STOP OS™ Version 7.3.1
Security Target

Version 1.08
December 9, 2011

BAE SYSTEMS

This document was originally developed by Science Applications International Corporation, 6841 Benjamin Franklin Drive, Columbia, MD 21046. It has since been modified by BAE Systems.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. Security Target Introduction

This Security Target (ST) describes the IT security requirements for the STOP OS™ Version 7.3.1 product. STOP OS is a multi-level secure (MLS) operating system that provides flexible security policies and a Linux® compatible Application Programming Interface (API). STOP OS includes trusted applications as well as a number of untrusted packages.

STOP OS is developed by BAE Systems Information Solutions Inc., 2525 Network Place, Herndon, VA 22171; hereafter, referred to as BAE Systems.

The TOE is software-only; the hardware on which STOP OS runs is defined to be in the IT environment.

1.1 Security Target, TOE and CC Identification

ST Title –STOP OS Security Target

ST Version – Revision 1.08

ST Date – December 9, 2011

TOE Identification – STOP OS 7.3.1

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 3, July 2009.

1.2 Conformance Claims

This TOE conforms to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional requirements, Version 3.1, Revision 3, July 2009.
    - Part 2 extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance requirements, Version 3.1, Revision 3, July 2009.
    - Part 3 conformant
- Package Conformance:
    - Evaluation Assurance Level 4 (EAL 4) augmented with ALC_FLR.3
- Protection Profile:
    - There are no claims to any currently active Protection Profile. [1,2]

[1] A large amount of the language in the SFRs, ASRs, threats, assumptions and policies are taken from the US Government Protection Profile for General-Purpose Operating Systems in a Networked Environment, Version 0.7 (GPOSPP), 10 August 2009.

[2] The ST includes all of the requirements from the following archived PPs: Labeled Security Protection Profile, Version 1.b, 8 October 1999; Role-Based Access Control Protection Profile, Version 1.0, July 30 1998; Controlled Access Protection Profile, Version 1.d, 8 October 1999.

1.3 Conventions, Terminology, Acronyms

This section specifies the formatting information used in the ST.

1.3.1 Conventions

The following conventions have been applied in this document:

- All requirements in this ST are reproduced relative to the requirements defined in CC v3.1r3.
- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
  - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a letter placed at the end of the component identifier. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
  - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that in cases where a selection operation is combined with an assignment operation and the assignment is null, the assignment operation is simply deleted leaving only the completed selection to identify the combination of operations. Alternately, if the assignment is not null the assignment is identified with embedded brackets which are bolded and italicized (e.g., [***[selected-assignment]***]).
  - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
  - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Extended requirements (i.e., those not taken from the CC) are identified by '_EXT' appearing as an element of the requirement label.
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.3.2 Acronyms

CC	Common Criteria
CM	Configuration Management
DAC	Discretionary Access Control
GPOSPP	US Government Protection Profile for General-Purpose Operating Systems in a Networked Environment
IPC	Inter-Process Communication
LSPP	Labeled Security Protection Profile
MLS	Multi-Level Secure
OS	Operating System
PI	Programming Interface
PP	Protection Profile
RBAC	Role Based Access Control
ST	Security Target
STOP OS	STOP Operating System
TCSEC	Trusted Computer System Evaluation Criteria
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSP	TOE Security Policy
TOE	Target Of Evaluation

1.3.3 Glossary of Terms

Access	Interaction between an entity and an object that results in the flow or modification of data.
Access control	Security service that controls the use of resources[3] and the disclosure and modification of data[4].
Accountability	Tracing each activity in an IT system to the entity responsible for the activity.
Administrator	An authorized user who has been specifically granted the authority to manage some portion or the entire TOE and thus whose actions may affect the TSP. Administrators may possess special privileges that provide capabilities to override portions of the TSP.
Assurance	A measure of confidence that the security features of an IT system are sufficient to enforce its security policy.
Attack	An intentional act attempting to violate the security policy of an IT system.
Authentication	Security measure that verifies a claimed identity.
Authentication data	Information used to verify a claimed identity.
Authorization	Permission, granted by an entity authorized to do so, to perform functions and access data.
Authorized user	An authenticated user who may, in accordance with the TSP, perform an operation.
Availability	Timely[5], reliable access to IT resources.
Compromise	Violation of a security policy.
Confidentiality	A security policy pertaining to disclosure of data.
Critical cryptographic security parameters	Security-related information appearing in plaintext or otherwise unprotected form and whose disclosure or modification can compromise the security of a cryptographic module or the security of the information protected by the module.
Cryptographic boundary	An explicitly defined contiguous perimeter that establishes the physical bounds (for hardware) or logical bounds (for software) of a cryptographic module.
Cryptographic key (key)	A parameter used in conjunction with a cryptographic algorithm that determines: – the transformation of plaintext data into ciphertext data, – the transformation of ciphertext data into plaintext data, – a digital signature computed from data, – the verification of a digital signature computed from data, or a data authentication code computed from data.

[3] hardware and software
[4] stored or communicated
[5] according to a defined metric

Cryptographic module	The set of hardware, software, and/or firmware that implements approved security functions (including cryptographic algorithms and key generation) and is contained within the cryptographic boundary.
Cryptographic module security policy	A precise specification of the security rules under which a cryptographic module must operate.
Defense-in-depth	A security design strategy whereby layers of protection are utilized to establish an adequate security posture for an IT system.
Discretionary Access Control (DAC)	A means of restricting access to objects based on the identity of subjects and groups to which they belong. The controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject.
Enclave	A collection of entities under the control of a single authority and having a homogeneous security policy. They may be logical, or based on physical location and proximity.
Entity	A subject, object, user or external IT device.
General-Purpose Operating System	A general-purpose operating system is designed to meet a variety of goals, including protection between users and applications, fast response time for interactive applications, high throughput for server applications, and high overall resource utilization.
Hypervisor	A software product that allows one or more virtual machines to run on a single physical machine.
Identity	A means of uniquely identifying an authorized user of the TOE.
Integrity	A security policy that pertains to preserving unauthorized data modification and preserving internal and external consistency of the data.
Mandatory Access Control (MAC)	A means of restricting access to objects in a manner where the user does not have full control of defining the access.
Named object	An object that exhibits all of the following characteristics: - The object may be used to transfer information between subjects of differing user identities within the TSF. - Subjects in the TOE must be able to request a specific instance of the object. - The name used to refer to a specific instance of the object must exist in a context that potentially allows subjects with different user identities to request the same instance of the object.
Object	An entity under the control of the TOE that contains or receives information and upon which subjects perform operations.
Operating environment	The total environment in which a TOE operates. It includes the physical facility and any physical, procedural, administrative and personnel controls.
Persistent storage	All types of data storage media that maintain data across system boots (e.g., hard disk, CD, DVD).
Public object	An object for which the TSF unconditionally permits all entities “read” access. Only the TSF or authorized administrators may create, delete, or modify the public objects.
Resource	A fundamental element in an IT system (e.g., processing time, disk space, and memory) that may be used to create the abstractions of subjects and objects.

Secure State	Condition in which all TOE security policies are enforced.
Security attributes	TSF data associated with subjects, objects and users that is used for the enforcement of the TSP.
Security Target (ST)	A set of security requirements and specifications to be used as the basis for evaluation of an identified TOE.
Subject	An active entity within the TSC that causes operations to be performed. Subjects can come in two forms: trusted and untrusted. Trusted subjects are exempt from part or all of the TOE security policies. Untrusted subjects are bound by all TOE security policies.
Target of Evaluation (TOE)	An IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation.
Threat	Capabilities, intentions and attack methods of adversaries, or any circumstance or event, with the potential to violate the TOE security policy.
User	Any person who interacts with the TOE.
Vulnerability	A weakness that can be exploited to violate the TOE security policy.

1.4 Security Target Overview and Organization

This ST describes the STOP OS TOE, intended environments, security objectives, security requirements (for the TOE), security functions, and all necessary rationale. This information is organized into the following additional sections:

- TOE Description (Section 2)
- Security Problem Definition (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8)

2. TOE Description

STOP OS has a strong history of evaluated products in its ancestry, including:

- SCOMP (Secure Communications Processor) evaluated at A1 against the Trusted Computer Security Evaluation Criteria (TCSEC a.k.a Orange Book);
- XTS-200 (STOP 3) and XTS-300® (STOP 4 and STOP 5) evaluated at B3 against the TCSEC;
- XTS-400® (STOP 6) evaluated twice at Evaluation Assurance Level (EAL) 5 against the Common Criteria
    - STOP 6.1.E: http://www.niap-ccevs.org/cc-scheme/st/?vid=3012
    - STOP 6.4 U4: http://www.niap-ccevs.org/cc-scheme/st/vid10293

STOP OS builds on that history but has been significantly re-engineered to provide the following key features:

- A security label is assigned to all subjects and objects in the system with security checks performed on all operations.
- Security labels consist of three parts:
  - Multi-Level Security (both sensitivity and integrity)
  - Role-Based Access Control (RBAC)
  - Discretionary Access Control (DAC)
- All users are assigned a clearance identifying the maximum security label of data that they may access.
- Simultaneous use of the system by multiple users, all with different clearances and needs-to-know.
- Simultaneous network connectivity to networks of differing sensitivities/classifications
- Encrypted file systems using the AES-256 encryption algorithm.
- Provides an operating environment that includes common Linux commands and tools.
- Provides a Linux-like programmatic interface to enable developers to port or develop applications easily on the system.

The access control mechanisms provided by STOP OS include two types of Mandatory Access Control (MAC):

- Multi-Level Security (MLS): Incorporating the Bell-LaPadula security and Biba integrity models (BL/B), STOP OS labels all objects on the system with a sensitivity label (includes a sensitivity level and optional compartments) and an integrity label (includes an integrity level and optional compartments). This mandatory, hierarchal security policy utilizes sensitivity levels to protect data from unauthorized (lesser cleared) access, while protecting processes, configuration data, and logs from attacks on the system's integrity.
- Role-Based Access Control (RBAC): New to STOP OS, RBAC provides a flexible mandatory access control mechanism allowing administrators to tie specific actions to specific roles and roles to subjects and objects. RBAC allows the security architect to restrict actions to specific object and subject combinations. The RBAC implementation is based on "The NIST Model for Role-Based Access Control: Towards a Unified Standard" by Ravi Sandhu, David Ferraiolo, and Richard Kuhn (2000).

The access control mechanisms provided by STOP OS also include a Discretionary Access Control (DAC):

- Discretionary Access Control (DAC) allows for simple configuration of read, write, and execute of objects to users and groups. DAC enforced by STOP OS is similar to the traditional UNIX-style DAC. A user may be assigned to only one group in STOP OS7.

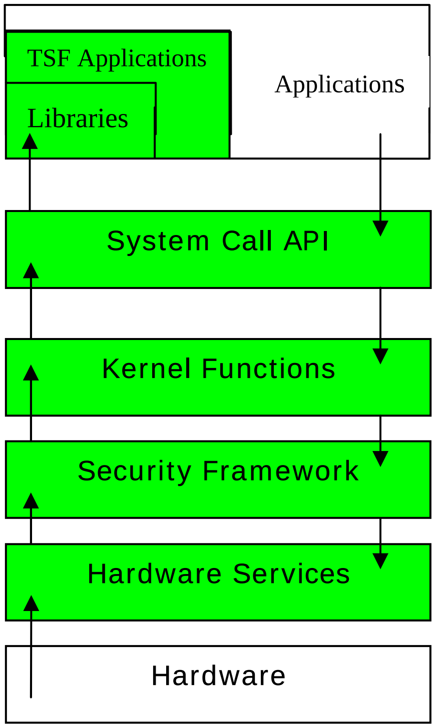
The security policy can be tailored to implement a customized security policy that meets unique customer needs.

The product provides user identification and authentication used for policy enforcement through user identifiers and passwords (both local and remote via SSH) and/or public-key authentication (only remote via SSH) and individual accountability through its auditing capability. Data scavenging is prevented through the control of object and subject reuse.

The TOE is not a distributed system, though it can be networked with other STOP OS systems and non-STOP OS systems.

2.1 TOE Architecture

The STOP OS architecture is shown in the figures below. STOP OS is mostly contained within a monolithic kernel that includes Hardware Services, Security Framework, Kernel Functions, and the System Call API. Additionally, STOP OS includes several applications to support the security functions. All security decisions are enforced by the kernel and not in applications. Figure 1 provides an architectural overview that describes STOP OS in its operational environment. The STOP OS TOE is identified in the shaded boxes while the IT Environment is shown in the unshaded boxes.



TSF Applications (and the libraries they use) needed to support the TOE security functions (TSF) (e.g. login, getty, passwd, shell).

Non-TSF applications

Linux-compatible API, small STOP-specific API

Thread Management, scheduling, virtual memory, virtual file system, sockets, IPC

BL/B, RBAC, Unix DAC, Metapolicy

Memory Management, Device Access, Timers, Interrupt and Machine Exception Vectors.

IA-32 Compatible processor with page-level memory

🟩 TOE Component   □ IT Environment

Figure 2-1STOP OS Architectural Overview

2.1.1 Additional Architecture-based Protections

STOP OS uses the protection mechanisms provided by the CPU on which it executes, including descriptor privilege levels, gate descriptors, segment attributes (read, write, execute), and call/return instructions. The privilege level (PL) protection mechanism ranges from PL0 (the most privileged) to PL3 (the least privileged). These privilege levels are also referred to as “rings”.

STOP OS incorporates a two-ring architecture to protect data and functionality. The monolithic kernel that includes Kernel Functions, a Security Framework, System Call API, and Hardware services shown above in Figure 1 executes in ring PL0, where it is protected from applications that execute in ring PL3.

2.2 Scope of the Evaluation

STOP OS includes additional optional administrator-installable packages which are within the scope of the TOE. A few of these packages are responsible for some of the claimed security functions (TSF packages), but most are not responsible for the claimed security functions and rely on the TSF for the claimed security.

The following packages are considered to be TSF packages:

- OpenSSH: assist in the implementation of secure remote access
- OpenSSL: supports the general-purpose cryptographic operations support (FCS)
- PAM: supports application programs that require authentication allowing them to be written independent of the underlying authentication scheme

2.3 Security Environment TOE Boundary

2.3.1 Physical Boundary

The physical boundary of the TOE is the software boundary of the STOP operating system.

2.3.1.1 Hardware Requirements of the TOE

STOP OS must be hosted on a hardware platform which can either be one of a variety of models supplied by BAE Systems as part of the XTS® product line or may be other hardware that meets the minimum system requirements.

The hardware platform must provide the following:

- x86 CPU(s) with the instruction set and features of an i686 or later (e.g. Intel™ Pentium™, Core™, Core2 or AMD™ Athlon™).
- At least 128 MB of system memory

The hardware platform may also provide the following peripherals:

- Storage devices and backup devices supported by the TOE software (hard disks, optical drives, tape drives, floppy disk drives)
- Network adapters supported by the TOE software

The TOE will use up to 8 cores on the provided CPUs. The CPU may support 64-bit mode, but the TOE will only execute in 32-bit mode.

The hardware platform may consist of physical hardware on which the TOE executes directly, or may consist of virtual hardware provided by a hypervisor which provides virtualized access to a hardware platform capable of the protection mechanisms described in 2.1.1.

2.3.2 Logical Boundaries

The logical boundaries of TOE are realized in the security functions it implements.

2.3.2.1 Security Audit

STOP OS records security relevant events related to the security functions it provides. These events are associated with individual users for individual accountability and can be accessed only by authorized administrators[6]. Audit management functionality is available to search the audit log as well as select the audit events to be audited.

STOP OS protects the audit records collected by ensuring that sufficient disk space is available to write all records that have been generated. The system can be configured to automatically shut down when disk space is not available for future audit records to be written to disk or an error occurs when writing audit records to disk. STOP OS also provides reliable time information to record in its audit records.

2.3.2.2 Access Control

STOP OS enforces a metapolicy that calls the following sub-policies, thereby, enforcing access control on its subjects and objects.

- Role-Based Access Control (RBAC): a flexible, permission-based mandatory access control policy

[6] The term *authorized administrator* is used to generally refer to an administrator authorized (e.g., by assigned role or action assigned to the role) to perform a corresponding function depending on the context in which the term is used.

- Bell-LaPadula multi-level security policy: a mandatory access control (MAC) read-down/write-up policy
- Biba multi-level integrity policy: a mandatory integrity control (MIC) read-up/write-down policy[7]
- Traditional UNIX-style discretionary access control (DAC)

The user is only allowed to by-pass specific access checks if the user is authorized by the role assigned to the user to perform specific actions that exempt a user from a specific check. Actions included in a role definition (i.e. role-actions) can exempt a check from being made. All users assigned exemptions are considered authorized administrators. STOP OS exports user data with and without security labels, and imports user data with and without security labels. No object or subject residual information is made available to users.

STOP OS includes a host-based Packet Filtering Firewall that allows the administrator to control inbound/outbound TOE network traffic.

2.3.2.3 Cryptographic Support

STOP OS provides cryptographic services to applications and to support the OpenSSH protocol used to establish a secure channel to allow remote access, and to optionally encrypt filesystems. The cryptographic services provided to applications are encryption/decryption, digital signatures, and hashing. The cryptographic services provided to support OpenSSH include encryption/decryption, digital signatures, public key authentication, and Diffie-Hellman key agreement. Encrypted filesystems are supported by a kernel-internal cryptographic library.

2.3.2.4 Identification & Authentication

STOP OS ensures that all users are authenticated prior to allowing access to the TOE. STOP OS only allows authorized administrators to manage user accounts (e.g. define role(s), clearance, default labels). Upon successful authentication, the security attributes for the subject are assigned including the user identifier, group, and clearance. STOP OS allows for the configuration of password criteria allowing the TOE the ability to meet the strength of secrets requirement. During authentication, the authentication information is not echoed. Remote authentication is enforced using SSH and either public-key authentication or a password. STOP OS can be configured to lock user accounts when the amount of failed authentication attempts reaches an administrator defined limit.

2.3.2.5 Security Management

STOP OS includes the security management roles of administrator and privileged users. Privileged users are users assigned to roles that authorize them to perform specific privileged actions. Users in the administrator role can perform all management functions. Privileged users are those users assigned to roles with exemptions. Exemptions (also referred to sometimes as privileged actions) are actions assigned to roles that bypass security policy checks or allow specific security management related functionality. Management of user accounts, objects, including management of security labels, roles and audit policies is restricted to only those authorized by their role membership.

2.3.2.6 Protection of the TSF

STOP OS preserves a secure state upon failure to access TSF data (role definitions); power failure; or unrecoverable hardware, firmware, or TSF software failure. STOP OS provides the ability to run a suite of self tests upon initial start-up or on-demand to demonstrate the correct operation of the cryptographic module(s) and provides a capability to verify the integrity of TSF executable code.

2.3.2.7 TOE Access

STOP OS is able to restrict the security attribute assigned to a session to be within the clearance assigned to the user on whose behalf the session is running. Additionally, a session will not be initiated if a label within the user's clearance cannot be assigned to the user. STOP OS allows for an administrator defined limit to be placed on the

[7] Bell-LaPadula multi-level (MAC) and Biba multi-level integrity policy (MIC) are implemented as two parts of the same policy – referred to as the Bell-LaPadula/Biba (BL/B) policy.

amount of concurrent interactive sessions a user may start. STOP OS will automatically terminate sessions that have been inactive for an administrator configured period of time. Additionally, upon the establishment of a session, STOP OS provides an administrator defined advisory banner and a history of the user's last successful authentication as well as the number of unsuccessful authentication attempts.

2.3.2.8 Trusted Path

The TSF provides the ability to access the TOE remotely and protects that communication using OpenSSH which uses the cryptographic mechanisms described in the Cryptographic Support security function that protect the communication from disclosure and undetected modification. The TSF provides a trusted communication path between local users and the TSF.

