RSA, The Security Division of EMC
RSA® Access Manager v6.1

Security Target

Evaluation Assurance Level: EAL3+
Augmented with ALC_FLR.2
Document Version: 0.8

Prepared for:

RSA, The Security Division of EMC
174 Middlesex Turnpike
Bedford, MA 01730
Phone: (877) RSA-4900

http://www.rsa.com

Prepared by:

Corsec Security, Inc.
10340 Democracy Lane, Suite 201
Fairfax, VA 22030
Phone: (703) 267-6050

http://www.corsec.com

Table of Contents

Table of Figures

Table of Tables

1 Security Target Introduction

This section identifies the Security Target (ST), Target of Evaluation (TOE), and the ST organization. The Target of Evaluation is the RSA® Access Manager, v6.1 and will hereafter be referred to as the TOE throughout this document. The TOE (Access Manager) is a software product designed to fulfill identity management needs. The TOE provides a central management interface which allows efficient administration of the security policy being enforced, and the users upon which the policy is enforced.

1.1 Purpose

This ST provides mapping of the Security Environment to the Security Requirements that the TOE meets in order to remove, diminish or mitigate the defined threats in the following sections:

- Security Target Introduction (Section 1) – Provides a brief summary of the ST contents and describes the organization of other sections within this document. It also provides an overview of the TOE security functions and describes the physical and logical scope for the TOE.
- Conformance Claims (Section 2) – Provides the identification of any Common Criteria (CC), ST Protection Profile, and Evaluation Assurance Level (EAL) package claims.
- Security Problem Definition (Section 3) – Describes the threats, policies, and assumptions that pertain to the TOE and its environment.
- Security Objectives (Section 4) – Identifies the security objectives that are satisfied by the TOE and its environment.
- Extended Components Definition (Section 5) – Identifies new components (extended Security Functional Requirements (SFRs) and extended Security Assurance Requirements (SARs)) that are not included in CC Part 2 or CC Part 3.
- Security Requirements (Section 6) – Presents the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) met by the TOE and by the TOE's environment.
- TOE Summary Specification (Section 7) – Describes the security functions provided by the TOE that satisfy the security functional requirements and objectives.
- Rationale (Section 8) – Presents the rationale for the security objectives, requirements, and SFR dependencies as to their consistency, completeness, and suitability.
- Terminology and Acronyms (Section 9) – Defines the acronyms and terminology used within this ST.

1.2 Security Target and TOE References

Table 1 – ST and TOE References

ST Title	RSA, The Security Division of EMC RSA® Access Manager v6.1 Security Target
ST Version	Version 0.8
ST Author	Corsec Security, Inc.
ST Publication Date	November 5, 2009
TOE Reference	RSA® Access Manager v6.1 build 20090806085904-0400-496214

Note: The U.S. Government Protection Profile Authorization Server for Basic Robustness Environments, Version 1.1 dated 25 July 2007 contains IT Environment Security Functional Requirements requiring that the IT Environment be *compliant with the Controlled Access Protection Profile or an Operating System Protection Profile at the Basic Level of Robustness or Greater. As v3.1 of the Common Criteria Standard does not require Environmental SFRs, these SFRs have been removed, but the associated Environmental Objectives remain.*

1.3 TOE Overview

The TOE Overview summarizes the usage and major security features of the TOE. The TOE Overview provides a context for the TOE evaluation by identifying the TOE type, describing the product, and defining the specific evaluated configuration.

RSA Access Manager (the TOE) is a software product designed to fulfill identity management needs. The TOE has a central management interface which allows efficient administration of the security policy being enforced, and the users the policy is being enforced upon. The TOE manages user accounts and determines whether or not users have the correct permissions to view protected resources.

In its typical deployment scenario, an Access Manager Agent is installed on a server that hosts a protected resource. Each time a user attempts to access the protected resource, the server will redirect the user to the TOE server to authenticate himself. If the user is successfully authenticated, and if the user has the permissions required allowing the user to view the resource, then the user will be granted access to the protected resource.

The TOE has the following features:

- Resource Access Management – The TOE controls access to protected resources over the web. If a user attempts to view a protected resource, the user must successfully authenticate himself before Access Manager will grant or deny him access to the resource based on his privilege profile.
- Interoperability – The TOE is capable of being successfully deployed throughout multi-vendor environments. The TOE additionally provides native support for user LDAP (Lightweight Directory Access Protocol) data stores.
- Single Sign-On – With Single Sign-On, users are able to seamlessly access protected resources deployed on various servers throughout the network without having to re-authenticate their identities.
- Identity Management and User Privilege Management – Identities can be centrally managed, and trust in each identity is established through user authentication. Additionally, privileges are centrally managed and can be mapped to user identities statically (for example: each department can have its own set of privileges) or dynamically (*e.g.* each account has privileges specific only to that account).
- SNMP (Simple Network Management Protocol) Support – Access Manager provides support for a third-party Network Management System (NMS) using SNMP. An NMS reveals how the Access Manager Servers are functioning in a production environment, making it easier to configure them for optimal performance.
- Auditing and Reporting – Logs are recorded, and reports can be generated based on logs that capture the actions of users, administrators, and system processes.
- Protection – The TOE uses the RSA BSAFE module in the TOE's operational environment to obfuscate all traffic over the network (via TLS (Transport Layer Security). Additionally, the TOE can optionally obfuscate component configuration files on-disk.
- Authentication Support - Authentication methods are configured on each non-TOE Agent. Access Manager can work with Oracle and iPlanet data stores, and supports the following authentication methods[1]:
    - **Basic**. This is the default authentication method. At logon, users enter their user names and passwords, which are authenticated with the user account information in the Access Manager data store.
    - **RSA SecurID**. Access Manager supports RSA SecurID two-factor authentication. At logon, a user name and password are authenticated against the credentials stored in the Authentication Manager.

[1] Note that only the Basic authentication method is included in the evaluated configuration of the TOE.

- **X.509 Certificates**. Access Manager supports X.509 certificates. The web server must be configured to accept browser certificates for authentication.
- **Windows**. Access Manager can use several methods to authenticate users against the Windows environment. These include NT, NT LAN (Local Area Network) Manager (NTLM), and Integrated Windows Authentication (IWA).
- **Custom**. Developers can use an Access Manager Web Agent Extension (WAX) or the Access Manager Runtime API (Application Programming Interface) to create their own custom authentication methods, with custom error messages and logging. They can also create WAX programs that integrate with existing legacy authentication methods.

Each authentication method prompts the user to provide the appropriate identification credentials. For added security, administrators can combine different authentication methods. For example, one could require that a user first authenticate by Basic authentication and then by Windows NT authentication.

Figure 1 shows the typical deployment configuration of the TOE:

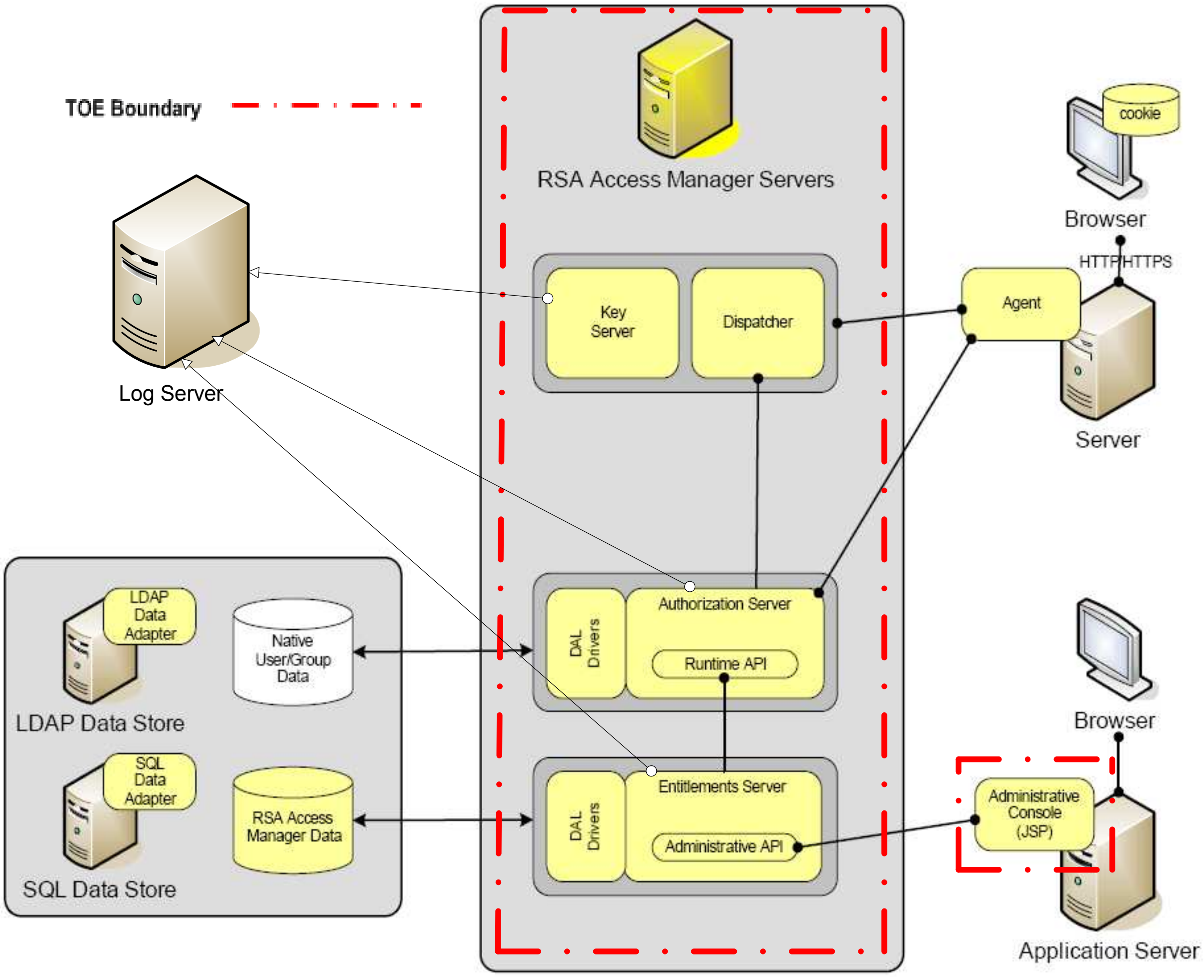

Figure 1 – Typical Deployment Configuration of the TOE

1.3.1 TOE Components

The TOE comprises multiple components which work together to perform identity and access management services. The following components compose the TOE.

1.3.1.1 Entitlements Server

The Entitlements Server provides Administrative API clients (including the Administrative Console) with read/write access to the Access Manager data store. This allows the Administrator to establish security policies, or modify existing policies. The Entitlements Server can be configured to selectively update the cached data on the Authorization Servers. By doing so, changes made to the Access Manager data store, such as entitling a user to access a resource, take effect immediately. The Entitlements Server provides auditing to track administrator actions on the server.

1.3.1.2 Authorization Server

The Authorization Server performs the authentication and authorization checks for users at runtime. As a user attempts to access a protected resource, the Authorization Server determines whether the authentication method validated the user and if the user is allowed to access the resource. The Authorization Server reads the user and policy information directly from the data store. The Authorization Server can be configured to cache a variety of data. When properly configured, the Authorization Server does not have to access the data store to check access privileges for users who have already been allowed or denied access to a resource. The Authorization Server provides auditing to track administrator actions on the server.

1.3.1.3 Dispatcher/Key Server

The Dispatcher/Key Server has two functions. The Dispatcher keeps track of all available Authorization Servers. By default, Agents are configured to query the Dispatcher at startup for available Authorization Servers. Agents then connect to the Authorization Servers that are available. The Key Server generates single sign-on (SSO) token encryption keys (or secret keys) via a cryptographic module in the TOE’s operational environment, which carry a limited lifetime. When a user authenticates to the Access Manager system, the Authorization Server issues a token, encrypted with one of these keys via the environmental cryptographic module, which encapsulates the user's session state. The Agent returns this token to the user's browser in the form of a cookie. On subsequent requests, the token is sent back to the Authorization Server for decryption by the environmental cryptographic module as needed. The Dispatcher/Key Server provides auditing to track administrator actions on the server.

1.3.1.4 RSA Access Manager Data Adapters

Access Manager uses the Data Abstraction Layer (DAL) to access user data in data stores, such as an LDAP directory or an SQL database. The user data store contains all information about TOE users and their access privileges. Access Manager adds additional policy, resource, and administration data schemas to the directory server or database, which can be managed separately from the user data store. This allows the administrator to consolidate users and security policies into one central location, making administration of the enterprise security less time-consuming.

Access Manager policy, resource, and administrative data can also be kept in separate data stores from user and group data. The Administrator can configure the Data Adapter to control the location and setup of the user and policy data stores. Keep in mind that data stores must all be of the same type. For example, Administrators cannot store some users in an LDAP directory, and other users in an SQL database.

