes of this document, the acronym VRBAC is sometimes used as shorthand to distinguish between VIOS RBAC and AIX RBAC. VIOS does not support the concept of normal users, only administrative users. By default, VIOS RBAC contains the following administrative roles:

- **Prime Administrator (PAdmin)** – This role can execute every command provided by the VIOS command-line interface including the user ID commands and security commands. This role is limited to a single user ID: *padmin*. This user ID is defined in the installation image.
- **System Administrator (SYSAdm)** – This role can execute every command provided by the VIOS command-line interface except for the security commands and user ID commands (exception: they can change their own passwords). System Administrator user accounts do not exist until the Prime Administrator creates one or more.
- **Development Engineer (DEUser)** – This role is used only by IBM personnel to debug problems and run diagnostics. Development Engineer user accounts do not exist until the Prime Administrator creates one or more.
- **Service Representative (SRUser)** – This role allows a service representative to run commands that are required to service the system (shutdown, restart, update system microcode, configure/unconfigure devices, certify, format, etc.). Service Representative user accounts do not exist until the Prime Administrator creates one or more.
- **Administrator (Admin)** - This role is used to manage OEM environments.
- **Run Diagnostics (RunDiagnostics)** - This role is used to run the VIOS diagnostic commands.
- **View (ViewOnly)** - This role is a read-only role and can perform only list-type functions. Users with this role do not have the authority to change the system configuration.

VIOS RBAC provides the ability to define other site-based roles. The TOE uses authorizations to implement roles. It associates authorization sets with a role and ensures that the authorizations required to assume a role are satisfied before allowing operations associated with the role to be performed. Through the use of authorizations, the scope of a role can be reduced or increased.

VIOS supports up to 8 roles per user session. A user can switch between assigned roles by using the *swrole* (switch role) command. This command creates a new shell process and assigns the requested roles to the new shell. This allows a user to add or delete roles from their active set of roles within their login session. Users can only *swrole* to roles that have been assigned to them.

The roles and authorizations tables for VIOS RBAC are maintained in user space, compiled, and loaded into the kernel. When the tables are compiled, they are checked for consistency. The kernel loads the compiled tables during the boot process (if compiled tables do not exist, the boot process will attempt to create compiled tables). Through this methodology, the TOE preserves a secure state and can recover to a consistent, secure state.

This section maps to the following SFR(s):

- FDP_ACC.1(VRBAC), FDP_ACF.1(VRBAC)
- FMT_MSA.1(VRBAC-ADM)
- FMT_MSA.1(VRBAC-AUTH)
- FMT_MSA.1(VRBAC-DFLT)
- FMT_MSA.1(VRBAC-USR)

- FMT_MSA.2(VRBAC)
- FMT_MSA.3(VRBAC)
- FMT_MTD.1(VRBAC)
- FMT_MTD.3(VRBAC)
- FMT_SMR.1
- FTA_LSA.1(VRBAC)
- FTA_TSE.1(VRBAC)

7.2.3.4 Security management (VIOS.SM)

7.2.3.4.1 VIOS roles (VIOS.SM.1)

VIOS provides a role-based access control mechanim known as VIOS RBAC (VRBAC). See section 7.2.3.3 "Role-based access control (VIOS.RA)" for more details. Users that have the appropriate authorizations associated with their account may run administrative programs associated with those authorizations/roles.

This section maps to the following SFR(s):

- FMT_SMR.1

7.2.3.4.2 Access control configuration and management (VIOS.SM.2)

The VIOS SCSI discretionary access control and the VIOS Ethernet discretionary access control are managed by the system administrators. VIOS provides an administrative interface for managing these functions. VIOS SCSI device drivers acting on behalf of the LPAR partitions are not allowed to access a logical or physical volume until the mapping is created in VIOS. A VIOS Ethernet device driver acting on behalf of a group of LPAR partitions sharing a virtual network cannot access a VIOS Ethernet adapter device driver and vice versa until a mapping is created in VIOS.

