CA Access Control R12 SP1
Security Target

Version 2.0

October 10, 2009

Prepared for:
CA
100 Staples Drive
Framingham, MA 01702

Prepared by:
Booz Allen Hamilton
Common Criteria Testing Laboratory
900 Elkridge Landing Road, Suite 100
Linthicum, MD 21090-2950

Table of Contents

List of Figures

List of Tables

1 Security Target Introduction

This chapter presents the Security Target (ST) identification information and an overview. An ST contains the Information Technology (IT) security requirements of an identified Target of Evaluation (TOE) and specifies the functional and assurance security measures offered by the TOE.

1.1 ST Reference

This section provides information needed to identify and control this ST and its Target of Evaluation. This ST targets Evaluation Assurance Level 3 (EAL3).

1.1.1 ST Identification

ST Title:	CA Access Control r12 SP1 Security Target
ST Version:	2.0
ST Publication Date:	October 10, 2009
ST Author:	Booz Allen Hamilton

1.1.2 Document Organization

Chapter 1 of this ST provides identifying information for the CA Access Control r12 SP1. It includes an ST Introduction, ST Reference, ST Identification, TOE Reference, TOE Overview, and TOE Type.

Chapter 2 describes the TOE Description, which includes the physical and logical boundaries.

Chapter 3 describes the conformance claims made by this ST.

Chapter 4 describes the Security Problem Definition as it relates to threats, Operational Security Policies, and Assumptions met by the TOE.

Chapter 5 identifies the Security Objectives of the TOE and of the operational environment.

Chapter 6 describes the Extended Security Functional Requirements.

Chapter 7 describes the Security Functional Requirements (SFRs).

Chapter 8 describes the Security Assurance Requirements (SARs).

Chapter 9 is the TOE Summary Specification (TSS), a description of the functions provided by CA Access Control r12 SP1 to satisfy the security functional and assurance requirements.

Chapter 10 provides a rationale, or pointers to a rationale, for security objectives, assumptions, threats, requirements, dependencies, and PP claims.

1.1.3 Terminology

Term	Definition
Access Authority	A permission owned by an access to perform a specified access on a resource. Also known as access rights.
Accessor	Users and groups of users in the TOE. Accessors are both end users and administrators.
ACL	An Access Control List (ACL) specifies the accessors that are granted access to a resource and the type of access to which the user is granted.
Administrator	A trusted user who has the authority to stop Access Control services, modify all or part of the rules, policies, and accessor information in Seosdb.
Agent	Also known as Seagent. Responsible for providing Access Control client applications access to Seosd and local OS management.
Authorization daemon	Also known as Seosd. Daemon responsible to manage access requests decision and CA Access Control database updates. Also responsible for restarting Seagent if it has stopped.
CACL	Conditional Access Control List. Provides an extension to the ACL. Specifies access to a resource where the access is by a particular method.
Class	Defines the properties that a record can have (Terminal, Process, Program, etc). Also defines a type of resource.
Client (OS and machine)	The machine from where Selang is used.
Database	Also known as Seosdb. The main repository that contains information on accessors, resources and the policies that govern them.
Default Record	The permissions which are applied to a resource if no specific record for that resource exists.
End User	A person who can log on, or can be the owner of a program batch, or daemon program. An administrator is an end user when trying to access local files (audit data). They’re governed the same way as

Term	Definition
	normal end users.
Enterprise user store	On the native operating system. Access Control pulls user information from here to Seosdb, or refers to the enterprise user store if the OS_user option is on.
Group	A collection of users who usually shares the same access authorizations.
Host (OS and machine)	The machine where CA Access Control components are installed.
NACL	Negative Access Control List. It specifies the accessors that are denied authorization to a resource, together with the type of access they are denied.
Object	A record on the TOE or a resource on the OS.
Ownership	A user or a group that has been explicitly assigned to a record.
Operation	Any action on an object (create, delete, read, write, execute, none, etc.).
PACL	Program Access List. Specific to an ACL that has a program tied to it.
Policy	A rule or group of rules assigned to a record of an accessor or resource (ex. ACL, PACL, CACL, NACL).
Record	A record is an instantiation of an accessor or a resource which the TOE protects, which includes the attributes an administrator can manage to control access to a resource.
Resource	An object that is protected by the access control mechanisms of the TOE (e.g. file, program, or service).
Rule	A rule is written by an administrator to determine a user's access to a resource.
Security Level	An integer between 0 and 255 that can be assigned to accessors and resources.
Selang	Command Language Interface.
SEOS_syscall	Used to intercept security related kernel events.
Subject	An individual (end user or administrator) in the context of attempting to access protected resources (either managed by the

Term	Definition
	TOE or part of it).
Superuser	A Superuser is the default administrator upon installation of the TOE. This account is disabled once the TOE is in an operational state.
User	A user is an Administrator or End User.
Watchdog	Also known as Seoswd. This daemon constantly checks that the other Access Control Services are running. If Seosd has stopped, Seoswd restarts it.

Table 1-1: Customer Specific Terminology

Term	Definition
Authorized user	A user who may, in accordance with the TSP, perform an operation.
External IT entity	Any IT product or system, untrusted or trusted, outside of the TOE that interacts with the TOE.
TOE Security Functions (TSF)	A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.

Table 1-2: CC Specific Terminology

1.1.4 Acronyms

Acronym	Definition
CA	CA Incorporated
EAL	Evaluation Assurance Level
IT	Information Technology
OS	Operating System
PP	Protection Profile
SAR	Security Assurance Requirements
SFR	Security Functional Requirements
ST	Security Target
TOE	Target of Evaluation
TSS	TOE Summary Specification

Table 1-3: ST Acronyms

1.1.5 References

The documentation referenced to populate this ST is identified below:

[1] *Common Criteria for Information Technology Security Evaluation*, CCMB-2009-07-002, Version 3.1 Revision 3, July 2009.

[2] CA™ Access Control Endpoint Administration Guide for UNIX r12

[3] CA™ Access Control Implementation Guide r12

[4] Integrated_arch_IAM_whitepaper.pdf

[5] CA™ Access Control Product Brief

[6] CA™ Access Control selang Reference Guide r12

[7] CA™ Access Control Release Notes r12

1.2 TOE Reference

1.2.1 TOE Identification

CA Access Control r12 SP1

1.3 TOE Overview

CA Access Control is a security software product that is tied to the operating system. The UNIX/LINUX Operating Systems (OS) are used in the evaluated configuration. In addition to supplying the regular security functions – such as an access rule database, an audit log, and administration tools – CA Access Control intercepts in memory the operating system events that are to be protected. No changes are made to system files other than the OS configuration files, and the UNIX kernel is not modified at all. CA Access Control either denies or allows the operation based upon rules and policies in Seosdb. The TOE enforces policy-based control of who can access objects protected by the PROGRAM, PROCESS, TERMINAL, FILE, USER, GROUP, SEOS, SURROGATE, XUSER, and XGROUP classes. In addition, the TOE enforces policy based controls to determine what users can do with their respective access rights and under what circumstances that access is allowed.

CA Access Control is not a replacement for the operating system, but works in conjunction with the underlying OS. CA Access Control hooks security related syscalls that must be protected and an interception is put on the Access Control kernel module at load time. This means control is passed to CA Access Control before the action or operation is executed. Following the syscall interception, CA Access Control then decides whether the user is allowed to perform the requested operation.

> ***Note:*** *The standard OS functionality which can be performed by CA Access Control is not evaluated in this ST.*

The TOE:

- Provides policy enforcement on TOE resources
- Executes allowed operations for a particular user if bound by a rule or policy
- Provides secure audit logs

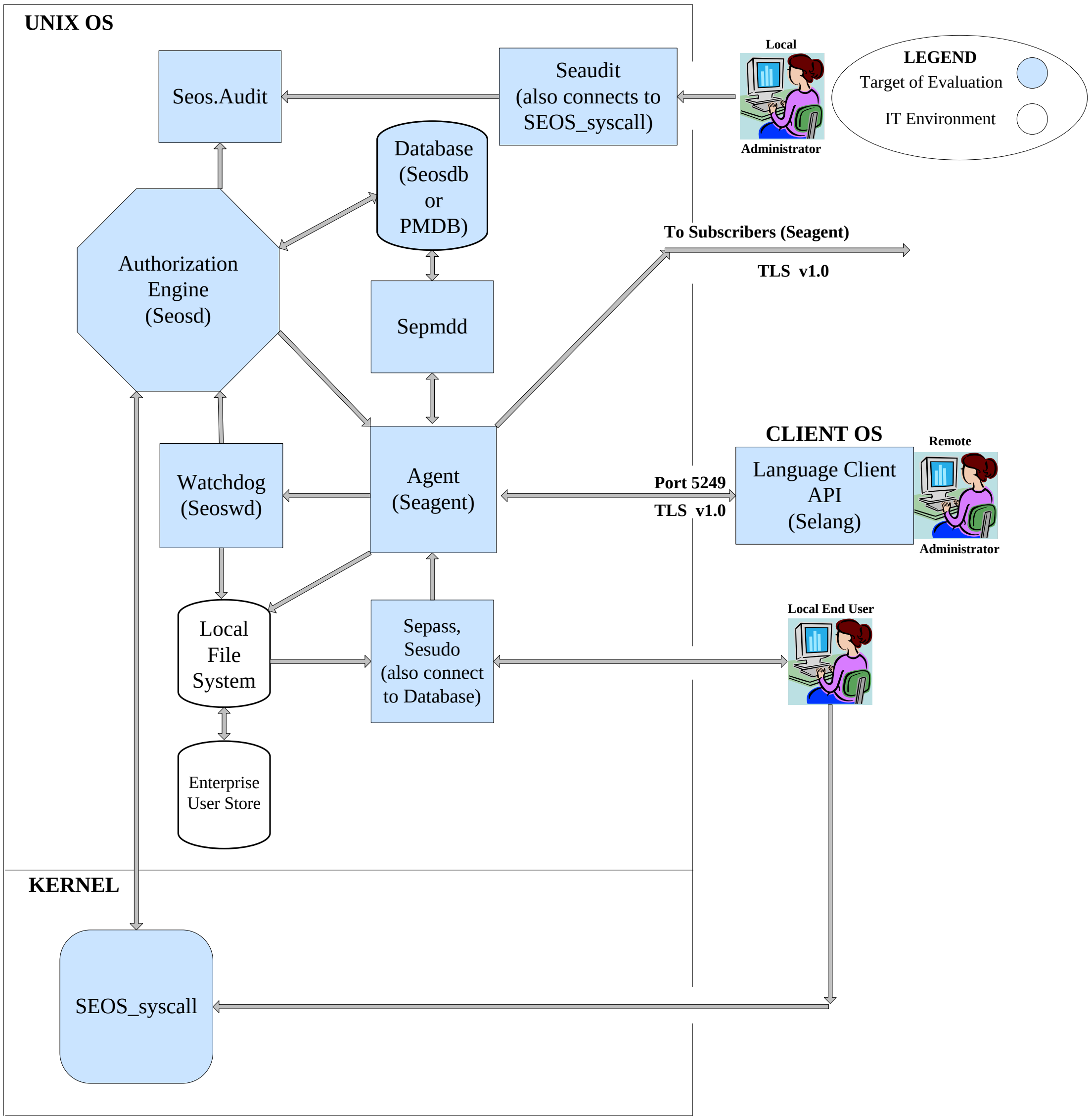


Figure 1-1: TOE Boundary

In the evaluated configuration, the TOE is able to manage distributed systems simultaneously by utilizing the Policy Model approach. In the Policy Model, a Policy Model Database (PMDB) is used as a central repository for a configuration. Other endpoints subscribe to the PMDB, and when an administrator updates the PMDB, the updates are made to all subscribers as well, as illustrated in the following diagram:

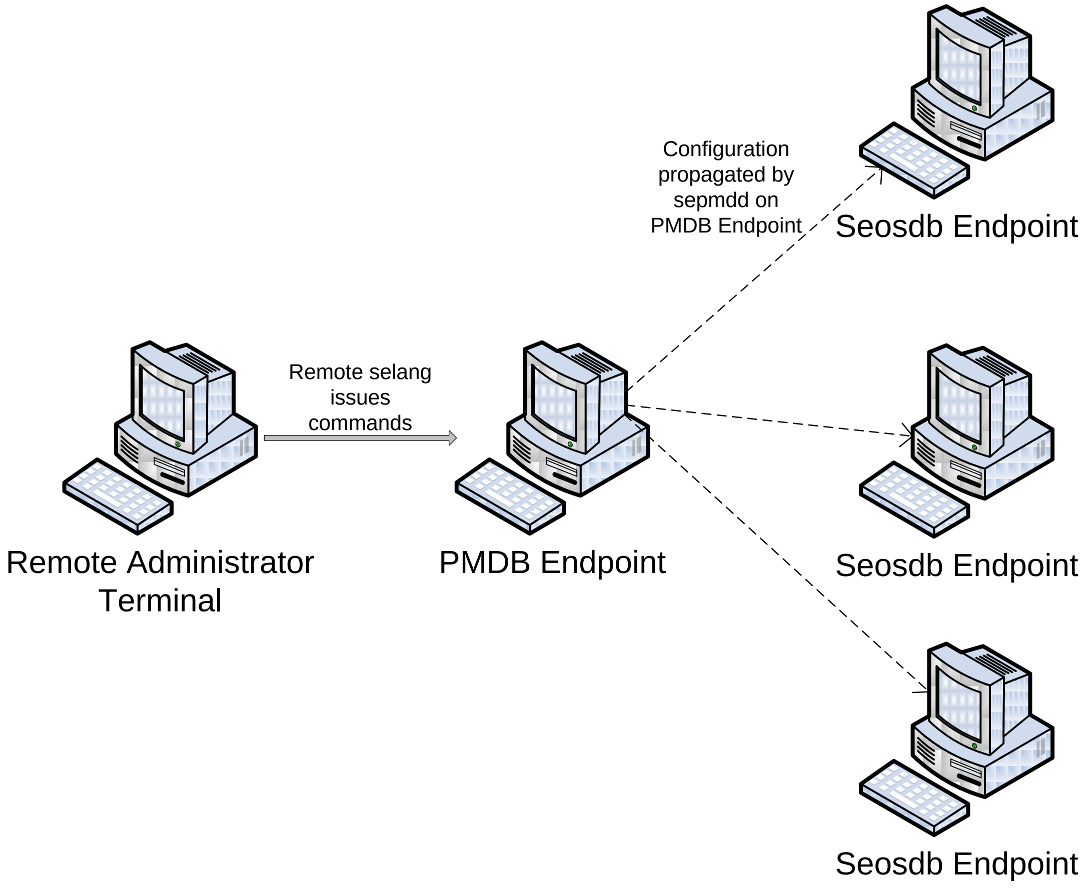

Configuration propagated by sepmdd on PMDB Endpoint
Seosdb Endpoint
Remote selang issues commands
Remote Administrator Terminal
PMDB Endpoint
Seosdb Endpoint
Seosdb Endpoint

Figure 1-2 Policy Model Implementation

When sepmdd detects that its PMDB has been updated, it propagates the updates by communicating with the seagents of the subscriber endpoints. The seagents parse these commands and execute them as if they had been issued by selang.

1.4 TOE Type

CA Access Control r12 SP1 provides the following: System Access Control.

2 TOE Description

This section provides a description of the TOE in its evaluated configuration. This includes the physical and logical boundaries of the TOE.

2.1 Evaluated Components of the TOE

CA Access Control r12 SP1 is comprised of multiple components. The components are identified below:

- Seosd
- Seoswd (Watchdog)
- Seagent (Agent)
- Seosdb (Database)
- SEOS_syscall
- Selang
- Policy Model Database (PMDB)
- Seos.audit
- Seaudit
- Sepmdd (PMDB – Policy Model Database)
- Sepass

> ***Note:*** *Selang is installed on the local machine, but can also be used to manage remote hosts. Selang is used for remote administration in the evaluated configuration. The TOE's components will run with the root privileges*

How an accessor's request to access a resource is intercepted is dependent upon the underlying OS. Table 1-4: Component Definitions defines each component of CA Access Control r12 SP1. For a detailed description of each component see section 9-1: Evaluated Components of the TOE.

Component	Definition
Policy Model Database (PMDB)	A PMDB is a repository of CA Access Control and contains information on two types of objects: accessor records and resource records. It also contains the rules and policies which govern accessor access to objects. It is identical to Seosdb except for the fact that a Seosdb (or other PMDB)

Component	Definition
	can subscribe to a PMDB so that any changes made to the PMDB will be made to all subscriber databases as well. PMDB functions as a virtual instance of Access Control that pushes updates automatically based on the commands issued to it from an actual instance of Access Control.
Seaudit	Seaudit is the application used by the TOE to access and interpret the audit records in a human-readable format.
Seos.audit	Seos.audit is the local storage for the end user's behavior on a local machine. It audits how end users interact with resources protected by Access Control on their own machine. While seos.audit contains the raw audit data, it is accessed by the seaudit application. The seos.audit file can be backed up to one or more files, which are labeled seos.audit.bak.*, where * represents the date the backup was created.
Seosd	Seosd is the main CA Access Control authorization daemon/service. The Seosd makes the runtime decisions required to grant or deny access to a resource. In addition, Seosd monitors the Agent to ensure it is running. If the Agent stops, Seosd will restart it. Seosd is also responsible for keeping track of a user's initially authenticated name so that they can't circumvent TOE rules via the su command.
Seoswd (Watchdog)	Seoswd monitors file information and digital signatures of programs that are defined in Seosdb as trusted programs. Seoswd also monitors the status of seosd and restarts it if it is terminated.
Sepass	Sepass is a replacement for the local passwd command that allows password policies defined by Access Control to be applied to the system accounts of end users.
Sepmdd	Sepmdd runs on the same machine as any PMDB which has been configured. It checks to see when the PMDB is updated (either by selang or a PMDB to which it subscribes), applies the update, and pushes the update to all subscriber seosdb and PMDB instances when an update has been detected.
Agent	Agent is responsible for communicating with CA Access Control clients through port 5249 over TLS v1.0. Additionally, it manages security for the remote administrators and monitors the Watchdog daemon/service.
Seosdb	Seosdb is the main repository of CA Access Control and contains information on two types of objects: accessor

Component	Definition
	records and resource records. Seosdb also contains the rules and policies which govern accessor access to objects.
Selang	Selang is a command line interface which is used remotely by administrators to manage the TOE. Selang allows administrators to manage the records of the accessors and resources in their environment. Administrators can create new accessor records, delete and modify accessor records, modify all or part of Seosdb, and assign administrative attributes to other administrators. .
SEOS_syscall	SEOS_syscall typically hooks into the operating system at boot up time (though it can be performed after boot as well) and intercepts all access and privilege requests. SEOS_syscall works in conjunction with Seagent and Seosd to allow or deny access to the TOE.

