HP OpenView Select Access

Security Target

Version 3.0

February 2006

Prepared by

Betrusted Limited
Level 4, Building C
33 Saunders Street
Pyrmont NSW 2009
Australia

www.betrusted.com

Document Control:

	Written by	Checked by	Approved by
Name	John Bludhorn	George Sarandrea	Gene Admur
Title	Security Consultant	Security Consultant	Hewlett-Packard
Document Reference			
PSG Document Reference Number: 03/272			
Document Classification			

Trademarks

"Select Access" is a trademark of Hewlett Packard Ltd.

"OpenView" is a trademark of Hewlett Packard Ltd.

Company Titles

Within this document, the following shortened forms of company titles may be used:

Hewlett Packard Limited – 'HP'

Copyright Statement

Reproduction of this document is authorised provided this document is reproduced in full, and Hewlett Packard is acknowledged as the owner and copyright holder of this document.

Contents

Tables

Figures

Conventions and Terminology

Conventions

The notation, formatting, and conventions used in this Security Target are consistent with those used in Version 2.1 of the Common Criteria [CC]. Selected presentation choices are discussed here to aid the Security Target reader. The CC allows several operations to be performed on functional and assurance requirements: The allowable operations defined in paragraph 2.1.4 of Part 2 of the CC [CC2] are *refinement, selection, assignment* and *iteration*.

- The assignment operation is used to assign a specific value to an unspecified parameter, such as the length of a password. An assignment operation is indicated by showing the value in square brackets, i.e. [assignment_value(s)].
- The refinement operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is denoted by **bold text**.
- The selection operation is picking one or more items from a list in order to narrow the scope of a component element. Selections are denoted by *<u>underlined italicised</u>* text.
- Iterated functional and assurance requirements are given unique identifiers by appending to the base requirement identifier from the CC an iteration number inside parenthesis, for example, FMT_MTD.1.1 (1) and FMT_MTD.1.1 (2) refer to separate instances of the FMT_MTD.1 security functional requirement component.

All operations described above are used in this Security Target. *Italicised text* is used for both official document titles and text meant to be emphasised more than plain text.

Terminology

The terminology used in the Security Target is that defined in the Common Criteria [CC1, CC2]. The following additional TOE specific terminology is included to assist the consumer of the Security Target:

Term	Description
Access Control	The process of limiting access to resources to only authorised users.
Administrator	The role responsible for installation and configuration of the TOE, and the management of all security functions. (See also, administrators and Delegated Administrator).
administrators	Used in lower case plural form, the term is used as a generic reference to both the Administrator and Delegated Administrator roles.
Applet	A restricted Java program, downloaded to a client system by commands embedded in a Web page, which executes within a Web browser under the control of a Java Runtime Environment (JRE) built-in to the browser.
Application Server	A J2EE application server
Authentication	The process of verifying an individual's right to access resources.
Authorisation	The process of granting or denying access to a network resource.
CC	Common Criteria
Delegated Administrator	The role delegated with responsibility for managing a sub-set of policy administration and user management.
Digital Signature	A digital code that is generated from a piece of data to assure source authenticity and integrity.
EAL	Evaluation Assurance Level
External Entity	a (non-administrative) person attempting to access the resources that the TOE controls – this person may be an attacker or an authorised user, depending on the context.
GUI	Graphical User Interface.
HTML	HyperText Markup Language
IT	Information Technology
J2EE	Java 2 platform, Enterprise Edition. J2EE is a platform-independent, Java-centric environment from Sun for developing, building and deploying Web-based enterprise applications online. The J2EE platform consists of a set of services, APIs, and protocols that provide the functionality for developing multi-tiered, Web-based applications. http://www.webopedia.com/TERM/J/J2EE.html
JRE	Java Runtime Environment, consisting of a Java Virtual Machine (JVM) and the core class libraries that define a complete environment within which Java applications execute.
PP	Protection Profile
Privilege	The ability to perform certain actions that would otherwise be prevented.
Resource	A resource is a network service or resource, such as a URL, dynamic page or portal link. A resource resides on a Web server or J2EE application server.
SF	Security Function
SFP	Security Function Policy
SOF	Strength of Function

Term	Description
SSL	Throughout this document, the term "SSL" without a version qualifier (see SSL v3.0) should be taken as a generic reference to either SSL v3.0 or TLS v1.0.
SSL v3.0	Secure Sockets Layer v3.0
ST	Security Target
TLS v1.0	Transport Layer Security v1.0
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSFI	TSF Interface
TSP	TOE Security Policy
URL	Uniform Resource Locator. The global address of documents and other resources on the World Wide Web. The first part of the address indicates what protocol to use, and the second part specifies the IP address or the domain name where the resource is located. http://www.webopedia.com/TERM/U/URL.html
XML	Extensible Markup Language. A specification developed by the W3C. XML is a pared-down version of SGML, designed especially for Web documents. It allows designers to create their own customized tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations. http://www.webopedia.com/TERM/X/XML.html
ISAPI	Internet Server API. An API for Microsoft's IIS (Internet Information Server) Web server. ISAPI enables programmers to develop Web-based applications that run much faster than conventional CGI programs because they're more tightly integrated with the Web server. In addition to IIS, several Web servers from companies other than Microsoft support ISAPI. http://www.webopedia.com/TERM/I/ISAPI.html
NSAPI	Netscape Server API. An API for Netscape's Web servers. NSAPI enables programmers to create Web-based applications that are more sophisticated and run much faster than applications based on CGI scripts. http://www.webopedia.com/TERM/N/NSAPI.html
HTTP	HyperText Transfer Protocol. The underlying protocol used by the World Wide Web. HTTP defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands. http://www.webopedia.com/TERM/H/HTTP.html
JDBC	Java Database Connectivity. A programming interface that lets developers using the Java programming language to gain access to a wide range of databases and other data sources, either directly or through middleware.
X.509	The most widely used standard for defining digital certificates. X.509 is actually an ITU Recommendation, which means that has not yet been officially defined or approved. As a result, companies have implemented the standard in different ways. http://www.webopedia.com/TERM/X/X_509.html

References

[3DES] Federal Information Processing Standard (FIPS) Publication 46-3, "Data Encryption Standard", National Institute of Standards and Technology, 25 October 1999.

[AES] Federal Information Processing Standard (FIPS) Publication 197, "Advanced Encryption Standard (AES)", National Institute of Standards and Technology, 26 November 2001.

[CC] Common Criteria for Information Technology Security Evaluation, Version 2.1, August 1999.

[CC1] Common Criteria Part 1: "Introduction and General Model, Version 2.1", CCIMB-99-031, August 1999.

[CC2] Common Criteria Part 2: "Security Functional Requirements, Version 2.1", CCIMB-99-032, August 1999.

[CC3] Common Criteria Part 3: "Security Assurance Requirements, Version 2.1", CCIMB-99-033, August 1999.

[DHE] RFC 2631, "Diffie-Hellman Key Agreement Method", Rescorla, E., June 1999.

[RSA] RFC 2437, "PKCS #1: RSA Cryptography Specifications Version 2.0", Kaliski, B. and J. Staddon, October 1998.

[SHA-1] Federal Information Processing Standard (FIPS) Publication 180-1, "Secure Hash Algorithm", National Institute of Standards and Technology, 17 April 1995.

[SSL3] "The SSL Protocol, Version 3.0", Freier, A., P. Karlton and P. Kocher, November 1995.

[TLS1] RFC 2246, "The TLS Protocol, Version 1.0", Dierks, T. and C. Allen, January 1999. SSLv3.0 is the Netscape implementation of TLSv1.0.

Document Organisation

Section 1 provides the introductory material for the Security Target.

Section 2 provides general purpose and TOE description.

Section 3 provides a discussion of the expected environment for the TOE. This section also defines the set of threats that are to be addressed by either the technical countermeasures implemented in the TOE hardware or software, or through the environmental controls.

Section 4 defines the security objectives for both the TOE and the TOE environment.

Section 5 contains the functional and assurance requirements derived from the Common Criteria, Part 2 and 3 [CC2, CC3], respectively that must be satisfied by the TOE.

Section 6 identifies the IT security functions provided by the TOE and also identifies the assurance measures targeted to meet the assurance requirements.

Section 7 makes any protection profile claims applicable to the TOE.

Section 8 provides a rationale to explicitly demonstrate that the information technology security objectives satisfy the policies and threats. Arguments are provided for the coverage of each policy and threat. The section then demonstrates how the set of requirements are complete, relative to the objectives, and that each security objective is addressed by one or more component requirements. Arguments are provided for the coverage of each objective. Next, Section 8 provides a set of arguments that address dependency analysis, strength of function issues, and the internal consistency and mutual supportiveness of the security target requirements.

1 Introduction

This introductory section presents *security target (ST)* identification information and an overview of the ST. A statement of Common Criteria conformance is also provided.

1.1 ST and TOE Identification

This section provides information needed to identify and control this ST and its Target of Evaluation (TOE). This ST targets an **Evaluation Assurance Level 2 (EAL2)** level of assurance for the TOE.

ST Title:	HP OpenView Select Access Security Target v3.0
TOE Identification:	HP OpenView Select Access v5.2 with Engineering Patch G, March 2004 HP Media Part Number T2593-15002
CC VERSION:	Common Criteria for Information Technology Security Evaluation, Version 2.1 Final.
ST Evaluation:	Australasian Information Security Evaluation Program, Defence Signals Directorate, Australian Department of Defence.
Author(s):	John Bluhdorn, George Sarandrea.
Keywords:	Access Control, Authentication, Authorisation.

***NOTE**: Wherever HP OpenView Select Access v5.2 is referenced it should now be considered to include Engineering Patch G as stated in the TOE Identification.

1.2 Security Target Overview

Select Access is an authorisation management solution, utilising an XML-based architecture that provides a Privilege Management Infrastructure (PMI), allowing the administration and enforcement of user privileges and transaction entitlements to enterprise resources in a distributed environment.

Select Access integrates with leading Web and Java2 Enterprise (J2EE) application servers. All Select Access policies are stored and accessed directly using LDAP to a range of directory servers.

Select Access provides a centralised user, resource and policy management capability. Authorisation rules may be defined down to the URL or transaction level. Authorisation decisions are based on dynamic role-based identities. A flexible policy inheritance scheme for users and resource groups increases scalability and reduces management overhead.

Select Access provides native password and profile management that supports multiple authentication techniques including passwords and X.509 certificates. Native password management facilitates definition and implementation of security policy for maintenance of passwords based on attributes including password length, required characters, user name match, dictionary match and password history. Secure session-based credentials are created and maintained by Select Access to allow users a "single sign-on like" capability.

To assist administrators with the potentially complex and time-consuming task of defining all network resources to be protected by Select Access, the resource discovery capability allows administrators to automatically discover and enumerate their networked resources.

A Java-based administration interface is provided to centrally administer and configure all of the distributed components of Select Access. Support is also provided for multi-level delegated administration through a Web-based interface to provide administrators with the capability to administer groups of users and/or resources for which they are responsible.

The highly secure, distributed architecture of Select Access provides for load-balancing and fail-over capabilities while enforcing strong authentication between all components.

The Select Access Secure Audit Server consolidates runtime and policy administration logs with digitally signed entries to provide for secure records of events. Select Access also provides reporting and alerting facilities. The Select Access Reporting facility uses the Secure Audit Server to enable organisations to define reporting procedures commensurate with their operational and audit policy. The Select Access Real Time Alerts allows custom alerts to be configured based on authorisation information, alert levels and alert handling instructions.

1.3 Common Criteria Conformance

The TOE is conformant with Part 2 of the CC, version 2.1 [CC2] and the assurance requirements of EAL 2 as defined in Part 3 of the CC, version 2.1 [CC3].

2 TOE Description

This section provides context for the TOE evaluation by identifying the product type and describing the evaluated configuration.

2.1 Overview of the TOE

Based on an open XML architecture, HP OpenView Select Access v5.2 (the TOE) is an easy-to-use and quick-to-deploy authorisation management software product. Select Access:

- Ensures the right people get access to the right online data and applications, and manages associated business; and
- Applies existing corporate policy throughout distributed network environments including extranets, intranets and portals.

Select Access has been designed with a directory-enabled modular architecture that allows fast deployment of an authorisation solution and can be centrally managed. The architecture of Select Access allows for a distributed system that incorporates load sharing and redundancy in many aspects of the solution. The runtime environment is a three-tiered architecture consisting of the directory server (policy repository), Validator (policy decision server), and Enforcer (application plug-in). Logs are consolidated to a Secure Audit Server. The Administrator and Delegated Administrators connect to the Administration Server using a Web browser to run the Policy Builder, a Java-based management Graphical User Interface (GUI).

Select Access is designed to be integrated into an existing security infrastructure. The Web server and its associated Enforcer plug-in will normally be located either behind a border or outer firewall, or in the DMZ (De-Militarised Zone) of that firewall, thus providing a basic degree of protection from the Internet. The remaining TOE components will normally be installed behind a second or inner firewall providing further protection for those components. Communications between the Enforcer and the Validator are encrypted and should be tunnelled through the inner firewall.

An example architecture including Select Access components is shown in Figure 1.

The logical components of the TOE (HP OpenView Select Access v5.2) include:

The Enforcer Plugin;

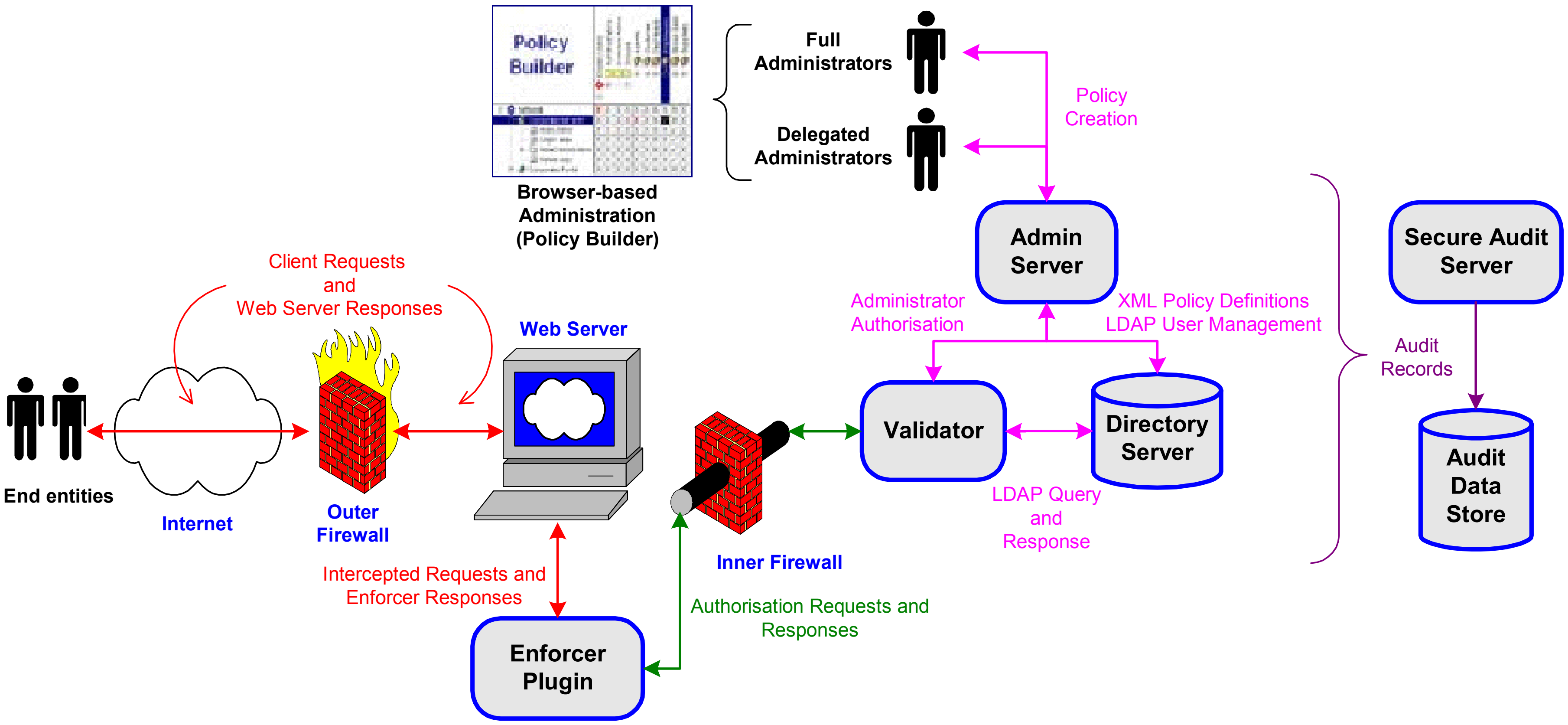
The Policy Validator;

The Administration Server;

The Policy Builder; and

The Secure Audit Server, and its associated Audit Data Store(s).

Figure 1: Select Access Component Architecture



2.1.1 Operational Summary

The Administrator and Delegated Administrators create and manage access and authorisation privileges using the Policy Builder. The Enforcer intercepts requests for access to network resources, querying the Validator to see if a given access or command is authorised. The Validator retrieves the relevant policy data from the directory server, evaluates the logic based on the information passed from the Enforcer, and returns the authorisation decision. The Enforcer then enforces the decision.