Changes to the Access Manager data stores are made through the Administrative Console (or through the Administrative API, which has larger capabilities than those provided by the Administrative Console). Both the Entitlements Server and Authorization Server use the Data Abstraction Layer (DAL) drivers to connect directly to data servers. A single Data Adapter is needed for each of the data stores.

1.3.1.5 RSA Access Manager Administrative Console

The access Manager Administrative Console is used to administer the system. The Administrative Console is a web-based, Java Server Page (JSP) application installed on any supported application server or servlet engine. For a list of supported application servers and servlet engines, see the Servers Installation and Configuration Guide.

The Administrative Console can be accessed from any computer with a web browser. From the Administrative Console, administrative groups and roles can be set up, resources added, and security policies defined. The Administrative Console can be used to add and edit users and groups, and store the information in its data store.

1.3.2 Excluded Components

The following product components can be implemented, depending on an Enterprise's security needs, system load, existing network architecture, and logging plans. However, they are not part of the TOE.

1.3.2.1 Web Server and Application Server Agents

An Agent must be installed on each of the servers Administrators want to protect. There are two types of Agents: Web Server Agents and Applications Server Agents. These are described below:

1.3.2.1.1 Web Server

RSA Access Manager Web Server Agents supplement the native security mechanisms of a web server. They run in the same process as the web server itself and are invoked whenever the web server needs to determine access rights for a particular Uniform Resource Locator (URL). The Agents forward access requests to an Authorization Server, which passes the answers it receives back to the web server.

1.3.2.1.2 Application Server

RSA Access Manager Application Server Agents supplement the native security on application servers with Access Manager, and extend single sign-on to the web application environment. This allows Administrators to protect web resources, such as servlets, Enterprise Java Beans (EJBs), and Java Server Pages (JSPs) with Access Manager.

1.3.2.2 Redundant RSA Access Manager Servers and LDAP Directories

Additional Access Manager Servers and LDAP directories can be deployed to increase runtime performance, stability, and to eliminate single points of failure in the Access Manager system.

1.3.2.3 RSA Access Manager Log Server

The Access Manager Log Server allows the administrator to configure the system so that all servers write to a single log file, regardless of where the servers are physically located.

1.3.2.4 RSA Access Manager Instrumentation Server

The Instrumentation Server provides Simple Network Management Protocol (SNMP) support for a third-party Network Management System (NMS). Using an NMS, a system administrator can query the Instrumentation Server for information about the Access Manager Servers that are running in a production environment. This allows for real-time monitoring of Server activity and performance.

1.3.2.5 RSA Access Manager Secure Proxy Server

The Access Manager Secure Proxy Server (SPS) is a self-contained reverse proxy and access control solution that consists of these components:

- Secure Apache-based HTTP server

- Fully integrated RSA Access Manager Agent
- Proxy engine

The SPS can be used as a proxy-based gateway to secure web servers. This allows the protection of web servers not currently supported by an Access Manager Web Server Agent. The Access Manager SPS can be placed in the Demilitarized Zone (DMZ) of the network to ensure that non-authenticated users cannot access the web resources, even though the web servers where these resources reside are not protected by an RSA Access Manager Agent. The SPS provides rapid, out-of-the-box access control in a robust and scalable fashion. The proxy engine can be configured to dynamically route incoming requests to the appropriate web server.

1.3.2.6 RSA Security Certificate Manager

The Security Certificate Manager creates keystores that can be used by Access Manager for inter-component security. RSA Access Manager Software Development Kit Access Manager is a highly customizable solution. The following Access Manager Application Programming Interfaces (APIs) and Service Provider Interfaces (SPIs) can be used to create custom applications that work with the Access Manager components.

1.3.2.7 Administrative APIs (Java and DCOM)

The Java and DCOM versions of the Administrative API allow Administrators to develop applications that interact with the Entitlements Server to create user accounts and the security policies that protect resources. A security policy identifies protected resources, defines the entitlements and Smart Rules that control access to these resources, and identifies the administrative groups and administrative roles in these groups that manage the security policy itself. In addition, if Access Manager is configured for write access to the user data store, then the Administrative API applications can create and update users and user groups. This allows Administrators to write custom programs to perform various administrative functions. For example:

- Load a large quantity of data from another source directly into the Access Manager data store.
- Develop custom web applications to perform self-registration and self-service account management for Access Manager.
- Develop custom policy administration applications that enhance the functionality provided by the Administrative Console.

Note: The C Administrative API was deprecated in RSA ClearTrust 5.5[2] and is not included with Access Manager 6.0. It is still available to developers who download an older version of the Software Development Kit (SDK).

1.3.2.8 Runtime APIs (Java, C, and DCOM)

The Java, C, and DCOM versions of the Runtime API allow Administrators to develop custom programs that use or extend the runtime functionality of the Authorization Server. The Runtime API provides efficient and scalable read-only access to certain Access Manager objects and security policy settings. Administrators can use the Runtime API to:

- Authenticate users.
- Control user access to protected resources.
- Personalize a user's online experience.

Allow SSO tokens created by the Runtime API to be passed to application servers and web servers.

[2] RSA Access Manager was formerly known as RSA ClearTrust.

1.3.2.9 Web Agent Extension (WAX) API

The WAX API, implemented in C, extends the functionality of RSA's Access Manager Web Server Agents. This allows Administrators to customize or control the behavior of the Agent during the authentication and authorization processing. For example, Administrators can:

- Create an extension to do custom logging.
- Create an extension to do custom authentication of users without connecting to an Access Manager Authorization Server.

Create an extension to direct the web server to custom HTML pages based on the return codes returned from the Authorization Server.

1.3.2.10 Service Provider Interfaces

The Service Provider Interfaces (SPIs) allow Administrators to extend the Access Manager Servers in various ways by implementing code that is run in-process as part of the Servers. This code is registered with the Servers to be invoked at certain points during client request processing. This allows Administrators to:

- Alter or override default Administrative and Runtime API call behavior, or to perform arbitrary operations (for example, sending notifications to remote systems) when such calls are executed.
- Retrieve user properties from third-party data sources for use in Smart Rule evaluation and by RSA Access Manager Agents.

Make additional Runtime API calls within the context of a client call execution within the Authorization Server. This makes it possible to have more complex combinations of authentication and authorization logic.

1.3.3 TOE Environment

The necessary hardware and software for the TOE to operate is described in Table 2 below.

Enforcement of RSA® Access Manager's access control decisions on principals are enforced by servers and programs in the TOE Environment.

The host computer will need to have a network connection. Additionally, the operating system must be in a secure location, operate in a secure state, and run a FIPS 140-2 validated version of RSA BSAFE.

1.4 TOE Description

This section will primarily address the physical and logical components of the TOE included in the evaluation.

1.4.1 Physical Scope

Figure 1 above illustrates the physical scope and the physical boundary of the overall solution and ties together all of the components of the TOE and the constituents of the TOE Environment.

The TOE is a software-only TOE designed to fulfill identity management needs. The server software components which make up the TOE are typically installed on a single host computer that is compliant with the minimum requirements as listed in Table 2. The non-TOE servers hosting resources requiring protection will typically also host a non-TOE Access Manager Agent which will refer the principal (any user or application) to the Access Manager Server when the principal attempts to access a protected resource, as depicted in Figure 1 above. There are no hardware components that come with the TOE.

1.4.1.1 TOE Software

The TOE is a software product designed to fulfill identity management needs. The software that makes up the TOE is typically installed on a single host computer.

1.4.1.2 Guidance Documentation

The following guides are required reading and part of the TOE:

- RSA Access Manager 6.1 Getting Started
- RSA Access Manager 6.1 Common Criteria Installation and Configuration Guide
- RSA Access Manager 6.1 Servers Installation and Configuration Guide
- RSA Access Manager 6.1 Administrator's Guide
- RSA Access Manager 6.1 Planning Guide
- RSA Access Manager 6.1 Upgrade Guide (if upgrading from a previous version of Access Manager (which was formerly called ClearTrust)).

1.4.2 Logical Scope

The security functional requirements implemented by the TOE are usefully grouped under the following Security Function Classes:

- Security Audit
- User Data Protection
- Identification and Authentication
- Security Management
- Protection of the TOE Security Functions
- TOE Access

1.4.2.1 Security Audit

The Security Audit function provides the TOE with the functionality for generation of audit records. As administrators manage and configure the TOE, their activities are automatically logged. All security-relevant configuration settings and changes are recorded to ensure accountability of the administrator's actions.

1.4.2.2 User Data Protection

The User Data Protection function implements functionality for TOE security functions and TOE security function policies related to protecting user data. The user data that the TOE is protecting is any resource(s) that the TOE is assigned by the administrator to protect.

The TOE uses its Authorization Server Access Control Policy to provide an access decision and enforce the decision on principals, protected resources, and all operations between the two. An access decision is provided and enforced by the TOE through the comparison of user attributes with the Authorization Server Access Control Policy, which is composed of Entitlements and/or Smart Rules implemented by the administrator.

1.4.2.3 Identification and Authentication

The Identification and Authentication function identifies and authenticates users to the TOE. End users must identify and authenticate themselves to the TOE anytime they wish to access a resource protected by the TOE. Access Manager provides its own internal authentication mechanism for identifying and authenticating users to the TOE. It does this by validating the user's username and password against the Access Manager Data store. The TOE can integrate with several external authentication types such as Microsoft NT Primary Domain Controller (PDC). Additionally, each administrator must identify and authenticate himself before he can administer the TOE.

1.4.2.4 Security Management

The Security Management features provide management and administration functions of the TOE for its administrators. The TSF is capable of associating users with roles. The TOE uses customizable user roles. The Security Administrator is the only user who can configure or modify the Authorization Server Access Control Policy settings. Additionally, only the Security Administrator is authorized to modify security attributes, for principals or protected resources, used by the Authorization Server Access Control Policy to make access decisions.

1.4.2.5 Protection of the TOE Security Functions

The Security Administrator is the sole role capable of modifying and verifying the integrity of the TOE system configuration files. Additionally, only the Security Administrator is capable of verifying the integrity of the stored TSF executable code.

1.4.2.6 TOE Access

When an administrator initiates an administrative user session, the TOE displays an advisory message warning about unauthorized use of the TOE.

1.4.2.7 Security Considerations in the TOE Environment:

The TOE's underlying operating system must be maintained in a secure state and physical access to the computers hosting the Access Manager components must be kept secure. Table 2 specifies the minimum system requirements for the proper operation of the TOE.

Table 2 – TOE Minimum Requirements

Category	Windows	Solaris
Operating System	Operating System: Microsoft Windows Server 2003 SP2 including R2 (64 bit) x86 Architecture: x86/x86-64/EM64T, 500 MHz (Megahertz) or faster Disk Space: 200 MB	Operating System: Solaris 10 – on SPARC – 64 bit Architecture: UltraSPARC, 500 MHz or faster Disk Space: 200 MB
Administrative Console	BEA WebLogic Server 10	BEA WebLogic Server 10
Data Store	Oracle 10g Release 2 (10.2)	iPlanet 6.3
RSA Access Manager Agents	IIS Web Agent 4.8 on Win2k3 R2 SP2 64 bit	Apache2.x.x Agent 4.8 on Solaris 10 Sparc
Browser	IE 6	IE 6

2 Conformance Claims

This section provides the identification for any CC, Protection Profile (PP), and EAL package conformance claims. Rationale is provided for any extensions or augmentations to the conformance claims. Rationale for CC and PP conformance claims can be found in Section 8.1.

Table 3 – CC and PP Conformance

Common Criteria (CC) Identification and Conformance	Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 2, September 2007; CC Part 2 extended; CC Part 3 conformant; PP claim.
PP Identification	US Government Protection Profile Authorization Server for Basic Robustness Environments, version 1.1
Evaluation Assurance Level	EAL3+ Augmented with Flaw Remediation (ALC_FLR.2)

3 Security Problem Definition

This section describes the security aspects of the environment in which the TOE will be used and the manner in which the TOE is expected to be employed. It provides the statement of the TOE security environment, which identifies and explains all:

- Known and presumed threats countered by either the TOE or by the security environment
- Organizational security policies with which the TOE must comply
- Assumptions about the secure usage of the TOE, including physical, personnel and connectivity aspects

3.1 Threats to Security

This section identifies the threats to the IT assets against which protection is required by the TOE or by the security environment. The threat agents are divided into two categories:

- Attackers who are not TOE users: They have public knowledge of how the TOE operates and are assumed to possess a low skill level, limited resources to alter TOE configuration settings/parameters and no physical access to the TOE.
- TOE users: They have extensive knowledge of how the TOE operates and are assumed to possess a high skill level, moderate resources to alter TOE configuration settings/parameters and physical access to the TOE. (TOE users are, however, assumed not to be willfully hostile to the TOE.)

Both are assumed to have a low level of motivation. The IT assets requiring protection are the user data saved on or transitioning through the TOE and the hosts on the protected network. Removal, diminution, and mitigation of the threats are through the objectives identified in Section 4 - Security Objectives.

