



## **Security Target**

# **IBM Directory Server 5.1 FixPak510-01**

**Version:** 1.2  
**Status:** Final  
**Last update:** May 7, 2003



## Document History

<b>Version</b>	<b>Date</b>	<b>Changes</b>	<b>Summary</b>	<b>Author</b>
0.8	2002-12-01	Initial version	Initial version produced by IBM	Steve Omrani
0.9	2002-12-20	Updated	Version submitted to BSI	Helmut Kurth
0.91	2003-02-13	Updated	Updated and extended after review from Stephan Müller	Helmut Kurth
0.92	2003-02-18	Updated	Update based on the internal review from Staffan Persson	Helmut Kurth
1.00	2003-02-24	Updated	Changed based on the ESE evaluator findings	Helmut Kurth
1.1	2003-03-12	Updated	Changes based on EFSP evaluator findings.	Helmut Kurth
1.2	2003-05-07	Updated	Changes based on EADM evaluator findings and on comments from BSI	Steve Omrani

## Table of contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	ST Identification	7
1.2	ST Overview	7
1.3	IBM Directory Server Overview	7
1.4	CC Conformance Claim	7
1.5	Strength of Function Claim	7
<b>2</b>	<b>TOE Description</b>	<b>8</b>
2.1	Product Type	8
2.1.1	Defining a Directory	8
2.1.2	Directory clients and servers	9
2.1.3	Directory security	9
2.1.4	Directory Architecture and Operations	9
2.2	Security Roles and Services	10
2.2.1	Delivered Core Services	10
2.2.2	Security Roles	10
2.2.2.1	Directory Administrator	10
2.2.2.2	End User	11
2.2.3	Auxiliary Services	11
2.3	TOE Boundary	12
<b>3</b>	<b>TOE Security Environment</b>	<b>14</b>
3.1	Secure Usage Assumptions	14
3.2	Threats to security	14
3.2.1	Threats addressed by TOE	14
3.2.2	Threats addressed by the operating environment	15
3.3	Organizational Security Policies	15
<b>4</b>	<b>Security Objectives</b>	<b>16</b>
4.1	TOE Security Objectives	16
4.2	Environmental Security Objectives	16
<b>5</b>	<b>IT Security Requirements</b>	<b>17</b>
5.1	TOE Security Functional Requirements	17
5.1.1	FAU_GEN.1 Audit data generation	17
5.1.2	FAU_GEN.2 User identity association	18
5.1.3	FAU_SAR.1 Audit review	18
5.1.4	FAU_SAR.2 Restricted audit review	18
5.1.5	FDP_ACC.2 Complete access control	18
5.1.6	FDP_ACF.1 Security attribute based access control	19
5.1.7	FIA_AFL.1 Authentication failure handling	20
5.1.8	FIA_ATD.1 User attribute definition	20
5.1.9	FIA_SOS.1 Verification of secrets	20
5.1.10	FIA_UAU.1 Timing of authentication	21
5.1.11	FIA_UID.1 Timing of identification	21
5.1.12	FMT_MOF.1a Management of security functions behaviour	21
5.1.13	FMT_MOF.1b Management of security functions behaviour	21
5.1.14	FMT_MSA.1 Management of security attributes	21
5.1.15	FMT_MSA.3 Static attribute initialisation	22
5.1.16	FMT_MTD.1 Management of TSF data	22
5.1.17	FMT_SMF.1 Specification of Management Functions	22
5.1.18	FMT_SMR.1 Security roles	22

- 5.1.19 FPT\_RVM.1 Non-bypassability of the TSP ..... 22
- 5.2 TOE Environment Security Functional Requirements ..... 22
  - IT Security Requirements for the underlying Operating System ..... 22
  - 5.2.1 FPT\_STM.1 Reliable time stamps ..... 22
  - Non-IT Security Requirements for the TOE Environment ..... 23
- 5.3 TOE Security Assurance Requirements ..... 23
- 6 TOE Summary Specification ..... 25**
  - 6.1 TOE Security Functions ..... 25
    - 6.1.1 F.AUDIT ..... 25
      - 6.1.1.1 Audit Generation (F.AUDIT.1) ..... 25
    - 6.1.2 F.ACCESS\_CONTROL ..... 27
      - 6.1.2.1 Order of Evaluation ..... 28
      - 6.1.2.2 Access Control Attributes ..... 28
    - 6.1.3 F.I&A ..... 29
      - 6.1.3.1 User Authentication (F.I&A.1) ..... 29
    - 6.1.4 F.MANAGEMENT ..... 29
      - 6.1.4.1 Roles (F.MANAGEMENT.1) ..... 29
      - 6.1.4.2 Authentication Function (F.MANAGEMENT.2) ..... 30
      - 6.1.4.3 Authorization Functions (F.MANAGEMENT.3) ..... 31
      - 6.1.4.4 Audit Function (F.MANAGEMENT.4) ..... 31
    - 6.1.5 F.REF\_MEDIATION ..... 32
    - 6.1.6 TOE Security Functions rationale ..... 32
  - 6.2 Assurance Measures ..... 32
- 7 Protection Profile Claims ..... 34**
  - 7.1 PP Reference ..... 34
- 8 Rationale ..... 35**
  - 8.1 Security Objectives Rationale ..... 35
  - 8.2 Security Requirements Rationale ..... 37
    - 8.2.1 Security Functional Requirements Rationale ..... 37
    - 8.2.2 Dependency analysis ..... 40
    - 8.2.3 Demonstration of mutual support between security requirements ..... 41
    - 8.2.4 Appropriateness of assurance requirements ..... 42
  - 8.3 TOE Summary Specification Rationale ..... 42
    - 8.3.1 TOE Security Functions rationale ..... 42
    - 8.3.2 Minimum Strength of Function Level rationale ..... 45
  - 8.4 Assurance measures rationale ..... 45

## List of figures

Figure 1: IBM Directory Architecture and TOE Boundary .....	13
---	----

## List of tables

Table 1: Summary of Security Functional Requirements for the TOE .....	17
Table 2: TOE assurance components .....	24
Table 3: Assurance Measures .....	33
Table 4: Mapping of Security Objectives to Threats and Assumptions .....	35
Table 5: Mapping of Security Functional Requirements to TOE Security Objectives .....	37
Table 6: Dependency Mapping of Security Functional Requirements .....	40
Table 7: Mapping of Security Functions to Security Functional Requirements .....	42
Table 8: Mapping of Assurance Measures to Assurance Requirements .....	45

## References

- [CC] Common Criteria for Information Technology Security Evaluation, CCIMB-99-031, Version 2.1, August 1999
- [CEM] Common Methodology for Information Technology Security Evaluation, CEM-99/045, Part 2 - Evaluation Methodology, Version 1.0, 1999
- [GUIDE] ISO/IEC PDTR 15446 Title: Information technology – Security techniques – Guide for the production of protection profiles and security targets, ISO/IEC JTC 1/SC 27 N 2449, 2000-01-04
- [RFC1777] Lightweight Directory Access Protocol, RFC 1777
- [RFC2251] Lightweight Directory Access Protocol (v3), RFC 2251
- [RFC2252] Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions, RFC 2252
- [RFC2253] Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names, RFC 2253
- [RFC2254] The String Representation of LDAP Search Filters, RFC 2254
- [RFC2255] The LDAP URL Format, RFC 2255
- [RFC2256] A Summary of the X.500(96) User Schema for use with LDAPv3, RFC 2256
- [ACLTest] ACL Tests, Version 1.1
- [AdminGuide] IBM Directory Server Version 5.1, Administration Guide, Version 5.1
- [AUDCERT] AUDCERT, Version 2.0
- [C4Guide] IBM Directory Server Version 5.1, Security Guide, Version 0.8
- [ConfMgt] Configuration Management Version Control: For the IBM Directory Server
- [FSP] IBM Directory Server Version 5.1, Functional Specification version 0.7
- [HLD] IBM Directory Server Version 5.1, High-level Design version 1.2
- [InstGuide] IBM Directory Server Version 5.1, Installation and Configuration Guide, Version 5.1
- [RCR] IBM Directory Server Version 5.1, Correspondence Analysis, Version 1.4
- [SOF] IBM Directory Server Version 5.1, Strength of Function Analysis
- [TestPlan] IBM Directory Server Version 5.1, Secure Mode Test Plan, Version 3.0
- [TQA] Download Director Applet: Functional Specification, Version 5.0
- [ProgRef] IBM Directory Server C-Client SDK Programming Reference Version 5.1

[PWPCERT] PWPCERT, Version 2.0

[VA] IBM Directory Server Version 5.1, Vulnerability analysis

# 1 Introduction

## 1.1 ST Identification

**Title:** Security Target for the IBM Directory Server Version 5.1 FixPak510-01

**Assurance level:** EAL2

**Keywords:** Light weight Directory Access Protocol (LDAP), Access Control List (ACL), Password Policy (PP), Audit Service (AS), IBM Directory Server (IDS)

## 1.2 ST Overview

This document is the Security Target (ST) for the IBM Directory Server Version 5.1 FixPak510-01. This Directory Server is running on Microsoft Windows 2000, IBM AIX 5.2, Sun Solaris 8, Red Hat Advanced Server 2.1 and SuSE Linux Enterprise Server 8. The Security Target has been developed in accordance with the Common Criteria for Information Technology Security Evaluation (CC) version 2.1, for a claimed Evaluation Assurance Level 2 (EAL 2).

## 1.3 IBM Directory Server Overview

IBM Directory Server version 5.1 is an implementation of Lightweight Directory Access Protocol (LDAP), which is compliant with the Internet Engineering Task Force (IETF) LDAP Version 2 specifications, i.e. RFC 1777 and Version 3 specifications, i.e. RFC 2251 - 2256. The server is a software only product and can be installed and operated on variety of hardware/software platforms.

LDAP is essentially a specialized database where the update operation is less frequent and dedicated to the common goal within the enterprise on consolidating and unifying the management of identity. IBM Directory Server is built for identity management with role supports, fine-grained access control and entry ownership. It provides the foundation for improved security, rapid development and deployment of Web applications. Using the power of the IBM DB2 Universal Database as back end data store, IBM Directory Server provides high performance, reliability and stability in an enterprise or e-business. As the central repository for data within an enterprise, it is a powerful, secure and standards compliant enterprise directory for corporate intranets and the Internet.

## 1.4 CC Conformance Claim

The ST is Part 2 conformant and Part 3 conformant to the CC. This means that it is conformant with the security functional requirements as specified in CC Part 2, and with the security assurance requirements for Evaluation Assurance Level 2 (EAL 2), as specified in Part 3 of the CC.

## 1.5 Strength of Function Claim

The TOE contains one non-cryptographic security function that is realized by a probabilistic or permutational mechanism, i.e. password based authentication. The minimum strength level claimed for this function is **SOF-basic**. Thus, the global minimum strength level claimed for the TOE is also **SOF-basic**.



## 2 TOE Description

This section describes the Target of Evaluation (TOE) in terms of the class of product, the operational environment, and the provided security functionality. This chapter provides a general description of the product without focusing on the evaluated configuration.