This section maps to the following SFR(s):

- FMT_MSA.1(VIOS)
- FMT_MSA.3(VIOS)
- FMT_MTD.1(VIOS-NV)

7.2.3.4.3 Management of user, group, and authentication data (VIOS.SM.3)

This section and its subsections map to the following SFR(s):

- FMT_MTD.1(VIOS-ADI)
- FMT_MTD.1(VIOS-ADM)
- FMT_MTD.1(VIOS-SA)
- FMT_REV.1(VIOS)
- FMT_SMF.1(VIOS)

7.2.3.4.3.1 Creating new users

The Prime Administrator can create a new user and can assign a role to the user's account. The initial password and other security attributes are defined by the Prime Administrator using various utilities (e.g., the *passwd* command). The new user will be disabled until the initial password is set.

Attributes that can be set for each user are among others (a complete list can be found in the description of the VIOS *chuser* command):

- Lock attribute (i.e., temporarily locking a user account)
- List of roles the user belongs to
- List of groups the user belongs to
- Home directory for this user
- Number of consecutive unsuccessful login attempts allowed before the user account is locked
- Password parameter including the maximum password age, minimum password length, etc.

Those attributes are stored in the following files:

- */etc/group*
- */etc/passwd*
- */etc/security/passwd*
- */etc/security/user*
- */etc/security/user.roles*

7.2.3.4.3.2 Modification of user attributes

User attributes can be modified by the Prime Administrator. Modifications of user attributes require the modification of the administration database that contains the user attributes (mainly */etc/security/user*).

7.2.3.4.3.3 Management of authentication data

The Prime Administrator has the capability to define rules and restrictions for passwords used to authenticate users. Table 27 contains the VIOS password parameters.

VIOS Password Parameter	Description
maxage	Maximum number of weeks that can pass before a password must be changed.
maxexpired	Maximum number of weeks beyond maxage that a password can be changed before administrative action is required to change the password.
minother	Minimum number of non-alphabetic characters the new password must contain. (Other characters are any ASCII printable characters that are non-alphabetic and are not national language code points).
minlen	Minimum number of characters the new password must contain.
maxrepeats	Maximum number of times a character can be used in the new password.
histexpire	Number of weeks that a user is unable to reuse a password.
histsize	Number of previous passwords that cannot be reused.

Table 27: VIOS password parameters

Users are also allowed to change their own password using the *passwd* command. The password restrictions defined by the system administrator are enforced by the *passwd* command.

7.2.3.4.3.4 Revoking users

The Prime Administrator is authorized to revoke the authentication data, group memberships, and security-relevant roles of VIOS users. When the Prime Administrator revokes or modifies the security attributes of a VIOS user, the changes take effect the next time the user logs in.

8 Abbreviations, Terminology and References

8.1 Abbreviations

ACE

Access Control Entry

ACF

AIX Cryptographic Framework

ACL

Access Control List

AES

Advanced Encryption Standard

AIX

Advanced Interactive Executive

AIXC

AIX Classic

ANSI

American National Standards Institute

API

Application Programming Interface

BAS

Basic AIX Security

CBC

Cipher-Block Chaining

CBC-MAC

Cipher-Block Chaining Message Authentication Code

CC

Common Criteria

CC

Common Criteria for Information Technology Security Evaluation

CCM

Counter with CBC-MAC

CDE

Common Desktop Environment

CDRFS

CD-ROM File System

CD-ROM

Compact Disc Read Only Memory

CID

Corral ID

CIPSO

Common IP Security Option

CLiC

IBM CryptoLite for C

CM

Configuration Management

CTR

Counter

CTS

Ciphertext Stealing

DAC

Discretionary Access Control

DLPAR

Dynamic LPAR

DRNG

Deterministic Random Number Generator

EAL

Evaluation Assurance Level

ECD

Extended Component Definition

EFS

Encrypted File System

EGID

Effective Group ID

EOF

End of File

EPS

Effective Privilege Set

EUID

Effective User ID

FIFO

First In First Out

FIPS

Federal Information Processing Standard

FIV

File Integrity Verification

FPR

Floating Point Register

FSF

File Security Flag

FSO

File System Object

FSP

Functional Specification

FTP

File Transfer Protocol

GA

General Availability

GCM

Galois/Counter Mode

GID

Group ID

GMAC

Galois Message Authentication Code

GPR

General Purpose Register

GSKit

IBM Global Security Kit

HLD

High Level Design

HTML

Hypertext Markup Language

I&A

Identification and Authentication

ID

Identification

IEC

International Electrotechnical Commission

IEEE

Once known as the Institute of Electrical and Electronics Engineers

IKE

Internet Key Exchange

IP

Internet Protocol

IPC

Inter-Process Communication

IPsec

Internet Protocol Security (a.k.a. IPSEC)