Table 2-1: Component Definitions

2.2 Excluded from the TOE

- Access Control Endpoint Management (Policy Manager)
- The following classes:

ADMIN	AGENT	AGENT_TYPE	APPL	AUTHHOST	CALENDAR
CATEGORY	CONNECT	CONTAINER	GAPPL	GAUTHHOST	GFILE
GHOST	GSUDO	GTERMINAL	HNODE	HOSTNET	HOSTNP
HOLIDAY	HOST	LOGINAPPL	MFTERMINAL	POLICY	PWPOLICY
RESOURCE_DESC	RESPONSE_TAB	RULESET	SECFILE	SECLABEL	SPECIALPGM
TCP	UACC	USER_ATTR	USER_DIR		

- Unicenter TNG
- Unicenter TNG user-defined classes
- Reporting Service
- The native Operating System of the host platform
- The GUI Administrator Interface
- The Task Delegation Service
- B1 Security
- Global Access Check (GAC)
- Resource Cache
- Network Cache
- Concurrent Logins
- Secons utility
- Other utilities as stated in Access Control Reference Guide pp. 13-17 that have not been discussed

2.3 Physical Boundary

The TOE includes the following CA Access Control components:

- Seosdb (Database)
- Seosd

- SEOS_syscall
- Seagent
- Seoswd (Watchdog)
- Seos.Audit
- Seaudit
- Selang Command Line Interface
- Sepmdd (PMDB – Policy Model Database)

The following table identifies the CC testing platforms for CA Access Control r12 SP1

Product Component	Version	Platforms
CA Access Control	r12 SP1	Linux Red Hat Advanced Server 5.0
		Solaris 10

Table 2-2: Supported Operating Systems

> ***Note:*** *In addition to the platforms listed in the table above, TLS implementation is also required to run the TOE.*

<u>Minimum Requirements</u>

Component	Solaris Unix	Linux
CPU	Sparc Workstation 64-bit	X86 64-bit
Memory (RAM)	128 MB	128 MB
Hard Disk Space	100 MB – minimal installations	100 MB – minimal installations
	150 MB – general installations	150 MB – general installations
Client Package	60,000 KB	60,000 KB

In addition to the above requirements, disk space is needed for the CA Seosdb, which is the repository of records describing trusted programs, accessors and resources, and the authorizations that permit controlled access to the resources. For example, a database for one thousand accessors, one thousand files, and five hundred access rules, occupies approximately 2 MB of disk memory.

2.4 Logical Boundary

The logical boundary of the TOE includes the CA Access Control r12 SP1 software and can be described in the terms of the security functionalities that the TOE provides for accessor access control to resources.

It is assumed that there will be no un-trusted users of CA Access Control. The TOE environment is one where the potential attacker is unsophisticated, with access to only standard equipment and public information about the product. The TOE relies upon the underlying operating system and platform to provide reliable time for its audit records.

2.4.1 Identification and Authorization

There are two types of users of the TOE: Administrators and end users. Administrators manage the TOE remotely through the command line interface: selang. One or more of them will also be given the ability to access the audit records locally using seaudit. End users access the TOE directly by logging onto their respective local machine. Both types of users are authenticated by the underlying Operating System before they are allowed to access the TOE. The TOE can define password composition requirements to be applied to system accounts for one or more endpoint users. This is accomplished by using the sepass utility. Identification and Authorization is explained in more detail in section 9.2.4 Identification and Authorization.

2.4.2 Access Control

Every attempt to access a resource is performed by an accessor. These accessors must be governed to ensure the proper access authorities or access rights are assigned and enforced. In CA Access Control, these access rights are assigned and managed in a variety of way, however, to gain access to a resource the accessor must meet one or more of the following criteria:

- The accessor must have the proper authority as granted by the resource Access Control List (ACL)
- The accessor must be a member of a group that has access authority
- The accessor must be running a program that has the access authority. For example, the accessor has the authority to run a program in the PROGRAMS class.
- The default access of the resource allows some degree of interaction to accessors for which there’s no specific authority.

For a more detailed description of the Access Controls provided by CA Access Control, review section 9.2.5 Access Control for Accessors.

2.4.3 Classes

Each object belongs to a predefined class which is a collection of objects of the same type. In CA Access Control, the class of a record defines the properties that the record can have. All records in a class have the same properties but will have different values for these properties. Each record contains values for the properties appropriate to the record's class.

The classes included in the evaluated configuration are: PROGRAM, PROCESS, TERMINAL, FILE, USER, GROUP, SUDO, SURROGATE, XUSER and XGROUP. For more information on Classes, review section 9.2.6 Classes.

2.4.4 Security Audit

CA Access Control generates secure and reliable audit logs which associate usernames to all resource actions. It maintains a user's "true" username so that rules cannot be circumvented by the su command. The audit records are stored in an audit log called seos.audit. The location of the audit log is specified in the seos.ini file.

For more detailed information on Security Audit review section 9.2.7 Security Audit.

2.4.5 Security Management

The TOE provides management capabilities through selang, the command line interface that is used by remote administrators. Through the use of selang, CA Access Control allows administrators to manage accessors and resources in their environment. Administrators can create new accessor records, delete and modify accessor records, modify all or part of Seosdb, and assign administrative attributes to other administrators. In addition, administrators can perform distributive management of multiple endpoints simultaneously, applying single rules or a collection of them to a target subset of the environment. For more information on Security Management review section 9.2.8 Security Management.

2.4.6 Degraded Fault Tolerance

Once the TOE is started, its applications monitor each other so that if one is terminated, it can continuously be restored by another. Seoswd is responsible for restarting seosd if it shuts down, seosd is responsible for restarting seagent if it shuts down, and seagent is responsible for restarting seoswd if it shuts down. This ensures that the TOE cannot be shut down on a local system without authorization and also ensures continued operation in the event of an unexpected failure. In addition, seosd will refuse any kill attempt made against, including kill -9. The kernel module of the TOE is able to intercept attempts to shut down the TOE and reject them.

2.4.7 TOE Security Environment

The TOE is an application installed on a UNIX/Linux OS and controls access to resources through Identification and Authorization and System Management. The TOE relies upon the underlying OS to provide authentication which ensure users are associated with the proper security attributes. The originally authenticated username of the accessor is maintained by the TOE so that a trusted user cannot impersonate another trusted user. In addition, the TOE relies upon the underlying OS to provide reliable time stamps for audit records. These requirements are outlined in section 6.3 Extended Security Functional Requirements for the Operational Environment.

2.4.8 Encrypted Communications

In the evaluated configuration, Access Control employs the AES and RSA encryption algorithms. The AES encryption algorithm uses 128-bit HMAC keys for symmetric cipher. The RSA asymmetric-key encryption algorithm is used with SHA-256 for TLS connections and key generation. The TLS connection is used to protect the disclosure and modification of information between Seagent and the selang shell on the remote client. It's also used to protect the communications between endpoints when sepmdd is updating subscriber databases when the Policy Model is used; which is performed between PMDB Endpoint's Seagent and the subscriber's Seagent. For more information on Encrypted Communications, review section 9.2.8 Encrypted Communications.

3 Conformance Claims

3.1 CC Version

This ST is compliant with *Common Criteria for Information Technology Security Evaluation*, CCMB-2009-07-002, Version 3.1 Revision 3, July 2009.

3.2 CC Part 2 Conformant

This ST and Target of Evaluation (TOE) is Part 2 conformant for EAL3, to include all applicable NIAP and International interpretations through 13 May 2008.

3.3 CC Part 3 Conformant Plus Flaw Remediation

This ST and Target of Evaluation (TOE) is Part 3 conformant plus flaw remediation for EAL3, to include all applicable NIAP and International interpretations through 13 May 2008.

3.4 PP Claims

This ST does not claim Protection Profile (PP) conformance.

3.5 Package Claims

This TOE has a package claim of EAL3.

3.6 Package Name Conformant or Package Name Augmented

This ST and Target of Evaluation (TOE) is conformant to EAL package claims augmented with ALC_FLR.1 and ASE_TSS.2**.**

3.7 Conformance Claim Rationale

There is no Conformance Claim rationale for this ST.

4 Security Problem Definition

4.1 Threats

The TOE itself has threats and the TOE is also responsible for addressing threats to the environment in which it resides. The assumed level of expertise of the attacker for all the threats is unsophisticated. The following are threats addressed by the TOE.

T.ACCESS: Unauthorized users could gain local or remote access to protected objects that they are not authorized to access.

T.ADMIN_ERROR: An administrator may incorrectly install or configure the TOE, or install a corrupted TOE resulting in ineffective security mechanisms.

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.EAVESDROPPING: A malicious user could eavesdrop on network traffic to gain unauthorized access to TOE data.

T.MASK: Users whether they be malicious or non-malicious, could gain unauthorized access to the TOE by bypassing identification and authentication countermeasures.

4.1.1 Organizational Security Policies

There are no Organizational Security Policies that apply to the TOE.

4.1.2 Assumptions

The specific conditions listed in this section are assumed to exist in the environment in which the TOE is deployed. These assumptions are necessary as a result of practical realities in the development of the TOE security requirements and the essential environmental conditions on the use of the TOE.

4.1.3 Personnel Assumptions

A.ADMIN: One or more authorized administrators will be assigned to install, configure and manage the TOE.

A.PATCHES: System Administrators exercise due diligence to update the TOE with the latest patches and patch the Operational Environment (e.g., OS and database) to ensure all known system vulnerabilities are not exploited.

A.NOEVIL: Administrators of the TOE are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided by the guidance documentation.

4.1.4 Physical Assumptions

A.LOCATE: The TOE will be located within controlled access facilities that will prevent unauthorized physical access.

5 Security Objectives

5.1 Security Objectives for the TOE

The following security objectives are to be satisfied by the TOE.

O.ACCESS: The TOE will provide measures to authorize users to access specified TOE resources once the user has been authenticated. User authorization is based on access rights configured by the authorized users of the TOE.

O.AUDIT: The TOE will provide measures for recording security relevant events that will assist the authorized users in detecting misuse of the TOE and/or its security features that would compromise the integrity of the TOE and violate the security objectives of the TOE.

O. FILESYS: The Security features offered by the TOE protect the audit files used by the TOE.

O.MANAGE: The TOE will provide authorized administrators with the resources to manage and monitor user accounts, resources, and security information relative to the TOE.

O.EAVESDROPPING: The TOE will encrypt TSF data between the Seagent and administrators on the Client operating system to prevent malicious users from gaining unauthorized access to TOE data.

O.IDENTIFY: The TOE will provide measures to uniquely identify all users and will maintain their original identity if they issue commands as a super user in the environment.

O.PASSWORD: The TOE will enforce defined organizational password complexity requirements.

O.SELF_PROTECTION: The TOE will preserve a secure state and ensure access control to resources when a component of the TOE fails.

O.ROBUST_ADMIN_GUIDANCE: The TOE will provide administrators with the necessary information for secure delivery and management.

5.2 Security Objectives for the Operational Environment

The following security objectives for the operational environment of the TOE must be satisfied in order for the TOE to fulfill its security objectives.

OE.ADMIN: One or more authorized administrators will be assigned to install, configure and manage the TOE.

OE.ROBUST_ACCESS: The operational environment will provide mechanisms that control a user’s logical access to the TOE and to explicitly deny access to specific users when appropriate.

OE.NOEVIL: Administrators of the TOE are not careless, wilfully negligent, or hostile and will follow and abide by the instructions provided by the guidance documentation.

OE.LOCATE: The TOE will be located within controlled access facilities that will prevent unauthorised physical access.

OE.AUTH: The Operational Environment will provide measures to uniquely identify all users and will authenticate the claimed identity prior to granting a user access to the resources protected by the TOE. The Operational Environment will provide measures to uniquely identify all administrators and will authenticate the claimed identity prior to granting an administrator access to the TOE.

OE.SYSTIME: The operating environment will provide reliable system time.

6 Extended Security Functional Requirements

6.1 Extended Security Functional Requirements for the TOE

There are no extended Security Functional Requirements for the TOE in this ST.

6.2 Extended Security Assurance Requirements

There are no extended Security Assurance Requirements in this ST.

6.3 Extended Security Functional Requirements for the Operational Environment

The Table below contains the extended security functional requirements for the Operational Environment:

Security Function	Security Functional Components
Identification and Authentication (FIA)	FIA_UAU_EXT.2 (1) User Authentication Before Any Action
	FIA_UAU_EXT.2 (2) User Authentication Before Any Action
	FIA_UID_EXT.2 (1) User Identification Before Any Action
	FIA_UID_EXT.2 (2) User Identification Before Any Action
Protection of the TSF (FPT)	FPT_STM_EXT.1 Reliable Time Stamps

Table 6-1: Extended Security Functional Requirements for the Operational Environment

6.3.1 Class FIA: Identification and Authentication

Identification and Authentication is required to ensure that users are associated with the proper security attributes (e.g. username, password). The following extended requirements for the FIA class have been included in this ST because the Operational Environment invokes separate methods for administrator and end user authentication.

6.3.1.1 FIA_UAU_EXT.2 (1) User authentication before any action

Hierarchical to: FIA_UAU.1 Timing of authentication

FIA_UAU_EXT.2.1 (1)The Operational Environment shall require each end user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that end user.

Dependencies:	FIA_UID.1 Timing of identification.

FIA_UAU_EXT.2 (2) User authentication before any action

Hierarchical to:	FIA_UAU.1 Timing of authentication
FIA_UAU_EXT.2.1	(2)The Operational Environment shall require each administrator to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that administrator.
Dependencies:	FIA_UID.1 Timing of identification.

6.3.1.2 FIA_UID_EXT.2 (1) User identification before any action

Hierarchical to:	FIA_UID.1 Timing of identification
FIA_UID_EXT.2.1	(1)The Operational Environment shall require each end user to be successfully identified before allowing any other TSF-mediated actions on behalf of that end user.
Dependencies:	No dependencies.

FIA_UID_EXT.2 (2) User identification before any action

Hierarchical to:	FIA_UID.1 Timing of identification
FIA_UID_EXT.2.1	(2)The Operational Environment shall require each administrator to be successfully identified before allowing any other TSF-mediated actions on behalf of that administrator.
Dependencies:	No dependencies.

6.4 Class FPT: Protection of the TSF

This class contains families of functional requirements that relate to the integrity and management of the mechanisms that constitute the TSF and to the integrity of TSF data. The following extended requirements for the FPT class have been included in this ST because the operational environment is capable of performing protection of the TSF events that are not covered by CC Part 2.

6.4.1 Reliable time stamp

Hierarchical to: No other components.

FPT_STM_EXT 1.1 The Operational Environment shall provide reliable time stamps.

Dependencies: No dependencies.

Application Note: *The Underlying OS needs to provide reliable time stamps from the system clock that is used for inclusion in the audit records generated by the TOE.*

6.5 Proper Dependencies

All dependencies for the extended security functional requirements were pulled from CC Part 2.

7 Security Functional Requirements

7.1 Security Functional Requirements for the TOE

The following table provides a summary of the Security Functional Requirements implemented by the TOE.

Security Function	Security Functional Components
User Data Protection (FDP)	FDP_ACC.1 Subset Access Control
	FDP_ACF.1 (1) Security Attribute Based Access Control
	FDP_ACF.1 (2) Security Attribute Based Access Control
Identification and Authentication (FIA)	FIA_ATD.1 User Attribute Definition
	FIA_SOS.1 Verification of Secrets
	FIA_UID.2 User Identification Before Any Action
Security Audit (FAU)	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 Protected Audit Trail Storage
Security Management (FMT)	FMT_MOF.1 Management of Security Functions Behavior
	FMT_MSA.1 Management of Security Attributes
	FMT_MSA.3 Static attribute initialization
	FMT_SMF.1 Specification of Management Functions
	FMT_SMR.2 Restrictions Security Roles
	FMT_MTD.1 Management of TSF Data
	FMT_REV.1 Revocation

Security Function	Security Functional Components
Cryptographic Support (FCS)	FCS_CKM.1 Cryptographic Key Generation
	FCS_CKM.4 Cryptographic Key Destruction
	FCS_COP.1 Cryptographic Operation
Protection of the TSF (FPT)	FPT_FLS.1 Failure w/ Preservation of State
Resource Utilization (FRU)	FRU_FLT.1 Degraded Fault Tolerance
TOE Access (FTA)	FTA_TSE.1 TOE Session Establishment
Trusted Path/Channels (FTP)	FTP_TRP.1 Trusted Path

Table 7-1: Security Functional Requirements

7.1.1 Class FAU: Security Audit

7.1.1.1 FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

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

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

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

c) [***all operations listed in Table 7-2 Audit Properties***].

Value of AUDIT	What is Logged	Applicable Objects
FAILURE	Access failures	Accessors and resources
SUCCESS	Access successes	Accessors and resources
LOGINFAILURE	Login failures	Accessors
LOGINSUCCESS	Login successes	Accessors

ALL	Equivalent to FAIL, SUCCESS, LOGINFAIL and LOGINSUCCESS	Accessors and resources
TRACE	Equivalent to ALL plus all system events	Accessors
NONE	No logging	Accessors and resources

Table 7-2: Audit Properties

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, 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 PP/ST, [***contents as listed in Table 7-3 Common Audit Records***].

Column	Contents	Description
1	Date	The date the access or attempted access occurred.
2	Time	The time the access or attempted access occurred.
3	Return code	The CA Access Control return code that indicates what happened. Valid values are: • D - CA Access Control denied access to a resource or did not permit an update to Seosdb because the accessor did not have sufficient authorization. • F - An attempt to update Seosdb failed. • M - CA Access Control was started or shut down. • O - An accessor or administrator logged out. • P - CA Access Control permitted access to a resource or permitted a login. • S - Seosdb was successfully updated. • U - A trusted PROGRAM or SECFILE was changed, so it is now un-trusted. • W - An accessor's authority was insufficient to access the

Column	Contents	Description
		specified resource; however, CA Access Control allowed the access because warning mode is set in the resource.
4	Event type/ Class	The type of event being audited or the class on which the action was performed.
5	Accessor/ Class	If the previous column contains a class name, this column contains the name of the accessor who executed the command. If the previous column contains UPDATE, this column contains the class in which the action was performed. Otherwise, this column contains the name of the accessor who executed the command or any other relevant information about the class.
6	Access type/ Accessor	If the previous column contains the accessor name, this column contains the access type, if relevant. If the previous column contains the class name, this column contains the name of the accessor who executed the command. Otherwise, this column contains the access type, if relevant, or any other relevant information according to the class.
7	Stage code	A number (up to three digits) that indicates at which stage CA Access Control decided what action to take and why.
8	Audit record code	A number that represents the reason that CA Access Control wrote an audit record.
9	Resource	This column contains the name of the resource being accessed or updated.
10	Terminal/ Program	If column four contains UPDATE, this column contains the name of the terminal from which the update was made. Otherwise, this column contains the name of the program that accessed the resource.
11	Command	If column four contains UPDATE, this column contains a complete copy of the command entered by the accessor. If the command is a password update, the password itself is replaced by a series of asterisks. If column four does not contain UPDATE and an action is being performed on the CLASS object via a remote terminal, then this column displays the IP address of remote terminal.

Table 7-3: Common Audit Records

7.1.1.2 FAU_GEN.2 User identity association

Hierarchical to: No other components.

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.

Dependencies: FAU_GEN.1 Audit data generation

FIA_UID.1 Timing of identification

Application Note: *FIA_UID_EXT.2 (1,2) has been included to satisfy the FIA_UID.2 requirement.*

7.1.1.3 FAU_SAR.1 Audit Review

Hierarchical to: No other components.

FAU_SAR.1.1 The TSF shall provide [***Authorized user***] with the capability to read [***all audit information in seos.audit***] from the audit records.

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

Dependencies: FAU_GEN.1 Audit data generation

Application note: *Authorization to read audit logs is managed by the host OS and is typically set up during install by delegating access privileges to a group.*

Application note: *The term "Authorized user" applies to end users. An administrator is an end user when trying to access local files (audit data). They're governed the same way as normal end users.*

7.1.1.4 FAU_SAR.2 Restricted audit review

Hierarchical to: No other components.

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.