### 2.1 Product Type

The IBM Directory Server (IDS) Version 5.1 FixPak510-01 is an implementation of the Lightweight Directory Access Protocol (LDAP) and meets the requirements of LDAP Version 3 as defined in RFC 2251–2256 and LDAP Version 2 as defined in RFC 1777. The product consists of two major components: the LDAP Server and the LDAP Server Administration Daemon. The TOE described in this ST includes both of them. The server may be partitioned into two parts; the Front-end and the Back-end. The Front-end is the network interface to LDAP clients and the Back-end is the interface to the DB2 database. The Administration Daemon provides an LDAP interface to clients, used for the administration of the LDAP server.

The IBM Directory Server (IDS) Version 5.1 is a software product only, delivered over the Internet as a package including the TOE, user and administrative tools, a WebSphere HTTP server, and a DB2 database. The user and administrator tools, the HTTP server and the DB2 database are all excluded from the TOE and are considered part of the environment.

The TOE environment must also include applications that are not delivered with the IDS product, but are used as unprivileged tools, for example the Netscape browser needed to administrate the TOE or the Adobe Acrobat Reader to access the supplied online documentation.

The IBM Directory Server (IDS) Version 5.1 delivery also consists of on-line documentation. The on-line documents include, in addition to the READMEs, the QuickStart, the Installation and Configuration Guide, the Administration Guide, the Tuning Guide, the C-Client SDK Programming References, the Server Plug-in Reference are provided in PDF and HTML formats.

#### 2.1.1 Defining a Directory

A directory is a collection of information about objects arranged in some order that gives details about each object. It is a specialized database, which stores typed and ordered information about objects. Directories enable users or applications to find resources that have the characteristics needed for a particular task.

If the name of an object is known, its characteristics can be retrieved. If the name of a particular individual object is not known, the directory can be searched for a list of objects that meet a certain requirement. Directories can usually be searched by specific criteria, not just by a predefined set of categories.

A directory is a specialized database that has characteristics that set it apart from general purpose relational databases. A characteristic of a directory is that it is accessed (read or searched) much more often than it is updated (written). Because directories must be able to support high volumes of read requests, they are typically optimized for read access. Because directories are not intended to provide as many functions as general-purpose databases, they can be optimized to economically provide more applications with rapid access to directory data in large distributed environments.

A directory can be centralized or distributed. If a directory is centralized, there is one directory server (or a server cluster) at one location that provides access to the directory. If the directory is distributed, there is more than one server, usually geographically dispersed, that provides access to the directory.

When a directory is distributed, the information stored in the directory can be partitioned or replicated. When information is partitioned, each directory server stores a unique and non-overlapping subset of the information. That is, one and only one server stores each directory entry. The technique to partition the directory is to use LDAP referrals. LDAP referrals allow the users to refer Lightweight Directory Access Protocol (LDAP) requests to either the same or different name spaces stored in a different (or same) server. When information is replicated, more than one server stores the same directory entry. In a distributed directory, some information may be partitioned, and some information may be replicated.

### 2.1.2 Directory clients and servers

Directories are usually accessed using the client-server model of communication. The client and server processes might or might not be on the same machine. A server is capable of serving many clients. An application that wants to read or write information in a directory does not access the directory directly. Instead, it calls a function or application programming interface (API) that causes a message to be sent to another process. This second process accesses the information in the directory on behalf of the requesting application. The results of the read or write are then returned to the requesting application.

An API defines the programming interface a particular programming language uses to access a service. The format and contents of the messages exchanged between client and server must adhere to an agreed upon protocol. LDAP defines a message protocol used by directory clients and directory servers. There is also an associated LDAP API for the C language and ways to access the directory from a Java application using the Java Naming and Directory Interface (JNDI).

### 2.1.3 Directory security

A directory should support the basic capabilities needed to implement a security policy. The directory might not directly provide the underlying security capabilities, but it might be integrated with a trusted network security service that provides the basic security services. First, a method is needed to authenticate users. Authentication verifies that users are who they say they are. A user ID and password is a basic authentication scheme. Once users are authenticated, it must be determined if they have the authorization or permission to perform the requested operation on the specific object.

Authorization is often based on access control lists (ACLs). An ACL is a list of authorizations that may be attached to objects and attributes in the directory. An ACL lists what type of access each user or a group of users is allowed or denied. In order to make ACLs shorter and more manageable, users with the same access rights are often put into groups.

### 2.1.4 Directory Architecture and Operations

LDAP defines the content of messages exchanged between an LDAP client and an LDAP server. The messages specify the operations requested by the client (search, modify, delete, and so on), the responses from the server, and the format of data carried in the messages. LDAP messages are carried over TCP/IP, a connection-oriented protocol; so there are also operations to establish and disconnect a session between the client and server.

The general interaction between an LDAP client and an LDAP server takes the following form:

- The client establishes a session with an LDAP server. This is known as *binding* to the server. The client specifies the host name or IP address and TCP/IP port number where the LDAP server is listening. The client can provide a user name and a password to properly authenticate with the server. Or the client can establish an anonymous session using bind with default access rights. The client and server can also establish a session that uses stronger security methods such as encryption of data.

- The client then performs operations on directory data. LDAP offers both read and update capabilities. This allows directory information to be managed as well as queried. LDAP also supports searching the directory for data meeting arbitrary user-specified criteria. Searching is a very common operation in LDAP. A user can specify what part of the directory to search and what information to return. A search filter that use Boolean conditions, specifies what directory data matches the search.
- When the client is finished making requests, it closes the session with the server. This is also known as *unbinding*.

**Note:** Unauthenticated users are users that have not performed a bind, while anonymous users are users that have performed a bind without providing a Distinguished Name (DN) or password. Apart from distinguishing between the two when logging, there is no difference in the way unauthenticated and anonymous users are being treated by the TOE. For this reason we will use the term unauthenticated both for the anonymous and the unauthenticated.

## 2.2 Security Roles and Services

### 2.2.1 Delivered Core Services

The server allows authorized clients to:

- Add entries to the directory,
- Delete entries from the directory,
- Modify the attributes of an entry,
- Modify the Distinguished Name (DN). For LDAPv2 this operation is called Modify Relative Distinguished Name (RDN),
- Search the directory for entries that meet certain filter criteria,
- Compare to check for presence of entries or attributes matching the compare criteria,
- Abandon (terminate) a LDAP operation,
- Extended operations, which are server the side enhancements to the LDAP operations as delivered by IBM.

In addition, the server allows unauthenticated users to perform any operation on any entries or attributes that are not blocked by any ACL.

### 2.2.2 Security Roles

The IBM Directory Server supports two security roles: *Directory Administrator* and *End User*. The following sections describe the capability of each one.

Note that IBM Directory Server has its own definition of roles, which is similar to LDAP groups and should not be confused with the security roles. The only difference between roles and groups is that when a user is assigned to a role, there is an implicit expectation that the necessary authority has already been set up to perform the job associated with that role. With group membership, there is no built-in assumption about what permissions are gained (or denied) by being a member of that group. In this evaluation, the groups and roles are regarded as authorization attributes instead.

#### 2.2.2.1 Directory Administrator

The Directory Administrator security role is created during product installation. It consists of a user ID and a password and predefined authorization to manipulate the entire directory. The Directory Administrator creates the End User security role. This is an LDAP entry with a

specific Distinguished Name (DN), User Password and other attributes that represent the particular End user. The Directory Administrator also defines the level of authorization the End User will have over entries.

### 2.2.2.2 End User

Each End User is identified with an LDAP entry containing the authentication and authorization information for that End User. The authentication and authorization information may also allow the End User to query and update other entries. The user ID and password are validated during the bind operation. Once the End User ID and password are validated, the End User may access any of the attributes of any entry to which he has permissions.

## 2.2.3 Auxiliary Services

In addition to the implementation of Lightweight Directory Access Protocol (LDAP), IBM Directory Server version 5.1 also includes enhancements added by IBM in functional and performance areas. This version uses the IBM DB2<sup>®</sup> as the backing store to provide per LDAP operation transaction integrity, high performance operations, and on-line backup and restore capability. It interoperates with the IETF LDAP V3 and V2 based clients. Besides supporting the standard LDAP operations, it additionally includes the following features:

- Administration and configuration for the IBM Directory Server is provided by LDAP clients being either a Web browser-based GUI or command line clients. The administration and configuration functions allow the Directory Administrator to:
  - Perform the initial setup of the directory
  - Change configuration parameters and options
  - Manage the daily operations of the directory

The administrative tasks of the TOE are mainly done through LDAP operations on the IBM Directory Server. For starting, stopping, restarting and querying the status of the IBM Directory Server, LDAP operations are performed on the administration daemon, which then will carry out the operations on the directory server.

- A dynamically extensible directory schema – This means that administrators can define new attributes and object classes to enhance the directory schema. Changes can also be made to the directory schema, which are subject to consistency checks. Users may dynamically modify the schema content without restarting the directory server. Since the schema itself is part of the directory, schema update operations are done through standard LDAP APIs. The following functions are provided by the IBM Directory Server to determine the status of the server and the LDAPv3 dynamic extensible schema:
  - Queryable schema information through LDAP APIs
  - Dynamic schema changes through LDAP APIs
  - Server rootDSE can be queried for status information of the server
  - Dynamic configuration changes using LDAP APIs
- UTF-8 (Universal Character Set Transformation Format) – An IBM Directory server supports data in multiple languages, and allows users to store, retrieve and manage information in a native language code page.
- Simple Authentication and Security Layer (SASL) – This support provides for additional authentication mechanisms, which is a plug-in facility to allow different authentication methods to be used. The SASL mechanism is not part of the TOE. The user ID and password is the basic authentication scheme used and being part of this TOE.

- The Secure Sockets Layer (SSL) provides encryption of data and authentication using X.509v3 public-key certificates. A server may be configured to run with or without SSL support. However, SSL support is not part of the TOE but part of the TOE environment.
- Replication – Replication is supported, which makes additional read-only copies of the directory available, improving performance and reliability of the directory service. However, replication is not part of the TOE.
- Referrals – Support for LDAP referrals, allowing directories to be distributed across multiple LDAP servers where a single server may only contain a subset of the whole directory data.
- Access Control Model – A powerful, easy-to-manage access control model on LDAP objects (entries or attributes) is supported through ACLs.
- Change log – Changes made to the LDAP data are being logged internally in the LDAP server to support replication of these information. However, replication is not part of the TOE.
- Security audit logging – Auditing of the LDAP operations is provided by the security auditing logging.