The Policy Builder also allows administrators to delegate management responsibilities to process and data owners. Management of both user profiles, and access and authorization privileges can be delegated in any combination. Delegated Administrators, using a Web-based interface, can configure policies and user information for those parts of the user and resource space they have been given permission to access. They can also sub-delegate portions of their control to others.

LDAP directory servers are used as data stores for user information, and for the set of network resources and policy information configured using the Policy Builder. Select Access can integrate with existing corporate directory servers, eliminating unnecessary data replication and mapping to the existing user schema.

Although they provide important storage function that is necessary for the TOE to operate, the LDAP servers do not provide any security-enforcing functionality within the TOE. Therefore, LDAP servers are not part of the TOE described in this Security Target.

The TOE components summarised above are covered here in more detail.

2.1.2 Enforcer TOE Component

The Enforcer is implemented as a plug-in to the Web server that manages access to network resources to be protected. A network resource could be a URL local to the Web server or an application server accessed through a Web server. The Enforcer intercepts information about incoming access requests and sends it as an XML document to the Validator for an authentication decision. The Enforcer is responsible for examining the XML-based response from the Validator and to enforce the authorisation decision.

The Enforcer provides the interface through which users must authenticate prior to being granted access to the requested resource. The Enforcer supports authentication via numerous methods, including passwords or X.509 certificates.

The Enforcer can be configured to operate with multiple Validators, with communications secured using SSL. If the Enforcer cannot receive a response to a validation request, it will deny access to the resource.

Since an Enforcer plug-in is configured on each of the servers providing access to managed resources, Enforcer redundancy and load sharing mirrors that of the servers themselves.

2.1.3 Validator TOE Component

The Validator is the component that determines whether or not access to a given resource is authorised. The Validator receives requests from one or more Enforcers in the form of an XML document. Based on this information, the Validator reads policies from the Policy Store held on the LDAP directory server and determines whether or not the access request is allowed. The result is passed back to the requesting Enforcer, which then enforces the decision. If the policies have been digitally signed, then the Validator will verify the policy signature.

The Validator performs validation of authentication information in support of authorisation decisions. Select Access provides built-in functionality to enable authentication services using mechanisms such as passwords and X.509 certificates.

The Validator supports SSL communications to all other components including the Enforcer and the LDAP directory server. Multiple Validators can be used to provide load balancing and redundancy within any given network configuration.

2.1.4 Administration Server TOE Component

The Administration Server includes a Web server to provide administrative access to the Select Access system using a Java-based browser applet, the Policy Builder. Administrative functionality provided by the Administration Server includes:

- Configuration of Select Access components via the Policy Builder applet;
- Management of SSL connections between the Administration Server and administrator Web browsers executing the Policy Builder applet.
- Management of certificates and SSL connections between Select Access components; and
- Management of policy data via Administration and Delegated Administration modes.

Due to the central co-ordination role played by the Administration Server, it is not feasible to operate more than one Administration Server within any given network configuration.

2.1.5 Policy Builder TOE Component

The Policy Builder is a signed Java-based GUI applet that operates through the administrator’s Web browser. It is used to configure aspects of the system, including authorisation policy, user management, networked resource data discovery, and delegation of management responsibilities. The Policy Builder transfers and stores its information in an LDAP-compliant directory server. Stored policies can be digitally signed and verified by the Policy Builder.

The Policy Builder supports secure SSL communications to the Administration Server.

Providing the Administration Server is available, multiple administrators can use the Policy Builder at the same time, with display refresh capabilities available to help ensure that the current state of the policy matrix is displayed.

Policy Builder - Administration and Delegated Administration Modes

The Policy Builder supports two operational modes - Administration or Delegated Administration.

Administration mode displays the complete version of the Policy Builder, which supports all policy management functions including add, modify or delete users, resources, rules, user access policies or user delegation privileges with respect to the Policy Matrix. Administration mode is only accessible to the Administrator.

Delegated Administration mode displays a limited version of the Policy Builder supporting management functions dependent upon the privileges granted to the Delegated Administrator. Delegated Administrators manage or sub-delegate the user and policy information over which they have been given authority. Delegated Administrators only see those parts of the user and resource space for which they have permission to manage – everything else is hidden. The Validator determines the Delegated Administrator's authority to access and manage user and policy privileges, in the same manner as it does for other resource access requests.

Policy Builder - Native Password and Profile Management

Select Access supports multiple authentication mechanisms including passwords and X.509 certificates. In addition to these facilities, Select Access provides the ability to define password policies based on selectable password attributes including:

- password length – minimum and/or maximum;
- required characters – alpha, numeric and/or special;
- user name match – must not contain user names or user ID;
- dictionary match – must not contain dictionary words; and
- password history – must not match user's last *x* number of passwords.

Select Access provides password policy management capabilities, which include:

- enforcement of password policies;
- providing a temporary password for new accounts;
- defining password expiration periods and changing passwords on a regular basis;
- locking accounts against which a number of failed password attempts have been made; and
- automatic reactivation of locked accounts after a set timeout period.

Policy Builder - Reporting

The Select Access Policy Builder provides a Reporting Engine that interacts with audit stores produced by the Secure Audit Server for defining reporting procedures commensurate with the operational and auditing policies of an organisation. The Reporting Engine enables administrators to create (view, save and print) reports specific to policy administration and authorisation which may be sorted and/or filtered on the basis of audit data including users, servers, administrators, dates and specific events.

The Reporting Engine verifies the digital signature of the audit records to ensure the validity of the audit data.

Policy Builder - Real Time Alerts

The Select Access Policy Builder enables custom alerts to be set, based on any authorisation information. Alerts may be set to one of five levels and configured so that each level activates a different alert handling instruction. Alert handling instructions may also be customised with respect to the recorded audit events including administrator operations, policy administration, system configuration, authorisation, caching, Validator queries and certificate operations. Alert handling instructions may include writing additional information to the audit records or sending e-mails to a list of recipients.

2.1.6 Secure Audit Server TOE Component

The Secure Audit Server provides a consolidated security audit trail. Audit entries are logged from the Select Access management components (the Administration Server, the Policy Builder) and the runtime environment (via the Policy Validator). All access requests, authorisation decisions, and administrative changes are logged. Log entries may be digitally signed to protect the integrity of audit data.

The Secure Audit Server outputs log data to multiple destinations and audit stores. These include JDBC-enabled databases, UNIX syslog, Windows Event Log, XML-formatted flat files and/or the standard error device. Different output destinations can be configured based on a combination of audit data attributes, such as audit component (i.e., administration activities, authorisation decision, access query) and/or event level (i.e., information, warning).

The Secure Audit Server supports SSL communications from all other components including the Enforcer Plugin, the Validator and the Administration Server.

Multiple Audit servers may be configured to provide load sharing, redundancy and hierarchical recording of audit information.

2.2 Physical Scope of the TOE

The physical scope of the TOE includes only software elements for either a Unix-based or a Windows-based environment as identified in Table 1. The first column identifies the physical components of the TOE (and associated logical components). Some TOE components may be located on the same machine, although this is not required or necessarily recommended.

The second column describes the platforms (software and hardware) that comprise the evaluated configuration.

Table 1: Components of the TOE and the Evaluated Configuration

Physical TOE Components	Supported Platforms
Policy Builder Applet and Administration Server Select Access Policy Builder Select Access Administration Server	**Windows Platform** Operating System: • Windows 2000 Server with SP2 or higher Recommended Hardware Configuration: • Intel Pentium 4, 1.2GHz processor • 256Mb RAM
Policy Validator Select Access Validator	**Unix Platform** Operating System: • Solaris 8 (2.8) patch 108940-07 or higher Hardware: • 440MHz UltraSPARC-III • 2Mb Cache • 512Mb RAM
	Windows Platform Operating System: • Windows 2000 Server with SP2 or higher Recommended Hardware Configuration: • Intel Pentium 4, 1.2GHz processor • 256Mb RAM
Enforcer plug-in Select Access Enforcer	**Unix Platform** Software: • iPlanet Web server 4.1 Operating System: • Solaris 8 (2.8) patch 108940-07 or higher Recommended Hardware Configuration: • 440MHz UltraSPARC-III • 2Mb Cache • 512Mb RAM

Physical TOE Components	Supported Platforms
	Windows Platform Software: • Microsoft IIS 5.0 with SP1 or higher Operating System: • Windows 2000 Server with SP2 or higher Recommended Hardware Configuration: • Intel Pentium 4, 1.2GHz processor • 256Mb RAM
Secure Audit Server Select Access Secure Audit Server	**Unix Platform** Operating System: • Solaris 8 (2.8) patch 108940-07 or higher Recommended Hardware Configuration: • 440MHz UltraSPARC-III • 2Mb Cache • 512Mb RAM
	Windows Platform Operating System: • Windows 2000 Server with SP2 or higher Recommended Hardware Configuration: • Intel Pentium 4, 1.2GHz processor • 256Mb RAM

2.3 Security Features

A summary of the security features offered by the TOE is described in Table 2.

Table 2: Summary of TOE Security Features

Service	Description
Validation of access requests	The TOE provides a policy validation and authorisation function through the Validator component.
Enforcement of access control	The TOE provides a policy enforcement function through the Enforcer component. The Enforcer component is the plug-in on a particular network resource. This resource can be a Web server or J2EE application server.
Authentication of users	The TOE provides support for authenticating users requesting access to resources using methods including passwords and X.509 certificates.
Native password management	The TOE provides facilities for the enforcement of password policies based on specified password attributes, changing of password at regular intervals and locking of accounts that have been the subject of suspicious activity.
Flexible policy definition	The TOE provides a Java-based GUI to define authorisation policies for users and resources. Users and resources are displayed on the axes of an expandable matrix, which allows administrators to "allow", "deny" or "conditionally" authorise access, by users, or groups of users, to controlled resources.
Delegation of policy-based administrative functions	The TOE allows the Administrator to delegate administration of authorisation policy to another trusted individual, a Delegated Administrator. That person can then administer the authorisation policy for a defined subset of users and/or resources. A Delegated Administrator can further delegate administration within their defined subset.
Secure audit collection and storage	The TOE provides a consolidated security audit trail. A specific TOE component, the Secure Audit Server, is dedicated to collecting, storing and protecting audit data. Audit entries are logged from the other TOE components. The level of logging can be configured by the Administrator both specifically for each TOE component, as well as at the Secure Audit Server for the whole TOE.
Reporting of audit data	The TOE provides a Reporting Engine which enables the definition of detailed reporting procedures commensurate with operational and audit policies. The Administrator and Delegated Administrators may create (view, save, print) reports on policy administration and authorisation based on specific sorting and/or filtering criteria. Audit data are verified by the Reporting Engine on the bases of the digital signatures of the audit records.
Alerting to audit events	The TOE provided the ability to set custom audit alert based on authorisation information, level of severity of the alert and the alert handing instruction.
Secure communications between distributed components	The TOE implements a secure subset of both the SSL v3.0 and TLS v1.0 protocols. TLS connections between TOE components use Ephemeral Diffie-Hellman key exchange, including RSA keys of at least 1024 bits and AES keys of 256 bits to provide confidentiality of data transmitted between components and authentication of the components. SSL connections between the Administration Server and administrator browsers use RSA key exchange, including RSA keys and 3DES keys to provide confidentiality of TOE administration data.

2.4 Features Outside of Scope

Select Access features outside the scope of the defined TOE Security Functions (TSF) and thus not evaluated are:

- The LDAP directory server;
- The Web server or J2EE application server protected by the Enforcer component;
- Audit storage device; e.g. NT event log or flat file;
- Select Access APIs for expanding or further tailoring the product
- Application, portal and wireless support;
- User self-registration;
- Storage of audit information other than as controlled by the Secure Audit Server;
- Authentication methods not specifically included;
- Data replication between redundant distributed components;
- High availability functions;
- User profile self management;
- Security Assertion Markup Language (SAML); and
- Any functions of SSL or TLS other than those implementing SSL v3.0/TLS v1.0 ciphersuites:
  - "SSL_RSA_WITH_3DES_EDE_CBC_SHA"
    (i.e.. algorithms other than 3DES and RSA, or key lengths less than 128 bits); or
  - "TLS_DHE_RSA_WITH_AES256_CBC_SHA"
    (i.e. algorithms other than AES and RSA, or key lengths other than 265 bits).

3 TOE Security Environment

In order to clarify the nature of the security problem that the TOE is intended to solve, this section describes the following:

- Any assumptions about the security aspects of the environment and/or of the manner in which the TOE is intended to be used.
- Any known or assumed threats to the assets against which specific protection within the TOE or its environment is required.
- Any organisational security policy statements or rules with which the TOE must comply.

3.1 Secure Usage Assumptions

The assumptions relating to the operation of the TOE are described in Table 3.

Table 3: Assumptions

Name	Description
A.ADMIN-DOCS	The Administrator will follow all policies and procedures described in the TOE system documentation to ensure secure administration of the TOE. Those delegated with administrative functions will follow all policies and procedures in the TOE systems documentation applicable for their delegated responsibilities to ensure secure administration of the TOE.
A.NO_EVIL	The Administrator, and those delegated with administrative functions, are assumed to be non-hostile and trusted to perform all their duties in a competent manner.
A.RESOURCES	The TOE will be used for authorising and authenticating users for granting or denying access to IT resources protected by the TOE, e.g. Web Pages.
A.PHYSICAL	It is assumed that strong physical security measures will be in place to prevent unauthorised physical access to all components of the TOE.
A.NETWORK	It is assumed that the TOE will be installed in a network that provides appropriate logical protection, against access or modification, to all components of the TOE.
A.CREDENTIALS	It is assumed that there exists an appropriate means of securely generating, distributing and managing good TOE user authentication credentials, for example operating the TOE within a securely managed PKI.

3.2 Threats to Security

Threats may be addressed either by the TOE or by its intended environment (for example, using personnel, physical, or administrative safeguards). These two classes of threats are discussed separately.

3.2.1 Threats Addressed by the TOE

The threats addressed by the TOE are described in Table 4.

Table 4: Threats Addressed by the TOE

Name	Description
T.NO_DETECT	An external entity may attempt to mount a network-based attack against the TOE security functions, which succeeds without detection.
T.NO_ACCESS	An external entity may attempt to gain network-based access to IT resources protected by the TOE for which they are not authorised.
T.IMPERSON	An external entity may attempt to impersonate an authorised user to gain access to the IT resources protected by the TOE.
T.CAPTURE	An external entity may attempt to capture plain data transmitted between components of the TOE.
T.INTEGRITY	The integrity of authentication and/or authorisation information may be compromised due to network transmission errors, application errors or malicious actions by external entities.
T.FAILURE	The IT resources protected by the TOE may be compromised due to the failure of one or more components of the TOE.

3.2.2 Threats Addressed by the Operating Environment

The TOE Operating Environment addresses the following threats listed in Table 5 below.

Table 5: Threats Addressed by the Operating Environment

Name	Description
TE.INSTALL	Those responsible for receiving and installing the TOE may unintentionally receive or install the TOE in a manner that undermines overall security.

3.3 Organisational Security Policies

Table 6 following describes the organisational security policies relevant to the operation of the TOE.

Table 6: Organisational Security Policies

Name	Description
P.AUTHORISE	Organisational policy must define the rules for granting or denying access, and requiring authenticated access, to the IT resources protected by the TOE.
P.AUDIT	Details of user activity will be recorded in an audit trail that must be preserved in accordance with relevant organisational archive requirements.
P.CRYPTO	All cryptographic material is to be the subject of physical and technical controls as defined in the relevant National Authority Standards.
P.TRAIN	All individuals who access any administrative functions of the TOE must receive training on the proper use of the functions and interactions of the TOE with supporting technologies.
P.ROLES	Organisational policy must define responsibilities for and assign individuals to the Administrator role. In addition, the organisational policy must define the mechanism for delegation of administrative duties.
P.NETWORK	The organisation's overall IT security policy must consider the placement and protection of the TOE within the context of maintaining organisational security.

4 Security Objectives

The security objectives are a concise statement of the intended response to the security problem. These objectives indicate, at a high level, how the security problem, as characterised in the "Security Environment" section of the ST, is to be addressed. Just as some threats are to be addressed by the TOE and others by its intended environment, so some security objectives are for the TOE and others are for its environment. These two classes of security objectives are discussed separately.

4.1 Security Objectives for the TOE

The security objectives for the TOE are as described in Table 7.

Table 7: Security Objectives for the TOE

Name	Description
O.AUTHORISE	The TOE will provide the means to grant or deny access to individuals or groups of individuals to IT resources protected by the TOE.
O.AUTHENTICATE	The TOE will provide the means to support authentication of individuals using multiple authentication mechanisms before granting access to IT resources protected by the TOE.
O.AUDIT	The TOE will provide the means of recording any security relevant events so as to assist a Administrator in the detection of potential attacks or misconfiguration of the TOE security functions.
O.ADMIN	The TOE will provide the means to enable the Administrator to effectively manage the TOE and its security functions. Further, the TOE will enable the delegation of subsets of administrative functions.
O.ENCRYPT	The TOE will provide the means to protect the confidentiality and integrity of data when transmitted between components of the TOE.
O.INTEGRITY	The TOE will provide the means to maintain the integrity of stored authentication and authorisation information.
O.FAIL_SAFE	The TOE will provide the means to ensure that access to the IT resources protected by the TOE is not granted in the event of a failure of a Validator or Enforcer.