The following threats are applicable:

Table 4 **– Threats**

Name	Description
T.ACCIDENTAL_ADMIN_ERROR	An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
T.ACCIDENTAL_AUDIT_COMPROMISE	An administrative user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.
T.ACCIDENTAL_CRYPTO_COMPROMISE	An administrative user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the cryptographic mechanisms and the data protected by those mechanisms.
T.LOW_PRIORITY	A low priority process may exhaust resources required by the TOE.
T.MASQUERADE	A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.

Name	Description
T.POOR_DESIGN	Unintentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a casually mischievous user or program.
T.POOR_IMPLEMENTATION	Unintentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a casually mischievous user or program.
T.POOR_TEST	Developers or test engineers may implement tests that are insufficient to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being discovered thereby causing potential security vulnerabilities.
T.RESIDUAL_DATA	A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.
T.TSF_COMPROMISE	An attacking user or process may cause, through an unsophisticated attack, TSF data, or executable code to be inappropriately accessed (viewed, modified, or deleted).
T.UNATTENDED_SESSION	A user may gain unauthorized access to an unattended session.
T.UNAUTHORIZED_ACCESS	A user or application may gain access to the data for which they are not authorized according to the TOE security policy.
T.UNIDENTIFIED_ACTIONS	The administrator may not have the ability to notice potential security violations, this limiting the administrator's ability to identify and take action against a possible security breach,

3.2 Organizational Security Policies

An Organizational Security Policy (OSP) is a set of security rules, procedures, or guidelines imposed by an organization on the operational environment of the TOE. The following OSPs are presumed to be imposed upon the TOE or its operational environment by any organization implementing the TOE in the CC evaluated configuration:

Table 5 **– Organizational Security Policies**

Name	Description
P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by access the system.
P.ACCOUNTABILITY	The TOE shall log all actions by authorized users such that the authorized

Name	Description
	users can be held accountable for their actions within the TOE.
P.BASIC_ROBUSTNESS	The TOE must be developed in accordance with the Basic Robustness guidelines.
P.CAPP_OS	The operating system the TOE operates on top of must be evaluated to be compliant with the Controlled Access Protection Profile.
P.COMMS	Communications exist between the TOE components (internally) and between the TOE components and the IT components.
P.CRYPTOGRAPHY	Only NIST FIPS 140-2 validated cryptography (methods and implementations) are acceptable for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e., encryption, decryption, signature, hashing, key exchange, and random number generation services).
P.HIGH_AVAILABILITY	The TOE shall include providing resource allocations to support priority of service and fault tolerance.
P.NO_GENERAL_PURPOSE	There will be no general-purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the hardware platforms that the TOE administrative and authorization policy engine software are installed. If Authorization Server "Agent" software is part of the TOE, then the system on which the Agent operates is exempt from the assumption.
P.TOE_ENVIRONMENT_ACCESS	The TOE environment will provide mechanisms that control a user's logical access to the TOE environment components.
P.WEB_BROWSER_PP	If administrators use a web browser to access the TOE for remote administration, they must use software that has been evaluated to the Web Browser Protection Profile.

Note to Consumer: The Authorization Server Protection Profile states "If the TOE supports remote administration via web browser, then the guidance documents shall instruct administrators to use a web browser that has been evaluated to be compliant with the Web Server Protection Profile (if any such web browsers exist at the time of the TOE evaluation)." No web browsers have been evaluated against the Web Browser Protection Profile because the Web Browser Protection Profile is still a draft. Therefore, any of the web browsers defined in Table 2 above are acceptable for remote administration of the TOE.

3.3 Assumptions

This section describes the security aspects of the intended environment for the evaluated TOE. The operational environment must be managed in accordance with assurance requirement documentation for delivery, operation, and user guidance. The following specific conditions are required to ensure the security of the TOE and are assumed to exist in an environment where this TOE is employed.

Table 6 – Assumptions

Name	Description
A.IT_ACCESS	The TOE has access to all the IT System data it needs to perform its functions.
A.LOWEXP	The threat of malicious attacks aimed at discovering exploitable vulnerabilities is considered low.
A.MANAGE	There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.
A.NO_EVIL	Administrators are non-hostile, appropriately trained and follow all administrator guidance.
A.NO_TOE_BYPASS	Principals cannot gain access to resources protected by the TOE without passing through the TOE access control mechanisms.
A.PHYSICAL	The IT environment provides the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE.
A.SCALABLE	The TOE environment is appropriately scalable to provide support to the IT Systems in the organization it is deployed.

4 Security Objectives

Security objectives are concise, abstract statements of the intended solution to the problem defined by the security problem definition (see Section 3). The set of security objectives for a TOE form a high-level solution to the security problem. This high-level solution is divided into two part-wise solutions: the security objectives for the TOE, and the security objectives for the TOE's operational environment. This section identifies the security objectives for the TOE and its supporting environment, as well as providing a mapping of the objectives to the threats, OSPs, and assumptions included in the security problem definition. This mapping also provides rationale for how the threats, OSPs, and assumptions are effectively and fully addressed by the security objectives.

4.1 Security Objectives for the TOE

The specific security objectives for the TOE are as follows:

Table 7 **– Security Objectives for the TOE**

Name	Description
O.ADMIN_GUIDANCE	The TOE will provide administrators with the necessary information for secure management.
O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security-relevant events associated with users.
O.CORRECT_TSF_OPERATION	The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site.
O.DISPLAY_BANNER	The TOE will display an advisory warning regarding use of the TOE to the administrative users.
O.MANAGE	The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
O.MEDIATE	The TOE must protect user data in accordance with its security policy.
O.PARTIAL_SELF_PROTECTION	The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces.
O.RESIDUAL_INFORMATION	The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.
O.TOE_ACCESS	The TOE will provide mechanisms that control a user's logical access to the TOE.

4.2 Security Objectives for the Operational Environment

4.2.1 IT Security Objectives

The following IT security objectives are to be satisfied by the environment:

Table 8 **– IT Security Objectives**

Name	Description
OE.AUDIT_PROTECTION	The IT Environment will provide the capability to protect audit information.
OE.CAPP_OS	Operating systems the TOE operates on top of must be compliant with the Controlled Access Protection Profile. The operating system will therefore provide all the capabilities outlined in the CAPP security function requirements and will have been evaluated against the CAPP assurance requirements.
OE.COMMS	Sites deploying the TOE will ensure that adequate communications exist between the TOE components (internally) and between the TOE components and the IT components.
OE.CRYPTOGRAPHY	The IT environment components shall use NIST FIPS 140-2 validated cryptographic modules if they provide cryptographic services.
OE.DISPLAY_BANNER	The underlying operating system of the TOE will display an advisory warning regarding use of the TOE to administrative users logging on the platform where the TOE software is installed.
OE.IT_ACCESS	Sites deploying the TOE will ensure the TOE has access to all the IT System data it needs to perform its functions.
OE.FAULT_TOLERANCE	The IT environment will provided limited capabilities to support degraded fault tolerance and fail over for some TOE components.
OE.LOWEXP	Site deploying the TOE will establish a protective environment where the threat of malicious attacks aimed at discovering exploitable vulnerabilities is considered low.
OE.NO_GENERAL_PURPOSE	There will be no general-purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the hardware platforms that the TOE administrative and authorization policy engine software are installed. This objective does not apply to agent software that might reside on a web server.
OE.PRIORITY	The IT Environment will provide prioritization of resources to support the TOE.
OE.RESIDUAL_INFORMATION	The IT Environment will ensure that any information contained in a protected resource is not released when the resource is reallocated.

Name	Description
OE.SCALABLE	Sites using the TOE will deploy the appropriate hardware and software environment to ensure the TOE system is scalable to provide support to the IT Systems in the organization it is deployed.
OE.WEB_BROWSER_PP	If administrators use a web browser to access the TOE for remote administration, they must to use software that has been evaluated to the Web Browser Protection Profile.
OE.NO_TOE_BYPASS	Principals cannot gain access to resources protected by the TOE without passing through the TOE access control mechanisms.

4.2.2 Non-IT Security Objectives

The following non-IT environment security objectives are to be satisfied without imposing technical requirements on the TOE. That is, they will not require the implementation of functions in the TOE hardware and/or software. Thus, they will be satisfied largely through application of procedural or administrative measures.

Table 9 – Non-IT Security Objectives

Name	Description
OD.BASIC_ROBUSTNESS	The TOE shall be developed in accordance with the Basic Robustness requirements.
OD.CONFIGURATION_IDENTIFICATION	The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly.
OD.DOCUMENTED_DESIGN	The design of the TOE is adequately and accurately documented.
OD.PARTIAL_FUNCTIONAL_TESTING	The TOE will undergo some security functional testing that demonstrates the TSF satisfies its security functional requirements.
OD.VULNERABILITY_ANALYSIS	The TOE will undergo vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws.
OE.NO_EVIL	Sites using the TOE shall ensure that administrators are non-hostile, appropriately trained and follow all administrator guidance.
OE.PHYSICAL	Physical security will be provided within the domain for the value of the IT assets protected by the operating system and the value of the stored, processed, and transmitted information.

Name	Description
OE.MANAGE	The TOE environmental components will provide all the functions, facilities and competent individuals necessary to support the administrators in their management of the security of the environment, and restrict these functions and facilities from unauthorized use.
OE.TOE_ENVIRONMENT_ACCESS	The TOE environment will provide mechanisms that control a user's logical access to the environmental components.

5 Extended Components Definition

This section defines the extended Security Functional Requirements (SFRs) and extended Security Assurance Requirements (SARs) met by the TOE. These requirements are presented following the conventions identified in Section 6.1.

5.1 Extended TOE Security Functional Components

This section specifies the extended SFRs for the TOE. The extended SFRs are organized by class. Table 10 identifies all extended SFRs implemented by the TOE

Table 10 – Extended TOE Security Functional Requirements

Name	Description
FDP_ACF_(EXT).1	Security Attribute Based Access Control
FPT_TST_(EXT).1	TSF Testing

5.1.1 Class FDP: User Data Protection

User data protection functions involve functionality for TOE security functions and TOE security function policies related to protecting user data. User data protection function class was modeled after the CC FDP: user data protection. The extended component FDP_ACF_(EXT).1: Security Attribute Based Access Control was modeled after the CC component FDP_ACF.1: Security Attribute Based Access Control.

FDP_ACF_(EXT)
1

Figure 2 – FDP_ACF_(EXT) Family Decomposition

5.1.1.1 Security Attribute Based Access Control (FDP_ACF_(EXT))

Family Behavior

This family describes the rules for the specific functions that can implement an access control policy named in Access control policy (FDP_ACC). Access control policy (FDP_ACC) specifies the scope of the policy.

Component Leveling

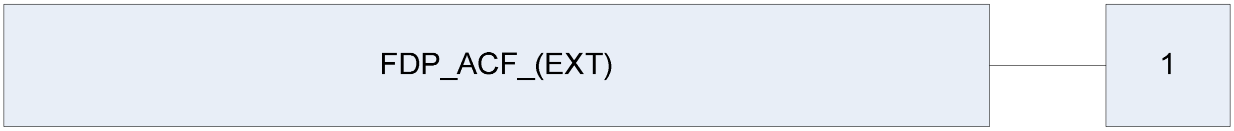

Figure 3 – FDP_ACF_(EXT) Family Decomposition

This family addresses security attribute usage and characteristics of policies. The component within this family is meant to be used to describe the rules for the function that implements the SFP as identified in Access control policy (FDP_ACC). The PP/ST author may also iterate this component to address multiple policies in the TOE.

Management: FDP_ACF_(EXT).1

The following actions could be considered for the management functions in FMT:

- Managing the attributes used to make explicit access or denial based decisions.

Audit: FDP_ACF_(EXT).1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- Minimal: Successful requests to perform an operation on an object covered by the SFP
- Basic: All requests to perform an operation on an object covered by the SFP.
- Detailed: The specific security attributes used in making an access check.

FDP_ACF_(EXT).1 Security Attribute Based Access Control

Hierarchical to: **[No other components]**

Dependencies: FDP_ACC.1 – Subset access control

FMT_MSA.3 – Static attribute initialization

FDP_ACF_(EXT).1.1

The TSF shall perform an access control decision and [selection of one or more by ST Author: enforce the decision, provide the decision] based on the [assignment: Access Control Policy] to objects based on the following: [assignment: list of subjects and objects controlled under the Authorization Server Access Control Policy, and for each, the relevant security attributes].

FDP_ACF_(EXT).1.2

The TSF shall [selection of one by ST Author: *enforce, provide an access control decision based on*] the following rules to determine if an operation among controlled subjects and controlled objects is allowed [assignment by ST Author: *rules governing access among controlled subject and controlled objects using controlled operations on controlled objects*].