IPSO

Internet Protocol Security Option

ISO

International Standards Organization

ISSO

Information System Security Officer

JFS

Journaled File System

JFS2

JFS version 2

KAT

Kernel Authorization Table

KCT

Kernel Privileged Command Table

KDC

Key Distribution Center

KDT

Kernel Privileged Device Table

KRT

Kernel Role Table

LAS

Labeled AIX Security

LDAP

Lightweight Directory Access Protocol

LFS

Logical File System

LPAR

Logical Partition

LPP

Licensed Product Package

LPS

Limiting Privilege Set

MAC

Mandatory Access Control

MPS

Maximum Privilege Set

NAS

IBM Network Authentication Service

NFS

Network File System

NIM

Network Install Manager

NPTRNG

Non-Physical True Random Number Generator

NVRAM

Non-Volatile Random Access Memory

OID

Object Identification

OR

Observation Report

OSP

Organizational Security Policy

PDF

Portable Data Format

PID

Process Identifier

PP

Protection Profile

PROCFS

Process File System

PRPQ

Programming Request for Price Quote

PTF

Program Temporary Fix

RAID

Redundant Array of Independent Disks

RAM

Random Access Memory

RBAC

Role-Based Access Control

RIPSO

Revised IP Security Option

RNG

Random Number Generator

RPC

Remote Procedure Call

RSH

Remote Shell

RTAS

Run-Time Abstraction Layer

SA

System Administrator

SAR

Security Assurance Requirement

SCSI

Small Computer System Interface

SED

Stack Execution Disable

SEM

Superuser Emulation Mode

SFP

Security Function Policy

SFR

Security Functional Requirement

SHA

Secure Hash Algorithm

SL

Sensitivity Label

SMIT

System Management Interface Tool

SO

System Operator

SPECFS

Special File System

SSF

System Security Flag

SSL

Secure Sockets Layer

ST

Security Target

SysV

UNIX System V

TCB

Trusted Computing Base

TCP

Transmission Control Protocol

TDES

Triple Data Encryption Standard

TDS

IBM Tivoli Directory Server

TID

Thread Identifier

TL

Integrity Label

TN

Trusted Network

TOE

Target of Evaluation

TSC

TSF Scope of Control

TSD

Trusted Signature Database

TSF

TOE Security Functionality

TSP

TOE Security Policy

UDFS

Universal Data Standard File System

UDP

User Datagram Protocol

UID

User ID

VFS

Virtual File System

VIOS

Virtual Input/Output Server

VLAN

Virtual Local Area Network

VMM

Virtual Memory Manager

VRBAC

VIOS RBAC

WPAR

Workload Partition

8.2 Terminology

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

Access

A right to interact with a system resource.

Administrative User

This term refers to an administrator of an AIX system. Some administrative tasks require the use of authorizations, which can be assigned to one or more user accounts while other tasks can be performed by specified users only. Authorizations can determine which privileges are assigned to a task.

AIX

This document uses the term AIX for AIX 7.1 when discussing AIX 7.1 in general terms (i.e. when it applies to both BAS mode and LAS mode).

Authentication data

This includes a user identifier, password and authorizations for each user of the product.

Authorization

An attribute associated with a user account that allows the user to run restricted programs or to run public programs with additional privilege.

Authorized Administrator

A user whose account and session has authorizations allowing privileged, administrative commands to be run.

Backward secrecy

For a random number generator, even when extracting an arbitrary sequence of random bits, it is not possible to deduce any previously extracted random bits.

BAS Mode

This term (Basic AIX Security Mode) refers to the evaluated version and installation method of AIX that complies with [OSPP] and several of the [OSPP] Extended Packages except [OSPP-LS].

Category

The non-hierarchical portion of the sensitivity label. The terms compartment and category are used interchangeably within this ST.

Classification

The component of a sensitivity label that is hierarchical.

Compartment

See category.

Discretionary Access Control (DAC)

A control mechanism that mediates access based on user identity and owner-controlled attributes on objects.

Dominates

Greater than or equal to, as used with labels, privileges, and authorizations.