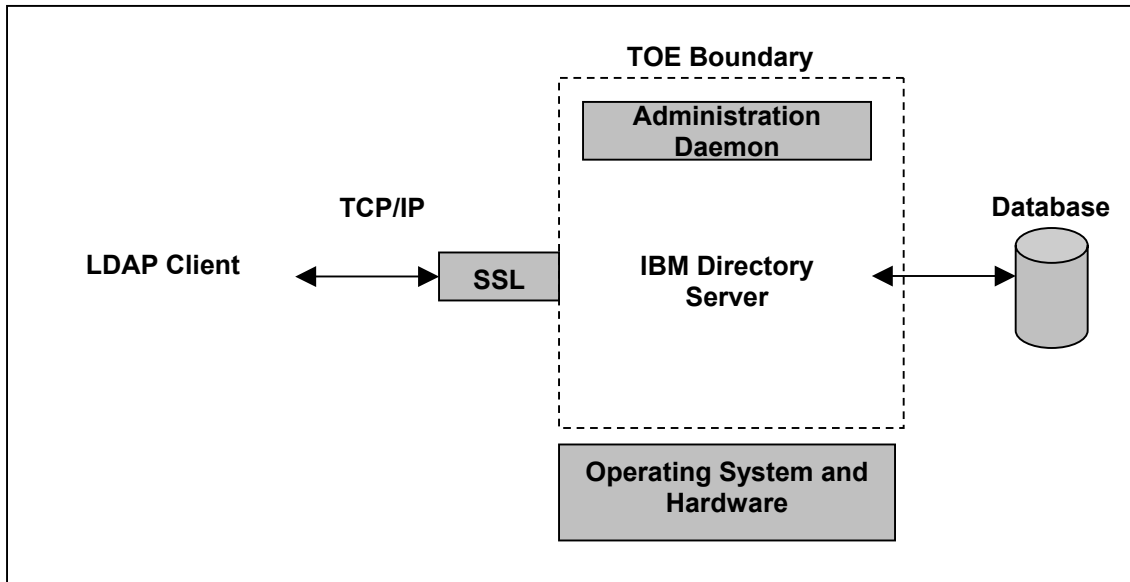
### 2.3 TOE Boundary

**Figure 1** illustrates the client/server based IBM Directory architecture. The rectangle represented by the dash lines indicates the TOE boundary, i.e. the standalone Directory Server and the Administration Daemon is the Target of Evaluation. Those, out of the scope of evaluation, are listed as below:

- The underlying hardware and operating system of the Directory Server
- Database, which serves as the back end data store of the directory
- LDAP client
- SSL module, which provides trust path between LDAP client and IBM Directory Server, and also provides trust path among replication servers
- Replication service

The underlying hardware and operating system, the database, the LDAP client and the SSL module are part of the TOE environment.

The TOE consists of two components: the directory server component and the administration daemon. User clients are connecting both to the LDAP server (shown in the picture as the IBM Directory Server) and to the administration daemon, using the LDAP protocol, but using different port numbers. The directory server is providing the LDAP functionality to users and administrators, while the administration daemon is only used by the administrators for starting, stopping and querying the status of the IBM Directory Server.



**Figure 1: IBM Directory Architecture and TOE Boundary**

Note that the hardware and the operating system the Directory Server uses are part of the TOE environment. Note also that the SSL protocol (when used) has to be provided by the TOE environment. The TOE assumes a secured communication link between itself and the client as well as between itself and the database.

## 3 TOE Security Environment

### 3.1 Secure Usage Assumptions

The following conditions are assumed to exist in the TOE operational environment. These assumptions include essential environmental constraints on the secure use of the TOE.

- A.PHYSICAL** The TOE is operated in a physically secure environment.
- A.NOEVIL** The TOE and TOE Environment Administrators are trustworthy to perform discretionary actions in accordance with security policies and not to interfere with the abstract machine.
- A.ADMIN** The TOE and TOE environment are competently installed and administered.
- A.COMM** It is assumed that a communication link exists from the TOE to LDAP clients on external systems, and that any communication links between the TOE and external systems are protected against unauthorized modification and disclosure of communication data.
- A.COOP** Authorized end users are trusted and expected to act in a cooperating manner in a benign environment.
- A.TIME** It is assumed that a reliable time is provided by the TOE environment.

### 3.2 Threats to security

The threats are categorized as those addressed by the TOE and those addressed by the environment.

The assets held in the TOE are information and resources under the control of the TOE, such as directory entries and TSF data. It is assumed that an attacker is either an unauthorized user of the TOE, or an authorized user of the TOE who has been granted rights to access the information or resources held by the TOE. The attacker is assumed to have limited resources and to come from a well-managed user community in a non-hostile working environment. The TOE is not intended to be used in an environment in which protection against determined or sophisticated attacks is required.

#### 3.2.1 Threats addressed by TOE

The TOE address the threats discussed below.

- **T.ENTRY:** An illegitimate user (i.e. one other than authenticated user) could gain unauthorized, malicious access to resources or information, other than public information, protected by the TOE.
- **T.ACCESS:** A legitimate user of the TOE could gain unauthorized access to a 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.

A legitimate user is someone who:

- is authenticated uniquely by the IBM Directory Server, or
- is unauthenticated, and appear as an anonymous user.

- **T.ACCOUNT:** Security relevant actions occur without awareness by Directory Administrators. Lack of accountability of security relevant events of user or system process may lead to failure in identifying possible security violations and holding those responsible accountable.
- **T.BYPASS:** An attacker may bypass TOE security functions to gain access to resources or information protected by the TOE.

### 3.2.2 Threats addressed by the operating environment

The threats discussed below must be countered in order to support the TOE security capabilities but are either:

- not addressed by, or
- only partly addressed by the TOE

Such threats must therefore be addressed in conjunction with the operating environment.

- **TE.USAGE:** The TOE may be configured, used and administered in an insecure manner, allowing an illegitimate user gaining access to resources or information protected by the TOE.
- **TE.CRASH:** Human error or a failure of software, hardware, power supply, or an accidental event may cause an abrupt interruption to the TOE operation, resulting in loss or corruption of data.
- **TE.SOPHISTICATED:** An unauthorized individual may gain access to TOE resources or information by using sophisticated technical attack, using IT security-defeating tools applied to the TOE or the underlying system components.
- **TE.PASS:** An attacker may bypass the TOE to access resources or resources protected by the TOE by attacking the underlying operating system or database, in order to gain access to TOE resources and information.

## 3.3 Organizational Security Policies

The TOE complies with the following organizational security policy:

- **P.PUBLIC:** Of the information under the control of the TOE, only information classified as public information should be made available to unauthenticated or anonymous users.



## 4 Security Objectives

This section defines the security objectives for the TOE and its environment respectively.

### 4.1 TOE Security Objectives

The following lists the security objectives that the TOE meets.

- |                       |  |
|-----------------------|--|
| <b>O.AUTHENTICATE</b> | The TOE must ensure that all users are identified and authenticated before being granted access to the TOE mediated resources except for allowing unauthenticated users to perform some operations on public data. Such limited access to the TOE is configured by Directory Administrators and should be compliant with the security policy of the organization responsible for the operation of the TOE. |
| <b>O.AUTHORIZE</b>    | The TOE must provide the ability to specify and manage access rights to objects and services by user and system process.   |
| <b>O.ACCOUNT</b>      | The TOE must ensure that all TOE users can subsequently be held accountable for their security relevant actions, except for unauthenticated users, who are granted limited access to TOE.  |
| <b>O.BYPASS</b>       | The TOE security policy enforcement functions must be invoked and succeed before access to TOE objects and services are allowed.   |

### 4.2 Environmental Security Objectives

Some security needs are beyond the capability of the TOE to adequately satisfy without support from the TOE operational environment. These security needs derive environmental security objectives which are listed as below:

- |                         |  |
|-------------------------|--|
| <b>OE.MANAGE</b>        | Those responsible for the TOE must ensure that the TOE is installed, and managed in a secure manner, which maintains the security of the TOE, TSF data and user data of the TOE.   |
| <b>OE.OS-MANAGE</b>     | Those responsible for the TOE must ensure that the underlying operating system and hardware is configured and managed in a secure manner.  |
| <b>OE.PHYSICAL</b>      | Those responsible for the TOE must ensure that the TOE and its underlying hardware and software are physically protected from unauthorized physical access and tampering.  |
| <b>OE.DATABASE</b>      | The database used to store the TSF and user data is configured and managed in a secure way that prohibits unauthorized access and tampering with the TSF data and user data of the TOE.  |
| <b>OE.SOPHISTICATED</b> | The TOE environment must sufficiently counter the threat of an individual (other than an authorized user) gaining unauthorized access via sophisticated technical attack.  |
| <b>OE.BACKUP</b>        | The TOE environment must provide backup facility for recovery to a secure state following a system failure, discontinuity of service, or detection of an insecure status. This includes the provisions necessary to ensure the availability of the TSF data and user data stored outside of the TOE. |
| <b>OE.COMMUNICATION</b> | The communication links between the TOE and LDAP clients on external systems are protected from unauthorized modification and disclosure of communication data   |
| <b>OE.TIME</b>          | The TOE environment must provide a reliable time source.   |

## 5 IT Security Requirements

This section contains the security functional requirements (SFRs) and security assurance requirements (SARs) that must be satisfied by the TOE. These requirements consist of functional components from Part 2 of the CC and Evaluation Assurance Level (EAL) 2 assurance components from Part 3 of the CC, respectively.

### 5.1 TOE Security Functional Requirements

This section identifies and specifies the SFR components that the TOE, IBM Directory Server, is intended to meet for the purposes of this CC evaluation. All of these SFR components are chosen from Part 2 of the CC to directly or indirectly (i.e., via a functional component dependency) satisfy the security objectives for the TOE, summarized in **Table 1**.

Operations that are completed on the SFR components are indicated throughout this section through the use of Bold Italic text. The iteration operation has only been performed once for FMT\_MOF.1. The two SFR components are identified as FMT\_MOF.1a and FMT\_MOF.1b.

**Table 1: Summary of Security Functional Requirements for the TOE**

SFR Components	
Identifier	Name
FAU_GEN.1	Audit data generation
FAU_GEN.2	User identity association
FAU_SAR.1	Audit review
FAU_SAR.2	Restricted audit review
FDP_ACC.2	Complete access control
FDP_ACF.1	Security attribute based access control
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_UID.1	Timing of identification
FMT_MOF.1a	Management of security functions behavior
FMT_MOF.1b	Management of security functions behavior
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
FPT_RVM.1	Non-bypassability of the TSP

#### 5.1.1 FAU\_GEN.1 Audit data generation

**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;
- c) Including the following events:
  - ***Bind operations (LDAP v2 and v3)***
  - ***Unbind operations (LDAP v2 and v3)***

- **Search operations (LDAP v2 and v3)**
- **Add operations (LDAP v2 and v3)**
- **Modify operations (LDAP v2 and v3)**
- **Delete operations (LDAP v2 and v3)**
- **ModifyDN (LDAP v3) and ModifyRDN (LDAP v2) operations**
- **LDAP v3 Event Notification extended operations**

**FAU\_GEN.1.2** The TSF shall record within each audit record at least the following information:

- a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
- b) For each audit event type, based on the auditable event definitions of the functional components included in the ST: **none**

**Application Note:** The subject identity for unauthenticated or anonymous users assigned will be the one of the unauthenticated or anonymous user identify. The subject identity for the start-up and shutdown of the audit function is not being audited.

#### 5.1.2 FAU\_GEN.2 User identity association

**FAU\_GEN.2.1** The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

**Application Note:** The unauthenticated or anonymous users will be assigned the unauthenticated or anonymous user identify.