4.2 Security Objectives for the Environment

The security objectives for the TOE environment are as described in Table 8.

Table 8: Security Objectives for the Environment

Name	Description
OE.NETWORK	Those responsible for the TOE shall ensure that procedures and/or mechanisms are in place to ensure that logical access to the TOE components is appropriately controlled.
OE.PHYS_ENV	Those responsible for the TOE shall ensure that procedures and/or mechanisms are in place to ensure that physical access to the TOE components is appropriately controlled.
OE.TRAIN	Those responsible for the security of the organisation shall provide initial and ongoing training for the Administrator and those delegated with administrative responsibilities. This training should include security awareness of vulnerabilities and familiarisation with supporting technologies. In addition, those responsible for the security of the organisation shall ensure that all appropriate background checks, psychological assessments, and security clearances, as required, are conducted for the Administrator and those delegated with administrative responsibilities.
OE.CREDENTIALS	Those responsible for the TOE shall ensure that an appropriate means of securely generating, distributing and managing TOE user authentication credentials exists in the TOE environment, for example operating the TOE within a securely managed PKI.

5 IT Security Requirements

5.1 TOE Security Functional Requirements

This section contains the functional requirements for the TOE. The functional requirements are listed in summary form in Table 9, below.

Table 9: TOE Security Functional Requirements

No.	Component	Component Name
Class FAU: Audit		
1	FAU_ARP.1	Security alarms
2	FAU_GEN.1	Audit data generation
3	FAU_GEN.2	User identity association
4	FAU_SAA.1	Audit review
5	FAU_SAR.1	Audit review
6	FAU_SAR.3	Selectable audit review
7	FAU_SEL.1	Selective audit
8	FAU_STG.1	Protected audit trail storage
Class FCS: Cryptographic support		
9	FCS_CKM.1	Cryptographic key generation
10	FCS_CKM.2	Cryptographic key distribution
11	FCS_CKM.4	Cryptographic key destruction
12	FCS_COP.1	Cryptographic operation
Class FDP: User data protection		
13	FDP_ACC.2	Complete access control
14	FDP_ACF.1	Security attribute based access control
Class FIA: Identification and authentication		
15	FIA_AFL.1	Authentication failure handling
16	FIA_UAU.1	Timing of authentication
17	FIA_UAU.5	Multiple authentication mechanisms
18	FIA_UAU.6	Re-authenticating
19	FIA_UID.2	User identification before any action
Class FMT: Security management		
20	FMT_MOF.1	Management of security functions behaviour
21	FMT_MSA.1	Management of security attributes
22	FMT_MSA.2	Secure security attributes
23	FMT_MSA.3	Static attribute initialisation
24	FMT_MTD.1	Management of TSF data

No.	Component	Component Name
25	FMT_SMF.1	Specification of Management Functions
26	FMT_SMR.1	Security roles
Class FPT: Protection of the TSF		
27	FPT_FLS.1	Failure with preservation of secure state
28	FPT_ITT.1	Basic internal TSF data transfer protection
29	FPT_RVM.1	Non-bypassability of the TSP
30	FPT_STM.1	Reliable time stamps

The following sections contain the functional components from the Common Criteria Part 2 [CC2] (CC) with the operations completed. The standard CC text is in regular font; the text inserted by the Security Target (ST) author is in accordance with the conventions described in at the beginning of this document.

5.1.1 Audit (FAU)

Security Alarms (FAU_ARP.1)

Hierarchical to: No other components.

FAU_ARP.1.1 The TSF shall take [action to produce alerts as specified by an Administrator] upon detection of a potential security violation.

Dependencies: FAU_SAA.1 Potential violation analysis

Audit data generation (FAU_GEN.1)

Hierarchical to: No other components.

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; and

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

c) [Access requests, authorisation decisions, policy changes, and custom alerts].

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, and [None].

Dependencies: FPT_STM.1 Reliable time stamps

User identity association (FAU_GEN.2)

Hierarchical to: No other components.

FAU_GEN.2.1 The TSF shall be able to associate each auditable event with identity of the user that caused the event.

Dependencies: FAU_GEN.1 Audit data generation

FIA_UID.1 Timing of identification

Potential Violation Analysis (FAU_SAA.1)

Hierarchical to: No other components.

FAU_SAA.1.1 The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP.

FAU_SAA.1.2 The TSF shall enforce the following rules for monitoring audited events:

a) Accumulation or combination of [administrator-defined events] known to indicate a potential security violation;

b) [None].

Dependencies: FAU_GEN.1 Audit data generation

Audit review (FAU_SAR.1)

Hierarchical to: No other components.

FAU_SAR.1.1 The TSF shall provide [authorised administrators] with the capability to read [policy administration and authorisation information regarding users, servers, administrators, dates and specifically defined audit events] from the audit records.

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

Dependencies: FAU_GEN.1 Audit data generation

Selectable audit review (FAU_SAR.3)

Hierarchical to: No other components.

FAU_SAR.3.1 The TSF shall provide the ability to perform *searches, sorting, ordering* of audit data based on [audit record attributes user, server, administrator, date and specifically defined audit events].

Dependencies: FAU_SAR.1 Audit review

Selective audit (FAU_SEL.1)

Hierarchical to: No other components.

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

a) *host identity, event type*

b) [Event source component, Event level].

Dependencies FAU_GEN.1 Audit data generation

FMT_MTD.1 Management of TSF data

Protected audit trail storage (FAU_STG.1)

Hierarchical to: No other components.

FCS_STG.1.1 The TSF shall protect the stored audit records from unauthorised deletion.

FCS_STG.1.2 The TSF shall be able to *detect* modifications to the audit records

Dependencies FAU_GEN.1 Audit data generation

5.1.2 Cryptographic support (FCS)

Cryptographic key generation (FCS_CKM.1 (1))

Hierarchical to: No other components

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [RSA] and specified cryptographic key sizes [at least 1024 bit] that meet the following [requirements for cryptographic key generation, as defined by RFC 2437 "PKCS #1: RSA Cryptography Specifications Version 2.0", October 1998.]

Dependencies: [FCS_CKM.2 Cryptographic key distribution **or**
FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction
FMT_MSA.2 Secure security attributes

Cryptographic key generation (FCS_CKM.1 (2))

Hierarchical to: No other components

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [3DES] and specified cryptographic key sizes [128 or 168 bit] that meet the following [requirements for cryptographic key generation, as defined by the Federal Information Processing Standard (FIPS) Publication 46-3, "Data Encryption Standard", 25 October 1999.]

Dependencies: [FCS_CKM.2 Cryptographic key distribution **or**
FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction
FMT_MSA.2 Secure security attributes

Cryptographic key generation (FCS_CKM.1 (3))

Hierarchical to: No other components

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [AES256] and specified cryptographic key sizes [256 bit] that meet the following [requirements for cryptographic key generation, as defined by the Federal Information Processing Standard (FIPS) Publication 197, "Advanced Encryption Standard (AES)", 26 November 2001.]

Dependencies: [FCS_CKM.2 Cryptographic key distribution **or**
FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction
FMT_MSA.2 Secure security attributes

Cryptographic key distribution (FCS_CKM.2(1))

Hierarchical to: No other components

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [RSA] that meets the following [requirements for cryptographic key distribution, as defined by RFC 2437 "PKCS #1: RSA Cryptography Specifications Version 2.0", October 1998.]

Dependencies: [FDP_ITC.1 Import of user data without security attributed **or**

FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

FMT_MSA.2 Secure security attributes

Cryptographic key distribution (FCS_CKM.2(2))

Hierarchical to: No other components

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [DHE] that meets the following [requirements for cryptographic key distribution, as defined by RFC 2631, Diffie-Hellman Key Agreement Method, June 1999.]

Dependencies: [FDP_ITC.1 Import of user data without security attributed **or**

FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

FMT_MSA.2 Secure security attributes

Cryptographic key destruction (FCS_CKM.4)

Hierarchical to: No other components

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [overwrite] that meets the following [requirements for cryptographic key destruction, as defined by:

"The SSL Protocol, Version 3.0", November 1995; and

RFC 2246, "The TLS Protocol, Version 1.0", January 1999.]

Dependencies: [FDP_ITC.1 Import of user data without security attributed **or**

FCS_CKM.1 Cryptographic key generation]

FMT_MSA.2 Secure security attributes

Cryptographic operation (FCS_COP.1 (1))

Hierarchical to: No other components

FCS_COP.1.1 The TSF shall perform [

"SSLV3.0 SSL_RSA_WITH_3DES_EDE_CBC_SHA"

] in accordance with a specified cryptographic algorithm [RSA, 3DES and SHA-1] and cryptographic key sizes [At least 1024 bit, 128 or 168 bit, and N/A] that meet the following: [requirements for cryptographic operations, as defined by:

"The SSL Protocol, Version 3.0", November 1995;

RFC 2437 "PKCS #1: RSA Cryptography Specifications Version 2.0", October 1998;

Federal Information Processing Standard (FIPS) Publication 46-3, "Data Encryption Standard", 25 October 1999; and

Federal Information Processing Standard (FIPS) Publication 180-1, "Secure Hash Algorithm", 17 April 1995.]

Dependencies: [FDP_ITC.1 Import of user data without security attributed **or**

FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

FMT_MSA.2 Secure security attributes

Cryptographic operation (FCS_COP.1 (2))

Hierarchical to: No other components

FCS_COP.1.1 The TSF shall perform [Digital signature creation and verification] in accordance with a specified cryptographic algorithm [RSA] and cryptographic key sizes [at least 1024 bit] that meet the following: [requirements for cryptographic operations, as defined by RFC 2437 "PKCS #1: RSA Cryptography Specifications Version 2.0", October 1998.]

Dependencies: [FDP_ITC.1 Import of user data without security attributed **or**

FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

FMT_MSA.2 Secure security attributes

Cryptographic operation (FCS_COP.1 (3))

Hierarchical to:	No other components
FCS_COP.1.1	The TSF shall perform [TLSv1.0 "TLS_DHE_RSA_WITH_AES256_CBC_SHA"] in accordance with a specified cryptographic algorithm [DHE, RSA, AES256 and SHA-1] and cryptographic key sizes [At least 1024 bit, 256 bit, and N/A] that meet the following: [requirements for cryptographic operations, as defined by: RFC 2246, "The TLS Protocol, Version 1.0", January 1999; RFC 2631, Diffie-Hellman Key Agreement Method, June 1999; RFC 2437 "PKCS #1: RSA Cryptography Specifications Version 2.0", October 1998; Federal Information Processing Standard (FIPS) Publication 197, "Advanced Encryption Standard (AES)", 26 November 2001; and Federal Information Processing Standard (FIPS) Publication 180-1, "Secure Hash Algorithm", 17 April 1995.]
Dependencies:	[FDP_ITC.1 Import of user data without security attributed **or** FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes

5.1.3 User data protection (FDP)

Complete access control (FDP_ACC.2)

Hierarchical to: FDP_ACC.1

FDP_ACC.2.1 The TSF shall enforce the [Authorisation SFP] on [all external entities, and all the resources protected by the TOE] 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.

Dependencies: FDP_ACF.1 Security attribute based access control

Security attribute based access control (FDP_ACF.1)

Hierarchical to: No other components

FDP_ACF.1.1 The TSF shall enforce the [Authorisation SFP] to objects based on [

1) The security policy attribute defined within the policy management matrix that links any external entity identity with a defined resource.
2) The value of the security policy attribute can be an ALLOW or DENY decision based on the application of the policy.
3) The application of the access policy which determines the value of the security policy attribute can make use of various combinations of the following attributes:
   a) The user's network address or domain name;
   b) The time of day when the user is attempting to access the resource;
   c) The user's encryption level;
   d) The directory attributes of the user's LDAP entry;
   e) The user's authentication method, which can include:
      i) A PKI certificate; and/or
      ii) A User ID and password.
   f) The port the user is attempting to access;
   g) The information embedded within a Policy Validator query, including:
      i) A PEM-encoded X.509 certificate;
      ii) The name and version of the client software;
      iii) The source or destination host names;
      iv) The source or destination host IP addresses;
      v) The source or destination port numbers;
      vi) The protocol used, e.g. http, https;
      vii) The protocol method requested, e.g. GRET, POST; or
      viii) The name and version of the server software.

].

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

a) if the external entity has been explicitly granted or denied access to the resource, then access is granted or denied according to the policy.

b) if the external entity has been conditionally granted or denied access to the resource, then access is granted or denied according to the policy.

c) if the external entity inherits an access policy (whether explicit or conditional) to the resource, then access is granted or denied according to the inherited policy.

d) if the "Unknown User" external entity has been explicitly or conditionally granted access to the resource, then access will be granted to all external entities.

e) if the external entity has been granted no access rights, then access to all resources is denied.

]

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

a) If a resource identity is configured as a "Pass-through Domain" or as an "Ignored File", access will be granted to all external entities.

]

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the [

a) If a resource identity is not listed within the policy matrix, access will be denied to all external entities.

]

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialisation

5.1.4 Identification and authentication (FIA)

Authentication failure handling (FIA_AFL.1)

Hierarchical to: No other components.

FIA_AFL.1.1 The TSF shall detect when [an Administrator defined number of] unsuccessful authentication attempts occur related to [password authentication].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [lock the associated account for an Administrator defined timeout period and raise a security alarm].

Dependencies: FIA_UAU.1 Timing of authentication

Timing of authentication (FIA_UAU.1)

Hierarchical to: No other components

FIA_UAU.1.1 The TSF shall allow [access to controlled resources that do not require authentication prior to access as defined in the policy management matrix] on behalf of the user to be performed before the user is authenticated.

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

Dependencies FIA_UID.1 Timing of identification

Multiple authentication mechanisms (FIA_UAU.5)

Hierarchical to: No other components.

FIA_UAU.5.1 The TSF shall provide [

a) external entity password mechanism

b) external entity X.509 certificate validation mechanism

c) Administrator password mechanism

] to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the [

authentication policy defined as follows:

a) If the user is an external entity requesting access to a controlled resource and the policy management matrix for that resource requires password authentication, then the external entity password mechanism is used.

b) If the user is an external entity requesting access to a controlled resource and the policy management matrix for that resource requires X.509 certificate based authentication, then the external entity X.509 certificate validation mechanism is used

c) If the user is a Administrator or has been delegated administrative duties and access is requested to administrative functions, then the Administrator password mechanism is used.

].

Dependencies No dependencies.

Re-authenticating (FIA_UAU.6)

Hierarchical to: No other components.

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions [

a) If the external entity has requested access to a controlled resource that requires a different or additional authentication mechanism to that already used to authenticate the entity as defined in the policy management matrix for that resource then external entity must re-authenticate.

b) If the Administrator configured session time limit has been reached, and the external entity requests access to a controlled resource that requires the same authentication mechanism as that already used to authenticate the entity, then the external entity must re-authenticate.

].

Dependencies No dependencies.

User identification before any action (FIA_UID.2)

Hierarchical to: FIA_UID.1

FIA_UID.2.1 The TSF shall require each user to identify itself before allowing any other TSF-mediated interactions on behalf of that user.

Dependencies No dependencies

5.1.5 Security Management (FMT)

Management of security functions behaviour (FMT_MOF.1(1))

Hierarchical to: No other components.

FMT_MOF.1.1 The TSF shall restrict the ability to *determine the behaviour of, disable, enable, modify the behaviour of* the functions [

audit

to [the Administrator].

Dependencies FMT_SMR.1 Security Roles

FMT_SMF.1 Specification of Management Functions

Management of security functions behaviour (FMT_MOF.1(2))

Hierarchical to: No other components.

FMT_MOF.1.1 The TSF shall restrict the ability to *determine the behaviour of, modify the behaviour of* the functions [authentication and authorisation] to [the Administrator and Delegated Administrators].

Dependencies FMT_SMR.1 Security Roles

FMT_SMF.1 Specification of Management Functions

Management of TSF data (FMT_MTD.1(1))

Hierarchical to: No other components.

FMT_MTD.1.1 The TSF shall restrict the ability to *query* the [

audit data

] to [the Administrator].

Dependencies FMT_SMR.1 Security Roles

FMT_SMF.1 Specification of Management Functions

Management of TSF data (FMT_MTD.1(2))

Hierarchical to: No other components.

FMT_MTD.1.1 The TSF shall restrict the ability to *query, modify, delete, clear* the [

TOE configuration information stored within the Configuration Data Store on the LDAP server used to hold the Policy Data Store

] to [the Administrator].

Dependencies FMT_SMR.1 Security Roles

FMT_SMF.1 Specification of Management Functions

Management of security attributes (FMT_MSA.1(1))

Hierarchical to: No other components.