Forward secrecy

For a random number generator, even with the knowledge of all extracted random bits, it is not possible to predict the next random bits that will be extracted.

Integrity Label (TL)

An attribute of a system resource that represents the level of trust associated with the integrity of the resource or data associated with the resource. A TL has only a hierarchical component.

LAS Mode

This term (Labeled AIX Security Mode) refers to the evaluated version and installation method of Trusted AIX that complies with [OSPP] and several of the [OSPP] Extended Packages including [OSPP-LS].

Mandatory Access Control (MAC)

A control mechanism that mediates access based on a label associated with the subject and a label associated with the object and where such labels are not generally under the control of the user/owner associated with the object or subject.

Mandatory Integrity Control (MIC)

A control mechanism that mediates access based on the integrity label associated with the subject and the integrity label associated with the object and where such labels are not generally under the control of the user/owner associated with the object or subject.

Mediation

The act of applying rules to determine access to TOE protected objects.

Object

In AIX, objects belong to one of four categories: file system objects, other kernel objects (such as processes, programs and inter-process communication), window system objects, and miscellaneous objects.

Privileges

A privilege is an attribute of a process that allows the process to bypass specific restrictions and limitations of the system. Privileges are used to override security constraints and are controlled by an administrator.

Product

The term product is used to define all hardware and software components that comprise the AIX system including VIOS.

Public Object

A type of object for which all subjects have read access, but only the TSF or the system administrators have write access.

Security Attributes

As defined by functional requirement FIA_ATD.1, the term ‘security attributes' includes the following as a minimum: user identifier; group memberships; user authentication data; and security-relevant roles.

Sensitivity Label (SL)

An attribute of system resources that represents the sensitivity of the resource or data associated with the resource. An SL has both a hierarchical and a non-hierarchical component.

Subject

There are two classes of subjects in AIX:

- untrusted internal subject - this is an AIX process running on behalf of some user, running outside of the TSF (for example, with no privileges).
- trusted internal subject - this is an AIX process running as part of the TSF. Examples are service daemons and the process implementing the identification and authentication of users.

System

Includes the hardware, software and firmware components of the AIX product which are connected/networked together and configured to form a usable system.

Trusted AIX

The term refers to the multi-level security version of AIX.

Trusted Computing Base (TCB)

The software components of the TOE that enforce the TSFs and which must remain inviolate in order to enforce the system security policies.

Trusted Execution (TE)

The integrity protection mechanism that enforces integrity verification at file load.

Trusted Network (TN)

The component of the system that labels internal and external network traffic and which mediates access between processes and network resources.

User

Any individual/person who has a unique user identifier and who interacts with the AIX product. Users can further be categorized as follows:

- **Authorized User** - A user who may, in accordance with the TSP, perform an operation.
- **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.
- **Role** - A predefined set of rules establishing the allowed interactions between a user and the TOE.
- **User** - Any entity (external IT entity or human user) outside the TOE that interacts with the TOE.
  - **External IT entity** - Any IT product or system, untrusted or trusted, outside of the TOE that interacts with the TOE.
  - **Human user** - Any person who interacts with the TOE.

8.3 References

BSI-AIS20 **BSI Application Notes and Interpretation of the Scheme (AIS), AIS 20**
Version 1
Date December 2, 1999
Location https://www.bsi.bund.de/cae/servlet/contentblob/478152/publicationFile/30265/ais20e_pdf.pdf

CC	**Common Criteria for Information Technology Security Evaluation**	
	Version	3.1R3
	Date	July 2009
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R3.pdf
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R3.pdf
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R3.pdf
FIPS180-3	**Secure Hash Standard (SHS)**	
	Version	FIPS PUB 180-3
	Date	October 2008
	Location	http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf
FIPS186-3	**Digital Signature Standard (DSS)**	
	Version	FIPS PUB 186-3
	Date	June, 2009
	Location	http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf
FIPS188	**Standard Security Label for Information Transfer**	
	Version	FIPS PUB 188
	Date	September 6, 1994
	Location	http://csrc.nist.gov/publications/fips/fips188/fips188.pdf
FIPS197	**Specification for the ADVANCED ENCRYPTION STANDARD (AES)**	
	Version	FIPS PUB 197
	Date	November 26, 2001
	Location	http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
OSPP	**BSI Operating System Protection Profile, Cert ID: BSI-CC-PP-0067**	
	Version	2.0
	Date	2010-06-01
	Location	https://www.bsi.bund.de/cae/servlet/contentblob/1098082/publicationFile/88584/pp0067b_pdf.pdf
OSPP-AM	**BSI OSPP Extended Package – Advanced Management, Cert ID: BSI-CC-PP-0067**	
	Version	2.0
	Date	2010-05-28
	Location	https://www.bsi.bund.de/cae/servlet/contentblob/1098148/publicationFile/88582/pp0067_EP_zip.zip
OSPP-CRYPTO	**BSI OSPP Extended Package – General Purpose Cryptography, Cert ID: BSI-CC-PP-0067**	
	Version	2.0
	Date	2010-05-28
	Location	https://www.bsi.bund.de/cae/servlet/contentblob/1098148/publicationFile/88582/pp0067_EP_zip.zip

