National Information Assurance Partnership

™

Common Criteria Evaluation and Validation Scheme Validation Report

CA Access Control R12 SP1

Table of Contents

1 Executive Summary

The Target of Evaluation (TOE) is R12 SP1 of the CA Access Control product. The TOE was evaluated by the Booz Allen Hamilton Common Criteria Test Laboratory (CCTL) in the United States and was completed in December 2009. The evaluation was conducted in accordance with the requirements of the Common Criteria, Version 3.1 Revision 3 and the Common Methodology for IT Security Evaluation (CEM), Version 3.1 Revision 3. The evaluation was for Evaluation Assurance Level 3 (EAL3) augmented with ALC_FLR.1 (Basic Flaw Remediation) and ASE_TSS.2 (TOE Summary Specification). The evaluation was consistent with National Information Assurance Partnership (NIAP) Common Criteria Evaluation and Validation Scheme (CCEVS) policies and practices as described on their web site (www.niap.ccevs.org).

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.

The CA Access Control product, when configured as specified in the installation guides and user guides, satisfies all of the security functional requirements stated in the TOE's Security Target.

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.

The technical information included in this report was largely derived from the Evaluation Technical Report and associated test reports produced by the evaluation team. The *CA Access Control r12 SP1 Security Target version 2.0, dated 10 October 2009* identifies the specific version and build of the evaluated TOE. This Validation Report applies only to that ST and is not an endorsement of the CA Access Control product by any agency of the US Government and no warranty of the product is either expressed or implied.

2 Evaluation Details

Evaluated Product	CA Access Control R12 SP1
Sponsor & Developer	CA, Inc., Framingham, MA
CCTL	Booz Allen Hamilton, Linthicum, Maryland
Completion Date	December 2009
CC	*Common Criteria for Information Technology Security Evaluation*, Version 3.1 Revision 3, July 2009
Interpretations	None.
CEM	*Common Methodology for Information Technology Security Evaluation*, Version 3.1 Revision 3, July 2009
Evaluation Class	EAL3 Augmented with ALC_FLR.1 and ASE_TSS.2
Description	The TOE is the Access Control R12 SP1 software, which is a security software product developed by CA, Inc.
Disclaimer	The information contained in this Validation Report is not an endorsement of the Access Control product by any agency of the U.S. Government, and no warranty of the Access Control product is either expressed or implied.
PP	None
Evaluation Personnel	Chris Gugel Kevin Micciche John Schroeder Amit Sharma Mark Landon
Validation Body	NIAP CCEVS

2.1 Threats to Security

Table 1 summarizes the threats that the evaluated product addresses.

Table 1 – Threats

Unauthorized users could gain local or remote access to protected objects that they are not authorized to access.
An administrator may incorrectly install or configure the TOE, or install a corrupted TOE resulting in ineffective security mechanisms.

| 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. |
| A malicious user could eavesdrop on network traffic to gain unauthorized access to TOE data. |
| Users whether they be malicious or non-malicious, could gain unauthorized access to the TOE by bypassing identification and authentication countermeasures. |

3 Identification

The product being evaluated is CA Access Control R12 SP1.

4 Security Policy

4.1 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.

4.2 Identification and Authentication

The TOE manages two types of users: 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.

4.3 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.

4.4 Audit

The TOE 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.

4.5 Encrypted Communications

The TOE 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.

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.

5 Assumptions

5.1 Personnel Assumptions

Table 2 – Personnel Assumptions

One or more authorized administrators will be assigned to install, configure and manage the TOE
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
Administrators of the TOE are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided by the guidance documentation

5.2   Physical Assumptions

Table 3 – Physical Assumptions

The TOE will be located within controlled access facilities that will prevent unauthorized physical access

5.3   Connectivity Assumptions

Table 4 – Connectivity Assumptions

The TOE will provide authorization based on already-authenticated sessions

6   Clarification of Scope

The TOE includes all the code that enforces the policies identified (see section 4).