2.4 TOE Documentation

STOP OS provides administrator and user guidance on how to utilize the TOE security functions and warnings to authorized administrators and users about actions that can compromise the security of the TOE. The installation and generation procedures, included in the administrator guidance, describe the steps necessary to install STOP OS in accordance with the evaluated configuration.

BAE Systems also provides delivery documentation and procedures to identify the TOE, allow detection of unauthorized modifications of the TOE and installation and generation instructions at start-up. BAE Systems' delivery procedures describe the electronic and non-electronic procedures to be used to detect modification to the TOE.

All of the administrator and user guidance is documented in:

- STOP OS Trusted Operating System Manual

3. Security Problem Definition

STOP OS is not compliant with any currently active Protection Profile. However, many of the assumptions, threats and policies have been taken from the US Government Protection Profile for General-Purpose Operating Systems in a Networked Environment (GPOSPP). In addition are included, all of the threats and policies defined in the LSPP and RBAC PPs, both of which are archived.

3.1 Secure Usage Assumptions

Generally, secure usage assumptions are organized in three categories: personnel (assumptions about administrators and users of the system as well as any threat agents), physical (assumptions about the physical location of the TOE or any attached peripheral devices), and connectivity (assumptions about other IT systems that are necessary for the secure operation of the TOE). The TOE makes physical usage assumptions that are described below.

3.1.1 Physical Assumptions

The TOEs are intended for application in user areas that have physical control and monitoring. It is assumed that the following physical conditions will exist:

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

.

3.2 Organization Security Policies

An Organizational Security Policy is a set of rules or procedures imposed by an organization upon its operations to protect its sensitive data. Although some of the organizational security policies described below are drawn from the GPOSPP they also apply to many non-defense related environments.

P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.
P.ACCOUNTABILITY	The users of the TOE shall be held accountable for their actions within the TOE.
P.AUTHORIZATION	The TOE shall limit the extent of each user’s abilities in accordance with the TSP.
P.AUTHORIZED_USERS	Only those users who have been authorized to access the information within the TOE may access the TOE.
P.CRYPTOGRAPHY	The TOE shall use NIST FIPS validated cryptography as a baseline for key management (i.e., generation and destruction) and for cryptographic operations (i.e., encryption, decryption, signature, hashing, and random number generation services).
P.I_AND_A	All users must be identified and authenticated prior to accessing any controlled resources with the exception of public objects.
P.NEED_TO_KNOW	The TOE must limit the access to data in protected resources to those authorized users who have a need to know that data.
P.ROLES	The TOE shall provide multiple administrative roles for secure administration of the TOE. These roles shall be separate and distinct from each other.
P.TRACE	The TOE shall provide the ability to review the actions of individual users.
P.TRUSTED_RECOVERY	Procedures and/or mechanisms shall be provided to assure that, after a TOE failure or other discontinuity, recovery without a protection compromise is obtained.

P.CLASSIFICATION	The system must limit the access to information based on sensitivity, as represented by a label, of the information contained in objects, and the formal clearance of users, as represented by subjects, to access that information. The access rules enforced prevent a subject from accessing information which is of higher sensitivity than it is operating at and prevent a subject from causing information from being downgraded to a lower sensitivity. The method for classification of information is made based on criteria set forth by the organization. This is usually done on a basis of relative value to the organization and its interest to limit dissemination of that information. The determination of classification of information is outside the scope of the IT system; the IT system is only expected to enforce the classification rules, not determine classification. The method for determining clearances is also outside the scope of the IT system. It is essentially based on the trust placed in individual users by the organization. To some extent is also dependent upon the individual's role within the organization.

3.3 Threats

3.3.1 Threats to be addressed by the TOE

T.ADMIN_ERROR	An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
T.ADMIN_ROGUE	An authorized administrator's intentions may become malicious resulting in user or TSF data being compromised.
T.AUDIT_COMPROMISE	A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.
T.CRYPTO_COMPROMISE	A malicious user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the cryptographic mechanisms and the data protected by those mechanisms.
T.MASQUERADE	A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources.
T.OPERATIONAL_ERRORS	While the TOE is operational, changes to the TOE may cause it to enter a configuration that is not able to enforce the security policies of the TOE.
T.RESIDUAL_DATA	A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.

T.RESOURCE_EXHAUSTION	A malicious process or user may block others from system resources (i.e., persistent storage) via a resource exhaustion denial of service attack.
T.TSF_COMPROMISE	A malicious user or process may cause TSF data or executable code to be inappropriately accessed (viewed, modified or deleted).
T.UNATTENDED_SESSION	A user may gain unauthorized access to an unattended session.
T.UNAUTHORIZED_ACCESS	A user may gain unauthorized access (view, modify, delete) to user data.
T.UNIDENTIFIED_ACTIONS	The administrator may fail to notice potential security violations, thus preventing the administrator from taking action against a possible security violation.
T.UNKNOWN_STATE	When the TOE is initially started or restarted after a failure, the security state of the TOE may be unknown.
T.SPOOFING	Legitimate system services are spoofed. An attacker tricks users into interacting with spurious system services. This could allow disclosure of the users' data to the attacker and unauthorized modification of the users' data by the attacker.
T.REMOTEACCESS	An attacker may intercept communication between the TOE and another trusted IT product and be able to read and modify the data in an undetected manner.
T.UNRESTRICTED_TRAFFIC	Network traffic into and out of the TOE may be uncontrollable by the TSF.

3.3.2 Threats to be addressed by the Operational Environment

There are no threats identified to be addressed by the Operational Environment.

4. Security Objectives

4.1 Security Objectives for the TOE

The following are the IT security objectives that are met by the TOE.

O.ACCESS	The TOE will ensure that users gain only authorized access to it and to resources that it controls.
O.ACCESS_HISTORY	The TOE will display information (to authorized users) related to previous attempts to establish a session.
O.ADMIN_ROLE	The TOE will provide administrator roles to isolate administrative actions.

O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security relevant events associated with users.
O.AUDIT_PROTECTION	The TOE will provide the capability to protect audit information.
O.AUDIT_REVIEW	The TOE will provide the capability to selectively view audit information and alert the administrator of identified potential security violations.
O.CORRECT_TSF_OPERATION	The TOE will provide a capability to test the TSF to ensure the correct operation of the TSF in its operational environment.
O.CRYPTOGRAPHIC_SERVICES	The TOE will make cryptographic services available to authorized users and/or user applications.[8]
O.DISCRETIONARY_ACCESS	The TOE will control access to resources based upon the identity of users and groups of users.
O.DISCRETIONARY_USER_CONTROL	The TOE will allow authorized users to specify which resources may be accessed by which users and groups of users.
O.DISPLAY_BANNER	The TOE will display an advisory warning regarding use of the TOE.
O.MANAGE	The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
O.PROTECT	The TOE will provide mechanisms to protect user data and resources.
O.RECOVERY	Procedures and/or mechanisms will be provided to assure that recovery is obtained without a protection compromise, such as from system failure or discontinuity.
O.RESIDUAL_INFORMATION	The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.
O.RESOURCE_SHARING	The TOE shall provide mechanisms that mitigate user attempts to exhaust TOE resources (e.g., concurrent sessions).
O.REFERENCE_MONITOR	The TOE will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.
O.USER_AUTHENTICATION	The TOE will verify the claimed identity of users.
O.USER_IDENTIFICATION	The TOE will uniquely identify users.

[8] O.CRYPTOGRAPHIC_SERVICES is presented differently in sections 4 and 7 of the GPOSPP in that the section 4 GPOSPP statement only focuses on “encryption” while the section 7 statement refers to “cryptographic” services.. This ST presentation is consistent with the section 7 GPOSPP statement as it seems most accurate to address the associated policy P.CRYPTOGRAPHY.

O.TRUSTED_PATH	The TOE will provide a means to ensure local users are not communicating with some other entity pretending to be the operating system and will provide a secure channel between itself and remote users.
O.MANDATORY_ACCESS	The TSF must control access to resources based upon the sensitivity, integrity, and categories of the information being accessed and the clearance of the subject attempting to access that information.
O. RESTRICT_TRAFFIC	The TOE provides the capability to restrict inbound and outbound network traffic.

4.2 Security Objectives for the Operational Environment

The following are the security objectives for the operational environment which are taken from the GPOSPP.

OE.PHYSICAL	Physical security will be provided for the TOE by the IT environment, commensurate with the value of the IT assets protected by the TOE.

5. IT Security Requirements

The following sections define the security functional and assurance requirements for the TOE. These security functional requirements have been drawn largely from the GPOSPP as well as the following archived Protection Profiles: Labeled Security Protection Profile, Version 1.b, 8 October 1999; Role-Based Access Control Protection Profile, Version 1.0, July 30 1998. The security assurance requirements have been drawn from EAL 4, as defined in the CC Part 3, augmented with ALC_FLR.3.

5.1 Extended Components Definition

The extended components included in this Security Target are taken from the GPOSPP. The naming convention for extended components is the same as that used in the CC. To ensure these requirements are identified, the word “Extended:” appears before the component behavior name to alert the reader. Additionally, the ending "_EXT" is appended to the newly created short name and the component and the element names are bolded. However, most of the extended requirements are based on existing CC requirements.

Table 5-1 TOE Extended Functional Security Requirements

Name	Description
FCS_BCM_EXT.1	Baseline Cryptographic Module
FCS_COA_EXT.1	Cryptographic Operations Availability
FCS_RBG_EXT.1	Random Number Generation
FIA_AFL_EXT.1	Authentication Failure Handling
FPT_TST_EXT.1	TSF testing

5.1.1 Baseline Cryptographic Module (FCS_BCM)

The cryptographic requirements are structured to accommodate use of the FIPS 140-2 standard and NIST's Cryptomodule Validation Program (CMVP) in meeting the requirements. Note that FIPS-approved cryptographic functions are required to be implemented in a FIPS-validated module running in FIPS-approved mode. FCS_BCM reflects this requirement, and it specifies the required FIPS validation levels for the security functions.

The term "FIPS-approved cryptographic function" is used in the following requirements. A FIPS-approved cryptographic function is a security function (e.g., cryptographic algorithm, cryptographic key management technique, or authentication technique) that is either: 1) specified in a Federal Information Processing Standard (FIPS), or 2) adopted in a FIPS and specified either in an appendix to the FIPS or in a document referenced by the FIPS. This family is an addition to the families defined in the CC Part 2.

5.1.1.1 Extended: Baseline Cryptographic Module (FCS_BCM_EXT.1)

Dependencies: No dependencies.

See section 5.2.2.1 for more information.

5.1.2 Cryptographic Operation (FCS_COP)

Cryptographic keys must be managed throughout their life cycle. This family is intended to support that lifecycle and consequently defines requirements for the following activities: cryptographic key generation, cryptographic key distribution, cryptographic key access and cryptographic key destruction. This family should be included whenever there are functional requirements for the management of cryptographic keys. This family is found in the CC Part 2.

5.1.2.1 Extended: Cryptographic Operations Availability (FCS_COA_EXT.1)

Dependencies: FCS_BCM_EXT.1 Extended: Baseline cryptographic module

See section 5.2.2.6 for more information.

5.1.2.2 Extended: Random Number Generation (FCS_RBG_EXT.1)

Dependencies: FCS_BCM_EXT.1 Extended: Baseline cryptographic module

See section 5.2.2.10 for more information.

5.1.3 Authentication Failure Handling (FIA_AFL)

This family contains requirements for defining values for some number of unsuccessful authentication attempts and TSF actions in cases of authentication attempt failures. Parameters include, but are not limited to, the number of failed authentication attempts and time thresholds. This family is found in the CC Part 2.

5.1.3.1 Authentication Failure Handling (FIA_AFL_EXT.1)

Dependencies: FIA_UAU.1 Timing of authentication

See section 5.2.4.1 for more information.

5.1.4 TSF Self Test (FPT_TST)

The family defines the requirements for the self-testing of the TSF with respect to some expected correct operation. Examples are interfaces to enforcement functions, and sample arithmetical operations on critical parts of the TOE. These tests can be carried out at start-up, periodically, at the request of the authorised user, or when other conditions are met. The actions to be taken by the TOE as the result of self testing are defined in other families.

The requirements of this family are also needed to detect the corruption of TSF data and TSF itself (i.e. TSF executable code or TSF hardware component) by various failures that do not necessarily stop the TOE's operation (which would be handled by other families). These checks must be performed because these failures may not necessarily be prevented. Such failures can occur either because of unforeseen failure modes or associated oversights in the design of hardware, firmware, or software, or because of malicious corruption of the TSF due to inadequate logical and/or physical protection. This family is found in the CC Part 2.

5.1.4.1 Extended: TSF Testing (FPT_TST_EXT.1)

Dependencies: FCS_COP.1 Cryptographic operation

FCS_RBG_EXT.1 Random number generation

See section 5.2.6.5 for more information.

5.2 TOE Security Functional Requirements

This section specifies the security functional requirements that are applicable to the TOE.

Table 5-2 TOE Functional Security Requirements

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1: Audit data generation
	FAU_GEN.2: User identity association
	FAU_SAR.1: Audit review
	FAU_SAR.2: Restricted audit review
	FAU_SAR.3: Selectable audit review
	FAU_SEL.1: Selective audit
	FAU_STG.1 Permanent Audit Trail Storage
	FAU_STG.3: Action in case of possible audit data loss
	FAU_STG.4: Prevention of audit data loss
FDP: User data protection	FDP_ACC.2: Subset access control/DAC
	FDP_ACC.1: Subset access control/RBAC
	FDP_ACF.1a: Security attribute based access control/DAC
	FDP_ACF.1b: Security attribute based access control/RBAC
	FDP_ETC.1: Export of user data without security attributes
	FDP_ETC.2: Export of user data with security attributes
	FDP_IFC.1: Subset information flow control /firewall
	FDP_IFC.2a: Subset information flow control /MAC
	FDP_IFC.2b: Subset information flow control /MIC
	FDP_IFF.1: Simple security attributes /firewall
	FDP_IFF.2a: Hierarchical security attributes /MAC
	FDP_IFF.2b: Hierarchical security attributes /MIC
	FDP_ITC.1: Import of user data without security attributes
	FDP_ITC.2: Import of user data with security attributes
	FDP_RIP.2: Full residual information protection
FCS: Cryptographic Support	FCS_BCM_EXT.1: Extended: Baseline Cryptographic Module
	FCS_CKM.1a: Cryptographic Key Generation (for symmetric keys)
	FCS_CKM.1b: Cryptographic Key Generation (for asymmetric keys)
	FCS_CKM.2: Cryptographic key distribution
	FCS_CKM.4.1: Cryptographic Key Destruction

Requirement Class	Requirement Component
	FCS_COA_EXT.1: Extended: Cryptographic Operations Availability
	FCS_COP.1a: Cryptographic operation
	FCS_COP.1b: Cryptographic operation
	FCS_COP.1c: Cryptographic operation
	FCS_RBG_EXT.1: Extended: Random Number Generation
FIA: Identification and authentication	FIA_AFL_EXT.1: Authentication Failure Handling
	FIA_ATD.1: User attribute definition
	FIA_SOS.1: Verification of secrets
	FIA_UAU.2: User authentication before any action
	FIA_UAU.6.1: Re-authenticating
	FIA_UAU.7: Protected authentication feedback
	FIA_UID.2: User identification before any action
	FIA_USB.1: User-subject binding
FMT: Security Management	FMT_MOF.1a: Management of security functions behaviour
	FMT_MOF.1b: Management of security functions behaviour
	FMT_MSA.1a: Management of security attributes/DAC
	FMT_MSA.1b: Management of security attributes/DAC2
	FMT_MSA.1c: Management of security attributes/RBAC
	FMT_MSA.1d: Management of security attributes/MAC
	FMT_MSA.1e: Management of security attributes/MIC1
	FMT_MSA.1f: Management of security attributes/MIC2
	FMT_MSA.1g: Management of security attributes
	FMT_MSA.1h: Management of security attributes/firewall
	FMT_MSA.2: Secure Security Attributes
	FMT_MSA.3a: Static attribute initialization
	FMT_MSA.3b: Static attribute initialization
	FMT_MTD.1a: Management of TSF data
	FMT_MTD.1b: Management of TSF data
	FMT_MTD.1c: Management of TSF data
	FMT_MTD.1d: Management of TSF data
	FMT_MTD.1e: Management of TSF data
	FMT_MTD.1f: Management of TSF data
	FMT_MTD.1g: Management of TSF data
	FMT_MTD.3: Secure TSF Data
	FMT_REV.1a: Revocation/Subjects
	FMT_REV.1b: Revocation/Objects
	FMT_SAE.1: Time-limited authorization
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.2: Security management roles
FPT: Protection of the TSF	FPT_FLS.1: Failure with preservation of secure state
	FPT_RCV.1: Manual recovery
	FPT_RCV.4: Function recovery
	FPT_STM.1: Reliable time stamps
	FPT_TST_EXT.1: TSF testing
FTA: TOE access	FTA_LSA.1: Limitation on scope of selectable attributes
	FTA_MCS.1: Basic limitation on multiple concurrent sessions
	FTA_TAB.1: Default TOE access banners
	FTA_TAH.1: TOE Access History
	FTA_TSE.1: TOE session establishment
FTP: Trusted Path / Channels	FTP_ITC.1: Inter-TSF trusted channel
	FTP_TRP.1: Trusted path

5.2.1 Security audit (FAU)

5.2.1.1 Audit data generation (FAU_GEN.1)

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

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

b) Start-up and shutdown of the TOE,

c) Uses of special permissions that circumvent the access control policies,

d) All auditable events listed in Table 6-1, and

e) All **other security relevant** auditable events for the [***minimal level***] of audit.

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 the outcome (success or failure) of the event; and

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

[(i) For each invocation of a security function, the RBAC Administrator role that made invocation of that security function possible.
(ii) For each access control action on the user data, the role that made possible the invocation of that action.
(iii) the additional details identified in Table 6-1 Auditable Events].

See Table 6-1 Auditable Events for the TOE in section 6 (TOE Summary Specification) for a list and description of all auditable events

5.2.1.2 User identity association (FAU_GEN.2)

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

5.2.1.3 Audit review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide [**authorized administrators and users authorized by role assignment**] with the capability to read [**all audit information**] from the audit records**.**

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the **authorized administrator** to interpret the information using a tool to access the audit records.

5.2.1.4 Restricted audit review (FAU_SAR.2)

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

5.2.1.5 Selectable audit review (FAU_SAR.3)

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

a) user identity,

b) object identity,

c) date of the event,

d) time of the event,

e) **type of event,**

f) **success of auditable security events,**

g) **failure of auditable security events, and**

h) **subject sensitivity label**

i) **object sensitivity label**

j) **object name & type of access**

k) **role that enabled the access**

l) **any combination of date and time of event; user identity; object name & type of access; and the role that enabled the access**]

5.2.1.6 Selective audit (FAU_SEL.1)

FAU_SEL.1.1 The TSF shall be able to ~~select the set of events to be audited~~ **include or exclude auditable events** from the set of audited events based on the following attributes: [9] [

a) object identity
b) user identity;
c) event type;]
d) **[success of auditable security events;**
e) **failure of auditable security events;**
f) **Subject sensitivity and integrity label (subject MAC and MIC label);**
g) **Object sensitivity and integrity label (object MAC and MIC label);**
h) **Subject identity; and**
i) **Users belonging to a specified Role and Access types (e.g. delete, insert) on a particular object.**]

5.2.1.7 Permanent Audit Trail Storage (FAU_STG.1)

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.
FAU_STG.1.2 The TSF shall be able to [***prevent***] unauthorized modifications to the stored audit records in the audit trail.

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

FAU_STG.3.1 The TSF shall [**notify an authorized administrator of the possible audit data loss**] if the audit trail exceeds [**an authorized administrator selectable, pre-defined limit**].

