CA Directory r12.0 SP3
Security Target
for EAL3+

Version 3.0
November 30, 2010

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 EAL3.

1.1.1 ST Identification

ST Title:	CA Directory r12.0 SP3
ST Version:	3.0
ST Publication Date:	November 30, 2010
ST Author:	Booz Allen Hamilton

1.1.2 Document Organization

Chapter 1 of this ST provides identifying information for the CA Directory r12 SP3. 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, and describes the components and/or applications that are excluded from the evaluated configuration.

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 (SFRs) and Security Assurance Requirements (SARs).

Chapter 7 describes the Security Functional Requirements.

Chapter 8 describes the Security Assurance Requirements.

Chapter 9 is the TOE Summary Specification (TSS), a description of the functions provided by CA Directory to satisfy the SFRs and SARs.

Chapter 10 is the Security Problem Definition Rationale and provides a rationale or pointers to a rationale, for security objectives, assumptions, threats, requirements, dependencies, and PP claims for the chosen EAL, any deviations from CC Part 2 concerning SFR dependencies, and a mapping of threats to assumptions, objectives, and SFRs. It also identifies the items used to satisfy the Security Assurance Requirements for the evaluation.

1.1.3 Convention Definitions

The requirements in this document are divided into assurance requirements and two sets of functional requirements. The first set of functional requirements, which were drawn from the Common Criteria, is designed to address the core System requirements for self-protection.

The CC permits four functional component operations—assignment, refinement, selection, and iteration —to be performed on functional requirements. This ST will highlight the four operations in the following manner:

- Assignment: allows the specification of an identified parameter. Indicated with ***bold text and italics*** if further operations are necessary by the Security Target author;
- Refinement: allows the addition of details. Indicated with *italics* if further operations are necessary by the Security Target author;
- Selection: allows the specification of one or more elements from a list. Indicated with <u>underlined text</u>; and
- Iteration: allows a component to be used more than once with varying operations. The Security Target author has identified iterations with a numbering value between two parenthesis (Example: FAU_SEL.1(1)). As can be seen in the example, the value (1) is an iteration of selectable auditing. Each iteration follows this structure, such that an SFR iteration is mapped to a value (X) where “X” represents the number of the iteration.

1.1.4 Terminology

This section defines the terminology used throughout this ST. The terminology used throughout this ST is defined in Table 1-2: Terminology Definitions. This table is to be used by the reader as a quick reference guide for terminology definitions.

Terminology	Definition
Abstract Syntax Notation	The method of sending data over dissimilar communication systems.
Access Control List	An Access Control List (ACL) specifies the users that are granted access to a resource and the type of access to which the user is granted.

Access Control Agent	An Access Control (AC) Agent is any of three types of roles that can be assigned to a user who wishes to access the TOE. These types include Registered User, Administrative User, and SuperUser. A user of the TOE can be both an Administrative User and a Registered User. This is dependent upon what subtrees the roles are set to. (e.g. Administrative User for one subtree and Registered User for another subtree).
Administrative User	An Administrative User has read/write access to a subtree of the Directory Information Tree (DIT). The DIT may be contained in a single Data Store or distributed over a number of Data Stores, each server by its own DSA.
Anonymous User	Anonymous users are those users that only have read access to those files stored in the public repository and cannot modify any information stored on the TOE.
Data DSA	A Data DSA is a TOE configuration in which there is a Data Store. The TOE can perform the operations and execute them locally on its Data Store. A Data DSA can also receive commands from a Router DSA to perform operations on its Data Store.
Data Store	The main repository that contains information on users and attributes assigned to those users.
Directory Administrator	A Directory Administrator is an operating system user on the machine on which the DXserver is installed who has access to the Configuration files for the TOE.
Directory Information Base	The DIB is the set of information managed by the Directory. DIB applies to the Data Store of the TOE.
Directory Server Agent	The DSA is a single instance of the Directory Server. An X.500 program manages the Directory Information Base (DIB), also known as white pages. It accepts requests from the Directory User Agent (DUA) counterpart in the workstation.
Directory User Agent	A DUA is a client that can send a number of different requests to the Directory Server Agent (DSA) via the DAP protocol. Similarly an LDUA is a client that uses the LDAP protocol.
Distinguished Name	The attribute within the Directory which uniquely identifies any entry in the Directory, not necessarily a user. The DN is also used to authenticate to the TOE along with a password and based on the privileges or ACLs defined.
Host	The machine where CA Directory components are installed.

Lightweight Directory Access Protocol	LDAP is a network protocol used for accessing of information directories.
Object	A record on the TOE or third party trusted DSA or a resource on the OS.
Policy	A rule or group of rules assigned to a record of a user or resource (ex. ACL).
Public User	Synonymous with Anonymous User
Registered User	Any user that has an entry that is stored in the Data Store. A Registered User can read from a particular subtree of TSF data.
Replicated DSA	When configured for High Availability, the TOE has a second DSA that is paired with the original DSA on the TOE. If an issue occurs that kills the original DSA, the TOE will transfer communication to the Replicated DSA for 180 second intervals while waiting for the original DSA to recover.
Router DSA	A Router DSA is a TOE configuration where there is no Data Store. Operations sent to the Router DSA are directed to Data DSAs which can then perform the operation.
Rule	A rule is written by a System Administrator or Directory Administrator to determine a user's access to a resource.
Subject	An individual (Registered User, Administrative User SuperUser, Trusted Peer DSA) in the context of attempting to access protected resources (either managed by the TOE or part of it).
Subtree	A portion of the Directory data. Often used when defining scope of a user, allowing them to access only a certain portion of information.
SuperUser	A SuperUser is a user of the TOE that has read/write to all Directory data within the DIT/DIB.
System Administrator	A System Administrator is any user, not on the TOE, who has root access to the underlying OS. This can include a Directory Administrator.
Trusted Peer DSA	See information in Table 2-2. The Trusted Peer DSA component includes DAP servers, LDAP servers, and any other DSA (a single instance of a Directory Server).
User	A user of the TOE that can have differing levels of access to different subtrees of the TOE. These levels include: Administrative User, Registered User, or SuperUser with an entry in the DIT.

Table 1-1: Customer Specific Terminology

Term	Definition
Authorized user	A user who, in accordance with proper authentication/authorization, performs an operation.
External IT entity	Any IT product or system, untrusted or trusted, outside of the TOE that interacts with the TOE. If trusted, it is referred to as Trusted third party entity.
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

1.1.5 Acronyms

The acronyms used throughout this ST are defined in Table 1-3: Acronym Definitions. This table is to be used by the reader as a quick reference guide for acronym definitions.

Acronym	Definition
ACL	Access Control List
ASN	Abstract Syntax Notation
CLI	Command Line Interface
DAP	Directory Access Protocol
DIB	Directory Information Base
DISP	Directory Information Shadowing Protocol
DN	Distinguished Name
DSA	Directory Server Agent
DSP	Directory Server Protocol
DUA	Directory User Agent
EAL	Evaluation Assurance Level
FSP	Functional Specification
GUI	Graphical User Interface
I&A	Identification and Authentication
IT	Information Technology
LDAP	Lightweight Directory Access Protocol
OS	Operating System
PP	Protection Profile
SAR	Security Assurance Requirements
SFR	Security Functional Requirements
SNMP	Simple Network Management Protocol
ST	Security Target
TCP/IP	Transmission Control Protocol / Internet Protocol
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Function
TSFI	TOE Security Function Interface
TSS	TOE Summary Specification
X.500	International Standard for Electronic Directories including DAP, DSP, DISP, DOP
X.501	X.500 Series Standard for Directory Models

Table 1-3: Acronym Definitions

1.1.6 References

[1] Common Criteria for Information Technology Security Evaluation, CCMB-2009-07-004, Version 3.1 Revision 3, July 2009