The evaluated configuration of the TOE includes the CA Access Control R12 SP1 product that is comprised of the following:

- Seosd - Seosd is the main CA Access Control authorization daemon/service

- Seoswd (Watchdog) - Seoswd monitors file information and digital signatures of programs that are defined in Seosdb as trusted programs

- Seagent (Agent) - Agent is responsible for communicating with CA Access Control clients through port 5249 over TLS v1.0

- Seosdb (Database) - Seosdb is the main repository of CA Access Control

- SEOS_syscall - SEOS_syscall typically hooks into the operating system at boot up time

- Selang - Selang is a command line interface which is used remotely by administrators to manage the TOE

- 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.

- Seos.audit - Seos.audit is the local storage for the end user's behavior on a local machine

- Seaudit - Seaudit is the application used by the TOE to access and interpret the audit records in a human-readable format

- Sepmdd (PMDB – Policy Model Database) - Sepmdd runs on the same machine as any PMDB which has been configured

- Sepass - Sepass is a replacement for the local passwd command

The scope and requirements for the evaluated configuration are summarized as follows:

1. The Access Control R12 SP1 software (i.e., the TOE) will be installed with the aforementioned components.

Note that the TOE, in its evaluated configuration, was tested on Solaris 10 and Linux Red Hat Advanced 5.0.

2. In addition to the platforms listed in the table above, TLS implementation is also required to run the TOE.

6.1 System Requirements

This section identifies the hardware and software requirements for the platforms described in the evaluated configuration. The TOE was evaluated using Linux Red Hat Advanced Server 5.0 and Solaris 10. The minimum system requirements for each component are illustrated below:

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

7 Architectural Information

The TOE maintains a chained architecture that consists of remote admin to PMDB endpoint to Seosdb endpoints. Figure 2 illustrates the implementation used for the policy model. Additionally, Figure 1 provides an overview of the TOE boundary.

UNIX OS

Seos.Audit

Seaudit
(also connects to
SEOS_syscall)

Local
Administrator

LEGEND
Target of Evaluation
IT Environment

Database
(Seosdb
or
PMDB)

Authorization
Engine
(Seosd)

To Subscribers (Seagent)

TLS v1.0

Sepmdd

Watchdog
(Seoswd)

Agent
(Seagent)

CLIENT OS

Port 5249

TLS v1.0

Language Client
API
(Selang)

Remote
Administrator

Local
File
System

Sepass,
Sesudo
(also connect
to Database)