5.2.1.9 Prevention of audit data loss (FAU_STG.4)

FAU_STG.4.1 The TSF shall [***prevent audited events, except those taken by the authorised user with special rights***] and **[cause the system to shut down]** if the audit trail is full.

5.2.2 Cryptographic Support (FCS)

5.2.2.1 Extended: Baseline Cryptographic Module (FCS_BCM_EXT.1)

[9] Attributes that pertain to distributed TOEs such as “host identifier” are not included even though the RBACPP and GPOSPP had specified them.

FCS_BCM_EXT.1.1 All FIPS-approved cryptographic functions implemented by the TOE shall be implemented in a cryptomodule that is FIPS 140-2 validated, and perform the specified cryptographic functions in a FIPS-approved mode of operation. The FIPS 140-2 validation shall include an algorithm validation certificate for all FIPS-approved cryptographic functions invoked by the TOE.

5.2.2.2 Cryptographic Key Generation (for symmetric keys) (FCS_CKM.1a)

FCS_CKM.1.1a The TSF shall generate cryptographic keys ~~in accordance with a specified cryptographic key generation algorithm~~ [**using a FIPS-Approved Random Number Generator as specified in FCS_RBG_EXT.1, and provide integrity protection to generated keys that leave the cryptomodule in accordance with NIST SP 800-57 "Recommendation for Key Management—Part 1: General," paragraph 6.2.2.2a.**] ~~and specified cryptographic key sizes that meet the following~~ **in the following manner: [SHA256 MAC].**

5.2.2.3 Cryptographic Key Generation (for asymmetric keys) (FCS_CKM.1b)

FCS_CKM.1.1b The TSF shall generate **asymmetric** cryptographic keys in accordance with ~~a specified cryptographic key generation algorithm [assignment: cryptographic key generation algorithim] and specified cryptographic key sizes~~ **domain parameter sizes** [**for rDSA-based keys, *[1024 bits*],**] **t**hat meet the following: [**FIPS 140-2**].

5.2.2.4 Cryptographic key distribution (FCS_CKM.2)

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [**diffie-hellman**] that meets the following: **[SSH (RFC 4252) and Diffie-Hellman Key Agreement Method (RFC 2631)]**.

5.2.2.5 Cryptographic Key Destruction (FCS_CKM.4)

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified [**cryptographic key zeroization method**] that meets the following: [**Key zeroization requirements of FIPS PUB 140-2, "Security Requirements for Cryptographic Modules"**].

5.2.2.6 Extended: Cryptographic Operations Availability (FCS_COA_EXT.1)

FCS_COA_EXT.1 The TSF shall provide the following cryptographic operations to applications:

a) Encryption/Decryption,

b) Cryptographic Signature (Digital Signature),

c) Hashing, and

d) [none].

5.2.2.7 Cryptographic operation (for encryption/decryption) (FCS_COP.1a)

FCS_COP.1.1a The TSF shall perform [**encryption and decryption**] ~~in accordance with a specified cryptographic algorithm~~ **using the FIPS-approved security function [AES algorithm] operating in [ECB, CBC, and CFB modes]** and cryptographic key size **of [*128, 192, and 256 bits*]** that meets the following: [**FIPS 140-2**].

5.2.2.8 Cryptographic operation (for cryptographic signature) (FCS_COP.1b)

FCS_COP.1.1b The TSF shall perform [**cryptographic signature services**] ~~in accordance with a specified cryptographic algorithm~~ **using the FIPS-approved security function [*RSA Digital Signature Algorithm (rDSA)*** ~~and cryptographic key sizes~~ **with a key size (modulus) of [*1024 bits or greater*]]** that meet the following~~:~~ [**FIPS 140-2**].

5.2.2.9 Cryptographic Operation (for cryptographic hashing) (FCS_COP.1c)

FCS_COP.1.1c The TSF shall perform [**cryptographic hashing services**] in accordance with ~~a specified cryptographic algorithm~~ **[*SHA 256, SHA 384, SHA 512*]** ~~and cryptographic key sizes~~ **message digest sizes** [***256, 384, or 512 bits***] that meet the following: [**FIPS 140-2**].

5.2.2.10 Extended: Random Number Generation (FCS_RBG_EXT.1)

FCS_RBG_EXT.1.1 The TSF shall perform all random bit generation (RBG) services in accordance with[***FIPS Pub 140-2 Annex C***] implemented in a FIPS-validated cryptomodule operating in FIPS mode seeded by an entropy source that accumulates entropy from [***one or more independent software-based noise sources].***

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded with a minimum of **[*256 bits*]** of entropy at least equal to the greatest bit length of the keys that it will generate.

5.2.3 User data protection (FDP)

5.2.3.1 Complete access control/DAC (FDP_ACC.2)

FDP_ACC.2.1 The TSF shall enforce the [**Discretionary Access Control policy**] on [**all subjects and all named objects and all operations among them**].

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.

5.2.3.2 Subset access control/RBAC (FDP_ACC.1)

FDP_ACC.1.1 The TSF shall enforce the **[Role-based Access Control (RBAC) policy]** on **[**
subjects: processes;
objects: processes, devices, semaphores, sockets, file system objects (files, directories, device special files, symbolic links, and named First-In First-Outs (FIFOs), and memory objects (shared memory and unnamed pipes)
and all operations among subjects and objects covered by the RBAC policy].

5.2.3.3 Security attribute based access control/DAC (FDP_ACF.1a)

FDP_ACF.1.1a The TSF shall enforce the [**Discretionary Access Control Policy**] to **named** objects based on the following **types of subject and object security attributes**:

a) [**the authorized user identity and group membership(s) associated with a subject, and**

b) **the authorized user (or group) identity, access operations pairs associated with a named object.**]

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

[**The Discretionary Access Control policy mechanism shall, either by explicit authorized user action or by default, provide that named objects are protected from unauthorized access according to the following ordered rules:**

1) **If the requested mode of access is denied to that authorized user, deny access.**
2) **If the requested mode of access is permitted to that authorized user, permit access.**
3) **If the requested mode of access is denied to the ~~every~~ group of which the authorized user is a member, deny access**
4) **If the requested mode of access is permitted to the ~~any~~ group of which the authorized user is a member, grant access**
5) **If the requested mode of access is permitted for the world (i.e. all user identities), grant access**
6) **Else deny access.**]

FDP_ACF.1.3a The TSF shall explicitly authorize access of subjects to **named** objects based on the following additional rules:

a) [**Authorized administrators must follow the above-stated Discretionary Access Control policy, except after taking the following specific actions: [assigns a role allowing an exemption that bypasses the above stated DAC policy to the subject requesting access and the subject requests access]**

:

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

5.2.3.4 Security attribute based access control/RBAC (FDP_ACF.1b)

FDP_ACF.1.1b The TSF shall enforce the **[Role-based Access Control Policy]** to objects based on the following: **[Subject Security Attributes**

a) Subject Identity
b) Role(s)

Object Security Attributes:

c) Object Identity
d) Role(s)

FDP_ACF.1.2b The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[The subject invoking the operation on an object is assigned to a role whose action set includes the operation on the object and that role is also associated with the object].**

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

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

5.2.3.5 Export of user data without security attributes (FDP_ETC.1)

FDP_ETC.1.1 The TSF shall enforce the [**MAC, MIC, DAC, and RBAC policies]** when exporting user data, controlled under the SFP(s), outside of the TOE.

FDP_ETC.1.2 The TSF shall export the **unlabeled** user data without the user data's associated security attributes **and shall enforce the following rules when unlabeled user data is exported from the TSC:**

a) **Devices used to export data without security attributes cannot be used to export data with security attributes unless the change in device state is performed manually and is auditable;**
b) **The devices used for export do have MAC, MIC, DAC, and RBAC attributes and these are implicitly the attributes of the exported information;**
c) **Normal MAC, MIC, DAC, and RBAC policy rules govern access from subjects to any export device and such access will be auditable;**
d) **Only an administrator can modify the MAC, MIC, or RBAC attributes of a device and only the owner of a device can modify the device's DAC attributes;**
e) **All changes to device security attributes are auditable;**

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

FDP_ETC.2.1 The TSF shall enforce the **[MAC, MIC, DAC, and RBAC policies]** when exporting **labeled** user data, controlled under the SFP(s), outside of the TOE.

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

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

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

a) **If a device is capable of maintaining data security attributes, and the device is being managed by the TSF, the security attributes shall be exported with the data and the device shall completely and unambiguously associate the security attributes with the corresponding data.**
b) **Devices used to export data with security attributes cannot be used to export data without security attributes unless the change in device state is performed manually and is auditable**
c) **The security attributes of exported data shall include MAC, MIC, DAC and RBAC security attributes. This applies to mounted file systems and tape backups.**

5.2.3.7 Subset Information flow control/Firewall (FDP_IFC.1)

FDP_IFC.1.1 The TSF shall enforce the **[Host-Based Packet Filter Firewall Policy]** on **[**

Subjects:

The TOE itself that sends/receives information to/from another product

Information:

IP network traffic

Operations:
send or receive traffic].

5.2.3.8 Complete Information flow control/MAC (FDP_IFC.2a)

FDP_IFC.2.1a The TSF shall enforce the **[Mandatory Access Control policy]** on **[**
all subjects:
processes
and all objects:
processes, devices, semaphores, sockets, file system objects (files, directories, device special files, symbolic links, and named First-In First-Outs (FIFOs), and memory objects (shared memory and unnamed pipes)]
and all operations that cause that information to flow to and from subjects covered by the SFP.

FDP_IFC.2.2a 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.

5.2.3.9 Complete Information flow control/MIC (FDP_IFC.2b)

FDP_IFC.2.1b The TSF shall enforce the **[Mandatory Integrity Control policy]** on **[**
all subjects:
processes
and all objects:
processes, devices, semaphores, sockets, file system objects (files, directories, device special files, symbolic links, and named First-In First-Outs (FIFOs), and memory objects (shared memory and unnamed pipes)]
and all operations that cause that information to flow to and from subjects covered by the SFP.

FDP_IFC.2.2b 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.

5.2.3.10 Simple security attributes/Firewall (FDP_IFF.1)

FDP_IFF.1.1 The TSF shall enforce the **[Host-Based Packet Filter Firewall Policy]** based on the following types of subject and information security attributes: **[**

a) **subject (The TOE itself that sends/receives information to/from another product)**
security attribute:
Filter Rules: define if the TOE can send or receive IP network traffic from another product based on any combination of the information security attributes

b) **information (IP network traffic)**
security attributes:
source IP address, destination IP address, source port, destination port and protocol] .

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 filter rules must allow for the information flow (IP network traffic to either be sent by the TOE or received by the TOE)].**

FDP_IFF.1.3 The TSF shall enforce the **[no additional rules].**

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

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

5.2.3.11 Hierarchical security attributes/MAC (FDP_IFF.2a)

FDP_IFF.2.1a The TSF shall enforce the [**MAC SFP**] based on the following types of subject and information security attributes: [

Subject Security Attributes

a) a sensitivity label (current MAC label), consisting of at least 256 site definable hierarchical levels and a set of 64 site definable nonhierarchical categories;

b) the user's clearance which consists of a minimum, maximum and default MAC label (which are also a sensitivity labels respectively)

Information Security Attributes

c) a sensitivity label (current MAC label), consisting of at least 256 site definable hierarchical levels and a set of 64 site definable nonhierarchical categories.

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

a) If the sensitivity label of the subject is greater than (see FDP_IFF.2.6a) 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);

Application Note: where 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).

c) If the information flow is between objects, the sensitivity label of the destination object must be greater than (see FDP_IFF.2.6a) or equal to the sensitivity label of the source object].

FDP_IFF.2.3a The TSF shall enforce the **[**

a) subjects can not operate at mandatory labels above, create objects at mandatory labels above, or change the mandatory label of an object to a label above, the clearance of the owning user;

b) subjects can not operate at mandatory labels below, create objects at mandatory labels below, or change the mandatory label of an object to a label below, the clearance of the owning user;

c) writes to file system and device objects must be at the label of the object (write-up is not allowed)]

FDP_IFF.2.4a The TSF shall explicitly authorize an information flow based on the following rules: **[**

a) each user must be assigned a clearance by an authorized administrator where the clearance is within the mandatory range allowed by the system;

b) **Authorized users may bypass MAC if they are assigned a role that includes an exemption to bypass the MAC policy. Some commands may also require a user to be associated with a role by the system administrator.**

].

FDP_IFF.2.5a The TSF shall explicitly deny an information flow based on the following rules: **[**

a) **a user can not upgrade nor downgrade the mandatory label of an object, unless s/he has been associated with a role that permits this change by an authorized administrator].**

FDP_IFF.2.6a 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

1) **Sensitivity labels are equal if the hierarchical level of both labels are equal and the non-hierarchical category sets are equal;**

2) **Sensitivity label A is greater than sensitivity label B if one of the following conditions exist:**

I. **If the hierarchical level of A is greater than the hierarchical level of B, and the non-hierarchical category set of A is equal to the non-hierarchical category set of B.**

II. **If the hierarchical level of A is equal to the hierarchical level of B, and the non hierarchical category set of A is a proper superset of the nonhierarchical category set of B.**

III. **If the hierarchical level of A is greater than the hierarchical level of B, and the non-hierarchical category set of A is a proper super-set of the non-hierarchical category set of B.**

3) **Sensitivity labels are incomparable if they are not equal and neither label is greater than the other.**

b) There exists a "least upper bound" in the set of 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.

5.2.3.12 Hierarchical security attributes/MIC (FDP_IFF.2b)

FDP_IFF.2.1b The TSF shall enforce the [**Mandatory Integrity Control (MIC) SFP**] based on the following types of subject and information security attributes: **[**

Subject Security Attributes

a) **an integrity label (current MIC label) containing at least 256 site-definable hierarchical levels and at least 32 non-hierarchical categories;**

b) **the user's clearance which consists of a minimum, maximum and default MIC label (which is also an integrity label);**

Information Security Attributes

c) **an integrity label (current MIC label) containing at least 256 site-definable hierarchical levels and at least 32 non-hierarchical categories].**

FDP_IFF.2.2b 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 a write (the flow of information from the subject to the object) is permitted;

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

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

FDP_IFF.2.3b The TSF shall enforce the **[**

a) subjects can not operate at integrity labels below, create objects at integrity labels below, or change the integrity label of an object to a label below, the clearance of the owning user;

b) subjects can not operate at integrity labels above, create objects at integrity labels above, or change the integrity label of an object to a label above, the clearance of the owning user;

c) writes to file system and device objects must be at the label of the object (write-down is not allowed);

d) each user must be assigned a clearance by an authorized administrator where the clearance is within the integrity range allowed by the system;

e) user can not upgrade nor downgrade the integrity label of an object, unless s/he has been associated with a role that permits this change by an authorized administrator].

FDP_IFF.2.4b The TSF shall explicitly authorize an information flow based on the following rules: **[**

a) Authorized users may bypass MIC if they are assigned a role that includes an exemption to bypass the MIC policy. Some commands may also require a user to be associated with a specific role by the system administrator].

FDP_IFF.2.5b The TSF shall explicitly deny an information flow based on the following rules: **[none]**

FDP_IFF.2.6b 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

1) Integrity labels are equal if the hierarchical level of both labels are equal and the non-hierarchical category sets are equal;

2) Integrity label A is greater than integrity label B if one of the following conditions exist:

I. If the hierarchical level of A is greater than the hierarchical level of B, and the non-hierarchical category set of A is equal to the non-hierarchical category set of B.

II. If the hierarchical level of A is equal to the hierarchical level of B, and the non-hierarchical category set of A is a proper superset of the nonhierarchical category set of B.

III. If the hierarchical level of A is greater than the hierarchical level of B, and the non hierarchical category set of A is a proper super-set of the non-hierarchical category set of B.

3) Integrity labels are incomparable if they are not equal and neither label is greater than the other.

b) There exists a "least upper bound" in the set of 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.

5.2.3.13 Import of user data without security attributes (FDP_ITC.1)

FDP_ITC.1.1 The TSF shall enforce the **[MAC, MIC, DAC and RBAC policies]** when importing **unlabeled** user data, controlled under the SFP, from outside of the TOE.

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

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

a) The TSF shall label the data with the MAC and MIC labels of the device by which the data is imported;

b) When importing data, the data is to be given the effective owner, group, and role attributes of the importer of the data;

c) Devices used to import data without security attributes cannot be used to import data with security attributes unless the change in device state is performed manually and is auditable].

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

FDP_ITC.2.1 The TSF shall enforce **the [MAC, MIC, DAC and RBAC policies]** when importing **labeled** 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 **labeled** user data.

FDP_ITC.2.3 The TSF shall ensure that the protocol used provides for the **correct** unambiguous association between the **imported** security attributes and the **imported, labeled** 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 **labeled** user data.

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

a) Devices used to import data with security attributes cannot be used to import data without security attributes unless the change in device state is performed manually and is auditable;

b) Sensitivity and integrity labels imported consist of the following:

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

5.2.3.15 Full residual information protection (FDP_RIP.2)[10]

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.

[10] Subjects (processes) are also objects in the TOE (see the Access Control security function description) and are, therefore, subject to this requirement. Therefore, this SFR is equivalent to the LSPP "Note 1" SFR.

5.2.4 Identification and authentication (FIA)

5.2.4.1 Authentication Failure Handling (FIA_AFL_EXT.1)

FIA_AFL_EXT.1.1 The TSF shall detect when an authorized administrator configurable positive integer of consecutive unsuccessful authentication attempts occur related to any authorized user authentication process.

FIA_AFL_EXT.1.2 When the defined number of consecutive unsuccessful authentication attempts has been met or surpassed, the TSF shall:

a) For all administrator accounts, "disable" the account for an authorized administrator configurable time period such that there can be no more than ten attempts per minute.

b) For all other accounts, disable the user logon account until it is re-enabled by the authorized administrator.

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

5.2.4.2 User attribute definition (FIA_ATD.1)

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

a) [**unique identifier,**

b) **group ~~memberships~~ identifier**

c) **authentication data,**

d) **security-relevant roles (see FMT_SMR.2),**

e) **[public key used for SSH**

f) **The clearance of the user including:**

a Minimum label
a Default label (including a Default MAC and Default MIC label)
a Maximum label]]

5.2.4.3 Verification of secrets (FIA_SOS.1)

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [**the following:**

a) Passwords are at least 16 characters in length, consisting of any combination of upper and lower case letters, numbers, and symbols, and

b) **Passwords are not reused within the last administrator-settable number of passwords used by that user.**]

5.2.4.4 User authentication before any action (FIA_UAU.2)

FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated **(i.e., an exact match between the internal representation of the user's entered data and the stored TSF authentication data)** before allowing any other TSF-mediated actions on behalf of that user.

5.2.4.5 Re-authenticating (FIA_UAU.6)

FIA_UAU.6.1 The TSF shall re-authenticate the user ~~under the conditions~~ **when** [changing authentication data].

5.2.4.6 Protected authentication feedback (FIA_UAU.7)

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

5.2.4.7 User identification before any action (FIA_UID.2)

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.