#### 5.1.3 FAU\_SAR.1 Audit review

**FAU\_SAR.1.1** The TSF shall provide **Directory Administrators** with the capability to read **all audit information** from the audit records.

**FAU\_SAR.1.2** The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

#### 5.1.4 FAU\_SAR.2 Restricted audit review

**FAU\_SAR.2.1** The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-write-access.

#### 5.1.5 FDP\_ACC.2 Complete access control

**FDP\_ACC.2.1** The TSF shall enforce the **Directory Access Control SFP** on **users as subjects and directory entries and attributes as objects** and all operations among subjects and objects covered by the SFP.

**FDP\_ACC.2.2** The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.

**Application Note:** Access rights may be attached to users, groups and roles. Subjects that can access directory entries or attributes are users, but the decision if to grant the requested access takes the user's membership in groups and the user's role into account.

## 5.1.6 FDP\_ACF.1 Security attribute based access control

**FDP\_ACF.1.1** The TSF shall enforce the *Directory Access Control SFP* to objects based on **two attributes sets**:

1. **Entry Owner Information:**
  - **entryOwner:** define entry owner.
  - **ownerPropagate:** indicate whether to propagate of the ownership to all of descendant entries of current entry.
2. **Access Control Information (ACI)**
  - **Non-filter based**
    - **aclEntry:** define the access control information.
    - **aclPropagate:** indicate whether to propagate access control information to all of descendant entries of current entry.
  - **Filter base**
    - **ibm-filterAclEntry:** define filter-based access control information.
    - **ibm-filterAclInherit:** indicate whether to terminate accumulation of access control information.

**FDP\_ACF.1.2** The TSF shall enforce the following rule to determine if an operation among controlled subjects and controlled objects is allowed: **in the following evaluation order**:

1. **By comparing the subject's bind DN with the effective entryOwner attribute values. The entry owner has full access to the target entry. A Directory Administrator is always the owner of all entries in the directory tree.**
2. **If the subject does not possess the entry ownership, the check for access continues by comparing the subject's DN with the effective ACI of the target entry. Depending on the ACI type two access control modes are possible:**
  - I. **In non filter-based ACL this means matching the subject DN with the subject of the ACI information. If a match on the subject is found the permissions defined in the corresponding ACI is enforced.**
  - II. **In filter-based ACL this means matching the subject DN and the requested object, with the subject and object of the ACI information. If a match on both the subject and the object is found the permissions defined in the corresponding ACI is enforced.**
3. **If no ACI information is found for the target object either explicitly or through inheritance, then default access is given.**

**FDP\_ACF.1.3** The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **by allowing any subject access to public information.**

**FDP\_ACF.1.4** The TSF shall explicitly deny access of subjects to objects based on the: **none.**

**Application Note:** Section 6.1.2 defines the rather complex rules for access control. A later version of this Security Target may try to express those rules in a FDP\_ACF.1.

### 5.1.7 FIA\_AFL.1 Authentication failure handling

**FIA\_AFL.1.1** The TSF shall detect when **3** unsuccessful authentication attempts occur related to ***consecutive authentication attempt of the same End User.***

**FIA\_AFL.1.2** When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall ***suspend the End User from further login attempts for a time period defined by the Directory Administrator, or prohibit further login of the user until the Directory Administrator has reset the user's password.***

**Application Note:** The number of unsuccessful authentication attempts can be specified by the Directory Administrator. The value of three is the one selected for the secure configuration. The authentication failure handling applies to all End Users, but not to the Directory Administrator.

### 5.1.8 FIA\_ATD.1 User attribute definition

**FIA\_ATD.1.1** The TSF shall maintain the following list of security attributes belonging to individual users:

- ***userPassword: it specifies the user's password***
- ***pwdChangedTime: it specifies the last time the user's password was changed***
- ***pwdFailureTime: it holds the times of the consecutive authentication failures, which is internally maintained by Directory***
- ***pwdAccountLockedTime: it holds the time that the user's account was locked***
- ***pwdReset: it holds a flag to indicates if the password has been reset and therefore must be changed by the user on first authentication***
- ***Distinguished Name of the user***

**Application Note:** In addition to these attributes, there are several other user attributes, which are not considered to be security attributes. All these attributes are maintained for End Users. None of the attributes above, with the exception of the password, is maintained for the Directory Administrator.

### 5.1.9 FIA\_SOS.1 Verification of secrets

**FIA\_SOS.1.1** The TSF shall provide a mechanism to verify that secrets meet ***the password policy constraints, defined by the following attributes:***

- ***Minimum length of 8 character***
- ***Minimum number of 2 non-alphabetic characters***
- ***Minimum of 4 alphabetic characters***
- ***Maximum of 2 identical consecutive characters***
- ***Maximum age of 90 days***
- ***Minimum time of 1 day to expire before a password can be changed again***

**Application Note:** The Directory Administrator has the ability to specify different values than those listed above. The above listed values are those that are considered for a secure configuration. The claimed SOF for the authentication mechanism is SOF-basic. This claim is

based on the setting above for the verification of secrets.. The Directory Administrator password is not subject to the constraints of the password policy.

#### 5.1.10 FIA\_UAU.1 Timing of authentication

**FIA\_UAU.1.1** The TSF shall allow *limited operations as assigned by the Directory Administrator in compliance with the security policy* on behalf of the user to be performed before the user is authenticated.

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

**Application Note:** The Directory Administrator may allow unauthenticated users to perform search, add, modify, compare and extended operations on public information, without a previous authentication. The bind/unbind is part of the user authentication and abandon is an allowed operation also for unauthenticated users.

#### 5.1.11 FIA\_UID.1 Timing of identification

**FIA\_UID.1.1** The TSF shall allow *limited operations as assigned by the Directory Administrator in compliance with security 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:** The note for FIA\_UAU.1 also applies for the identification since this is performed as one bind operation.

#### 5.1.12 FMT\_MOF.1a Management of security functions behaviour

**FMT\_MOF.1.1** The TSF shall restrict the ability to *modify the behaviour of* the functions: *authentication mechanism* to *Directory Administrators*.

#### 5.1.13 FMT\_MOF.1b Management of security functions behaviour

**FMT\_MOF.1.1** The TSF shall restrict the ability to *disable, enable, and modify the behaviour of* the functions *audit service* to *Directory Administrators*.

#### 5.1.14 FMT\_MSA.1 Management of security attributes

**FMT\_MSA.1.1** The TSF shall enforce the *Directory Access Control SFP* to restrict the ability to *modify, delete, or read* the security attributes:

- a) *Password – End User (owner) or Directory Administrator*
- b) *Password policy – The attributes of password policy for the whole directory can be read by everybody, but only changed by Directory Administrator.*
- c) *Entry Owner Information – End User (entry owner) or Directory Administrator*
- d) *Access Control Information – End User (entry owner) or Directory Administrator*
- e) *Audit options – Directory Administrator*

to *those users or roles as identified above*.

**Application Note:** The note the passwords cannot be read as clear text, since they are stored as in an encrypted form.

#### 5.1.15 FMT\_MSA.3 Static attribute initialisation

**FMT\_MSA.3.1** The TSF shall enforce the *Directory Access Control SFP* to provide *restrictive* default values for security attributes that are used to enforce the SFP.

**FMT\_MSA.3.2** The TSF shall allow the *Directory Administrator, or authorized End Users with roles of proper privileges* to specify alternative initial values to override the default values when an object or information is created.

#### 5.1.16 FMT\_MTD.1 Management of TSF data

**FMT\_MTD.1.1** The TSF shall restrict the ability to *modify* the *TOE security configuration made in the schema file and the configuration file, to the Directory Administrator*.

#### 5.1.17 FMT\_SMF.1 Specification of Management Functions

**FMT\_SMF.1.1** The TSF shall be capable of performing the following security management functions:

- a) *Password management*
- b) *Password policy management*
- c) *User management*
- d) *Access Control Management*
- e) *Audit management.*

#### 5.1.18 FMT\_SMR.1 Security roles

**FMT\_SMR.1.1** The TSF shall maintain the roles *End User and Directory Administrator*.

**FMT\_SMR.1.2** The TSF shall be able to associate users with roles.

#### 5.1.19 FPT\_RVM.1 Non-bypassability of the TSP

**FPT\_RVM.1.1** The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

## 5.2 TOE Environment Security Functional Requirements

This section identifies and specifies the environmental SFR components that the environment of the TOE is intended to meet for the purposes of this CC evaluation. All of these SFR components are chosen to directly or indirectly (i.e., via a functional component dependency) satisfy the environmental security objectives for the TOE.

### IT Security Requirements for the underlying Operating System

#### 5.2.1 FPT\_STM.1 Reliable time stamps

**FPT\_STM.1.1** The TSF shall be able to provide reliable time stamps for its own use.

## Non-IT Security Requirements for the TOE Environment

The following requirements for the non-IT environment of the TOE need to be satisfied:

### **ER.ATTACK – Sophisticated Attacks**

The TOE as well as the database need to be protected against sophisticated attacks by appropriate measures in the TOE environment if such protection is required by the operational policy. This includes also potential denial-of-service attacks when the functions of the TOE are used for time critical applications.

Note that this requirement may be satisfied by physical / organizational controls, IT controls or a combination of both. This Security Target does not prescribe, how the required protection is achieved.

### **ER.BACKUP – Backup and Recovery**

Backup of the TSF data and the TOE user data have to be taken which allow to restore the TOE with its TSF data and user data in a known and secure state. Backup frequency and strategy must be defined in the security policy of the organization responsible for the operation of the TOE. Backup data needs to be adequately protected to prevent loss, authorized access or modification.

### **ER.COMMUNICATION – Protection of the Communication**

Communication links between the TOE and LDAP clients on external systems need to guarantee the authenticity of the communication partner and need to be protected from unauthorized access to and modification of communication data.

### **ER.OS-MANAGE – Management of the Operating System Environment**

The operating system and hardware used to run the TOE on must be configured and managed by competent and trustworthy personnel in a way that ensures the security, preventing unauthorized access to the TOE, TSF data and user data.

Note that this requirement may be satisfied by physical / organizational controls, IT controls, or a combination of both. This Security Target does not prescribe, how the required protection is achieved.

### **ER.MANAGE – Management of the TOE**

The TOE must be installed and managed by competent and trustworthy personnel in a way that ensures the security, preventing unauthorized access to the TOE, the TSF data and user data.

### **ER.DATABASE – Management of the Database**

The database used to store the TSF and user data must be protected from unauthorized access, tampering and loss of data. The database must be configured and managed by competent and trustworthy personnel to ensure this protection at all operational stages.