FMT_MSA.1.1 The TSF shall enforce the [Authorisation SFP] to restrict the ability to *change default, query, modify, or delete* the security attributes [

a) The security policy attribute defined within the policy management matrix that links any external entity identity with any defined resource;

b) The value of the security policy attribute (an ALLOW or DENY decision) resulting from the application of the policy;

c) The combinations of the following attributes that determines the value of the security policy attribute:

- i) The user's network address or domain name;
- ii) The time of day when the user is attempting to access the resource;
- iii) The user's encryption level;
- iv) The directory attributes of the user's LDAP entry;
- v) The user's authentication method, which can include:
  - (1) A PKI certificate; and/or
  - (2) A User ID and password.
- vi) The port the user is attempting to access;
- vii) The information embedded within a Policy Validator query, including:
  - (1) A PEM-encoded X.509 certificate;
  - (2) The name and version of the client software;
  - (3) The source or destination host names;
  - (4) The source or destination host IP addresses;
  - (5) The source or destination port numbers;
  - (6) The protocol used, e.g. http, https;
  - (7) The protocol method requested, e.g. GRET, POST; or
  - (8) The name and version of the server software.

] to [the Administrator].

Dependencies: [FDP_ACC.1 Subset access control
or
FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles

Management of security attributes (FMT_MSA.1(2))

Hierarchical to: No other components.

FMT_MSA.1.1 The TSF shall enforce the [Authorisation SFP] to restrict the ability to *change default, query, modify, or delete* the security attributes [

a) The security policy attribute defined within the policy management matrix that links an external entity identity with a defined resource within the subset of the the policy management matrix that has been delegated to the Delegated Administrator;

b) The value of the security policy attribute (an ALLOW or DENY decision) resulting from the application of the policy within the subset of the the policy management matrix that has been delegated to the Delegated Administrator;

c) The combinations of the following attributes that determines the value of the security policy attribute within the subset of the the policy management matrix that has been delegated to the Delegated Administrator:

i) The user's network address or domain name;

ii) The time of day when the user is attempting to access the resource;

iii) The user's encryption level;

iv) The directory attributes of the user's LDAP entry;

v) The user's authentication method, which can include:

(1) A PKI certificate; and/or

(2) A User ID and password.

vi) The port the user is attempting to access;

vii) The information embedded within a Policy Validator query, including:

(1) A PEM-encoded X.509 certificate;

(2) The name and version of the client software;

(3) The source or destination host names;

(4) The source or destination host IP addresses;

(5) The source or destination port numbers;

(6) The protocol used, e.g. http, https;

(7) The protocol method requested, e.g. GRET, POST; or

(8) The name and version of the server software.

] to [Delegated Administrators].

Dependencies: [FDP_ACC.1 Subset access control
or
FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles

Secure security attributes (FMT_MSA.2)

Hierarchical to: No other components.

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for security attributes.

Dependencies: ADV_SPM.1 Informal TOE security policy model
[FDP_ACC.1 Subset access control
or
FDP_IFC.1 Subset information flow control]
FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

Static attribute initialisation (FMT_MSA.3)

Hierarchical to: No other components.

FMT_MSA.3.1 The TSF shall enforce the [Authorisation 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 Administrator and Delegated Administrators] to specify alternative initial values to override the default values when an object or information is created.

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

Specification of Management Functions (FMT_SMF.1)

Hierarchical to: No other components.

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: [management of audit functions and data, management of authentication and authorisation functions and data, management of TOE roles, management of TOE configuration information].

Dependencies No dependencies

Security management roles (FMT_SMR.1)

Hierarchical to: No other components.

FMT_SMR.1.1 The TSF shall maintain the roles: [Administrator, Delegated Administrators].

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

Dependencies FIA_UID.1 Timing of identification

5.1.6 Protection of the TSF (FPT)

Failure with preservation of secure state (FPT_FLS.1)

Hierarchical to: No other components

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: [If the Enforcer fails to receive a response to a request to a Validator for access to a controlled resource, then the Enforcer shall deny access to that resource].

Dependencies: ADV_SPM.1 Informal TOE security policy model

Basic internal TSF data transfer protection (FPT_ITT.1)

Hierarchical to: No other components

FPT_ITT.1.1 The TSF shall protect TSF data from *disclosure* when it is transmitted between separate parts of the TOE.

Dependencies: No dependencies

Non-bypassability of the TSP (FPT_RVM.1)

Hierarchical to: No other components

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.

Dependencies: No dependencies

Reliable time stamps (FPT_STM.1)

Hierarchical to: No other components

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps for its own use.

Dependencies: No dependencies

5.2 TOE Security Assurance Requirements

This section contains the assurance requirements for the TOE. The assurance requirements are listed in summary form in Table 10, below.

Table 10: TOE Security Assurance Requirements

No.	Component	Component Name
Class ACM: Configuration management		
1	ACM_CAP.2	Configuration items
Class ADO: Delivery and Operation		
2	ADO_DEL.1	Delivery procedures
3	ADO_IGS.1	Installation, generation and start-up
Class ADV: Development		
4	ADV_FSP.1	Informal functional specification
5	ADV_HLD.1	Descriptive high level design
6	ADV_RCR.1	Informal representational correspondence
Class AGD: Guidance documents		
7	AGD_ADM.1	Administrator guidance
8	AGD_USR.1	User guidance
Class ATE: Tests		
9	ATE_COV.1	Evidence of coverage
10	ATE_FUN.1	Functional testing
11	ATE_IND.2	Independent testing- sample
Class AVA: Vulnerability Assessment		
12	AVA_SOF.1	Strength of TOE security function evaluation
13	AVA_VLA.1	Developer vulnerability analysis

5.2.1 Configuration management (ACM)

Configuration Items (ACM_CAP.2)

ACM_CAP.2.1D The developer shall provide a reference for the TOE.

ACM_CAP.2.2D The developer shall use a CM system.

ACM_CAP.2.3D The developer shall provide CM documentation.

ACM_CAP.2.1C The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.2.2C The TOE shall be labelled with its reference.

ACM_CAP.2.3C The CM documentation shall include a configuration list.
The configuration list shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.2.4C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.2.5C The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.2.6C The CM system shall uniquely identify all configuration items.

5.2.2 Delivery and operation (ADO)

Delivery procedures (ADO_DEL.1)

ADO_DEL.1.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.

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

ADO_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

Installation, generation, and start-up procedures (ADO_IGS.1)

ADO_IGS.1.1D The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

ADO_IGS.1.1C The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation, and start-up of the TOE.

5.2.3 Development (ADV)

Informal functional specification (ADV_FSP.1)

ADV_FSP.1.1D The developer shall provide a functional specification.

ADV_FSP.1.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV_FSP.1.2C The functional specification shall be internally consistent.

ADV_FSP.1.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

ADV_FSP.1.4C The functional specification shall completely represent the TSF.

Descriptive high-level design (ADV_HLD.1)

ADV_HLD.1.1D The developer shall provide the high-level design of the TSF.

ADV_HLD.1.1C The presentation of the high-level design shall be informal.

ADV_HLD.1.2C The high-level design shall be internally consistent.

ADV_HLD.1.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.1.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.1.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.1.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.1.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

Informal correspondence demonstration (ADV_RCR.1)

ADV_RCR.1.1D The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

ADV_RCR.1.1C For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

5.2.4 Guidance documents (AGD)

Administrator guidance (AGD_ADM.1)

AGD_ADM.1.1D The developer shall provide administrator guidance addressed to system administrative personnel.

AGD_ADM.1.1C The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD_ADM.1.2C The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.4C The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.

AGD_ADM.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD_ADM.1.6C The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

User guidance (AGD_USR.1)

AGD_USR.1.1D The developer shall provide user guidance.

AGD_USR.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behaviour found in the statement of TOE security environment.

AGD_USR.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

5.2.5 Tests (ATE)

Evidence of coverage (ATE_COV.1)

ATE_COV.1.1D The developer shall provide evidence of the test coverage.

ATE_COV.1.1C The evidence of the test coverage shall show the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

Functional testing (ATE_FUN.1)

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

ATE_FUN.1.2D The developer shall provide test documentation.

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

ATE_FUN.1.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

Independent testing - sample (ATE_IND.2)

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

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

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

5.2.6 Vulnerability Analysis (AVA)

Strength of TOE security functions (AVA_SOF.1)

AVA_SOF.1.1D The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

AVA_SOF.1.1C For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.2C For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

Vulnerability analysis (AVA_VLA.1)

AVA_VLA.1.1D The developer shall perform a vulnerability analysis.

AVA_VLA.1.2D The developer shall provide vulnerability analysis documentation.

AVA_VLA.1.1C The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for obvious ways in which a user can violate the TSP.

AVA_VLA.1.2C The vulnerability analysis documentation shall describe the disposition of obvious vulnerabilities.

AVA_VLA.1.3C The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

5.3 Security Requirements for the IT Environment

The TOE has no requirements for the IT environment.

5.4 Security Requirements for the Non-IT Environment

The TOE has the following security requirements for the Non-IT Environment.

ENV_NONIT.1 TOE components are to be physically protected.

The TOE components shall be located within a controlled access facility that will prevent unauthorised physical access.

ENV_NONIT.2 The Administrator and individuals delegated with administrative functions are well-trained according to their role.

The TOE environment shall ensure that the Administrator, and individuals delegated with administrative functions are trained and motivated to make the right choices when providing administrative support to the TOE. This training should include security awareness of vulnerabilities and familiarisation with supporting technologies.

ENV_NONIT.3 Controlled Administrator Access to TOE components

The TOE environment shall provide procedures for installing, configuring and maintaining the underlying operating system for each of the TOE components such that access is limited to persons undertaking the role of the authorised Administrator. For example, accounts on the TOE component platforms should only exist for those persons undertaking the role of the authorised Administrator.

ENV_NONIT.4 Configuration of infrastructure

The TOE environment shall provide procedures and guidance for the Administrator to ensure that the infrastructure surrounding and supporting the TOE is installed configured and maintained in accordance with the organisational policy as defined by P.NETWORK.

ENV_NONIT.5 Management of authentication credentials

The TOE relies on the use of credentials for authenticating individuals and for the authentication of TOE components. An appropriate means of securely generating, distributing and managing TOE authentication credentials should exist within the TOE operating environment.

6 TOE Summary Specification

This section presents the Security Functions implemented by the TOE and the Assurance Measures applied to ensure their correct implementation.

6.1 Security Functions

Table 11 describes the Security Functions performed by the TOE.

Table 11: Security Functions

Security Function Label	Security Function Description
VALIDATE	The TOE provides a policy validation and authorisation function through the Validator component. The Validator component receives queries from the Enforcer component, on behalf of the controlled network resources, in the form of an XML document. Based on this information, the Validator component retrieves policy information from the LDAP directory server and determines whether or not the access request is allowed. The result is passed back to the Enforcer component, which then enforces the decision. The Validator component evaluates the access control and authorisation rules defined in the policies for both the requesting user and the requested resource. Where authentication is required, the Validator component will request the Enforcer component to prompt for authentication. When the authentication information is then provided, the Validator component will validate the authentication information. The TOE provides session-based authentication through the use of digitally signed credentials. The Validator component creates and signs a credential when a user is successfully authenticated, and the credential is forwarded to the Enforcer component and on to the user. The Validator component then maintains state information relating to that user and will allow the digitally signed credential to be used for on-going authentication. The administrator configures the lifetime of the credential. The Validator component supports SSL communications to both the Enforcer component and the LDAP directory server.
ENFORCE	The TOE provides a policy enforcement function through the Enforcer component. The Enforcer component is the plug-in on a particular network resource. This resource can be a Web server or J2EE application server. The Enforcer component intercept all incoming access attempts and gathers information about those requests and sends it as an XML query to the Validator component to be validated. The Enforcer component is responsible for examining the XML-based response from the Validator component and enforcing the authorisation decision. The Enforcer component supports secure fail-over between multiple Validator components, with communications secured using SSL. If no Validator component is responding then the Enforcer component will, by default, deny access to the requested resource. If the Validator component response requires authentication prior to granting access to the requested resource then the Enforcer component will request the necessary authentication information from the user. The Enforcer component will handle different types of authentication information (e.g. passwords and X.509 certificates) and will request the appropriate type of authentication based on information from the Validator component.

Security Function Label	Security Function Description
POLICY	The TOE provides a Java-based GUI to define authorisation policies for users and resources. Users and resources are displayed on the axes of an expandable matrix, which allows the Administrator and Delegated Administrators to "allow", "deny" or "conditionally" authorise access, by users, or groups of users, to controlled resources. The Administrator and Delegated Administrators may select from a range of authentication mechanisms for requests to access resources. Conditional rules can be defined for access to resources based on the authentication method, source addresses, date and time of request, level of encryption. The Administrator and Delegated Administrators may define password policies for authentication with characterisation including password expiration periods, temporary passwords, failed password attempt account locking and account reactivation latency. Initial Policy settings are restrictive, whereby access is denied to all resources. Access by users to resources must be explicitly enabled by administrators. Policy definition scales to support organisations with many users and/or resources by providing policy inheritance. As Administrators apply policy to groups of users and resources, policy is inherited over the members of those groups. As new users and resources are added, the appropriate policy is inherited from policies set on the user and resource groups above.
DELEGATE	The TOE allows the Administrator to delegate administration of authorisation policy to trusted individual or group, a Delegated Administrator. These people can then administer the authorisation policy for a defined subset of users and/or resources. The TOE permits multiple Delegated Administrators for the same or different sections of the Policy Matrix. Each Delegated Administrator can, in turn, further delegate administration to another individual, and so on. The Delegated Administrator can only see the user data and the resource structure for which they have been given permission to administer. The Delegated Administrators use the same Java-based GUI interface as the Administrator to manage a limited subset of the authorisation policy. The Delegated Administration function relies on the Validator component to determine and enforce access policy decisions. Thus, the TOE provides authorisation for its own delegated administration function.

Security Function Label	Security Function Description
ADMIN	The Policy Builder is a remote, Java-based GUI client of the Administration Server that one or more administrators can use to administer the TOE, access policies and delegation/sub-delegation of administration duties to other users. The Administration Server is a Web-server based component of the TOE that serves the Policy Builder over SSL and which provides administrative functions including component configuration, certificate management, and policy data administration. The Policy Builder administration interface configures all aspects of the TOE, including: • Create, manage and view authorisation policy; • Manage resources, users, groups and attribute-based roles; • Define and manage delegated administration; • Manage authentication and encryption requirements; and • Audit and reporting of policy changes. From the Policy Builder GUI, the users and resources of an entire network are displayed from a single administration point, within which the distributed components of the TOE can be managed and configured. Administration and configuration information is stored in the LDAP directory server. Users, resources and associated policy rules are displayed in a scalable matrix – the Policy Matrix. This hierarchical matrix not only makes it easy to understand the overall policy, but also makes it easy to see the specific policy for a particular network resource for a given person. Access to the Policy Builder is controlled through two mechanisms. Firstly, a user ID and password combination is required for Administrator and Delegated Administrator access. If that is successfully authenticated then a Delegated Administrator must also present a Delegated Administrator "role" certificate. All communications from the Administration GUI (Policy Builder) to other components of the TOE and to the LDAP directory server are secured using SSL. The Administrator may also access local administration functions through a "Setup" utility on each TOE component. Local operating system level editing of TOE configuration files also allows the Administrator to configure some parameters of a specific TOE component. Configuration information for all TOE components (other than the Secure Audit Server) is stored on the LDAP server used to hold the Policy Store. Each component reads this configuration information (Configuration Data Store) at startup. Local XML-format text files hold just enough information to allow the Validator and Enforcer components to connect to the Administration Server, and to allow the Administration Server components to connect to the LDAP server where component configuration information is stored.
SIGN	The TOE provides a digital signature function, based on X.509 certificates, to provide protection for the integrity of critical TOE information. The Administrator can require all policies to be signed prior to storage in the LDAP directory. In addition, the Administrator can configure the Secure Audit Server to require either individual audit records or streams of audit data to be signed prior to storage in a file or database.

Security Function Label	Security Function Description
AUDIT	The TOE provides a consolidated security audit trail. A specific TOE component, the Secure Audit Server, is dedicated to collecting, storing and protecting audit data. Audit entries are logged from the other TOE components. The level of logging can be configured by the Administrator both specifically for each TOE component, as well as at the Secure Audit Server for the whole TOE. The startup and shutdown of audit services are logged to the Secure Audit Server. All access requests, authorisation decisions, and policy changes are logged and digitally signed to protect the integrity of audit data. The Secure Audit Server outputs to multiple destinations including databases supporting JDBC, UNIX syslog, NT Event Log, email messages, and/or flat files. Different output destinations can be configured based on audit data attributes, such as audit component (i.e., administration activities, authorisation decision, access query) and event level (i.e., information, warning). In flat files or databases audit records are stored as XML messages containing data fields including data and time, user identity, originating TOE component, event level, and result of action. Modification and deletion of audit records are detectable by the failure of digital signature verification, even if an entire block of records, together with its attendant digital signature, is deleted. The Administrator can identify possible record deletions by identifying gaps in audit record timestamps. In the case of flat file audit logs, log names are numbered sequentially to allow the deletion of entire files to be detected. All communications to the Secure Audit Server from other components of the TOE are secured using SSL. The TOE provides a Reporting Engine that enables an organisation to define reporting procedures commensurate with their operational and audit policies. Administrators can create (view, save, print) reports on policy administration and authorisation based on sorted and/or filtered audit data containing attribute such as users, servers, administrators, dates and specific audit events. Audit data held in flat files and databases can be used as the source for such reports. Report formats may be customised and report output data may be sent to comma separated value files (CSV) for integration with appropriate report analysis and spreadsheet tools. The TOE provides the ability to define custom alerts based on authorisation information. Alerts can be set to one of five levels (Debug, Info, Warning, Error, Fatal) for which each level can be configured to a different alert handling instruction. Custom alerts can be created based on condition points in a condition rule, e.g. failed login, attempts to access restricted resources, successful login. The TOE provides the ability to customise alert handling procedures for all recorded audit events based on attributes including user, server, administrator, date and specifically defined audit events. Alert handling instructions may be written for the inclusion of additional information in audit logs or sending emails notifying critical event to a select list of recipients.