[2] CA Directory Administration Guide for r12.0

[3] CA Directory Installation Guide for r12.0

[4] CA Directory Reference Guide for r12.0

[5] CA Directory Release Summary for r12.0

1.2 TOE Reference

CA Directory r12.0 SP3 Build 4346

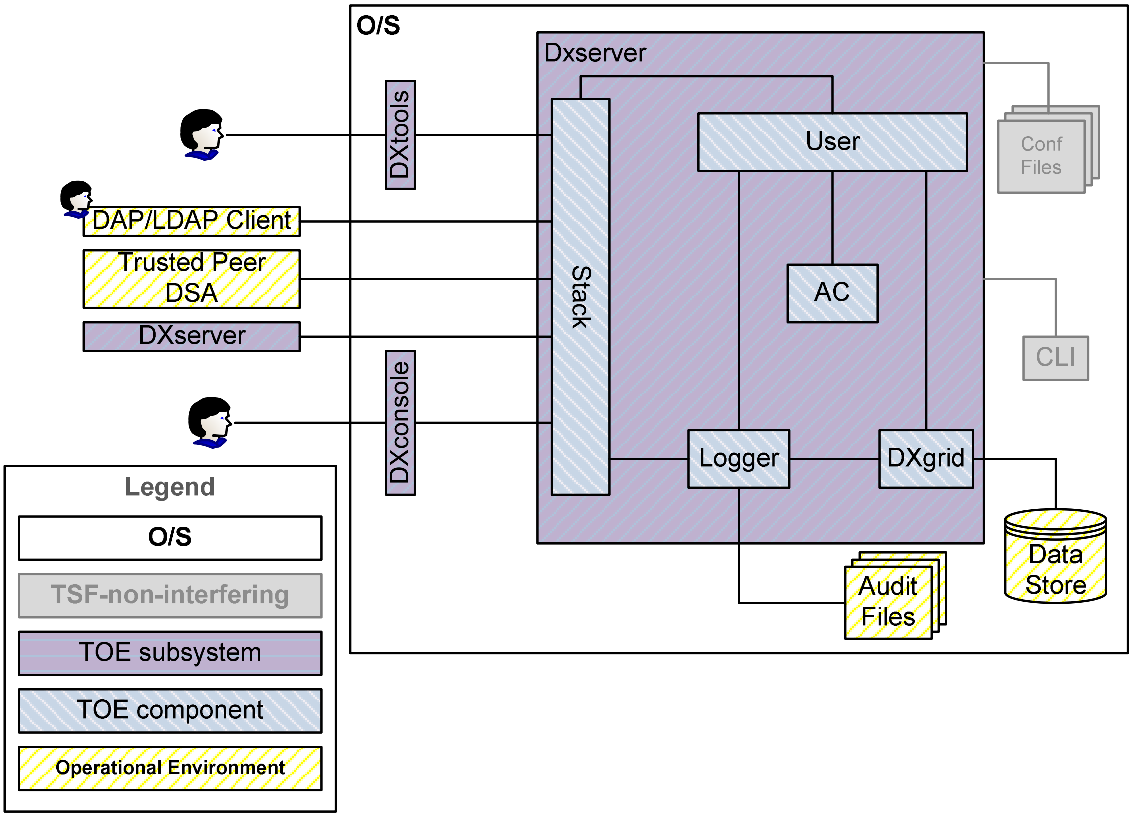

Figure 1 – TOE Boundary as Data DSA

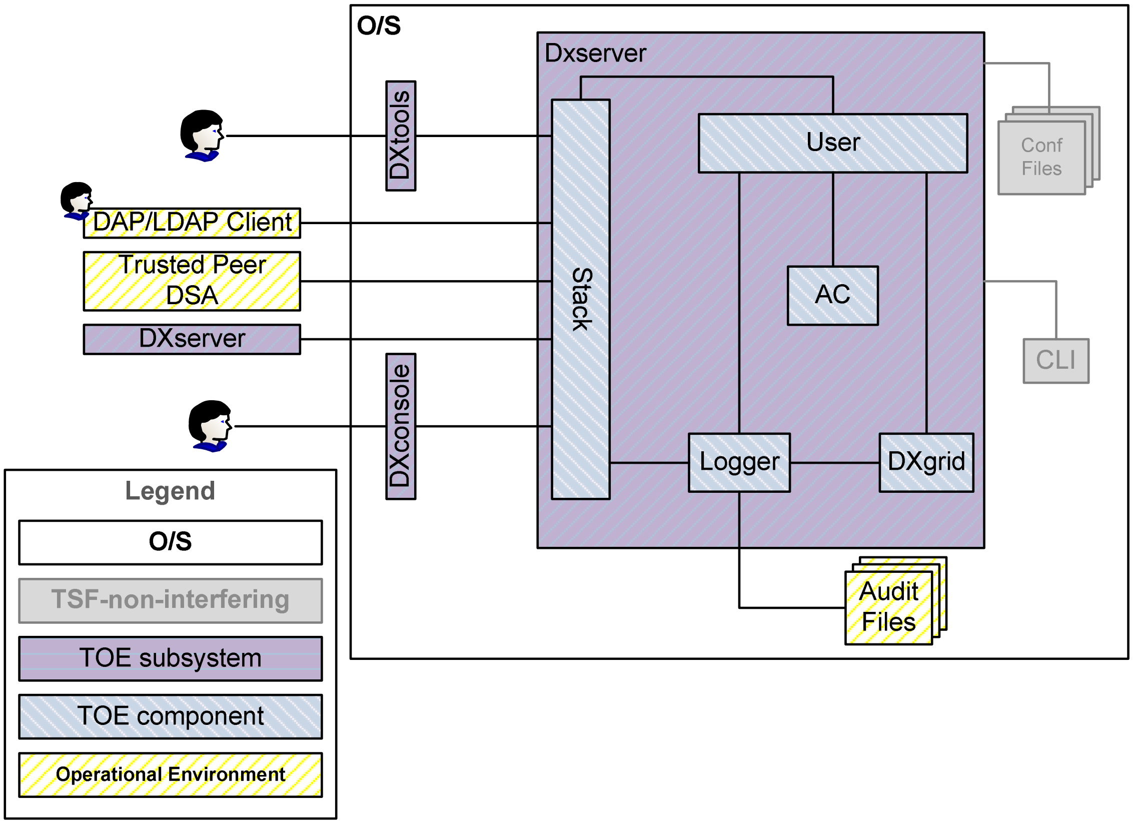

Figure 2: TOE Boundary as Router DSA

As shown above, the TOE can be configured into two modes. These modes are a Router DSA and Data DSA. The functionality performed in these modes is identical except for the configuration of data. In a Data DSA, the TOE contains a Data Store which can be used to locally maintain Directory data. In a Router DSA, the TOE contains no Data Store but has a list of Data DSAs it communicates with. When an operation is sent to the TOE via a TSFI, the TOE finds the location of that data or subtree and passes the operation to that Data DSA.

As illustrated in Figures 1 and 2, the TOE boundary contains three subsystems – DXserver, DXconsole, and DXtools. Additionally, the external interfaces outside of the underlying Operating System provide connections to the Directory Server. The DXserver provides the main enforcement for all TSFs, including Audit Generation, Access Control, Data Protection, Identification and Authentication, Protection of TSF Data, Resource Utilization, and Security Management. The LDAP client (JXplorer in the evaluated configuration) is also capable of performing management of TSF data as long as the user is authorized on the DSA. Listed below are operations available to be performed on the TOE.

- Authentication through an LDAP/DAP client, DXconsole, DXtools, or by a trusted peer DSA. This occurs when one of the aforementioned entities is attempting to connect to a DSA. Once a bind has been performed and a connection between the client and DSA has been established no more authentications are performed.

- With authorization through DXtools, LDAP/DAP clients, and DXconsole, a user with the proper ACL (access control level) can perform search, modify, or delete operations on the TOE.
- If the DSA on the TOE fails, a Replicated DSA can be passed operations to ensure functionality. The trusted peer DSAs also serve as Distributed DSAs, providing remote connection to data that is not stored on the DXserver Data Store.

The Directory can be setup as a standalone, Replicated DSA (for high availability), and/or distributed Directory. For more information, the Administration guide contains information regarding the set up of Replication and set up of Distribution and Routing. DXconsole and DXtools provide interfaces into the TOE, enforcing Security Management requirements by allowing an authorized user to manage the TOE through the DXconsole and DXtools. The DXtools are a collection of binaries that can be used for managing the directory data (when the DSA is stopped) and for accessing the data via the LDAP interface when the DSA is running. A user with proper authorization can perform management on the data store through this interface. DXconsole allows for viewing of DSA data as well as management of that data and access control levels.

Finally, the Data Store is accessed by users of the TOE as well as by third party trusted entities to perform data management of the TOE. It is stored on the operational environment along with the audit and configuration files and requires Directory Administrator/System Administrator access to the underlying Operating System to perform any modifications to the data.

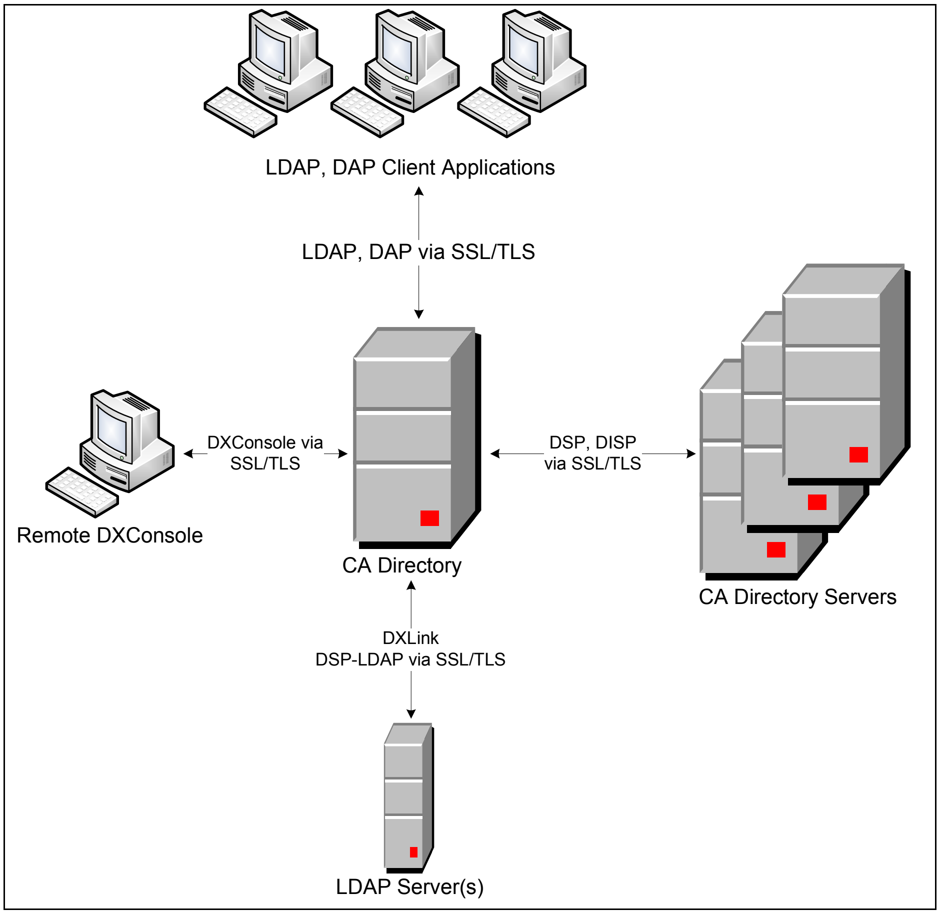

Figure 3 – TOE Deployment

As illustrated in Figure 2, the TOE deployment contains a remote console connected through a secure connection along with LDAP and DAP client applications that access the main CA Directory which provides access control permissions to data.

1.3 TOE Type

The TOE type for CA Directory r12 SP3 is a network access control. A directory is a service for information management. It stores information about people, resources, and systems. A directory allows users to instantly look up critical everyday information. By applying access control rules to the directory and requiring certain levels of access to perform security management on the TOE, CA Directory r12 SP3 is working as an access control to the network by maintaining a Data Store of those user's Distinguished Names and their privileges.

2 TOE Description

CA Directory is a Directory software application, and as such it provides a system to store and manage electronic information. CA Directory R12 SP3 can operate in a standalone mode or, as typical for directories, provide Directory services to other applications, operating as part of larger systems. CA Directory can also operate in a large distributed Directory system and itself be deployed as a large distributed Directory, supporting Directory functionality such as replication and chaining, involving distributed authentication mechanisms. The scope of the evaluation includes the CA Directory server (DXserver), DXtools and the administrative interface (DXconsole) within the TOE and the Data Store in the Operational Environment. The DXserver and DXconsole implement the Directory security services to its users through DAP, LDAP, DSP, and DISP interfaces. These are the X.500 interfaces visible to administrative and non-administrative users. DAP and LDAP are used for human users or Directory-enabled applications to access the Directory repository information. DSP and DISP are used when the Directory works with other standard Directory servers (DSAs) as part of a Directory system, and are used for distributed authentication and replication, respectively. These other DSAs, external to the TOE, are referred to in this ST as ‘Trusted Peer DSAs’.