## **5.3 TOE Security Assurance Requirements**

The security assurance requirements for EAL2, as specified in Part 3 of the CC, are summarized in the following table:



**Table 2: TOE assurance components**

<b>Component</b>	<b>Component ID</b>	<b>Component Title</b>
Configuration Management	ACM_CAP.2	Configuration items
Delivery and Operation	ADO_DEL.1	Delivery procedures
	ADO_IGS.1	Installation, Generation, and Start-up Procedures
Development	ADV_FSP.1	Informal functional specification
	ADV_HLD.1	Descriptive high-level design
	ADV_RCR.1	Informal Correspondence Demonstration
Guidance Documents	AGD_ADM.1	Administrator Guidance
	AGD_USR.1	User Guidance
Tests	ATE_COV.1	Evidence of coverage
	ATE_FUN.1	Functional Testing
	ATE_IND.2	Independent Testing – Sample
Vulnerability Assessment	AVA_SOF.1	Strength of TOE security function evaluation
	AVA_VLA.1	Developer vulnerability analysis

## 6 TOE Summary Specification

This section provides a description of the TOE security functions and assurance measures, which meet the TOE security requirements specified in Section 5.

### 6.1 TOE Security Functions

The TOE security functions are described as follows:

#### 6.1.1 F.AUDIT

The TOE provides audit generation service, generating audit entries into the **audit log file** in chronological order to keep traces of users' activities and to keep track of any changes made to the audit function. The audit service also generates entries into an **error log file**. The administration daemon writes its audit records for its auditable events into an admin daemon log file.

The audit log file, admin daemon log file and error log file entries are stored as text files, which are managed and reviewed using the audit review as part of F.MANAGEMENT.4. But, these files may also be viewed by the Directory Administrators using any text editor.

##### 6.1.1.1 Audit Generation (F.AUDIT.1)

The audit service will generate at least one audit log entry for each valid LDAP request received by the server. The entry is generated when the server is about to return the corresponding LDAP response to the client, i.e., when the result of the processing is known.

In addition to the LDAP audit events, audit events are also generated by the audit service for starting or stopping of the Audit Service, or any changes made of the audit configuration options. These messages will be logged in the audit log as well as in the error log. This is to cover the case that the audit log is full and no more entries could be added.

The audit log will grow and requires the auditor to manage (delete, save, or replace) the content of the audit log (as part of the TOE environment). In case the audit log file is full, an error message will be written in the error log file; the IBM Directory Server will continue to operate, but the auditing function will no longer generate audit entries into the audit log file.

To accurately track a user's activity, each message entry contains the following information:

- time when the activity occurs,
- the user's identity or unauthenticated and anonymous for unauthenticated and anonymous users respectively,
- the activity, and
- the result of the activity.

The format of each message entry will be:

```
Timestamp 1 -- message text
```

Each non-message entry will contain a header (general information) and operation specific data. The header will be in the following format:

```
Timestamp 1 -- Version number + [SSL] + [unauthenticated or
anonymous] Operation-bindDN: server encoded DN string --
client:Client IP address:Port number -- ConnectionID: xxxx --
received: Timestamp 2 -- Result or Status string
```

Where:

**Timestamp 1**

Is the local time the entry is logged (i.e., when the processing of the request is done).

**Version number +[SSL]+[unauthenticated or anonymous] Operation** Shows the LDAP request that was received and processed. Version number is either V2 or V3. SSL indicates whether SSL was used. The SSL within the brackets will show if SSL was used or not. Unauthenticated indicates whether the request was sent by an anonymous or unauthenticated client. If the client was anonymous or unauthenticated the text “authenticated” or “anonymous” will be shown before the Operation. For authenticated requests, this will be omitted and the bindDN user ID will be shown as described below.

**bindDN: server encoded DN string**

Shows the bind DN. The encoding done by the server is to hide the DN from trivial viewing. The server will provide a decoding function through an LDAP search of special audit related attributes. For V3 unauthenticated or anonymous requests, this field will not be shown.

**client:Client IP address:Port number**

Shows the client’s IP address and port number.

**ConnectionID: xxxx**

Shows the connection identification number used to tie all the entries of a connection together (i.e. all events between a Bind and an Unbind).

**received: Timestamp 2**

Is the local time when the request was received, or more precisely, the beginning time when the request was processed. Its format is the same as Timestamp 1.

**Result or Status**

Shows the result or status of the LDAP operation. The result is a text message like “success” or “operationsError”.

For the LDAP operation a message will follow the header and display the operation specific data of that LDAP operation. The following lists the operation specific data for each auditable operation.

**Bind**

**name** – the DN of the client authenticating to the server; **authenticationChoice** – the authentication choice (simple authentication, sasl or Kerberos, but only simple is part of the evaluation); **authenticationMechanism** – the authentication mechanisms used, but only in case a sasl mechanisms is being used (which is not considered part of the evaluated configuration).

**Unbind**

No operation specific data.

**Search**

**base** – the base DN for the search; **scope** – the scope of the search; **derefAliases** – the dereferencing option supplied by the client indicating how aliases dereferencing should be done in the request (either never, always, when searching, or dereferenced only when locating the base object for the search); **typesOnly** – a Boolean flag indicating whether the client wants to have the attribute types and values, or the attributes types only; **filter** – the filter string used for the search request; **attributes** – and finally the list of all attributes that are requested, provided they are specified by the client in the search request.

**Add**

**entry** – the DN that is added to the DIT (directory information tree); **attributes** – the list of attributes the added attribute contains.

**Modify**

**object** – the DN of the entry subject to modification; **operation** – and the operation type (add, delete or replace) and attribute subject to the operation. This entry is repeated for each operation/attribute pair affected by the request from the client.

**Delete**

**entry** - the DN of the entry requested to be being deleted.

**ModifyDN** (for LDAP v2, this is called ModifyRDN)

**entry** – the DN of the entry subject to renaming; **newrdn** - the new DN after the renaming; **deleteoldrdn** – a Boolean value indicating if the old DN is being deleted (or copied); **newSuperior** – the name of the new parent DN for the new entry.

**Event notification extended operation: registration**

**eventID** – the event notification ID created by the server during registration; **base** – the base entry containing the baseDN, at which the notification starts in the DIT (directory information tree); **scope** – the cope of the operation; **type** – the type of event for which the client is requesting event notification.

**Event notification extended operation: unregister**

**eventID** – the event notification ID created by the server during registration

**Start-up of the audit function**

**name** – the DN of the client requesting the starting of the audit function

**Shut-down of the audit function**

**name** – the DN of the client requesting the starting of the audit function

**6.1.2 F.ACCESS\_CONTROL**

Permission to perform a particular LDAP operation on a specified target object is granted or denied based on the subject's DN (Distinguished Name), established by the bind operation. Users, who have not performed a bind or an anonymous bind, will have an empty DN (NullDN) and are called unauthenticated or anonymous. There is no difference between the access rights given to unauthenticated and anonymous user.

Each entry within the LDAP directory contains the distinguished name of the entry as well as a set of attributes and their corresponding values. Each entry has a list of entry owners kept in the attribute entryOwner. In addition each entry has a set of associated ACIs (Access Control Information). When determining access, the entryOwner information and the ACI information are being used.

The access control using the ACI information is either filter-based or non filter-based. The attributes are mutually exclusive within a single containing directory entry. However, both types can co-exist in the directory tree in separate entries. The ACI type of the target entry ACI determines the mode of calculation. In filter based mode, non-filter based ACLs are ignored in effective access calculation, and vice versa.

The ACIs for an entry is determined in the following way:

- a) If there is a set of explicit access control attributes at the entry, then the entry's ACI applies.
- b) If there is no explicitly defined access control attributes, then traverse the directory tree upwards until an ancestor node is reached with a set of propagating access control attributes.
- c) If no such ancestor node is found, the default access rights will apply. The default access rights are predefined and cannot be changed by the Directory Administrator.

It is possible to grant access to groups by associate multiple subject DN's to DN's representing a group. The LDAP server also maintains dynamic groups called pseudo DN's. These group DN's can be granted rights, which will apply to all group members.

The ACI information applicable to an entry is compiled and used in the following way:

1. Specificity Rule
  - a) The rights assigned to the owner DN dominate over the right of the group DN.
  - b) Within the same entry, individual attribute permissions dominate over the attribute class permissions.
  - c) Within the same attribute or attribute class, deny dominates over grant.
2. Combinatory Rule
  - For each entry the permissions granted to subjects of equal importance, as described under a), b) and c) above are combined.
  - If the access cannot be determined within the same specificity level, the access definitions of a less specific level are used.
  - If the access is not determined after defined ACIs are applied, the access is denied.

#### 6.1.2.1 Order of Evaluation

When determine access, processing stops as soon as access can be determined based on access evaluation order, evaluation mode and evaluation rules as described below:

1. The first check for access is done by comparing the subject's bind DN with the effective entryOwner attribute values. The entry owner has full access to the target entry. A Directory Administrator is always the owner of all entries in the directory tree.
2. If the subject does not possess the entry ownership, the check for access continues by comparing the subject's DN with the effective ACI of the target entry. Depending on the ACI type two access control modes are possible:
  - i. In non filter-based ACL this means matching the subject DN with the subject of the ACI information. If a match on the subject is found the permissions defined in the corresponding ACI are enforced.
  - ii. In filter-based ACL this means matching the subject DN and the requested object, with the subject and object of the ACI information. If a match on both the subject and the object is found the permissions defined in the corresponding ACI is enforced.
3. If no ACI information is found for the target object either explicitly or through inheritance, then default access is given.

#### 6.1.2.2 Access Control Attributes

The TOE controls access to all directory entry objects based on the following security attributes:

- Entry Owner Information
  - entryOwner – identifying the DN of the owner of the entry
  - ownerPropagate – specifying the inheritance of ownership should in case no entry owner is specified in descendants
- Access Control Information (ACI)

- Non filter based
  - aclEntry – specifying the access control for the non filter-based ACI
  - aclPropagate – specifying the inheritance of access control rights in case no ACI is specified in descendants
- filter based
  - ibm-filterAclEntry – specifying the access control for the filter-based ACI
  - ibm-filterAclInherit – specifying the inheritance of access control rights in case no ACI is specified in descendants
- Groups and roles

### 6.1.3 F.I&A

#### 6.1.3.1 User Authentication (F.I&A.1)

In order to give the users access rights to other than operations on public data, the TOE requires each user to be successfully identified and authenticated. The subject's DN is established by the identification and authentication using a bind operation. The TOE uses a simple user ID / password authentication method to authenticate users as defined in RFC 2251. Users are maintained in the directory.

The behavior of the authentication mechanism is controlled by the password policy specified by the Directory Administrator (see F.MANAGEMENT.2 below).

The administrator can also specify that a user has to change his password after it has been initially set or after it has been reset by the administrator. The administrator can also define the number of consecutive failed authentication attempts after which one of the following happens:

- any further authentication attempt is prohibited for a time period defined by the administrator
- or
- the user will not be able to log in until the administrator has reset the user's password

### 6.1.4 F.MANAGEMENT

The TOE allows Directory Administrators to manage the behavior of the following functions:

- Authentication function
- Authorization function
- Audit function

The TOE allows Directory Administrators to configure the following security attributes:

- Password policy (Authentication function)
- Entry Owner Information (Authorization function)
- Access Control Information (Authorization function)
- Audit options (Audit function)

#### 6.1.4.1 Roles (F.MANAGEMENT.1)

The TOE supports two security roles: Directory Administrator and End User.