Dependencies: FAU_SAR.1 Audit review

Application note: *The term "users" applies to end users. An administrator is an end user when trying to access local files (audit data). They're governed the same way as normal end users.*

7.1.1.5 FAU_SAR.3 Selectable Audit Review

Hierarchical to: No other components.

FAU_SAR.3.1 The TSF shall provide the ability to apply [***sorting***] of audit data based on [***time or event type/Class as seen in column 4 of the Audit log***].

Dependencies: FAU_SAR.1 Audit review

7.1.1.6 FAU_SEL.1 Selective Audit

Hierarchical to: No other components.

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

a) [***object identity, user identity***]

b) [***class name, group name, program name, access rights, authorization result***]

Dependencies: FAU_GEN.1 Audit data generation

FMT_MTD.1 Management of TSF data

7.1.1.7 FAU_STG.1 Protected Audit Trail Storage

Hierarchical to: No other components.

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.

Dependencies: FAU_GEN.1 Audit data generation

7.1.2 Class FCS: Cryptographic Support

The Cryptography used in this product has not been FIPS certified nor has it been analyzed or tested to conform to cryptographic standards during this evaluation. All cryptography has only been asserted as tested by the vendor.

7.1.2.1 FCS_CKM.1 Cryptographic Key Generation

Hierarchical to: No other components.

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [***RSA***] and specified cryptographic key sizes [***1024-bits***] that meet the following: [***RFC 2313***].

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or

FCS_COP.1 Cryptographic operation]

FCS_CKM.4 Cryptographic key destruction

Application note: This SFR supports key generation for TLS.

7.1.2.2 FCS_CKM.4 Cryptographic Key Destruction

Hierarchical to: No other components.

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [***overwrite method***] that meets the following: [***none***].

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation]

Application note: This SFR supports key destruction for TLS.

7.1.2.3 FCS_COP.1 Cryptographic Operation

Hierarchical to: No other components.

FCS_COP.1.1 The TSF shall perform [***encryption and decryption***] in accordance with a specified cryptographic algorithm [***AES***] and cryptographic key sizes [***128-bits***] that meet the following: [***RFC 3268***].

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

Application note: *This SFR supports the symmetric key usage for TLS.*

7.1.3 Class FDP: User Data Protection

7.1.3.1 FDP_ACC.1 Subset Access Control

Hierarchical to: No other components.

FDP_ACC.1.1 The TSF shall enforce the ***[Access Control rules and policies***] on [***Accessors and resource objects and operations as listed in Table 7-4 Access Control Predefined Classes***].

Dependencies: FDP_ACF.1 Security attribute based access control

Class	Class Type	Description	ACLs Used	Valid Access Values	Operation Allowed
PROGRAM	Resource	Each record in this class defines a trusted program that can be used with conditional access rules. Trusted programs are setuid/setgid programs that are monitored by Seoswd to ensure they are	ACL NACL	Execute	Execute a program

Class	Class Type	Description	ACLs Used	Valid Access Values	Operation Allowed
		not tampered with.			
TERMINAL	Resource	Each record in this class defines a terminal—a device from which an administrator can log in.	ACL NACL PACL	Read	Log into terminal
				Write	Administer terminal
PROCESS	Resource	Each record in this class defines an executable file.	ACL NACL PACL	Read	Kill the process
FILE	Resource	Allows for the protection of a file, a directory, or a file name mask.	ACL NACL PACL	Changepermissions, sec	Modify the ACL of the resource
				Chdir	Access to the directory with the equivalent of read and execute permissions
				Chmod	Change file systems modes.
				Chown	Change the owner of the record/resource
				Control	Perform all valid operations except delete and rename.
				Create	Create records in this class.
				Delete	Delete records/resources in this class

Class	Class Type	Description	ACLs Used	Valid Access Values	Operation Allowed
				Execute	Execute a program. The accessor must also have read access
				Read	Access a resource without changing it. To rename a file, a user must have delete access to the source and rename access to the target. The audit log reflects this order of events.
				Rename	Rename a record in this class.
				Update	Perform the combined operations of Read, Write, and Execute.
				Utime	Change the modification time of a file.
				Write	Change the file or directory/modify the resource.
				All	Perform all valid operations for this class
				None	Deny all valid operations for this class
USER	Accessor	Each record in this class defines an	N/A	N/A	N/A

Class	Class Type	Description	ACLs Used	Valid Access Values	Operation Allowed
		internal user			
GROUP	Accessor	Each record in this class defines an internal group	N/A	N/A	N/A
SUDO	Resource	Each record in this class defines a command that is authorized via sudo	ACL NACL PACL	Execute	Execute the command
SURROGATE	Resource	Each record in this class contains access rules for an accessor, which define who can use that accessor as a surrogate.	ACL NACL PACL	Execute	Surrogate to the user
XUSER	Accessor	Each record in this class defines an enterprise user to CA Access Control	N/A	N/A	N/A
XGROUP	Accessor	Each record in this class defines an enterprise group to CA Access Control	N/A	N/A	N/A

Table 7-4: Access Control Predefined Classes

7.1.3.2 FDP_ACF.1 (1) Security Attribute Based Access Control

Hierarchical to:	No other components.
FDP_ACF.1.1 (1)	The TSF shall enforce the [***Access Control rules and polices***] to objects based on the following: [***Objects with attributes listed in Table 7-5 Security Attributes***].

Object	Security Attribute	Seosdb Record Attribute
Resource Record	Object Identity	RESOURCENAME
	Resource Class	CLASSNAME
	Resource Group Membership	GROUPS
	Resource Owner	OWNER
	Access Control List	ACL
	Negative Access Control List	NACL
	Program Access Control List	PACL
	Default Access	UACC
	Un-trusted Program	UNTRUST
Accessor Record	User Identity	UID
	User Group Membership	GROUPS
	Authority Attributes	OBJ_TYPE
	Accessor's Access Control Lists	REVACL

Table 7-5: Security Attributes

FDP_ACF.1.2 (1) The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [

- ***An accessor is explicitly granted access to a resource by association of the username.***
- ***An accessor is implicitly granted access to a resource if he/she belongs to a group which has been granted access.***
- ***A rule or set of rules specifies the resources protected by the policy***].

FDP_ACF.1.3 (1) The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [

- ***An accessor is explicitly granted access to a resource by an Access Control List.***
- ***An accessor is explicitly granted access to a resource by a Program Access Control List.***
- ***An accessor is explicitly granted access to a resource by the owner parameter of the record.***
- ***An accessor is explicitly granted access to a resource according to the default access field of the resource's record***].

FDP_ACF.1.4 (1) The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [***Negative Access Control List, or a rule denying access to an explicit resource by a specific accessor***].

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialization

Application Note: *The root account on UNIX is subject to authorization restriction like all other users of the system.*

Application Note: *The term accessor for this SFR is applicable to any user accessing the machine the TOE protects, affectively making them an end user of the TOE. This includes administrators which are accessing the machine to view the audit logs.*

7.1.3.3 FDP_ACF.1 (2) Security Attribute Based Access Control

Hierarchical to: No other components.

FDP_ACF.1.1 (2) The TSF shall enforce the [***Access Control rules and policies***] to objects based on the following: [***administrators are allowed to manage the TOE by performing the operations as listed in Table 7-6 Global and Group Authorization Attributes for Administrators***].

FDP_ACF.1.2 (2) The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [***only Administrators with the attributes listed in Table 7-6 Global and Group Authorization Attributes for Administrators are allowed to perform the operations listed in the table***].

FDP_ACF.1.3 (2) The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [***an administrator must have WRITE access from the approved terminal, approval of terminal name or IP address in the TERMINAL class, and an administrator attribute must be included in the administrator's respective record in Seosdb and a rule must be written depicting his access to the TOE in order to manage the TOE through selang***].

Application Note: *The TOE contains a superuser account by default which is mapped to the user that installs the TOE.*

FDP_ACF.1.4 (2) The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [***administrator has not been assigned one of the administrator attributes listed in Table 7-6 Global and Group Authorization Attributes for Administrators, the administrator does not have WRITE access from the approved terminal, and the terminal name/IP address has not been approved by the TERMINAL class***].

Attribute	Privileges/Operations	Restrictions
ADMIN	Read/modify record properties	The last admin with the ADMIN attribute cannot be deleted.
	Create new records	The ADMIN attribute cannot be removed from the last admin with the ADMIN attribute.

Attribute	Privileges/Operations	Restrictions
	Delete records	Unable to update the audit mode without the AUDITOR attribute.
	Can set root to be a non-ADMIN administrator.	Unable to delete root.
	Set up Policy Model	None
	Set up password policies	Administrator with AUDITOR or OPERATOR attribute has read-only access to the policies.
AUDITOR	Set up audit policy	Unable to perform any other operations than those listed.
	Modify/set the audit mode for existing records	
OPERATOR	READ access to all files	Unable to perform any other operations than those listed.
	List information in Seosdb	
	Run backup jobs	
	Shut down the TOE	
PWMANAGER	Change passwords of any user	Unable to change the number of grace logins.
		Unable to change the password interval of another accessor.
		Unable to change general password rules.
GROUP-ADMIN	Read/modify record properties	Unable to make resources inaccessible to themselves
	Create new records	Unable to assign a security level that is higher than their own security level.
	Delete records	Unable to assign a security category or security label that they do not have.
	Connect users to a group or separate users from a group	Unable to delete the root administrator from Seosdb.

Attribute	Privileges/Operations	Restrictions
		Unable to delete the only administrator with the ADMIN attribute in Seosdb.
		Unable to remove the ADMIN attribute from the record of the last ADMIN administrator in Seosdb.
		Unable to set the ADMIN, AUDITOR, OPERATOR, PWMANAGER, and SERVER authorization attributes for any administrator.
		Those without the AUDITOR attribute cannot update the audit mode.
GROUP-AUDITOR	List the properties of any audit record within the group scope	Unable to perform any other operations than those listed.
	Set the audit mode for any record within the group scope	
GROUP-OPERATOR	List the properties of any record within the group scope	Unable to perform any other operations than those listed.
GROUP-PWMANAGER	Change the password of any user within the group scope	Unable to perform any other operations than those listed.

Table 7-6: Global and Group Authorization Attributes for Administrators

7.1.4 Class FIA: Identification and Authentication

7.1.4.1 FIA_ATD.1 User Attribute Definition

Hierarchical to: No other components.

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [***attributes of the USER class as listed in Table 7-7 Security Attributes of USER class***].

Dependencies: No dependencies.

User Security Attribute	Definition
APPLS	Displays the list of applications that the accessor is authorized to access.
AUDIT_MODE	Defines the activities that CA Access Control records in the audit log.
FULLNAME	Defines the full name associated with an accessor.
GAPPLS	Indicates the list of application groups that the accessor is authorized to access.
GROUPS	Displays the list of groups that the accessor or administrator belongs to. This property also contains any group authorities, such as group administration authority (GROUP-ADMIN), assigned to the administrator for each group the administrator belongs to.
LAST_ACC_TERM	Displays the terminal from which the last login was performed.
LAST_ACC_TIME	Displays the date and time of the last login.
LOGININFO	Defines the information needed to log the user in to a specific application and audit data. LOGININFO contains a separate list for each application that the user is authorized to access.
NOTIFY	Defines the accessor to be notified when a resource or accessor generates an audit event.

User Security Attribute	Definition
OBJ_TYPE	Specifies the administrator authority attributes. An administrator can have are one or more of the following authority attributes: ADMIN, AUDITOR, PWMANAGER, OPERATOR.
OWNER	Defines the user or group that owns the record.
PGMINFO	Defines the program information automatically generated by CA Access Control.
REVACL	Displays the access control lists of the accessor.

Table 7-7: Security Attributes of the USER class

7.1.4.2 FIA_SOS.1 Verification of Secrets

Hierarchical to: No other components.

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [***password length, age, and composition requirements***].

Dependencies: No dependencies.

Application Note: Password policies are applied to accessor OS user accounts and profiles via the SEOS class and enforced via the sepass utility on each endpoint.

7.1.4.3 FIA_UID.2 User identification before any action

Hierarchical to: FIA_UID.1 Timing of identification

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

Dependencies: No dependencies.

Application Note: *This instantiates the TOE's ability to maintain the original identity a user users to access the host OS to prevent sudo operations circumventing rules.*

7.1.5 Class FMT: Security Management

7.1.5.1 FMT_MOF.1 Management of Security Functions Behavior

Hierarchical to: No other components.

FMT_MOF.1.1 The TSF shall restrict the ability to [***modify the behavior of***] the functions [***as listed in Table 7-2 Audit Properties***] to [***administrators with the AUDITOR attribute and/or administrators with the GROUP-AUDITOR attribute***].

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

7.1.5.2 FMT_MSA.1 Management of Security Attributes

Hierarchical to: No other components.

FMT_MSA.1.1 The TSF shall enforce the [***Access Control rules and policies***] to restrict the ability to [***Perform actions as defined in the Privileges/Operations column of table 7-6, on***] the security attributes [***listed as modifiable in Table 7-8 Class Properties***] to [***Administrators with attributes listed in the attribute column of table 7-6***].

Dependencies: [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]

FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

Class Property	Class	Modifiable vs. Informational
ACL	PROGRAM, TERMINAL, PROCESS, FILE, SUDO	Modifiable
APPLIST	USER, XUSER	Modifiable. Used by CA SSO
APPLIST_TIME	USER, XUSER	Modifiable. Used by CA SSO
AUDIT_MODE	USER, GROUP, XGROUP, XUSER	Modifiable
BADPASSWD	USER, XUSER	Modifiable. Used by CA SSO
BLOCKRUN	PROGRAM	Modifiable
CALACL	FILE, SUDO	Modifiable
CALENDAR	FILE, SUDO	Modifiable
CATEGORY	FILE, USER, SUDO	Modifiable
COMMENT	PROGRAM, TERMINAL, PROCESS, FILE, USER, GROUP, XGROUP, XUSER, SUDO	Modifiable
COUNTRY	USER	Modifiable
CREATE_TIME	SUDO	Modifiable
DAYTIME	PROGRAM, PROCESS, TERMINAL, FILE, USER, GROUP, XGROUP, XUSER	Modifiable
EMAIL	USER	Modifiable
EXPIRE DATE	USER, GROUP	Modifiable
FULLNAME	USER, GROUP, XGROUP, XUSER	Modifiable
GAPPLS	USER, GROUP, XGROUP, XUSER	Modifiable
GRACELOGIN	USER, XUSER	Modifiable
GROUP_MEMBER	USER, GROUP, XGROUP	Modifiable
GROUPS	SUDO	Modifiable

Class Property	Class	Modifiable vs. Informational
INACTIVE	USER, GROUP, XUSER	Modifiable
LAST_ACC_TERM	USER, XUSER	Modifiable
LAST_ACC_TIME	USER, XUSER	Modifiable
LOCALAPPS	USER, XUSER	Modifiable. Used by CA SSO
LOCATION	USER, XUSER	Modifiable
LOGININFO	USER, XUSER	Modifiable. Used by CA SSO
LOGSHIFT	USER, XUSER	Modifiable
MAXLOGINS	USER, XUSER	Modifiable
MEMBER_OF	USER, GROUP, XGROUP	Modifiable
MIN_TIME	USER, XUSER	Modifiable
NACL	PROGRAM, TERMINAL, PROCESS, FILE, SUDO	Modifiable
NOTIFY	PROGRAM, TERMINAL, PROCESS, FILE, USER, XUSER, SUDO	Modifiable
OBJ_TYPE	USER, XUSER	Modifiable
OIDCRDDATA	USER, XUSER	Modifiable. Used by CA SSO
OLD_PASSWD	USER, XUSER	Modifiable
ORG_UNIT	USER, XUSER	Modifiable
ORGANIZATION	USER, XUSER	Modifiable
OWNER	PROGRAM, TERMINAL, PROCESS, FILE, USER, GROUP, XGROUP, SUDO	Modifiable
PACL	PROGRAM, TERMINAL, PROCESS, FILE, SUDO	Modifiable
PASSWD_A_C_W	USER, XUSER	Modifiable
PASSWD_INT	USER, XUSER	Modifiable
PASSWD_L_A_C	USER, XUSER	Modifiable
PASSWD_L_C	USER, XUSER	Modifiable

Class Property	Class	Modifiable vs. Informational
PASSWDRULES	GROUP, SEOS	Modifiable
PASSWORDREQ	SUDO	Modifiable
PGMINFO	PROGRAM, USER, XUSER	Modifiable
PHONE	USER, XUSER	Modifiable
POLICYMODEL	USER, GROUP, SUDO	Modifiable
PROFUSR	GROUP, XGROUP	Modifiable
PWD_AUTOGEN	USER, GROUP, XGROUP, XUSER	Modifiable. Used by CA SSO
PWD_SYNC	USER, GROUP, XGROUP, XUSER	Modifiable. Used by CA SSO
PWPOLICY	GROUP, XGROUP	Modifiable. Used by CA SSO
RAUDIT	PROGRAM, TERMINAL, PROCESS, FILE	Modifiable
RESUME_DATE	USER, GROUP, XUSER	Modifiable
REVACL	GROUP, XGROUP, XUSER	Modifiable
REVOKE_COUNT	USER, XUSER	Modifiable. Used by CA SSO
SCRIPT_VARS	USER, XUSER	Modifiable. Used by CA SSO
SECLABEL	FILE, USER, SUDO	Modifiable
SECLEVEL	FILE, USER, SUDO	Modifiable
SESSION_GROUP	USER, XUSER	Modifiable. Used by CA SSO
SHELL	GROUP, XGROUP	Modifiable
SHIFT	USER, XUSER	Modifiable. Used by CA SSO
SUBGROUP	GROUP, XGROUP	Modifiable
SUPGROUP	GROUP, XGROUP	Modifiable
SUSPEND DATE	USER, GROUP	Modifiable
SUSPEND WHO	USER, GROUP	Modifiable
TARGUSR	SUDO	Modifiable
UACC	PROGRAM, TERMINAL, PROCESS, FILE, SUDO	Modifiable

Class Property	Class	Modifiable vs. Informational
UALIAS	USER	Modifiable
UNTRUST	PROGRAM, FILE	Modifiable
UPDATE_TIME	PROGRAM, TERMINAL, PROCESS, FILE, USER, GROUP, XGROUP, SUDO	Informational
UPDATE_WHO	PROGRAM, TERMINAL, PROCESS, FILE, USER, GROUP, XGROUP, SUDO	Informational
USERLIST	GROUP, XGROUP	Modifiable
WARNING	PROGRAM, TERMINAL, PROCESS, FILE, SUDO	Modifiable

Table 7-8: Class Properties

Application Note: *Management functions can be performed against a single endpoint or distributed simultaneously to multiple endpoints by using the Policy Model.*

Application Note: *This SFR also applies to the ability of end users to change their own password, which is mediated by sepass.*

7.1.5.3 FMT_MSA.3 Static Attribute Initialization

Hierarchical to: No other components.

FMT_MSA.3.1 The TSF shall enforce the [***Access Control rules and policies***] to provide [***restrictive***] default values for security attributes that are used to enforce the SFP.

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

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

7.1.5.4 FMT_REV.1 Revocation

Hierarchical to: No other components.