FDP_ACF_(EXT).1.3

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [selection: *[assignment: rules, based on security attributes, that explicitly authorize access of subjects to objects], "no additional rules"*]

FDP_ACF_(EXT).1.4

The TSF shall explicitly deny access of subjects to objects based on the [selection: *[assignment: rules, based on security attributes, that explicitly deny access of subjects to objects], "no additional explicit denial rules"]*.

Application Note: This requirement (FDP_ACF_(EXT).1) is applicable only if the TOE enforces or provides an access control decision. If the TOE acts only as attribute authority, then this requirement is not applicable.

5.1.2 Class FPT: Protection of the Toe Security Functions

This class contains families of functional requirements that relate to the integrity and management of the mechanisms that constitute the TSF and to the integrity of TSF data. The extended component FPT_TST_(EXT).1: TSF Testing was modeled after the CC component FPT_TST.1: Internal TSF without FPT_TST.1.

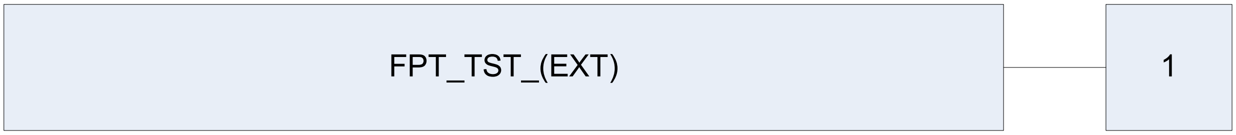

Figure 4 – FPT_TST_(EXT) Family Decomposition

5.1.2.1 TSF Testing (FPT_TST_(EXT))

Family Behavior

The requirements of this family are needed to detect the corruption of TSF executable code (i.e. TSF software) and TSF data by various failures that do not necessarily stop the TOE's operation (which would be handled by other families). These self-tests must be performed because these failures may not necessarily be prevented. Such failures can occur either because of unforeseen failure modes or associated oversights in the design of hardware, firmware, or software, or because of malicious corruption of the TSF due to inadequate logical and/or physical protection.

Component Leveling

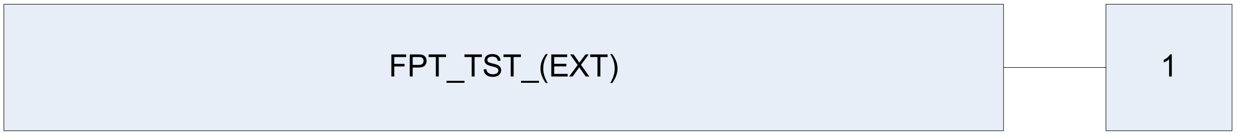

Figure 5 – FPT_TST_(EXT) Family Decomposition

FPT_TST_(EXT).1 TSF testing, provides the ability to verify the integrity of TSF data and executable code.

Management: FPT_TST_(EXT).1

- Management of the conditions under which TSF self-testing occurs.

Audit: FPT_TST_(EXT).1

The following actions could be considered for the management functions in FMT:

- Basic: Execution of the TSF self tests and the results of the tests.

FPT_TST_(EXT).1 TSF testing

Hierarchical to: No other components

Dependencies: No other dependencies

FPT_TST_(EXT).1.1

The TSF shall provide security administrator with the capability to verify the integrity of the following TSF data: [selection: *[assignment: parts of TSF], TSF data*].

FPT_TST_(EXT).1.2

The TSF shall provide security administrator with the capability to verify the integrity of stored TSF executable code.

5.2 Extended TOE Security Assurance Components

No extended Security Assurance Requirements have been defined for this Security Target.

6 Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) met by the TOE. These requirements are presented following the conventions identified in Section 6.1.

6.1 Conventions

There are several font variations used within this ST with respect to the Common Criteria standard and Protection Profile. Selected presentation choices are discussed here to aid the Security Target reader.

The CC allows for assignment, refinement, selection and iteration operations to be performed on security functional requirements. All of these operations are used within this ST. These operations are performed as described in Parts 2 and 3 of the CC, and are shown as follows:

- Completed assignment statements are identified using [*italicized text within brackets*].
- Completed selection statements are identified using [<u>*underlined italicized text within brackets*</u>].
- Refinements are identified using **bold text**. Any text removed is stricken (Example: ~~TSF Data~~) and should be considered as a refinement.
- Extended Functional and Assurance Requirements are identified using "EXT_" at the end of the short name.

Iterations are identified by appending a letter in parentheses following the component title. For example, FAU_GEN.1(a) Audit Data Generation would be the first iteration and FAU_GEN.1(b) Audit Data Generation would be the second iteration.

6.2 Security Functional Requirements

This section specifies the SFRs for the TOE. This section organizes the SFRs by CC class. Table 11 identifies all SFRs implemented by the TOE and indicates the ST operations performed on each requirement.

Table 11 – TOE Security Functional Requirements

Name	Description	Selection	Assignment	Refinement	Iteration
FAU_GEN.1	Audit Data Generation	✓	✓	✓	
FAU_GEN.2	User Identity Association				
FDP_ACC.1	Access Control Policy		✓		
FDP_ACF_(EXT).1	Access Control Functions	✓	✓		
FDP_RIP.2	Full Residual Information Protection	✓			
FIA_AFL.1	Authentication Failure Handling	✓	✓	✓	
FIA_ATD.1(1)	User Attribute Definition - Administrator		✓	✓	✓

Name	Description	Selection	Assignment	Refinement	Iteration
FIA_ATD.1(2)	User Attribute Definition - Principal		✓	✓	✓
FIA_ATD.1(3)	User Attribute Definition - Authorized Application		✓	✓	✓
FIA_SOS.1	Verification of Secrets		✓		
FIA_UAU.2	Timing of Authentication				
FIA_UID.2	Timing of Identification				
FMT_MOF.1(1)	Management of Security Functions Behavior (Access Policy)	✓	✓		✓
FMT_MOF.1(2)	Management of Security Functions Behavior (Authorized Applications)	✓	✓		✓
FMT_MOF.1(3)	Management of Security Functions Behavior (Audit)	✓	✓		✓
FMT_MSA.1(1)	Management of Security Attributes - Attribute Management	✓	✓		✓
FMT_MSA.1(2)	Management of Security Attributes - Attribute Authority	✓	✓		✓
FMT_MSA.2	Secure Security Attributes		✓		
FMT_MSA.3	Static Attribute Initialization	✓	✓		
FMT_MTD.1	Management of TSF Data	✓	✓		
FMT_SMF.1	Specification of Management Functions		✓		
FMT_SMR.1	Security Management Roles		✓		
FPT_TST_(EXT)1.1	TSF Testing	✓	✓		

Name	Description	Selection	Assignment	Refinement	Iteration
FTA_TAB.1	Default TOE Access Banners			✓	

6.2.1 Class FAU: Security Audit

FAU_GEN.1 Audit Data Generation

Hierarchical to: No other components.

FAU_GEN.1.1

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

- Start-up and shutdown of the audit functions;
- All auditable events, for the [*basic*] level of audit as identified in **Table 12 – Auditable Events;**
- [*no additional events*].

Table 12 – Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents (As Needed)
FDP_ACF_(EXT).1	All requests to perform an operation on an object covered by the SFP.	The specific security attributes used in making an access check.
FIA_AFL.1.1	The reaching of the threshold for the unsuccessful authentication attempts.	The claimed identity of the user attempting to gain access.
FIA_AFL.1.2	The actions (e.g., disabling of a terminal) taken and the subsequent, if appropriate, restoration to the normal state (e.g., re-enabling of a terminal).	The claimed identity of the user attempting to gain access.
FIA_SOS.1	Rejection or acceptance by the TSF of any tested secret.	Identification of any changes to the defined quality metrics.
FIA_UAU.2	All use of the authentication mechanism.	Claimed identity of user being authenticated, if that used exists in PADS.
FIA_UID.2	All use of the user identification mechanism, including the user identity provided.	Claimed identity of the user using the identification mechanism, if that user exists in PADS.
FMT_MOF.1(1)	All modifications to the access policy settings.	Identity of administrator making the modifications.
FMT_MOF.1(2)	All modification to the list of authorized applications.	Identity of the administrator making the modifications.

Requirement	Auditable Events	Additional Audit Record Contents (As Needed)
FMT_MOF.1(3)	All modifications to the audit behavior.	Identity of administrator making the modifications.
FMT_MSA.1(1)	All modifications of the values of security attributes.	Identity of administrator making the modifications.
FMT_MSA.1(2)	All queries of the values of security attributes.	Identity of authorized application making the queries.
FMT_MSA.2	All offered and rejected values for a security attribute.	All offered and accepted secure values for a security attribute.
FMT_MSA.3	All modifications of the default settings of permissive or restrictive rules.	Identity of the administrator making the modifications.
FMT_MSA.3	All modifications of the initial values of static security attributes.	Identity of the administrator making the modifications.
FMT_MTD.1	All modifications to the values of TSF data.	Identity of administrator making the modifications.
FMT_SMR.1	Modifications to the group of users that are part of a role.	Identity of administrator making the modifications
FMT_SMF.1	Use of the management functions.	Identity of administrator making the modifications.
FPT_TST_(EXT)1.1	Execution of the TSF self-tests and the results of the tests.	
FRU_FLT.1	Any failure detected by the TSF. Plus all TOE capabilities being disconnected due to a failure.	Identity of component that failed.

FAU_GEN.1.2

The TSF shall record within each audit record at least the following information:

- Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and
- For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*information specified in column three of* Table 12 *above*].

Application Note: The SFR text has been updated from the text used in the Protection Profile in order to reflect changes in CC 3.1 Rev 2.

Dependencies: FPT_STM.1 Reliable time stamps

FAU_GEN.2 User identity association

Hierarchical to: No other components.

FAU_GEN.2.1

For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

Dependencies: FAU_GEN.1 Audit data generation
FIA_UID.1 Timing of identification

6.2.2 Class FDP: User Data Protection

FDP_ACC.1 Subset access control

Hierarchical to: No other components.

FDP_ACC.1.1

The TSF shall enforce the [*Authorization Server Access Control Policy*] on [*principals as subjects, protected resources as objects, and all the operations among subjects and objects covered by the Authorization Server Access Control policy*].

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACF_(EXT).1 Security attribute based access control

Hierarchical to: No other components.

FDP_ACF_(EXT).1.1

The TSF shall perform an access control decision and [*enforce the decision, provide the decision*] based on the [*Authorization Server Access Control Policy*] to objects based on the following *[Entitlements and Smart Rules]*.

FDP_ACF_(EXT).1.2

The TSF shall [*enforce, provide an access control decision based on*] the following rules to determine if an operation among controlled subjects and controlled objects is allowed [*Entitlements or Smart Rules created by the Administrator in the Authorization Server Access Control Policy]*.

FDP_ACF_(EXT).1.3

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [the username or group membership of the user].

FDP_ACF_(EXT).1.4

The TSF shall explicitly deny access of subjects to objects based on the [the username or group membership of the user].

Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialization

FDP_RIP.2 Full residual information protection

Hierarchical to: FDP_RIP.1 Subset residual information protection

FDP_RIP.2.1

The TSF shall ensure that any previous information content of a resource is made unavailable upon the [*allocation of the resource to*] all objects.

Dependencies: No dependencies

6.2.3 Class FIA: Identification and Authentication

FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components.

FIA_AFL.1.1

The TSF shall detect when [*a **security** administrator configurable positive integer within [an administratively assigned range of values]"*] unsuccessful authentication attempts occur related to administrators attempting to authenticate to the TOE, and [*principals authenticating to the TOE*].

FIA_AFL.1.2

When the defined number of unsuccessful authentication attempts has been [*met or surpassed*], the TSF shall [*prevent the principal from performing actions that require authentication until an action is taken by the Security Administrator*].

Dependencies: FIA_UAU.1 Timing of authentication

Application Note: The administrator can assign any limit he wishes to set for unsuccessful login attempts.

FIA_ATD.1(1) User attribute definition - Administrator

Hierarchical to: No other components.

FIA_ATD.1.1(1)

The TSF shall maintain the following list of security attributes belonging to individual **administrative** users:

- [*Administrative user identifier,*
- *Administrator class (i.e. Security Administrator vs. Audit Administrator)*],
- *Authentication data,*
- [*username, group membership, any user attribute a Smart Rule uses for an access decision*]].

FIA_ATD.1(2) User attribute definition - Principal

Hierarchical to: No other components.

FIA_ATD.1.1(2)

The TSF shall maintain the following list of security attributes belonging to individual **principal** users:

- [*User identifier,*
- *Group membership,*
- [*username, group membership, any user attribute a Smart Rule uses for an access decision*]].

FIA_ATD.1(3) User attribute definition – Authorized Programs

Hierarchical to: No other components.

FIA_ATD.1.1(3)

The TSF shall maintain the following list of security attributes belonging to individual **authorized applications**: [Application Name, Group Membership, any other administrator definable attribute used in a Smart Rule].