Both roles are defined within the TOE. The Directory Administrator has the ability to define groups and other “roles” to assist in the management of access rights and privileges. Those administrator defined groups and roles are not considered to be roles in the sense of the CC requirement FMT\_SMR.1 but are just ways to manage access rights more easily.

The End Users have no administrative rights.

#### **6.1.4.2 Authentication Function (F.MANAGEMENT.2)**

The TOE has the following password criteria, called the password policy, that can be specified by the Directory Administrator to enhance the quality of the passwords selected by the users:

- Minimum length of passwords
- Minimum number of alphabetic characters
- Minimum number of non-alphabetic characters
- Maximum number of repeated characters
- Maximum lifetime of a password
- Minimum lifetime of a password

The Directory Administrator can also specify that a user has to change his password after it has been initially set or after it has been reset by the administrator. The Directory Administrator can also define the number of consecutive failed authentication attempts after which one of the following happens:

- any further authentication attempt is prohibited for a time period defined by the administrator
- or
- the user will not be able to log in until the administrator has reset the user's password

The Directory Administrator can also specify if a user is allowed to change his own password.

The following section defines the password security policy settings that are considered for the evaluated configuration:

- pwdCheckSyntax – 1 (activate the password checking mechanism)
- Minimum length of passwords (pwdMinLength) – default for secure configuration is 8 characters
- Minimum number of alphabetic characters (passwordminalphachars) – default for secure configuration is 4
- Maximum number of repeated characters (passwordmaxrepeatedchars) – default for secure configuration is 2
- Minimum number of non-alphabetic characters (passwordMinOtherChars) – default for secure configuration is 2
- Maximum lifetime of a password (pwdMaxAge) – default for secure configuration is 90 days (7776000 seconds)
- Minimum lifetime of a password (pwdMinAge) – default for secure configuration is 1 day (86400 seconds)

- Maximum number of consecutive failed login attempts (PwdMaxFailure) – default for secure configuration is 3
- Action to be taken when PwdMaxFailure is reached or exceeded – default for secure configuration is: user can not log in until the administrator has reset the password (pwdLockout – TRUE and pwdLockoutDuration – 0)
- User must change his password after initialization and after reset by the Directory Administrator (pwdMustChange – TRUE)
- Users are allowed to change their own password (pwdAllowUserChange – TRUE)
- pwdSafeModify – TRUE
- ibm-pwdpolicy – TRUE

#### 6.1.4.3 Authorization Functions (F.MANAGEMENT.3)

With the exception of the user password entry and the system attributes, the End User being the entry owner or the Directory Administrator have full access rights for an entry and are able to use the authorization function to modify the authorization information on an entry.

The following functions for management of security attributes are available:

- **Entry owner information** – the entry owner information (entryOwner) of an entry can be set by the End User being the entry owner and by the Directory Administrator. This means that the entry owner can give away an entry to any other user. The entry owner information is either inherited from an ancestor or directly specified for each entry.
- **Access Control Information (ACI)** – the access control information can be specified by the End User being the entry owner and by the Directory Administrator. This means that these users can specify explicit access rights by specifying the access control mode and the associated attributes. These are:
  - **Non filter-based ACL** – containing the aclEntry defining the access control information and aclPropagate indicating whether to propagate the ACL information to descendants.
  - **Filter-based ACL** – containing the ibm-filterAclEntry defining the filter-based access control information, and aclPropagate indicating whether to propagate the ACL information to descendants.

There are attributes, so called system attributes that can only be changed by the system itself, and not be changed by neither any the End User nor by the Directory Administrator. E.g., the pwdChagedTime, specifying the last time the user's password was changed.

Changes to the user password entries are not only subject to the access control, as described above, but in addition subject the constrains of the password policy.

#### 6.1.4.4 Audit Function (F.MANAGEMENT.4)

The audit management is divided into management of the audit function and audit file management. The audit management is restricted to the Directory Administrator.

##### *Audit function management*

The directory administrator can perform the following management functions:

- The audit subsystem can be enabled and disabled.
- The audit file name can be specified.



- The auditable events can separately be activated or deactivated for the following events: bind, unbind, search, add, modify, delete, modifyDN (including modifyRDN), the event notification extended operation: registration and unregistration
- Select that only operations that failed are to be audited, or both failed and successful operations.

Note: The modifyDN and modifyRDN are indistinguishable by the audit function such that they will both be audited as modifyDN and also cannot be configured individually.

#### *Audit file management*

There are three different extended operations available to the Directory Administrator for managing the log files. With the extended operation LDAP requests, the audit file can be queried, read or deleted as follows:

- **Request number of lines** – inquiry of how many lines are in the log file
- **View request** – request to view a subset of the log file
- **Clear request** – request to clear the log file

The LDAP server reads the lines of the log file and truncates a line down to 400 characters in case the line is larger before sending it to the client. This is set to prevent the server from sending endless lines.

Each of the extended operation must contain at least the type of the requested log file.

### **6.1.5 F.REF\_MEDIATION**

The TOE is designed such that all security policy enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed. Any request for access to a directory entry the TOE receives is checked for access according to the rules defined for the TOE.

### **6.1.6 TOE Security Functions rationale**

For a justification of the TOE security functions to meet the security functional requirements is provided in TOE Security Functional Rationale, section 8.3.1.

## **6.2 Assurance Measures**

The assurance requirements for EAL2 are met by this TOE by the following assurance measures. These assurance requirements provide, primarily via review of supplied evidence, independent confirmation that these actions have been competently performed. They also include the following independent, third-party analysis:

- a) Confirmation of effective configuration management
- b) Confirmation of product delivery and installation procedures
- c) Confirmation that the guidance documentation is adequate
- d) Verification of a sample of the vendor functional testing
- e) Verification of the developers analysis for vulnerabilities and resistance against obvious penetration attacks
- f) Independent functional testing

To define the assurance measures claimed to satisfy the security assurance requirements specified in Section 5.2, a mapping is provided between the Assurance Requirements and the Assurance Measures, which are intended to satisfy the Assurance Requirements. As

shown in Table 3, the Assurance Measures are provided in the form of references to the relevant and appropriate document associated with each requirement.

**Table 3: Assurance Measures**

<b>CC Assurance Component</b>	<b>Assurance Measure</b>
ACM_CAP.2 Configuration items	M.CAP Configuration Management Version Control. CMVC and Lotus team rooms are used to manage configuration items [ConfMgt].
ADO_DEL.1 Delivery procedures	M.DEL The software is delivered via the Internet using a secure download mechanism as defined in the document Download Director Applet Function Specification [TQA].
ADO_IGS.1 Installation, generation, and start-up procedures	M.IGS This process is described in the document: IBM Directory Server, Installation and Configuration Guide Version 5.1 [InstGuide] and in the IBM Directory Server, Security Guide [SG].
ADV_FSP.1 Informal functional specification	M.FSP The functional specification is provided in the document IBM Directory Server Version 5.1 Functional Specification [FSP]
ADV_HLD.1 Descriptive high-level design	M.HLD The high level design is defined in the document IBM Directory Server Version 5.1 High Level Design [HLD].
ADV_RCR.1 Informal correspondence demonstration	M.RCR Correspondence demonstration is provided in a separate document [RCR] that maps the TOE Summary Specification to the Functional Specification and the Functional Specification to the High Level Design
AGD_ADM.1 Administrator guidance	M.ADM IBM Directory Server Version 5.1: Administration Guide [AdminGuide]
AGD_USR.1 User guidance	M.USR IBM Directory Server C-Client SDK Programming Reference Version 5.1 [UserGuide]
ATE_COV.1 Evidence of coverage	M.COV Test coverage is provided as tables that map test cases to the functional specification [TestPlan]
ATE_FUN.1 Functional testing	M.FUN [AUDCERT], [PWPCERT], [ACLTest], IBM Directory Server Version 5.1 – Secure Mode Test Plan [TestPlan]
ATE_IND.2 Independent testing – sample	M.IND Independent testing will be performed by the evaluation facility
AVA_SOF.1 Strength of TOE security function evaluation	M.SOF A Strength of Functional analysis is provided for the password mechanism [SOF]
AVA_VLA.1 Developer vulnerability analysis	M.VLA A Vulnerability Analysis is provided in a separate document. [VA]

## **7 Protection Profile Claims**

### **7.1 PP Reference**

This Security Target does not claim conformance with any Protection Profile that has been registered and / or evaluated.

## 8 Rationale

### 8.1 Security Objectives Rationale

Table 4 provides a mapping of TOE security objectives to threats and assumptions. Then it is followed by rationale of how each threat, assumption and organizational security policy is addressed by the corresponding security objectives.

**Table 4: Mapping of Security Objectives to Threats, Assumptions and Policies**

	T.ENTRY	T.ACCESS	T.ACCOUNT	T.BYPASS	TE.SOPHISTICATED	TE.USAGE	TE.CRASH	TE.PASS	A.PHYSICAL	A.NOEVIL	A.ADMIN	A.COMM	A.COOP	A.TIME	P.PUBLIC
O.AUTHENTICATE	X		X												
O.AUTHORIZE		X													X
O.ACCOUNT			X												
O.BYPASS				X											
OE.MANAGE						X				X	X		X		
OE.OS-MANAGE						X		X		X	X				
OE.PHYSICAL								X	X						
OE.DATABASE		X				X		X		X	X				
OE.SOPHISTICATED					X										
OE.BACKUP							X								
OE.COMMUNICATION												X			
OE.TIME														X	

T.ENTRY

O.AUTHENTICATE ensures that all users are identified and authenticated before being granted access to TOE mediated resources except for allowing unauthenticated users to perform some operations on public data, configured by administrators. It prevents unauthenticated users from access to TOE resources and services.

T.ACCESS

O.AUTHORIZE provides the capability to specify and manage access rights to TOE resources and services. Thus it prevents any user from access to data or performing operations without proper permissions.

Unauthorized access during communication and while stored in the external database needs to be ensured by measures in the TOE environment and are addressed by the objectives OE.DATABASE and OE.COMMUNICATION.