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

FMT_REV.1.2 The TSF shall enforce the rules [***revocation of an administrator with the following attributes: ADMIN, AUDITOR, OPERATOR, PWMANAGER, GROUP-ADMIN, GROUP-AUDITOR, GROUP-OPERATOR, GROUP-PWMANAGER***]

Dependencies: FMT_SMR.1 Security roles

Application Note: *The last administrator with the ADMIN attribute cannot revoke the ADMIN attribute from itself.*

7.1.5.5 FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

FMT_MTD.1.1 The TSF shall restrict the ability to [***query, modify or delete***] the [***records in Seosdb as listed in Table 7-6 Global and Group Authorization Attributes for Administrators***] to [***administrators with the ADMIN, AUDITOR, PWMANAGER, OPERATOR, GROUP-ADMIN, GROUP-AUDITOR, GROUP-PWMANAGER, and/or GROUP-OPERATOR attribute***].

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

Application Note: *Management functions can be performed against a single endpoint or distributed simultaneously to multiple endpoints by using the Policy Model.*

7.1.5.6 FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [***as described in Table 7-6 Global and Group Authorization Attributes for Administrators***].

Dependencies: No dependencies.

Application Note: *Management functions can be performed against a single endpoint or distributed simultaneously to multiple endpoints by using the Policy Model.*

7.1.5.7 FMT_SMR.2 Restrictions on Security Roles

Hierarchical to: FMT_SMR.1 Security Roles

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.2.1 The TSF shall maintain the roles [***end user and administrator***].

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 [That only administrators with the ADMIN, AUDITOR, OPERATOR, PWMANGER, GROUP-ADMIN, GROUP-AUDITOR, GROUP-OPERATOR, GROUP-PWMANAGER attributes are able to perform the privileges associated with those attributes defined in table 7-6] are satisfied.

Application Note:	*Management functions can be performed against a single endpoint or distributed simultaneously to multiple endpoints by using the Policy Model.*
Application Note:	*The only management functions afforded to end users are the ability to change their own password on the local OS via sepass.*

7.1.6 Class FPT: Protection of the TSF

7.1.6.1 FPT_FLS.1 Failure with Preservation of Secure State

Hierarchical to:	No other components.
FPT_FLS.1.1	The TSF shall preserve a secure state when the following types of failures occur: [***when the Seosd, Seagent, and/or the Seoswd daemons go down***].
Dependencies:	No dependencies.

7.1.6.2 FPT_ITT.1 Basic Internal TSF Data Transfer Protection

Hierarchical to:	No other components.
FPT_ITT.1.1	The TSF shall protect the TSF data from [***disclosure, modification***] when it is transmitted between separate parts of the TOE.
Dependencies:	No dependencies.
Application Note:	*This addresses the confidentiality of TSF data when selang is used to manage a seosdb or PMDB remotely and when sepmdd is updating distributed subscriber seosdb instances as a result of PMDB updates.*

7.1.7 Class FRU: Resource Utilization

7.1.7.1 FRU_FLT.1 Degraded Fault Tolerance

Hierarchical to:	No other components.
FRU_FLT.1.1	The TSF shall ensure the operation of [***access control to resources***] when the following failures occur: [***when the Seosd, Seagent, and/or the Seoswd daemons go down***].
Dependencies:	FPT_FLS.1 Failure with preservation of secure state

7.1.8 Class FTA: TOE Access

7.1.8.1 FTA_TSE.1 TOE Session Establishment

Hierarchical to: No other components.

FTA_TSE.1.1 The TSF shall be able to deny session establishment based on [***administrators may not use selang to manage the TOE unless they have one or more of the following attributes: ADMIN, AUDITOR, PWMANAGER, OPERATOR, GROUP-ADMIN, GROUP-AUDITOR, GROUP-PWMANAGER, GROUP-OPERATOR, and the following applies: the administrator does not have WRITE access from the approved terminal, and the terminal name/IP address of the terminal is not approved in the TERMINAL class***].

Dependencies: No dependencies.

7.1.9 Class FTP: Trusted Path/Channels

7.1.9.1 FTP_TRP.1 Trusted Path

Hierarchical to: No other components.

FTP_TRP.1.1 The TSF shall provide a communication path between itself and [***remote***] 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 or disclosure.

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

FTP_TRP.1.3 The TSF shall require the use of the trusted path for [***user authorization, management of the TOE through selang***].

Dependencies: No dependencies.

Application Note: *The term "remote users" applies to administrators*

7.2 Security Functional Requirements for the Operational Environment

There are no security functional requirements for the Operational Environment in this ST beyond the extended requirements.

7.3 Operations Defined

The notation, formatting, and conventions used in this security target (ST) are consistent with version 3.1 of the Common Criteria for Information Technology Security Evaluation. All of the components in this ST are taken directly from Part 2 of the CC except the ones noted with “_EXT” in the component name. Font style and clarifying information conventions were developed to aid the reader.

The CC permits four functional component operations: assignment, iteration, selection, and refinement to be performed on functional requirements. These operations are defined in Common Criteria, Part 1 as:

7.3.1 Assignments Made

An assignment allows the specification of parameters and is specified by the ST author in [***italicized bold text***].

7.3.2 Iterations Made

An iteration allows a component to be used more than once with varying operations and is identified with the iteration number within parentheses after the short family name.

7.3.3 Selections Made

A selection allows the specification of one or more items from a list and is specified by the ST author in [***italicized bold text***].

7.3.4 Refinements Made

A refinement allows the addition of details and is identified with "Refinement:" right after the short name. Additions to the CC text are specified in ***italicized bold and underlined text***.

8 Security Assurance Requirements

This section identifies the Security Assurance Requirement components met by the TOE. These assurance components meet the requirements for EAL3 augmented with ALC_FLR.1 and ASE_TSS.2.

8.1 Security Architecture

8.1.1 Security Architecture Description (ADV_ARC.1)

ADV_ARC.1.1D The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2D The developer shall design and implement the TSF so that it is able to protect itself from tampering by un-trusted active entities.

ADV_ARC.1.3D The developer shall provide a security architecture description of the TSF.

ADV_ARC.1.1C The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2C The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3C The security architecture description shall describe how the TSF initialization process is secure.

ADV_ARC.1.4C The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5C The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

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

8.1.2 Functional Specification with Complete Summary (ADV_FSP.3)

ADV_FSP.3.1D The developer shall provide a functional specification.

ADV_FSP.3.2D The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.3.1C The functional specification shall completely represent the TSF.

ADV_FSP.3.2C The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.3.3C The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.3.4C For each SFR-enforcing TSFI, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.

ADV_FSP.3.5C For each SFR-enforcing TSFI, the functional specification shall describe direct error messages resulting from SFR-enforcing actions and exceptions associated with invocation of the TSFI.

ADV_FSP.3.6C The functional specification shall summarize the SFR-supporting and SFR-non-interfering actions associated with each TSFI.

ADV_FSP.3.7C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_FSP.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.3.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

8.1.3 Architectural Design (ADV_TDS.2)

ADV_TDS.2.1D The developer shall provide the design of the TOE.

ADV_TDS.2.2D The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

ADV_TDS.2.1C The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.2.2C The design shall identify all subsystems of the TSF.

ADV_TDS.2.3C The design shall describe the behavior of each SFR non interfering subsystem of the TSF in detail sufficient to determine that it is SFR non-interfering.

ADV_TDS.2.4C The design shall describe the SFR-enforcing behavior of the SFR enforcing subsystems.

ADV_TDS.2.5C The design shall summarize the SFR-supporting and SFR-non interfering behavior of the SFR-enforcing subsystems.

ADV_TDS.2.6C The design shall summarize the behavior of the SFR-supporting subsystems.

ADV_TDS.2.7C The design shall provide a description of the interactions among all subsystems of the TSF.

ADV_TDS.2.8C The mapping shall demonstrate that all TSFIs trace to the behavior described in the TOE design that they invoke.

ADV_TDS.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_TDS.2.2E The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

8.2 Guidance Documents

8.2.1 Operational User Guidance (AGD_OPE.1)

AGD_OPE.1.1D The developer shall provide operational user guidance.

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

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

8.2.2 Preparative Procedures (AGD_PRE.1)

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

AGD_PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

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

AGD_PRE.1.2E The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

8.3 Lifecycle Support

8.3.1 Authorization Controls (ALC_CMC.3)

ALC_CMC.3.1D The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.3.2D The developer shall provide the CM documentation.

ALC_CMC.3.3D The developer shall use a CM system.

ALC_CMC.3.1C The TOE shall be labeled with its unique reference.

ALC_CMC.3.2C The CM documentation shall describe the method used to uniquely identify the configuration items.

ALC_CMC.3.3C The CM system shall uniquely identify all configuration items.

ALC_CMC.3.4C The CM system shall provide measures such that only authorized changes are made to the configuration items.

ALC_CMC.3.5C The CM documentation shall include a CM plan.

ALC_CMC.3.6C The CM plan shall describe how the CM system is used for the development of the TOE.

ALC_CMC.3.7C The evidence shall demonstrate that all configuration items are being maintained under the CM system.

ALC_CMC.3.8C The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.

ALC_CMC.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.3.2 CM Scope (ALC_CMS.3)

ALC_CMS.3.1D The developer shall provide a configuration list for the TOE.

ALC_CMS.3.1C The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; and the implementation representation.

ALC_CMS.3.2C The configuration list shall uniquely identify the configuration items.

ALC_CMS.3.3C For each TSF relevant configuration item, the configuration list shall indicate the developer of the item. Evaluator action elements:

ALC_CMS.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.3.3 Delivery Procedures (ALC_DEL.1)

ALC_DEL.1.1D The developer shall document and provide procedures for delivery of the TOE or parts of it to the consumer.

ALC_DEL.1.2D The developer shall use the delivery procedures.

ALC_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

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

8.3.4 Identification of Security Measures (ALC_DVS.1)

ALC_DVS.1.1D The developer shall produce and provide development security documentation.

ALC_DVS.1.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

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

ALC_DVS.1.2E The evaluator shall confirm that the security measures are being applied.

8.3.5 Life-cycle Definition (ALC_LCD.1)

ALC_LCD.1.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC_LCD.1.2D The developer shall provide life-cycle definition documentation.

ALC_LCD.1.1C The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

ALC_LCD.1.2C The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE. Evaluator action elements:

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

8.3.6 Basic Flaw Remediation (ALC_FLR.1)

ALC_FLR.1.1D The developer shall document and provide flaw remediation procedures addressed to TOE developers. Content and presentation elements:

ALC_FLR.1.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.1.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.1.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.1.4C The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users. Evaluator action elements:

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

8.4 Security Target Evaluation

8.4.1 Conformance Claims (ASE_CCL.1)

ASE_CCL.1.1D The developer shall provide a conformance claim.

ASE_CCL.1.2D The developer shall provide a conformance claim rationale.

ASE_CCL.1.1C The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.

ASE_CCL.1.2C The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.

ASE_CCL.1.3C The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.

ASE_CCL.1.4C The CC conformance claim shall be consistent with the extended components definition.

ASE_CCL.1.5C The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.

ASE_CCL.1.6C The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.

ASE_CCL.1.7C The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.

ASE_CCL.1.8C The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.

8.4.2 Extended Components Definition (ASE_ECD.1)

ASE_ECD.1.1D The developer shall provide a statement of security requirements.

ASE_ECD.1.2D The developer shall provide an extended components definition.

ASE_ECD.1.1C The statement of security requirements shall identify all extended security requirements.

ASE_ECD.1.2C The extended components definition shall define an extended component for each extended security requirement.

ASE_ECD.1.3C The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.

ASE_ECD.1.4C The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.

ASE_ECD.1.5C The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.

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

ASE_ECD.1.2E The evaluator shall confirm that no extended component can be clearly expressed using existing components.

8.4.3 ST Introduction (ASE_INT.1)

ASE_INT.1.1D The developer shall provide an ST introduction.

ASE_INT.1.1C The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.

ASE_INT.1.2C The ST reference shall uniquely identify the ST.

ASE_INT.1.3C The TOE reference shall identify the TOE.

ASE_INT.1.4C The TOE overview shall summarize the usage and major security features of the TOE.

ASE_INT.1.5C The TOE overview shall identify the TOE type.

ASE_INT.1.6C The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.

ASE_INT.1.7C The TOE description shall describe the physical scope of the TOE.

ASE_INT.1.8C The TOE description shall describe the logical scope of the TOE.

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

ASE_INT.1.2E The evaluator shall confirm that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

8.4.4 Security Objectives (ASE_OBJ.2)

ASE_OBJ.2.1D The developer shall provide a statement of security objectives.

ASE_OBJ.2.2D The developer shall provide a security objectives rationale.

ASE_OBJ.2.1C The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.

ASE_OBJ.2.2C The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.

ASE_OBJ.2.3C The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.

ASE_OBJ.2.4C The security objectives rationale shall demonstrate that the security objectives counter all threats.

ASE_OBJ.2.5C The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.

ASE_OBJ.2.6C The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.

ASE_OBJ.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.4.5 Security Requirements (ASE_REQ.2)

ASE_REQ.2.1D The developer shall provide a statement of security requirements.

ASE_REQ.2.2D The developer shall provide a security requirements rationale.

ASE_REQ.2.1C The statement of security requirements shall describe the SFRs and the SARs.

ASE_REQ.2.2C All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.

ASE_REQ.2.3C The statement of security requirements shall identify all operations on the security requirements.

ASE_REQ.2.4C All operations shall be performed correctly.

ASE_REQ.2.5C Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.

ASE_REQ.2.6C The security requirements rationale shall trace each SFR back to the security objectives for the TOE.

ASE_REQ.2.7C The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.

ASE_REQ.2.8C The security requirements rationale shall explain why the SARs were chosen.

ASE_REQ.2.9C The statement of security requirements shall be internally consistent.

ASE_REQ.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.4.6 Security Problem Definition (ASE_SPD.1)

ASE_SPD.1.1D The developer shall provide a security problem definition.

ASE_SPD.1.1C The security problem definition shall describe the threats.

ASE_SPD.1.2C All threats shall be described in terms of a threat agent, an asset, and an adverse action.

ASE_SPD.1.3C The security problem definition shall describe the OSPs.

ASE_SPD.1.4C The security problem definition shall describe the assumptions about the operational environment of the TOE.

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

8.4.7 TOE Summary Specification (ASE_TSS.2)

ASE_TSS.2.1D The developer shall provide a TOE summary specification.

ASE_TSS.2.1C The TOE summary specification shall describe how the TOE meets each SFR.

ASE_TSS.2.2C The TOE summary specification shall describe how the TOE protects itself against interference and logical tampering.

ASE_TSS.2.3C The TOE summary specification shall describe how the TOE protects itself against bypass.

ASE_TSS.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_TSS.2.2E The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

8.5 Tests

8.5.1 Analysis of Coverage (ATE_COV.2)

ATE_COV.2.1D The developer shall provide an analysis of the test coverage.

ATE_COV.2.1C The analysis of the test coverage shall demonstrate the correspondence between the tests in the test documentation and the TSFIs in the functional specification.

ATE_COV.2.2C The analysis of the test coverage shall demonstrate that all TSFIs in the functional specification have been tested.

ATE_COV.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.5.2 Basic Design (ATE_DPT.1)

ATE_DPT.1.1D The developer shall provide the analysis of the depth of testing.

ATE_DPT.1.1C The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems in the TOE design.

ATE_DPT.1.2C The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.

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

8.5.3 Functional Tests (ATE_FUN.1)

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation

ATE_FUN.1.1C The test documentation shall consist of test plans, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.3C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4C The actual test results shall be consistent with the expected test results.

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

8.5.4 Independent Testing (ATE_IND.2)

ATE_IND.2.1D The developer shall provide the TOE for testing.

ATE_IND.2.1C The TOE shall be suitable for testing.

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

ATE_IND.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

ATE_IND.2.3E The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

8.6 Vulnerability Assessment

8.6.1 Vulnerability Analysis (AVA_VAN.2)

AVA_VAN.2.1D The developer shall provide the TOE for testing.

AVA_VAN.2.1C The TOE shall be suitable for testing.

AVA_VAN.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.2.2E The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.2.3E The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.

AVA_VAN.2.4E The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

9 TOE Summary Specification

9.1 Evaluated Components of the TOE

CA Access Control r12 SP1 is comprised of multiple components. They are as follows:

- Seosd
- Seoswd (Watchdog)
- Seagent (Agent)
- Seosdb (Database)
- SEOS_syscall
- Selang
- Policy Model Database (PMDB)
- Seos.audit
- Seaudit
- Sepmdd
- Sepass

9.1.1 Components at the Kernel Level

The Kernel level in the operational environment contains SEOS_syscall which is the CA Access Control interception module. This interception module is located in the SEOSDIR bin directory. The SEOS_load utility controls this kernel module, which must be loaded before running many CA Access Control utilities.

SEOS_syscall hooks into the operating system on SEOS_syscall kernel module load time or after boot time and intercepts all access and privilege requests. SEOS_syscall works in conjunction with Seagent and Seosd to allow or deny access to the TOE by administrators and end users on UNIX operating systems.

SEOS_syscall protects all of the CA Access Control resources by performing the following tasks:

- Intercepting every request to access a resource or terminate a process.
- Passing these requests to Seosd and receiving the decision of Seosd whether the request should be granted or denied.
- Seosd returns an answer through SEOS_syscall, for OS access, if allowed, and then the request is passed to the original OS function. If access is not granted SEOS_syscall will return an error to the calling function so it fails the request.

9.1.2 OS Level Components

9.1.2.1 Seosd

Seosd is the enforcement service for CA Access Control. This component is responsible for making all decisions on what's allowed or denied on the TOE and the machine the TOE protects, based on the records stored in Seosdb. This includes but is not limited to the identification, authorization of accessors, the creation of rules and their inclusion in Seosdb, and the ability of subjects to perform operations on objects. In addition, Seosd captures all events according to the audit policy, and all security events on the TOE in audit records and writes the information to the seos.audit file. When a user session is initiated, seosd stores the UID of the user and uses it to identify the user for all actions taken against protected resources. The reason the UID is captured is so that an accessor cannot circumvent rules using the su command (unless a SUDO record explicitly authorizes it).

It is important to note, CA Access Control relies on the underlying OS to conduct authentication. An end user must authenticate to the underlying OS to confirm their identity prior to accessing the resources which the TOE protects. Then CA Access Control can *authorize* the end user to access a protected resource based upon the UID. Since the password is not used here for access to a protected resource, Seosd is not authenticating the end user but instead is simply authorizing or denying access.

A daemon is a program that runs in the background in UNIX environment. Seosd is the main CA Access Control authorization daemon and is a binary program on UNIX. The installation location can be found from the seos.ini file. Seosd makes the runtime decisions required to grant or deny access to a resource.

Only the accessors with the UNIX root account can invoke seosd, and only an administrator with the ADMIN or OPERATOR attribute can shut it down. The command to start seosd is "seload" or "seosd", while the command to shut it down is "secons –s" or "secons –sk."

Seosd opens, reads, and updates Seosdb. No other process can access Seosdb while Seosd is running except under read access. Seosd also blocks any write, delete, or rename access to critical files, such as the CA Access Control audit and, optionally, the CA Access Control binary files. During startup, seosd also invokes Seagent, the CA Access Control Agent daemon.

Seosd can get policy queries and updates only after the Seagent and Seoswd daemons are initialized. After initialization, these three daemons maintain a type of handshaking protocol to ensure they are all alive and responding. In the UNIX environment Seoswd only restarts Seosd if it has gone down and Seosd restarts Seagent if it has gone down. If one of these three daemons go down, the TOE will still provide access control to resources for end users.

9.1.2.2 Seoswd (Watchdog)

Seoswd monitors the file information and digital signatures of programs that are defined in Seosdb as trusted programs. Monitoring is performed in the background. Seoswd automatically starts Seosd.