2.1 Evaluated Components of the TOE

The following table describes the TOE components in the evaluated configuration:

Component	Definition
DXserver	DXserver is the central component of CA Directory. Each DSA uses a DXserver process. Many DSAs can be run on one computer, but DXserver needs to be installed only once on each. The DXserver provides: • Network connection between the TOE, a Replicated DSA, users, telnet, and trusted peers DSAs. • Security management of the TOE and authentication to the TOE. • Processing operations generated from user interaction with the TOE through any of the possible interfaces. • Creation of audit entries in the configured audit files The DXserver in the operational environment is provided as a Replicate of the pre-existing TOE or as a distributed TOE for sharing of data over several DSAs. This is still a TOE-to-TOE communication. If the DXserver is used for replication from the original TOE DSA, it receives updates made during

	regular operation or at initialization and keeps up to date with the original DSA. If that DSA fails, the TOE checks the configuration data in memory and finds the Replicated DSA, quickly changing to that DSA and minimizing downtime to only a brief second.
DXconsole	DXconsole is actually a component of DXserver but is a user interface and therefore shown as a component of the operational environment because it can be accessed either locally or remotely by a user through a telnet connection. The DXconsole is a command line interface (CLI) through telnet which provides Administrators of the TOE access to audit information and repository data. In the evaluated configuration, the DXconsole is the only user-interface accessing the TOE. The web server interface is disabled in the evaluated configuration and is rationalized in Section 2.3.1.
DXtools	Directory Administration maintenance utilities. Used during initialization and configuration before TOE is in use. Some operations are available during regular use.

Table 2-1: Evaluated Components of the TOE

2.2 Components and Applications in the Operational Environment

The following table lists components and applications in the environment that the TOE relies upon in order to function properly:

Component	Definition
LDAP Client	An interface used by authorized users of the DSA to search, modify, and query data stored on the DSA. In the evaluated configuration, the LDAP client will be JXplorer. For testing purposes of the TOE, DUA and LDUA are also included in the evaluation. These agents allow for command line entries as well as test scripts to be run for automation of tests.
Data Store	The Data Store is installed on the Underlying Operating System and is provided with the installation of CA Directory R12 SP3. User data is stored on the data store while access levels (if configured to have RBAC) are mapped into memory to be leveraged by the TOE during runtime.
Audit Files	The audit files on the TOE are a configurable set of flat files that collect information specific to the audit file type. These audit types, as specified in the forms of selectable auditing are: Summary,

	Stats, Query, Connect, Update, Cert, Alert, Warn, Diag, Time, SNMP, and Alarm. Additional information about configuring these files can be found in the Administrative Guidance for CA Directory R12.
Configuration Files	The configuration files are stored on the underlying Operating System where the TOE DXserver has been installed and provides the knowledge files that are used during initialization of the TOE. These configurations determine access control rules and mapping of DSAs and their Replicates or distributions. Once initialized, the configuration data is stored in runtime memory and any changes made are only made to those configurations in memory. If restarted or shutdown, the TOE returns to the configuration defined in the configuration files. A user who wishes to alter these configurations requires Directory/System Administrator access to the underlying OS.
Trusted Peer DSA	The trusted peer DSAs mapped to the TOE includes both LDAP and X.500 servers. It is similar to that of the distributed capability discussed in DXserver below. If a user on the TOE or a Trusted Peer DSA wishes to communicate with the other to gather Directory data, authentication occurs between the two points and if successful, data is passed to result in a successful operation. The Trusted Peer DSA component includes DAP servers, LDAP servers, and any other DSA (a single instance of a Directory Server).

Table 2-2: Evaluated Components of the Operational Environment

2.3 Excluded from the TOE

The following optional products, components, and/or applications can be integrated with CA Directory but are not included in the evaluated configuration. They provide no added security related functionality. They are separated into three categories: not installed, installed but requires a separate license, and installed but not part of the TSF.

2.3.1 Not Installed

1. DXmanager – This allows monitoring and configuring of CA Directory. This package is not required to run CA Directory and is therefore excluded from the evaluated configuration. If this is not installed then the core Directory is configured with text configuration files.

2.3.2 Installed but Requires a Separate License

No components installed with CA Directory required a separate license.

2.3.3 Installed but not part of the TSF

1. The Local CLI (Command Line Interface) - Part of the underlying Operating System and is used solely for the initialization or the TOE. This requires Directory Administrator access to the TOE and is not used once the TOE is in a successful running state. For this reason, it was determined that this component is outside of the scope of any TSF.

2. The Configuration Files – Files which are stored on the underlying Operating System where the TOE DXserver has been installed and provides the knowledge files that are used during initialization of the TOE. These configurations determine access control rules and mapping of DSAs and their Replicates or distributions.

   Once initialized, the configuration data is stored in runtime memory and any changes made are only made to those configurations in memory. If restarted or shutdown, the TOE returns to the configuration defined in the configuration files. A user who wishes to alter these configurations requires Directory Administrator access to the underlying OS. These files are not part of the TSF because they are used only during initialization. They do not provide configuration decisions once CA Directory R12 SP3 is initialized. When the Local CLI initializes the TOE, the configuration data is loaded into the TOE's runtime memory and the files are no longer used during regular use of the CA Directory. Memory is then responsible for maintaining the evaluated configuration.

3. DXadmind – DXadmind is the interface between the user and the TOE when DXmanager is installed. Even though DXadmind is installed, it is not configured for the reason that DXmanager is not incorporated into the evaluated configuration.

2.4 Physical Boundary

The Target of Evaluation (TOE) is a distributed, software-only TOE. The TOE requires dedicated hardware.

CA Directory supports the following OS platforms:

- Solaris 10 SPARc or x86: Used in the Evaluated Configuration
- Windows Server 2003 64-bit: Used in the Evaluated Configuration
- Linux x86/ servers: Out of Scope
- Windows x86/64 servers: Out of Scope
- AIX servers: Out of Scope
- HPUX RISC: Out of Scope
- HPUX IA64: Out of Scope

The following table lists the minimum hardware requirements of the TOE.

Computer Role	Minimum RAM	Minimum Disk Space	Require Software
Directory Host	2 GB	100 GB	JRE 1.6.0_16 for JXplorer

Table 2-3: Minimum Hardware Requirements for the TOE

2.5 Logical Boundary

The logical boundary of the TOE is described in the terms of the security functionalities that the TOE provides to the systems that utilize this product for intrusion detection.

The logical boundary of the TOE will be broken down into six security classes: Security Audit, Data Protection, Identification and Authentication, Security Management, and High Availability. Listed below are the security functions with a listing of the capabilities associated with them:

2.5.1 Security Audit

The TOE generates audit records for selected security events. The records are stored in the log text files on the DXserver platform and are viewable provided that an application in the TOE environment is available to read the audit records. Any standard text viewing application is capable of viewing the audit records.

A CA Directory Administrator can define an action to take when a violation occurs (e.g. failed authentication attempt limit reached). These actions are stored in the configuration store and are available as SNMP triggers to alert SNMP client applications. In the evaluated configuration of CA Directory, the audit records are stored as flat files and are separated into: an alarm log, warn log (errors and warnings), statistics log, and an update log which informs a Directory Administrator of who is performing updates to the Directory at a given time. Refer to Section 7.1.1 for additional log files under the application notes. Audit logs are rolled over each day and only restricted by the amount of disk space still available on the machine. To view audit information, a user must have read access to the audit file location. This access is controlled by the underlying OS.

2.5.2 User Data Protection

The TOE uses the X.501 access control scheme to control access to its repository data for users accessing the Directory using DAP and LDAP. These users are the relying parties (AC Agents) using a Directory-enabled interface. DAP and LDAP are the only interfaces for these users. Access controls are configured externally to the Directory data, therefore normal access control rules are not determined by any security attributes. The individual access control rules specify what level of access to what subtrees, entries or attributes a user DN is allowed.

The access control policy is a single set of access controls that are static in nature. The access controls are defined and stored in configuration files, and are copied into the TOE memory during startup. This memory copy is used during operation. Once copied into memory and the TOE is initialized, these configuration files are not referenced and moved out of scope.

2.5.3 Identification & Authentication

CA Directory provides the following levels of authentication on the TOE: Client to DSA authentication (via DAP or LDAP), and DSA to DSA authentication (via DSP or DISP). When a client binds to the Directory, the client initially chooses an authentication type. If that authentication type is permitted by the DSA and the authentication requirements are met by the client, then the authentication is permitted. The three forms of authentication allowed are anonymous, clear password, and SSL authentication. For the evaluated configuration, only clear password will be enabled. Anonymous authentication is enabled but it is the responsibility of a Directory Administrator to ensure that the information presented to an Anonymous User is not security relevant. SSL authentication is not included in the evaluation of the TOE.

The TOE provides DAP and LDAP users with two authentication mechanisms: password-based and distributed authentication for users in a distributed Directory environment. The remote trusted peer DSAs that access the TOE using DSP and DISP are required to authenticate using the mutual authentication when establishing a connection to the DSP and DISP sessions. The DXserver validates the identity provided by the client for the mutual authentication and returns its identity to create a mutual, trusted session between the two servers. The DXconsole users are authenticated by the TOE using a password mechanism. A TOE configuration file specifies which users are allowed access to the DXconsole based on their roles and then those users are authenticated using the same password mechanism as DAP users.

With the evaluated password policy, there is policy audit setting that is set to restrict the number of authentication attempts before an account is suspended. When an account is suspended for reaching the limit of failed attempts that unique DN is then locked out for a configured length of time before being allowed to attempt his or her login again.

When logging in to the TOE, a user can be assigned a specific role appropriate to their Distinguished Name (DN) and password. The four roles provided are SuperUser, Administrative User, Registered User, and Public User. SuperUsers have unrestricted read/update access to all parts of the DSA, Administrators have read/update privileges within their specified Administrator scope (a subtree, entry, or designated attributes in an entry or subtree), and Registered Users have read access within the same style of scope. Public users have read access only to those entries that are non-TSF relevant.

It should be noted that a user may have different access control rights over different parts of the DIT. For example, a user may have Administrative rights over one subtree but only Registered User rights over another.