T.ACCOUNT	<p>O.AUTHENTICATE ensures that all users are identified and authenticated before being granted access to the TOE mediated resources except for allowing unauthenticated users to perform some operations on public data. Such limited access to the TOE is configured by Directory Administrators and should be compliant with the security policy of the organization responsible for the operation of the TOE. Based on the identity information, O.ACCOUNT enforces that any security related events can be further associated with those accountable for such activities.</p> <p>Note that unauthenticated users and anonymous users are granted limited access to public data. For those audit entries for activities performed by unauthenticated users or anonymous users, no identity information is kept in such entries.</p> <p>The two objectives together prevent security relevant actions from occurring without traceability of those accountable for such actions.</p>
T.BYPASS	<p>O.BYPASS enforces that the TOE security policy enforcement functions must always be invoked and succeed before access to TOE objects and services are allowed. It prevents the user from circumventing the TOE security functions.</p>
TE.SOPHISTICATED	<p>OE. SOPHISTICATED ensures that the TOE environment have sufficient capabilities to counter the threat of illegal users gaining unauthorized access via sophisticated attacking tools applied to the underlying system.</p>
TE.USAGE	<p>OE.MANAGE and OE.OS-MANAGE require that those responsible for the TOE must ensure that TOE is delivered, installed, operated, and administered in a secure manner. They address the threat introduced by misuse, misconfiguration, and mismanagement.</p> <p>This threat is also addressed by OE.DATABASE, which requires proper configuration and operation of the database system, which is part of the TOE environment.</p>
TE.CRASH	<p>OE.BACKUP requires that TOE environment have the backup services. Hence, upon human error, software/hardware failure, etc. which result in data loss or corruption, the system is still able to restore to a previous secure state.</p>
TE.PASS	<p>OE.PHYSICAL, OE.OS-MANAGE and OE.DATABASE requires that the hardware and software is physically protected, that the underlying operating system and hardware is configured and managed in a secure manner and that the database is configured and managed in a secure way, preventing the bypassing of the TOE security functions.</p>
A.PHYSICAL	<p>OE.PHYSICAL requires that TOE and its underlying hardware and software are physically protected from unauthorized physical modification and from technical attacks at the hardware and operating system level. Hence, this assumption is maintained.</p>
A.NOEVIL	<p>OE.MANAGE, OE.OS-MANAGE and OE.DATABASE require that TOE and the TOE environment is managed and administered in a secure manner. Assuming that the administrators should be trusted to perform discretionary actions in accordance with security policies.</p>

A.ADMIN	OE.MANAGE, OE.OS-MANAGE and OE.DATABASE requires that TOE and the underlying OS and HW, and database is managed and administered in a secure manner, which implies that TOE and TOE environment are competently installed and administered.
A.COMM	OE.COMMUNICATION requires that communication links between the TOE and LDAP clients on external systems are protected to against unauthorized modification and disclosure of communication data.
A.COOP	OE.MANAGE requires that the TOE is managed in a secure manner to maintain the security of the TSF data and user data. Including the protection of user passwords and setting of the access control rights under the control of the individual users.
A.TIME	OE.TIME requires that the TOE environment provides a reliable time.
P.PUBLIC	O.AUTHORIZE provides the capability to specify and manage the access rights to TOE resources and services. Thus, preventing users from access to data or performing operations without proper permissions by only making public information available to any user.

## 8.2 Security Requirements Rationale

### 8.2.1 Security Functional Requirements Rationale

Table 5 shows a mapping of Security Functional Requirements to TOE Security Objectives and Security Objectives for the TOE environment. Following such a mapping, further discussion is given on how each Security Objective is addressed by the corresponding Security Functional Requirements.

The *italic text* used in the table represents those functional components that are met by the TOE environment.

**Table 5: Mapping of Security Functional Requirements to TOE Security Objectives**

	O.AUTHENTICATE	O.AUTHORIZE	O.ACCOUNT	O.BYPASS	OE.MANAGE	OE.OS-MANAGE	OE.PHYSICAL	OE.DATABASE	OE.SOPHISTICATED	OE.BACKUP	OE.COMMUNICATION	OE.TIME
FAU_GEN.1			X									
FAU_GEN.2			X									
FAU_SAR.1			X									
FAU_SAR.2			X									
FDP_ACC.2		X										
FDP_ACF.1		X										
FIA_AFL.1	X											

FIA_ATD.1		X										
FIA_SOS.1	X											
FIA_UAU.1	X											
FIA_UID.1	X											
FMT_MOF.1a	X	X										
FMT_MOF.1b			X									
FMT_MSA.1		X										
FMT_MSA.3		X										
FMT_MTD.1		X										
FMT_SMF.1	X	X	X									
FMT_SMR.1		X										
FPT_RVM.1				X								
<i>FPT_STM.1</i>												X

O.AUTHENTICATE

FIA\_AFL.1 ensures that an attacker does not have an unlimited number of authentication attempts he could use to guess an end user's password.

FIA\_UID.1 ensures that, except that unauthenticated users and anonymous users are allowed access to public information and services configured by Directory Administrators in accordance to security policies, each user is successfully identified before allowing any TSF-mediated actions for that user.

FIA\_UAU.1 ensures that, except that unauthenticated users and anonymous users are allowed access to public information and services configured by Directory Administrators in accordance to security policies, each user is successfully authenticated before allowing any TSF-mediated actions for that user.

FIA\_SOS.1 ensures that password rules, for password based identification and authenticated, are enforced against all end users, preventing the bypassing or circumventing security policies.

FMT\_MOF.1a ensures that the authentication function is managed by the Directory Administrator to enforce the appropriate password rules.

FMT\_SMF.1 ensures that the TSF is capable of performing management of the authentication function by password management and password policy management.

Such security requirements work together to ensure successful identification and authentication prior to any TSF-mediated actions for each user.

O.AUTHORIZE

FDP\_ACC.2 ensures that complete access control is enforced on access to TOE resources and services.

FDP\_ACF.1 ensures that the access control security policy is actually implemented by relevant security functions, based on user security attributes.

FIA\_ATD.1 ensures that user security attributes are maintained and managed by the TOE to provide supports for access control.

FMT\_MOF.1a ensures that the TSF behavior is administered and managed by Directory Administrators, so that any change to it is restricted to authorized users.

FMT\_MSA.1 ensures that the TOE security attributes can only be administered and managed by the Directory Administrators authorized users.

FMT\_MSA.3 ensures that TOE access control is enforced to restrict the capability to specify default security attributes to authorized users.

FMT\_MTD.1 ensures that access to TSF data is restricted to the Directory Administrator only.

FMT\_SMF.1 ensures that the TSF is capable of performing management of the users and of the access control.

FMT\_SMR.1 ensures that roles/groups are maintained by the TOE and can be associated with users to facilitate the access control.

Such security requirements work together to ensure full control and management of user, data, and services, providing authorized user access to resources and functionality.

#### O.ACCOUNT

FAU\_GEN.1 ensures that audit log of security related activity and events are recorded.

FAU\_GEN.2, which is a security requirement on TOE environment, ensures that each audit event can be associated with the identity of the user that caused the event so that the user can be held accountable for security related action, except for unauthenticated user and anonymous user. However, as access to resources and services by them is defined by Directory Administrators in accordance to security policy, it doesn't pose threats to TOE.

FPT\_STM.1, which is a security requirement on TOE environment, ensures that each audit entry is associated with reliable time stamp. Note that this time stamp is taken from the underlying operating system, which is part of the TOE environment.

FAU\_SAR.1 provides the user with the capability to review audit log.

FAU\_SAR.2 restricts the access to the audit log to users of Directory Administrator role. It further prevents unauthorized users from manipulating the audit log and preserves the integrity audit log.

FMT\_MOF.1b ensures that the behavior of the audit function is managed by the Directory Administrator to enforce accountability. Such security requirements, as a whole, ensure that users can be held accountable for their actions.

FMT\_SMF.1 ensures that the TSF is capable of performing management of the audit function.

#### O.BYPASS

FPT\_RVM.1 ensures that security policy enforcement functions are invoked and succeed before each function is allowed to proceed so the access control is always enforced.

These security requirements work together to prevent from



bypassing and circumvention of TOE security policy. Note that, although unauthenticated user and anonymous user have limited access to TOE resources and services, such access is defined by Directory Administrators are under control by TOE security policy, hence it is not regarded as circumvention of security policy.

OE.MANAGE	No dependency to any SFR.
OE.OS-MANAGE	No dependency to any SFR.
OE.PHYSICAL	No dependency to any SFR.
OE.DATABASE	No dependency to any SFR.
OE.SOPHISTICATED	No dependency to any SFR.
OE.BACKUP	No dependency to any SFR.
OE.COMMUNICATIO N	No dependency to any SFR.
OE.TIME	FPT_STM.1 ensures that a reliable timestamp is provided by the TOE environment.

**8.2.2 Dependency analysis**

Table 6 demonstrates that the security functional requirements meet all functional dependencies. The *italic text* used in the table represents those functional components that are met by the TOE environment.

As the assurance components are those standard ones for EAL2, all dependencies for such EAL2 assurance components are satisfied automatically.

**Table 6: Dependency Mapping of Security Functional Requirements**

<b>Component</b>	<b>Name</b>	<b>Dependencies</b>
FAU_GEN.1	Audit data generation	FPT_STM.1
FAU_GEN.2	User identity generation	FAU_GEN.1, FIA_UID.1
FAU_SAR.1	Audit review	FAU_GEN.1
FAU_SAR.2	Restricted audit review	FAU_SAR.1
<i>FPT_STM.1</i>	Time stamps	—
FDP_ACC.2	Complete access control	FDP_ACF.1
FDP_ACF.1	Security attribute based access control	FDP_ACC.1, FMT_MSA.3
FIA_AFL.1	Authentication failure handling	FIA_UAU.1
FIA_ATD.1	User attribute definition	—
FIA_SOS.1	Selection of secrets	—
FIA_UAU.1	Timing of authentication	FIA_UID.1
FIA_UID.1	Timing of identification	—
FMT_MOF.1a	Management of security functions behavior	FMT_SMF.1, FMT_SMR.1
FMT_MOF.1b	Management of security functions behavior	FMT_SMF.1, FMT_SMR.1
FMT_MSA.1	Management of security attributes	FDP_ACC.1 [or FDP_IFC.1] FMT_SMF.1, FMT_SMR.1

Component	Name	Dependencies
FMT_MSA.3	Static attribute initialization	FMT_MSA.1, FMT_SMR.1
FMT_MTD.1	Management of TSF data	FMT_SMF.1, FMT_SMR.1
FMT_SMF.1	Specification of Management Functions	—
FMT_SMR.1	Security roles	FIA_UID.1
FPT_RVM.1	Non-bypassability of the TSP	—

### 8.2.3 Demonstration of mutual support between security requirements

The dependency analysis provided in the previous section has shown how supportive dependencies between SFRs are satisfied. This section further demonstrates that the SFRs are mutually supportive by highlighting and discussing the additional supportive dependencies, which ensures that security policies are enforced.

FPT\_RVM.1 ensures that SFRs cannot be bypassed.

FIA\_UAU.1 provides additional protection as it ensures that access to the TOE cannot be made by impersonate as a different user. FIA\_UID.1 ensures that no security mediated functions can be initiated on behalf of a user until the user is uniquely identified to the TOE, except that limited access (to public information) is granted to unauthenticated user and anonymous user prior to identification and authentication. However, such restricted access is defined by Directory Administrators in accordance to the organizational security policy. Then, it poses no threats to the TOE.

FMT\_SMR.1 enforces which roles users may take in the TOE and the conditions associated with assuming the role. Based on FMT\_SMR.1, FMT\_MOF.1a FMT\_MOF.1b restricts the ability of users under specific roles to modify the behavior of functions that control security attributes or configuration data. With the aid of FMT\_SMR.1, FMT\_MSA.1 restricts the ability to modify security attributes or configuration data, protecting against tampering attacks through unauthorized modification of data. FMT\_MSA.3 restricts default security attributes or configuration data controlled under FMT\_MOF.1a, FMT\_MOF.1b and FMT\_MSA.1.

FMT\_MTD.1 restricts the ability to modify any other security relevant data, protecting against tampering attacks through unauthorized modification of data.