Seoswd is the Watchdog of Access Control. Seoswd monitors trusted programs to ensure they have not been modified in the local file system. All trusted programs must be listed as a trusted program in Seosdb in order to be monitored by Seoswd.

The Seoswd daemon performs the following functions:

- Seoswd monitors the programs that are defined in the PROGRAM class of Seosdb. If Seoswd detects that a program was modified, it notifies Seosd, which marks the program as untrusted. The Seosd daemon marks the program's status change to untrusted in Seosdb and creates an audit record. Depending on the value of the blockrun property for the PROGRAM that gets untrusted, the Seosd daemon either allows a program to run or does not allow it to run until it's re-trusted. If the blockrun property is set to yes, the program is not executed until it is re-trusted and is not allowed to access any file that the relevant PACL would allow. The PACL is effectively disabled until the program is re-trusted. If the blockrun property is set to no, the program is executed and the PACL for it is inactive.
- Seoswd reports several events to Seosd, which creates audit records for programs that were found to be altered.
- Seoswd allows for the specification of interval and fixed scanning schedules for trusted programs.
- Seoswd ignores any signal except SIGHUP; the Seoswd daemon cannot be killed unless Seosd is shut down first. However, if the command kill -SIGHUP pid is executed, Seoswd scans all trusted programs in Seosdb.
- Seoswd monitors Seosd and ensures it's running. If Seoswd detects a problem with Seosd, it automatically restarts it.
- The Seoswd daemon uses the system log syslog to notify the security administrators when it detects that Seosd has stopped responding. All system log messages are submitted as AUTH facility.

The seos.ini file contains several tokens that control the scanning and time-out values of Seoswd. This file also contains the documentation on these values.

There are two ways in which the Seoswd scanning mechanism can be set up by an administrator:

1. Determine a start time and then repeat scans at a given interval. For example, when checking trusted programs, Seoswd will start the first scan at *PgmTestStartTime* and will check all the trusted programs. Rescanning will take place *PgmTestInterval* seconds after the beginning of the previous scan.

2. Scanning on specific predefined times during the day.

9.1.2.3 Seagent (Agent)

Agent is responsible for communicating with CA Access Control clients through port 5249 over TLS v1.0. Additionally, it manages security for the remote administrators. Seagent also checks that the Watchdog daemon, seoswd, is running, and can restart it directly once Seagent is restarted by Seosd. The Agent is also responsible for starting sepmdd. It does not monitor sepmdd; instead, when an API call to a PMDB is made and sepmdd is not running, a call to seagent is made which starts the daemon.

9.1.2.4 Seosdb (Database)

Seosdb is the main repository (set of files) of CA Access Control and contains information on two types of objects: accessors and resources. *Accessors* are the users and groups of users in the TOE (administrators and end users). *Resources* are objects which are protected, such as file, program, or service. Each record in Seosdb describes an accessor or a resource. Seosdb also contains the rules and policies which govern accessor access to resources.

For accessors that are located in the host Operating System and are pulled from the enterprise user store for identification and authentication, a record will be created in the XUSER or XGROUP class.

9.1.2.5 Selang

The command line language, selang, allows the administrators to perform all of their functions of CA Access Control. To use selang commands, open a command prompt window and start selang. The Selang “hosts” command allows the administrator to simultaneously manage one or more remote hosts if those hosts have a TERMINAL rule which allows them this access. This is how the TOE will be managed in the evaluated configuration.

The selang utility uses the following two files for configuration options:

- seos.ini - Contains CA Access Control configuration options. This is the main configuration file for CA Access Control.
- lang.ini - Contains configuration information that selang uses.

Selang uses the lang.ini files in *one or both* of the following locations:

- The directory where the seos.ini file is located

- The administrator’s home directory

If a token is specified in only one of these lang.ini files, selang uses the value from that file. If a token is specified differently in the two lang.ini files, the value in the administrator’s home directory overrides the other one.

9.1.2.6 Seos.audit and Seaudit

Seos.audit is the audit file that’s created on each endpoint managed by the TOE. Seaudit is the application used to parse seos.audit into a human-readable format. Seaudit is protected by the TOE to restrict access to an individual with the authority to authenticate locally to the endpoint and run the application.

9.1.2.7 PMDB and Sepmdd

The Policy Model Database (PMDB) can be set up as a “master” database that stores configuration information for an endpoint. Other endpoints can subscribe to this database so that when the PMDB is updated (by a remote administrator over selang), the subscriber databases can receive the same updates. The daemon responsible for identifying when the PMDB has been updated and propagating the changes to all subscriber seagents is called sepmdd.

9.2 TOE Security Functions

This section describes the security functions provided by the TOE.

9.2.1 TP-1 TSF Domain Separation

Domain separation is the security architecture property whereby the TSF defines separate security domains for each user and for the TSF and ensures that no user process can affect the contents of a security domain of another user or of the TSF.

The underlying operating system must authenticate administrators and end users before they are allowed to access the TOE. Once the accessor has authenticated to the underlying OS (host and client) by submitting their username and password, CA Access Control fetches the user ID of the accessor and authorizes his or her access based on the rules and polices defined in Access Control. In other words, access granted to the accessor is specific to the user ID of the respective accessor. It is important to note, the accessor’s password is not used by CA Access Control for authorization to objects. Therefore, the accessor’s logon information remains separate from the TOE. All Access Control resources are protected by the TOE on the host machine, therefore no accessors outside of CA Access Control can modify the logon information when Access Control is running.

All access requests for protected resources are intercepted at the Kernel level and authorized against Seosdb. If Seosdb authorizes the requested action the response is relayed to the Kernel level which then allows the underlying OS to perform the requested action. Therefore the Kernel protects the TOE from the user space in the underlying OS.

The underlying assumption regarding the operation of CA Access Control is that it's maintained in a physically secure environment.

9.2.2 TOE Protection by the Operational Environment

End users and administrators must authenticate to their respective Operating System before they are allowed access to the TOE, and its resources. Once authenticated, the accessor's username is fetched from the underlying OS and maintained by seosd to determine access to objects. The accessor's authorization to objects is based on the rules and polices set within CA Access Control. More importantly, for end users to access protected resources, the permissions/privileges set within the underlying OS are superseded by the rules and policies set in Access Control. Therefore, TOE resources are governed by the rules and polices set within Access Control and not the underlying OS. Additionally, administrators are not allowed to manage the TOE remotely through selang unless they have WRITE access from an approved terminal, the terminal name/IP address is included in the TERMINAL class, and they have one of the administrator attributes assigned to them.

9.2.3 Self Protection Module

The CA Access Control Self-Protection Module (also referred to as the CA Access Control watchdog or Seoswd) knows which program is in control at a particular time and checks whether the program has been modified or moved since it was classified as trusted. If a trusted program is modified or moved, the program is no longer considered trusted and CA Access Control does not allow it to run.

The Watchdog daemon also continually checks whether the seosd daemon is alive and responding. (If necessary, the watchdog daemon restarts the seosd daemon.). When the seosd daemon forks, it automatically executes the seagent program to start the Agent, and seagent starts the watchdog This helps ensure that recovery of failure of one component of Access Control is secure and automatic so that its rule enforcement is continuous.

In addition, CA Access Control protects against various deliberate and accidental threats, including:

- **Kill attempts** - CA Access Control can be used to protect critical services or daemons against kill attempts.
- **Password Delinquency** - CA Access Control policies delineate rules that force accessors to create and use passwords of sufficient quality. To ensure that accessors create and use acceptable passwords, CA Access Control can set maximum and minimum lifetimes for passwords, restrict certain words, prohibit repetitive characters, and enforce other restrictions.
- **Account Management** - CA Access Control policies ensure that dormant accounts are dealt with appropriately.

9.2.4 Identification and Authorization

As illustrated in Figure 1-1 , a user of the TOE can either be a remote administrator who manages the TOE, or an end user who logs onto the machine by being authenticated by the underlying Operating System (OS). Both the remote administrator and the end user are policed by the rules and policies of the TOE.

During installation of CA Access Control, a superuser account is automatically created inside Seosdb and is used to create the initial administrator accounts. Administrators must be given one or more of the following attributes, normally ADMIN and AUDITOR, in order to manage the TOE: ADMIN, AUDITOR, PWMANAGER, or OPERATOR (see section 9.2.7.1 Global Authorization Attributes). All users with write access to the terminal are able to operate selang, however, only administrators with the ADMIN attribute have appropriate authority to update or view the access control rules and policies defined by the TOE. The selang shell is on the remote client Operating System, which communicates securely with CA Access Control's Seagent over port 5249 using TLS v1.0. A user will be denied access to administer the TOE through selang if one or more of the following is true: the user has not been assigned one of the above administrator attributes for managing the TOE, the user's username is not verified by the TOE, or the remote IP terminal is not an authorized workstation.

An end user, group of end users, or an administrator of CA Access Control is also known as an accessor. An end user is an accessor which uses the local interface (host machine without administrative privileges) to authenticate directly to the host Operating System. An administrator is an accessor which accesses the TOE using Selang. No administrative actions can be performed via the local interface.

All Access Control policy information is stored in Seosdb (Database) in the form of records. Each record in Seosdb describes an accessor or a resource. For example, the combination of the Terminal and User records contain the information such as username, password, and the remote IP terminal address or terminal name from which the administrator is allowed to access the TOE.

CA Access Control supports enterprise user stores; that is, stores for users and groups that are native to the OS. For example, it supports LDAP directory configurations. This support means that access rules can be defined for enterprise users and groups without having to synchronize or import the users and groups into Seosdb. Seosdb stores the access rules but Seosd fetches user and group information from the enterprise user stores. In the evaluated configuration only the local enterprise user store on the OS will be used.

Accessor information is stored in Seosdb either as a USER record (for Access Control accessors) or an XUSER record (for accessors referenced through the enterprise user store). Access Control first checks Seosdb for accessor information. If there is no information about an accessor in Seosdb, then Access Control requests this information from the accessor's host OS (when external user is enabled). The host OS retrieves this information from the enterprise user store. If the accessor information was not found, Seosd gives that accessor the attributes of the _undefined USER record.

The TSF maintains the following security attributes of the USER class for individual users.

User Security Attribute	Definition
APPLS	Displays the list of applications that the accessor is authorized to access.
AUDIT_MODE	Defines the activities that CA Access Control records in the audit log.
FULLNAME	Defines the full name associated with an accessor.
GAPPLS	Indicates the list of application groups that the accessor is authorized to access.
GROUPS	Displays the list of groups that the accessor or administrator belongs to. This property also contains any group authorities, such as group administration authority (GROUP-ADMIN), assigned to the administrator for each group the administrator belongs to.
LAST_ACC_TERM	Displays the terminal from which the last login was performed.
LAST_ACC_TIME	Displays the date and time of the last login.
LOGININFO	Defines the information needed to log the user in to a specific application and audit data. LOGININFO contains a separate list for each application that the user is authorized to access.
NOTIFY	Defines the accessor to be notified when a resource or accessor generates an audit event.
OBJ_TYPE	Specifies the administrator authority attributes. An administrator can have are one or more of the following authority attributes: ADMIN, AUDITOR, PWMANAGER, OPERATOR.
OWNER	Defines the user or group that owns the record.

User Security Attribute	Definition
PGMINFO	Defines the program information automatically generated by CA Access Control.
REVACL	Displays the access control lists of the accessor.

Table 9-1: Security Attributes of the USER Class

A *resource* is an entity that can be accessed by an end user accessor and protected by the access rules of the Seosdb record that corresponds to that entity. The main purpose of creating resource records in CA Access Control is to define access permissions for the resource that corresponds to the resource record. Administrators create rules on the resources that accessors are allowed to access, and these access rules are stored in Seosdb. In other words, administrators create rules if the resource protection needs to be different than the default access (this is known as a default record – the permissions which are applied to a resource if no specific record for that resource exists). Administrators are also able to create a set of rules which encompass a policy.

The main functionality of the TOE enables administrators to create rules and policies stored in Seosdb, which allow subjects to perform operations on objects within the TOE boundary. A subject is an accessor or IT entity that attempts to perform an operation on an object within or managed by the TOE. An operation is any function performed on an object within or managed by the TOE, such as creating a rule, executing a process, or creating a hash. These operations are performed on objects, which can be a record on the TOE or a resource on the OS.

Each resource belongs to an associated class, which is a collection of protected objects of the same type. Each record contains values for the properties appropriate to the record class. For example, TERMINAL is a class containing objects that are terminals (workstations) protected by CA Access Control. CA Access Control contains many predefined classes; however, only the TERMINAL, PROCESS, PROGRAM, FILE, USER, GROUP, SUDO, SURROGATE, XUSER and XGROUP classes are included in the evaluated configuration. Therefore, in the evaluated configuration, administrators are limited to write rules and policies only to resources which belong to these classes. The following table lists each class in the evaluated configuration and identifies the subject, object, and operation of a rule corresponding to the respective class.

Class	Subject	Object	Operation	Note
TERMINAL	▪ Accessor	▪ Terminal Name ▪ IP address	▪ Administrative privilege the administrator is authorized to perform ▪ Authorize log-in operation	
PROCESS	▪ Accessor	Process defined by the administrator in the rule	Ending the Process	Applies only to the accessor on the local machine
PROGRAM	▪ Seoswd (Watchdog) ▪ Seosd (Authorization Engine)	Path to the trusted program	Execution of the program, hash creation or the validation of the hash that Seoswd performs	
FILE	▪ Accessor	The file which the accessor is requesting access	Privileges assigned by administrator: e.g. Read, Write, Execute, Delete, Rename, etc.	Access can be specified regarding an individual file or to a set of similarly named files (using * or ? as wildcards). Access can also be specified for all files in a specific directory and below it. In addition, access can be specified via a specific program only.
USER	▪ Accessor	N/A	N/A	N/A
GROUP	▪ Accessor	N/A	N/A	N/A
SUDO	▪ Accessor	A command not authorized to the accessor, but is authorized to another account the accessor can use.	The accessor can issue the command with sudo.	
SURROGATE	▪ Accessor	N/A	N/A	N/A

Class	Subject	Object	Operation	Note
XUSER	▪ Accessor	N/A	N/A	CA Access Control creates the XUSER as needed
XGROUP	▪ Accessor	N/A	N/A	CA Access Control creates the XGROUP as needed.

Table 9-2: Class Mapping

When an access rule that references an enterprise user or group is created, or when an accessor logs in to the host operating system, CA Access Control creates a record in Seosdb for that accessor, if one did not exist before. These records have the class XUSER

or XGROUP (for access rules only). They hold the properties that CA Access Control requires to enforce access rules. Administrators do not need to manage the creation of records for new users because CA Access Control creates them as required.

All rules and policies created by the administrators through selang are first evaluated by Seosd to verify if the rules are valid. If they are valid, the rules and policies are then populated into Seosdb, which is the main repository of CA Access Control. In order for an end user to execute an operation on a resource, he or she must be included in a rule set by an administrator or given permission through the defaccess parameter of the rule; on the other hand, an administrator must have the appropriate administrative attribute to execute an operation on a record. The scenarios below describe an accessor's attempt to execute the operation via an end user's local terminal or an administrator's remote one.

End User: Executing a request

The end user's attempt to execute the requested action is intercepted by SEOS_syscall. If the requested action is to a FILE in Unix, SEOS_syscall first verifies its local cache to determine if the requested action can be performed. If the information is not present in the local cache, SEOS_syscall forwards the requested action to Seosd. The requested action is evaluated within Seosd to determine if the end user has previously been granted or denied access; if this information is not found in cache, the requested action is evaluated against rules that sit within Seosdb to determine if a rule associated with the record for the requested action is present within the database. Seosd fetches the relevant information from Seosdb and then makes the decision to grant or deny the end user to perform the requested action on the resource. This decision is forwarded to SEOS_syscall, which then relays the Seosd's decision to the underlying OS. If the requested action was granted by Seosd, the underlying OS will continue with the requested action. On the contrary, if the requested action was denied the underlying OS will return an error without performing the requested action. If there is no resource record in the seosdb, the end user is granted or denied access as determined by the default permissions of the underlying OS.

Note: *Review* [Figure 1-1: TOE Boundary] *to follow along with end user: Executing a Request.*

Administrator Authorization to the TOE:

The administrator executes Selang and establishes a session with Agent over port 5249 using the TLS v1.0 protocol using the hosts command. Agent on the target machine collects the administrator's username and the terminal name from the connected socket and passes them to the seosd for validation. The seosd validates that the terminal IP address/terminal name is in the TERMINAL class and ensures the appropriate administrator attribute(s) are present in Seodb. Seosd then checks if the user has the ADMIN attribute and if the user has 'WRITE' access in this TERMINAL record against the Seosdb. Seosd uses this information to make the decision to allow or deny access to the TOE. The decision to grant or deny access is sent back to Selang. Once an administrator has been authorized access to the TOE, he or she is allowed to create rules or manage the TOE to the level of the administrator attributes assigned to them.

Note: *Review* [Figure 1-1: TOE Boundary] *to follow along with Administrator authorization.*

Once the administrator is given the authority to manage the TOE over Selang, the TOE will only check that authority and not the information provided to establish a session. The process of creating rules is detailed below.

Creating a rule:

The agent gets the rule from selang and verifies the rule's author has sufficient permission to create the rule. Seosd determines if a similar rule was previously created and if it is valid based on syntax and object existence. To ensure the rule can be created, Seosd queries Seosdb for the administrator attributes required for the objects named in that rule. If the administrator meets the specified attributes and if the rule is valid, the rule can then be created. Once the rule is written and accepted, it is stored in Seosdb for future verification. Seosd then writes audit information on all administrator and end user actions in the Seos.audit file.

Note: *An audit record is not automatically generated, it depends on the audit policy of the defined resource in the database.*

In order to protect end user access to the system account, the TOE is capable of supplementing the native passwd command with the sepass utility. This allows the password policies defined by the TOE to be applied to the end user accounts on the host endpoints. When an accessor attempts to change their password, sepass intercepts the request and determines if the chosen password meets the password policy defined by the TOE (in a record of the SEOS class). If it does not, they will be prompted to enter a new password. If it does, the password is forwarded to passwd and the password change is made. Because the TOE is not responsible for the password checking once it has been set in the OS, any authentication failure handling will be the responsibility of the host OS.

9.2.5 Access Control for End Users

Every access attempt on the machine CA Access Control protects is performed by an end user. CA Access Control uses accessor information from Seosdb and from the enterprise user stores in the host operating system to make decisions on the end users access. If it cannot find a record for an accessor defined in Seosdb, it looks for, and uses the information from, the users and the group memberships defined in the enterprise stores. Seosdb stores accessor information in either a USER record or an XUSER record. If it has no information about a user (or the enterprise user store ability is disabled) it gives that user the attributes of the _undefined USER record.

The TOE can enforce password composition rules against passwords used for the accessor's OS login. If an accessor runs the sepass utility (either directly or by an administrator-defined link between passwd and sepass), it will receive the accessor's desired password and compare it against the defined password policy and determine if the new password is allowed. If it is, the validated password is changed. Otherwise, the accessor will be notified that the password they have chosen is not allowed.

After the OS login process passes the authentication stage for the end user, CA Access Control intercepts the process and performs the following steps until it finds a record associated with the accessor:

1. CA Access Control searches the Seosdb for a record of an accessor with the same username as the end user which authenticated to the OS.
2. CA Access Control uses the operating system to search the enterprise user stores for an accessor of that name.