5.2.4.8 User-subject binding (FIA_USB.1)

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: **The security attributed identified in FIA_ATD.1a, b, d**, **and [e and f]**

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

a) [**For administrative users, provide restrictive defaults for security attributes identified in FIA_ATD.1, and**

b) **Restrict the ability to specify alternative initial user security attributes (that override the default attributes) to authorized administrators, and**

c) **The sensitivity label associated with a subject shall be within the clearance range of the user;**
d) **The integrity label associated with a subject shall be within the clearance range of the user]**

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: **[**

a) [**Restrict the ability to change user security attributes to authorized administrators, and**

b) **User security attribute changes shall take effect at next user logon, and**

c) **Only subjects associated with roles that have special role-actions associated with them can change any component of their label within the allowed range;**

d) **Only subjects associated with roles that have special role-actions associated with them can change their user ID;**

e) **Only subjects that were started as set-auth programs can change their user ID to the real and effective user ID for the subject;**

f) **All user ID changes are auditable;**

g) **Only administrators can change the clearance, role(s), and group identifier of a user.]**

5.2.5 Security management (FMT)

5.2.5.1 Management of Security Functions Behavior (for specification of auditable events) (FMT_MOF.1a)

FMT_MOF.1.1a The TSF shall restrict the ability to [***disable and enable***] the ~~functions~~ [**audit functions and to specify which events are to be audited (see FAU_SEL.1.1)]** to [**the authorized administrators and privileged users.]**

5.2.5.2 Management of Security Functions Behavior (for authentication data) (FMT_MOF.1b)

FMT_MOF.1.1b The TSF shall restrict the ability to [~~modify the behavior of~~ ***manage the values of***[11] **[security attributes associated with user authentication data]** to [**authorized administrators**].

5.2.5.3 Management of security attributes/DAC (FMT_MSA.1a)

FMT_MSA.1.1a The TSF shall enforce the [**Discretionary Access Control policy**] to restrict the ability to [***[change the value of]***] ~~the~~ **object** security attributes to [**authorized administrators and owners of the object**]

5.2.5.4 Management of security attributes/DAC (FMT_MSA.1b)

FMT_MSA.1.1b The TSF shall enforce the [**Discretionary Access Control policy**] to restrict the ability to [***[change object ownership]***] to [**authorized administrators**]

5.2.5.5 Management of security attributes/RBAC (FMT_MSA.1c)

FMT_MSA.1.1c The TSF shall enforce the **[RBAC Policy]** to restrict the ability to **[*modify* *and* *[create]*]** the security attributes **[**

a) Role Definitions & Role Attributes (Role definitions & Role assignments)
b) Role Hierarchies (by assigning one or more roles to other roles)
c) Constraints among Role Relationships
d) roles associated with a user and roles associated with a protected object]

to **[authorized administrators and users in roles defined to allow this].**

5.2.5.6 Management of security attributes/MAC (FMT_MSA.1d)

FMT_MSA.1.1d The TSF shall enforce the [**Mandatory Access Control policy]** to restrict the ability to [***modify***] the security attributes **[value of the sensitivity label associated with an object (see FDP_IFF.2.1a)]** to [**authorized administrators or users associated with the appropriate role].**

5.2.5.7 Management of security attributes/MIC (FMT_MSA.1e)

FMT_MSA.1.1e The TSF shall enforce the **[Mandatory Integrity Control policy]** to restrict the ability to **[*modify, delete, and [add]*]** the security attributes **[set of users with read access rights to the audit records (see FAU_SAR.1.1)]** to **[authorized administrators].**

[11] The word “manage” includes but is not limited to create, initialize, change default, modify, delete, clear, append, and query. The security attributes associated with user authentication data referenced by this requirement include those that are specified by FIA_AFL and FIA_SOS.

5.2.5.8 Management of security attributes/MIC2 (FMT_MSA.1f)

FMT_MSA.1.1f The TSF shall enforce the [**Mandatory Integrity Control policy]** to restrict the ability to [***modify***] the security attributes **[value of the integrity label associated with an object (see FDP_IFF.2.1 b)]** to **[authorized administrators users associated with the appropriate role(s)].**

5.2.5.9 Management of security attributes (FMT_MSA.1g)

FMT_MSA.1.1g The TSF shall enforce the **[DAC, MAC, MIC, and RBAC policies]** to restrict the ability to **[*change_default]*** the security attributes [**default user security attributes including clearance, group, and MAC label]** to **[authorized administrators and users associated with the appropriate role(s)].**

5.2.5.10 Management of security attributes (FMT_MSA.1h)

FMT_MSA.1.1h The TSF shall enforce the **[Host-Based Packet Filter Firewall Policy]** to restrict the ability to **[*modify]*** the security attributes **[Filter Rules]** to **[authorized administrators and users associated with the appropriate role(s)].**

5.2.5.11 Secure Security Attributes (FMT_MSA.2)

FMT_MSA.2.1 The TSF shall ensure that only valid values are accepted for [**all security attributes**].

5.2.5.12 Static attribute initialization (FMT_MSA.3a)

FMT_MSA.3.1a The TSF shall enforce the [**DAC, MAC, MIC, and RBAC policies]** to provide [***restrictive***] default values for security attributes that are used to enforce the SFPs.

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

5.2.5.13 Static attribute initialization (FMT_MSA.3b)

FMT_MSA.3.1b The TSF shall enforce the **[Host-Based Packet Filter Firewall Policy]** to provide **[*permissive*]** default values for security attributes that are used to enforce the SFPs.

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

5.2.5.14 Management of TSF data (FMT_MTD.1a)

FMT_MTD.1.1a The TSF shall restrict the ability to [***[manage]***] the [**TSF data except for audit records, user security attributes, authentication data, and critical cryptographic security parameters**] to [**authorized administrators and privileged users].**

5.2.5.15 Management of TSF data (FMT_MTD.1b)

FMT_MTD.1.1b The TSF shall restrict the ability to [***query, delete and clear***] the [**audit records**] to [**authorized administrators and privileged users].**

5.2.5.16 Management of TSF data (FMT_MTD.1c)

FMT_MTD.1.1c The TSF shall restrict the ability to [***[initialize]***] the [**user security attributes**] to [**authorized administrators and privileged users]**.

5.2.5.17 Management of TSF data (FMT_MTD.1d)

FMT_MTD.1.1d The TSF shall restrict the ability to [***modify***] the [**user security attributes, other than authentication data,**] to [**authorized administrators, and privileged users].**

5.2.5.18 Management of TSF data (FMT_MTD.1e)

FMT_MTD.1.1e The TSF shall restrict the ability to [***modify***] the [**authentication data**] to [**authorized administrators and users modifying their own authentication data**].

5.2.5.19 Management of TSF data (FMT_MTD.1f)

FMT_MTD.1.1f The TSF shall [***[prevent]***] **the [reading of] authentication data**

5.2.5.20 Management of TSF data (FMT_MTD.1g)

FMT_MTD.1.1g The TSF shall restrict the ability to [***[manage]***] the [**critical cryptographic security parameters and data related to cryptographic configuration**] to [**authorized administrators and privileged users].**

5.2.5.21 Secure TSF Data (FMT_MTD.3)

FMT_MTD.3.1 The TSF shall ensure that only secure values are accepted for **[TSF data].**

5.2.5.22 Revocation/Subjects (FMT_REV.1a)

FMT_REV.1.1a The TSF shall restrict the ability to revoke [**security attributes**] associated with the [***users***] under the control of the TSF to [**authorized administrators**].

FMT_REV.1.2a The TSF shall enforce the ~~rules~~ [**revocation of security-relevant authorizations at the next logon**].

5.2.5.23 Revocation/Objects (FMT_REV.1b)

FMT_REV.1.1b The TSF shall restrict the ability to revoke [**security attributes**] ~~associated with the~~ **of** [***[named objects]***] to [**owners of the named object and authorized administrators**].

FMT_REV.1.2b The TSF shall enforce the ~~rules~~ [**revocation of access rights associated with named objects when an access check is made**]

5.2.5.24 Time-limited authorization (FMT_SAE.1)

FMT_SAE.1.1 The TSF shall restrict the capability to specify an expiration time for [**authorized user authentication data**] to [**the authorized administrator**].

FMT_SAE.1.2 ~~For each of these security attributes, t~~The TSF shall be able to [**force the associated authorized user to change their authentication information prior to being able to successfully log on**] after the expiration time ~~for the indicated security attribute~~ has passed

5.2.5.25 Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [**all security management functions identified in other sections of this ST].**

5.2.5.26 Security management roles (FMT_SMR.2)

FMT_SMR.2.1 The TSF shall maintain the roles:

a) **[authorized administrator; and**
b) **Privileged Users (users authorized by their role definition to perform privileged actions); and**
c) **Object owners].**

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
b) The set of RBAC Administrative Roles can modify security attributes for all objects under the control of the TOE.
] are satisfied.

5.2.6 Protection of the TSF (FPT)

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

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

5.2.6.2 Manual recovery (FPT_RCV.1)

FPT_RCV.1.1 After [**a failure or service discontinuity that may lead to a violation of the TSP,**] the TSF shall enter a maintenance mode where the ability to return **the TOE** to a secure state is provided.

5.2.6.3 Reliable time stamps (FPT_STM.1)

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

5.2.6.4 Function recovery (FPT_RCV.4)

FPT_RCV.4.1 The TSF shall ensure that **[the following security functions and failure scenarios:**

a) **The SF 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 SF 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.

5.2.6.5 TSF testing (FPT_TST_EXT.1)

FPT_TST_EXT.1.1 The TSF shall run a suite of self tests in accordance with FIPS PUB 140-2, during initial start-up (on power on) to demonstrate the correct operation of the cryptographic modules**.**

FPT_TST_EXT.1.2 The TSF shall provide authorized administrators with the capability to verify the integrity of stored TSF executable code through the use of the TSF-provided cryptographic services.

5.2.7 TOE Access (FTA)

5.2.7.1 Limitation on scope of selectable attributes (FTA_LSA.1)

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**].

5.2.7.2 Basic limitation on multiple concurrent sessions (FTA_MCS.1)

FTA_MCS.1.1 The TSF shall ~~restrict~~ **enforce** a maximum number of concurrent **interactive** sessions ~~that belong to the same~~ per user.
FTA_MCS.1.2 The TSF shall ~~enforce by default, a limit of~~ **allow an authorized administrator to set the maximum number of concurrent interactive** sessions per user.

5.2.7.3 Default TOE access banners (FTA_TAB.1)

FTA_TAB.1.1 Before establishing a user session, the TSF shall display an authorized-administrator specified advisory notice and consent warning message regarding unauthorized use of the TOE.

5.2.7.4 TOE Access History (FTA_TAH.1)

FTA_TAH.1.1 Upon successful interactive session establishment, the TSF shall display to the authorized user the date and time of that authorized user's last successful interactive session establishment.

FTA_TAH.1.2 Upon successful interactive session establishment, the TSF shall display to the authorized user the date and time of the last unsuccessful attempt and the number of unsuccessful attempts at interactive session establishment for that user identifier since the last successful interactive session establishment**.**

FTA_TAH.1.3 The TSF shall not erase the access history information from the authorized user interface without giving the authorized user the opportunity to review the information.

5.2.7.5 TOE Session Establishment (FTA_TSE.1)

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].**

5.2.8 Trusted path/channels (FTP)

5.2.8.1 Inter-TSF trusted channel (FTP_ITC.1)

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

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

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for **[remote access using SSH]**.

5.2.8.2 Trusted path (FTP_TRP.1)

FTP_TRP.1.1 The TSF shall provide a communication path between itself and ***[local*]** users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [***modification and disclosure***].

FTP_TRP.1.2 The TSF shall permit [***local users***] to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for **[*initial local user authentication*].**

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the Evaluation Assurance Level 4 (EAL 4) components as specified in Part 3 of the Common Criteria, augmented with ALC_FLR.3 as indicated in bold the following table. No operations are applied to the assurance components.

Assurance Class	Assurance Component	Component Description
Development	ADV_ARC.1	Security architecture description
	ADV_FSP.4	Complete functional specification
	ADV_IMP.1	Implementation representation of the TSF
	ADV_TDS.3	Basic modular design
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
Lifecycle support	ALC_CMC.4	Production support, acceptance procedures and automation
	ALC_CMS.4	Problem tracking CM coverage
	ALC_DEL.1	Delivery procedures
	ALC_DVS.1	Identification of security measures

Assurance Class	Assurance Component	Component Description
	ALC_FLR.3	Systematic flaw remediation
	ALC_LCD.1	Developer defined life-cycle model
	ALC_TAT.1	Well-defined development tools
Security Target evaluation	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.2	Security objectives
	ASE_REQ.2	Derived security requirements
	ASE_SPD.1	Security problem definition
	ASE_TSS.1	TOE summary specification
Tests	ATE_COV.2	Analysis of coverage
	ATE_DPT.1	Testing: basic design
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing – sample
Vulnerability Assessment	AVA_VAN.3	Focused vulnerability analysis

6. TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

6.1 TOE Security Functions

6.1.1 Security Audit

STOP OS audit functionality is a kernel based logging system called System LOGger (slog). Slog assesses whether or not the record should be incorporated into the current slog file based on filter rules. If the record is to be incorporated, slog populates slog records into a plain text file. The list of auditable events include those listed in the Event column below along with any associated information identified in the Additional Details column for each event

Table 6-1 Auditable Events

Component	Event	Additional Details
FAU_GEN.1	Start-up and shutdown of the audit functions	
FAU_SAR.1	Opening the audit records	Name of object (audit log file)
FAU_SAR.2	Unsuccessful attempts to read information from the audit records.	
FAU_SEL.1	All modifications to the audit configuration that occur while the audit collection functions are operating	
FAU_STG.3	Actions taken due to exceeding of a threshold	Message sent to administrator
FAU_STG.4	Actions taken due to exceeding of a threshold	
FCS_BCM_EXT.1	Failure of the cryptographic operation	
FCS_CKM.1	Successful key generation and Failure of the key generation process.	
FCS_CKM.2	Success and failure of the activity	
FCS_CKM.4	Failure of key zeroization process	Identity of subject requesting or causing zeroization, identity of object or entity being cleared.
FCS_COP.1a	Failure in encryption or decryption.	Cryptographic mode of operation, name of object being encrypted/decrypted.
FCS_COP.1b	Failure in cryptographic signature	Cryptographic mode of operation, name of object being signed/verified.
FCS_COP.1c	Failure in hashing function	Cryptographic mode of operation, name of object being hashed.
FCS_RBG_EXT.1.1	Failure in the randomization process.	
FDP_ACF.1a, b	All requests to perform an operation on an object covered by the SFP. Use of privilege to bypass the access control mechanism	The name of the object being accessed.
FDP_ETC.1	All attempts to export information	
FDP_ETC.2	All attempts to export information	Manual change in device state
FDP_ETC.2	Overriding of human-readable output marking. (Additional)	
FDP_IFF.1	All decisions on requests for information flow.	
FDP_IFF.2a, b	All decisions on requests for information flow.	
FDP_ITC.1	All attempts to import user data, including any security attributes	
FDP_ITC.2	All attempts to import user data, including any security attributes	
FIA_AFL_EXT.1	The reaching of the threshold for the unsuccessful authentication attempts The action taken (disable for non-administrators, delay for	

Component	Event	Additional Details
	administrator) The re-enablement of disabled non-administrative account	
FIA_SOS.1	Rejection or acceptance by the TSF of any tested secret	
FIA_UAU.2	All use of the authentication mechanism	Origin of the attempt (e.g., terminal identifier, source IP address)
FIA_UAU.6	All re-authentication attempts when changing authentication data	Origin of the attempt (e.g., terminal identifier, source IP address)
FIA_UID.2	All use of the user identification mechanism, including the identity provided during successful attempts	Provided user identity, origin of the attempt (e.g., terminal identifier, source IP address)
FIA_USB.1	Binding of user security attributes to a subject (e.g. creation of a subject) Unsuccessful binding of user security attributes to a subject	
FMT_MOF.1a	All modifications in the behavior of the functions in the TSF.	The old and new values for audit events specified by this function.
FMT_MOF.1a	All modifications in the behavior of the functions in the TSF.	
FMT_MSA.1a – h	All modifications of the values of security attributes	The name of the object, the old and new values of the attributes Assignment of Users, Roles and Privileges to Roles Deletion of Users, Roles and Privileges from Roles Creation and Deletion of Roles
FMT_MSA.2	All modifications of the values of security attributes.	All offered and rejected values for a security attribute.
FMT_MSA.3a	Modifications of the default setting of permissive or restrictive rules. All modifications of the initial value of security attributes	The old and new values of the attributes
FMT_MSA.3b	Modifications of the default setting of permissive or restrictive rules. All modifications of the initial value of security attributes	The old and new values of the attributes
FMT_MTD.1a	All modifications to the values of TSF data	The old and new values of the TSF data
FMT_MTD.1b	Actions taken with respect to the audit records	The specific action that was performed.
FMT_MTD.1c	All initializations of the values of user security attributes.	The initial values for the user security attributes.

Component	Event	Additional Details
FMT_MTD.1d	All modifications of the values of user security attributes.	The old and new values of the attributes.
FMT_MTD.1e	All actions associated with modifications of the values of authentication data.	(none)
FMT_MTD.1f	(none)	(none)
FMT_MTD.1g	All actions associated with modifications of the values of critical cryptographic security parameters.	The old and new values of the parameters, excluding any sensitive information, such as secret or private keys.
FMT_MTD.3	All rejected values of TSF data	
FMT_REV.1a	All attempts to revoke security attributes.	The security attributes that are attempting to be revoked
FMT_REV.1b	All attempts to revoke security attributes	The security attributes that are attempting to be revoked, the object with which the security attributes are associated.
FMT_SAE.1	Specification of the expiration time for an attribute. Action taken due to attribute expiration	
FMT_SMF.1	Use of the management functions	
FMT_SMR.2	Modifications to the group of users that are part of a role Unsuccessful attempts to use a role due to the given conditions on the roles; Every use of the rights of a role (Additional/Detailed)	The role the user is associated with or disassociated from.
FPT_FLS.1	Failure of the TSF	
FPT_RCV.1	The fact that a failure or service discontinuity occurred; Resumption of the regular operations; Resumption of the regular operation;	Type of failure or service discontinuity
FPT_RCV.4	If possible, the impossibility to return to a secure state after a failure of the TSF; If possible, the detection of a failure of a function	
FPT_TST.1	Execution of the cryptography self tests.	For each test, the identification of the test and the results of that test.
FTA_MCS.1	Rejection of a new session based on the limitation of multiple concurrent sessions. Setting the limit on the number of multiple concurrent sessions by an authorized administrator.	The old and new values of the number of multiple concurrent sessions (for setting the session limit).
FPT_STM.1	Setting the time to a specific value	The old and new values for the time.
FTA_LSA.1	All attempts at selecting a session security attributes	
FTA_TSE.1	All attempts at establishment of a user session	
FTP_TRP.1	Failures of the trusted path functions	

Component	Event	Additional Details
	Identification of the user associated with all trusted path failures, if available	

Each audit record includes a date and time of event, type of event, outcome of the event, and the identity of the subject that caused the event.

Slog has audit filtering capabilities wherein filter rules can be defined only by an authorized administrator and are then applied by slog. Filtering rules can include object and subject-specific filtering to include or exclude audit records. The filtering is based upon a high level setting which is set by default and can only be changed by an authorized administrator to one of the following values:

- all: include all events, ignoring filter rules (impractical)
- none: do not include any events, ignore filter rules (insecure)
- filter: include all events, apply filter rules (default)

Slog filtering rules appear as files in a specific directory consisting of subdirectories which dictate the following:

- global_pre: rules to be processed before any other rules
- event_name (i.e., open): rules to be processed for that event
- global_post: rules to be processed after any other rules

Filtering rules are processed in order; first, any rules in global_pre; next any event-specific rules; finally any rules in global_post. Filtering rules can be defined to select events to be included by slog based on the following criteria (FAU_SEL.1):

object identity
user identity;
event type;
success of auditable security events;
failure of auditable security events
Subject sensitivity and integrity label (subject MAC and MIC label);
Object sensitivity and integrity label (object MAC and MIC label);
Subject identity;
Users belonging to a specified Role and Access types (e.g. delete, insert) on a particular object

The audit records written to a slog file are in a delimited text format which is protected with the TSF access control mechanisms from unauthorized access, modification, and deletion. The slogfmt command can be used to produce more readable output from the slog file. The slog file can also be viewed with grep, perl, awk, and sed to search and filter the slog file. Slog files can only be viewed by an authorized administrator or users authorized by role assigned (see the Security Management section for detail on management roles) and can be searched based upon user identity; object identity; date of event; time of event; type of event; success of auditable security events; failure of auditable security events; subject sensitivity label; object sensitivity label; object name and type of access; role that enabled the access; and any combination of date and time of event; user identity; object name & type of access; and the role that enabled the access.