6.2 Assurance Measures

The TOE claims to satisfy the assurance requirements for the Common Criteria Evaluation Assurance Level EAL2 (CC EAL2). Table 12 identifies the assurance measures relevant to the TOE that satisfy the CC EAL2 assurance requirements defined in the CC Part3 [CC3].

Table 12: Assurance Measures

Assurance Measure Label	Assurance Measure Description
CM_DOC	Configuration management documentation that includes a configuration list, a description of the configuration items comprising the TOE and a description of the method used to uniquely identify the configuration items. Document(s) title: HP OpenView Select Access Configuration Management Documentation, Version 1.3, April 2004; and Appendix A of this document.
DEL_DOC	Delivery documentation that describes all procedures necessary to maintain security for distribution of the TOE to a user's site. Document(s) title: HP OpenView Select Access Version 5.2 Delivery Document, version 1.1 May 2004.
IGS_DOC	Installation and generation documentation that describes the steps necessary for secure installation, generation and startup of the TOE. Document(s) title: HP OpenView Select Access Version 5.2 Installation Guide, March 2004. HP OpenView Select Access Version 5.2 Network Integration Guide, October 2003. HP OpenView Select Access Integration Paper – Select Access & IBM WebSphere application server 4.0, October 2003. HP OpenView Select Access Integration Paper – Plumtree Corporate Portal version 4.5, October 2004. HP OpenView Select Access Integration Paper – Select Access & Oracle Internet Directory version 3.0.1.1, October 2003. HP OpenView Select Access Integration Paper – Select Access & iPlanet application server, October 2003 HP OpenView Select Access Integration Paper – Select Access & eTrust Directory Server version 3.6, October 2003. HP OpenView Select Access Version 5.2 Policy Builder Guide, March 2004. HP OpenView Select Access Version 5.2 Developer's Guide, October 2003. HP OpenView Select Access Version 5.2 Release Notes, October 2003
FUN_SPEC	Functional specification that describes the TSF and its external interfaces and the purpose and method of use of external TSF interfaces, including details of effects, exceptions and error messages. Document(s) title: HP OpenView Select Access Functional Specification, Version 2.4, February 2004.
HLD_DOC	High-level design that describes the structure of the TSF in terms of sub-systems and describes the security functionality provided by each sub-system. Document(s) title: HP OpenView Select Access High Level Design, Version 1.4, April 2004.

Assurance Measure Label	Assurance Measure Description
RCR_DOC	Representation correspondence analysis that, for each adjacent pair TSF representations, demonstrates that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation. Document(s) title: HP OpenView Select Access Representational Correspondence, Version 1.4, April 2004.
ADMIN	Administrator guidance that describes the administrative functions and interfaces available to the administrator of the TOE, describes how to administer the TOE in a secure manner, describes warnings about functions and privileges that should be controlled in a secure processing environment, describes all assumptions about user behaviour relevant to secure operation, describes all security parameters under the control of the administrator, and describes each type of security relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF. Document(s) title: HP OpenView Select Access Version 5.2 Developer's Guide October 2003. HP OpenView Select Access Version 5.2 Policy Builder Guide, March 2004. HP OpenView Select Access Version 5.2 Network Integration Guide, October 2003. HP OpenView Select Access Version 5.2 Release Notes, October 2003.
USER	User guidance that describes the functions and interfaces available to the non-administrative users of the TOE, describes the use of user accessible security functions, describes warnings about user accessible functions and privileges that should be controlled in a secure processing environment, and describes all user responsibilities necessary for secure operation of the TOE. Document(s) title: None. HP OpenView Select Access is designed to be operated by trained administrators only. The TOE therefore makes no distinction between Administrator and User Guidance. The guidance documents that meet the AGD_ADM.1 Assurance Requirement (see "ADMIN", above) also meet the assurance requirements for AGD_USR.1.
TEST_COV	Test evidence that shows the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification. Document(s) title: : HP OpenView Select Access Functional Testing and Coverage Analysis, Version 1.8, April 2004.
TEST_DOC	Test documentation consisting of test plans, test procedure descriptions, expected test results and actual test results. The test plan identifies the security functions to be tested and the goal of the tests to be performed. The test procedure descriptions identify the tests to be performed, and describes the scenarios for testing each security function. The expected test results show the anticipated outputs from successful test execution. The actual test results demonstrate that each tested security function behaved as specified. Document(s) title: : HP OpenView Select Access Functional Testing and Coverage Analysis, Version 1.8, April 2004.
TEST_DOC	The TOE and necessary supporting infrastructure suitable for testing. Document(s) title & Equipment: : HP OpenView Select Access Functional Testing and Coverage Analysis, Version 1.8, April 2004.

Assurance Measure Label	Assurance Measure Description
SOF_DOC	For each mechanism identified in the Security Target, an analysis shows that the claimed strength of TOE security function meets or exceeds the minimum strength level defined in the Security Target. Document(s) title: HP OpenView Select Access Strength of Function Analysis, Version 1.4, April 2004.
VLA_DOC	A vulnerability analysis that shows that for all identified vulnerabilities, the vulnerability cannot be exploited in the intended environment of the TOE. Document(s) title: HP OpenView Select Access Vulnerability Analysis, Version 1.5, April 2004.

7 PP Claims

The HP OpenView Select Access Security Target was not written to address a published Protection Profile.

8 Rationale

8.1 Security Objectives Rationale

The purpose of this rationale is to demonstrate that the identified security objectives are *suitable*, that is:

- that they are *necessary*, i.e., there are no redundant security objectives; and
- they are *sufficient* to address the security needs.

8.1.1 Security Objectives are Necessary

The need to demonstrate that there are no redundant security objectives is satisfied as follows:

- The first section (Table 13) shows that all of the secure usage assumptions, organisational security policies, and threats to security have been addressed.
- The second section (Table 14) shows that each security objective for the TOE and its environment counters at least one assumption, policy, or threat.

Table 13: Mapping of Threats, Assumptions and OSPs to Security Objectives

Label	Associated Security Objective
A.ADMIN-DOCS	OE.TRAIN
A.NO_EVIL	OE.TRAIN
A.RESOURCES	O.AUTHENTICATE O.AUTHORISE
A.PHYSICAL	OE.PHYS_ENV
A.NETWORK	OE.NETWORK OE.TRAIN
A.CREDENTIALS	OE.CREDENTIALS
T.NO_DETECT	O.AUDIT OE.TRAIN
T.NO_ACCESS	O.AUTHORISE O.AUTHENTICATE O.AUDIT
T.IMPERSON	O.AUTHENTICATE O.AUDIT
T.CAPTURE	O.ENCRYPT
T.INTEGRITY	O.ENCRYPT O.INTEGRITY
T.FAILURE	O.FAIL_SAFE
TE.INSTALL	OE.TRAIN
P.AUTHORISE	O.AUTHORISE O.AUTHENTICATE

Label	Associated Security Objective
P.AUDIT	O.AUDIT O.ADMIN
P.CRYPTO	O.ADMIN OE.PHYS_ENV OE.NETWORK
P.TRAIN	OE.TRAIN
P.ROLES	O.ADMIN
P.NETWORK	OE.PHYS_ENV OE.NETWORK

Table 14 shows that there are no unnecessary IT security objectives.

Table 14: Mapping of Security Objectives to Threats, Policies and Assumptions

Objective Label	Threat / Policy/ Assumption
O.AUTHORISE	A.RESOURCES T.NO_ACCESS P.AUTHORISE
O.AUTHENTICATE	A.RESOURCES T.NO_ACCESS T.IMPERSON P.AUTHORISE
O.AUDIT	T.NO_DETECT T.NO_ACCESS T.IMPERSON P.AUDIT
O.ADMIN	P.AUDIT P.CRYPTO P.ROLES
O.ENCRYPT	T.CAPTURE T.INTEGRITY
O.INTEGRITY	T.INTEGRITY
O.FAIL_SAFE	T.FAILURE
OE.NETWORK	A.NETWORK P.CRYPTO P.NETWORK
OE.PHYS_ENV	P.CRYPTO

Objective Label	Threat / Policy/ Assumption
OE.TRAIN	A.ADMIN-DOCS A.NO_EVIL A.NETWORK T.NO_DETECT TE.INSTALL P.TRAIN
OE.CREDENTIALS	A.CREDENTIALS

8.1.2 Security Objectives are Sufficient

The arguments provided in Table 15 demonstrate the sufficiency of the Security Objectives outlined above.

Table 15: Sufficiency of Security Objectives

Label	Argument to support Security Objective sufficiency
A.ADMIN-DOCS	This assumption is upheld by the objective OE.TRAIN which ensures that the Administrator, and those delegated with administrative responsibilities, receive appropriate training to enable them to operate and manage the TOE, and its supporting technologies, properly.
A.NO_EVIL	This assumption is upheld by the security objective OE.TRAIN, which ensures that the Administrator, and those delegated with administrative functions, receive appropriate training to enable them to operate and manage the TOE, and its supporting technologies, properly. It also ensures that the organisation carries out the appropriate background checks and clearances for the individuals carrying out the Administrator and Delegated Administrator roles.
A.RESOURCES	This assumption is upheld by the following security objectives: • O.AUTHORISE, which provides the means to grant or deny access for individuals or groups of individuals to resources. • O.AUTHENTICATE, which provides the means to authenticate individuals using the appropriate authentication mechanism before access to a specific resource or group of resources protected by the TOE is granted.
A.PHYSICAL	This assumption is upheld by OE.PHYS_ENV, which ensures that physical access to all TOE components is appropriately controlled.
A.NETWORK	This assumption is upheld by the following security objectives: • OE.NETWORK, which ensures that the critical TOE components are protected against logical attacks that could arise due to its distributed architecture. • OE.TRAIN, which ensures that administrators receive appropriate training in security vulnerabilities and familiarisation of supporting technologies.
A.CREDENTIALS	This assumption is upheld by the security objective OE.CREDENTIALS, which ensures that an appropriate means of securely generating, distributing and managing TOE authentication credentials, exists in the TOE environment, such that good authentication credentials are generated.

Label	Argument to support Security Objective sufficiency
T.NO_DETECT	The threat of an undetected attack is countered by the following security objectives: • O.AUDIT, which ensures that all security-relevant events that may indicate an attack on the TOE security functions are recorded and that information recorded is sufficient to hold individuals accountable for their security-relevant actions, and that administrators are alerted to potential security violations. • OE.TRAIN supports O.AUDIT by ensuring that the Administrator, and those delegated with administrative functions, receive appropriate training for the secure management and operation of the TOE. This training includes procedures for the regular inspection and review of the audit trails, and awareness training to detect possible attacks on the TOE.
T.NO_ACCESS	The threat of attacker gaining unauthorised access to resources protected by the TOE is countered by the following security objectives: • O.AUTHORISE, which provides the means to grant or deny access for individuals or groups of individuals to resources. • O.AUTHENTICATE, which provides the means to authenticate individuals using the appropriate authentication mechanism before access to a specific resource or group of resources protected by the TOE is granted. • O.AUDIT supports O.AUTHORISE and O.AUTHENTICATE by ensuring that all security-relevant events that may indicate attempts to gain unauthorised access to resources protected by the TOE are recorded.
T.IMPERSON	The threat that an attacker may attempt to impersonate and authorised user of the TOE is countered by the following security objectives: • O.AUTHENTICATE, which provides the means to authenticate individuals using the appropriate authentication mechanism before access to a specific resource or group of resources protected by the TOE is granted. • O.AUDIT supports O.AUTHENTICATE by ensuring that all security-relevant events that may indicate attempts to gain unauthorised access to resources protected by the TOE are recorded.
T.CAPTURE	The threat that an attacker may attempt to capture plain data transmitted between the TOE components and between the TOE and the resources it protects is countered by O.ENCRYPT, which provides the means to protect the confidentiality and integrity of transmitted data.
T.INTEGRITY	The threat that the integrity of authentication and/or authorisation information is compromised due to transmission errors, application errors or malicious actions is countered by the following security objectives: • O.ENCRYPT, which protects the confidentiality and integrity of information transmitted between TOE components and between TOE components and the resources protected by the TOE. • O.INTEGRITY, which protects the integrity of stored authentication and authorisation information.
T.FAILURE	The threat that the resources protected by the TOE may be compromised by a failure of one or more of its components is countered by O.FAIL_SAFE, which ensures that access to IT resources is not granted in the event of a failure of one or more components.

Label	Argument to support Security Objective sufficiency
TE.INSTALL	The threat of the TOE being delivered or installed in a manner which undermines security is countered by the security objective OE.TRAIN, which ensures that the Administrator, and those delegated with administrative functions, receive appropriate training to enable them to operate and manage the TOE and its supporting technologies properly.
P.AUTHORISE	The OSP requirement for definition of rules for granting or denying access, and requiring authenticated access, is met by: • O.AUTHORISE, which provides the means to grant or deny access for individuals or groups of individuals to resources. • O.AUTHENTICATE, which provides the means to authenticate individuals using the appropriate authentication mechanism before access to a specific resource or group of resources protected by the TOE is granted.
P.AUDIT	The OSP requirement for recording user activity in an audit trail is met by: • O.AUDIT, which provides functionality to record security-relevant events in such a way that an individual may be held accountable for their security-relevant actions. • O.ADMIN, which provides the means for the authorised Administrator to manage audit trail information.
P.CRYPTO	The OSP requirement for appropriate physical and technical controls for the protection of cryptographic material is met by: • O.ADMIN, which provides the means for the authorised Administrator to manage cryptographic functionality of the TOE. • OE.PHYS_ENV, which ensures that physical access to TOE components (which includes cryptographic material used by those components) is strictly controlled. • OE.NETWORK, which ensures that procedures and/or mechanisms are in place to control logical access to TOE components (which includes cryptographic material used by those components).
P.TRAIN	The OSP requirement for appropriate ongoing training for the Administrator and individuals delegated administrative functions is met by OE.TRAIN, which ensures that the Administrator, and those delegated with administrative functions, receive appropriate training to enable them to operate and manage the TOE, and its supporting technologies, properly.
P.ROLES	The OSP requirement for definition of the Administrator role and the mechanism for delegation of administrative functions is met by O.ADMIN, which provides the means for those individuals identified as administrators, to manage the TOE and its security functions. This includes the ability to delegate administration of subsets of security functions.
P.NETWORK	The OSP requirement for consideration of the placement and protection of the TOE within the context of maintaining organisational security is met by the following security objectives: • OE.PHYS_ENV, which ensures that strong physical access controls are employed to protect TOE components so as to maintain organisational security. • OE.NETWORK, which ensures that appropriate procedures and/or mechanisms are employed to protect TOE components so as to maintain organisational security.

8.2 Security Requirements Rationale

The purpose of this rationale is to demonstrate that the identified security requirements are *suitable* to meet the TOE security objectives, that is:

- that they are *necessary*, i.e., there are no redundant security requirements; and
- they are *sufficient* to address the TOE security objectives.

8.2.1 Security Requirements are Necessary

The need to demonstrate that there are no redundant security requirements is satisfied as follows:

- The first section (Table 16) shows that each TOE security objective is addressed by at least one security requirement.
- The second section (Table 17) shows that each security requirement for the TOE and its environment addresses at least one TOE security objective.

Note that several objectives are partially satisfied by the TOE and partially satisfied by the TOE environment. Security Objectives for the TOE are satisfied by Common Criteria functional components.