Note: If no user can be found for any reason, the default username for an accessor is "_undefined."

The main purpose of CA Access Control is to assign and enforce access authorities, also known as access rights, to accessors which log in to the local OS.

An access authority is a permission owned by an accessor to perform a specified access on a resource.

An access authority always has the following components:

- The resource that the access applies to
- Whether or not the accessor has access to that resource
- The accessor, which is either a user or a group

An accessor has the authority to access a resource in a certain way because one or more of the following are true:

- The accessor is the owner, as granted by the resource ACL.

- The accessor has the access authority, as granted by the inclusion of their username in the resource ACL.
- The accessor is a member of a group that has access authority.
- The accessor is running a program that has the access authority. For example, the accessor has the authority to run a program in the PROGRAMS class. The program has a separate PACL that defines privileges based on either the program itself, a group it belongs to, or a wildcard.
- The “defaccess” parameter of the resource allows default access to it.

9.2.5.1 Access Control Lists

The access authorities to a resource are specified in an access control list (ACL). Every resource record has at least two access control lists: ACL, which specifies the accessors that are granted access to the resource, together with the type of access they are granted.

Each resource record also has a negative access control list (NACL), which specifies the accessors that are denied authorization to the resource, together with the type of access that they are denied. For the TERMINAL class, the access authority can also depend on the circumstances around the access, such as whether the accessor is logged in locally or not.

Conditional Access Control Lists (CACLs) provide an extension to ACLs. When an accessor attempts to access a resource, if the resource's ACL and NACL do not define an access authority for the accessor, CA Access Control examines the conditional access control lists.

The conditional access control lists specify access to resource where the access is by a particular method, for example by using a specified program. Program Access Lists (PACLs) are a type of Conditional Access Control List. In the evaluated configuration, the only type of CACL that will be evaluated is the PACL.

The order in which it runs through these checks is important. For each resource, CA Access Control checks its record for access control rights in the following order by default:

1. The resource's ownership (owners are allowed access)
2. The resource's NACL
3. The resource's ACL
4. The resource's PACL
5. The resource's defaccess field

One access control list can contain more than one entry that affects an accessor. For example, it can contain an entry that mentions an accessor explicitly, and also entries for each of the groups to which the accessor belongs. CA Access Control checks all the possible entries at each level before it goes to the next level.

The record for a resource can include a default access field, defaccess. The value of the defaccess field specifies the access authority that is allowed to accessors who are not covered by any of the resource access control lists.

9.2.6 Classes

The classes included in the evaluated configuration include: PROGRAM, PROCESS, TERMINAL, FILE, USER, GROUP, SURROGATE, XGROUP and XUSER.

The information about CLASS status (that is, whether the class was active or inactive) is held in Seosdb. Every attempt to access a resource is intercepted by CA Access Control, which checks the status in Seosdb. If the class is inactive, access is allowed without further checking for authorization. CA Access Control issues an instantiation of active classes when Seosd starts up and when an accessor changes the CLASS activity status.

9.2.7 Access Control for Administrators

Administrators can create new accessor records, delete and modify accessor records, modify all or part of Seosdb, shut down AC, and assign administrative attributes to other administrators.

To use selang commands that change records in the Seosdb, an administrator must have sufficient authority. For most commands, one of the following conditions must be met:

- The administrator is the owner of the resource.
- The administrator has an administrator attribute (ADMIN, AUDITOR, etc.).
- The resource record is within the scope of a group in which the administrator has the GROUP-ADMIN attribute.
- The administrator has CREATE or MODIFY access authority in the ACL of the record in the ADMIN class.
- An administrator must be a member of the CA Access Control Administrators group in the security files of the local host.

Only users with the ADMIN attribute are able to assign and revoke the administrative attributes to administrators. Immediately upon the revocation of an administrative attribute, the administrator which was assigned that attribute loses the administrative operations allowed from assignment of that attribute. When a user with administrative attribute(s) has all administrative attribute(s) revoked, CA Access Control prevents them from managing the TOE through Selang. The last administrator with the ADMIN attribute cannot revoke the ADMIN attribute from himself.

9.2.7.1 Global Authorization Attributes

Global authorization attributes are set in the administrator's record. Each global authorization attribute permits the administrator to perform certain types of functions. This section describes the functions and the limits of each global authorization attribute.

The following table lists each administrative attribute in the evaluated configuration.

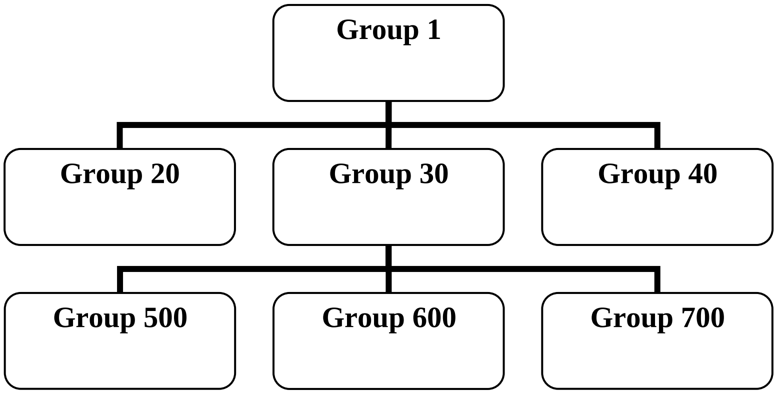
Attribute	Privileges/Operations	Restrictions
ADMIN	Read/modify record properties	The last admin with the ADMIN attribute cannot be deleted.
	Create new records	The ADMIN attribute cannot be removed from the last admin with the ADMIN attribute.
	Delete records	Unable to update the audit mode without the AUDITOR attribute.
	Can set root to be a non-ADMIN administrator	Unable to delete root.
	Set up Policy Model	None
	Set up password policies	Administrators with the AUDITOR or OPERATOR attribute have read-only access to the password policies.
AUDITOR	List information in Seosdb	Unable to perform any other operations than those listed.
	Modify/set the audit mode for existing records	
OPERATOR	READ access to all files	Unable to perform any other operations than those listed.
	List information in Seosdb	
PWMANAGER	Change passwords of other users	Unable to change the number of grace logins.
		Unable to change the password interval of another accessor.

		Unable to change general password rules.

Table 9-3: Global Authorization Attributes for Administrators

9.2.7.2 Group Authorization Attributes

The concept of subordinate and superior groups, also known as parentage, is important when discussing group administration privileges. One group can be the parent-superior-of one or more groups. A child or subordinate group can have only one parent. Assigning a parent to a group is optional. Consider the following diagram:



Group 1 is the parent of the three Groups 20, 30, and 40. Group 30 is also the parent of three groups-500, 600, and 700. Group 600 has only one parent-Group 30. Group 1 has no parent.

All records, including resource records and accessor records alike, have owners. Owning a record means having authorization to view, edit, and remove it, as described in Ownership in this section.

A group can own its own records. However, within a group that owns records, only certain privileged administrators can manage the records. These special administrators have a group authorization attribute set in their own administrator records. The group authorization attributes are the following:

- GROUP ADMIN
- GROUP AUDITOR
- GROUP OPERATOR
- GROUP PWMANAGER

The join command-which only a properly authorized administrator can issue-sets these attributes. The join command serves the purpose of both putting an administrator into a group, and specifying the administrator's group authorization attribute (if any).

The privileged members of the group may or may not be authorized to manage the administrator records that define the members of the group, depending on who owns those records.

The following table lists the administrator group attributes in the evaluated configuration.

Group Attribute	Privileges/Operations	Restrictions
GROUP-ADMIN	Read/modify record properties	Unable to make resources inaccessible to themselves
	Create new records	Unable to assign a security level that is higher than their own security level.
	Delete records	Unable to assign a security category or security label that they do not have.
	Connect users to a group or separate users from a group	Unable to delete the root administrator from Seosdb.
		Unable to delete the only administrator with the ADMIN attribute in Seosdb.
		Unable to remove the ADMIN attribute from the record of the last ADMIN administrator in Seosdb.
		Unable to set the ADMIN, AUDITOR, OPERATOR, PWMANAGER, and SERVER authorization attributes for any administrator.
		Those without the AUDITOR attribute cannot update the audit mode.
GROUP-AUDITOR	List the properties of any record within the group scope	Unable to perform any other operations than those listed.
	Set the audit mode for any record within the group scope	

GROUP-OPERATOR	List the properties of any record within the group scope	Unable to perform any other operations than those listed.
GROUP-PWMANAGER	Change the password of any user within the group scope	Unable to perform any other operations than those listed.

Table 9-4: Group Authorization Attributes for Administrators

Ownership

Every record in Seosdb, including both accessor records and resource records, has an owner. When a record is added to Seosdb, its owner can be explicitly assigned either by using the owner parameter or by allowing CA Access Control to assign the administrator who defines the record as the owner of the record. The lang.ini file can also be edited to change this assignment. Note that the accessor assigned to own a record does not need to be an administrator or have any special privileges in order to own a record.

When a group owns a record, only accessors who have joined the group with the GROUP ADMIN property can use the privileges of ownership. If an accessor that owns a record is removed from Seosdb, the records no longer have an owner.

Accessors who own records have the following access authority for the records they own:

- Read – Shows the properties of the record.
- Modify – Changes the properties of the record.
- Delete – Removes the record from Seosdb.
- Connect – Joins a user to a group or separates a user from a group.

If it is not desirable for an accessor to have ownership authority over a particular record, the owner nobody can be assigned to the record.

- The limits of the ownership privileges are as follows:
- The owner of the last ADMIN user in Seosdb cannot delete that administrator record.
- Owners who do not have the AUDITOR attribute cannot update the audit mode. Only an owner with the AUDITOR attribute can update the audit mode.
- The owner of root cannot delete root from Seosdb.
- Owners cannot set the global authorization attributes-ADMIN, AUDITOR, OPERATOR, and PWMANAGER-for the administrators they own.
- Owners cannot make resources inaccessible to themselves, so:

- Owners cannot assign a security level that is higher than their own security level.
- Owners cannot assign a security category or security label that they do not have.

Owners can change password policy for a user they own or for a profile group they own to influence password policy for all users which have a certain profile.

9.2.8 Security Audit

CA Access Control generates secure and reliable audit logs which associate usernames to all resource actions. The TOE maintains the username that an accessor originally authenticated as, so even if the individual using the TOE uses the su command or attempts to change their username in some other manner, the TOE will record these actions as their originally authenticated username.

The audit records are stored in an audit log called seos.audit. The location for the audit log is specified in the *seos.ini* file. This file also stores the configuration for audit log backup. By default, only a single backup log is created (seos.audit.bak), but seos.ini can be configured to establish a configuration where audit logs are rolled over indefinitely based on a certain time period or log file size.

By default, Seosd creates the audit logs with root ownership, since the Seosd program is executed with root privileges. For the same reason, the audit logs are created with read/write permissions granted only to root. CA Access Control auditing daemons and logs are self-protected and cannot be shutdown or modified. Their functioning is monitored by the watchdog which ensures they remain running in an unaltered state.

The seaudit utility can be used to list recorded events in the audit log, and sort events by time restrictions or event type/Class (column 4 of the audit log). In order to execute the seaudit utility, an administrator must log in to the local machine as an end user and have access rights as defined by the TOE. An administrator with the AUDITOR or GROUP-AUDITOR attributes will be able to change the audit property of a resource in the database. When displaying audit records that include passwords, seaudit protects password identity by substituting a series of asterisks (***) in place of the password text.

In addition to an administrator's ability to change the audit property of a resource, CA Access Control has the capability to additionally filter, or block, audit records from being written by Seosd into the seos.audit file. Through the audit.cfg file, a filter can be supplied that Seosd reads during startup, defining audit records that should not be generated. This filter helps to limit the size of the seos.audit file by keeping only the records needed. Filtering rules can be set for class name, object name, user name, group name, program name, access rights, and authorization result. Audit filter rules are written in the *ACInstallDir*/etc/audit.cfg file.

Administrators with the AUDITOR or GROUP-AUDITOR attributes are permitted to perform auditing tasks such as changing the auditing attribute that is assigned to accessors and resources. CA Access Control includes two entries in the seos.ini file that specify which group ownership is assigned to the log files.

- One entry is for the audit log. The log routing daemons consult the same token to see who should have access rights to the audit logs that the daemons produce and collect. Note that the audit logs are subject to access control like any other files, and CA Access Control rules can keep accessors from accessing them.
- The other entry is for the error log, and it is used in the same way to specify group ownership for that file.

9.2.8.1 Audit Records

Each record that seaudit displays contains data arranged in columns. The data in the first three columns has the same meaning for all types of records. The remaining data displayed depends on the type of record.

The following table describes the format of the output for the most common types of records, by column:

Column	Contents	Description
1	Date	The date the access or attempted access occurred.
2	Time	The time the access or attempted access occurred.
3	Return code	The CA Access Control return code that indicates what happened. Valid values are: • D - CA Access Control denied access to a resource or did not permit an update to Seosdb because the accessor did not have sufficient authorization. • F - An attempt to update Seosdb failed. • M - CA Access Control was started or shut down. • O - An accessor or administrator logged out. • P - CA Access Control permitted access to a resource or permitted a login. • S - Seosdb was successfully updated. • U - A trusted PROGRAM or SECFILE was changed, so it is now untrusted. • W - An accessor's authority was insufficient to access the specified resource; however, CA Access Control allowed the access because warning mode is set in the resource or the class.

Column	Contents	Description
4	Event type/ Class	The type of event being audited or the class on which the action was performed.
5	Accessor/ Class	If the previous column contains a class name, this column contains the username of the accessor who executed the command. If the previous column contains UPDATE, this column contains the class in which the action was performed. Otherwise, this column contains the name of the accessor who executed the command or any other relevant information about the class.
6	Access type/ Accessor	If the previous column contains the accessor username, this column contains the access type, if relevant. If the previous column contains the class name, this column contains the username of the accessor who executed the command. Otherwise, this column contains the access type, if relevant, or any other relevant information according to the class.
7	Stage code	A number (up to three digits) that indicates at which stage CA Access Control decided what action to take and why.
8	Audit record code	A number that represents the reason that CA Access Control wrote an audit record.
9	Resource	This column contains the name of the resource being accessed or updated.
10	Terminal/ Program	If column four contains UPDATE, this column contains the name of the terminal from which the update was made. Otherwise, this column contains the name of the program that accessed the resource.
11	Command	If column four contains UPDATE, this column contains a complete copy of the command entered by the accessor. If the command is a password update, the password itself is replaced by a series of asterisks. If column four does not contain UPDATE and an action is being performed on the CLASS object via a remote terminal, then this column displays the IP address of remote terminal.

Table 9-5: Common Audit Records

9.2.8.2 Audit Events and Properties

CA Access Control keeps audit records for events of access denial and access grants according to the audit rules defined in Seosdb. The decision whether to log a certain event is based on the following rules:

Value of AUDIT	What is Logged	Applicable Objects
FAIL	Access failures	Accessors and resources
SUCCESS	Access successes	Accessors and resources
LOGINFAIL	Login failures	Accessors
LOGINSUCCESS	Login successes	Accessors
ALL	Equivalent to FAIL, SUCCESS, LOGINFAIL and LOGINSUCCESS	Accessors and resources
TRACE	Equivalent to ALL plus all system events	Accessors
NONE	No logging	Accessors and resources

Table 9-6: Audit Properties

The records in the audit log accumulate according to these audit rules. The decision whether to log an event is based on the following:

- If the resource or accessor has AUDIT(ALL), all login events for the accessor and all events concerning resources protected by CA Access Control are logged, regardless of whether access failed or succeeded.
- If access to a resource protected by CA Access Control is successful and the accessor or resource has AUDIT(SUCCESS), the event is logged.
- If access to a resource protected by CA Access Control fails and the accessor or resource has AUDIT(FAIL), the event is logged.
- If access to a resource protected by CA Access Control has AUDIT(NONE), neither success nor failure events will be logged.

These settings are used to define the selected set of events to be audited by the TOE.

Whenever an accessor or resource record is created or updated in Seosdb, the AUDIT property can be specified. To define which access events CA Access Control should log, the value of the AUDIT property of the resource or accessor is changed. Additionally, the method to specify that CA Access Control should log every trace event to the audit log can be used.

When reviewing the audit log, an administrator locally accesses an endpoint that is managed by Access Control as a user with an account that is privileged by the TOE to access seaudit. This could be the root account, a separate privileged account that was used to install the Access Control endpoint, or a separate auditor account that was created just for that purpose. If an account other than the root account is used, OS protection and Access Control rules will need to be updated in order to delegate this privilege. Access

Control has the ability to route audit log files to a central repository using the selogrd utility, but this is beyond the scope of the evaluation.

Seaudit allows an administrator to filter and search information by various criteria, such as filtering for all audit data for a specific time period or searching for failed actions performed by a specific user.

The audit log can be configured to have one or indefinite backups. If it's set to have one backup, the seos.audit file is backed up as seos.audit.bak, which is then overwritten when the next seos.audit file needs to be backed up. If indefinite backup is desired, a new audit file with a datestamp in the filename will be added on the desired interval, one of daily, weekly, or monthly. If the audit log fills before the interval is over, a new log is started early. The audit log is filled when the size threshold has been exceeded. The default size is 1024 KB and can be modified by using a parameter in the invocation of seaudit. Since audit data is written as unformatted text, disk space will not be an issue on modern systems; however, since there is no notification provided when an audit threshold is close to or over the limit, administrators are instructed to periodically ensure that space is available on the endpoint.

Seos.audit also records an entry when a component of the TOE goes down or up so that attempts to disable its operation can be identified.

9.2.9 Security Management

Through the use of selang, CA Access Control allows administrators to manage accessors and resources on the host machine. Selang is the CA Access Control command language. The selang command language is the command definition language that lets administrators make rules in Seosdb. Administrators can create, modify and delete resource records and accessor records, including assigning administrative attributes to accessor records. See sections on Access Control for End Users and Access Control for Administrators for more information on security management.

For the evaluated configuration, the following minimum password policy is defined and should be implemented in the TOE during operation:

- At least 8 characters long
- At least one lowercase letter, one uppercase letter, and one number
- No more than two repeating characters (i.e. 1Aaaaaaa would not be allowed but 1Aaabbcc would be)

In order to manage the TOE remotely, multiple options are made available by administrators. By using the “hosts” command within selang and specifying one or more target machines, an administrator can issue commands via selang to a single remote terminal or to multiple terminals simultaneously (if more than one host is specified).

In addition to the hosts command within selang, the TOE allows administrators to propagate management functions across a number of terminals in an enterprise by using the Policy Model service.

The Policy Model service involves setting up a database called a Policy Model Database (PMDB). This database stores users, groups, rules, resources, and other elements that comprise a seosdb. A number of endpoints can subscribe to the PMDB, and when any changes are made to the PMDB, the change is propagated to the seosdb of each subscriber. In this way, multiple endpoints can be configured by issuing a single set of commands. One computer can manage multiple PMDBs, each with a differing set of subscribers, so that multiple types of policies can be centrally managed.

In order to manage a Policy Model, the administrator must enter the pmd environment in selang with the command “env pmd”. The selang command “createpmd” allows an administrator to designate a host database as a PMDB and set up the administrator access and management privileges for it. The “subs” command allows the administrator to set up the list of subscribers to the PMDB.

When the PMDB is updated, the sepmdd daemon pushes any updates made to all subscribers. Subscribers do not need to poll the PMDB continuously because the updates are only a push operation.