2.5.4 Security Management

The TOE, through the DXconsole, provides the TOE's Directory Administrator access to some of the security functionality of the TOE. Access Controls cannot be changed via the DXconsole as any changes made will be reset to the configuration file settings at restart/startup. While all the security functions and data can be accessed from the DXconsole, some of the trusted data resides in configuration text files on the DXserver and some in the repository. The data in the configuration files requires a Directory Administrator to modify the files using a text editor on the operating system for the modifications to be persistent when the DXserver restarts. In addition, Directory Administrator-specified remote trusted peer DSAs are able to update defined portions of the repository data through replication.

Supporting the password-based authentication mechanism, a Directory Administrator can specify a policy for passwords that includes authentication failure mechanisms and rules that define acceptable passwords.

The console interface is used solely by the Directory Administrator. It can be password protected and can be limited to being available only on the machine running the DSA. The console can be used to change some DSA settings, but not access controls, and to access the Directory itself. Directory access requires a bind as per the DAP/LDAP interface and access control rules are applied as per the DAP/LDAP interface. A Directory Administrator can also view audit information in text format and modify some policies through the CLI.

2.5.5 High Availability (FRU + FPT)

The TOE protects the TSF by providing high availability support and confidential data transmission. As discussed in Table 2-2, Evaluated Components of the Operational Environment, the concept of replication is discussed. The TOE leverages this capability (if configured to do so) by keeping an up-to-date duplication of the DXserver to provide a functional Data Store if the original DSA fails. In the case that the original DSA fails during a user operation, the TOE will perform a failover based on the configuration data in memory and transfer the request to the Replicated DSA. The TOE will check to see that the first DSA is unreachable and then checks to identify the name of the backup DSA. If located based on configuration, the TOE will switch and process the operation successfully as long as the user is authorized to do so.

The TOE is capable of ensuring the operation of all security operations (query, modify, or delete TSF data) by providing a Replicated DSA that the TOE can switch to when a failure of the original DSA occurs.

Failover is usually performed by a DXserver. The DXserver will have knowledge of two or more data DSAs and will switch between them.

3 Conformance Claims

3.1 CC Version

This ST is compliant with Common Criteria for Information Technology Security Evaluation, CCMB-2007-09-004, Version 3.1 Revision 3 July 2009.

3.2 CC Part 2 Conformance Claims

This ST and Target of Evaluation (TOE) is Part 2 conformant for EAL3, to include all applicable NIAP and International interpretations through 30 November 2010.

3.3 CC Part 3 Conformance Claims

This ST and TOE is Part 3 augmented with ALC_FLR.1. This includes all applicable NIAP and International interpretations through 30 November 2010.

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 EAL3 package claims augmented with ALC_FLR.1.

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 A legitimate user of the TOE could gain unauthorized access to resources or information protected by the TOE, or perform operations for which no access rights have been granted, via user error, system error, or other actions.

T.ADMIN_ERROR A Directory 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.MASK Users, whether they are malicious or non-malicious, could gain unauthorized access to the TOE by bypassing identification and authentication measures.

T.MASQUERADE A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.

4.2 Assumptions

This section contains assumptions regarding the security environment and the intended usage of the TOE.

4.2.1 Personnel Assumptions

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

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

4.2.2 Physical Assumptions

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

5 Security Objectives

This section identifies the security objectives of the TOE and its supporting environment. The security objectives identify the responsibilities of the TOE and its environment in meeting the security needs.

5.1 Security Objectives for the TOE:

The following are the TOE security objectives:

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 administrators.

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

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

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 Directory Administrators with the necessary information for secure delivery and management.

5.2 Security Objectives for the Operational Environment

The TOE’s operating environment must satisfy the following objectives.

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

OE.NOEVIL: Administrative Users 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 Directory/System Administrators and will authenticate the claimed identity prior to granting them access to the TOE configuration files stored on the underlying OS.

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

OE.FILESYS The Security features offered by the underlying Operating System protect the audit files used by the TOE by requiring authentication to the OS before reviewing audit files.

6 Extended Security Functional and Assurance Requirements

6.1 Extended Security Functional Requirements for the TOE

There are no extended Security Functional Requirements for this ST.

6.2 Extended Security Assurance Requirements

There are no extended Security Assurance Requirements in this ST.

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
Security Audit (FAU)	FAU_GEN.1 (Audit Generation)
	FAU_SEL.1(1) (Selective Audit)
	FAU_SEL.1(2) (Selective Audit)
User Data Protection (FDP)	FDP_ACC.1 (Subset Access Control)
	FDP_ACF.1 (Security Attribute Based Access Control)
Identification and Authentication (FIA)	FIA_AFL.1 (Authentication Failure Handling)
	FIA_ATD.1 (User Attribute Definition)
	FIA_SOS.1 (Verification of Secrets)
	FIA_UAU.1 (Timing of Authentication)
	FIA_UAU.5 (Multiple Authentication Mechanisms)
	FIA_UID.1 (Timing of Identification)
Security Management (FMT)	FMT_MSA.1 (Management of Security Attributes)
	FMT_MSA.3 (Static attribute initialization)
	FMT_MTD.1 (Management of TSF Data)
	FMT_SMF.1 (Specification of Management Functions)
	FMT_SMR.1 (Security Roles)
Protection of TSF Data (FPT)	FPT_FLS.1 (Failure with preservation of secure state)
Resource Utilization (FRU)	FRU_FLT.1 (Degraded fault tolerance)

Table 7-1: Security Functional Requirements for the TOE

Note: High Availability is a grouping of the requirements FRU_FLT.1 and FPT_FLS.1.

7.1.1 Class FAU: Security Audit

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) ***[the auditable events listed in the table below].***

Application Note: *The auditable events for the non-specific level of auditing are included in the table below.*

Component	Event	Additional Details
FAU_GEN.1	Start-up and shutdown of audit functions	N/A
FAU_SEL.1(1)	Modifications to the audit configuration regarding the event types that have been selected for auditing.	None
FAU_SEL.1(2)	Modifications to the audit configuration regarding the log types that have been selected for auditing.	None
FDP_ACC.1	None	N/A
FDP_ACF.1	Enforcement of the user policy	The identity of the user that caused the event.
FIA_AFL.1	Incorrect password entered for authentication and attempts to log in through DXconsole when a session is already currently established.	N/A
FIA_ATD.1	None	N/A
FIA_SOS.1	None	N/A
FIA_UAU.1	Unsuccessful use of the password based authentication mechanism.	The identity of the user that caused the event.
FIA_UAU.5	The final decision on authentication	The identity of the user that caused the event. Must exclude all password information in the audit record.
FIA_UID.1	Unsuccessful use of the password based identification mechanism.	The identity of the user that caused the event.
FMT_MSA.1	Modifications of access control rules	None
FMT_MTD.1	1. Operations on the TSF data located in the repository. 2. Operations performed on the operating memory from the console.	None
FMT_SMF.1	None	N/A
FMT_SMR.1	None	N/A

Table 7-2: Auditable Events

Application Note: *The TOE defines the startup and shutdown of the audit function as the same as the startup and shutdown of the DXserver.*

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) b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, ***[information specified in Column Three of the table above]***.

FAU_SEL.1(1) Selective audit

Hierarchical to: No other components.
Dependencies: FAU_GEN.1 Audit data generation
FMT_MTD.1 Management of TSF data

FAU_SEL.1(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) ***[event types***

b) ***No additional attributes]***.

Application Note: *The TOE defines the following information that is captures for the event types: alert, cert, connect, diag, DSA/all, error, LDAP, Limit, Query, Stack, stats, summary, time, update, warn, X.500, and ASN.*

FAU_SEL.1(2) Selective audit

Hierarchical to: No other components.
Dependencies: FAU_GEN.1 Audit data generation
FMT_MTD.1 Management of TSF data

FAU_SEL.1(2).1 The TSF shall be able to select the set of events to be audited from the set of all auditable events based on the following attributes:

a) ***[type of daily log to record***

b) ***No additional attributes]***.

Application Note: *Daily logs include the following audit files: Summary, Trace, Stats, Query, Connect (only successful authentication, IP/address, form of authentication), Update, Cert, Alert, Warn, Diag, Time, and Alarm.*

7.1.2 Class FDP: User Data Protection

FDP_ACC.1 Subset Access Control

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1 The TSF shall enforce the *[**user policy**]* on ***[***

- ***Subject: LDAP and DAP Sessions, DSAs***
- ***Objects: DSP protocol, the repository information, both information entries and information attribute types; and***
- ***Operations: add, search, list, modify, mod-rdn, read, compare, delete].***

FDP_ACF.1 Security Attribute Based Access Control

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1 The TSF shall enforce the [***user policy***] to objects based on the following: ***[***

- ***Subject Security Attributes:***
  - ***Distinguished Name,***
  - ***User Groups,***
  - ***User Roles,***
  - ***Authentication Level;***
- ***Object Security Attributes:***
  - ***Access Control rule(s) each specifying the following:***
    - ***Objects for which the access control rule applies***
    - ***Subjects for which the access control rule applies***
    - ***Priority of the access control rule permissions: SuperUser, Administrative User, Registered User, Public User***
    - ***Optional permissions: read, add, remove, rename, all, and modify.***
    - ***Authentication level required].***

Application Note: *Authentication level refers to how the subject authenticated to the Directory: anonymous, clear password, or ssl* *authentication. In the evaluated configuration, only clear password is leveraged by the TOE to perform authentication.*

Application Note: *'Permissions' translates to both X.501 permissions and to X.501 precedence in the access control decision function specified at the end of FDP_ACF.1.2. It is not possible to change precedence of the rule, as defined in X.501.*

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

- ***The access control rules must include rules where:***
    - ***the subject requestor (distinguished name, user group, role) is in the access control rule subject set;***
    - ***the protected object of the operation is in the access control rule object set;***
    - ***the subject requestor is authenticated at the required level;***
- ***The access control decision must apply the following rules to the 'associated access control rules':***
    - ***only access control rules with the highest priority are considered;***
    - ***grant access only if all access control decisions protecting that object are approved.]***

Application Note: *The policy implements the X.501 Simplified Access Control requirements.*

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [***no additional rules***].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the [***rule that protected items are never available to Registered Users***].