Table 16: Mapping of Security Objectives to Security Requirements

Objectives	Requirements
O.AUTHORISE	FDP_ACC.2 FDP_ACF.1 FPT_RVM.1
O.AUTHENTICATE	FIA_AFL.1 FIA_UAU.1 FIA_UAU.5 FIA_UAU.6 FIA_UID.2
O.AUDIT	FAU_ARP.1 FAU_GEN.1 FAU_GEN.2 FAU_SAA.1 FAU_SAR.1 FAU_SAR.3 FAU_SEL.1 FAU_STG.1 FCS_COP.1(2) FMT_MOF.1(1) FMT_MTD.1 FPT_STM.1

Objectives	Requirements

O.ADMIN	FCS_COP.1(2)
	FMT_MOF.1(1)
	FMT_MOF.1(2)
	FMT_MSA.1(1)
	FMT_MSA.1(2)
	FMT_MSA.2
	FMT_MSA.3
	FMT_MTD.1
	FMT_SMR.1
	FMT_SMF.1

O.ENCRYPT	FCS_CKM.1(1)
	FCS_CKM.1(2)
	FCS_CKM.1(3)
	FCS_CKM.2(1)
	FCS_CKM.2(2)
	FCS_CKM.4
	FCS_COP.1(1)
	FCS_COP.1(3)
	FPT_ITT.1

O.INTEGRITY	FCS_COP.1(2)
O.FAIL_SAFE	FPT_FLS.1
OE.NETWORK	ENV_NONIT.3 ENV_NONIT.4
OE.PHYS_ENV	ENV_NONIT.1
OE.TRAIN	ENV_NONIT.2
OE.CREDENTIALS	ENV_NONIT.5

Table 17: Mapping of Security Requirements to Security Objectives

Component	Objective
FAU_ARP.1	O.AUDIT
FAU_GEN.1	O.AUDIT
FAU_GEN.2	O.AUDIT
FAU_SAA.1	O.AUDIT
FAU_SEL.1	O.AUDIT
FAU_SAR.1	O.AUDIT
FAU_SAR.3	O.AUDIT
FAU_STG.1	O.AUDIT
FCS_CKM.1(1)	O.ENCRYPT
FCS_CKM.1(2)	O.ENCRYPT

Component	Objective
FCS_CKM.1(3)	O.ENCRYPT
FCS_CKM.2(1)	O.ENCRYPT
FCS_CKM.2(2)	O.ENCRYPT
FCS_CKM.4	O.ENCRYPT
FCS_COP.1(1)	O.ENCRYPT
FCS_COP.1(2)	O.AUDIT O.ADMIN O.INTEGRITY
FCS_COP.1(3)	O.ENCRYPT
FDP_ACC.2	O.AUTHORISE
FDP_ACF.1	O.AUTHORISE
FIA_AFL.1	O.AUTHENTICATE
FIA_UAU.1	O.AUTHENTICATE
FIA_UAU.5	O.AUTHENTICATE
FIA_UAU.6	O.AUTHENTICATE
FIA_UID.2	O.AUTHENTICATE
FMT_MOF.1(1)	O.ADMIN O.AUDIT
FMT_MOF.1(2)	O.ADMIN
FMT_MTD.1	O.ADMIN O.AUDIT
FMT_MSA.1(1)	O.ADMIN
FMT_MSA.1(2)	O.ADMIN
FMT_MSA.2	O.ADMIN
FMT_MSA.3	O.ADMIN
FMT_SMR.1	O.ADMIN
FMT_SMF.1	O.ADMIN
FPT_FLS.1	O.FAIL_SAFE
FPT_ITT.1	O.ENCRYPT
FPT_RVM.1	O.AUTHORISE
FPT_STM.1	O.AUDIT
ENV_NONIT.1	OE.PHYS_ENV
ENV_NONIT.2	OE.TRAIN
ENV_NONIT.3	OE.NETWORK
ENV_NONIT.4	OE.NETWORK
ENV_NONIT.5	OE.CREDENTIALS

8.2.2 Security Requirements are Sufficient

The arguments in Table 18 demonstrate that security requirements are *sufficient* to satisfy the TOE security objectives, whether in a principal or supporting role.

Table 18: Sufficiency of Security Requirements

Objectives	Argument to support sufficiency of Security Requirements
O.AUTHORISE	The objective to provide the means to grant or deny access to individuals or groups of individuals to IT resources is met by the following security requirements: • FDP_ACC.2 requires that all requests for access to resources protected by the TOE by external entities is mediated by the TOE through the Authorisation SFP. • FDP_ACF.1 supports FDP_ACC.2 by enforcing the Authorisation SFP based on a set of rules. • FPT_RVM.1 prevents bypass of the TSP enforcement functions (in this case authentication and authorisation), requiring that these functions are invoked and succeed before each function in the TOE scope of control is allowed to proceed (in this case the resource requested).
O.AUTHENTICATE	The objective to provide the means to authenticate individuals using multiple authentication mechanisms is met by the following security requirements: • FIA_AFL.1 requires that a user is correctly authenticated within a defined number of attempts, else the user's account is locked and an alarm is issued. • FIA_UAU.1 and FIA_UID.2 requires that all users requesting access to resources protected by the TOE are identified and if required authenticated for access to those resources. • FIA_UAU.5 supports FIA_UAU.1 by requiring support for multiple authentication mechanisms that can be applied to a request for access to a resource protected by the TOE. • FIA_UAU.6 supports FIA_UAU.1 by requiring that re-authentication occurs if and when required for a request to access resources protected by the TOE.

Objectives	Argument to support sufficiency of Security Requirements
O.AUDIT	The objective to provide the means of detecting and recording security relevant events is met by the following security requirements: • FAU_ARP.1 supports FAU_SAA.1 by requiring that automated alarms may be generated in the event that potential security violations are detected using the collected audit data. This security requirement helps to ensure that audit data is actively used to defend the TOE, and the resources that it protects. • FAU_GEN.1 requires the capability to generate records of security relevant events, including the identity of the user responsible in order to be able to hold a user accountable for their actions. • FAU_GEN.2 supports FAU_GEN.1 by requiring that the identity of the user that caused the event is associated with each audit record. • FAU_SAA.1 requires that audit functions include the ability to detect potential security violations from the collected audit data according to rules defined by administrators • FAU_SAR.1 supports FAU_GEN.1 and FAU_GEN.2 by requiring that authorised administrators are able to read audit information from the audit records. • FAU_SAR.3 supports FAU_SAR.1 by requiring that audit records may be sorted and/or filtered based on logical rules applied to audit record attributes. • FAU_SEL.1 supports FAU_GEN.1 by allowing control of the level of audit information captured by the TOE. • FPT_STM.1 supports FAU_GEN.1 by requiring that reliable time stamps can be generated to associate with security-relevant events. • FAU_STG.1 and FCS_COP.1(2) together support FAU_GEN.1 by requiring that audit entries are protected against unauthorised deletion and that modifications can be detected through use of digital signatures. • FMT_MOF.1 (1) and FMT_MTD.1 require that the ability to manage the audit functions and the audit trail be restricted to administrators. These security requirements help to ensure that appropriate audit data is collected and maintained by the TOE.

Objectives	Argument to support sufficiency of Security Requirements
O.ADMIN	The objective to provide the means for administrators to effectively manage TOE security functions and delegate administrative functions is met by the following security requirements: • FMT_MOF.1(1) requires the capability to manage TOE security functions (audit, authentication and authorisation) are restricted to administrators. • FMT_MOF.1(2) requires the capability for those individuals delegated with administrative functions to manage the authentication and authorisation functions is restricted to only those individuals. • FMT_MSA.1(1) and FMT_MSA.1(2) limit the management of TOE security attributes to the Administrator and to Delegated Administrators.. • FMT_MSA.2 supports FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM.2, FCS_CKM.4, FCS_COP.1(1), and FCS_COP.1(2) by requiring that only secure values of those security attributes may be used within the TOE • FMT_MSA.3 supports FDP_ACF by specifying that no user can modify initial values for the security attributes required to maintain the TOE in a secure state. • FMT_MTD.1 supports FMT_MOF.1(1) by requiring restricting the ability to manage TSF data to administrators. • FMT_SMR.1 supports FMT_MOF.1(1), FMT_MOF.1(2) and FMT_MTD.1 by requiring that the TOE maintain the roles of Administrator and Delegated Administrator. • FMT_SMF.1 requires the capability for administrators to manage the TOE through TOE management functions. • FCS_COP.1(2) supports FMT_MOF.1(1), and FMT_MOF.1(2) by requiring that policy changes made by the Administrator role and the Delegated Administrator role are digitally signed. In addition, the audit records of such policy changes and of administrator logins are digitally signed within the audit store.
O.ENCRYPT	The objective to provide the means to protect the confidentiality and integrity of information transmitted between the TOE and resources it protects, and between components of the TOE is met by the following security requirements: • FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM.2(1) and FCS.CKM.4 require the capability to generate, distribute and destroy cryptographic keys used for confidentiality and integrity of data transmission between the Administration Server and the Web browser used by administrators to access the Policy Builder. • FCS_COP.1(1) requires the capability to perform both 3DES encryption and decryption for confidentiality of data transmissions and RSA encryption and decryption for the integrity of TSF data in transit between the Administration Server and the Web browser used by administrators to access the Policy Builder. • FCS_CKM.1(1), FCS_CKM.1(3), FCS_CKM.2(2) and FCS.CKM.4 require the capability to generate, distribute and destroy cryptographic keys used for confidentiality and integrity of data transmission between TOE components. • FCS_COP.1(3) requires the capability to perform both AES encryption and decryption for confidentiality of data transmissions and RSA encryption and decryption for the integrity of TSF data in transit between TOE components. • FPT_ITT.1 requires the capability to protect data in transit between separate parts of the TOE.

Objectives	Argument to support sufficiency of Security Requirements
O.INTEGRITY	The objective to provide the means to protect the integrity of stored authorisation and authentication information is met by the following security requirements: • FCS_COP.1(2) requires the capability to perform digital signature operations on stored authorisation and authentication information.
O.FAIL_SAFE	The objective to provide the means of ensuring that access to IT resources protected by the TOE is not granted in the event of a failure of a Validator is met by the following security requirement: • FPT_FLS.1 requires that a secure state is preserved when the TOE is unable to determine a response to a request to access a controlled resource.
OE.NETWORK	The objective to ensure that appropriate logical access controls exist in the TOE environment is met by the following security requirements: • ENV_NONIT.3 ensures that procedures are defined in the TOE environment for an Administrator to effectively manage the underlying operating system for each of the TOE components, and that access to TOE components is limited to only authorised administrators. • ENV_NONIT.4 ensures that procedures and guidance are provided in the TOE environment for the effective administration of the infrastructure surrounding and supporting the TOE.
OE.PHYS_ENV	The objective to ensure that appropriate physical access controls exist in the TOE environment is met by the following security requirements: • ENV_NONIT.1 ensures that TOE components are located in a controlled access facility to prevent unauthorised access.
OE.TRAIN	The objective to ensure that the Administrator and Delegated Administrators (as defined by FPT_STM.1) are appropriately trained and motivated is met by the following security requirements: • ENV_NONIT.2 ensures that the Administrator and those individuals delegated with administrative functions are trained, and includes security awareness of vulnerabilities and familiarisation with supporting technologies.
OE.CREDENTIALS	The objective to ensure that an appropriate means of generating, distributing and managing TOE authentication credentials is met by the following security requirements: • ENV_NONIT.5 ensures that an appropriate means of securely generating, distributing and managing TOE authentication credentials exists in the TOE operating environment.

8.2.3 Satisfaction of Dependencies

Table 19 shows the dependencies between the functional requirements. All of the dependencies are satisfied. Note that:

(H) indicates the dependency is satisfied through the inclusion of a component that is hierarchical to the one required.

(*) indicates that this dependency is not satisfied by the TOE. Refer to the supporting rationale following Table 19.

Table 19: Dependency Analysis

Component Reference	Requirement	Dependencies	Dependency Reference
Functional Requirements			
1	FAU_ARP.1	FAU_SAA.1	4
2	FAU_GEN.1	FPT_STM.1	36
3	FAU_GEN.2	FUA_GEN.1, FIA_UID.1	2, 24(H)
4	FAU_SAA.1	FAU_GEN.1	2
5	FAU_SAR.1	FAU_GEN.1	2
6	FAU_SAR.3	FAU_SAR.1	5
7	FAU_SEL.1	FAU_GEN.1, FMT_MTD.1	2, 30
8	FAU_STG.1	FAU_GEN.1	2
9	FCS_CKM.1(1)	FCS_CKM.2, FCS_CKM.4, FMT_MSA.2	12/13, 14, 29
10	FCS_CKM.1(2)	FCS_CKM.2, FCS_CKM.4, FMT_MSA.2	12/13, 14, 29
11	FCS_CKM.1(3)	FCS_CKM.2, FCS_CKM.4, FMT_MSA.2	12/13, 14, 29
12	FCS_CKM.2(1)	FCS_CKM.1, FCS_CKM.4, FMT_MSA.2	9/10, 14, 29
13	FCS_CKM.2(2)	FCS_CKM.1, FCS_CKM.4, FMT_MSA.2	9/11, 14, 29
14	FCS_CKM.4	FCS_CKM.1, FMT_MSA.2	9/10/11, 29
15	FCS_COP.1(1)	FCS_CKM.1, FCS_CKM.4, FMT_MSA.2	9/10, 14, 29
16	FCS_COP.1(2)	FCS_CKM.1, FCS_CKM.4, FMT_MSA.2	9, 14, 29
17	FCS_COP.1(3)	FCS_CKM.1, FCS_CKM.4, FMT_MSA.2	9/11, 14, 29
18	FDP_ACC.2	FDP_ACF.1	19
19	FDP_ACF.1	FDP_ACC.1, FMT_MSA.3	18(H), 30
20	FIA_AFL.1	FIA_UAU.1	21

Component Reference	Requirement	Dependencies	Dependency Reference
21	FIA_UAU.1	FIA_UID.1	24(H)
22	FIA_UAU.5	No dependencies	-
23	FIA_UAU.6	No dependencies	-
24	FIA_UID.2	No dependencies	-
25	FMT_MOF.1(1)	FMT_SMF.1, FMT_SMR.1.1	32, 33
26	FMT_MOF.1(2)	FMT_SMF.1, FMT_SMR.1	32, 33
27	FMT_MSA.1(1)	FDP_ACC.1, FMT_SMF.1, FMT_SMR.1	18(H), 32, 33
28	FMT_MSA.1(2)	FDP_ACC.1, FMT_SMF.1, FMT_SMR.1	18(H), 32, 33
29	FMT_MSA.2	FDP_ACC.1, FMT_MSA.1, FMT_SMR.1, ADV_SPM.1*	18(H), 27, 33, -
30	FMT_MSA.3	FMT_MSA.1, FMT_SMR.1	27, 33
31	FMT_MTD.1	FMT_SMF.1, FMT_SMR.1	31. 33
32	FMT_SMF.1	No dependencies	-
33	FMT_SMR.1	FIA_UID.1	24(H)
34	FPT_FLS.1	ADV_SPM.1*	-
35	FPT_ITT.1	No dependencies	-
36	FPT_RVM.1	No dependencies	-
37	FPT_STM.1	No dependencies	-
Assurance Requirements			
38	ACM_CAP.2	No dependencies	-
39	ADO_DEL.1	No dependencies	-
40	ADO_IGS.1	AGD_ADM.1	44
41	ADV_FSP.1	ADV_RCR.1	43
42	ADV_HLD.1	ADV_FSP.1, ADV_RCR.1	41, 43
43	ADV_RCR.1	No dependencies	-
44	AGD_ADM.1	ADV_FSP.1	41
45	AGD_USR.1	ADV_FSP.1	41
46	ATE_COV.1	ADV_FSP.1, ATE_FUN.1	41, 47
47	ATE_FUN.1	No dependencies	-
48	ATE_IND.2	ADV_FSP.1, AGD_ADM.1, AGD_USR.1, ATE_FUN.1	41, 44, 45, 47
49	AVA_SOF.1	ADV_FSP.1, ADV_HLD.1	41, 42
50	AVA_VLA.1	ADV_FSP.1, ADV_HLD.1, AGD_ADM.1, AGD_USR.1	41, 42, 44, 45

The dependencies on ADV_SPM.1 are not satisfied in this Security Target because all of the requirements for maintaining the TOE in a secure state are explicitly stated within the Security Target.

ADV_SPM.1 is identified as a dependency for FPT_FLS.1 and FMT_MSA.2.

In the context of FPT_FLS.1, the TOE will maintain a secure state by denying access to a resource whenever a TOE response cannot be determined for a request from an external entity to access a controlled resource. This informally states the rules for access by subjects to objects in the case of a failure in the TSF, such that the TSP is preserved.

In relation to FMT_MSA.2, the allowable values of the cryptographic security attributes required to maintain the TOE in a secure state:

- Are explicitly stated in FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM.1(3), FCS_CKM.2(1), FCS_CKM.2(2), FCS_CKM.4, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), and FMT_MSA.1; and
- They are secure values,

Therefore, the requirement FPT_FLS.1 and FMT_MSA.2 are met without satisfying this dependency.

8.3 TOE Summary Specification Rationale

8.3.1 IT security functions satisfy the SFRs.

Table 20 and Table 21 show that each SFR is mapped to at least one IT security function and each IT security function is mapped to at least one SFR.