For performance, generated audit records are stored in a 128 KB memory buffer. When a slog file is opened, reserve space sufficient for at least one slog buffer is acquired. In the event of an unclean system shutdown (for example, due to a power outage or system failure), at most one slog buffer of generated audit records may be lost. The sync_interval parameter can be used to limit the time that a particular audit event stays in the audit buffer, thereby further limiting the potential audit data lost in the case of an unclean system shutdown.

The system can be configured such that a warning record is generated when storage for slog files is low. If storage is exhausted when trying to write records to disk or when allocating a new slog file, auditing will stop until an authorized administrator explicitly re-enables it. The system can also be configured to shut down when the system has run out of space for auditing. In such a case, the reserve space is released to allow any pending records to be committed to disk, preventing the loss of records due to space exhaustion.

During system startup, if auditing cannot be started due to an error (for example, the configured slog path is missing or cannot be accessed) the system will inhibit startup script processing and notify the administrator that slog has been disabled, allowing the administrator an opportunity to re-configure auditing before the system reaches a fully operational state.

The Audit security function substantiates the following requirements: FAU_GEN.1, FAU_GEN.2, FAU_SAR.1, FAU_SAR.2, FAU_SAR.3, FAU_SEL.1, FAU_STG.1, and FAU_STG.4.

6.1.2 Access Control

The TSF enforces an access control policy upon requested access by a subject to protected objects. Subjects are processes acting on behalf of users and objects are listed in Table 6-2. The objects listed in Table 6-2 include all the named objects in the TOE according to the definition of named objects as follows.

> A named object is an object that exhibits all of the following characteristics:
> - The object may be used to transfer information between subjects of differing user identities within the TSF.
> - Subjects in the TOE must be able to request a specific instance of the object.
> - The name used to refer to a specific instance of the object must exist in a context that potentially allows subjects with different user identities to request the same instance of the object.

6.1.2.1 Metapolicy

To enforce its access control policy, STOP OS invokes a Metapolicy which implements several sub-policies comprised of a Mandatory Access Control (MAC) Policy, a Mandatory Integrity Control Policy (MIC), a Role-based Access Control Policy (RBAC), and a Discretionary Access Control (DAC) Policy. The MAC and MIC policies are implemented as a single policy and referred to as the Bell-LaPadula and Biba (BL/B) policy given that the MAC policy is based upon the Bell-LaPadula policy and the MIC policy is based on the Biba policy.

Every user is defined with a clearance. A clearance consists of a minimum label, a default label, and a maximum label.

Every subject and object has a security label. A security label consists of 3 parts, one for each component of the security policy:

- Role part -- a comma-separated list of roles
- BL/B part -- the name of a BL/B label representing the sensitivity level and categories and the integrity level and categories.
- DAC part -- the owner, group, and mode in the traditional UNIX discretionary access control policy. The fields are separated by periods.

Table 6-2 Objects

Object	Description
Processes	Processes differ from file system objects in that they are considered to be both active subjects and storage objects. The storage object determination arises from the fact that (a) processes may be the target of an inter-process communication (IPC) message, and (b) processes contain accessible status information. When viewed as an object, processes have the following security

Object	Description
	relevant characteristics: • The real and effective label of the process • The real and effective auth of the process
Semaphores	Semaphores, or more accurately semaphore sets, are used to coordinate access to resources. A semaphore set consists of one or more individual semaphores. Unlike file system objects and devices, semaphores are not included in the process address space. Semaphore sets contain the following security-relevant characteristics: • The label for the semaphore set.
Devices	Devices are classified as objects because (a) devices serve as gateways for information, and, as such, may be viewed as abstract repositories of information; and (b) devices contain accessible status information. They have the following security-relevant characteristics: • The label of the device.
Sockets	Sockets (network and Unix-domain) are classified as objects because they are channels through which data can flow out of and into a process. Sockets are different than other TSF objects in the following ways: the object data may not be stored on the system, there may be no static "name" for the container of the data, data written cannot be read back, and specific "records" cannot be read. Instead, the object data is a dynamic set of packets to which there is no beginning, ending, or maximum size. The name of a network socket is also dynamic in that it can change according to what the sending and receiving processes agree upon. Like pseudo-terminals and unnamed pipes, sockets cease to exist once all processes have closed them. The label of the socket is set to the label of the creating process. Each socket can bind to a local port or identifier suitable for the protocol that is being used. The ability to bind to TCP and UDP ports is controlled by the files in /xts/cfg/net/inet/tcp/ports and /xts/cfg/net/inet/udp/ports, respectively. The TSF ensures that only the creator of a socket can attach to a socket.
File System Objects	File System objects are the basic information repository. File System objects possess the following security-relevant characteristics: • The label for the object. File system objects are used to create a hierarchically structured file system. There are six types of file system objects: files, directories, device special files, symbolic links, Unix-domain sockets, and named FIFOs (pipes).
• Files	The most common type of file system object in the TOE is the regular file. The file header for an executable file also contains the following additional security relevant information: • The auth to which the process should be set when the file is executed
• Directories	A directory is a special type of file system object that is used to construct a hierarchical file system.

Object	Description
• Device Special Files	Device special files serve as the way that devices are designated through the file system. Device special files can only be created in /dev and are automatically created by the TSF. Users cannot create device special files.
• Named FIFOs	Named FIFOs provide a method to support the UNIX "named pipe" concept. They provide a permanent "First-In First-Out" communication path between multiple processes, at the request of the processes, within the bounds of the system security policy. Named FIFOs support multiple readers and multiple writer; locks are used to serialize access. DAC is enforced through the same mechanisms used for files.
• Unix-Domain Socket File System Objects	Unix-Domain socket file system objects are simply a handle for getting to Unix-domain sockets (mentioned above). They are created by a "bind" operation and must be explicitly removed from the system when no longer wanted (i.e., they are not implicitly removed when all processes close the socket). Their security attributes can be changed. Normal opens are not allowed against this kind of file system object.
• Symbolic Links	Symbolic links appear as names in the file system, but simply "point" at another file system object. When an application performs a normal file system operation on a symbolic link, the operation is actually performed on the target of the link. If a symbolic link appears in a pathname, the target of the link is used in the path. There are special system calls to read the content, or get the status, of a symbolic link. Unlike "hard" links, the target object can be on another file system, can be downgraded with respect to the link, or can be non-existent.
Memory Objects	Memory objects do not have a presence in the file system and are not persistent across system re-boots. They exist solely in system memory.
• Shared Memory	Shared memory objects follow the Unix System V shared memory model. They can be shared between processes with a key or ID, or they can be private to the creating process. They persist until the system shuts down or until they are explicitly removed by the creator or owner. Shared memory is "mapped" into the address space of a process. The label of shared memory objects can be changed and is set to the label of the creating process by default.
• Unnamed Pipes	Unnamed pipes serve as a data transfer conduit between processes. They have no interprocess "handle" and can only be shared between predecessor and ancestor in a "fork" relationship. An unnamed pipe ceases to exist as soon as the last process using it either closes the pipe or exits. Pipes are created at the label of the creating process.

The Metapolicy queries each sub-policy, assembles a result from each sub-policy, and makes a final decision. The sub-policies are queried in the following order:

- RBAC
- BL/B
- DAC

Each sub-policy called by the Metapolicy is described in the following subsections. The ability to bypass a policy check is implemented by RBAC roles which define specific actions referred to as exemptions and described below as part of the RBAC Policy. Note that there are no exemptions defined to bypass the RBAC policy.

6.1.2.1.1 RBAC Policy

The RBAC policy supports the Metapolicy by returning a decision regarding a request to access an object based upon the role of the subject. A role is defined by a role identifier and a set of actions a subject in that role may perform. A subject has a currently activated set of roles associated with it which may be all or a subset of those roles assigned to the user on whose behalf the subject is acting. Only authorized administrators may assign roles to users. Actions mostly correspond to system calls such as open, read, stat, mount, etc. STOP OS implements a fixed list of actions that can be associated with roles. Roles are also associated with objects. Upon attempted access (which identifies specific actions) to objects, the RBAC policy checks to ensure that the attempted action is included in a role definition that is associated with that object and, if found, that the role allowing that action is associated with the subject requesting the action.

A role can be defined to inherit from one or more other roles which is implemented by adding other role names to the definition of a role. These other roles are referred to as parents. Inheriting grants the child role the same permissions (actions) to the same objects that are accessible to the parent role. Note that the role(s) associated with subjects and objects may both include inherited roles.

The RBAC policy implements the following computations:

For every role in subject’s session:

- Compute set of effective roles by including inherited roles
    - ➢ For every role on object:
- Compute set of effective roles by including inherited roles
    - ➢ If a role in the intersection of these two effective sets contains the requested action, access is granted

The Metapolicy implements the concept of exemptions in the context of actions that are assigned to roles. Some actions provide exemptions to other parts of the security policy such as blb_exempt_read_security_sub, and dac_exempt_perms_sub. STOP OS includes default roles that contain these actions that implement exemptions.

The RBAC policy is used internally by the TSF to prevent modification or deletion of TSF data, including the audit trail and configuration parameters.

6.1.2.1.2 BL/B (MAC and MIC) Policy

The BL/B policy supports the Metapolicy by returning a decision after invoking both the MAC and MIC policies

6.1.2.1.2.1 MAC Policy

The MAC policy is based upon a sensitivity level and sensitivity categories of the subject and sensitivity (level and categories) of the object.

.

The TSF enforces a mandatory access control policy over all identified system resources (i.e., subjects, storage objects, and I/O devices) that are accessible, either directly or indirectly, to subjects external to the TSF. As the basis of its enforcement, this policy uses MAC labels that are associated with every subject and object in the system. These MAC labels consist of 256 hierarchical sensitivity levels, and 64 nonhierarchical sensitivity categories.

The TOE uses a *dominates* function that is used to compare sensitivity labels; this comparison is done whenever a subject external to the TSF accesses an object. Every user has a clearance that includes a MAC label. The TSF enforces the restriction that any subject created on behalf of a user has a current MAC label dominated by the user’s clearance. A subject cannot operate at mandatory labels below, create objects at mandatory labels below, or change the mandatory label of an object to a label below, the clearance of the owning user

The kinds of access that are relevant are read and write – execute is considered the same as read.. Only authorized administrators can change the MAC label of an object. A MAC label change to an object will take effect immediately, even if that means denying access to the object by a process which already has the object “open”.

6.1.2.1.2.2 MIC Policy

The MIC policy is based upon an integrity level and an integrity categories of the subject and the integrity level and integrity categories of the object.

The TSF enforces a mandatory integrity control policy over all identified system resources (i.e., subjects, storage objects, and I/O devices) that are accessible, either directly or indirectly, to subjects external to the TSF. As the basis of its enforcement, this policy uses MIC labels that are associated with every subject and object in the system. These MIC labels consist of 256 hierarchical integrity levels, and 32 nonhierarchical integrity categories.

The TOE uses a *dominates* function that is used to compare integrity labels; this comparison is done whenever a subject external to the TSF accesses an object. Every user has a clearance that includes a MIC label. The TSF enforces the restriction that any subject created on behalf of a user has a current MIC label that dominates the user’s MIC clearance. A subject can not operate at integrity labels above, create objects at integrity labels above, or change the integrity label of an object to a label above, the clearance of the owning user

The kinds of access that are relevant are read and write – execute is considered the same as read. Only authorized administrators can change the MIC label of an object. A MIC label change to an object will take effect immediately, even if that means denying access to the object by a process which already has the object “open”.

Mandatory integrity control is used internally by the TSF to prevent modification or deletion of TSF data, including the audit trail and configuration parameters.

6.1.2.1.3 DAC Policy

The DAC policy is based on: the user and group identity of the subject requesting access to an object; the object’s owner identity, group identity, and the object’s permissions. The object’s permissions are expressed as Unix permission bits with operations allowed to the owner, operations allowed to the object’s group, and operations allowed to the world.

The kinds of operations or access that are controlled are read, write, and execute. Write does not imply the ability to delete and some objects cannot be executed.

There are three allowed access mode bits in the object’s permission which controls whether or not the specific access will be allowed to the owner, subjects in the object’s group, and the remaining users (world).

- read: If this bit is set, the user or group is allowed read access to the object
- write: If this bit is set, the user or group is allowed write access to the object. This also allows append and deletion, though deletion of file system objects depends not on the object’s permissions, but on the permissions of the containing directory.
- execute: If this bit is set, the user or group is allowed “execute” access to the object. This bit may be ignored or be meaningless for certain types of objects (for example: devices, named First-In First-Out (FIFOs), processes). For directory file system objects, this bit is not interpreted as “execute," but as “search."

If the particular bit for that type of access is not set, the corresponding user or group is denied that mode of access. Hence, to deny access to a given user or group, the bits for all types of access would not be set.

Only administrators can introduce new users and groups to the system and establish the group membership of users. Normal users can change the discretionary attributes of only the objects they own, but administrators can change the attributes of any object.

6.1.2.2 Import and Export of User Data

The TSF imports unlabelled user data and associates the MAC and MIC label of the device to imported unlabelled user data imported from that device. The TSF will associate the effective owner, group, and role of the user importing the data with the data.

The TSF uses the security attributes associated with imported labeled user data and ensures the protocol used correctly and unambiguously associates the security attributes with the data, and that the security attributes are interpreted as intended by the source of the labeled user data. Note that security labels produced by a non-TOE product are never interpreted as labels. The only labels that are ever imported as security labels are always generated by another instance of the TOE. These include file systems and tar files only. In those cases, the administrator should ensure that the source and target TOE have compatible security policies and are thus interpreting the labels in the same way.

Devices used to import data without security attributes cannot be used to import data with security attributes unless the change in device state is performed manually and is auditable. Additionally, devices used to import data with security attributes cannot be used to import data without security attributes unless the change in device state is performed manually and is auditable.

The TSF exports data with and without security attributes. However, devices used to export data without security attributes cannot be used to export data with security attributes unless the change in device state is performed manually and is auditable. Additionally, devices used to export data with security attributes cannot be used to export data without security attributes unless the change in device state is performed manually and is auditable.

Only an administrator can modify the MAC, MIC, or RBAC attributes of a device and only the owner of a device can modify the device's DAC attributes. All changes to device security attributes are auditable.

For data that is exported without security attributes, the devices used for export do have MAC, MIC, DAC, and RBAC attributes and these are implicitly the attributes of the exported information. For data that is exported with security attributes, the TSF ensures the security attributes are unambiguously associated with the exported labeled user data, and the attributes exported consists of MIC, MAC, DAC, and RBAC security attributes.

The MAC, MIC, DAC, and RBAC policy rules govern access from subjects to any export device and such access will be auditable.

6.1.2.3 Residual Protection

The TSF is designed to prevent a process from seeing information left over in an object or memory area used by another process (or the TSF itself).

Generally, main memory and device memory/media/registers are cleared as needed, in a manner dependent on the resource in question. When a file is deleted, the name of the file is deleted, leaving no residual information in the directory (this is a variation from the standard UNIX behavior). When a file shrinks, the TSF clears residual data in partial pages.

The Access Control function substantiates the following requirements: FDP_ACC.1a, FDP_ACC.1b FDP_ACF.1a, FDP_ACF.1b, FDP_ETC.1, FDP_ETC.2, FDP_IFC.2a, FDP_IFC.2b, FDP_IFF.2a, FDP_IFF.2b, FDP_ITC.1, FDP_ITC.2, and FDP_RIP.2.

6.1.2.4 Host-Based Packet Filter Firewall

The TSF provides functionality that can be configured by an administrator to control network traffic (IP traffic) inbound to the TOE or outbound from the TOE. This control is based upon the configuration of filter rules. The filter rules consist of the indication of whether traffic can flow to or from the TOE based upon any combination of the following attributes of IP network traffic: source IP address, destination IP address, source port, destination port and protocol.

Filter rules are used by the TSF to create tables that maintain the filter criteria that are checked by the TSF to control inbound/outbound traffic.

The TSF includes a TSF application (pf) that provides administrative support for this firewall functionality.

6.1.3 Cryptographic Support

6.1.3.1 Cryptography Available to Applications

The TSF includes two FIPS 140-2 validated cryptographic modules which are used by the TSF to provide cryptographic services to applications. The first is a user space cryptographic library that provides FIPS 140-2 validated crypto services to applications (CMVP Certificate #1051). The cryptographic services provided are AES encryption/decryption, RSA digital signature algorithm (rDSA), and SHS hashing. The TSF generates asymmetric keys in accordance with a 1024 bit domain parameter size for rDSA-based keys. The TSF uses a FIPS approved RNG to generate keys that is seeded by an entropy source that accumulates entropy from one or more independent software-based noise sources. When the keys leave the cryptographic module (which has been FIPS 140-2 validated), they are digitally signed using rDSA to support their integrity. Keys are destroyed by zeroization in accordance with FIPS 140-2. The cryptographic algorithms are implemented by openssl and adhere to the following standards:

Algorithm	Mode(s)	Key Size(s)	FIPS Algorithm Certificate #
AES	ECB, CBC, and CFB	128, 192, and 256 bits	#695
rDSA	FIPS 186-3	1024 or greater	#264
SHA256, SHA384, SHA512	N/A	N/A	#723
RNG	N/A	N/A	#407

The second module provides FIPS 140-2 validated kernel cryptographic support (CMVP Certificate #1590) for the FIPS approved DRBG to generate keys that are seeded by an entropy source that accumulates entropy from one or more independent hardware based noise sources. It also provides the ability to encrypt filesystems using the AES algorithm to perform the encryption. The cryptographic algorithms are implemented by the Kernel Cryptographic Module and adhere to the following standards:

Algorithm	Mode(s)	Key Size(s)	FIPS Algorithm Certificate #
AES	ECB and CBC	128, 192, and 256 bits	#1603
DRBG	AES_CTR DRBG	SP800-90	#78
HMAC	SHA-256	Tested KS < BS and KS = BS	#939

Algorithm	Mode(s)	Key Size(s)	FIPS Algorithm Certificate #
SHS	SHA-1, SHA-256, SHA-384, and SHA-512	Byte-only	#1416
TRIPLE DES	ECB and CBC	112 and 168-bit	#1048

6.1.3.2 OpenSSH Cryptographic Support

The TOE implements the OpenSSH Version 2 to establish a secure channel to allow remote access to the TOE. OpenSSH is a protocol for secure remote login and other secure network services over an insecure network. The TOE implements OpenSSH as specified in RFC 4252. OpenSSH provides public key authentication, encryption/decryption, to key exchange using the following algorithms in the below table. The method of compliance with these standards is based upon vendor assertion. The TOE implements a random bit generation according to FIPS 140-2 Annex C which is seeded with a minimum of 256 bits of entropy at least equal to the greatest bit length of the keys it is used to generate. The random bit generation algorithm from FIPS 140-2 Annex C is in accordance with ANSI X9.31-1998 - Appendix A.2.4. The TSF uses this random number generator to generate keys and protects the integrity of the generated keys that are exported from the module according to NIST SP 800-57 “Recommendation for Key Management Part 1: General” (para 6.2.2.2a) using RSA digital signatures.