OSPP-IV **BSI OSPP Extended Package – Integrity Verification, Cert ID: BSI-CC-PP-0067**
Version 2.0
Date 2010-05-28
Location https://www.bsi.bund.de/cae/servlet/contentblob/1098148/publicationFile/88582/pp0067_EP_zip.zip

OSPP-LS **BSI OSPP Extended Package – Labeled Security, Cert ID: BSI-CC-PP-0067**
Version 2.0
Date 2010-05-28
Location https://www.bsi.bund.de/cae/servlet/contentblob/1098148/publicationFile/88582/pp0067_EP_zip.zip

OSPP-VIRT **BSI OSPP Extended Package – Virtualization, Cert ID: BSI-CC-PP-0067**
Version 2.0
Date 2010-05-28
Location https://www.bsi.bund.de/cae/servlet/contentblob/1098148/publicationFile/88582/pp0067_EP_zip.zip

PKCS1 **PKCS #1: RSA Cryptography Standard**
Version 1.5
Date November 1, 1993
Location http://www.rsa.com/rsalabs/node.asp?id=2125

PKCS11 **PKCS #11: Cryptographic Token Interface Standard**
Version 2.20
Date June 28, 2004
Location http://www.rsa.com/rsalabs/node.asp?id=2133

PwdSecHist **Password Security: A Case History (from Communications of the ACM)**
Author(s) Robert Morris and Ken Thompson
Date November 1979

RFC1108 **U.S. Department of Defense Security Options for the Internet Protocol**
Author(s) S. Kent
Date November 1991
Location http://www.ietf.org/rfc/rfc1108.txt

RFC2460 **Internet Protocol, Version 6 (IPv6) Specification**
Author(s) S. Deering, R. Hinden
Date December 1998
Location http://www.ietf.org/rfc/rfc2460.txt

RFC3961 **Encryption and Checksum Specifications for Kerberos 5**
Author(s) K. Raeburn
Date February 2005
Location http://www.ietf.org/rfc/rfc3961.txt

RFC3962 **Advanced Encryption Standard (AES) Encryption for Kerberos 5**
Author(s) K. Raeburn
Date February 2005
Location http://www.ietf.org/rfc/rfc3962.txt

RFC4120 **The Kerberos Network Authentication Service (V5)**
Author(s) C. Neuman, T. Yu, S. Hartman, K. Raeburn
Date July 2005
Location http://www.ietf.org/rfc/rfc4120.txt

RFC4301 **Security Architecture for the Internet Protocol**
Author(s) S. Kent, K. Seo
Date December 2005
Location http://www.ietf.org/rfc/rfc4301.txt

RFC4869 **Suite B Cryptographic Suites for IPsec**
Author(s) L. Law, J. Solinas
Date May 2007
Location http://www.ietf.org/rfc/rfc4869.txt

SecGuide **AIX Version 7.1: Security**
Date October 2011

SP800-38A **Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode**
Version Addendum to NIST Special Publication 800-38A
Date October 2010
Location http://csrc.nist.gov/publications/nistpubs/800-38a/addendum-to-nist_sp800-38A.pdf

SP800-38D **Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC**
Version NIST Special Publication 800-38D
Date November 2007
Location http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf

SP800-67 **Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher**
Version NIST Special Publication 800-67 Version 1.1
Date May 2008
Location http://csrc.nist.gov/publications/nistpubs/800-67/SP800-67.pdf