Dependencies: No dependencies

FIA_SOS.1 Verification of secrets

Hierarchical to: No other components.

FIA_SOS.1.1

The TSF shall provide a mechanism to verify that secrets meet [*the condition that passwords must contain a minimum of 8 alpha numeric characters with at least one numeric character, and shall not be reused within a Security Administrator defined window of password changes*].

Dependencies: No dependencies

FIA_UAU.2 User authentication before any action

Hierarchical to: FIA_UAU.1 Timing of authentication

FIA_UAU.2.1

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

Dependencies: FIA_UID.1 Timing of identification

FIA_UID.2 User identification before any action

Hierarchical to: FIA_UID.1 Timing of identification

FIA_UID.2.1

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

Dependencies: No dependencies

Application Note: The SFR text has been updated from the text used in the Protection Profile in order to reflect changes in CC 3.1 Rev 2.

6.2.4 Class FMT: Security Management

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

Hierarchical to: No other components.

FMT_MOF.1.1(1)

The TSF shall restrict the ability to [*determine the behavior of, modify the behavior of*] the functions [*Configure the Authorization Server Access Control Policy settings*] to [*the Security Administrator*].

Dependencies: FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

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

Hierarchical to: No other components.

FMT_MOF.1.1(2)

The TSF shall restrict the ability to [*determine the behavior of, modify the behavior of*] the functions [*Configure the list of Authorized Applications and specify their security attributes*] to [*the Security Administrator*].

Dependencies: FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MOF.1(3) Management of security functions behaviour (audit)

Hierarchical to: No other components.

FMT_MOF.1.1(3)

The TSF shall restrict the ability to [*determine the behavior of, disable, enable, modify the behavior of*] the functions [*related to the security audit generation*] to [*the Audit Administrator*].

Dependencies: FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

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

Hierarchical to: No other components.

FMT_MSA.1.1(1)

The TSF shall enforce the [*Authorization Server Access Control Policy*] to restrict the ability to [change_default, query, modify or delete] the security attributes [*associated with both principals and protected resources which are used for access control permission rules*] to [*a designated Security Administrator*].

Dependencies: [FDP_ACC.1 Subset access control or
FDP_IFC.1 Subset information flow control]
FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MSA.1(2) Management of security attributes – Attribute Authority

Hierarchical to: No other components.

FMT_MSA.1.1(2)

The TSF shall enforce the [*Authorization Server Access Control Policy*] to restrict the ability to [query] the security attributes [*associated with both principals and protected resources which are used for access control permission rules*] to [*a designated Authorized Application*].

Dependencies: [FDP_ACC.1 Subset access control or
FDP_IFC.1 Subset information flow control]
FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MSA.2 Secure security attributes

Hierarchical to: No other components.

FMT_MSA.2.1

The TSF shall ensure that only secure values are accepted for [*security attributes, in particular, user authentication passwords shall be considered insecure if they have been previously used within a Security Administrator configurable number of password changes*].

Dependencies: [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

FMT_MSA.3 Static attribute initialisation

Hierarchical to: No other components.

FMT_MSA.3.1

The TSF shall enforce the [*Authorization Server Access Control Policy]* to provide [*restrictive*] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2

The TSF shall allow the [*the Security Administrator*] 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

FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

FMT_MTD.1.1

The TSF shall restrict the ability to [*change, default, query, modify, delete, clear,*] the [*all TSF data, including system configuration files, and the advisory warning messaged referenced in FTA_TAB.1*] to [*the Security Administrator role*].

Dependencies: FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

FMT_SMF.1.1

The TSF shall be capable of performing the following management functions: *[*

- *modify the Authorization Server Access Control Policy*
- *configure the list of Authorized Applications and their attributes*
- *enable, disable, determine, and modify Audit functions*
- *modify, create, delete, and query attributes associated with principals*
- *query attributes associated with protected resources*
- *manage user roles]*

Dependencies: No Dependencies

FMT_SMR.1 Security roles

Hierarchical to: No other components.

FMT_SMR.1.1

The TSF shall maintain the roles [*user roles are customizable by the Administrator*].

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

Dependencies: FIA_UID.1 Timing of identification

6.2.5 Class FPT: Protection of the TSF

FPT_TST_(EXT).1 TSF testing

Hierarchical to: No other components.

FPT_TST_(EXT).1.1

The TSF shall provide security administrator with the capability to verify the integrity of the following TSF data: [*TOE system configuration files*].

FPT_TST_(EXT).1.2

The TSF shall provide security administrators with the capability to verify the integrity of stored TSF executable code.

Dependencies: No dependencies

6.2.6 Class FTA: TOE Access

FTA_TAB.1 Default TOE access banners

Hierarchical to: No other components.

FTA_TAB.1.1

Before establishing **an administrative** session, the TSF shall display an advisory warning message regarding unauthorized use of the TOE.

Dependencies: No dependencies

6.3 Security Assurance Requirements

This section defines the assurance requirements for the TOE. Assurance requirements are taken from the CC Part 3 and are EAL3+ augmented with ALC_FLR.2. Table 13 – Assurance Requirements summarizes the requirements.

Table 13 – Assurance Requirements

Assurance Requirements	
Class ALC : Life Cycle Support	ALC_CMC.3 Authorisation controls
	ALC_CMS.3 Implementation representation CM coverage
	ALC_DEL.1 Delivery procedures
	ALC_DVS.1 Identification of security measures
	ALC_LCD.1 Developer defined life-cycle model
	ALC_FLR.2 Flaw reporting procedures
Class ADV: Development	ADV_ARC.1 Security architecture description
	ADV_FSP.3 Functional specification with complete summary
	ADV_TDS.2 Architectural design
Class AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
Class ATE: Tests	ATE_COV.2 Analysis of coverage
	ATE_DPT.1 Testing: basic design
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing – sample
Class AVA: Vulnerability assessment	AVA_VAN.2 Vulnerability analysis

7 TOE Summary Specification

This section presents information to detail how the TOE meets the functional and assurance requirements described in previous sections of this ST.

7.1 TOE Security Functions

Each of the security requirements and the associated descriptions correspond to the security functions. Hence, each function is described by how it specifically satisfies each of its related requirements. This serves to both describe the security functions and rationalize that the security functions satisfy the necessary requirements.

Table 14 – Mapping of TOE Security Functions to Security Functional Requirements

TOE Security Function	SFR ID	Description
Security Audit	FAU_GEN.1	Audit Data Generation
	FAU_GEN.2	User Identity Association
User Data Protection	FDP_ACC.1	Access Control Policy
	FDP_ACF_(EXT).1	Access Control Functions
	FDP_RIP.2	Full Residual Information Protection
Identification & Authentication	FIA_AFL.1	Authentication Failure Handling
	FIA_ATD.1(1)	User Attribute Definition - Administrator
	FIA_ATD.1(2)	User Attribute Definition - Principal
	FIA_ATD.1(3)	User Attribute Definition - Authorized Application
	FIA_SOS.1	Verification of Secrets
	FIA_UAU.2	Timing of Authentication
	FIA_UID.2	Timing of Identification
Security Management	FMT_MOF.1(1)	Management of Security Functions Behavior (Access Policy)

TOE Security Function	SFR ID	Description
	FMT_MOF.1(2)	Management of Security Functions Behavior (Authorized Applications)
	FMT_MOF.1(3)	Management of Security Functions Behavior (Audit)
	FMT_MSA.1(1)	Management of Security Attributes - Attribute Management
	FMT_MSA.1(2)	Management of Security Attributes - Attribute Authority
	FMT_MSA.2	Secure Security Attributes
	FMT_MSA.3	Static Attribute Initialization
	FMT_MTD.1	Management of TSF Data
	FMT_SMF.1	Specification of Management Functions
	FMT_SMR.1	Security Management Roles
TOE Access	FTA_TAB.1	Default TOE Access Banners
Protection of the TOE Security Functions	FPT_TST_(EXT)1.1	TSF Testing

7.1.1 Security Audit

The Security Audit function provides the TOE with the functionality for generating audit records. As administrators manage and configure the TOE, their activities are tracked by recording audit records into the logs. All security-relevant configuration settings and changes are recorded to ensure accountability of the administrator's actions. The TOE can be configured to generate audit logs at four different audit levels. The minimum audit level required in order for the TOE to operate in the Common Criteria-evaluated mode is the highest level (level 40).

Table 15 provides a list of the auditable events and the audit level at which they are logged. The audit levels are cumulative: higher logging levels include all events recorded in the lower levels.

Table 15 – Audit Record Contents

	Level 10	Level 20	Level 30	Level 40
Authorization Server	• Server Startup • Invalid Argument • Data Store Error • Authentication Errors • Internal Error • Server Test Failure • Unknown Error	• Access Denied • Bas Password • Locked Out • Expired Account • Inactive Account • Failed Authentication • Password Expired • Entitlement Denied • Smart Rule Denied • Unknown User	• Access Allowed • Protected Resource • Entitlement Allowed • Valid User	• Server Test Succeeded • Unknown Resource • Unprotected Resource • Cache Preload Overflow
Entitlements Server	• Internal Error	• Logon Failed • Create Failure • Delete Failure • Modify Failure • Administrator Permissions Denied • Read Access Denied	• Administrator Logon • Create • Delete • Modify • Administrator Permissions • Read Access	• Startup Events and all events are recorded
Dispatcher/Key Server	• Server Startup • Internal Error • Unknown Error	• Dispatcher Down		• Dispatcher List Request • Register with Dispatcher • Send Session Key • New Session Key • Receive Session Key • Lead Key Server Selection

By default, all log output is stored in separate log files on the host computer (all servers are installed on the same host computer). However, if centralized logging is enabled, log output from the Access Manager Servers will be sent to the optional Access Manager Log Server, which will consolidate the logged events from the Access Manager Servers.

Each server has its own set of auditable events, and therefore has its own method of identifying each auditable event and recording it. RSA has provided appendices which helps the TOE Administrator understand the contents of each server's audit log. The appendices describing the contents of any auditable record for each server is located in the Installation and Configuration Guide for Access Manager in Appendices C, D, and F.

For example, an Authorization server audit log entry might have an event with code 1003 and the reason for the event code might be 1002. In this case, event code 1003 means that an Authorization Failure has occurred, and the code for the reason means that there was a bad password.

TOE Security Functional Requirements Satisfied: FAU_GEN.1, FAU_GEN.2

7.1.2 User Data Protection

The User Data Protection function implements functionality for TOE security functions and TOE security function policies related to protecting user data. The user data that the TOE is protecting is any resource(s) that the TOE is assigned by the administrator to protect.

The TOE uses its Authorization Server Access Control Policy to provide an access decision and enforce the decision on principals, protected resources, and all operations between the two. An access decision is provided and enforced by the TOE through the comparison of user attributes with the Authorization Server Access Control Policy composed of Entitlements and/or Smart Rules the administrator implemented. Additionally, the Security Administrator can explicitly allow or deny access of principals to resources based on their username or group membership.

Finally, the TOE ensures that any previous information content of a resource is made to be unavailable upon the allocation of the resource to all objects (*i.e.* system memory*)*.

TOE Security Functional Requirements Satisfied: FDP_ACC.1, FDP_ACF_(EXT).1, FDP_RIP.2

7.1.3 Identification and Authentication

The Identification and Authentication functions identify and authenticate users to the TOE. End users must identify and authenticate with the TOE anytime they wish to access a resource protected by the TOE. The TOE has the ability to work with multiple types of user data stores (such as LDAP) to identify a user.

The Security Administrator of the TOE will set a number of unsuccessful attempts a user has to authenticate himself. As the number of attempts is surpassed, the TOE will detect the event, and will lock the account with action pending on the Security Administrator’s behalf to unlock the account. This includes users attempting to authenticate when accessing a protected resource, and Administrators attempting to log directly into the TOE. Each principal has a unique set of attributes related to its respective account. These attributes can be used to determine access decisions.

In order to access any protected resource or administer the TOE, the user or administrator must be successfully identified and authenticated. Without successful identification and authentication, the TOE will not allow any actions to be performed. A user will typically identify himself with a username and authenticate himself with a password. The TOE will verify the secret information. In addition, the secret must be at the minimum of eight characters long and contain at least one number. The user cannot use the same password (after the previous password expires) within a certain period of time which is defined by the Security Administrator.

TOE Security Functional Requirements Satisfied: FIA_AFL.1, FIA_ATD.1(1), FIA_ATD.1(2), FIA_ATD.1(3), FIA_SOS.1, FIA_UAU.2, FIA_UID.2

7.1.4 Security Management

The Security Management features provide management and administration functions of the TOE for its administrators.

The TSF associates users with customizable roles. It is from the customizable permissions available, when creating a user, that the administrator can create a “Security Administrator” or an “Audit Administrator”. This can be done by assigning the Security Administrator permissions to modify the security functions of the TOE, and the Audit Administrator permissions to modify the Audit functions of the TOE.