Application Note: *By default, no access control rules are defined, denying access to all subjects as long as access control is enabled with no configured roles or privileges.*

7.1.3 Class FIA: Identification and Authentication

FIA_AFL.1 Authentication failures

Hierarchical to: No other components
Dependencies: FIA_UAU.1 Timing of Authentication

FIA_AFL.1.1 The TSF shall detect when ***[a Directory Administrator configurable positive integer]*** unsuccessful authentication attempts occur related to ***[unsuccessful LDAP/DAP client binds].***

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall ***[disable the account for the Directory Administrator-specified period of time]***.

Application Note: *The user is required to use a value that supports the security policy as specified in the Administrator Guide Supplement.*

FIA_ATD.1 User attribute definition

Hierarchical to: No other components
Dependencies: No dependencies

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

a) ***Distinguished Name;***

b) ***Role;***

c) ***Group;***

d) ***Password;***

e) ***User status].***

FIA_SOS.1 Verification of Secrets

Hierarchical to: No other components.
Dependencies: No dependencies.

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

Application Note: *The configured password policy will state that the minimum length of a user's password contains at least 6 characters as well as the following requirements: at least one capital* *letter, one number, and one special character, cannot repeat a character more than 2 times, cannot match the user's previous 3 passwords, cannot contain the user's own name or DN, and cannot be older than 90 days.*

FIA_UAU.1 Time of Authentication

Hierarchical to: No other components.
Dependencies: No dependencies.

FIA_UAU.1.1 The TSF shall allow [***Read access to public repository information in accordance with the user policy***] on behalf of the users to be performed before the user is authenticated.

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

FIA_UAU.5 Multiple Authentication Mechanisms

Hierarchical to: No other components.
Dependencies: No dependencies.

FIA_UAU.5.1 The TSF shall provide ***[***

a) Clear password,

b) Password (console)

c) Distributed authentication via 'Peer DSA Password Check',

d) and Distributed authentication via 'conveyed originator']

to support user authentication.

Application Note: *Distributed authentication is used when the TOE works as part of a distributed Directory system. The credential information required to make an authentication decision is in a different Directory server than the one that holds the information the user wants to access. A user can authenticate to the TOE with their password credentials stored on a Trusted Peer DSA using 'Peer DSA Password Check'.*

FIA_UID.1 Timing of identification

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FIA_UID.1.1 The TSF shall allow ***[Read access to public repository information in accordance with the user policy]*** on behalf of the user to be performed before the user is identified.

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

Application Note: *Users do not need to provide an identifier when viewing information in the public repository. The information is read only and cannot be modified without authenticating to the TOE.*

7.1.4 Class FMT: Security Management

FMT_MSA.1 Management of Security Attributes

Hierarchical to:	No other components.
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

FMT_MSA.1.1 The TSF shall enforce the ***[user policy]*** to restrict the ability to ***[query, modify, and delete]*** the security attributes ***[access control rules]*** to ***[Directory Administrators via the configuration files].***

Application Note: It's important to note that changes can be made at the DXconsole but only take effect until the TOE is restarted. The only method to permanently affect the user policy is for a Directory Administrator to edit the configuration files.

FMT_MSA.3 Static Attribute Initialization

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.3.1 The TSF shall enforce the ***[user policy]*** to provide ***[restrictive]*** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the ***[Administrative Users with appropriate scope or SuperUser]*** to specify alternative initial values to override the default values when an object or information is created.

Application Note: Administrative Users and SuperUsers define the variables for entries into the Directory when created. There are no default values. All initial values are specified by these roles.

FMT_MTD.1 Management of TSF Data

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security Roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1 The TSF shall restrict the ability to ***[perform]*** the ***[operations against TSF data as determined in Table 7-4 below]*** to ***[the roles listed in Table 7-4 below].***

Operations	TSF data	Roles
	Repository Data	

Operations	TSF data	Roles
Query **Modify** **Delete**	• (FMT_MTD.1) Managing the group of roles that can interact with the security attributes. • (FMT_SMR.1) Managing the group of roles except Trusted Peer DSA that can interact with the TSF data. • (FIA_UID.1) DN for users for Directory Administrator-specified portions of the repository.	• **Directory Administrator-specified SuperUser** • **Directory Administrator-specified Trusted Peer DSA** • **Directory Administrator**
Modify	• (FIA_UAU.1) Passwords	• **Directory Administrator** • **Directory Administrator-specified SuperUser** • **Registered User for the password when specified by the Administrator**
Query	• Directory Data	• **Administrative User via LDAP/DAP client** • **Directory Administrator** • **Registered Users with scope to subtree** • **Directory Administrator-specified SuperUser**
Configuration File Data in TOE Memory		
Query **Modify**	• (FAU_SEL.1) the TSF data that determines which events are being audited. • (FIA_AFL.1) management for the threshold for unsuccessful authentication attempts and action to be taken in the event of an authentication failure. • (FIA_SOS.1) metrics used to verify secrets	• **Directory Administrator**

Table 7-3: Management of TSF Data

FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.
Dependencies: No dependencies.

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: ***[as specified in FMT_MSA.1 and Table 7-4 above].***

FMT_SMR.1 Security Roles

Hierarchical to: No other components.
Dependencies: FIA_UID.1 Timing of Identification.

FMT_SMR.1.1 The TSF shall maintain the roles ***[***

- ***SuperUser;***
- ***Administrative User;***
- ***Registered User;***
- ***Trusted Peer DSA].***

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

7.1.5 Class FPT: Protection of the TSF

FPT_FLS.1 Failure with Preservation of Secure State

Hierarchical to: No other components.
Dependencies: No dependencies.

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

Application Note: *The TOE preserves a secure state by providing a replication of the DSA. When the primary server fails, the infrastructure fails over to the Replicated DSA.*

7.1.6 Class FRU: Resource Utilization

FRU_FLT.1 Degraded Fault Tolerance

Hierarchical to: No other components.
Dependencies: FPT_FLS.1 Failure with preservation of secure state

FRU_FLT.1.1 The TSF shall ensure the operation of ***[all security functions]*** when the following failures occur: ***[failure of a DSA].***

.

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.

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 Security-enforcing functional specification (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 security enforcing effects 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 TOE 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 behavior described in the TOE design is mapped to the TSFIs that invoke it.

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 Use of a CM system (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. Parts of the TOE

8.3.2 CM coverage (ALC_CMS.2)

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

8.3.3 Delivery Procedures (ALC_DEL.1)

ALC_DEL.1.1D The developer shall document 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 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 Flaw reporting procedures (ALC_FLR.1)

ALC_FLR.1.1D The developer shall document flaw remediation procedures addressed to TOE developers.

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.

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 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 Derived 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 requirement’s 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.1)

ASE_TSS.1.1D The developer shall provide a TOE summary specification.

ASE_TSS.1.1C The TOE summary specification shall describe how the TOE meets each SFR.

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

ASE_TSS.1.2E The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

8.5 Tests

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.

8.5.1 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.

8.5.2 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.

8.5.3 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 TOE Security Functions

The following sections identify the security functions of the TOE. They include Security Audit, User Data Protection, Identification and Authentication, Security Management, and High Availability.

9.2 Security Audit

The TOE generates audit records for TSF-relevant events in its audit files which are located on the operating system supporting the DXserver. The audit mechanism is always active while the DXserver is running, therefore starting and stopping of the audit function is done when the DXserver is started and stopped. This action is recorded in the alarm log. Any action relevant to the TSFs is captured in the audit files on the OS, such as LDAP binds, LDAP authentication, searches, and modifications.

The Directory Administrator selects the auditable events to include and exclude by opening and closing the associated audit file. The types of audit events are organized by the event type of the audit file as well as the level of the trace as determined by the configuration in memory. Any user who has access to the audit file folder location can view audit files. This is an Operating System level permission management issue. Additionally, the identity of the user that caused the event is included in the audit records except for updates to the operating memory from the local console.

9.2.1 Event Types

The audit trace file is responsible for collecting all information relating to each event on the TOE. When an operation is being performed, the trace file, if configured to "all," will record every step the DXserver and its modules perform in the process of executing the task. This includes authentication processing, LDAP protocols, errors, and updates to the DSA. The trace levels include: none (not to be used in the evaluated configuration), warn, stats, diag, alert, and those listed below in Table 9-1.

Trace Level	Content
Alert	Displays authentication errors
Cert	Displays certificate operations
Connect	Displays connections
Diag	Traces local DXserver operations that were refused
DSA/All	Similar to the x500 trace, but also includes tracing of the module flow inside the DSA. This is a full code tracing for each operation.
Error	Displays error messages of high severity. Compare with trace warn. These are events that may have an impact on the ability of DXserver to perform a requested operation. This is the default trace level.
Ldap	Traces detailed LDAP operations. The output can become quite large when searches return a large number of entries.

Limit	Traces any violation of size or time limits
Query	Displays a one-line summary containing the server request and result.
Stack	Displays detailed protocol tracing. The output can become quite large
Stats	Displays statistical information for each minute the DSA is not idle
Summary	Displays a one-line summary containing the service request and result
Time	Displays the time taken for successful operations
Update	Displays update operations-add, delete, modify, and rename
Warn	Displays error messages of moderate severity. Compare with trace error. Warn messages usually represent a user error, rather than a problem with DXserver
X.500	Displays the full details of the service request, confirmation, or error. This traces DAP, DSP, and LDAP operations. The output can become quite large when searches return a large number of entries.
ASN	ASN protocol encoding

Table 9-1: Trace Levels (Event Types)

9.2.2 Log Types

CA Directory R12 allows for the configuration of audit files based on event types. A Directory Administrator with access to the underlying OS can configure the auditable events on the TOE based on which audit files are enabled. Listed below in the table is the audit files and what is collected by each.

Audit file	Content
Summary	Summary of all operations written. The summary log displays a summary (no real detail) of each operation that is processed. Depending upon what the operation is, there is a fixed set of fields summarizing that operation. A summary log is useful to give more details about operations than the stats log. e.g. gauge the ratio of updates to searches or gauge the ratio of simple searches to complex searches.
Trace	Current level of tracing written. Further information regarding the Trace log can be found in Section 9.2.2. This log does not rollover daily and is configured by default to be running.
Stats	Each stats entry is the sum of all the activity for the past 60 seconds. Entries are only written when the DSA is active.
Query	The query log records all operations processed by the DSA. Each operation has a result written to indicate the success or failure of the operation. Each operation is paired with the DSA's response. A query log is useful for finding rogue queries (ones that take a long time because they are very complex or return huge amounts of data)
Connect	This log contains a list of all successful connections to the CA Directory DSA.
Update	The update log generates one line for each UPDATE operation performed. This includes ADD, MODIFY, MODDN (Rename) or Delete (REMOVE) operations.
Cert	The cert log captures all operations that pertain to certificate objects within the Directory.
Alert	The alert log captures all security related exceptions generated by the DSA.
Warn	The warn log captures all warnings generated by the DSA. The warn message itself is free form in nature, dependent on the warning generated, but the first few fields are fixed. A warn log is useful as an early warning

	mechanism.
Diag	The diag log records all protocol or client errors (e.g. bad DN, credentials, filter)
Time	If an operation takes longer to process than the time log limits specify, operation details are written to this log. This can be used to determine what operations are taking a long time to process.
Alarm	The alarm log captures all alarms generated by the DSA. DSA alarms are defined as any message that condition which threatens or impacts the stability of the DSA itself. The alarm log should always be monitored. This log does not rollover daily and cannot be disabled. If the DSA if running, the alarm log is active.