A subscriber database can be assured that the PMDB to which it’s subscribed is a valid database because of the synchronization of UIDs which is performed when the subscription is first made. If an attacker wished to push insecure policies through an imposter PMDB, they would require a comprehensive list of UIDs managed by the original PMDB, which would require root access to the PMDB’s host OS.

9.2.10 Encrypted Communications

Communications between Agent and the selang shell use AES with HMAC-SHA-256 with key sizes of 128 bits.

The remote selang client connects to the TOE using TLS v1.0 with CipherSuite TLS_RSA_WITH_AES_128_CBC_SHA, as defined by RFC 3268. The same encryption is used when a seagent connects to a subscribed PMDB via sepmdd.

TLS uses AES and RSA. AES (Advanced Encryption Standard) is an encryption algorithm approved by FIPS 140 and defined in FIPS 197. AES is a secret-key cipher that enciphers data in blocks of 128 bits and whose key size may be 128, 192, or 256 bits. In the evaluated configuration, AES is used by Access Control with 128-bit HMAC keys for symmetric cipher. RSA is a standard asymmetric-key encryption algorithm used with SHA-256 for TLS connections and is used by Access Control for key generation. TLS is used to protect the disclosure and modification of information between Seagent and the selang shell on the remote client. This secure connection is used to authorize administrators’ access to the TOE and for the management of the TOE, and is initiated by the administrator when the host command is used. The secure connection is established in the same way by sepmdd when it propagates PMDB changes to subscriber databases.

This TLS connection is performed between PMDB Endpoint’s Seagent and the subscriber’s Seagent.

In addition, SHA-1 is used for cryptographic hash functions and is used in Access Control with trusted programs.

9.3 TOE Summary Specification Rationale

This section identifies the security functions provided by the TOE mapped to the security functional requirement components contained in this ST. This mapping is provided in the following table.

Security Function	Security Functional Components
User Data Protection	FDP_ACC.1 Subset access control
	FDP_ACF.1(1) Security attribute based access control
	FDP_ACF.1(2) Security attribute based access control
Identification and Authentication	FIA_ATD.1 User attribute definition
	FIA_SOS.1 Verification of Secrets
	FIA_UID.2 User Identification before Any Action
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 Protected audit trail storage
Security Management	FMT_MOF.1 Management of security functions behavior
	FMT_MSA.1 Management of security attributes
	FMT_MSA.3 Static attribute initialization
	FMT_SMR.2 Restrictions on Security Roles
	FMT_SMF.1 Specification of management functions
	FMT_MTD.1 Management of TSF data
	FMT_REV.1 Revocation
Cryptographic Support	FCS_CKM.1 Cryptographic key generation

Security Function	Security Functional Components
	FCS_CKM.4 Cryptographic key destruction
	FCS_COP.1 Cryptographic operation
Protection of the TSF	FPT_FLS.1 Failure with preservation of secure state
Resource Utilization	FRU_FLT.1 Degraded fault tolerance
TOE Access	FTA_TSE.1 TOE session establishment
Trusted Path/Channels	FTP_TRP.1 Trusted path

Table 9-7: Security Functional Components

9.3.1 User Data Protection

The User Data Protection function of the TOE enforces the FDP_ACC.1, FDP_ACF.1 (1), and FDP_ACF.1 (2) requirements.

When an accessor attempts to access an object protected by the TOE, CA Access Control's Seosd uses the records in Seosdb to determine if the accessor should be granted or denied access. CA Access Control's rules and policies are enforced by the TSF on accessors who attempt to access objects. The TSF enforces the rules and policies to objects based on username, group name, access authorities (administrative attributes for administrators); access control lists (ACLs) and class. If an accessor supplies the correct user/group name, has the approved access authority, and/or is included in an access control list or class, he is able to access the object.

Components of the TOE interact to enforce access control. The Seagent interprets an administrator's request to access a record. An administrator must have WRITE access from the approved terminal, approval of terminal name or IP address in the TERMINAL class, and an administrator attribute must be included in the administrator's respective record in Seosdb and a rule must be written depicting his access to the TOE in order to manage the TOE through selang. For end users, SEOS_syscall intercepts the request to access a file, program, or service. It then checks with Seosd to determine if the Seosdb has a resource record for the file, program, or service being accessed. If a resource record is not found, then it is considered unprotected and the end user is granted or denied access as determined by the default permissions of the underlying OS. If a resource record is found, the requested file, program, or service is protected and Seosd checks for the end user's information in Seosd. Once the end user's information is found and verified, the end user is then checked against the rules and/or policy of the resource record determining if access is allowed to the resource.

9.3.2 Identification and Authentication

The identification and authentication function of the TOE enforces FIA_ATD.1, FIA_UID.2, and FIA_SOS.1.

The client OS provides user identification, authentication and authorization through the use of user records and passwords for Administrators and accessors. End users have to identify and authenticate to the host operating system before being allowed access to TOE resources. The TOE is able to provide password composition checking to user password changes to the host operating system, ensuring passwords of sufficient strength to meet an organization's security requirements. However, once passwords are set in the OS, the OS retains the responsibility for handling authentication failure.

Administrators have to identify and authenticate themselves to the Operational Environment before being able to remotely manage the TOE through selang. The TOE is able to maintain the originally authenticated identity of an end user so that access control rules are still enforced on them even if they perform a superuser operation (sudo). This is accomplished by using seosd to cache the accessor's UID, which is the identifier that is checked against all access requests, ensuring that all TSF-mediated actions require identification.

Authentication failures can be handled by the OS at the administrator's discretion; the evaluated configuration of the TOE provides no enforcement of authentication, only authorization based on already-authenticated sessions.

During authorization, CA Access Control's Seosd communicates with Seosdb to determine that the UID and IP address (for administrators) are retrieved from an accessor who is requesting access to the TOE or its resources. When an accessor attempts to access an object, Seosd uses the accessor's information along with the rules and policies stored in Seosdb to determine how to authorize the accessor.

The rules and policies defined in Seosdb specify the objects that the accessor is allowed to access.

9.3.3 Security Audit

The security audit function of the TOE enforces the FAU_GEN.1, FAU_GEN.2, FAU_SAR.1, FAU_SAR.2, FAU_SAR.3, FAU_SEL.1 and FAU_STG.1 requirements. FAU_GEN.1 requires a reliable time-stamp, which is provided by FPT_STM_EXT.1.1 (provided by the Operational environment).

By default, the authorization daemon Seosd creates the audit logs with root ownership, since the Seosd program is executed by the root user. For the same reason, the audit logs are created with read/write permissions granted only to root. In the evaluated configuration, the seos.audit file stores the record of the startup and shutdown of the TOE's audit functions. The audit report includes access failures for accessors and resources, access successes for accessors and resources, login failures for accessors and administrators, login successes for accessors and administrators, and all system events on the TOE that go to the trace files.

The minimum contents of each entry in the audit report include the following: Date and time of the event, type of event, subject identity, the outcome (success or failure) of the event, return code, accessor/class, stage code, audit record code, resource, terminal/program and command (see Table 9-4 Common Audit Records for more information).

The TOE shall protect the stored audit records from unauthorized deletion, and shall be able to prevent unauthorized modifications to the audit records in the audit trail. The audit log is compressed and internally protected by Access Control. However, the user can read the audit log on the local machine as long as the appropriate access rights are provided. It is important to note the users only have READ access to the audit log.

The TOE relies on the underlying operating system to provide accurate time stamps to be used for audit records.

9.3.4 Security Management

The security management function of the TOE enforces the FMT_MOF.1, FMT_MSA.1, FMT_MSA.3, FMT_SMR.2, FMT_REV.1, FMT_SMF.1 and FMT_MTD.1 requirements.

The TOE provides management capabilities through selang, the command line interface that is used remotely by administrators. The TSF shall provide the ability to manage its security functions including the management of accessor accounts and accessor access rights, TOE resources and security information recorded in the audit logs.

The TSF shall maintain the roles end user and administrator for Access Control. An end user is a user that accesses the machine which the TOE protects. An administrator is a user that accesses the TOE via Selang. To be an administrator, a user must have one of the administrative attributes to use Selang. The TOE shall provide the ability to set attributes to provide security relevant authority to an administrator (e.g. give a user the AUDITOR attribute). The TSF shall allow the Administrator to specify alternative initial values to override the default values set by the TOE when an object or information is created (e.g. how long an administrator can be idle before their session is closed).

The TSF shall restrict the ability to modify the auditable events to those administrators with the AUDITOR or GROUP-AUDITOR attribute. The TSF shall enforce the Access Control rules and policies to restrict the ability to modify or delete security attributes of the TERMINAL, PROCESS, PROGRAM, FILE, USER, GROUP, XUSER, and XGROUP classes to those administrators with ADMIN, GROUP-ADMIN, AUDITOR, GROUP-AUDITOR, PWMANAGER, GROUP-PWMANANGER, OPERATOR, and/or GROUP-OPERATOR attributes.

By using the Policy Model service, the administrator is performing management functions of the TOE simultaneously against multiple endpoints.

9.3.5 Cryptographic Support

The Cryptographic Support function of the TOE enforces the FCS_CKM.1, FCS_CKM.4 and FCS_COP.1 requirements.

The TOE uses TLS v1.0 encryption for secured communication between the administrators on the client OS and Access Control's Seagent through port 5249 as well as communications between seagent and a remote sepmdd instance. TLS uses RSA with 1024-bit keys for key generation and AES with 128-bit keys for symmetric key usage (encryption and decryption). The same encryption schemes are used when data in a PMDB is propagated to subscriber databases. The TSF shall destroy cryptographic keys in accordance with the overwrite method.

CA Access Control will be installed in 'Fips-only' mode. In FIPS-only mode CA Access Control uses the ETPKI 3.2.1 encryption library. On UNIX systems it uses the OS encryption library for password encryption ("crypt" method).

A trusted path is established from the client OS to the TOE via selang over TLS to ensure that all traffic communications to and from the TOE are protected from unauthorized disclosure.

CA Access Control uses FIPS 140-2 certified encryption toolkits to implement cryptographic operations. A FIPS 140-2 compliant mode of operation is provided and can be implemented during installation.

9.3.6 Protection of the TSF

The Protection of the TSF function of the TOE enforces the FPT_FLS.1, FPT_ITT.1, and FPT_STM_EXT.1 requirements.

The TOE maintains and controls individual sessions for administrators and end users. The TSF, when invoked by the underlying host OS, ensures that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed. The TSF maintains a security domain for its own execution that protects it from interference and tampering by un-trusted subjects initiating actions through its own TSFI. It also protects TSF data that's transferred between distributed components of the TOE. The TSF shall preserve a secure state when the Seoswd, Seosd, or Seagent daemon goes down. This is done in the following manner:

- Seoswd monitors seosd and restarts seosd in the event of failure or termination.
- Seosd monitors seagent and restarts seagent in the event of failure or termination.
- Seagent monitors seoswd and restarts seoswd in the event of failure or termination.

The TOE relies on the host operating system to provide reliable timestamps for audit records.

9.3.7 Resource Utilization

The Resource Utilization function of the TOE enforces the FRU_FLT.1 requirement.

The TSF shall ensure the operation of access control to resources when the Seoswd, Seosd, or Seagent daemon goes down. This is accomplished by the mutual monitoring defined in the previous section. When one of the daemons goes down, the daemon responsible for its operation restarts it. This ensures that there is no discontinuity of resource protection.

9.3.8 TOE Access

The TOE access function of the TOE enforces the FTA_TSE.1 requirement.

The TSF shall deny access to the TOE when administrators attempt to remotely login to manage the TOE through selang and the following applies: the IP address or terminal name is not approved by the TOE, the administrator does not have WRITE access from the terminal, and the username/password given by the administrator cannot be verified by the TOE if they are supplied. Once the administrator has access to the TOE, the ADMIN, AUDITOR, PWMANAGER, OPERATOR, GROUP-ADMIN, GROUP-AUDITOR, GROUP-PWMANAGER, and GROUP-OPERATOR attributes determine the scope of the management functions available to them.

9.3.9 Trusted Path/Channels

The Trusted Path function of the TOE enforces the FPT_ITT.1 and FTP_TRP.1 requirements.

The TSF shall provide a path for communication between itself and remote administrators 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.   The TSF shall allow initial communication to the trusted path by remote administrators, and it shall require the use of the trusted path for administrator authorization and all other TSF mediated actions by the administrator. The TOE also uses the trusted path for inter-TSF communications, specifically the propagation of configuration information to subscriber databases in the Policy Model implementation.

The TOE uses TLS v1.0 encryption for the trusted path secured communication between the administrators on the client OS and Access Control’s Seagent through port 5249.

10 Rationale

10.1 Security Objective Rationale

The following table provides a mapping with rationale to identify the security objectives that address the stated assumptions and threats.

Assumption	Objective	Rationale
A. ADMIN One or more authorized administrators assigned to install, configure, and manage the TOE.	OE. ADMIN One or more authorized administrators will be assigned to install, configure and manage the TOE.	OE. ADMIN maps to A. ADMIN in order to ensure that authorized administrators install, manage and operate the TOE in a manner that maintains its security objectives.
A. PATCHES System Administrators exercise due diligence to update the TOE with the latest patches and patch the Operational Environment(e.g., OS and database) to ensure all known system vulnerabilities are not exploited.	OE. ADMIN One or more authorized administrators will be assigned to install, configure and manage the TOE.	OE. ADMIN maps to A. PATCHES in order to ensure that the authorized administrators properly patch the TOE and the Operational environment in a manner that maintains its security objectives.
A.NOEVIL Administrators of the TOE are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided by the organization's guidance documentation.	OE.NOEVIL Administrators of the TOE are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided by the organization's guidance documentation.	OE.NOEVIL maps to A.NOEVIL in order to ensure that there are no careless, willfully negligent, or hostile administrators of the TOE.
A.LOCATE The TOE will be located within controlled access facilities that will prevent unauthorized physical access.	OE.LOCATE The TOE will be located within controlled access facilities that will prevent unauthorized physical access.	OE.LOCATE maps to A.LOCATE in order to ensure the physical security in which the TOE operates.

Table 10-1: Assumption to Objective Mapping

Component	Documents(s)	Rationale
ACCESS Unauthorized users could gain local or remote access to protected objects that they are not authorized to access.	O.ACCESS The TOE will provide measures to authorize users to access specified TOE resources once the user has been authenticated. User authorization is based on access rights configured by the authorized users of the TOE.	O.ACCESS (FDP_ACC.1, FDP_ACF.1(1), FDP_ACF.1(2), FTA_TSE.1) addresses T.ACCESS by providing the authorized users with the capability to specify access restrictions on the protected TOE resources to authenticated users.
	O.SELF_PROTECTION The TOE will preserve a secure state and ensure access control to resources when a component of the TOE fails.	O.SELF_PROTECTION (FPT_FLS.1, FRU_FLT.1) addresses T.ACCESS by ensuring connectivity of failed components are reinitialized prior to resources being accessed.
	OE.ROBUST_ACCESS The operational environment will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate. The operational environment will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate.	OE.ROBUST_ACCESS (FIA_UAU_EXT.2(1), FIA_UAU_EXT.2(2), FIA_UID_EXT.2(1), and FIA_UID_EXT.2 (2)) addresses T.ACCESS by controlling the logical access to the Operational environment and its resources.
	OE.AUTH The Operational Environment will provide measures to uniquely identify all users and will authenticate the claimed identity prior to granting a user access to the resources protected by the TOE.	OE.AUTH (FIA_ATD.1, FIA_UAU_EXT.2(1), FIA_UAU_EXT.2(2), FIA_UID_EXT.2(1), FIA_UID_EXT.2(2)) helps to mitigate T.ACCESS by providing measures to uniquely identify and authenticate users through the OS authentication.
ACCESS Unauthorized users could gain local or remote access to protected objects	O.ACCESS The TOE will provide measures to authorize users to access specified TOE resources once the user has	O.ACCESS (FDP_ACC.1, FDP_ACF.1(1), FDP_ACF.1(2),

Component	Documents(s)	Rationale
that they are not authorized to access.	been authenticated. User authorization is based on access rights configured by the authorized users of the TOE.	FTA_TSE.1) addresses T.ACCESS by providing the authorized users with the capability to specify access restrictions on the protected TOE resources to authenticated users.
	O.SELF_PROTECTION The TOE will preserve a secure state and ensure access control to resources when a component of the TOE fails.	O.SELF_PROTECTION (FPT_FLS.1, FRU_FLT.1) addresses T.ACCESS by ensuring connectivity of failed components are reinitialized prior to resources being accessed.
	OE.ROBUST_ACCESS The operational environment will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate. The operational environment will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate.	OE.ROBUST_ACCESS (FIA_UAU_EXT.2(1), FIA_UAU_EXT.2(2), FIA_UID_EXT.2(1), and FIA_UID_EXT.2 (2)) addresses T.ACCESS by controlling the logical access to the Operational environment and its resources.
	OE.AUTH The Operational Environment will provide measures to uniquely identify all users and will authenticate the claimed identity prior to granting a user access to the resources protected by the TOE.	OE.AUTH (FIA_ATD.1, FIA_UAU_EXT.2(1), FIA_UAU_EXT.2(2), FIA_UID_EXT.2(1), FIA_UID_EXT.2(2)) helps to mitigate T.ACCESS by providing measures to uniquely identify and authenticate users through the OS authentication.
	O.PASSWORD The TOE will enforce defined organizational password complexity requirements.	O.PASSWORD (FIA_SOS.1) helps to mitigate T.ACCESS by ensuring that the system passwords of accessors cannot be easily guessed or cracked.

Component	Documents(s)	Rationale
T.ADMIN_ERROR An administrator may incorrectly install or configure the TOE, or install a corrupted TOE resulting in ineffective security mechanisms.	O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management.	O.ROBUST_ADMIN_GUIDANCE (ALC_DEL.1, AGD_PRE.1, and AGD_OPE.1) helps to mitigate T.ADMIN_ERROR by ensuring the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner and to provide the administrator with instructions to ensure the TOE was not corrupted during the delivery process. Having this guidance helps to reduce the mistakes that an administrator might make that could cause the TOE to be configured in a way that is unsecure.
	O.MANAGE The TOE will provide authorized users with the resources to manage and monitor user accounts, TOE resources and security information relative to the TOE.	O.MANAGE (FMT_MOF.1, FMT_MTD.1, FMT_MSA.1, FMT_MSA.3, FMT_SMR.2, FMT_REV.1, FMT_SMF.1) addresses T.ADMIN_ERROR by ensuring only authorized administrators can use the provided resources for managing and monitoring user accounts, TOE resources and security information relative to the TOE.
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	OE.SYSTIME The operating environment will provide reliable system time.	OE.SYSTIME (FPT_STM_EXT.1) is necessary for the Audit logs to contain the accurate system time of events.