Cryptographic Algorithms/Protocols and Standards		
Algorithm or Protocol	**Standard**	**Method of Compliance**
OpenSSH V2	RFC 4252	Vendor Assertion
AES	FIPS 197	FIPS Certificate #695
rDSA	FIPS 186	FIPS Certificate #264
Diffie-Hellman	RFC 2631	Vendor Assertion

The Cryptographic Support function substantiates the following requirements: FCS_BCM_EXT.1, FCS_CKM.1a, FCS_CKM.1b, FCS_CKM.2, FCS_CKM.4, FCS_COA_EXT.1, FCS_COP.1a, FCS_COP.1b, FCS_COP.1c, FCS_RBG_EXT.1.

6.1.4 Identification and Authentication

Users are required to authenticate themselves prior to performing any actions either with a user identify and password or with a public key using SSH. The TSF maintains an authentication database in which accounts are defined for users that associates attributes with users. The attributes associated with users are:

user identifier,
group identifier,
authentication data,
security-relevant roles
public key used for SSH
The clearance of the user;

The default Mandatory Access Control (MAC) label;
The default Mandatory Integrity Control (MIC) label.

By default new users are created with a clearance that grants the user no access to anything on the system. Either the default clearance must be changed before the user is created or the user's clearance must be set appropriately after the user is created. If the clearance is not set the user will not be able to log in.

Upon successful authentication, the TSF associates the attribute associated with the user account (listed above) with the subject acting on behalf of that user. The sensitivity and integrity label associated with the subject must be within the clearance range of the user. The ability to specify different initial user security attributes is restricted to authorized administrators or users authorized by their role definition. Changes to security attributes are restricted to the administrator and users in roles that authorize the user including the below:

- change any component of their label within the allowed range;
- change the user ID associated with auditing to a value other than a set-auth program has used in the subject's history (and all ID changes are auditable);
- The user and group ID used for DAC policy checks can only be changed by (and after) starting a program which an administrator has specified as a "set-auth" program;
- change the clearance, role(s), and group membership of a user.

Changes to user security attributes take effect upon subsequent logins. While a user is logged on, they may change their own authentication data upon successfully re-authenticating themselves. Only subjects that were started as set-auth programs can change their user ID to the real and effective user ID for the subject;

To enforce restrictive security attributes by default, users are not assigned to be in the administrative role by default and users are not assigned roles that grant the user privileged actions by default. Furthermore, the administrative role is subdivided into several pre-defined sub-roles that can be granted to an account to provide the ability to limit the authority granted to users based upon need. Exemptions (as further described in the Security Management section) are also granular such that only those actions or exemptions needed can be assigned to a role.

The TSF can detect when the amount authentication attempts related to a user reaches an administrator defined amount. Upon meeting this amount, the account will be disabled for non-administrator users and remain disabled until it is re-enabled by the authorized administrator. Upon the account being disabled, the TSF will present a message to the user identifying that the account is disabled. For administrator accounts, when the amount of authentication attempts meet the administrator defined amount, the TSF will ensure that there can be no more than ten authentication attempts per minute for an authorized administrator configurable time period.

The TSF includes TSF applications used to authenticate (e.g. login, passwd) by calling the operating system specific API.

6.1.4.1 Password Authentication

For password authentication, users are required to enter their user name and password prior to gaining access to any other functions of the system. When defined, passwords are first hashed, using a 120 bit MD5 algorithm, before stored in the authentication database. Upon entry of the password by a user, the entered password is first hashed and then compared to the stored hashed password for that user-id. If the hashed value of the entered password matches the hashed password value stored for that user, authentication succeeds. Passwords are not echoed back to the user when entered and the users will receive an authentication failure message upon a failed authentication attempt. The authentication database is protected by an RBAC role with specific action that allows access. Even users cleared for maximum security cannot read it. Users can only change their own passwords and an authorized administrator may change all passwords.

The administrator can specify the following system-wide parameters for passwords:

- minimum password length to be at least 16 characters

- mixed case required
- non-alphanumeric characters required (symbols and numbers)
- password history size (if the history is of size N, when a user changes his or her password, it can not be the same as any of his or her last N passwords)
- expiration time (after which the user must change the password)

Using these configurable parameters, the administrator can specify a password policy that provides sufficient strength. Additionally, feedback given in response to authentication attempts will be obscured and will not include any diagnostic information with respect to the bad password.

The TSF associates attributes with a user following a successful login including the label (as defined above includes RBAC, BL/B (MAC and MIC), and DAC labels). These attributes are set by an administrator. The TSF starts a user's session at the default MAC and MIC after checking that the default MAC and MIC labels, respectively, are within the user's clearance.

6.1.4.2 Public Key-based Authentication

The TSF allows remote access to the TOE using OpenSSH. The TSF can be configured to include a SSH daemon that performs public key-based authentication. If public-key authentication is desired for a user, that user must have their public key stored in the authentication database. (ssh_authorized_keys). Public-key authentication is implemented using RSA digital signature authentication. See the Cryptographic Support function for a description of the cryptographic mechanisms implemented by OpenSSH. The TSF can be configured to additionally require a password with public-key authentication, if desired.

The Identification and authentication function described above substantiates the following requirements: FIA_AFL_EXT.1, FIA_ATD.1, FIA_SOS.1, FIA_UAU.2, FIA_UAU.6, FIA_UAU.7, FIA_UID.2, and FIA_USB.1.

6.1.5 Security Management

The TSF restricts specific management functionality to only those users that are authorized to perform such functions as allowed by a user's role membership. The definition of a role includes identifying what actions users in that role can perform.

By default, the TOE provides the following roles

- admin (administrator role) -- this role can do anything. The default admin user account has this role. This role inherits from the "slog" role.
- slog (audit role) -- Processes with this role can create application slog (audit) events in the slog file. Note that the auditor role is not listed as a separate role in FMT_SMR.2 as it is considered within the role of users assigned to roles with specific actions and is considered a privileged user.

Note that because new roles can be created with actions that are exemptions (described below), users in these roles that allow exemptions to the policy are also referred to as privileged users and are considered to be in a management role.

The admin role is subdivided into several predefined administrative sub-roles that can be granted to an account to provide that account with limited authority. If a site wants all administrators to have limited power, the admin role can be removed entirely, leaving only the limited component roles. Sub-roles can be used to implement the concept of separation of duties. Users assigned to administrative sub-roles are also considered privileged users.

An action is referred to as privileged or as an exemption if by granting this action to a role, the role is then allowed to operate outside the BL/B or DAC policies. Note that there are no actions that exempt the RBAC policy. Some privileged actions or exemptions involve objects while others do not. The object related and non-object related exemptions are identified below. Non-object related exemptions are also sometimes referred to as capabilities.

Table 6-3 Object Related Exemptions

Object Related Exemptions	
Object Exemptions	**Description**
auth_set	Make a program run as another arbitrary user. Change the current process to run as an arbitrary user that is not the current real or effective user. This action is often needed by privileged code that needs to proxy on behalf of unprivileged users or to become an unprivileged user
auth_exempt_kill_obj	Allows a process to receive a signal from processes whose auth does not match (SIGKILL and SIGSTOP are excluded)
auth_exempt_kill_sub	Allow a subject to send a signal to a process whose auth does not match
blb_exempt _read_security_obj	Allows any subject to read the object even if the BL/B security level specified by the label of the object is higher than or incomparable to the BL/B security level specified by the label of the subject
blb_exempt _read_security_sub	Allow a subject to read an object even if the BL/B security label specified by the label of the object is higher than or incomparable to the BL/B security label specified by the label of the subject
blb_exempt _read_integrity_obj	Allows any subject to read the object even if the BL/B integrity level specified by the label of the object is lower than or incomparable to the BL/B integrity level specified by the label of the subject
blb_exempt _read_integrity_sub	Allow a subject to read an object even if the BL/B integrity label specified by the label of the object is lower than or incomparable to the BL/B integrity label specified by the label of the subject
blb_exempt _write_security_obj	Allows any subject to write an object even if the BL/B security level specified by the label of the object is lower than or incomparable to the BL/B security level specified by the label of the subject
blb_exempt _write_security_sub	Allow a subject to write any object even if the BL/B security label specified by the label of the object is lower than or incomparable to the BL/B security label specified by the label of the subject
blb_exempt _write_integrity_obj	Allows any subject to write the object even if the BL/B integrity level specified by the label of the object is higher than or incomparable to the BL/B integrity level specified by the label of the subject
blb_exempt _write_integrity_sub	Allow a subject to write any object even if the BL/B integrity label specified by the label of the object is higher than or incomparable to the BL/B integrity label specified by the label of the subject
dac_exempt _perms_obj	Allows any subject to bypass DAC permission checks on this object
dac_exempt _perms_sub	Allows a subject to bypass DAC permission checks on any object
dac_exempt _owner_obj	Allows any subject to bypass DAC owner checks on the object

Object Related Exemptions	
Object Exemptions	**Description**
dac_exempt _owner_sub	Allows a subject to bypass DAC owner checks on any object
dac_exempt_owner_set _obj	Allow any subject to set the object's DAC owner or group to arbitrary values
dac_exempt_owner_set _sub	Allows a subject to set an object's DAC owner or group to arbitrary values
label_blb_down_set	Allow a subject to the BL/B portion of a security label downwards. Downwards refers to a lower numeric BL/B security label or a higher numeric BL/B integrity label. Note that this action will not allow a user to set a label outside of that user's clearance, all label changes are confined within the user's clearance. Note that this action will allow a user to downgrade an object
label_blb_up_set	Allow a subject to set BL/B portion of a label upwards. Upwards refers to a higher numeric BL/B security label or a lower numeric BL/B integrity label. Note that this action will not allow a user to set a label outside of that user's clearance; all label changes are confined within the user's clearance. While this action does only permit changing labels upwards, there are a number of ways that a process with this action (and without the label_blb_down_set action) can cause information to flow "down".

.

Table 6-4 Non-Object Related Exemptions

Non-Object Related Exemptions	
Non-Object Exemptions	**Description**
sched	Increase a process's priority or change the priority of another process
shutdown	Shut down or reboot the system
slog	Create slog (audit) events

Any user which is assigned any of the exemptions (both the object exemptions and non-object exemptions) is considered a privileged user.

A normal user (a user not granted any exemptions and not in any of the admin roles) is allowed to change some of the security attributes of their own login session or their own objects such as those below:

- Users may change their own passwords
- Object owners may change the object permission bits used to enforce DAC, if they are in a role that includes the action to allow this modification.

By default new users are created with a clearance that grants the user no access to anything on the system. Either the default clearance must be changed before the user is created or the user's clearance must be set appropriately after the user is created. If the clearance is not set the user will not be able to log in.

By default, if not specified upon object creation, object security attributes are restrictive. Some objects receive the attributes from the creator, others from a parent object, and others may not be accessed until a label is explicitly applied. These methods all result in the assignment of restrictive labels when not explicitly specified.

The TSF ensures that the security attributes used by the defined policies (DAC, RBAC, MAC, and MIC) assigned are valid in that they meet the criteria (e.g. fall within a certain range) of the security attribute. Security attributes that do not meet the criteria are determined to be invalid and are not accepted by the TSF. A valid range may be different for different types of security attributes and is documented in the user guidance.

The TSF ensures that only secure values are accepted for TSF data. The TSF performs checks on TSF data to ensure the TSF data received meets any pre-defined criteria including ensuring that entered labels are structured appropriately, default MAC and MIC labels assigned to users fall within the defined clearance for that user, and passwords meet password defined criteria. TSF data is considered any data used by the TSF to make security relevant decisions (e.g. security attributes), data that is used for monitoring to ensure the TSF is functioning properly (audit data), data that is used to configure security functions (limits on the size of the audit trail), and any data that if corrupted could cause a violation of the security policy.

The TSF provides the ability to revoke security attributes associated with users and restricts this capability to authorized administrators and users in roles assigned an action to allow this authorization. The changes will be enforced upon subsequent logons by that user.

The TSF provides the ability to revoke security attributes associated with objects and restricts this capability to authorized administrators and owners of the object, and users in roles assigned actions that provide this authorization. The changes to the object security attributes are enforced when an access check is next made as follows:

a) a change in MAC, MIC, or RBAC label of an object will be enforced the next time any subject attempts an access to the object;
b) a change in DAC attributes of an object will be enforced the next time any subject attempts to “open” the object. Subjects which already have the object open will be allowed to continue accessing the object; and
c) an owner of an object will only be allowed to change the MAC or MIC label of an object if s/he possesses the appropriate role-action.

Rebooting the system is not necessary for revocation of subject and object security attributes to be enforced. Labels with items already in use will continue to work. However, there will be no new uses of the deleted item upon subsequent attempts to use it because the label will not be found.

The security function described above substantiates these requirements, however, further detail is provided below to meet the more specific requirements.
The Security Management function meets the FMT_MOF.1, FMT_MSA.1a thru FMT_MSA.1f, FMT_MSA.3a, FMT_MSA.3b, FMT_MTD.1a thru FMT_MTD.1d, FMT_MTD.3, FMT_REV.1a, FMT_REV.1b, FMT_SMF, and FMT_SMR.2.

FMT_MOF.1a, b – the ability to manage (enable, disable, and select events to be audited) the audit function is restricted to the authorized administrator and users authorized by a role-action. Additionally, authentication data can be managed only by the authorized administrator and users authorized by a role-action. Users may change their own passwords.

FMT_MSA.1a, b – only object owners in roles that allow this action and users with DAC exemptions may change the DAC permission bits on an object. Only authorized administrators can change object ownership.

FMT_.MSA.1c – only the administrator may create roles and assign them (there are no RBAC exemptions)

FMT_MSA.1d – only an administrator may change an object's MAC label inside of the user's clearance

FMT_MSA.1e – only an administrator may determine what users may read the audit data

FMT_MSA.1f - only an administrator may change an object's MIC label inside of the user's clearance

FMT_MSA.1g – only an administrator may change default security attributes of objects and the object owners.

FMT_MSA.2 – all security attributes are ensured to be valid

FMT_MSA.3a – only an administrator may change default security attributes associated with DAC, MAC, MIC and RBAC.

FMT_MTD.3b – by default, a Host-based Packet Filter firewall policy is enabled which can only be changed or disabled by an administrator or users authorized by a role-action.

FMT_MTD.1a – TSF data (except for audit data, user security attributes, authentication data, and critical cryptographic security parameters) are managed only by authorized administrators and users authorized by an assigned role that includes the appropriate action (role-action).

FMT_MTD.1b – only an administrator and users authorized by a role-action can query, delete, and clear, modify, and read the audit data

FMT_MTD.1c – only an administrator and users authorized by a role-action can initialize user security attributes.

FMT_MTD.1d – only an administrator and users authorized by a role-action can modify user security attributes, other than authentication data.

FMT_MTD.1e – only an administrator can modify authentication data and users can modify their own authentication data.

FMT_MTD.1f – no user can read authentication data

FMT_MTD.1g – only an administrator and users authorized by a role-action can manage the critical security parameters and data related to cryptographic configuration

FMT_MTD.1h – only an administrator and users authorized by a role-action can modify the Host-based Packet Filter firewall policy security attributes (port rules).

FMT_MTD.3 – only an administrator or users authorized by a role-action can enter TSF data (e.g. define passwords, define labels) with the exception of the user that can change their own passwords.

FMT_REV.1a, 1b – revocation of security attributes for subjects and objects can only be performed by an administrator or an object owner authorized by a role-action (for objects) and are immediate.

FMT_SAE.1 – Expiration dates can be assigned to authentication data. Upon expiration, the TSF will not authentication the user using the expired authentication data. Therefore, the user is forced to change their authentication data.

FMT_SMF.1 – The TTSF provides the management functions referenced in all other management requirements listed above.

FMT_SMR.2 – The TSF provides the management roles for which certain management action can be restricted to perform allowing the management of security functions.

6.1.6 Protection of the TSF

The reliable time stamp provided by the TSF is done partly by the hardware and partly by the software. The hardware provides primitive building blocks to support this as follows:

- It stores, and updates, the time-of-day while the TOE is not running.
- It provides periodic timers/counters that the TSF can use at run-time to calculate the current time-of-day.

However, the hardware does not “know”:

a) whether the stored time is correct;
b) which time zone or daylight savings time offsets to apply;
c) whether relative or absolute times are needed;
d) what format times will be kept or displayed in.

Also, some aspects of providing reliable time stamps are implemented completely in software, with no reliance on the hardware building blocks. The TSF (software) implements the protection, with full awareness of the security state and requirements, by:

a) applying appropriate time zone and daylight savings modifications to the time;
b) setting up the Local APIC timer (if available) or the RTC registers to count and interrupt in a particular way;
c) setting up hardware descriptors to route PIT interrupts to the appropriate TSF interrupt handler;
d) using PIT interrupts to update the running (absolute) time stored in the TSF (software);
e) notifying the interrupt hardware when interrupt processing is complete;
f) serializing access to the PIT hardware counter and to the TSF-stored time;
g) placing time stamps in audit records and databases of login activity;
h) displaying time stamps associated with audit records and previous login activity.
i) providing an interface for an administrator to display and change the absolute time kept in software and to change the time zone and daylight savings values;
j) placing corrected times into the hardware for storage when the TOE is not active;
k) providing a programmatic interface for non-TSF software to obtain the current absolute time.

The TSF reacts to the following by recovering the TOE to a consistent and secure state: the corruption or inaccessibility of TSF data including the RBAC role definitions and role assignments to users; an abrupt power failure or unrecoverable hardware, firmware, or TSF software failure.

Upon corruption or inaccessibility to TSF role related data, access checks will fail, and the TOE will enter into a maintenance mode. From this maintenance role, the ability to return to a secure state is provided. The TSF performs the following checks that should fail upon inaccessibility to TSF data: checks whether a specific action is associated to any role; checks whether a specific role has been assigned to a user. The TSF should be able to return the TOE to a consistent and secure state.

The TSF protects its functions by providing the ability to perform self tests to demonstrate the correct operation of the TSF, the integrity of the TSF data, and the integrity of stored TSF executable code. Any user can run these tests whenever desired and the administrator can configure them to run periodically in the background. Only the administrator may generate new checksum information for the TSF files.

In the event that the TOE crashes, the file systems that were mounted at the time of the crash will have to be repaired before they can be used again. The TSF will prevent booting from, or mounting, a file system that was not previously properly unmount or repaired, since the security attributes for objects in the file system may be corrupt (in rare circumstances). The tests include checking the validity of file systems and the TSF offers the ability to repair them by the administrator. The boot system is repaired in a maintenance mode.

The Protection of the TSF security function satisfies the following requirements: FPT_FLS.1, FPT_RCV.1, FPT_RCV.4, FPT_STM.1, FPT_TST.1.

6.1.7 TOE Access

The TSF assigns a role(s) to a user by including the roles in the role part of the user's max clearance. The minimum role set must be a subset of the default role set, which must be a subset of the maximum role set. If the default role set is empty for a user, then upon that user's attempt to login, the user will fail authentication and a session will not be established. Upon a user's attempt to login, the TSF will assign the subject the default role set assigned to the user or any other role that is within the user's maximum role set. Therefore, restriction of session security attributes is achieved by the TSF only assigning a role or role set to a subject if it is subset of the user's maximum role set.

A limit can be set on the amount of concurrent interactive sessions a user may start. The limit can be set only an authorized administrator. The TSF terminates an interactive session after the session is inactive for a specified time interval which can only be defined by the authorized administrator. Both when a user initiates the termination of their session and when the TSF terminates a session because of inactivity, the TSF clears or overwrites the display devices making the current contents unreadable.

The TSF displays an advisory notice and consent warning message concerning unauthorized use of the TOE. Only the authorized administrator can specify the content of the message. Upon authentication by the TSF and session establishment, the TSF shall display to the user the following information: date and time of that authorized user's last successful interactive session establishment; date and time of that user's last unsuccessful attempt and the number of unsuccessful attempts at interactive session establishment for that user identifier since the last successful interactive session establishment. The history shall not be erased by the TSF from the authorized user interface until the user is provided the opportunity to review the information.

The TOE Access security function satisfies the following requirements: FTA_LSA.1, FTA_MCS.1, FTA_TSE.1, FTA_TAB, FTA_TAH.