The Security Administrator is the only user who can modify the Authorization Server Access Control Policy settings. Additionally, only the Security Administrator is authorized to modify security attributes, for principals or protected resources, used by the Authorization Server Access Control Policy to make access decisions. The TOE’s Authorization Server Access Control Policy ensures that only Authorized Applications have the ability to query security attributes (for the purpose of comparing principals’ and protected resources’ attributes for an access decision). The Security Administrator is the sole role capable of creating/modifying the TSF data, system configuration files, and access banner.

The Audit Administrator is the only user role which has the ability to modify the behavior of the audit generation functions.

The TOE ensures that only secure values are accepted for security attributes. Passwords are considered insecure if they have been used within a specific number of password changes (configurable by the Security Administrator) and would therefore not be accepted by the TOE.

In the case of a new principal being added to the data repository, all security attributes, which are used to make an access decision, are initially provided with a restrictive value. Only the Security Administrator can override the initial values for the security attributes.

TOE Security Functional Requirements Satisfied: FMT_MOF.1(1), FMT_MOF.1(2), FMT_MOF.1(3), FMT_MSA.1(1), FMT_MSA.1(2), FMT_MSA.2, FMT_MSA.3, FMT_MTD.1, FMT_SMF.1, FMT_SMR.1

7.1.5 Protection of the TSF

The Security Administrator is the sole role capable of modifying and verifying the integrity of the TOE system configuration files. Additionally, only the Security Administrator is capable of verifying the integrity of the stored TSF executable code. No other roles have access to view or modify any configuration or executable file. This allows the Security Administrator to ensure the TOE is operating as expected, and ensures that no rogue processes are executed, thus ensuring the secure operation of the TOE.

TOE Security Functional Requirements Satisfied: FPT_TST_(EXT).1

7.1.6 TOE Access

When an administrator initiates an administrative user session, the TOE displays an advisory message warning about unauthorized use of the TOE.

TOE Security Functional Requirements Satisfied: FTA_TAB.1

8 Rationale

8.1 Conformance Claims Rationale

This Security Target conforms to Parts 2 and 3 of the Common Criteria Standard for Information Technology Security Evaluations, version 3.1, revision 2. There are two extended SFRs contained within this ST: FDP_ACF_(EXT).1 and FPT_TST_(EXT).1.

This Security Target claims conformance with U.S. Government Protection Profile Authorization Server for Basic Robustness Environments. All SFRs identified in the PP are included in this ST, and all the operations applied to the SFRs derived from the PP are in accordance with the requirements of the PP.

Note: The U.S. Government Protection Profile Authorization Server in Basic Robustness Environments, Version 1.1 contains IT Environment Security Functional Requirements requiring that the IT Environment be compliant with the Controlled Access Protection Profile or an Operating System Protection Profile at the Basic Level of Robustness or Greater. As v3.1 of the Common Criteria Standard does not require Environmental SFRs, these SFRs have been removed, but the associated Environmental Objectives remain.

8.2 Security Objectives Rationale

This section provides a rationale for the existence of each threat, policy statement, and assumption that compose the Security Target. Sections 8.2.1, 8.2.2, and 8.2.3 demonstrate the mappings between the threats, polices, and assumptions to the security objectives is complete. The following discussion provides detailed evidence of coverage for each threat, policy, and assumption.

Table 16 – Mapping of TOE Security Objectives to Threats and Policies

Security Objectives for the TOE Threats and Policies	O.ADMIN_GUIDANCE	O.AUDIT_GENERATION	0.CORRECT_TSF_OPERATION	O.DISPLAY_BANNER	O.MANAGE	O.MEDIATE	O.PARTIAL_SELF_PROTECTION	O.RESIDUAL_INFORMATION	O.TOE_ACCESS	OE.AUDIT_PROTECTION	OE.CAPP_OS	OE.COMMS	OE.CRYPTOGRAPHY	OE.DISPLAY_BANNER	OE.FAULT_TOLERANCE	OE.IT_ACCESS	OE.LOWEXP	OE.NO_GENERAL_PURPOSE	OE.NO_TOE_BYPASS	OE.PRIORITY	OE.RESIDUAL_INFORMATION	OE.SCALABLE	OE.WEB_BROWSER_PP	OD.BASIC_ROBUSTNESS	OD.CONFIGURATION_IDENTIFICATION	OD.DOCUMENTED_DESIGN	OD.PARTIAL_FUNCTIONAL_TESTING	OD.VULNERABILITY_ANALYSIS	OE.MANAGE	OE.NO_EVIL	OE.PHYSICAL	OE.TOE_ENVIRONMENT_ACCESS
Threats																																
T.ACCIDENTAL_ADMIN_ERROR	✓																															
T.ACCIDENTAL_AUDIT_COMPROMISE							✓	✓			✓																					
T.ACCIDENTAL_CRYPTO_COMPROMISE													✓								✓											
T.LOW_PRIORITY	✓																			✓												
T.MASQUERADE						✓			✓																							
T.POOR_DESIGN																									✓	✓		✓				
T.POOR_IMPLEMENTATION																									✓		✓	✓				

Security Objectives for the TOE / Threats and Policies	O.ADMIN_GUIDANCE	O.AUDIT_GENERATION	0.CORRECT_TSF_OPERATION	O.DISPLAY_BANNER	O.MANAGE	O.MEDIATE	O.PARTIAL_SELF_PROTECTION	O.RESIDUAL_INFORMATION	O.TOE_ACCESS	OE.AUDIT_PROTECTION	OE.CAPP_OS	OE.COMMS	OE.CRYPTOGRAPHY	OE.DISPLAY_BANNER	OE.FAULT_TOLERANCE	OE.IT_ACCESS	OE.LOWEXP	OE.NO_GENERAL_PURPOSE	OE.NO_TOE_BYPASS	OE.PRIORITY	OE.RESIDUAL_INFORMATION	OE.SCALABLE	OE.WEB_BROWSER_PP	OD.BASIC_ROBUSTNESS	OD.CONFIGURATION_IDENTIFICATION	OD.DOCUMENTED_DESIGN	OD.PARTIAL_FUNCTIONAL_TESTING	OD.VULNERABILITY_ANALYSIS	OE.MANAGE	OE.NO_EVIL	OE.PHYSICAL	OE.TOE_ENVIRONMENT_ACCESS
T.POOR_TEST			✓																							✓	✓	✓				
T.RESIDUAL_DATA								✓																								
T.TSF_COMPROMISE					✓		✓	✓																								
T.UNATTENDED_SESSION									✓																							✓
T.UNAUTHORIZED_ACCESS						✓																										
T.UNIDENTIFIED_ACTIONS		✓									✓																					
Policies																																
P.ACCESS_BANNER				✓										✓																		
P.ACCOUNTABILITY		✓							✓	✓	✓																					

Security Objectives for the TOE **Threats and Policies**	O.ADMIN_GUIDANCE	O.AUDIT_GENERATION	0.CORRECT_TSF_OPERATION	O.DISPLAY_BANNER	O.MANAGE	O.MEDIATE	O.PARTIAL_SELF_PROTECTION	O.RESIDUAL_INFORMATION	O.TOE_ACCESS	OE.AUDIT_PROTECTION	OE.CAPP_OS	OE.COMMS	OE.CRYPTOGRAPHY	OE.DISPLAY_BANNER	OE.FAULT_TOLERANCE	OE.IT_ACCESS	OE.LOWEXP	OE.NO_GENERAL_PURPOSE	OE.NO_TOE_BYPASS	OE.PRIORITY	OE.RESIDUAL_INFORMATION	OE.SCALABLE	OE.WEB_BROWSER_PP	OD.BASIC_ROBUSTNESS	OD.CONFIGURATION_IDENTIFICATION	OD.DOCUMENTED_DESIGN	OD.PARTIAL_FUNCTIONAL_TESTING	OD.VULNERABILITY_ANALYSIS	OE.MANAGE	OE.NO_EVIL	OE.PHYSICAL	OE.TOE_ENVIRONMENT_ACCESS
P.BASIC_ROBUSTNESS																								✓								
P.CAPP_OS											✓																					
P.COMMS												✓																				
P.CRYPTOGRAPHY													✓																			
P.HIGH_AVAILABILITY															✓					✓												
P.NO_GENERAL_PURPOSE																		✓														
P.TOE_ENVIRONMENT_ACCESS																																✓
P.WEB_BROWSER_PP																							✓									
Assumptions	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

Threats and Policies / Security Objectives for the TOE	O.ADMIN_GUIDANCE	O.AUDIT_GENERATION	0.CORRECT_TSF_OPERATION	O.DISPLAY_BANNER	O.MANAGE	O.MEDIATE	O.PARTIAL_SELF_PROTECTION	O.RESIDUAL_INFORMATION	O.TOE_ACCESS	OE.AUDIT_PROTECTION	OE.CAPP_OS	OE.COMMS	OE.CRYPTOGRAPHY	OE.DISPLAY_BANNER	OE.FAULT_TOLERANCE	OE.IT_ACCESS	OE.LOWEXP	OE.NO_GENERAL_PURPOSE	OE.NO_TOE_BYPASS	OE.PRIORITY	OE.RESIDUAL_INFORMATION	OE.SCALABLE	OE.WEB_BROWSER_PP	OD.BASIC_ROBUSTNESS	OD.CONFIGURATION_IDENTIFICATION	OD.DOCUMENTED_DESIGN	OD.PARTIAL_FUNCTIONAL_TESTING	OD.VULNERABILITY_ANALYSIS	OE.MANAGE	OE.NO_EVIL	OE.PHYSICAL	OE.TOE_ENVIRONMENT_ACCESS
A.IT_ACCESS																✓																
A.LOWEXP																	✓															
A.MANAGE																													✓			
A.NO_EVIL																														✓		
A.NO_TOE_BYPASS																			✓													
A.PHYSICAL																															✓	
A.SCALABLE																						✓										

8.2.1 Security Objectives Rationale Relating to Threats

Table 17 – Threats:Objectives Mapping

Threats	Objectives	Rationale
T.ACCIDENTAL_ADMIN_ERROR An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.	O.ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure management.	O.ADMIN_GUIDANCE, which states that the TOE will provide administrators with the necessary information for secure management. This helps to mitigate this threat by ensuring the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner and to provide the administrator with instructions to ensure the TOE was not corrupted during the delivery process. Having this guidance helps to reduce the mistakes that an administrator might make that could cause the TOE to be configured in a way that is insecure.
T.ACCIDENTAL_AUDIT_COMPROMISE An administrative user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.	OE.CAPP_OS Operating systems the TOE operates on top of must be compliant with the Controlled Access Protection Profile. The operating system will therefore provide all the capabilities outlined in the CAPP security function requirements and will have been evaluated against the CAPP assurance requirements.	OE.CAPP_OS, which states that Operating systems the TOE operates on top of must be compliant with the Controlled Access Protection Profile. This contributes to mitigating this threat by controlling access to the audit trail. No one is allowed to modify audit records, and only an authorized administrator is allowed to delete the audit trail. The operating system has the capability to prevent auditable actions from occurring if the audit trail is full.
	O.PARTIAL_SELF_PROTECTION The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces.	O.PARTIAL_SELF_PROTECTION, which states that the TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces. This contributes to countering this threat by ensuring that the TSF can protect itself from users. If the TSF could not maintain security domains of subjects in the TOE Scope of Control, it could not be trusted to control access to the resources under its control, which includes the audit trail.
	O.RESIDUAL_INFORMATION	O.RESIDUAL_INFORMATION, which

Threats	Objectives	Rationale
	The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.	states that the TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated. This prevents a user not authorized to read the audit trail from access to audit information that might otherwise be persistent in a TOE resource (e.g., memory). By ensuring the TOE prevents residual information in a resource, audit information will not become available to any user or process except those explicitly authorized for that data.
T.ACCIDENTAL_CRYPTO_COMPROMISE An administrative user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the cryptographic mechanisms and the data protected by those mechanisms.	OE.CRYPTOGRAPHY The IT environment components shall use NIST FIPS 140-2 validated cryptographic modules if they provide cryptographic services.	OE.CRYPTOGRAPHY, which states that the IT environment components shall use NIST FIPS 140-2 validated cryptographic modules if they provide cryptographic services. This provides assurance that the cryptographic modules do not permit accidental compromise.
	OE.RESIDUAL_INFORMATION The IT Environment will ensure that any information contained in a protected resource is not released when the resource is reallocated.	OE.RESIDUAL_INFORMATION, which states that the IT Environment will ensure that any information contained in a protected resource is not released when the resource is reallocated. This mitigates the possibility of malicious users or processes from gaining inappropriate access to cryptographic data, including keys. This objective ensures that the cryptographic data does not reside in a resource that has been used by the cryptographic module and then reallocated to another process.
T.LOW_PRIORITY A low priority process may exhaust resources required by the TOE.	O.ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure management.	O.ADMIN_GUIDANCE, which states that the TOE will provide administrators with the necessary information for secure management. This will instruct administrators to configure the IT Environment to support prioritization of the TOE's resources.
	OE.PRIORITY The IT Environment will provide prioritization of resources to support	OE.PRIORITY, which states that the IT Environment will provide prioritization of resources to support the TOE. This mitigates the threat by