Table 9-2: Audit files (Log Types)

All audit files are stored in flat file format and rolled over every 24 hours (except for the Alarm and Trace Log which are done at the discretion of the System/Directory Administrator) with time stamps applied to the file and events within each audit. Timestamps on audit records are obtained from the operating system where the product is running.

The only restriction is the amount of disk space available on the hard drive where the logs are being stored. The summary log shows association numbers and the binds recorded in this log show the Distinguished name of the user. The identity of the remote trusted peer for the DSP bind is included in the trace-log.

9.3 User Data Protection

Access controls are configured externally to the Directory data, therefore normal access control rules are not determined by any security attributes. The individual access control rules specify what level of access to what subtrees, entries or attributes a user DN is allowed.

CA Directory provides access control rules that are easy to work with and maps these rules to the X.501 access control standards. The access control rules work by answering the question of "Is the client permitted to perform an operation on a certain subtree of data?"

- Client: A user (based on distinguished name), a role as defined by RBAC (role based access control), a user group, or authentication level (clear password)
- Operation: Add, Search, List, Modify, Modify-dn, Delete
- Data: Any data stored by the TOE in its data stores which are part of the DSA
- Subtree: A subsection of the data store

Each Directory has one access control policy. This policy is the collection of rules that define who can access what, and under what circumstances. It is recommended that the same access control rules are applied to every DSA. This does not mean that only one rule is possible but rather that several access control policies cannot apply to various subtrees of the data stored on the DSA data store. The access control rules for the TOE are stored on the underlying OS where the DXserver has been installed. If a user of the TOE wishes to change an access control rule, they must have Directory/System Administrator access to the underlying OS and once updated, restart the TOE to apply the updated rules. Additional basic information can be found in Chapter 18 of the Administrative Guide for CA Directory R12.

9.3.1 Access Levels

By default, no access control rules are defined, denying access to all subjects as long as access control is enabled with no configured roles or privileges. If the TOE is configured to use access control rules, the order of access is hierarchical. The DSA, when receiving access requests for operations, will check for the highest applicable rule that apply to that user or trusted peer DSA based on the following levels in order of highest to lowest precedence:

1. SuperUser – Access rights at this level cannot be taken away.
2. Administrative User – Access rights at this level cannot be taken away.
3. Protected Items – Rules at this level deny access rights given to lower precedence levels.
4. Registered User – Access rights are granted but can be taken away.
5. Public User – Grant read rights, but can be taken away.

Access rights granted to these roles also operate in a hierarchical manner. If an Administrative User access level is given the modify permission only, this implicitly grants the Administrative User read access. If only given read access, a user with that access level could only perform read operations of data. In the evaluated configuration, Public Users only have read access to the Directory. If an item is protected, any specified permission means that operation is denied (e.g. if all operations are specified, all operations are denied). This does not operate hierarchically though. If modify is only specified for a protected item, all other operations are still available.

Finally, it should be noted that privileges are scoped, although the scope may be the entire Directory Data Store. Scoping can be by subtree, entry or attribute or by a combination of these.

9.3.2 Access by Administrative User or SuperUser

A user granted SuperUser access has their permissions set to allow all. This means that the specified user can modify and read all information within the entire Directory. Administrative user access gives a user read and update access rights over specified scope. This scope could be a subtree, entry, a particular set of attributes within an entry or subtree. If the latter is done, the Administrative User cannot add or remove entries but only modify the attributes they have access to (e.g. user passwords). These two access levels are not affected by the protected items access level.

9.3.3 Role Based Access Control

CA Directory access controls also include Role Based Access Controls (RBAC). A role can be viewed as being a "security attribute" for the user binding to the Directory. If role based access controls are enabled, and a user is a member of a specific role, then that user automatically inherits all of the access permitted by the role based access controls loaded by the DSA. An overview of the role based access controls can be found below.

When access control rules are applied to a role, they are known as role-based access controls. A role is a Directory entry that can be associated with user entries. These associated user entries are member entries. A role may contain any number of members. This can be useful in a large distributed Directory environment where people change their roles frequently, and when the Directory is to be used as an authorization engine. Roles are maintained in a subtree of the Directory information tree (DIT).

RBAC operates in the following manner:

1. A user attempts to bind to the DSA
2. The DSA searches the role subtree for the attempting user's DN in the member attribute
3. The DSA stores the names returned by the search as the roles pertaining to the connection, and uses them in access control decisions

Access controls are not active by default when a DSA is initially created. This is by design and required in the evaluated configuration. Without binding/connecting to the Directory, there are no operations that can be performed on the data contained in the DSA. A successful bind/connection needs to be established to a DSA in order to operate on the data contained in the DSA. If access controls are in place, only operations allowed by the enforced access controls will be permitted.

9.4 Identification and Authentication

When an LDAP/DAP client binds to the Directory, the LDAP client initially chooses an authentication type. If that authentication type is permitted by the DSA and the authentication requirements are met by the LDAP client, then the authentication is permitted.

9.4.1 Authentication Levels (LDAP client to DSA)

When a user is attempting to authenticate to the TOE through an LDAP client, they are subject to three possible authentication levels within the evaluated configuration. The authentication level is detailed below.

Clear password: This is configured in the DSA's settings file using the command "set min-auth = clear password". The value of "clear password" requires a password to be provided. For clear password simple authentication, the users connecting to the TOE must have corresponding user entries stored on the TOE with a password attribute. Authentication will fail if the entry named cannot be found, the user entry does not contain a password attribute, or the password provided does not match the stored password.

Clear password binds will also be accepted when the min-auth = none as min-auth specifies the minimum authentication level that is accepted by the DSA.

Note: This is different from the auth-level setting in the knowledge files which is used for DSA-DSA authentication. The auth-level setting only allows binds of the specified values.

9.4.2 Authentication (DSA to DSA)

The types of authentication levels supported in a DSA to DSA are controlled by the "auth-levels" specified in a DSA's knowledge file. When a DSA binds to another DSA it will attempt to use the same authentication level as the LDAP Client. Similar to client to DSA binding, the following level is implemented on a DSA to DSA basis:

Clear password: This is where a DSA to DSA mutual authentication (bind) will be performed based upon the originators "DSA-NAME" & "DSA-PASSWORD". When DSA1 binds to DSA2, using a clear password authentication level, it sends a bind request to DSA2 containing it's dsa-name and dsa-password. DSA2 compares the dsa-name and dsa-password passed within the bind request and compares it to the registered dsa-name and dsa-password it's cached for DSA1.

When a DSA attempts to bind to another DSA, it occurs through mutual authentication. This means that both DSAs have to be aware of each other and contain credentials to confirm that they are trusting of one another as well. The process of this authentication is explained below:

1. The sending DSA includes its credentials within the bind request which are collected from its knowledge file.
2. The receiving DSA checks the credentials as well as the DSA name, IP address, and password of the sending DSA against its own knowledge file of the sending DSA.
3. The receiving DSA then sends its own credentials back to the sending DSA to confirm that it is the DSA the sender actually wants to communicate with. The sending DSA will also check the IP address of the bind confirm sent by the receiving DSA with its knowledge file.
4. If the credentials match, the bind is successful. Otherwise, it is denied and the authentication failure is logged.

9.4.3 Password Policy

The TOE implements a password policy to ensure strength of secrets. By requiring certain minimum password structure, the risk of an attacker is greatly mitigated. The password structure, as defined in the password policy, requires the following items to be configured:

- At least 6 characters long
- At least one capital letter
- At least one number
- At least one special character
- Cannot repeat a character more than 2 times
- Cannot match the user's previous 3 passwords
- Cannot contain the user's own name or DN
- Cannot be older than 90 days.

If a password does not fit within the above defined policy, it is rejected and the user must attempt to configure their password again.

When the password policy is configured (based on the SFR information and supplemental guidance), there is a password policy audit setting that can be set to restrict the number of authentication attempts before an account is suspended. When an account is suspended for failing to successfully authenticate after a consecutive number of attempts, the account is locked out from anymore attempts. After a configured length of time, the account can attempt to log in again. The length of suspension is based on the "password-max-suspension" variable. This only applies to accounts that have been suspended due to "max-retries" being exceeded. These values are not configured by default but guidance is provided on how to configure the password policy properly.

Finally, when the password policy is active, a user account can be active, expired, suspended, or locked. Active means that the user is capable of logging in regularly. Expired means that the user can no longer log in because they have not changed their expired password. Suspended applies to a user who has failed to authenticate too many times in a row and locked applies to any account that has been manually locked by an Administrative User for any number of reasons (e.g. length of password, special characters missing, user is on vacation).

9.5 Security Management

The TOE, through the DXconsole, provides the TOE's Directory Administrator access to control the security functions and manage the trusted data. While all the security functions and data can be accessed from the DXconsole, some of the trusted data resides in configuration text files on the DXserver and some in the Data Store. The data in the configuration files requires a Directory Administrator to modify the files using a text editor on the operating system for the modifications to be persistent when the DXserver restarts. In addition, Directory Administrator-specified remote trusted peer DSAs are able to update defined portions of the repository data through replication.

Once authenticated to the TOE, a user can perform operations on the Directory based on the access controls in place and the access rights of the user. This can be determined by their scope of the Directory (subtree, entry, attribute), their level of access (SuperUser, Administrative User, Registered User, Public User, protected items), or the access controls in place (static, RBAC). If successfully authenticated, a user can perform queries, modifications, and deletions of data on the TOE as long as it is within their scope. Additionally, Administrative Users of the underlying OS can configure the configuration (knowledge) files and view both those and the audit files for review. Audit review is done by the Operational Environment and therefore not a responsibility of the TOE.