Table 20: Mapping of SFRs to IT Security Functions

Security Functional Requirement	IT Security Function
FAU_ARP.1	AUDIT
FAU_GEN.1	AUDIT
FAU_GEN.2	AUDIT
FAU_SAA.1	AUDIT
FAU_SAR.1	AUDIT
FAU_SAR.3	AUDIT
FAU_SEL.1	AUDIT
FAU_STG.1	AUDIT SIGN
FCS_CKM.1(1)	VALIDATE ADMIN AUDIT
FCS_CKM.1(2)	ADMIN AUDIT

Security Functional Requirement	IT Security Function
FCS_CKM.1(3)	ENFORCE VALIDATE ADMIN AUDIT
FCS_CKM.2(1)	ADMIN AUDIT
FCS_CKM.2(2)	ENFORCE VALIDATE ADMIN AUDIT
FCS_CKM.4	ENFORCE VALIDATE ADMIN AUDIT
FCS_COP.1(1)	ADMIN AUDIT
FCS_COP.1(2)	VALIDATE ADMIN AUDIT SIGN
FCS_COP.1(3)	ENFORCE VALIDATE ADMIN AUDIT
FDP_ACC.2	VALIDATE ENFORCE
FDP_ACF.1	VALIDATE ENFORCE POLICY
FIA_AFL.1	POLICY AUDIT
FIA_UAU.1	ENFORCE VALIDATE ADMIN
FIA_UAU.5	ENFORCE VALIDATE ADMIN
FIA_UAU.6	ENFORCE VALIDATE
FIA_UID.2	ENFORCE VALIDATE ADMIN

Security Functional Requirement	IT Security Function
FMT_MOF.1(1)	POLICY ADMIN DELEGATE
FMT_MOF.1(2)	DELEGATE
FMT_MSA.1(1)	ADMIN
FMT_MSA.1(2)	DELEGATE
FMT_MSA.2	ADMIN
FMT_MSA.3	ADMIN POLICY
FMT_MTD.1	ADMIN
FMT_SMR.1	ADMIN DELEGATE
FMT_SMF.1	ADMIN DELEGATE POLICY
FPT_FLS.1	ENFORCE
FPT_ITT.1	ENFORCE VALIDATE ADMIN AUDIT
FPT_RVM.1	ENFORCE
FPT_STM.1	AUDIT

Table 21: Mapping of IT Security Functions to SFRs

IT Security Function	Security Functional Requirement
VALIDATE	FCS_CKM.1(1) FCS_CKM.1(3) FCS_CKM.2(2) FCS_CKM.4 FCS_COP.1(2) FCS_COP.1(3) FDP_ACC.2 FDP_ACF.1 FIA_UAU.1 FIA_UAU.5 FIA_UAU.6 FIA_UID.2 FPT_ITT.1
ENFORCE	FCS_CKM.1(1) FCS_CKM.1(3) FCS_CKM.2(2) FCS_CKM.4 FCS_COP.1(3) FDP_ACC.2 FDP_ACF.1 FIA_UAU.1 FIA_UAU.5 FIA_UAU.6 FIA_UID.2 FPT_FLS.1 FPT_ITT.1 FPT_RVM.1
POLICY	FDP_ACF.1 FIA_AFL.1 FMT_MOF.1(1) FMT_MSA.3 FMT_SMF.1
SIGN	FAU_STG.1 FCS_COP.1(2)

DELEGATE	FMT_MOF.1(1)
	FMT_MOF.1(2)
	FMT_MSA.1(2)
	FMT_SMR.1
	FMT_SMF.1

IT Security Function	Security Functional Requirement

ADMIN	FCS_CKM.1(1)
	FCS_CKM.1(2)
	FCS_CKM.1(3)
	FCS_CKM.2(1)
	FCS_CKM.2(2)
	FCS_CKM.4
	FCS_COP.1(1)
	FCS_COP.1(2)
	FCS_COP.1(3)
	FIA_UAU.1
	FIA_UAU.5
	FIA_UID.2
	FMT_MOF.1(1)
	FMT_MSA.1(1)
	FMT_MSA.2
	FMT_MSA.3
	FMT_MTD.1
	FMT_SMR.1
	FMT_SMF.1
	FPT_ITT.1

AUDIT	FAU_ARP.1 FAU_GEN.1 FAU_GEN.2 FAU_SAA.1 FAU_SAR.1 FAU_SAR.3 FAU_SEL.1 FAU_STG.1 FCS_CKM.1(1) FCS_CKM.1(2) FCS_CKM.1(3) FCS_CKM.2(1) FCS_CKM.2(2) FCS_CKM.4 FCS_COP.1(1) FCS_COP.1(2) FCS_COP.1(3) FIA_AFL.1 FPT_ITT.1 FPT_STM.1

8.3.2 IT Security Function Suitability

Table 22 provides appropriate justification that the IT Security Functions are suitable to meet the TOE Security Functional Requirement and that when implemented, contributes to meeting that requirement.

Table 22: Suitability of IT Security Functions

Security Functional Requirement	Argument for suitability of IT Security Functions
FAU_ARP.1	The TOE SFR is satisfied by the IT Security Functions AUDIT as: • AUDIT provides the administrator with the ability to generate automated alerts from the audit data.
FAU_GEN.1	The TOE SFR is satisfied by the IT Security Function AUDIT as: • the function provides for the generation of audit records, which are written to an audit trail.
FAU_GEN.2	The TOE SFR is satisfied by the IT Security Function AUDIT as: • the function associates user identity with audit records.
FAU_SAA.1	The TOE SFR is satisfied by the IT Security Functions AUDIT as: • AUDIT provides the administrator with the ability to detect potential security violations from the audit data.
FAU_SAR.1	The TOE SFR is satisfied by the IT Security Functions AUDIT as: • AUDIT provides the administrator with the ability to review the audit data.
FAU_SAR.3	The TOE SFR is satisfied by the IT Security Function s AUDIT as: • AUDIT provides the administrator the ability to sort and/or filter and review the audit data based on selection by logical rules.
FAU_SEL.1	The TOE SFR is satisfied by the IT Security Function AUDIT as: • the function provides the means for the Administrator to configure the collection of audit information based on host identity and event type, and to output the audit trail information to different formats.
FAU_STG.1	The TOE SFR is satisfied by the IT Security Functions AUDIT and SIGN as: • the AUDIT function provides for the signing of audit trail entries such that a modification or deletion of audit records can be detected. • The SIGN function supports the AUDIT function as the TOE uses digital signatures based upon X.509 certificates prior to storage of audit data.
FCS_CKM.1(1)	The TOE SFR is satisfied by the IT Security Functions VALIDATE, ADMIN and AUDIT as: • the functions all provide for the generation of RSA public/private key pairs used by TOE components for digital signature generation. • the functions ADMIN and AUDIT provide for the generation of RSA public/private key pairs used by the Administration Server and the Administrator's browser for 3DES session key exchange.
FCS_CKM.1(2)	The TOE SFR is satisfied by the IT Security Functions ADMIN and AUDIT as: • the functions all provide for the generation of 3DES session keys used for securing communication between the Administration Server and the Administrator's browser using SSLv3.0.

Security Functional Requirement	Argument for suitability of IT Security Functions
FCS_CKM.1(3)	The TOE SFR is satisfied by the IT Security Functions ENFORCE, VALIDATE, ADMIN and AUDIT as: • the functions all provide for the generation of AES session keys used for securing communication between distributed TOE components using TLS v1.0. • the functions all provide for the generation of RSA public/private key pairs used by TOE components for AES session key exchange.
FCS_CKM.2(1)	The TOE SFR is satisfied by the IT Security Functions ADMIN and AUDIT as: • the functions all provide for the agreement of 3DES session keys used for securing communication between the Administration Server and the Administrator's browser using RSA.
FCS_CKM.2(2)	The TOE SFR is satisfied by the IT Security Functions ENFORCE, VALIDATE, ADMIN and AUDIT as: • the functions all provide for the agreement of AES session keys used for securing communication between distributed TOE components using RSA.
FCS_CKM.4	The TOE SFR is satisfied by the IT Security Functions ENFORCE, VALIDATE, ADMIN and AUDIT as: • the functions all provide for the destruction of 3DES session keys for securing communication between distributed TOE components, and for RSA public/private key pairs by overwriting cryptographic material.
FCS_COP.1(1)	The TOE SFR is satisfied by the IT Security Functions ADMIN, and AUDIT as: • ADMIN and AUDIT all provide for encryption/decryption of TOE data communicated between the Administration Server and the Administrator's browser using SSL v3.0.
FCS_COP.1(2)	The TOE SFR is satisfied by the IT Security Functions VALIDATE,ADMIN, AUDIT and SIGN as: • VALIDATE ADMIN and AUDIT provide for the generation and verification of digital signatures using RSA. • SIGN provides for the generation of digital signatures using RSA.
FCS_COP.1(3)	The TOE SFR is satisfied by the IT Security Functions ENFORCE, VALIDATE, ADMIN, and AUDIT as: • ENFORCE, VALIDATE, ADMIN and AUDIT all provide for encryption/decryption of TOE data communicated between components using TLS v1.0.
FDP_ACC.2	The TOE SFR is satisfied by the IT Security Functions ENFORCE and VALIDATE as: • VALIDATE provides the means to determine whether access to any resource protected by the TOE is permitted for an external entity, and what means of authentication may be required, in accordance with the Authorisation SFP. • ENFORCE provides the means to enforce the authorisation decision from the VALIDATE function, and to request authentication if required by the Authorisation SFP rules for access to any resource protected by the TOE.

Security Functional Requirement	Argument for suitability of IT Security Functions
FDP_ACF.1	The TOE SFR is satisfied by the IT Security Functions ENFORCE, VALIDATE and POLICY as: • VALIDATE provides the means to determine whether access to any resource protected by the TOE is permitted for an external entity, and what means of authentication may be required. • ENFORCE provides the means to enforce the authorisation decision from the VALIDATE function, and to request authentication if required by the Authorisation SFP rules for access to any resource protected by the TOE. • POLICY provides the means to specify the rules that the VALIDATE function will use to determine whether access to a resource will be granted or denied through the policy management matrix.
FIA_AFL.1	The TOE SFR is satisfied by the IT Security Functions POLICY and AUDIT as: • POLICY provides the means for the Administrator to manage and enforce the password authentication policy applied to the TOE. • AUDIT provides the means by which administrators can be alerted to the occurrence of failed password authentication attempts.
FIA_UAU.1	The TOE SFR is satisfied by the IT Security Functions ENFORCE, VALIDATE and ADMIN as: • ENFORCE and VALIDATE together provide the means to authenticate external entities if required by the rules for access as defined in the policy management matrix. The VALIDATE function also provides the means of authenticating Delegated Administrators prior to access being granted to the DELEGATE function. • ADMIN provides the means to authenticate the Administrator by password; and to authenticate Delegated Administrators by password and the role-based delegated administrator certificate.
FIA_UAU.5	The TOE SFR is satisfied by the IT Security Functions ENFORCE, VALIDATE and ADMIN as: • ENFORCE and VALIDATE together provide the means to authenticate external entities if required by the rules for access as defined in the policy management matrix using the authentication mechanism defined required for access to that resource. Supported authentication mechanisms are password and X.509 certificates. • ADMIN provides the means to authenticate the Administrator by password; and to authenticate Delegated Administrators by password and the role-based delegated administrator certificate.
FIA_UAU.6	The TOE SFR is satisfied by the IT Security Functions ENFORCE and VALIDATE as: • VALIDATE provides the means to re-authenticate by maintaining state information that allows the TOE to require external entities to re-authenticate as defined by the Administrator. • ENFORCE supports VALIDATE by requesting authentication information as required by the VALIDATE function and enforcing the results of the external entity request for access to a resource protected by the TOE.

Security Functional Requirement	Argument for suitability of IT Security Functions
FIA_UID.2	The TOE SFR is satisfied by the IT Security Functions ENFORCE, VALIDATE and ADMIN as: • ENFORCE and VALIDATE together ensure that an external entity requesting access to a resource protected by the TOE is identified before access to that resource is granted. VALIDATE also ensures that a Delegated Administrator is identified prior to access to the DELEGATE function being granted. • ADMIN ensures that the Administrator is identified by supplying a user ID; and that Delegated Administrators are identified by supplying a user ID and their role-based administrator certificate prior to gaining access to administrative functions.
FMT_MOF.1(1)	The TOE SFR is satisfied by the IT Security Functions POLICY, ADMIN and DELEGATE as: • POLICY provides the means for administrators to manage the policy management matrix that defines the Authorisation SFP for the TOE. • ADMIN provides the means for the Administrator to manage the behaviour of the audit function of the TOE and the authentication and authorisation functions of the TOE. • DELEGATE provides the means for administrators to delegate (i.e. authorise) administrative duties to specific TOE Users.
FMT_MOF.1(2)	The TOE SFR is satisfied by the IT Security Function DELEGATE as: • DELEGATE provides the means for Delegated Administrators to manage the policy management matrix that defines the Authorisation SFP for the TOE, for only that part of the policy management matrix for which they have been delegated administrative duties.
FMT_MSA.1(1)	The TOE SFR is satisfied by the IT Security Functions ADMIN as: • ADMIN limits the complete management of the Policy Matrix that controls access to TOE-protected resources to the Administrator.
FMT_MSA.1(2)	The TOE SFR is satisfied by the IT Security Functions DELEGATE as: • DELEGATE limits the management by the Delegated Administrator to a defined subset of the Policy Matrix that controls access to TOE-protected resources.
FMT_MSA.2	The TOE SFR is satisfied by the IT Security Functions ADMIN as: • ADMIN ensures that only secure values of the cryptographic security attributes required to maintain the TOE in a secure state may be introduced.
FMT_MSA.3	The TOE SFR is satisfied by the IT Security Functions POLICY and ADMIN as: • POLICY provides for restrictive initialisation values for access security attributes (viz. no access), which must be explicitly modified (directly or by inheritance) by the administrators. • ADMIN provides the only means for administrators to modify the restrictive initialisation values of the access security attributes required to maintain the TOE in a secure state.

Security Functional Requirement	Argument for suitability of IT Security Functions
FMT_MTD.1	The TOE SFR is satisfied by the IT Security Functions ADMIN as: • ADMIN provides the means for the Administrator to manage the audit data and configuration information such as resources, users and groups, authentication and encryption requirements.
FMT_SMR.1	The TOE SFR is satisfied by the IT Security Functions ADMIN and DELEGATE as: • ADMIN provides the means for the Administrator to identify administrators for access. Further, ADMIN provides the means to assign users with the Delegated Administrator role. • DELEGATE provides the means for Delegated Administrators to further delegate administrative functions (i.e. create sub-ordinate Delegate Administrators) for that part of the policy management matrix that they have been allocated administrative duties.
FMT_SMF.1	The TOE SFR is satisfied by the IT Security Functions ADMIN, DELEGATE and POLICY as: • ADMIN provides the means for the Administrator to manage the TOE security functions, including the delegation of administrative duties. • DELEGATE provides the means for Delegated Administrators to manage a defined subset of the TOE security functions, including the ability to further delegate administrative functions. • POLICY provides the means by which administrators (the Administrator and Delegated Administrators) manage and enforce organisational security policy.
FPT_FLS.1	The TOE SFR is satisfied by the IT Security Function ENFORCE as: • ENFORCE ensures that if a response to a request to access a resource protected by the TOE is not received from the VALIDATE function that access to the requested resource is denied.
FPT_ITT.1	The TOE SFR is satisfied by the IT Security Functions ENFORCE, VALIDATE, ADMIN and AUDIT as: • ENFORCE, VALIDATE, ADMIN and AUDIT together provide the means to protect the confidentiality of data transferred between components using 3DES over an TLSv1.0 session.
FPT_RVM.1	The TOE SFR is satisfied by the IT Security Function ENFORCE as: • this function ensures that every external entity requesting access to a resource protected by the TOE is passed to the VALIDATE function and a response received before access to that resource may be granted.
FPT_STM.1	The TOE SFR is satisfied by the IT Security Function AUDIT as: • this function generates and associates timestamps with each security-relevant event written to the audit trail. AUDIT uses the Secure Audit Server system time as the clock source.

8.3.3 Demonstration of Mutual Support

The dependency analysis provided at Table 19 and the analyses provided in Table 20, Table 21 and Table 22 demonstrate that the IT security functions work together to satisfy the TSFs, that is, they demonstrate mutual support between function components.

The primary function of the TOE, namely authorisation policy enforcement, is provided by the SFRs from the FIA and FDP class. These SFRs provide for the identification and authentication of external entities requesting access to controlled resources and the granting or denying of access to those resources based on a defined access control policy. The SFRs selected from the FAU class provide the auditing functions in support of the FIA and FDP requirements by detecting security relevant events that might indicate a potential compromise of those functions, and alerting administrators. These are in turn supported by the SFRs from the FMT, FCS and FPT classes as follows:

- SFRs from the FMT class provide Administrator and Delegated Administrator functions to support the secure management of TOE security functions and of TSF data such as the Authorisation SFP (through the policy management matrix) and the audit trail upon which FIA, FDP and FAU SFRs depend;
- SFRs from the FPT class provide appropriate protection of the TSF, preventing bypass of the TOE security policy (FPT_RVM.1), protecting TSF data (FPT_ITT.1) and supplying reliable time stamps (FPT_STM.1); and
- SFRs from the FCS class provide support to FDP, FAU and FPT class components as the confidentiality of audit data and TSF data transferred between TOE components is protected through encryption.

By definition, all assurance requirements support all SFRs since they provide confidence in the correct implementation and operation of the SFRs.

This analysis of the security functional and assurance requirements demonstrates that there are no conflicts between requirements. Therefore, the security requirements together form a mutually supportive and consistent whole.

8.3.4 Security Assurance Requirements Rationale

Table 23 below shows that all Security Assurance Requirements are met by the assurance measures.