Component	Documents(s)	Rationale
recorded, thus masking an user's action.	O. FILESYS The Security features offered by the TOE protect the audit files used by the TOE.	O.FILESYS (FAU_STG.1) addresses T.AUDIT_COMPROMISE by ensuring that the TOE provides the capability to protect the audit files used by the TOE.
T.EAVESDROPPING A malicious user could eavesdrop on network traffic to gain unauthorized access to TOE data.	O.EAVESDROPPING The TOE will encrypt TSF data between the Seagent and administrators on the Client operating system to prevent malicious users from gaining unauthorized access to TOE data.	O. EAVESDROPPING (FCS_CKM.1, FCS_CKM.4,FCS_COP.1, FPT_ITT.1,FTP_TRP.1) mitigates T.EAVESDROPPING by ensuring that all communication to and from the TOE or between components of the TOE are not sent unless they are encrypted.
T.MASK Users whether they be malicious or non-malicious, could gain unauthorized access to the TOE by bypassing identification and authentication countermeasures.	O.AUDIT The TOE will provide measures for recording security relevant events that will assist the authorized users in detecting misuse of the TOE and/or its security features that would compromise the integrity of the TOE and violate the security objectives of the TOE.	O.AUDIT (FAU_GEN.1, FAU_GEN.2, FAU_SAR.1, FAU_SAR.2, FAU_SAR.3, FAU_SEL.1, FAU_STG.1, FPT_STM_EXT.1), addresses T.MASK by providing the authorized users with tools necessary to monitor user activity to ensure that misuse of the TOE does not occur.
	O.IDENTIFY The TOE will provide measures to uniquely identify all users and will maintain their original identity if they issue commands as a super user in the environment.	O.IDENTIFY (FIA_UID.2) addresses T.MASK by eliminating the ability of users to bypass TOE rules via the su command in the environment.
	OE.SYSTIME The operating environment will provide reliable system time.	OE.SYSTIME (FPT_STM_EXT.1) helps to mitigate T.MASK by ensuring the accuracy of the tools necessary to monitor user activity as provided via O.AUDIT.

Component	Documents(s)	Rationale
	OE.AUTH The Operational Environment will provide measures to uniquely identify all users and will authenticate the claimed identity prior to granting a user access to the resources protected by the TOE. The Operational Environment will provide measures to uniquely identify all administrators and will authenticate the claimed identity prior to granting an administrator access to the TOE.	OE.AUTH (FIA_ATD.1, FIA_UAU_EXT.2(1), FIA_UAU_EXT.2(2), FIA_UID_EXT.2(1), FIA_UID_EXT.2(2)) helps to mitigate T.MASK by providing measures to uniquely identify and authenticate users through the OS authentication.

Table 10-2: Threat to Objective Mapping

10.2 Assurance Measures

Component	Document(s)	Rationale
ADV_ARC.1 Security Architecture Design	TOE Design Specification for CA Access Control R12 SP1 v0.4	This document describes the security architecture of the TOE.
ADV_FSP.3 Functional Specification with complete summary	Functional Specification Document for Access Control R12 SP1 v0.4	This document describes the functional specification of the TOE with complete summary.
ADV_TDS.2 Architectural Design	TOE Design Specification for CA Access Control R12 SP1 v0.4	This document describes the architectural design of the TOE.
AGD_OPE.1 Operational User Guidance	• CA Access Control selang Reference Guide • CA Access Control Reference Guide • CA Access Control Endpoint Administration Guide for UNIX • CA Access Control Enterprise Administration Guide • CA Access Control r12 SP1	This document describes the operational user guidance for CA Access Control selang.

Component	Document(s)	Rationale
	Admin Supplemental Guidance version 1.0	
AGD_PRE.1 Preparative Procedures	• CA Access Control Implementation Guide • CA Access Control Release Notes • Evaluated Configuration for CA Access Control r12 SP1	This document describes the preparative procedures that need to be done prior to installing CA Access Control r12 SP1.
ALC_CMC.3 Authorizations Controls	• CA Access Control selang Reference Guide • CA Access Control Endpoint Administration Guide for UNIX • CA Access Control Enterprise Administration Guide • Control of Source Code and Design Documents Policy • CA Access Control Product Documentation Configuration Management Plan r12.0 SP1 • CA AllFusion Harvest Change Manager Configuration Management Plan for CA Access Control r12 SP1	This document describes the authorization controls for the TOE.
ALC_CMS.3 CM Scope	• Control of Source Code and Design Documents Policy • CA Access Control Product Documentation Configuration Management Plan r12.0 SP1 • CA AllFusion Harvest Change Manager Configuration Management Plan for CA Access Control r12 SP1	These documents describe the CM scope of the TOE.
ALC_DEL.1 Delivery Procedures	CA Access Control 12.0 SP1 Download/Installation instruction	This document describes product delivery for CA Access Control and a description of all procedures used to ensure objectives are not compromised in the delivery process.
ALC_DVS.1	• CA Access Control	This document provides an

Component	Document(s)	Rationale
Identification of Security Measures	Implementation Guide • CA Access Control Release Notes • GRC - Global Security - Pre-employment Screening (PES) • Control of Source Code and Design Documents Policy • CA Access Control Product Documentation Configuration Management Plan r12.0 SP1 • CA AllFusion Harvest Change Manager Configuration Management Plan for CA Access Control r12 SP1 • Computer Usage Policy • GIS - Backup Procedure • Global Risk and Compliance / Global Safety and Asset Protection (GSAP) - Security Operations • GRC – BP&C - RIM - Records Security and Confidentiality Policy • RIM - Records Disposal Procedure • Privileged Access Procedure • Global Risk & Compliance - Business Practices & Compliance - Privacy and Data Protection • Inactive User Account Procedure • Server Security Procedure	identification of security measures for the TOE.
ALC_LCD.1 Life-Cycle Definition	Project 360 Reference Guide	This document provides the life-cycle definition of the TOE.
ASE_CCL.1 Conformance Claims	CA Access Control R12 SP1 Security Target v2.0	This document describes the CC conformance claims made by the TOE.
ASE_ECD.1 Extended Components Definition	CA Access Control R12 SP1 Security Target v2.0	This document provides a definition for all extended components in the TOE.

Component	Document(s)	Rationale
ASE_INT.1 Security Target Introduction	CA Access Control R12 SP1 Security Target v2.0	This document describes the Introduction of the Security Target.
ASE_OBJ.2 Security Objectives	CA Access Control R12 SP1 Security Target v2.0	This document describes all of the security objectives for the TOE.
ASE_REQ.2 Security Requirements	CA Access Control R12 SP1 Security Target v2.0	This document describes all of the security requirements for the TOE.
ASE_SPD.1 Security Problem Definition	CA Access Control R12 SP1 Security Target v2.0	This document describes the security problem definition of the Security Target.
ASE_TSS.2 TOE Summary Specification	CA Access Control R12 SP1 Security Target v2.0	This document describes the TSS section of the Security Target.
ATE_COV.2 Analysis of Coverage	• AC Test Plan_July_27.doc • CC_Map_list_Automated_July28.xlsx • BoozAllen (directory in zip BA_QASH_output_files_July29.zip)	This document provides an analysis of coverage for the TOE.
ATE_DPT.1 Basic Design	• AC Test Plan_July_27.doc • CC_Map_list_Automated_July28.xlsx • BoozAllen (directory in zip BA_QASH_output_files_July29.zip)	This document describes the basic design of the TOE.
ATE_FUN.1 Functional Tests	• AC Test Plan_July_27.doc • CC_Map_list_Automated_July28.xlsx • BoozAllen (directory in zip BA_QASH_output_files_July29.zip)	This document describes the functional tests for the TOE.
ATE_IND.2 Independent Testing	• Evaluation Team Test Plan for CA Access Control Version r12 SP1 v1.0 • Booz Allen_CA_AC_R12_IND_Test_Plan.xls	This document describes the independent testing for the TOE.
AVA_VAN.2	Vulnerability Analysis CA	This document describes the

Component	Document(s)	Rationale
Vulnerability Analysis	ACCESS CONTROL R12 SP1 v0.2	vulnerability analysis of the TOE.

Table 10-3: Assurance Requirements Evidence

10.3 EAL 3 Justification

The threats that were chosen are consistent with attacker of low attack potential, therefore EAL3 was chosen for this ST.

10.4 Requirement Dependency Rationale

All Security Functional Requirement component dependencies have been met by the TOE.

10.5 Security Functional Requirements Rationale

The following table provides a mapping with rationale to identify the security functional requirement components that address the stated TOE and environment objectives.

Objective	Security Functional Components	Rationale
O.ACCESS The TOE will provide measures to authorize users to access specified TOE resources once the user has been authenticated. User authorization is based on access rights configured by the authorized users of the TOE.	FDP_ACC.1 Subset access control	FDP_ACC.1 states the TSF shall enforce the Policy on Users to access resources.
	FDP_ACF.1(1) Security attribute based access control	FDP_ACF.1 (1) states the TSF shall enforce the access control rules and policies to objects based on the Resource Record and Accessor Record.
	FDP_ACF.1(2) Security attribute based access control	FDP_ACF.1 (2) states the TSF shall enforce the Administrator Policy to objects based on their respective attributes.
	FTA_TSE.1 TOE session establishment	FTA_TSE.1 ensures the denial of TOE session establishment via Selang unless the administrator does not have WRITE access from the approved terminal and the terminal name/IP address is not included in the TERMINAL class.

Objective	Security Functional Components	Rationale
O.AUDIT The TOE will provide measures for recording security relevant events that will assist the authorized users in detecting misuse of the TOE and/or its security features that would compromise the integrity of the TOE and violate the security objectives of the TOE.	FAU_GEN.1 Audit data generation	FAU_GEN.1 states that the TSF shall be able to generate an audit record for the start-up and shutdown of the audit functions and all auditable events for the level of audit. For each record, the TSF shall record the date/time/type of event/outcome of the event and subject identity. Also, the TSF shall generate an audit report based on user activity, administrator operations, authorized applications, denied authorizations and resources, policies per role, resources, authentication and authorization, and roles. The TSF shall also record the date/time, remote server host name and ID, account name and errors.
	FAU_GEN.2 User identity association	FAU_GEN.2 states the TSF shall be able to associate each auditable event with the identity of the user that caused the event.
	FAU_SAR.1 Audit Review	FAU_SAR.1 states the TSF shall provide the Authorized Administrator with the Auditor attribute with the capability to read all audit information in Seosdb from the audit records. Also, the TSF shall provide the audit records in a manner suitable for the user to interpret the information.
	FAU_SAR.2 Restricted audit review	FAU_SAR.2 states the TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

Objective	Security Functional Components	Rationale
	FAU_SAR.3 Selectable audit review	FAU_SAR.3 states the TSF shall provide the ability to apply filters of audit data based on time or event type/class.
	FAU_SEL.1 Selective audit	FAU_SEL.1 states the TSF shall be able to select the set of audited events from the set of all auditable events based on the following attributes: object identity, user identity, class name, group name, program name, access rights, and authorization result.
	FAU_STG.1 Protected audit trail storage	FAU_STG.1 states the TSF shall protect the stored audit records in the audit trail from unauthorized deletion. Also, the TSF shall be able to prevent unauthorized modifications to the stored audit records in the audit trail.
	FPT_STM_EXT.1 Reliable Time Stamps	FPT_STM_EXT.1 states the Operational environment shall be able to provide reliable time-stamps for use by the TOE.
OE.AUTH The Operational Environment will provide measures to uniquely identify all users and will authenticate the claimed identity prior to granting a user access to the resources protected by the TOE. The Operational Environment will provide measures to uniquely identify all	FIA_ATD.1 User attribute definition	FIA_ATD.1 specifies the security attributes that should be maintained at the level of the user. This means that the security attributes listed are assigned to and are changed at the level of the user. In other words, changing a security attribute in this list associated with a user should have no impact on the security attributes of any other user.

Objective	Security Functional Components	Rationale
administrators and will authenticate the claimed identity prior to granting an administrator access to the TOE.	FIA_UAU_EXT.2(1) User authentication before any action	FIA_UAU_EXT.2 (1) requires each accessor to be successfully authenticated by the Operational Environment before allowing access to the TOE and resources protected by the TOE.
	FIA_UAU_EXT.2(2) User authentication before any action	FIA_UAU_EXT.2 (2) requires each administrator to be successfully authenticated by the Operational Environment before allowing access to the TOE and resources protected by the TOE.
	FIA_UID_EXT.2(1) User identification before any action	FIA_UID_EXT.2 (1) requires each accessor to be successfully identified by the Operational Environment before allowing any other TSF-mediated actions on behalf of that accessor.
	FIA_UID_EXT.2(2) User identification before any action	FIA_UID_EXT.2 (2) requires each administrator to be successfully identified by the Operational Environment before allowing any other TSF-mediated actions on behalf of that administrator.

Objective	Security Functional Components	Rationale
	FTA_TSE.1 TOE session establishment	FTA_TSE.1 ensures the denial of TOE session establishment via Selang unless the administrator has one or more of the following attributes: ADMIN, AUDITOR, PWMANAGER, OPERATOR, GROUP-ADMIN, GROUP-AUDITOR, GROUP-PWMANAGER, GROUP-OPERATOR and the following is true: administrator does not have WRITE access from the approved terminal, and the terminal name/IP address is not included in the TERMINAL class.
O.FILESYS The Security features offered by the TOE protect the audit files used by the TOE.	FAU_STG.1 Protected audit trail storage	FAU_STG.1 states the TSF shall protect the stored audit records in the audit trail from unauthorized deletion. Also, the TSF shall be able to prevent unauthorized modifications to the stored audit records in the audit trail.
O.IDENTIFY The TOE will provide measures to uniquely identify all users and will maintain their original identity if they issue commands as a super user in the environment.	FIA_UID.2 User Identification before Any Action	FIA_UID.2 states that the TOE shall require all users to provide identification before any TSF-mediated actions are allowed. Within the context of the TOE, this refers to the ability of the TOE to track the user’s original claimed identity, even if they re-authenticate using su.
O.MANAGE The TOE will provide authorized administrators with the resources to manage and monitor user accounts, resources, and security information relative to the TOE.	FMT_MOF.1 Management of security functions behaviour	FMT_MOF.1 The TSF shall restrict the ability to modify the behaviour of the functions listed in Table 1-12 Audit Properties to administrators with the AUDITOR and/or GROUP-AUDITOR attribute(s).

Objective	Security Functional Components	Rationale
	FMT_MSA.1 Management of security attributes	FMT_MSA.1 Only Administrators with the ADMIN, AUDITOR, PWMANAGER, and/or OPERATOR authority attribute have the ability to query, modify or delete the security attributes listed as modifiable in Table 1-8: Class Properties.
	FMT_MSA.3 Static attribute initialization	FMT_MSA.3 states the TSF shall enforce the Access Control rules and policies to provide restrictive default values for security attributes that are used to enforce the SFP. It allows administrators with the ADMIN attributes to specify alternative initial values to override the default values when an object or information is created.
	FMT_SMF.1 Specification of management functions	FMT_SMF.1 states the TSF shall be capable of performing the management functions as described in Table 7-6: Global Group and Group Authorization Attributes for Administrators.
	FMT_SMR.2 Restrictions on Security Roles	FMT_SMR.2 requires the TOE to provide the ability to set roles for security relevant authority as well as to restrict the ability to define and assign roles to authorized administrators.
	FMT_REV.1 Revocation	FMT_REV.1 requires the TOE to restrict the ability to revoke security attributes to administrators with the ADMIN attribute.

Objective	Security Functional Components	Rationale
	FMT_MTD.1 Management of TSF data	FMT_MTD.1 The ability to query, modify, or delete the records in Seosdb shall be restricted to administrators with the ADMIN, AUDITOR, PWMANAGER, OPERATOR, GROUP-ADMIN, GROUP-AUDITOR, GROUP-PWMANAGER and/or GROUP-OPERATOR attribute(s).

Objective	Security Functional Components	Rationale
O.EAVESDROPPING The TOE will encrypt TSF data between the Seagent and administrators on the Client operating system to prevent malicious users from gaining unauthorized access to TOE data.	FCS_CKM.1 Cryptographic key generation	FCS_CKM.1 states the TSF shall generate cryptographic keys in accordance with the RSA cryptographic key generation algorithm with a cryptographic key size of 1024 bits and is compliant with RFC 2313.
	FCS_CKM.4 Cryptographic key destruction	FCS_CKM.4 states the TSF shall destroy cryptographic keys with the overwrite method and does not zero keys.
	FCS_COP.1 Cryptographic operation	FCS_COP.1 states the TSF shall perform encryption and decryption in accordance with the AES cryptographic algorithm and a cryptographic key size of 128 bits.
	FPT_ITT.1 Basic Internal TSF Data Transfer Protection	FPT_ITT.1 states the TSF shall protect data from modification and disclosure when that data is transmitted between distributed components of the TOE.
	FTP_TRP.1 Trusted Path	FTP_TRP.1 states the TSF shall provide a communication path between itself and remote 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 or disclosure. The TSF shall allow remote users to initiate communication via the trusted path, and it shall require the use of the trusted path for initial user authentication and all other TSF mediated actions.

Objective	Security Functional Components	Rationale
O.PASSWORD The TOE will enforce defined organizational password complexity requirements.	FIA_SOS.1 Verification of Secrets	FIA_SOF.1 states that the TSF shall provide a mechanism to verify that secrets meet password age, length, and composition requirements. This ensures that all passwords are sufficiently complex for a secure configuration.
O.SELF_PROTECTION The TOE will preserve a secure state and ensure access control to resources when a component of the TOE fails.	FPT_FLS.1 Failure with preservation of secure state	FPT_FLS.1 requires that the TSF shall preserve a secure state when a failure of Seosd, Seagent, and/or Seoswd.
	FRU_FLT.1 Degraded Fault Tolerance	FRU_FLT.1 ensures the operation of access control to resources when the Seosd, Seagent, and/or Seoswd daemons fail.
O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management.	ALC_DEL.1 Delivery Procedures	ALC_DEL.1 describes product delivery and a description of all procedures used to ensure objectives are not compromised in the delivery process.
	AGD_PRE.1 Preparative Procedures	AGD_PRE.1 documents the procedures necessary and describes the steps required for the secure installation, generation, and start-up of the TOE.
	AGD_OPE.1 Operational User Guidance	AGD_OPE.1 describes the proper use of the TOE from a user standpoint.

Table 10-4: Security Functional Requirements Rationale

10.6 Extended Requirements Rationale

10.6.1 FIA_UID

This family defines the conditions under which users shall be required to identify themselves before performing any other actions that are to be mediated by the TSF and which require user identification. The following are extended requirements of FIA_UID:

FIA_UID_EXT.2 (1) was made an extended requirement of FIA_UID.2 because it states that the operational environment instead of the TSF shall require each accessor to identify itself before allowing any other TSF-mediated actions on behalf of that accessor.

FIA_UID_EXT.2 (2) was made an extended requirement of FIA_UID.2 because it states that the operational environment instead of the TSF shall require each administrator to be successfully identified before allowing any other TSF-mediated actions on behalf of that administrator.

10.6.2 FIA_UAU

This family defines the types of user authentication mechanisms supported by the TSF. This family defines the required attributes on which the user authentication mechanisms must be based. The following are extended requirements of FIA_UAU:

FIA_UAU_EXT.2 (1) was made an extended requirement of FIA_UAU.2 because it states that the operational environment instead of the TSF shall require each accessor to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that accessor.

FIA_UAU_EXT.2 (2) was made an extended requirement of FIA_UAU.2 because it states that the operational environment instead of the TSF shall require each administrator to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that administrator.

10.6.3 FPT_STM

This family addresses requirements for a reliable time stamp function within a TOE. The following is an extended requirement for FPT_STM:

FPT_STM_EXT.1.1 was made an extended requirement because it refers to the OS in the operational environment as opposed to the TOE as stated in FPT_STM.1. It states that the underlying operating system in the operational environment shall provide a reliable time stamp from its system clock for use by the TOE’s audit records.

This Security Target does not include any extended Security Assurance Requirements.

10.7 PP Claims Rationale

This Security Target does not claim Protection Profile conformance.