9.5.1 Levels of Access

By engaging CA Directory access controls, a DSA has the following levels of access in the product:

- SuperUser: SuperUsers have unrestricted read and update access to all parts of the DSA. Add users to the list of SuperUsers either as single users, groups of users, roles, or all users in a specified subtree.
- Administrative User: This requires a clear password credentialed authentication bind. Within a specified subtree, you can assign Administrative User authority to users. Administrative users have read and update privileges over a specified administrative scope. This scope is a subtree, entry, or designated attributes in an entry or subtree. The Administrative User can view and update protected items within the administrative scope. Add users to the list of administrative users either as single users, groups of users, roles, or all users in a specified subtree.
- Protected Items: Protected items are those which have been configured by a Directory Administrator to disallow viewing by Registered and Public Users of the data marked as protected. It should be noted that protected items are not an access level for users but rather a classification applied to TSF data that is deemed viewable only by Administrative Users and SuperUsers.
- Regist ered User: This requires a clear password credentialed authentication bind. Registered Users have read privileges over a specified scope: a subtree, entry, or the designated attributes in an entry or subtree. “Protected items” in the scope are invisible.

Trusted Peer DSAs also authenticate to the TOE through mutual authentication. The access controls which apply to the DSA do not fall under the scope of any one specific access level. See Section 9.4.2 for more information regarding the DSA to DSA authentication.

The console interface is used solely by the Directory Administrator. It is password protected and limited to being accessible by only one user at a time. The console can be used to change some DSA settings, but not access controls, and to access the Directory itself. Directory access requires a bind as per the DAP/LDAP interface and access control rules are applied as per the DAP/LDAP interface.

The DSP interface is used for DSA-DSA communication. Operations performed via DSP are subject to DSP binds being performed and user credentials being passed with the operation. This means that access control rules are applied by the DSA receiving the request.

9.6 High Availability (FRU+FPT)

The TOE is capable of ensuring the operation of all security operations (query, modify, or delete TSF data) by providing a Replicated DSA that the TOE can switch to when a failure of the original DSA occurs.

In addition to the use of Replicated DSAs to provide full operation of TSF resources, the TOE can generate online backups of the Data Store. To ensure that in case of any failure of a Data Store, an authorized Administrative User can recover the data from an online location and load the data back into the TOE through the DXtools component.

10 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
Security Audit (FAU)	FAU_GEN.1 (Audit Generation)
	FAU_SEL.1(1) (Selective Audit)
	FAU_SEL.1(2) (Selective Audit)
User Data Protection (FDP)	FDP_ACC.1 (Subset Access Control)
	FDP_ACF.1 (Security Attribute Based Access Control)
Identification and Authentication (FIA)	FIA_AFL.1 (Authentication Failure Handling)
	FIA_ATD.1 (User Attribute Definition)
	FIA_SOS.1 (Verification of Secrets)
	FIA_UAU.1 (Timing of Authentication)
	FIA_UAU.5 (Multiple Authentication Mechanisms)
	FIA_UID.1 (Timing of Identification)
Security Management (FMT)	FMT_MSA.1 (Management of Security Attributes)
	FMT_MSA.3 (Static attribute initialization)
	FMT_MTD.1 (Management of TSF Data)
	FMT_SMF.1 (Specification of Management Functions)
	FMT_SMR.1 (Security Roles)
Protection of TSF Data (FPT)	FPT_FLS.1 (Failure with preservation of secure state)
Resource Utilization (FRU)	FRU_FLT.1 (Degraded fault tolerance)

Table 10-1: Security Functional Requirements for the TOE

Note: High Availability is a grouping of the requirements FRU_FLT.1 and FPT_FLS.1.

10.1 Security Audit

The security audit function of the TOE enforces the FAU_GEN.1, FAU_SEL.1(1), and FAU_SEL.1(2) requirements. FAU_GEN.1 requires a reliable time-stamp, which is provided by the Operational Environment via the underlying Operating System.

By default, DXserver creates the audit logs as flat files and stores them on the Operational Environment. For the same reason, the audit logs are created with read/write permissions granted only to users who have been granted access to the Operating System that the files are stored on. In the evaluated configuration, the audit files store the record of the startup and shutdown of the TOE's audit functions. The iterations of FAU_SEL.1

serve the purpose of identifying the two methods in which audit information can be collected. Audits can be collected based on trace levels and audit file configurations. For iteration one, the following event types can be set in the trace configuration: alert, cert, connect, diag, DSA/all, error, LDAP, limit, Query, Stack, stats, Summary, Time, Update, Warn, X.500, and ASN. The second iteration shows that a Directory Administrator can configure the audit files that are to be used in the collection of data. These files include the following: summary, trace, stats, query, connect, update, cert, alert, warn, diag, time, and alarm. Each audit file collects a certain type of information.

The minimum contents of each entry in the audit report include the following: Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event (see Table 7-2 Auditable Events for more information).

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

10.2 User Data Protection

The User Data Protection function of the TOE enforces the FDP_ACC.1and FDP_ACF.1 requirements.

The TOE uses the X.501 access control scheme to control access to its repository data for users accessing the Directory using DAP and LDAP. These users are the relying parties and Administrative Users using a Directory-enabled interface. DAP and LDAP are the only interfaces for these users. Access controls are configured externally to the Directory data, therefore normal access control rules are not determined by any security attributes. The individual access control rules specify what level of access to what subtrees, entries or attributes a user DN is allowed. CA Directory access controls also include Role Based Access Controls (RBAC). A role can be viewed as being a “security attribute” for the user binding to the Directory. This role, once applied to a user of the TOE, assists in determining whether the individual can read/write to data and specifically what subtrees of data that individual can even see.

10.3 Identification and Authentication

The identification and authentication function of the TOE enforces FIA_AFL.1, FIA_ATD.1, FIA_UAU.1, FIA_UAU.5, FIA_UID.1, and FIA_SOS.1.

The TOE provides authentication mechanisms to the TOE depending on the interface in which the communication is occurring. The DXtools that access the Directory using the LDAP interface and DAP/LDAP clients can use a Distinguished Name (DN) and for authentication and authorization. A user connecting through the DXconsole however only requires the secure password. If the user is viewing or modifying the DSA settings or if the Directory data is to be accessed via the DXconsole, a bind (similar to a DAP/LDAP bind) must be performed using the authentication methods above. Additionally, any interaction between the TOE and other DXservers and Trusted Peer DSAs goes through a process of mutual authentication so that both parties confirm one another.

The TOE also provides authentication between one DSA and another DSA. When an operation that requires distributed data to be accessed, the TOE or external DSA will authenticate to one another through mutual authentication. One DSA will send its credentials to the receiving DSA which will determine if it trusts the sender. The receiving DSA then sends its credentials back to allow the sending DSA to confirm its identity as well. Once this has occurred successfully, the operation can occur. For additional information, reference Section 9.4.2.

Administrative Users have to identify and authenticate themselves to the TOE before being able to remotely manage the TOE through DXserver. The TOE is capable of uniquely identifying a remote DSA or a user by their Distinguished Name (DN). Remote DSAs can also be identified by their IP address.

Authentication failures are handled by the TOE and are configured to lockout a user after a configured number of failed authentication attempts occurs. This lockout would automatically suspend the user for a period of time before attempting to let them attempt authorization again.

During authorization, CA Directory's DXserver communicates with the Data Store or Configuration data in memory to determine that the distinguished name (DN) and/or password is retrieved from a user who is requesting access to the TOE or its resources. When a user attempts to access an object, CA Directory uses the user's information along with any defined ACLs stored in memory to determine what the user can access and perform.

The rules and policies defined in the runtime memory specify the objects that the user is allowed to access.

10.4 Security Management

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

The TOE provides management capabilities through the LDAP client, DXtools, and DXconsole. The DXconsole interface can be used remotely through a telnet connection and allows an authorized user to perform searches and modifications to TSF data. The TSF shall provide the ability to manage its security functions including the management of user accounts and accessor access rights, TOE resources and security information recorded in the audit logs.

The TSF shall maintain the roles SuperUser, Administrative User, Registered User, and Trusted Peer DSA. An Administrative User and SuperUser have read/write access to certain subtrees or entries in the Data Store. An Administrative User with read/write access to all trees of the TOE is defined as a SuperUser.

10.5 High Availability (FRU + FPT)

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

The TSF shall ensure the operation of CA Directory to its resources when the DSA goes down by providing failover capabilities. This is accomplished by the period monitoring of the DSA through user operations. When one of the DSAs goes down, the Replicated DSA will take the place of the original DSA and take over user operations directed towards the original. The failed DSA receives attempts to restart every 180 seconds and will resume operation as the new Replicate to the now lead DSA. This ensures that there is no discontinuity of resource protection.

11 Security Problem Definition Rationale

11.1 Security Objectives 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 Directory Administrators will be assigned to install, configure and manage the TOE.	**OE. ADMIN** One or more authorized Directory Administrators will be assigned to install, configure and manage the TOE.	**OE. ADMIN** maps to **A. ADMIN** in order to ensure that authorized Directory Administrators install, manage and operate the TOE in a manner that maintains its security objectives.
A.PATCHES: Directory 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 Directory Administrators will be assigned to install, configure and manage the TOE.	**OE. ADMIN** maps **to A. PATCHES** in order to ensure that the authorized Administrative Users and Directory Administrators properly patch the TOE and the Operational environment in a manner that maintains its security objectives.
A.NOEVIL: Directory Administrators, Administrative Users, and SuperUsers of the TOE are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided by the guidance documentation.	**OE.NOEVIL** Administrative Users of the TOE are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided by the guidance documentation.	**OE.NOEVIL** maps to **A.NOEVIL** in order to ensure that there are no careless, willfully negligent, or hostile Administrative Users 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 11-1: Assumption to Objective Mapping

Threat	Objective	Rationale
T.ACCESS A legitimate user of the TOE could gain unauthorized access to resources or information protected by the TOE, or performs operations for which no access rights have been granted, via user error, system error, or other actions.	**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, FIA_AFL.1, FIA_UAU.1, FIA_UAU.5) 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.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 users and will authenticate the claimed identity prior to granting any access to the TOE.	**OE.AUTH** 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.
T.ADMIN_ERROR A Directory 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 Directory 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 Directory Administrators have guidance that instructs them how to administer the TOE in a secure manner and to provide the Directory Administrator with instructions to ensure the TOE was not corrupted during the delivery process. Having this guidance helps to reduce the mistakes that an Directory Administrator might make that could cause the TOE to be

Threat	Objective	Rationale
		configured in a way that is unsecure.
	O.MANAGE The TOE will provide authorized Directory Administrators and Administrative Users with the resources to manage and monitor user accounts, resources, and security information relative to the TOE.	**O.MANAGE** (FMT_MSA.1, FMT_MSA.3, FMT_MTD.1, FMT_SMF.1, and FMT_SMR.1) addresses T.ADMIN_ERROR by ensuring only authorized Directory Administrators and Administrative Users 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 recorded; thus masking a user's action.	**OE.SYSTIME** The operating environment will provide reliable system time.	**OE.SYSTIME** is necessary for the Audit logs to contain the accurate system time of events.
	OE.FILESYS The Security features offered by the underlying Operating System protect the audit files used by the TOE by requiring authentication to the OS before reviewing audit files.	**OE.FILESYS** addresses T.AUDIT_COMPROMISE by ensuring that the TOE provides the capability to protect the audit files used 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 users and will authenticate the claimed identity prior to granting any access to the TOE.	**OE.AUTH** helps to mitigate T.AUDIT_COMPROMISE by providing measures to uniquely identify and authenticate users through the OS authentication.
T.MASK Users, whether they are 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_SEL.1(1), and FAU_SEL.1(2)), 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 SuperUser in the	**O.IDENTIFY** (FIA_ATD.1, FIA_UID.1) addresses T.MASK by limiting the ability of users not fully authenticated by the TOE.