Threats	Objectives	Rationale
	the TOE.	ensuring that the TOE can have a higher priority than other processes in the Environment.
T.MASQUERADE A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.	O.MEDIATE The TOE must protect user data in accordance with its security policy.	O.MEDIATE, which states that the TOE must protect user data in accordance with its security policy. This works to mitigate this threat by constraining how and when authorized users can access the TOE.
	O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE.	O.TOE_ACCESS, which states that the TOE will provide mechanisms that control a user's logical access to the TOE. This mitigates this threat by controlling the logical access to the TOE and its resources. By identifying and authenticating all users (and principals if the TOE acts as an authentication server) this objective helps mitigate the possibility of a user attempting to login and masquerade as an authorized user or an unauthorized entity accessing a protected resource. In addition, this objective provides the administrator the means to control the number of failed login attempts a user can generate before an account is locked out, further reducing the possibility of a user gaining unauthorized access to the TOE.
T.POOR_DESIGN Unintentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a casually mischievous user or program.	OD.CONFIGURATION_IDENTIFICATION The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly.	OD.CONFIGURATION_IDENTIFICATION, which states that the configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly. This counters this threat by requiring the developer have a configuration item, a reference for each version of the TOE, and a Configuration Management (CM) system with CM documentation. The developer is also required to establish flaw remediation procedures for accepting and acting upon user reports of security flaws and ensuring that any reported flaws are corrected.
	OD.DOCUMENTED_DESIGN The design of the TOE is adequately	OD.DOCUMENTED_DESIGN, which states that the design of the TOE is adequately and accurately

Threats	Objectives	Rationale
	and accurately documented.	documented. This counters this threat, to a degree, by requiring that the TOE be developed using a documented design engineering approach. By providing at least a high level of informal documenting of the security mechanisms in the TOE, the design of the TOE can be understood, which increases the chances that design errors will be discovered.
	OD.VULNERABILITY_ANALYSIS The TOE will undergo vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws.	OD.VULNERABILITY_ANALYSIS, which states that the TOE will undergo vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws. This ensures that the design of the TOE is analyzed by the developer for obvious design flaws. Having the developer perform a vulnerability assessment and document that known vulnerabilities cannot be exploited may find errors in the design that may have been left undiscovered.
T.POOR_IMPLEMENTATION Unintentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a casually mischievous user or program.	OD.CONFIGURATION_IDENTIFICATION The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly.	OD.CONFIGURATION_IDENTIFICATION, which states that the configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly. This contributes to this objective by requiring the developer have a configuration item, a reference for each version of the TOE, and a Configuration Management (CM) system with CM documentation. The developer is also required to establish flaw remediation procedures for accepting and acting upon user reports of security flaws and ensuring that any reported flaws are corrected. Following a good CM process during development will reduce the risk of implementation errors.
	OD.PARTIAL_FUNCTIONAL_TESTING The TOE will undergo some security functional testing that demonstrates the TSF satisfies its security functional requirements.	O. PARTIAL_FUNCTIONAL_TESTING, which states that the TOE will undergo security functional testing that demonstrates the TSF satisfies some of its security functional requirements. This increases the likelihood that any errors that do exist in the implementation (with respect to the

Threats	Objectives	Rationale
		functional specification and high level design) will be discovered through testing.
	OD.VULNERABILITY_ANALYSIS The TOE will undergo vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws.	OD.VULNERABILITY_ANALYSIS, which states that the TOE will undergo vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws. This ensures that the design of the TOE is analyzed for obvious design flaws buy the developer. Having the developer perform a vulnerability assessment and document that known vulnerabilities cannot be exploited may find errors in the design that may have been left undiscovered.
T.POOR_TEST Developers or test engineers may implement tests that are insufficient to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being discovered thereby causing potential security vulnerabilities.	O.CORRECT_TSF_OPERATION The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site.	O.CORRECT_TSF_OPERATION, which states that the TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site. This provides administrators with the capability to verify the integrity TSF data, including stored TSF executable code and configuration files.
	OD.DOCUMENTED_DESIGN The design of the TOE is adequately and accurately documented.	OD.DOCUMENTED_DESIGN, which states that the TOE's design will be adequately and accurately documented. This ensures the existence of design documentation sufficient to permit adequate testing of the TOE.
	OD.PARTIAL_FUNCTIONAL_TESTING The TOE will undergo some security functional testing that demonstrates the TSF satisfies its security functional requirements.	OD.PARTIAL_FUNCTIONAL_TESTING, which states that the TOE will undergo security functional testing that demonstrates the TSF satisfies its security functional requirements. This ensures that functional testing is performed to ensure the TSF satisfies the security functional requirements and demonstrates that the TOE's security mechanisms operate as documented. While functional testing serves an important purpose, it does not ensure the TSFI cannot be used in unintended ways to circumvent the TOE's security policies.
	OD.VULNERABILITY_ANALYSIS	OD.VULNERABILITY_ANALYSIS, which states that the TOE will undergo

Threats	Objectives	Rationale
	The TOE will undergo vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws.	vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws. This ensures that the design of the TOE is analyzed by the developer for obvious design flaws. Having the developer perform a vulnerability assessment and document that known vulnerabilities cannot be exploited may find errors in the design that may have been left undiscovered.
T.RESIDUAL_DATA A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.	O.RESIDUAL_INFORMATION, which states that the TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated. This counters this threat by ensuring that TSF data and user data is not persistent when resources are released by one user/process and allocated to another user/process. This means that network packets will not have residual data from another packet due to the padding of a packet. This ensures successful access control decisions make for one user does not carry over to the next user.
T.TSF_COMPROMISE An attacking user or process may cause, through an unsophisticated attack, TSF data, or executable code to be inappropriately accessed (viewed, modified, or deleted).	O.MANAGE The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	O.MANAGE, which states that the TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use. This defines an access control policy to control access to TSF data or the resources being protected by the TOE. This objective is used to dictate who is able to view and modify TSF data, as well as the behavior of TSF functions.
	O.PARTIAL_SELF_PROTECTION The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces.	O.PARTIAL_SELF_PROTECTION, which states that the TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces. This contributes to countering this threat by ensuring that the TSF can protect itself from users. If the TSF could not maintain security domains of subjects in the TOE Scope of Control, it could not be trusted to control access to the resources under

Threats	Objectives	Rationale
		its control. It requires that the TSF be able to protect itself from tampering and that the security mechanisms in the TSF cannot be bypassed.
	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.	O.RESIDUAL_INFORMATION, which states that the TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated. This counters this threat by ensuring that TSF data and user data is not persistent when resources are released by one user/process and allocated to another user/process. This means that network packets will not have residual data from another packet due to the padding of a packet. This ensures successful access control decisions make for one user does not carry over to the next user.
T.UNATTENDED_SESSION A user may gain unauthorized access to an unattended session.	OE.TOE_ENVIRONMENT_ACCESS The TOE environment will provide mechanisms that control a user's logical access to the environmental components.	OE.TOE_ENVIRONMENT_ACCESS, which states that the TOE environment will provide mechanisms that control a user's logical access to the environmental components. This helps to mitigate this threat by including mechanisms that place controls on user's sessions. Local administrator's sessions are locked and remote sessions are dropped after a Security Administrator defined time period of inactivity. Locking the local administrator's session reduces the opportunity of someone gaining unauthorized access the session when the console is unattended. Dropping the connection of a remote session (after the specified time period) reduces the risk of someone accessing the remote machine where the session was established, thus gaining unauthorized access to the session.
	O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE.	O.TOE_ACCESS, which states that the TOE will provide mechanisms that control a user's logical access to the TOE, including the locking of sessions.
T.UNAUTHORIZED_ACCESS A user or application may gain	O.MEDIATE The TOE must protect user data in	O.MEDIATE, which states that the TOE must protect user data in accordance with its security policy.

Threats	Objectives	Rationale
access to the data for which they are not authorized according to the TOE security policy.	accordance with its security policy.	This works to mitigate this threat by ensuring that all requests to access user data, or data being protected by the TOE, are subject to an Authorization Server access control policy. A TOE policy engine enforces rules to determine if an operation among controlled subjects and controlled objects is allowed based on the security attributes of the user and the object. The TOE requires successful authentication to the TOE prior to gaining access to administrative services on or mediated by the TOE to protected resources. Communications between the TOE components must be protected from unauthorized disclosure to ensure integrity and confidentiality of the user data. Lastly, the TSF must ensure that all configured enforcement functions (authentication, access control rules, etc.) must be invoked prior to allowing a user to gain access to TOE or TOE mediated services. The TOE restricts the ability to modify the security attributes associated with access control rules, access to authenticated and unauthenticated services, etc to the Security Administrator. This feature ensures that no other user can modify the access control policy to bypass the intended TOE security policy.
T.UNIDENTIFIED_ACTIONS The administrator may not have the ability to notice potential security violations, this limiting the administrator's ability to identify and take action against a possible security breach,	O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security-relevant events associated with users.	O.AUDIT_GENERATION, which states that the TOE will provide the capability to detect and create records of security-relevant events associated with users. This means that actions that might result from security violations will be audited, and thus may be detected by administrators.
	OE.CAPP_OS Operating systems the TOE operates on top of must be compliant with the Controlled Access Protection Profile. The operating system will therefore provide all the capabilities outlined in the CAPP security function requirements and will have been evaluated against the CAPP assurance requirements.	OE.CAPP_OS, which states that operating systems in which the TOE operates must be compliant with the Controlled Access Protection Profile. This helps to mitigate this threat by providing the Security Administrator with a set of rules for monitoring the audited events and based upon these rules can indicate a potential violation of the TSP. A required minimum set of configurable audit events that could indicate a potential security violation. By configuring these auditable events,

Threats	Objectives	Rationale
		when the Security or Audit Administrator reviews the audit records, they can determine the occurrences of these events (e.g. set number of authentication failures, etc.). A search and sort capability provides an efficient mechanism for the Audit Administrator to view pertinent audit information.

Every Threat is mapped to one or more Objective in the table above. This complete mapping demonstrates that the defined security objectives counter all defined threats.

8.2.2 Security Objectives Rationale Relating to Policies

Table 18 – Policies:Objectives Mapping

Policies	Objectives	Rationale
P.ACCESS_BANNER The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by access the system.	O.DISPLAY_BANNER The TOE will display an advisory warning regarding use of the TOE to the administrative users.	O.DISPLAY_BANNER, which states that the TOE will display an advisory warning regarding use of the TOE to administrators.
	OE.DISPLAY_BANNER The underlying operating system of the TOE will display an advisory warning regarding use of the TOE to administrative users logging on the platform where the TOE software is installed.	OE.DISPLAY_BANNER, which states that the underlying operating system of the TOE will display an advisory warning regarding use of the TOE to administrative users logging on the platform where the TOE software is installed.
P.ACCOUNTABILITY The TOE shall log all actions by authorized users such that the authorized users can be held accountable for their actions within the TOE.	OE.AUDIT_PROTECTION The IT Environment will provide the capability to protect audit information.	OE.AUDIT_PROTECTION, which states that the IT Environment will provide the capability to protect audit information.
	O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security-relevant events associated with users.	O.AUDIT_GENERATION, which states that the TOE will provide the capability to detect and create records of security-relevant events associated with users. This addresses this policy by providing the Security Administrator with the capability of configuring the audit mechanism to

Policies	Objectives	Rationale
		record the actions of a specific user.
	OE.CAPP_OS Operating systems the TOE operates on top of must be compliant with the Controlled Access Protection Profile. The operating system will therefore provide all the capabilities outlined in the CAPP security function requirements and will have been evaluated against the CAPP assurance requirements.	OE.CAPP_OS, which states that Operating systems the TOE operates on top of must be compliant with the Controlled Access Protection Profile. This plays a role in supporting this policy by requiring the IT environment to provide a reliable time stamp (configured locally by the Security Administrator or via an external NTP server). The audit mechanism is required to include the current date and time in each audit record. All audit records that include the user ID, will also include the date and time that the event occurred.
	O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE.	O.TOE_ACCESS, which states that the TOE will provide mechanisms that control a user's logical access to the TOE. This supports this policy by requiring the TOE to identify and authenticate all authorized users prior to allowing any TOE access or access to any TOE protected resource that the TOE is mediating access on behalf of the users.
P.BASIC_ROBUSTNESS The TOE must be developed in accordance with the Basic Robustness guidelines.	OD.BASIC_ROBUSTNESS The TOE shall be developed in accordance with the Basic Robustness requirements.	OD. BASIC_ROBUSTNESS, which directly enforces P. BASIC_ROBUSTNESS.
P.CAPP_OS The operating system the TOE operates on top of must be evaluated to be compliant with the Controlled Access Protection Profile.	OE.CAPP_OS Operating systems the TOE operates on top of must be compliant with the Controlled Access Protection Profile. The operating system will therefore provide all the capabilities outlined in the CAPP security function requirements and will have been evaluated against the CAPP assurance requirements.	OE.CAPP_OS, which states that operating systems the TOE operates on top of must be compliant with the Controlled Access Protection Profile. OE.CAPP_OS directly enforces P.CAPP_OS.
P.COMMS Communications exist between the TOE components (internally) and between the TOE components and the IT components.	OE.COMMS Sites deploying the TOE will ensure that adequate communications exist between the TOE components (internally) and between the TOE	OE.COMMS, which states that Sites deploying the TOE will provide adequate communications exist between the TOE components (internally) and between the TOE components and the IT components. OE.COMMS directly enforces