Local End User

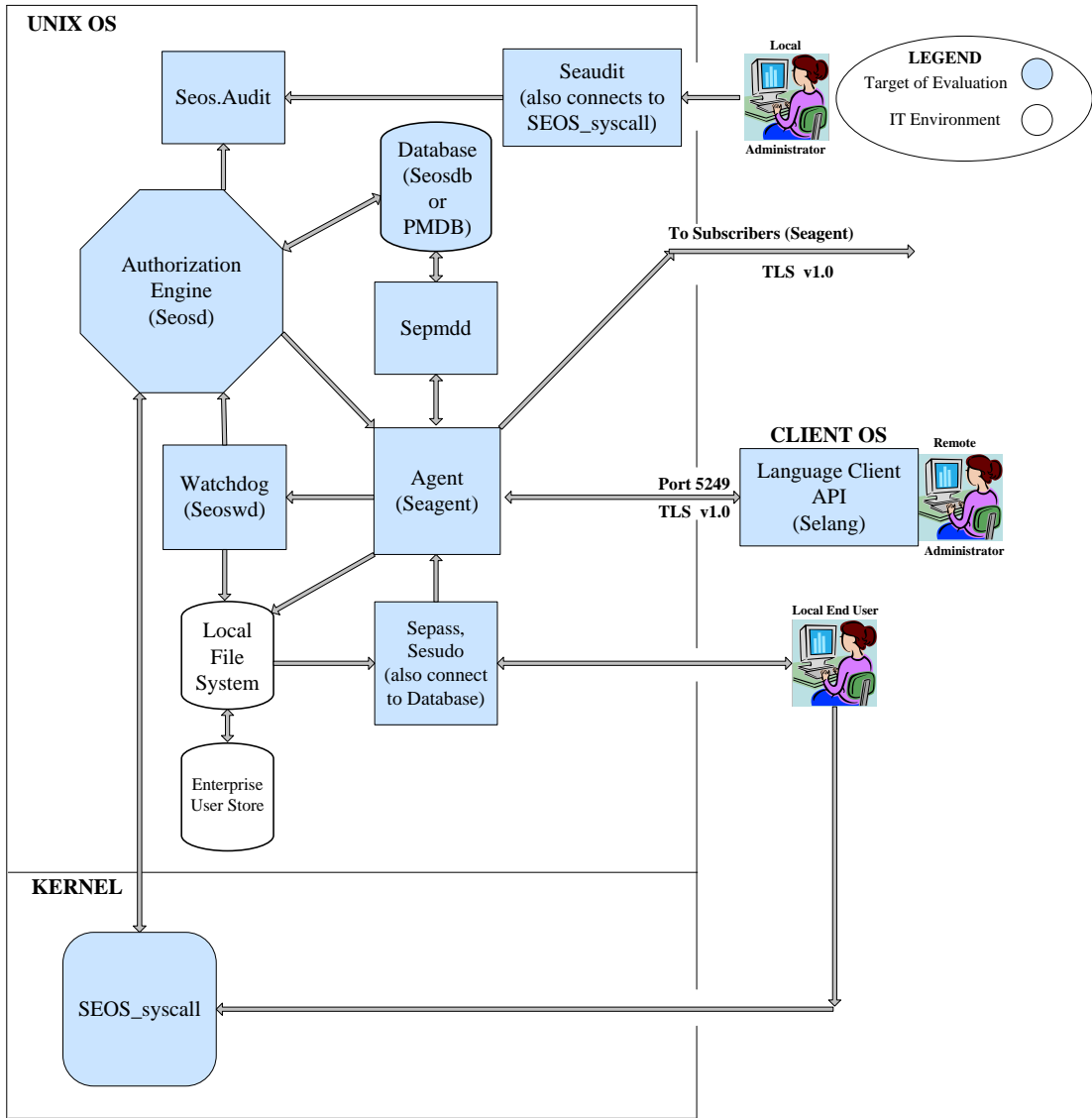Enterprise
User Store

KERNEL

SEOS_syscall

Figure 1 – CA Access Control R12 SP1 TOE Boundary

7.1.1 Policy Model Implementation

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:
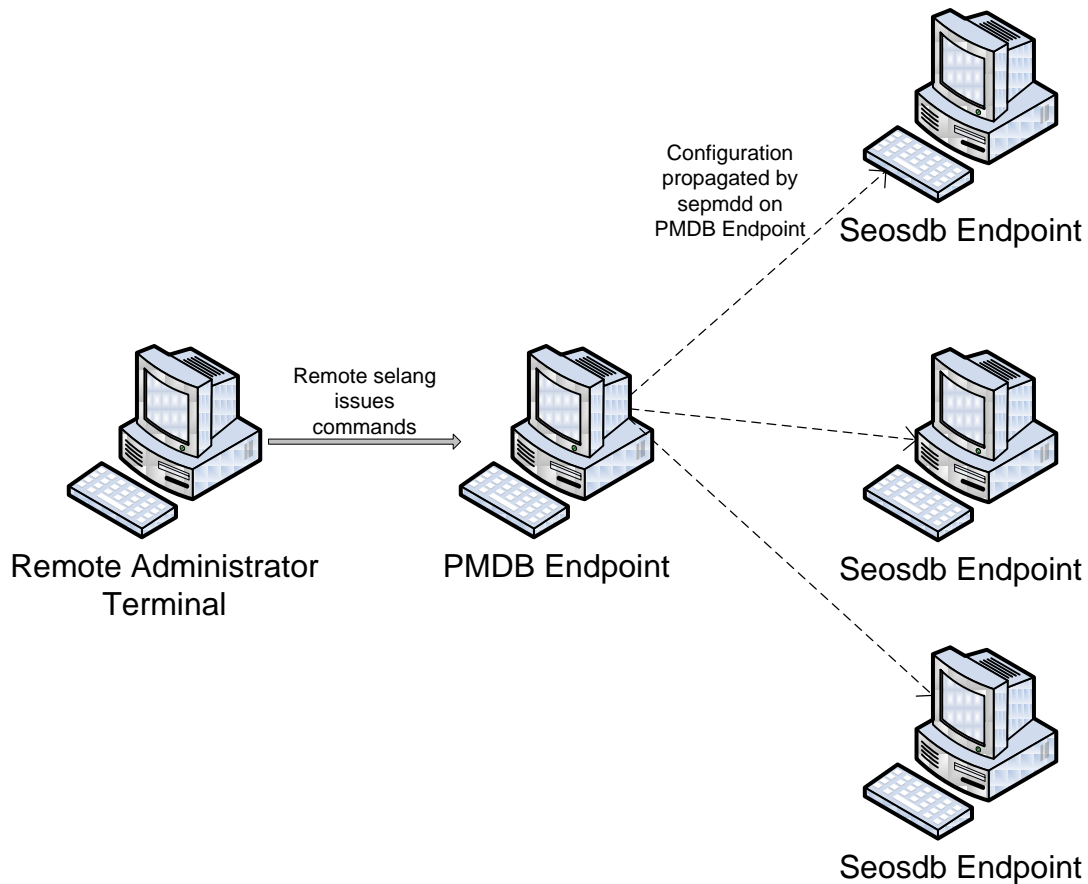
Figure 2 – Policy Model Implementation

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

7.2 TOE Components

7.2.1 Seosdb (Database)

Seosdb is the main repository of CA Access Control and contains information on two types of objects: accessor records and resource records. Seosdb also contains the rules and policies which govern accessor access to objects.

7.2.2 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.

7.2.3 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.

7.2.4 Seagent

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.

7.2.5 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.

7.2.6 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.

7.2.7 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.

7.2.8 Sepmdd (PMDB – Policy Model Database)

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) 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.

7.2.9 Selang Command Line Interface

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.

8 Documentation

The documents were evaluated to satisfy assurance requirements:

Component	Document(s)	Rationale
ADV_ARC.1 Security Architecture Design	TOE Design Specification for CA Access Control R12 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 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 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	This document describes the operational user guidance for CA Access Control selang.
AGD_PRE.1 Preparative Procedures	• CA Access Control Implementation Guide • CA Access Control Release Notes	This document describes the preparative procedures that need to be done prior to installing CA Access Control r12 SP1.

Component	Document(s)	Rationale
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 for UNIX 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 for UNIX 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.

Table 5 – Assurance Documents Evidence

These documents are provided to customers who have purchased the TOE.

9   TOE Acquisition

The NIAP-certified Access Control product is acquired via normal sales channels, and digital delivery of the TOE is coordinated with the end customer by CA, Inc.

10 **IT Product Testing**

The test team's test approach is to test the security mechanisms of the CA Access Control by exercising the external interfaces to the TOE and viewing the TOE behavior either remotely, or on the platform. Each TOE external interface is described in the appropriate design documentation (e.g., FSP) in terms of the relevant claims on the TOE that can be tested through the external interface. The ST, TOE Design Specification (TDS), Functional Specification (FSP), and the vendor's test plans were used to demonstrate test coverage of all *appropriate* EAL3 requirements for all *security relevant* TOE external interfaces. TOE external interfaces that were determined to be *security relevant* are interfaces that

- change the security state of the product,
- permit an object access or information flow that is regulated by the security policy,
- are restricted to subjects with privilege or behave differently when executed by subjects with privilege, or
- invoke or configure a security mechanism.

Security functional requirements were determined to be *appropriate* to a particular interface if the behavior of the TOE that supported the requirement could be invoked or observed through that interface.