Table 23: Mapping of SARs to Assurance Measures

Security Assurance Requirements	Assurance Measures
ACM_CAP.2	CM_DOC
ADO_DEL.1	DEL_DOC
ADO_IGS.1	IGS_DOC
ADV_FSP.1	FUN_SPEC
ADV_HLD.1	HLD_DOC
ADV_RCR.1	RCR_DOC
AGD_ADM.1	ADMIN
AGD_USR.1	USER
ATE_COV.1	TEST_COV
ATE_FUN.1	TEST_DOC
ATE_IND.2	TEST_DOC
AVA_SOF.1	SOF_DOC
AVA_VLA.1	VLA_DOC

Given that all Security Assurance Requirements are met by at least one Assurance Measure and that the implementation of each Assurance Measure will be the subject of evaluation activities, it is concluded that all of the Assurance Measures will meet all of the Security Assurance Requirements.

Given the TOE environment and security objectives outlined above (Sections 3 and 4 respectively), it could be argued that the appropriate assurance level for the TOE would be determined by the value of the assets the TOE is meant to protect. However, considering the value of the assets alone is not sufficient for determining the appropriate assurance level for the TOE. There are measures defined for the environment (described in Section 3 and illustrated in Figure 1) that significantly decrease the risks to the IT assets. These measures include:

- It is intended that the TOE components be sited in a physically secure environment that has constraints on the access of unauthorised individuals.
- It is intended that appropriate logical protection be provided to all TOE components. For example, use of a network firewall between network devices with the Enforcer component installed and the other TOE Components.
- It is intended that an appropriate means of securely generating, distributing and managing authentication credentials exists within the TOE environment to reduce the risk of compromise of these credentials.

- The associated cryptography is outside the scope of CC evaluation, and therefore subject to assessment for 'fitness of purpose' by the National Authority for cryptography. This assessment will determine that the strength of cryptographic functions, as specified within this Security Target is appropriate for the intended environment of the TOE, within the jurisdiction of the National Authority.

Given that the residual risks to the IT assets have been partially (and significantly) mitigated by the security measures in the environment, the attractiveness of the assets is similarly reduced. Thus, the combination of the reduced attractiveness of the IT assets and the security measures provided by the environment, it is considered that an EAL-2 level of assurance is entirely appropriate for the intended application of the TOE.

8.3.5 Strength of function claims

At CC EAL-2, the TOE security assurance requirements include the AVA_SOF.1 component. The minimum strength of function claim for the TOE security functional requirements is *SOF_basic*. A strength of function claim is appropriate for all TOE security functions that implement support for password and X.509 certificate authentication mechanisms. This applies to the following TOE Security Functions:

- VALIDATE; and
- ADMIN.

VALIDATE implements the FIA_UAU.1 security functional requirement, with authentication enforced through a password or X.509 certificate mechanism depending upon the rules defined by the Authorisation SFP. The support for multiple authentication mechanisms allows the Administrator to define stronger authentication requirements (e.g. use X.509 certificates instead of passwords) as appropriate for the value of the resources protected by the TOE, as part of the Authorisation SFP. Given that the X.509 certificate mechanism is implemented by cryptographic components, which falls outside the scope of the [CC], this will be subject to independent evaluation by the National Authority.

For the password mechanism, the requirements for strong physical and logical access controls for the TOE components limit an attackers access to the TOE and increase the level of expertise required to mount a successful attack. The requirements in the TOE environment for a secure means of generating good authentication credentials, and securely distributing and managing those credentials further increase the level of expertise required to mount a successful attack, and increase the elapsed time required to exploit the password mechanism. Therefore, a claim of *SOF-basic* is appropriate for the requirement FIA_UAU.1 and the VALIDATE function.

The ADMIN function also implements FIA_UAU.1 requiring the Administrator to be authenticated by password and requiring Delegated Administrators to be authenticated by password and also through the role-based X.509 administrator certificate. Given that authentication is implemented using both mechanisms, and that strong physical and logical controls required for TOE components, a strength of function claim of

SOF-basic is appropriate for the ADMIN function. The X.509 certificate authentication mechanism is implemented by cryptographic components. The cryptographic algorithms and values used in these components are specified within this Security Target, and will be subject to evaluation by the National Authority regarding their "fitness for purpose" within the National Authority's jurisdiction.

The ENFORCE function also traces to the implementation of FIA_UAU.1. However, the ENFORCE function merely prompts for authentication information and passes this information onto the VALIDATE function, and finally enforces the response from the VALIDATE function. Thus, the ENFORCE function only supports authentication performed by the VALIDATE function and, therefore, does not require an SOF claim.

8.4 Rationale for Extensions

Not applicable.

8.5 PP Claims Rationale

This ST makes no PP conformance claim therefore no rationale is required.

Appendix A: Delivery CD-ROM contents

Table 24: Configuration List for HP OpenView Select Access v5.2 software

Modified	Size(Bytes):	Name
Volume is Sea52R Volume Serial Number is 3C9C-D6EC		
Directory Of <CD-ROM>:		
29/03/04 06:31	<DIR>	.
29/03/04 06:31	<DIR>	..
29/03/04 06:29	<DIR>	docs
29/03/04 06:30	<DIR>	schema
29/03/04 06:31	<DIR>	solutions
28/10/03 07:31	56	autorun.inf
28/10/03 07:31	27,682	install_sidebar.gif
28/10/03 07:31	77,471	install_splash.jpg
28/10/03 07:31	71,094	license.pdf
28/10/03 07:31	2,238	selectaccess.ico
28/10/03 07:27	86,373,830	setup_hpux
28/10/03 07:23	53,514,394	setup_linux
30/10/03 07:34	92,505,631	setup_solaris
28/10/03 07:31	67,711,196	setup_win32.exe
28/10/03 07:31	61	version.txt
15 File(s)	300,283,653 bytes	
Directory of <CD-ROM>:\docs		
29/03/04 06:29	<DIR>	.
29/03/04 06:31	<DIR>	..
29/03/04 06:29	<DIR>	developer
29/03/04 06:29	<DIR>	InstallGuide
29/03/04 06:29	<DIR>	network
29/03/04 06:29	<DIR>	PolicyBuilder
29/03/04 06:29	<DIR>	Release_Notes
29/03/04 06:29	<DIR>	solutions
28/10/03 07:31	2,919	index.html
28/10/03 07:31	2,244	logo3cmwide300dpirgb.gif
10 File(s)	5,163 bytes	

```
Modified           Size(Bytes):      Name

Directory of <CD-ROM>:\docs\InstallGuide
29/03/04  06:29        <DIR>          .
29/03/04  06:29        <DIR>          ..
22/03/04  12:09            4,309,095 installation_guide.pdf
               3 File(s)     4,309,095 bytes
Directory of <CD-ROM>:\docs\PolicyBuilder
29/03/04  06:29        <DIR>          .
29/03/04  06:29        <DIR>          ..
22/03/04  12:10            3,834,736 policy_builder_guide.pdf
               3 File(s)     3,834,736 bytes
Directory of <CD-ROM>:\docs\Release_Notes
29/03/04  06:29        <DIR>          .
29/03/04  06:29        <DIR>          ..
28/10/03  07:31              414,637 relnotes.pdf
               3 File(s)       414,637 bytes
Directory of <CD-ROM>:\docs\developer
29/03/04  06:29        <DIR>          .
29/03/04  06:29        <DIR>          ..
28/10/03  07:31            1,426,444 developers_guide.pdf
               3 File(s)     1,426,444 bytes
Directory of <CD-ROM>:\docs\network
22/03/04  14:55        <DIR>          .
22/03/04  14:55        <DIR>          ..
28/10/03  10:31            1,408,156 network_integration_guide.pdf
               3 File(s)     1,408,156 bytes
Directory of <CD-ROM>:\docs\solutions
29/03/04  06:29        <DIR>          .
29/03/04  06:29        <DIR>          ..
28/10/03  07:31              505,086 a360_enrole.pdf
28/10/03  07:31              338,052 apache_2.pdf
28/10/03  07:31              519,466 apache_examples.pdf
```

```
Modified              Size(Bytes):     Name
28/10/03  07:31           367,490 apache_reverse_proxy.pdf
28/10/03  07:31           495,561 citrix_nfuse.pdf
28/10/03  07:31           391,258 domino.pdf
28/10/03  07:31           405,173 hp.pdf
28/10/03  07:31           351,413 iplanet.pdf
28/10/03  07:31           362,478 Oracle_app.pdf
28/10/03  07:31           368,768 outlook_webaccess.pdf
28/10/03  07:31           423,097 plumtree_portal.pdf
28/10/03  07:31           473,147 sa_servlet_plugin.pdf
28/10/03  07:31         1,201,730 saml_solutions_guide.pdf
28/10/03  07:31           400,842 siebel.pdf
28/10/03  07:31           399,738 silverstream.pdf
28/10/03  07:31           414,120 weblogic.pdf
28/10/03  07:31           363,137 websphere.pdf
              19 File(s)      7,780,556 bytes

Directory of <CD-ROM>:\schema

29/03/04  06:30       <DIR>          .
29/03/04  06:31       <DIR>          ..
29/03/04  06:30       <DIR>          Active-Directory
29/03/04  06:30       <DIR>          Aphelion
29/03/04  06:29       <DIR>          CA-eTrust
29/03/04  06:30       <DIR>          Critical-Path
29/03/04  06:29       <DIR>          Siemens-DirX
               7 File(s)              0 bytes

Directory of <CD-ROM>:\schema\Active-Directory

29/03/04  06:30       <DIR>          .
29/03/04  06:30       <DIR>          ..
28/10/03  07:31                   314 adupdate.reg
28/10/03  07:31                   240 IMPORTANT-READ-BEFORE-OPENING-REGISTRYFILE.txt
               4 File(s)            554 bytes
```

```
Modified              Size(Bytes):     Name

Directory of <CD-ROM>:\schema\Aphelion

29/03/04  06:30       <DIR>          .
29/03/04  06:30       <DIR>          ..
28/10/03  07:31              6,307 selectaccess.at.conf
28/10/03  07:31              4,188 selectaccess.oc.conf
               4 File(s)        10,495 bytes

Directory of <CD-ROM>:\schema\CA-eTrust

29/03/04  06:29       <DIR>          .
29/03/04  06:30       <DIR>          ..
28/10/03  07:31             16,601 SAeTrustSchema.dxc
               3 File(s)        16,601 bytes

Directory of <CD-ROM>:\schema\Critical-Path

29/03/04  06:30       <DIR>          .
29/03/04  06:30       <DIR>          ..
29/03/04  06:30       <DIR>          CP4.0
29/03/04  06:30       <DIR>          CP4.1
               4 File(s)             0 bytes

Directory of <CD-ROM>:\schema\Critical-Path\CP4.0

29/03/04  06:30       <DIR>          .
29/03/04  06:30       <DIR>          ..
28/10/03  07:31              4,487 oidslocal-SA
28/10/03  07:31             20,455 schema-SA
               4 File(s)        24,942 bytes

Directory of <CD-ROM>:\schema\Critical-Path\CP4.1

29/03/04  06:30       <DIR>          .
29/03/04  06:30       <DIR>          ..
28/10/03  07:31             20,385 ldapattribs.ldif
28/10/03  07:31              1,080 ldapnb.ldif
28/10/03  07:31              3,094 ldapocadd.ldif
28/10/03  07:31                919 ldapocupdate.ldif
               6 File(s)        25,478 bytes
```

Modified Size(Bytes): Name

Directory of <CD-ROM>:\schema\Siemens-DirX

```
29/03/04  06:29      <DIR>          .
29/03/04  06:30      <DIR>          ..
28/10/03  07:31              1,353 accessControl_SelectAccess.cp
28/10/03  07:31              1,123 add_goun_nf.adm
28/10/03  07:31              1,645 add_vendor_attributes.adm
28/10/03  07:31                737 bind.tcl
28/10/03  07:31              1,162 create_SelectAccess_admin.cp
28/10/03  07:31              8,661 dirxabbr-ext.SelectAccess
28/10/03  07:31                179 dirxabbr-VN-VV.SelectAccess
28/10/03  07:31              1,170 extend_LDAP_Root.cp
28/10/03  07:31              5,779 GlobalVar.tcl
28/10/03  07:31              1,287 goun_rule.cp
28/10/03  07:31                729 l-bind.cp
28/10/03  07:31             11,353 schema_ext_for_SelectAccess.adm
28/10/03  07:31              1,343 Seimens_user_loc_no_schema_instructions.txt
28/10/03  07:31              5,588 setup.bat
28/10/03  07:31              3,313 subschema_ext_for_SelectAccess.cp
              17 File(s)         45,422 bytes
```

Directory of <CD-ROM>:\solutions

```
29/03/04  06:31      <DIR>          .
29/03/04  06:31      <DIR>          ..
29/03/04  06:31      <DIR>          apache2
29/03/04  06:31      <DIR>          citrix
29/03/04  06:31      <DIR>          Domino
29/03/04  06:31      <DIR>          iwa
29/03/04  06:31      <DIR>          oracle
29/03/04  06:31      <DIR>          plumtree
               8 File(s)              0 bytes
```

```
Modified            Size(Bytes):     Name

Directory of <CD-ROM>:\solutions\Domino
29/03/04  06:31        <DIR>          .
29/03/04  06:31        <DIR>          ..
28/10/03  07:31             1,283 domino_site_data.c
28/10/03  07:31            45,056 domino_web.dll
               4 File(s)         46,339 bytes

Directory of <CD-ROM>:\solutions\apache2
29/03/04  06:31        <DIR>          .
29/03/04  06:31        <DIR>          ..
28/10/03  07:31         1,768,206 mod_enforcer2.so
               3 File(s)      1,768,206 bytes

Directory of <CD-ROM>:\solutions\citrix
29/03/04  06:31        <DIR>          .
29/03/04  06:31        <DIR>          ..
29/03/04  06:31        <DIR>          ASP
29/03/04  06:31        <DIR>          RuleBuilderPlugin
               4 File(s)              0 bytes

Directory of <CD-ROM>:\solutions\citrix\ASP
29/03/04  06:31        <DIR>          .
29/03/04  06:31        <DIR>          ..
29/03/04  06:31        <DIR>          NFuse1.6
29/03/04  06:31        <DIR>          NFuse1.7
               4 File(s)              0 bytes

Directory of <CD-ROM>:\solutions\citrix\ASP\NFuse1.6
29/03/04  06:31        <DIR>          .
29/03/04  06:31        <DIR>          ..
28/10/03  07:31            21,330 applist.asp
28/10/03  07:31             7,992 default.htm
28/10/03  07:31             2,453 frameset.asp
28/10/03  07:31             5,450 launch.asp
28/10/03  07:31             4,404 login.asp
```

```
Modified            Size(Bytes):     Name
    28/10/03  07:31                 202 logout.asp

                8 File(s)        41,831 bytes

    Directory of <CD-ROM>:\solutions\citrix\ASP\NFuse1.7

    29/03/04  06:31        <DIR>          .
    29/03/04  06:31        <DIR>          ..
    29/03/04  06:31        <DIR>          include
    28/10/03  07:31                 193 applist.asp
    28/10/03  07:31                 191 launch.asp
    28/10/03  07:31               4,741 login.asp
                6 File(s)         5,125 bytes

    Directory of <CD-ROM>:\solutions\citrix\ASP\NFuse1.7\include

    29/03/04  06:31        <DIR>          .
    29/03/04  06:31        <DIR>          ..
    28/10/03  07:31              26,683 applistImpl.vbs
    28/10/03  07:31               5,759 launchImpl.vbs
                4 File(s)        32,442 bytes

    Directory of <CD-ROM>:\solutions\citrix\RuleBuilderPlugin

    29/03/04  06:31        <DIR>          .
    29/03/04  06:31        <DIR>          ..
    28/10/03  07:31               6,037 component.jar
    28/10/03  07:31                 541 component.xml
    28/10/03  07:31               1,141 icon.gif
    28/10/03  07:31                 866 toolbaricon.gif
                6 File(s)         8,585 bytes

    Directory of <CD-ROM>:\solutions\iwa

    29/03/04  06:31        <DIR>          .
    29/03/04  06:31        <DIR>          ..
    28/10/03  07:31                 312 EnableIWA.reg
                3 File(s)           312 bytes
```

```
Modified              Size(Bytes):      Name

Directory of <CD-ROM>:\solutions\oracle

29/03/04  06:31         <DIR>          .
29/03/04  06:31         <DIR>          ..
28/10/03  07:31             40,960 oracle_apache_web32.dll
               3 File(s)         40,960 bytes

Directory of <CD-ROM>:\solutions\plumtree

29/03/04  06:31         <DIR>          .
29/03/04  06:31         <DIR>          ..
28/10/03  07:31              6,285 customsso.asp
               3 File(s)          6,285 bytes

Total Files Listed:          164 File(s)    321,536,017 bytes
                                                      0 bytes free
```

Appendix B: Engineering Patch G

Test case has been provided and included with these updated documents. Document is HP OpenView Select Access Engineering Patch G Test Case.doc.

Table 24: Configuration List for HP OpenView Select Access v5.2 Engineering Patch G software

Modified	Size(Bytes):	Name
Directory of Engineering Patch G:\		
23/03/04 15:47	372,736	enforcer32.dll
23/03/04 15:48	139,264	IISPlugin32.dll
31/03/04 07:30	11,403	SA52_Eng_Patch_G.htm
3 File(s)	403,403 bytes	
Total Files Listed:	**3 File(s)**	**403,403 bytes**