6.1.8 Trusted Path/Channels

The TSF provides the ability to access the TOE remotely and protects that communication using OpenSSH which uses the cryptographic mechanisms described in the Cryptographic Support security function described. The cryptographic mechanisms protect the communication from disclosure and undetected unauthorized modification.

The TSF provides a trusted communication path between local users and the TSF. On the console the sequence is <Alt-SysRq>. These are known as the SAK (Secure Attention Key). Any invocation of the SAK leads to a Trusted Path. SAK must be used to initiate a login. Any time SAK is used, any current login session will be terminated and a login will be initiated. An alternate SAK key can be configured by the administrator.

The TOE Trusted Path security function satisfies the following requirements: FTP_ITC.1, FTP_TRP.1

7. Protection Profile Claims

This section provides the PP conformance claim statements and supporting justifications of conformance with Protection Profiles.

7.1 No Protection Profile Conformance Claim Reference

This Security Target does not claim conformance with any PP.

8. Rationale

8.1 Security Objectives Rationale

This section provides a rationale for the existence of each threat, policy statement, security objective, and component that comprise the protection profile.

8.1.1 Complete Coverage - Threats

This section provides evidence demonstrating coverage of the threats by both the IT and Non-IT security objectives. The following table shows this objective to threat mapping, along with a discussion of the coverage for each threat.

Threats	Objectives Coverage	Objectives
T.Admin_Error	O.Manage contributes to mitigating this threat by providing the security mechanisms (e.g., tools for reviewing audit data) for administrators to perform TOE administration effectively, and to quickly alert the administrator of ineffective security policies on the TOE.	O.Manage
T.Admin_Rogue	It is important to limit the functionality of administrative roles. If the intentions of an individual in an administrative role become malicious, O.Admin_Role mitigates this threat by isolating the administrative actions within that role and limiting the functions available to that individual. This objective presumes that separate individuals will be assigned separate distinct roles with no overlap of allowed operations between the roles.	O.Admin_Role
T.Audit_Compromise	O.Audit_Generation provides the capability to detect and create records of security relevant events. Audit records identify the user responsible for the event and are an important form of evidence that can be used to track an attacker's actions. Tampering with or destruction of audit data by physical means is addressed by Oe.Physical, which provides physical security controls to the TOE environment. O.Audit_Protection provides the capability to specifically protect audit information from external interference, tampering, or unauthorized disclosure. O.Reference_Monitor protects the TOE and its resources (including audit data) by ensuring that the security policies implemented by the TOE to protect the audit information are always invoked.	OE.Physical O.Audit_Generation O.Audit_Protection O.Reference_Monitor
T.Crypto_Compromise	The cryptography is afforded external protection from viewing, modification, or deletion by malicious users through physical security measures provided by the IT environment [Oe.Physical]. Further, as part of the TOE's security functions (TSF), the cryptography is afforded internal protection from viewing, modification, or deletion by malicious processes and users through the domain isolation maintained by the TOE for its own execution [O.Reference_Monitor].	OE.Physical O.Reference_Monitor

Threats	Objectives Coverage	Objectives
T.Masquerade	To address this threat, O.User_Identification identifies the user as a legitimate user and O.User_Authentication authenticates this user preventing unauthorized users, processes, or external IT entities from masquerading as an authorized entity.	O.User_Authentication O.User_Identification
T.Operational_Errors	The TOE must continue to operate correctly and enforce its security policies once it has been fielded. Some level of testing must be available to authorized users to ensure the TOE's security mechanisms continue to operate correctly once the TOE is fielded. O.Correct_TSF_Operation ensures that once the TOE is installed at a customer's location, the capability exists that the integrity of the TSF (hardware and software) can be demonstrated, and thus provides end users the confidence that the TOE's security policies continue to be enforced.	O.Correct_TSF_Operation
T.Residual_Data	The sharing of hardware resources such as primary and secondary storage components between users introduces the potential for information flow in violation of the TOE security policy when hardware resources are deallocated from one user and allocated to another. In order to prevent such unintended consequences, the TOE prevents the compromise of the TOE security policy through mechanisms that ensure that residual information cannot be accessed after the resource has been reallocated (O.Residual_Information). The intent here is to prevent the unauthorized flow of information that would violate the TOE security policy. The intent is not to require explicit scrubbing or overwriting of data prior to reuse of the storage resource. Therefore, the presence of "residual" data in a storage resource is acceptable as long as it cannot be accessed by subsequent users such that a violation of the TOE security policy results.	O.Residual_Information
T.Resource_Exhaustion	The sharing of resources (e.g. multiple concurrent sessions) between users introduces the potential for a malicious process or user to obstruct users from access to resources via a resource exhaustion denial-of-service attack. O.Resource_Sharing mitigates this threat by requiring the TOE to provide controls to enforce maximum login sessions per user.	O.Resource_Sharing
T.TSF_Compromise	The tampering with or destruction of TSF hardware, software, or configuration data via physical means is addressed by the physical security controls present in the TOE environment [Oe.Physical]. O.Reference_Monitor addresses the threat of tampering with or destruction of TSF hardware,	OE.Physical O.Reference_Monitor

Threats	Objectives Coverage	Objectives
	software, or configuration data by other (non-physical) means. It ensures that the TSF maintains a security domain for its own execution that protects it from interference and tampering by untrusted subjects and enforces the separation between the security domains of subjects within the TSC.	
T.Unattended_Session	When an authorized user leaves an active session unattended, an unauthorized user may gain access to the unattended session. O.Protect mitigates this threat by providing mechanisms to protect user data and resources from unauthorized access by ensuring that the TSF will terminate an interactive session and make the visible contents unreadable after a specified time interval of session inactivity.	O.Protect
T.Unauthorized_Access	Unauthorized users may physically access TOE resources. To mitigate this threat, Oe.Physical restricts the physical access only to authorized personnel. Within the computing environment, O.Access restricts all access controls to authorized users based on their user identity. At the same time, O.Protect enforces access rules by providing mechanisms to prevent the user data from unauthorized disclosure and modification. O.Access_History helps users confirm their previously established session or may help detected possible unsuccessful attempts to their account by an unauthorized user.	OE.Physical O.Access O.Access_History O.Protect
T.Unidentified_Actions	The threat of an administrator failing to know about audit events may occur. To mitigate this threat, O.Audit_Review provides the capability to selectively view audit information, and alert the administrator of identified potential security violations.	O.Audit_Review
T.Unknown_State	After a failure, the security condition of the TOE may be unknown. To mitigate this threat O.Recovery provides procedures and/or mechanisms to ensure that recovery without a protection compromise is obtained.	O.Recovery
T.Spoofing	The threat of spoofing from another machine is mitigated by the use of the Trusted Path. Users are also told at login time of previous logins so that they may assess if another user has been using their account.	O.Access_History O.Trusted_Path
T.RemoteAccess	The threat that the communication between the TOE and remote users may be read and/or modified in an undetected manner is mitigated by the requirement that the channel is trusted.	O.Trusted_Path
T.Unrestricted_Traffic	The threat that inbound/outbound network traffic to/from the TOE may not be controlled is mitigated by including the capability to filter inbound/outbound network traffic.	O.Restrict_Traffic

8.1.2 Complete Coverage - Policy

This section provides evidence demonstrating coverage of the Organizational Security Policy by both the IT and Non-IT security objectives. The following table shows this objective to policy mapping, and the table is followed by a discussion of the coverage for each Security Policy.

Policy	Policy Coverage	Objectives
P.Access_Banner	O.Display_Banner satisfies this policy by ensuring that the TOE displays a banner that provides authorized users with an advisory warning about the unauthorized use of the TOE.	O.Display_Banner
P.Accountability	Enforcement of this policy requires that users be uniquely identified [O.User_Identification] and that their security relevant actions be monitored and recorded [O.Audit_Generation]. The recorded audit information can be selectively reviewed in search of any potential security violations [O.Audit_Review].	O.Audit_Generation O.Audit_Review O.User_Identification
P.Authorization	O.Access supports this policy by requiring the TOE to uniquely identify authorized users [O.User_Identification] prior to allowing any TOE access or any TOE mediated access on behalf of those users. Within the TOE, O.Protect provides mechanisms to prevent user data from unauthorized disclosure and modification.	O.Access O.Protect O.User_Identification
P.Authorized_Users	Within the set of all the users that may interact with the TOE, authorized users are those with access to the information within the TOE after being successfully identified and authenticated by the TOE. Access control policies are used to define the access permitted to the system and its resources. These policies are supported by the implementation of authorized user attributes that identify the user-allowed accesses to TOE information. O.Access supports this policy by ensuring that users only gain authorized access to TOE information and its resources by checking user attributes before system use.	O.Access
P.CRYPTOGRAPHY	By building upon NIST FIPS-validated, cryptography, the TOE not only provides, but also augments the cryptographic support offered solely by baseline NIST FIPS-validated cryptography. The TOE cryptography supports key management (i.e., generation and destruction of keys) and cryptographic operations (i.e., encryption, decryption, signature, hashing, and random number generation). O.Cryptographic_Services provides these cryptographic services to TOE authorized users and/or user applications.	O.Cryptographic_Services
P.I_And_A		O.User_Authentication O.User_Identification
P.Need_To_Know	The need-to-know policy is satisfied by the	O.Access

Policy	Policy Coverage	Objectives
	discretionary access control rules. O.Discretionary_Access protects resources based on the identity of authorized users where the access to objects is directed by owners of the object [O.Discretionary_User_Control]. O.Protect enforces these policy rules by providing the mechanisms to protect the user data from disclosure and modifications and lastly, O.Access ensures that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.	O.Discretionary_Access O.Discretionary_User_Control O.Protect
P.Roles	To appropriately administer the system, O.Admin_Role requires the system to provide multiple administrator roles to isolate actions performed by these different roles. To completely satisfy this policy, separate roles must be assigned separate individuals.	O.Admin_Role
P.Trace	A common organizational security policy is to maintain records allowing for individuals to be held responsible for the actions that they take with respect to organizational assets. Information can be one of the most valuable assets that an organization possesses. To satisfy this policy, O.Audit_Review provides suitable mechanisms to accurately and selectively review those records by authorized personnel to provide accountability at the individual user level to determine any potential security violation.	O.Audit_Review
P.Trusted_Recovery	After a failure or other discontinuity, the security condition of the TOE may be unknown. O.Recovery provides procedures and/or mechanisms to ensure that recovery to a known secure state is obtained without a protection compromise.	O.Recovery
P.Classification	The TOE shall restrict access by sensitivity label. Administrators must exist to configure the MAC labels of users and system devices and management functions must exist to allow such configuration.	O.Mandatory_Access O.Manage

8.1.3 Complete Coverage - Environmental Assumptions

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

Assumption	Objectives Coverage	Objectives
A.Physical	Physical security must be provided for the TOE by the IT environment to ensure the TOE is capable of addressing the threats to TOE assets [Oe.Physical].	OE.Physical

8.2 Security Functional Requirements Rationale

This section demonstrates that the functional components selected for this Security Target provide complete coverage of the defined IT security objectives. The mapping of components to IT security objectives is depicted in the following table.

.

Objective	Functional Component	Justification
O.Access	FDP_ACC.1 FDP_ACC.2 FDP_ACF.1 FDP_ACF.1b FIA_AFL_EXT.1 FIA_ATD.1 FMT_REV.1.a FMT_REV.1.b FTA_MCS.1 FDP_IFC.2a,b FDP_IFF.2a,b	The TOE must protect itself and the resources it controls from unauthorized access. FDP_ACC.2 enforces the Discretionary Access Control (DAC) policy on all subjects and all named objects and all operations among them. The DAC policy specifies the access rules between all subjects and all named objects controlled by the TOE. While authorized users are trusted to some extent, this requirement ensures only authorized access is allowed to named objects. FDP_ACF.1 specifies the DAC policy rules that will be enforced by the TSF and determines if an operation among subjects and named objects is allowed. Furthermore, it specifies the rules to explicitly authorize or deny access to a named object based upon security attributes. FIA_AFL_EXT.1 provides a detection mechanism for unsuccessful authentication attempts. The requirement enables an authorized administrator configurable threshold that prevents unauthorized users from gaining access to authorized user's account by guessing authentication data. This mechanism prevents access by either disabling the targeted account. Thus, limiting an unauthorized user's ability to gain unauthorized access to the TOE. FIA_ATD.1 defines the attributes of users, including a userid that is used by the TOE to determine a user's identity and enforce what type of access the user has to the TOE (e.g., the TOE associates a userid with any role(s) they may assume). FMT_REV.1.a ensures that the authorized administrator has the ability to revoke security attributes to a specific user. This revocation is immediate and helps authorized administrators control the ability of authorized users to log in or perform privileged operations. FMT_REV.1.b ensures that the authorized administrator and owners of named objects have the ability to revoke security attributes to a specific user. This revocation occurs when an access check is made and helps authorized administrators and owners control the ability of users accessing named objects. Additional policies are included in the TOE to support ensuring that only authorized access is granted.
O.Access_History	FTA_TAH.1	FTA_TAH.1 is used to provide information about previous interactive sessions (i.e., date and time). This information is displayed to the authorized user upon each successful interactive session establishment. This requirement gives the authorized users the ability to verify their last successful interactive session and thus, is a means for determining if the previous successful interactive session establishment was authorized or not.
O.Admin_Role	FMT_SMR.1	The TOE must maintain roles to isolate administrative actions. FMT_SMR.1 ensures that a minimum of an administrative role be maintained
O.Audit_Generation	FAU_GEN.1 FAU_GEN.2 FAU_SEL.1 FIA_USB.1 FPT_STM.1	FAU_GEN.1 defines the set of events that the TOE must be capable of recording. This requirement ensures that the authorized administrator has the ability to audit any security relevant event that takes place in the TOE. This requirement also defines the information that must be contained in the audit record for each auditable event. There is a minimum of information

Objective	Functional Component	Justification
		that must be present in every audit record and this requirement defines that, as well as the additional information that must be recorded for each auditable event. FAU_GEN.2 ensures that the audit records associate a user identity with the auditable event. The association is accomplished using the userid of the authorized user. FAU_SEL.1 allows the authorized administrator to configure which auditable events will be recorded in the audit trail. This provides the administrator with the flexibility in recording only those events that are deemed necessary by site policy, thus reducing the amount of resources consumed by the audit mechanism. FIA_USB.1 plays a role is satisfying this objective by requiring a binding of security attributes associated with users that are authenticated with the subjects that represent them in the TOE. This only applies to authenticated users, since the identity of unauthenticated users cannot be confirmed. Therefore, the audit trail may not always have the proper identity of the user that causes an audit record to be generated (e.g., an attacker/user providing another user's user identifier). FPT_STM.1 ensures that the time stamps used to create the audit records are reliable. The time and date included in the time stamp is crucial when generating the audit information to ensure accountability.
O.Audit_Protection	FAU_SAR.2 FAU_STG.1 FAU_STG.4	The audit trail must be protected so that only authorized users and authorized administrators may access it or delete it. FAU_SAR.2 ensures that only authorized users have read access to audit information and FAU_STG.1 ensures that audit information is not modified and protects it from unauthorized deletions. By preventing auditable events (except those taken by the administrator) when the audit trail is full, the objective to protect audit data is further supported.
O.Audit_Review	FAU_SAR.1 FAU_SAR.3 FAU_STG.	FAU_SAR.1 provides the ability for an authorized administrator to efficiently review audit records. This requirement also mandates the audit information be presented in a manner that is suitable for the administrators to interpret the audit trail. FAU_SAR.3 complements FAU_SAR.1 by providing the administrators the flexibility to specify criteria that can be used to search or sort the audit records residing in the audit trail. FAU_SAR.3 requires the administrators be able to establish the audit review criteria based on a user and identifier, date and time, so that the actions of a user can be readily identified and analyzed. Allowing the administrators to perform searches or sort the audit records based on dates, times, type of events, and success and failure of these events, provides the capability to extract the user activity to what is pertinent at that time in order facilitate the administrator's review. It is important to note that the intent of sorting in this requirement is to allow the administrators the capability to organize or group the records associated with a given criteria. FAU_STG.3 allows the authorized administrator to be alerted of the possible audit data loss if the audit trail exceeds an authorized administrator selectable, pre-defined limit.

Objective	Functional Component	Justification
O.Cryptographic_Services	FCS_BCM_EXT.1 FCS_COA_EXT.1 FCS_CKM.1.a FCS_CKM.1.b FCS_CKM.4 FCS_COP.1.a FCS_COP.1.b FCS_COP.1.c FCS_RBG_EXT.1 FCS_CKM.2	Baseline cryptographic services are provided in the TOE by FIPS PUB 140-2 compliant modules [FCS_BCM_EXT.1]. Specific functional requirements in the area of cryptographic operations address data encryption and decryption [FCS_COP.1.a]; cryptographic signatures [FCS_COP.1.b]; cryptographic hashing [FCS_COP.1.c]; random number generation [FCS_RBG_EXT.1]; and supporting key management services [FCS_CKM.1.a, FCS_CKM.1.b, FCS_CKM.4]. These TOE requirements support cryptographic services that can be called upon by the TOE itself, or by TOE authorized users and/or user applications [FCS_COA_EXT.1]. By supporting the distribution of cryptographic keys used to implement cryptographic services, the objective to provide cryptographic services is further supported.
O.Discretionary _Access	FDP_ACC.2 FDP_ACF.1 FIA_USB.1 FMT_MSA.3	Access to TOE resources is determined by the Discretionary Access Control policy. FDP_ACC.2 ensures that the Discretionary Access Control policy is enforced on all subjects and all named objects and all operations between them. FDP_ACF.1 defines the Discretionary Access Control rules to determine if any operation between subjects and named objects is allowed. These rules are based on the identity of the users and their group memberships. FIA_USB.1 defines the associations between user security attributes and subjects acting on behalf of that user by which policy decisions are based upon. FMT_MSA.3 ensures that the TOE provides protection by default for all named objects at creation time. This may allow authorized users to explicitly specify the desired access controls upon the object at its creation, provided that there is no window of vulnerability through which unauthorized access may be gained to newly-created objects.
O.Discretionary _User_Control	FMT_MSA.1.a FMT_MSA.1.b FMT_REV.1.b	To allow authorized users to specify which resources may be accessed, the TOE must provide the ability for object security attributes to be changed and revoked. FMT_MSA.1.a and FMT_MSA.1.b restrict the ability to change the value of object security attributes to authorized administrators and owners of objects. FMT_REV.1.b restricts the ability to revoke security attributes of named objects to authorized administrators and owners of these objects.
O.Display_Banner	FTA_TAB.1	Before identification and authentication and the establishment of a user session, the TOE allows limited access by any potential users of the system in order to convey warnings and agreements for system use. Through this limited access before establishing a user session, the TSF displays an authorized, administrator-specified advisory notice and consent warning message regarding unauthorized use of the TOE [FTA_TAB.1]. In typical applications a user who continues session establishment procedures (including their successful identification and authentication) after display of the notice and warning banner effectively acknowledges the banner content and consents to the stated conditions. This banner of information can be critical in supporting legal actions related to the use of the TOE.
O.Manage	FMT_MOF.1.a FMT_MOF.1.b FMT_MSA.1.a FMT_MSA.1.b FMT_MSA.3 FMT_MTD.1.a	In a variety of ways the TOE supports authorized administrators in the management of security functions, security attributes and data while also restricting unauthorized use. For example, the TOE provides for and restricts the following actions to authorized administrators only (except where specifically noted): • Disable and enable the audit functions, and specify which events