The evaluation team created a test plan that contained the vendor functional test suite, and supplemental functional testing of the vendor's tests. Booz Allen also performed vulnerability assessment and penetration testing.

10.1 TEST METHODOLOGY

10.1.1 Vulnerability Testing
The evaluation team executed the following vulnerability tests against CA Access Control R12 SP1:

- Eavesdropping on Communications (wireshark 1.0, arpspoof 2.4)
    - The team attempted to intercept any TOE involved network traffic. The attack machine executed an arp poisoning attack so that all network traffic between two nodes on a switched LAN would be tunneled through the attack machine before it reached its destination. A sniffer would then be used to analyze the network traffic and attempt to view any confidential information that may have passed over the network.
- Port Scanning
    - The team attempted to identify any way to subvert the security of the TOE by executing a side channel attack. A port scanner ran against all TOE systems in an attempt to identify any open ports. Any port on a system that accepted external connections could potentially represent an attack vector. This test identified any such ports and would attempt to enumerate them to determine their original purpose.

- Buffer Overflow/Format String/Unexpected Input Attack
  - The team attempted to discover and exploit any software errors that did not appropriately handle various non standard inputs. For this test a program known as a fuzzer was used. This program contains a list of malicious inputs. It injects these inputs into any given template then sends the result to a listening port. These malicious inputs form 3 categories.
    - Buffer Overflows: In this case, larger and larger inputs are injected to try to overflow a buffer on the server and corrupt its program stack.
    - Format Strings: In this case, format strings are injected to attempt to see if they are not handled correctly by the server.
    - Special Characters: In this case, unexpected special characters are injected in an attempt to induce non standard behavior.
- Vulnerability Scanner
  - The team used the Nessus Vulnerability scanner to test any and all open interfaces on any applicable systems of the TOE. The scanner probed all of the SSL cipher suites accepted by an SSL server and reported on the existence of long strength ciphers
- Denial of Service – TCP Malformed Packet Flooding
  - The team attempted to exercise the stability of the IP stack and its components by sending a large amount of TCP packets and malformed TCP packets in an attempt to overload the application. If successful, the TOE would crash and not allow any connections until the TOE is rebooted
- SSL/TLS – Eavesdropping on Communications
  - This test is a version of "Eavesdropping on Communications" that is specific for the SSL/TLS interfaces of the TOE. That means that this test analyzed all traffic between the `<AC_admin>` machine and the `<AC>` machine. All communications were expected to travel encrypted via SSL/TLS. Therefore, no confidential information should be leaked.
- Client Authentication Attack
  - Analysis showed that the SSL interfaces of the TOE perform mutual authentication via X.509 certificates on both the client and server sides of the connection. The client side authentication is available in SSL but is usually not used in higher level protocols (https, ssh, etc). Therefore, this test would attempt to exploit any incorrect use of client side authentication. The attack machine attempted to authenticate to the server computer using no certificate or using a self signed certificate.
- Local Authentication Bypass
  - The team attempted to bypass the restrictions placed on a user by changing the linux/unix root password. This was done by rebooting the server and booting into single logon mode, which allows a local user to escalate privileges and modify system files.
- Local Resource Tampering
  - This test attempted to tamper with some of the local resources used by Access Control. It dealt with system processes, audit data, configuration files, etc. The intent was to try to log into the system as an untrusted user and gain

access to sensitive data by modifying Access Control or the way that it operates.

10.1.2 Vulnerability Results

The following lists any issues that were discovered as a result of the vulnerability testing process. These issues along with the related guidance for mitigation have been included in the Common Criteria Addendum to the product Administrator Guidance.

- *Default Installation Required No TLS Encryption/Authentication*
    - o The default installation of Access Control does not require the use of TLS over the remote administrative interface. TLS encryption is needed to protect the confidentiality of the commands being sent and TLS authentication is required to protect the integrity of Access Control commands and the Access Control database. Without TLS configured, all traffic is sent via a symmetric encryption method and there is no authentication of remote management. TLS encryption was enabled in testing configuration and guidance was included in the installation documentation ensuring that any deployed system would be protected.