Threat	Objective	Rationale
	environment.	
	OE.SYSTIME The operating environment will provide reliable system time.	**OE.SYSTIME** is necessary for the Audit logs to contain the accurate system time of events.
	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 users and will authenticate the claimed identity prior to granting any access to the TOE.	**OE.AUTH** helps to mitigate T.AUDIT_COMPROMISE by providing measures to uniquely identify and authenticate users through the OS authentication.
T.MASQUERADE A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.	**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, FIA_AFL.1, FIA_UAU.1, and FIA_UAU.5) addresses T.ACCESS by providing the authorized users with the capability to specify access restrictions on the protected TOE resources to authenticated users.

Table 11-2: Threat to Objective Mapping

11.2 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

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 Security attribute based access control	FDP_ACF.1 states the TSF shall enforce the access control rules and policies to objects based on the Resource Record and Accessor Record.
	FIA_AFL.1 Authentication Failures	FIA_AFL.1 states that the TSF shall provide a configured policy that will deny users access if they have failed to properly authenticate to the TOE after a set number of attempts. They will then be locked out for a period of time before being allowed another attempt at authentication.
	FIA_UAU.1 Timing of Authentication	FIA_UAU.1 states that the TSF shall enforce the requirement that all users must be fully authenticated before performing TSF-mediated operations on the TOE.
	FIA_UAU.5 Multiple Authentication Mechanisms	FIA_UAU.5 states that the TOE will provide mechanisms for users to access the TOE to all for proper authentication by those users.
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 audits based on the configured audit files.
	FAU_SEL.1(1) User identity association	FAU_SEL.1(1) states that the TSF shall be able to select what is audited by the TOE based on trace level. These include: none, error, dsa, ldap, warn, stats, query, time, asn, stack, and fall.
	FAU_SEL.1(2) Audit Review	FAU_SEL.1(2) states that the TSF shall be able to select what is audited by the TOE based on which audit files have been configured. These files include: summary, stats, query, connect, update, cert, alert, warn, diag, time, SNMP, and Alarm.

Objective	Security Functional Components	Rationale
O.IDENTIFY The TOE will provide measures to uniquely identify all users and will maintain their original identity if they issue commands as a SuperUser in the environment.	FIA_UID.1 User Identification before Any Action	FIA_UID.1 states that the TSF shall allow users Read access to the public repository based on user policy configurations. This can be disabled.
	FIA_ATD.1 User Attribute Definition	FIA_ATD.1 states that the TSF shall maintain user attributes that are used for authentication. These attributes allow for the unique identification of users during their authentication.
O.MANAGE The TOE will provide authorized Directory Administrators and Administrative Users with the resources to manage and monitor user accounts, resources, and security information relative to the TOE.	FMT_MTD.1 Management of security functions behaviour	FMT_MTD.1 states that the TSF shall be able to search, query, modify, and delete repository and configuration file data based on role.
	FMT_MSA.1 Management of security attributes	FMT_MSA.1 states that the user policy levied by the TOE allows for querying, modification, and deletion of access control rules by Directory Administrators via the Configuration Files.
	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.
	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 7-3: Global Group and Group Authorization Attributes for Administrative Users.
	FMT_SMR.1 Security Roles	FMT_SMR.1 states that the TOE will provide the ability to set roles for security relevant authority as well as to restrict the ability to define and assign roles to authorize Administrative Users as well as to view Directory data by those users authorized to do so.

Objective	Security Functional Components	Rationale
O.PASSWORD The TOE will enforce defined organizational password complexity requirements.	FIA_SOS.1 Verification of Secrets	FIA_SOS.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 a DSA occurs.
	FRU_FLT.1 Degraded Fault Tolerance	FRU_FLT.1 ensures the operation of access control to resources when the DSA fails.
O.ROBUST_ADMIN_GUIDANCE The TOE will provide Directory 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.

Objective	Security Functional Components	Rationale
OE.NOEVIL Administrative Users of the TOE are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided by the guidance documentation.	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.
OE.ADMIN One or more authorized Directory Administrators will be assigned to install, configure and manage the TOE.	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 11-3: Security Functional Requirements Rationale

11.3 EAL Justification

The threats that were chosen are consistent with an attacker of low attack potential, therefore EAL3 augmented with ALC_FLR.1 was chosen for this ST.

11.4 Requirement Dependency Rationale

Functional Component	Dependency	Included
FAU_GEN.1	FPT_STM.1	No, in the evaluated configuration the OS is responsible for providing time stamps.
FAU_SEL.1(1)	FAU_GEN.1	Yes
	FMT_MTD.1	Yes
FAU_SEL.1(2)	FAU_GEN.1	Yes
	FMT_MTD.1	Yes
FDP_ACC.1	FDP_ACF.1	Yes
FDP_ACF.1	FDP_ACC.1	Yes
	FMT_MSA.3	Yes
FIA_AFL.1	FIA_UAU.1	Yes
FIA_ATD.1	None	N/A
FIA_SOS.1	None	N/A
FIA_UAU.1	None	N/A
FIA_UAU.5	None	N/A
FIA_UID.1	None	N/A
FMT_MSA.1	FDP_ACC.1	Yes
	FDP_IFC.1	No, Not required if FDP_ACC.1 is included
	FMT_SMR.1	Yes
	FMT_SMF.1	Yes

Functional Component	Dependency	Included
FMT_MSA.3	FMT_MSA.1	Yes
	FMT_SMR.1	Yes
FMT_MTD.1	FMT_SMR.1	Yes
	FMT_SMF.1	Yes
FMT_SMF.1	None	N/A
FMT_SMR.1	FIA_UID.1	Yes
FPT_FLS.1	None	N/A
FRU_FLT.1	FPT_FLS.1	Yes

Table 11-4: Requirement Dependencies

11.5 Assurance Measures

The SARs for this evaluation have been chosen because they are consistent with the package claim of EAL3. CA Directory R12 SP3 is also augmented with ALC_FLR.1.

The following table identifies the SARs for this ST.

Component	Document(s)	Rationale
ADV_ARC.1 Security Architecture Design	TOE Design Specification for CA Directory R12 SP3	This document describes the security architecture of the TOE.
ADV_FSP.3 Functional Specification with complete summary	Functional Specification Document for CA Directory R12 SP3	This document describes the functional specification of the TOE with complete summary.
ADV_TDS.2 Architectural Design	TOE Design Specification for CA Directory R12 SP3.	This document describes the architectural design of the TOE.
AGD_OPE.1 Operational User Guidance	• CA Directory Integration Guide R12 • CA Directory Administration Guide R12 • CA Directory Reference Guide R12	This document describes the operational user guidance for CA Directory.
AGD_PRE.1 Preparative Procedures	• CA Directory Installation Guide R12 • CA Directory Release Summary R12	This document describes the preparative procedures that need to be done prior to installing CA Directory R12 SP3.
ALC_CMC.3 Authorizations Controls	CA Directory R12 Authorizations Controls v1	This document describes the authorization controls for the TOE.
ALC_CMS.3 CM Scope	CA Directory R12 CM Scope v1	These documents describe the CM scope of the TOE.
ALC_DEL.1	CA Directory R12 Delivery Procedures v1	This document describes product delivery for CA

Component	Document(s)	Rationale
Delivery Procedures		Directory and a description of all procedures used to ensure objectives are not compromised in the delivery process.
ALC_DVS.1 Identification of Security Measures	CA Directory R12 Identification of Security Measures v1	This document provides an identification of security measures for the TOE.
ALC_LCD.1 Life-Cycle Definition	CA Directory R12 Life-Cycle Definition v1	This document provides the life-cycle definition of the TOE.
ASE_CCL.1 Conformance Claims	CA Directory R12 SP3 Security Target v3.0	This document describes the CC conformance claims made by the TOE.
ASE_ECD.1 Extended Components Definition	CA Directory R12 SP3 Security Target v3.0	This document provides a definition for all extended components in the TOE.
ASE_INT.1 Security Target Introduction	CA Directory R12 SP3 Security Target v3.0	This document describes the Introduction of the Security Target.
ASE_OBJ.2 Security Objectives	CA Directory R12 SP3 Security Target v3.0	This document describes all of the security objectives for the TOE.
ASE_REQ.2 Security Requirements	CA Directory R12 SP3 Security Target v3.0	This document describes all of the security requirements for the TOE.
ASE_SPD.1 Security Problem Definition	CA Directory R12 SP3 Security Target v3.0	This document describes the security problem definition of the Security Target.
ASE_TSS.2 TOE Summary Specification	CA Directory R12 SP3 Security Target v3.0	This document describes the TSS section of the Security Target.
ATE_COV.2 Analysis of Coverage	CA Directory R12 Analysis of Coverage v1.0	This document provides an analysis of coverage for the TOE.
ATE_DPT.1 Basic Design	CA Directory R12 Basic Design v1.0	This document describes the basic design of the TOE.

Table 11-5: Assurance Requirements Evidence