FMT\_FMF.1 specifies the management functions for the security behavior of the authentication function (FMT\_MOF.1a) and for the password and password policy (FMT\_MSA.1). FMT\_FMF.1 also specifies the management functions for the user and access control (FMT\_MSA.1 and FMT\_MTD.1), as well as for the management of the audit function (FMT\_MSA.1).

FDP\_ACF.1 controls rules governing user access to objects based on security attribute values and supports FDP\_ACC.2. FDP\_ACC.2 provides complete access control and enforces access controls on subjects and objects and all operations among the subjects and objects.

FIA\_SOS.1 reduces the likelihood of successful direct attack aimed at the identification and authentication functions, and thus supports FIA\_UAU.1.

FPT\_STM.1 support time entries in audit records for FAU\_GEN.1, as provided by the TOE environment. The FPT\_STM.1 is not subject to any administration by the TOE administrator.

FAU\_GEN.1 provides the ability to track the security related events and actions. FAU\_GEN.2 ensures that the individual responsible for generating an audit event is uniquely and unambiguously identified along with the audit data. FAU\_SAR.1 and FAU\_SAR.2 ensure that authorized users have the capability to review data from the audit records.

**8.2.4 Appropriateness of assurance requirements**

The TOE is supposed to thwart attackers of limited resources. The EAL2 assurance requirements bring enough assurance elements for the TOE, operating within its environment as described in this document. Furthermore, the EAL2 assurance level is technically feasible and achievable based on the requirements on life cycle support, development documents, secure delivery procedure, and configuration management. It is appropriate to satisfy users' expectations.

**8.3 TOE Summary Specification Rationale**

The TOE summary specification rationale is intended to show that the TOE security functions and assurance measures are suitable to meet the TOE security (functional and assurance) requirements.

To show that the selection of TOE security functions and assurance measures are suitable to meet TOE security requirements (functional and assurance), it is important to demonstrate the following:

- The specified TOE IT security functions work together so as to satisfy the TOE security functional requirements.
- That the started assurance measures are compliant with the assurance requirements.

**8.3.1 TOE Security Functions rationale**

This section is intended to provide a demonstration that the TOE security functions satisfy all TOE SFRs included in the ST. This is accomplished by mapping the TOE security functions onto the TOE SFRs by Table 7, which shows that:

- Each TOE SFR is mapped onto at least one TOE security function, and
- Each TOE security function is mapped onto at least one TOE SFR.

Note that FPT\_STM.1 is a TOE environment security functional requirement and is to be satisfied by the TOE environment.

**Table 7: Mapping of Security Functions to Security Functional Requirements**

	FAU_GEN.1	FAU_GEN.2	FAU_SAR.1	FAU_SAR.2	FDP_ACC.2	FDP_ACF.1	FIA_UAU.1	FIA_AFL.1	FIA_UID.1	FIA_ATD.1	FIA_SOS.1	FMT_MOF.1a	FMT_MOF.1b	FMT_MSA.1	FMT_MSA.3	FMT_MTD.1	FMT_SMF.1	FMT_SMR.1	FPT_RVM.1
F.AUDIT	X	X																	
F.ACCESS_CONTROL					X	X													
F.I&A		X					X	X	X	X	X								
F.MANAGEMENT			X	X								X	X	X	X	X	X	X	
F.REV_MEDIATION																			X

Table 7 shows that all the TOE security functional requirements are addressed by the TOE security functions. Below is described how the IT security functions of the TOE are suitable to meet the TOE security functional requirements.

FAU\_GEN.1      The requirement for generation of audit events is satisfied by the security functions F.AUDIT, which will generate audit records for the specified

events to an audit file. The audit record format and events are described as part of F.AUDIT. The audit record containing the timestamp is produced by F.AUDIT.

(A reliable time-stamp is provided by the environment as expressed by the IT requirement for the TOE environment FPT\_STM.1)

FAU_GEN.2	The requirement is satisfied by the association of user identities (F.I&A) with audit events (F.AUDIT).
FAU_SAR.1	The ability to read all audit records is provided by the security function F.MANAGEMENT, allowing only the Directory Administrator to view the audit file.
FAU_SAR.2	The requirement is satisfied by the function F.MANAGEMENT, which is restricted only to Directory Administrators, preventing any other users to read the audit file.
FDP_ACC.2	The requirement for complete access control is satisfied by the function F.ACCESS_CONTROL. All objects (directory entries and attributes) are subject to access control of F.ACCESS_CONTROL.
FDP_ACF.1	The requirement is satisfied by the F.ACCESS_CONTROL, which enforces the directory access control SFP based on the entry owner information and the Access Control Information.
FIA_AFL.1	The requirement for authentication failure is satisfied by F.I&A which will prevent any further authentication attempts of the same user after a specific number of consecutively failed authentication attempts. The number of failed attempts is three for a secure configuration, but the Directory Administrator can specify any number.
FIA_ATD.1	The requirement for user attributes is being fulfilled by F.I&A. The user attribute information is used and maintained by the F.I&A.
FIA_SOS.1	The requirement for the verification of secrets is fulfilled by the user ID / password mechanisms being part of the security function F.I&A. The password policy constrains for a secure configuration are given. The Directory Administrator has the ability to specify different values.
FIA_UAU.1	The requirement is being met by F.I&A, which will only assign the user an identity (Distinguished Name) as part of a successful identification and authentication. Without a successful authentication the user will be regarded as unauthenticated and will only be given access to public information.
FIA_UID.1	The requirement is being met by F.I&A. Since the identification and authentication is performed as one operation, see the description above of how FIA_UAU.1 is being met by F.I&A.
FMT_MOF.1a	The requirement is being fulfilled by F.MANAGEMENT, allowing the Directory Administrator to modify the behaviour of the identification and authentication function, by specifying the password policy.
FMT_MOF.1b	The requirement is being fulfilled by F.MANAGEMENT, allowing the Directory Administrator to disable, enable and modify the behaviour of the audit function.
FMT_MSA.1	The requirement is being fulfilled by F.MANAGEMENT, allowing the Directory Administrator to modify, delete and read the security attributes of the disable, enable and modify the behaviour of the audit function.
FMT_MSA.3	The requirement for restrictive default values are being fulfilled by the

	security function F.MANAGEMENT by assigning restrictive values by installation and configuration made by the Directory Administrator.
FMT_MTD.1	The requirement for management of TSF data is being satisfied by the security function F.MANAGEMENT.
FMT_SMF.1	The requirement for the TSF to provide management functions is being satisfied by the security function F.MANAGEMENT.
FMT_SMR.1	The requirement for roles maintained by the TOE, the End User and the Directory Administrator is satisfied by the security function F.MANAGEMENT.
FPT_RVM.1	The requirement for non-bypassability of the TSP is satisfied by the security function F.REV_MEDIATION, ensuring that the access control functions are being invoked before access is given granted.

The table above shows how the security functions work together to satisfy the security functional requirements.

The requirements FAU\_GEN.1, FAU\_GEN.2, FAU\_SAR.1, FAU\_SAR.2 define the requirements for the audit system by specifying the audit events, in relation to the other security functional requirements. The association of each audit event with user identities is consistent with the use of the identification and authentication function (FIA\_UID.1 and FIA\_UAU.1), so that FAU\_GEN.2 has a basis for associating the events to user identities causing that event. In order to record the time and date of the events, FAU\_GEN.1 requires a reliable time-stamp, which is provided by FPT\_STM.1 (provided by the IT environment). The requirements FAU\_SAR.1 provides the Directory Administrator with the capability to read all the audit information generated by the audit function. FAU\_SAR.2 is preventing all other users from accessing the audit records.

The requirements for access control is defined by the discretionary access control policy in FDP\_ACC.2, providing complete access control based on the Entry Owner information and the Access Control Information and according to the evaluation order defined in FDP\_ACF.1.

The requirements for identification and authentication are addressed by FIA\_AFL.1, FIA\_ATD.1, FIA\_SOS.1, FIA\_UAU.1 and FIA\_UID.1. The timing of identification and authentication is specified by FIA\_UID.1 and FIA\_UAU.1. Authentication failures are handled by FIA\_AFL.1, by detecting and suspending users or prohibit further login after a certain number of failed consecutive login attempts. The attributes, including passwords, associated with individual users are specified in FIA\_ATD.1. Restrictions on the passwords used for user authentication is specified by FIA\_SOS.1, preventing the user to select weak or easy to guess passwords.

The management functions are specified by FMT\_SMF.1, specifying management functions for password management, password policy management, user management, access control management and audit management. The roles identified by the TOE is specified in FMT\_SMR.1.

The management of the authentication function is described in FMT\_MOF.1a and the management of passwords, password policy, entry owner information and access control information and audit options is covered by FMT\_MOF.1b. These management functions are also restricted to specific identified, authenticated and authorized users. For the discretionary access control, restricted attributes are specified by FMT\_MSA.3. The management of TSF data is authorized users as specified in FMT\_MTD.1.

FPT\_RVM.1 defines that the TSP enforcement functions are invoked and succeed before any other function is allowed to proceed, preventing by-passability of the TSP.

The ability of the TOE security functions to fulfill the security functional requirements is demonstrated by the internal consistency of the security functional requirements, shown in section 8.2.3, and by the demonstration that each of these security requirements are being satisfied with one or more TOE security functions in combination as explained above.

### 8.3.2 Minimum Strength of Function Level rationale

The TOE mechanisms will resist technical attacks by unauthorized users. The TOE mechanisms will also resist user errors, system errors, or non-malicious actions by authorized users. Resistance to sophisticated types of attacks, when such resistance is required, is provided by the TOE operational environment. The environment also assumes that those individuals who have authorized physical access to the TOE are trusted to not behave maliciously.

Consequently, a level of strength of function basic (**SOF-basic**) which indicates that a function provides adequate protection against straightforward or intentional breach of TOE security by attackers possessing a basic attack potential is consistent with the security objectives of the TOE. This claim applies to identification and authentication as specified in FIA\_SOS.1, using the described settings, and satisfied by F.I&A.

## 8.4 Assurance measures rationale

This section is to show that the identified assurance measures are appropriate to meet the assurance requirements by providing mapping between the identified assurance measures and the assurance requirements, as shown in Table 8.

In this case, the specification of assurance measures is done by reference to the appropriate document (e.g., Configuration Management Plan, System Architecture, User Guide, etc.). Rationale is provided to show that the referenced document (assurance measure) meets the requirements of the associated assurance requirement.

**Table 8: Mapping of Assurance Measures to Assurance Requirements**

	ACM_CAP.2	ADO_DEL.1	ADO_IGS.1	ADV_FSP.1	ADV_HLD.1	ADV_RCR.1	AGD_ADM.	AGD_USR.1	ATE_COV.1	ATE_FUN.1	ATE_IND.2	AVA_SOR.1	AVA_VLA.1
M.CAP	X												
M.DEL		X											
M.IGS			X										
M.FSP				X									
M.HLD					X								
M.RCR						X							
M.ADM							X						
M.USR								X					
M.COV									X				
M.FUN										X			
M.IND											X		
M.SOF												X	
M.VLA													X