- *Default Encryption Kits*
    - o Access Control comes with preinstalled encryption keys and certificates that are used for TLS communications once they are enabled. They are included with the default UNIX installation and are available to anyone that has access to the installation media.   The authentication of remote management is performed using TLS certificates. The presence of default keys means that it is possible to remotely manage Access Control unauthenticated. This was shown in the testing environment.
    - o The keys can be rotated using the sechkey utility included with Access Control.   Instructions on how to rotate keys has been included in the administrative documentation to ensure the protection of a deployed system.

- *Starting Access Control on System Boot*
    - o It became apparent during testing that Access Control was able to be subverted if the proper steps were not taken to ensure that all Access Control daemons started upon system boot. Without this in place, a user could perform a hard kill of the system and then access it without the Access Control protections in place.
    - o Guidance has been included in the administrator documentation ensuring that the system is brought up in a secure fashion.

- *Access Control Administrative Access to UNIX Root User*
    - o There was one use case identified where Access Control was able to be subverted. If a system was configured so that the UNIX root user was not an Access Control admin, it was possible for that root user to escalate his privileges. This is due to the fact that the Access Control installation files as

well as critical system files are protected by UNIX file permissions by default and not by rules in Access Control. A root user could therefore tamper or delete Access Control installation files and restart the system in a state where the Access Control daemons could not be started.

o Guidance has been included stating that the root user should always be an Access Control admin until Access Control installation files and critical system files are protected. It is possible to create administrators having root as the super-admin, but it should not be assumed that privileges can be denied to root without first protecting those files.

11 Results of the Evaluation

The evaluation was carried out in accordance with the Common Criteria Evaluation and Validation Scheme (CCEVS) process and scheme. The evaluation demonstrated that the CA Access Control R12 SP1 TOE meets the security requirements contained in the Security Target.

The criteria against which the CA Access Control R12 SP1 TOE was judged are described in the Common Criteria for Information Technology Security Evaluation, Version 3.1 Revision 3, July 2009. The evaluation methodology used by the evaluation team to conduct the evaluation is the Common Methodology for Information Technology Security Evaluation, Version 3.1 Revision 3, July 2009. The Booz Allen Hamilton Common Criteria Test Laboratory determined that the evaluation assurance level (EAL) for the CA Access Control R12 SP1 TOE is EAL 3. The TOE, configured as specified in the installation guide, satisfies all of the security functional requirements stated in the Security Target.

The evaluation was completed in December 2009. Results of the evaluation and associated validation can be found in the Common Criteria Evaluation and Validation Scheme Validation Report.

12 Validator Comments/Recommendations

The "Supplemental Administrative Guidance" (version 1.0, October 19, 2009) and the "Evaluated Configuration for CA Access Control r12 SP1" (October 2009) define the recommendations and secure usage directions for the TOE as derived from testing.

System integrators should note that identification credentials come from the host machine. No end user remote credentials are passed to the host machines.

13 Security Target

The security target for this product's evaluation is *CA Access Control r12 Security Target version 2.0, dated October 10, 2009.*

14 **List of 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

15 **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.

Term	Definition
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 are governed the same way as 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 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.

Term	Definition
Watchdog	Also known as Seoswd. This daemon constantly checks that the other Access Control Services are running. If Seosd has stopped, Seoswd restarts it.
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.

16 **Bibliography**

1. Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, Version 3.1 Revision 3.
2. Common Criteria for Information Technology Security Evaluation – Part 2: Security functional requirements, Version 3.1 Revision 3.
3. Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance requirements, Version 3.1 Revision 3.
4. Common Evaluation Methodology for Information Technology Security Evaluation, Version 3.1 Revision 3.
5. *CA Access Control r12 Security Target version 2.0, October 10, 2009*
6. Evaluation Technical Report for a Target of Evaluation CA Access Control R12 SP1 Security Target v2.0 Evaluation Technical Report v3.0 dated 26 October 2009.