Objective	Functional Component	Justification
	FMT_MTD.1.b FMT_MTD.1.c FMT_MTD.1.d FMT_MTD.1.e FMT_MTD.1.g FMT_REV.1.a FMT_REV.1.b FMT_SAE.1 FMT_SMF.1 FDP_ACC.1, FDP_ACF.1b, FMT_MSA.1c, FMT_MSA.1g FMT_MTD.3 FMT_SMR.2 FMT_MSA.1d, e, f	are audited [FMT_MOF.1.a]. • Create, initialize, change default, modify, delete, clear, append, query, etc. the values of security attributes associated with user authentication data [FMT_MOF.1.b]. • Change the value of object security attributes. (Object owner is also allowed to perform this action.) [FMT_MSA.1.a, FMT_MSA.1.b]. • Provide restrictive default values for security attributes, and specify alternative initial values to override the default values when an object or information is created. [FMT_MSA.3]. • Create, initialize, change default, modify, delete, clear, append, query, etc. the security-relevant TSF data (except audit records, user security attributes, authentication data, and critical security parameters) [FMT_MTD.1.a]. • Query, delete, and clear audit records [FMT_MTD.1.b]. • Initialize user security attributes. [FMT_MTD.1.c]. • Modify user security attributes, other than authentication data. [FMT_MTD.1.d]. • Modify authentication data. (Also allows users authorized to modify their own authentication data to do so.) [FMT_MTD.1.e]. • In addition, the TOE restricts the management of the critical cryptographic security parameters to an authorized administrator [FMT_MTD.1.g]. • Revoke security attributes associated with the users within the TSC. [FMT_REV.1.a]. • Revoke security attributes of named objects within the TSC. (Object owner is also allowed to perform this action.) [FMT_REV.1b]. • Specify an expiration time for authorized user authentication data. [FMT_SAE.1]. FMT_SMF.1 provides a list of the management functions specified in this ST and is required as a dependency for the management functions. By providing role-based access control (RBAC) and controlling the ability to set/modify the security attributes used by the RBAC, the objective to provide the ability to manage the security functions is further supported. By controlling the ability to manage security attributes associated with RBAC, MIC, and MAC, the ability manage the security functions is further supported. By ensuring management roles can perform certain functionality, the ability manage the security functions is further supported.
O.Protect	FDP_ACC.2 FDP_ACF.1 FDP_RIP.2 FIA_SOS.1 FIA_UAU.7 FMT_MTD.1.f FMT_REV.1.b	O.PROTECT requires mechanisms be provided by the TOE to protect user data and resources. FIA_SOS.1 prescribes the metrics that must be satisfied for user authentication. If a user can’t authenticate, he or she will not have the ability to access user data and resources. FIA_SOS.1 requires that the authentication mechanism provide the ability for authorized users to have a “secret” up to 16 characters in length, consisting of any combination of upper and lower case letters, numbers, and punctuation. FIA_UAU.7 ensures that no feedback that affects the ability of users to circumvent the authentication mechanism is presented during the authentication process. The TOE is allowed to provide information that would allow the user to use the authentication mechanism in a correct manner (e.g., press CTRL-ALT-DELTE, slide card quickly, center your finger and press firmly, speak louder and slowly), but not provide information that may allow alteration to their presentation that would thwart

Objective	Functional Component	Justification
		the mechanism. FMT_MTD.1.f ensures that the authentication data is protected. No entity is allowed to read authentication data and the TSF must prevent any attempt to read it. To protect user data and resources, FDP_ACC.2, FDP_ACF.1, and FMT_REV.1.b require a Discretionary Access policy and rules that ensures the correct access to named objects by subjects acting on behalf of users. To ensure that user data is not disclosed before a resource is reused, FDP_RIP.2 ensures that the shared memory and operating system controlled files are not available to another user thus protecting the user data.
O.Recovery	FPT_RCV.1	FPT_RCV.1 ensures that the system enters a maintenance mode allowing the system to be returned to a secure state after a failure or service discontinuity. In a secure state, all security policies are enforced.
O.Residual_Information	FDP_RIP.2	FDP_RIP.2 is used to ensure the contents of resources are not available to subjects other than those explicitly granted access to the data.
O.Resource_Sharing	FTA_MCS.1	FTA_MCS.1 identifies user accounts as a system resource that could be exhausted (through multiple concurrent "logons" of a single individual). The requirement mandates that the administrator be able to limit the number of concurrent logon sessions by a single user. This ensures that a single individual could not mount a denial-of-service attack using multiple sessions as launching points. Resources (e.g., memory contained on the network card) that are not covered by the above are subject to denial of service attacks. Denial-of-service attacks of these resources should be addressed via other mechanisms such as redundant hardware.
O.Reference_Monitor	FPT_RCV.1	This objective requires the protection of the TSF (and its data) from external interference, tampering or inappropriate disclosure by mandating that the TSF create and maintain a domain for its execution. Domain is defined as the logical area that the TSF provides for itself in which to operate. Common mechanisms include hardware execution domains (e.g., processor execution rings as well as other isolation mechanisms that protect TSF data when it is in transit to other TSF components.) FPT_RCV.1 is used to ensure that the TSF offers a mechanism to recover from a failed state by mandating that the TSF provide maintenance mode from which to re-initiate (or establish) a known (secure) state. This ensures that once the TSF has established a domain for its own execution it can always return to that state with confidence that this domain continues to be present.
O.User_Authentication	FIA_SOS.1 FIA_UAU.2 FIA_UAU.6 FIA_UAU.2	FIA_UAU.2 plays a role in satisfying this objective by ensuring that every user is authenticated before the TOE performs any TSF-mediated actions on behalf of that user. FIA_UAU.6 ensures that the authorized user changing his authentication data re-authenticates before he or she is allowed to proceed. To verify the claimed identity of an authorized user, FIA_SOS.1 prescribes the metrics that must be satisfied. It provides the mechanism that will verify the secret for user authentication. In any case, FIA_SOS.1 requires that the authentication mechanism provide the ability for authorized users to have a "secret" up to 16 characters in length, consisting of any combination of upper and lower case letters, numbers, and punctuation This requirement replaces FIA_UAU.1 which is identified to support O.User_Authentication in the GPOSPP. It provides further security

Objective	Functional Component	Justification
		functionality by not allowing any actions on behalf of the user until the user is successfully authenticated.
O.User_Identtification	FIA_UID.2	FIA_UID.2 plays a role in satisfying this objective by ensuring that every user is identified before the TOE performs any TSF-mediated actions on behalf of that user. It also allows for the specification of a list of public objects that users are allowed read access before the user is identified. This requirement replaces FIA_UID.1 which is identified to support O.User_Identification in the GPOSPP. It provides further security functionality by not allowing any actions on behalf of the user until the user is successfully identified.
O.Recovery	FPT_FLS.1	By ensuring the secure state is preserved upon corruption or unavailability of RBAC security attributes, the objective to ensure the TOE can recover from failure is further supported.
	FPT_RCV.4	The TSF ensures that specific security checks regarding the checking of role membership and the enforcement of the RBAC policy complete successfully, or upon failure to complete, recovers to a consistent secure state. This behaviour further the objective to ensure the TOE can recover from failure.
O.Access	FTA_LSA.1	By restricting the scope of the session to a role assigned to the user, the objective to ensure that authorized users only gain access to the resources that are authorized to access.
	FTA_TSE.1	By ensuring the TSF can deny session establishment based upon the role set being empty, the objective to ensure that authorized users only gain access to the resources that are authorized to access.

Objective	Functional Component	Justification
O.Mandatory_Access	FDP_IFC.2a,b FDP_IFF.2a,b FMT_MSA.1d, e, f FDP_ETC.1, 2 FDP_ITC.1, 2 FDP_ACC.1, FDP_ACF.1b	These requirements ensure that information is labeled and users are assigned labels that are within their assigned clearance, and grants control by applying rules that are based upon the label of the information and the label of the user. Additionally, these requirements ensure that the labels are assigned and managed only by authorized users. Both labeled and unlabeled data can be imported and exported and the labels. RBAC also supports the definition of clearance.
O.Trusted_Path	FTP_ITC.1 FTP_TRP.1, FIA_UAU.6	The TOE is required to be able to establish a trusted path between local and remote users to the TSF. Additionally, the TOE requires the user to re-authenticate upon changing authentication data.
O.Restrict_Traffic	FDP_IFC.1, FDP_IFF.1, FMT_MSA.1h, FMT_MSA.3b	The TOE is required to be able to control TOE inbound/outbound network traffic

8.3 Security Assurance Requirements Rationale

This ST claims EAL 4 augmented with ALC_FLR.3 and does so to meet the needs of the product's customer base.

8.4 Requirement Dependency Rationale

The following table shows that all the security functional requirement dependencies that exist based on the security functional requirements (and iterations thereof) included in this Security Target are satisfied, with one exception for which rationale is provided below. All dependencies are met, with the exceptions noted below.

Note that all dependencies are met within the assurance package EAL4 as it is package taken directly from the Common Criteria. Therefore, a dependency rationale is not necessary.

Table 8-1 SFR Requirement Dependencies

ST Requirement	CC Dependencies	ST – How Met
FAU_GEN.1	FPT_STM.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1 and FIA_UID.1	FAU_GEN.1 and FIA_UID.2
FAU_SAR.1 FAU_SAR.2 FAU_SAR.3 FAU_STG.1	FAU_GEN.1 FAU_GEN.1 FAU_GEN.1 FAU_GEN.1	FAU_GEN.1 FAU_GEN.1 FAU_GEN.1 FAU_GEN.1
FAU_SEL.1	FAU_GEN.1 and FMT_MTD.1	FAU_GEN.1 and FMT_MTD.1
FAU_STG.3 FAU_STG.4	FAU_STG.1	FAU_STG.1
FCS_BCM_EXT.1	none	None
FCS_CKM.1a	FCS_RBG_EXE.1 and FCS_CKM.4	FCS_RBG_EXE.1 and FCS_CKM.4
FCS_CKM.1b	FCS_RBG_EXT.1	FCS_RBG_EXT.1
FCS_CKM.2	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] and FCS_CKM.4	FDP_ITC.1, FDP_ITC.2 and FCS_CKM.4
FCS_CKM.4.1	FDP_ITC.1 and FDP_ITC.2 or FCS_CKM.1	FDP_ITC.1, FDP_ITC.2 and FCS_CKM.4
FCS_COA_EXT.1	FCS_BCM_EXT.1	FCS_BCM_EXT.1
FCS_COP.1a FCS_COP.1b FCS_COP.1c	FDP_ITC.1 or FDP_ITC.2 and FCS_CKM.1 and FCS_CKM.4 FDP_ITC.1 or FDP_ITC.2 and FCS_CKM.1 and FCS_CKM.4 FDP_ITC.1 or FDP_ITC.2 and FCS_CKM.1 and FCS_CKM.4	FDP_ITC.1, FDP_ITC.2 and FCS_CKM.4 FDP_ITC.1, FDP_ITC.2 and FCS_CKM.4 FDP_ITC.1, FDP_ITC.2 and FCS_CKM.4
FCS_RBG_EXT.1	FCS_BCM_EXT.1	FCS_BCM_EXT.1
FDP_ACC.2		
FDP_ACC.1, FDP_ACC.2, FDP_ACF.1a, FDP_ACF.1b, FMT_MSA.1c	FDP_ACF.1 FDP_ACC.1 and FDP_ACF.1 FDP_ACC.2 and FMT_MSA.3 FDP_ACC.1 and FMT_MSA.3 FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FDP_ACF.1b FDP_ACC.1b and FMT_MSA.3 FDP_ACC.2 and FMT_MSA.3 FDP_ACC.1b and FMT_MSA.3 FMT_SMR.2 and FMT_SMF.1 and FDP_ACC.1b
FDP_IFC.2a, FDP_IFF.2a FMT_MSA.1d FMT_MSA.1e FMT_MSA.1f	FDP_IFF.1 FDP_IFC.1 and FMT_MSA.3 FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1) FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1) FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FDP_IFF.2a FDP_IFC.2a and FMT_MSA.3 FMT_SMR.2, FMT_SMF.1 and FDP_IFC.2a FMT_SMR.2, FMT_SMF.1 and FDP_IFC.2b FMT_SMR.2, FMT_SMF.1 and FDP_IFC.2b
FMT_MSA.1g	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FDP_IFC.2a, FDP_IFC.2b, FDP_ACC.1, FDP_ACC.2 , FMT_SMR.1 and FMT_SMF.1
FMT_MSA.1h	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FDP_IFC.1, FMT_SMR.1 and FMT_SMF.1

ST Requirement	CC Dependencies	ST – How Met
FMT_MSA.3b	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1h and FMT_SMR.1
FDP_IFC.1, FDP_IFF.1	FDP_IFF.1 FDP_IFC.1 and FMT_MSA.3	FDP_IFF.1 FDP_IFC.1 and FMT_MSA.3
FDP_IFC.2b, FDP_IFF.2b	FDP_IFF.1 FDP_IFC.1 and FMT_MSA.3	FDP_IFF.2b FDP_IFC.2b and FMT_MSA.3
FDP_ETC.1	(FDP_ACC.1 or FDP_IFC.1)	FDP_IFC.2a, FDP_IFC.2b, FDP_ACC.1, and FDP_ACC.2
FDP_ETC.2	(FDP_ACC.1 or FDP_IFC.1)	FDP_IFC.2a, FDP_IFC.2b, FDP_ACC.1, and FDP_ACC.2
FDP_ITC.1	(FDP_ACC.1 or FDP_IFC.1) and FMT_MSA.3a	FDP_IFC.2a, FDP_IFC.2b, FDP_ACC.1, FDP_ACC.2 , and FMT_MSA.3a
FDP_ITC.2	(FDP_ACC.1 or FDP_IFC.1) and (FTP_TRP.1 or FTP_ITC.1) and FPT_TDC.1	FDP_IFC.2a, FDP_IFC.2b, FDP_ACC.1, FDP_ACC.2, and FTP_TRP.1 FPT_TDC.1: See below rationale
FDP_RIP.2	None	None
FIA_AFL_EXT.1	FIA_UID.1	FIA_UID.2
FIA_ATD.1	None	None
FIA_SOS.1	None	None
FIA_UAU.2 FIA_UAU.7	FIA_UID.1 FIA_UID.1	FIA_UID.2 FIA_UID.2
FIA_UAU.6	None	None
FIA_UID.2	none	none
FIA_USB.1	FIA_ATD.1	FIA_ATD.1
FMT_MOF.1a FMT_MOF.1b	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1
FMT_MSA.1a FMT_MSA.1b	FDP_ACC.1, FDP_IFC.1, FMT_SMR.1, FMT_SMF.1	FDP_ACC.1, FDP_IFC.1, FMT_SMR.1, FMT_SMF.1
FMT_MSA.2	FDP_ACC.1 FDP_IFC.1 FMT_MSA.1 FMT_SMR.1	FDP_ACC.1 FDP_IFC.1 FMT_MSA.1 FMT_SMR.1
FMT_MSA.3 FMT_MTD.3	FMT_MSA.1 and FMT_SMR.1 FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1a-g and FMT_SMR.2 FMT_MSA.1a-g and FMT_SMR.2
FMT_MTD.1a	FMT_SMR.1	FMT_SMR.1
FMT_MTD.1b FMT_MTD.1c FMT_MTD.1d FMT_MTD.1e FMT_MTD.1f FMT_MTD.1g	FMT_SMR.1 and FMT_SMF.1 FMT_SMR.1 and FMT_SMF.1 FMT_SMR.1 and FMT_SMF.1 FMT_SMR.1 and FMT_SMF.1 FMT_SMR.1 and FMT_SMF.1 FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1 FMT_SMR.1 and FMT_SMF.1 FMT_SMR.1 and FMT_SMF.1 FMT_SMR.1 and FMT_SMF.1 FMT_SMR.1 and FMT_SMF.1 FMT_SMR.1 and FMT_SMF.1
FMT_REV.1a FMT_REV.1b	FMT_SMR.1 FMT_SMR.1	FMT_SMR.1 FMT_SMR.1
FMT_SAE.1	FMT_SMR.1 and FPT_STM.1	FMT_SMR.1 and FPT_STM.1
FMT_SMF.1	none	
FMT_SMR.1 FMT_SMR.2	FIA_UID.1 FIA_UID.1	FIA_UID.2 FIA_UID.2
FPT_FLS.1	none	none
FPT_TDC.1	none	none
FPT_RCV.1	AGD_OPE.1	AGD_OPE.1
FPT_RCV.4	none	none
FTA_LSA.1	none	none

ST Requirement	CC Dependencies	ST – How Met
FPT_STM.1	none	None
FTA_TSE.1	none	none
FPT_TST_EXT.1	FCS_COP.1 and FCS_RBG_EXT.1	FCS_COP.1 and FCS_RBG_EXT.1
FTA_MCS.1	FIA_UID.1	FIA_UID.2
FTA_TAB.1	none	none
FTA_TAH.1	none	none
FTP_TRP.1	none	none

FDP_ITC.2 (Import of user data without security attributes) includes a dependency on FPT_TDC.1 (Inter-TSF basic TSF data consistency). FPT_TDC.1 is focused on ensuring that security labels imported from another trusted IT product are consistently interpreted between the TOE and that product. However, the TOE only imports security attributes from removable media. Therefore, there is no receipt of labels from another trusted IT product and FPT_TDC.1 is not applicable.

8.5 TOE Summary Specification Rationale

The following table describes the association between the TOE Security Functions and the TOE Security Functional Requirements. This table in conjunction with rationale provided in Section 6.1 demonstrates that the TOE Security Functional Requirements are satisfied.

Table 8-2 Security Function to TOE SFR Mapping

	Security audit	Access Control	Cryptographic Support	Identification and authentication	Security management	Protection of the TSF	Resource Utilization	TOE Access	Trusted Path/Channel
FAU_GEN.1	X								
FAU_GEN.2	X								
FAU_SAR.1	X								
FAU_SAR.2	X								
FAU_SAR.3	X								
FAU_SEL.1	X								
FAU_STG.1	X								
FAU_STG.3	X								
FAU_STG.4	X								
FCS_BCM_EXT.1			X						
FCS_CKM.1			X						
FCS_CKM.2			X						
FCS_CKM.4			X						
FCS_COA_EXT.1			X						
FCS_COP.1a			X						
FCS_COP.1b			X						
FCS_COP.1c			X						
FCS_RBG_EXT.1.1			X						
FDP_ACC.2		X							

	Security audit	Access Control	Cryptographic Support	Identification and authentication	Security management	Protection of the TSF	Resource Utilization	TOE Access	Trusted Path/Channel
FDP_ACC.1		X							
FDP_ACF.1a, b		X							
FDP_ETC.1		X							
FDP_ETC.2		X							
FDP_IFC.1		X							
FDP_IFF.1		X							
FDP_IFC.2a, b		X							
FDP_IFF.2a, b		X							
FDP_ITC.1		X							
FDP_ITC.2		X							
FDP_RIP.2		X							
FIA_AFL_EXT.1				X					
FIA_ATD.1				X					
FIA_SOS.1				X					
FIA_UAU.2				X					
FIA_UAU.6				X					
FIA_UAU.7				X					
FIA_UID.2				X					
FIA_USB.1				X					
FMT_MOF.1a					X				
FMT_MOF.1a					X				
FMT_MSA.1a – h					X				
FMT_MSA.2					X				
FMT_MSA.3a					X				
FMT_MSA.3b					X				
FMT_MTD.1a – g					X				
FMT_MTD.3					X				
FMT_REV.1a					X				
FMT_REV.1b					X				
FMT_SAE.1					X				
FMT_SMF.1					X				
FMT_SMR.2					X				
FPT_FLS.1						X			
FPT_RCV.1						X			
FPT_RCV.4						X			
FPT_TST.1						X			
FPT_STM.1						X			
FTA_MCS.1								X	
FTA_TAB.1								X	
FTA_TAH.1								X	
FTA_LSA.1								X	
FTA_TSE.1								X	
FTP_ITC.1									X
FTP_TRP.1									X