Policies	Objectives	Rationale
	components and the IT components.	P.COMMS.
P.CRYPTOGRAPHY Only NIST FIPS 140-2 validated cryptography (methods and implementations) are acceptable for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e., encryption, decryption, signature, hashing, key exchange, and random number generation services).	OE.CRYPTOGRAPHY The IT environment components shall use NIST FIPS 140-2 validated cryptographic modules if they provide cryptographic services.	OE.CRYPTOGRAPHY, which states that the IT environment components shall use NIST FIPS 140-2 validated cryptographic modules if they provide cryptographic services. These services will provide confidentiality and integrity protection of TSF data while in transit to between software components of the TOE and for TSF data being transfer to/from trusted IT environment components.
P.HIGH_AVAILABILITY The TOE shall include providing resource allocations to support priority of service and fault tolerance.	OE.FAULT_TOLERANCE The IT environment will provided limited capabilities to support degraded fault tolerance and fail over for some TOE components.	OE.FAULT_TOLERANCE, which states that the IT environment will provide limited capabilities to support degraded fault tolerance and fail over for some TOE components. This helps satisfy the policy by ensuring that when a single instance of authorization server policy engine fails, operations are continued by an alternate authorization server policy engine.
	OE.PRIORITY The IT Environment will provide prioritization of resources to support the TOE.	OE.PRIORITY, which states that the IE Environment will provide prioritization of resources to support the TOE. This will ensure that priority of service is available to the TOE.
P.NO_GENERAL_PURPOSE There will be no general-purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the hardware platforms that the TOE administrative and authorization policy engine software are installed. If Authorization Server "Agent" software is part of the TOE, then the system on which the Agent operates is exempt from the assumption.	OE.NO_GENERAL_PURPOSE There will be no general-purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the hardware platforms that the TOE administrative and authorization policy engine software are installed. This objective does not apply to agent software that might reside on a web server.	OE.NO_GENERAL_PURPOSE, which states that there will be no general-purpose computing or storage repository capabilities available on the hardware platforms on which the TOE software is installed. OE.NO_GENERAL_PURPOSE directly enforces P.NO_GENERAL_PURPOSE.
P.TOE_ENVIRONMENT_ACCESS The TOE environment will provide	OE.TOE_ENVIRONMENT_ACCESS The TOE environment will provide mechanisms that control a user's	OE.TOE_ENVIRONMENT_ACCESS, which states that the TOE environment will provide mechanisms that control a user's logical access to

Policies	Objectives	Rationale
mechanisms that control a user's logical access to the TOE environment components.	logical access to the environmental components.	the environmental components. OE.TOE_ENVIRONMENT_ACCESS directly enforces P.TOE_ENVIRONMENT_ACCESS.
P.WEB_BROWSER_PP If administrators use a web browser to access the TOE for remote administration, they must use software that has been evaluated to the Web Browser Protection Profile.	OE.WEB_BROWSER_PP If administrators use a web browser to access the TOE for remote administration, they must to use software that has been evaluated to the Web Browser Protection Profile.	OE. WEB_BROWSER_PP, which directly enforces P. WEB_BROWSER_PP.

Every policy is mapped to one or more Objective in the table above. This complete mapping demonstrates that the defined security objectives enforce all defined policies.

8.2.3 Security Objectives Rationale Relating to Assumptions

Table 19 – Assumptions:Objectives Mapping

Assumptions	Objectives	Rationale
A.IT_ACCESS The TOE has access to all the IT System data it needs to perform its functions.	OE.IT_ACCESS Sites deploying the TOE will ensure the TOE has access to all the IT System data it needs to perform its functions.	OE.IT_ACCESS, which states that Sites deploying the TOE will ensure the TOE has access to all the IT System data it needs to perform its functions. OE.IT_ACCESS directly upholds A.IT_ACCESS.
A.LOWEXP The threat of malicious attacks aimed at discovering exploitable vulnerabilities is considered low.	OE.LOWEXP Site deploying the TOE will establish a protective environment where the threat of malicious attacks aimed at discovering exploitable vulnerabilities is considered low.	OE.LOWEXP, which states that Site deploying the TOE will establish a protective environment where the threat of malicious attacks aimed at discovering exploitable vulnerabilities is considered low.
A.MANAGE There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.	OE.MANAGE The TOE environmental components will provide all the functions, facilities and competent individuals necessary to support the administrators in their management of the security of the environment, and restrict these functions and facilities from	OE.MANAGE, which states that the TOE environmental components will provide all the functions, facilities and competent individuals necessary to support the administrators in their management of the security of the environment, and restrict these functions and facilities from unauthorized use.

Assumptions	Objectives	Rationale
	unauthorized use.	
A.NO_EVIL Administrators are non-hostile, appropriately trained and follow all administrator guidance.	OE.NO_EVIL Sites using the TOE shall ensure that administrators are non-hostile, appropriately trained and follow all administrator guidance.	OE.NO_EVIL, which states that sites using the TOE shall ensure that administrators are non-hostile, appropriately trained and follow all administrator guidance. OE.NO_EVIL directly upholds A.NO_EVIL.
A.NO_TOE_BYPASS Principals cannot gain access to resources protected by the TOE without passing through the TOE access control mechanisms.	OE.NO_TOE_BYPASS Principals cannot gain access to resources protected by the TOE without passing through the TOE access control mechanisms.	OE.NO_TOE_BYPASS, which states that Principals cannot gain access to resources protected by the TOE without passing through the TOE access control mechanisms. OE.NO_EVIL directly upholds A.NO_EVIL.
A.PHYSICAL The IT environment provides the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE.	OE.PHYSICAL Physical security will be provided within the domain for the value of the IT assets protected by the operating system and the value of the stored, processed, and transmitted information.	OE.PHYSICAL, which states that Physical security will be provided within the domain for the value of the IT assets protected by the operating system and the value of the stored, processed, and transmitted information. OE.PHYSICAL directly upholds A.PHYSICAL.
A.SCALABLE The TOE environment is appropriately scalable to provide support to the IT Systems in the organization it is deployed.	OE.SCALABLE Sites using the TOE will deploy the appropriate hardware and software environment to ensure the TOE system is scalable to provide support to the IT Systems in the organization it is deployed.	OE.SCALABLE, which states that Sites using the TOE will deploy the appropriate hardware and software environment to ensure the TOE system is scalable to provide support to the IT Systems in the organization it is deployed. OE.SCALABLE directly upholds A.SCALABLE.

Every assumption is mapped to one or more Objective in the table above. This complete mapping demonstrates that the defined security objectives uphold all defined assumptions.

8.3 Rationale for Extended Security Functional Requirements

FDP_ACF_(EXT).1 is necessary to ensure the TOE will be performing specific access decisions based on Security Attributes. While this SFR is based off of FDP_ACF.1, the original SFR would have required extensive editing beyond the scope of typical refinements. The extended SFR was then developed for the PP to explicitly document the TOE’s security feature.

FPT_TST_(EXT).1 is necessary to ensure the TOE is capable of allowing the Security Administrator the ability to verify the integrity of the TOE system configuration files and the TSF executable code. This extended SFR was added to meet the required inclusion of the SFR FPT_TST_(EXT).1 from the Protection Profile.

These requirements exhibit functionality that can be easily documented in the ADV assurance evidence and thus do not require any additional Assurance Documentation.

8.4 Rationale for Extended TOE Security Assurance Requirements

No extended Security Assurance Requirements have been defined for this Security Target.

8.5 Security Requirements Rationale

The following table and discussion provides detailed evidence of coverage for each security objective.

Table 20 – Mapping of SFRs to TOE Security Objectives

Security Objectives for the TOE Security Functional Requirements	O.ADMIN_GUIDANCE	O.AUDIT_GENERATION	O.CORRECT_TSF_OPERATION	O.DISPLAY_BANNER	O.MANAGE	O.MEDIATE	O.PARTIAL_SELF_PROTECTION	O.RESIDUAL_INFORMATION	O.TOE_ACCESS
FAU_GEN.1		✓							
FAU_GEN.2		✓							
FDP_ACC.1						✓			
FDP_ACF_(EXT).1						✓			
FDP_RIP.2								✓	
FIA_AFL.1									✓
FIA_ATD.1(1)									✓
FIA_ATD.1(2)						✓			
FIA_ATD.1(3)						✓			
FIA_SOS.1									✓
FIA_UAU.2									✓
FIA_UID.2									✓
FMT_MOF.1(1)					✓				
FMT_MOF.1(2)					✓				
FMT_MOF.1(3)					✓				

Security Objectives for the TOE / Security Functional Requirements	O.ADMIN_GUIDANCE	O.AUDIT_GENERATION	O.CORRECT_TSF_OPERATION	O.DISPLAY_BANNER	O.MANAGE	O.MEDIATE	O.PARTIAL_SELF_PROTECTION	O.RESIDUAL_INFORMATION	O.TOE_ACCESS
FMT_MSA.1(1)					✓				
FMT_MSA.1(2)						✓			
FMT_MSA.2					✓				
FMT_MSA.3					✓				
FMT_MTD.1					✓				
FMT_SMF.1					✓				
FMT_SMR.1					✓				
FPT_TST_(EXT).1			✓						
FTA_TAB.1				✓					

8.5.1 Rationale for Security Functional Requirements of the TOE Objectives

Table 21 – Objectives:SFRs Mapping

Objective	Requirements Addressing the Objective	Rationale
O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security-relevant events associated with users.	FAU_GEN.1 Audit Data Generation	FAU_GEN.1, which defines the set of events that the TOE must be capable of recording. This requirement ensures that the Security Administrator has the ability to audit any security relevant event that takes place in the TOE. This requirement also defines the information that must

Objective	Requirements Addressing the Objective	Rationale
		be contained in the audit record for each auditable event. There is a minimum of information that must be present in every audit record and this requirement defines that, as well as the additional information that must be recorded for each auditable event. This requirement also places a requirement on the level of detail that is recorded on any additional security functional requirements an ST author adds to this PP.
	FAU_GEN.2 User Identity Association	FAU_GEN.2, which ensures that the audit records associate a user identity with the auditable event. In the case of authorized users, the association is accomplished with the “userid”. When TOE components imitate actions that need to be audited, the TOE will ensure a mechanism is in place to identity the component as the entity conducting the action.
O.CORRECT_TSF_OPERATION The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site.	FPT_TST_(EXT)1.1 TSF Testing	FPT_TST_(EXT)1.1 and FPT_TST_(EXT)2.1, which ensure the correctness of the TSF configuration files, data and executable code. If TSF software is corrupted it is possible that the TSF would no longer be able to enforce the security policies. This also holds true for TSF data, if TSF data is corrupt the TOE may not correctly enforce its security policies. The FPT_TST_(EXT)1 functional requirement includes the critical nature and specific handling of the cryptographic related TSF data. Since the cryptographic TSF data has specific FIPS PUB requirements associated with them it is important to ensure that any fielded testing on the integrity of these data maintains the same level of scrutiny as specified in the FCS functional requirements.
O.DISPLAY_BANNER The TOE will display an advisory warning regarding use of the TOE to the administrative users.	FTA_TAB.1 Default TOE Access Banners	FTA_TAB.1, which meets this objective by requiring the TOE display a Security Administrator defined banner before an administrator can establish an authenticated remote session. This banner is under complete control of the Security Administrator in which they specify any warnings regarding unauthorized

Objective	Requirements Addressing the Objective	Rationale
		use of the TOE and remove any product or version information if they desire.
O.MANAGE The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	FMT_MOF.1(1) Management of Security Functions Behavior (Access Policy)	FMT_MOF.1(1) and FMT_MOF.1(2), which provide Security Administrators the ability to manage the TOE's access policy settings and the list of applications authorized to query the TOE.
	FMT_MOF.1(2) Management of Security Functions Behavior (Authorized Applications)	FMT_MOF.1(1) and FMT_MOF.1(2), which provide Security Administrators the ability to manage the TOE's access policy settings and the list of applications authorized to query the TOE.
	FMT_MOF.1(3) Management of Security Functions Behavior (Audit)	FMT_MOF.1(3), which provides the Audit Administrator the ability to manage the audit settings.
	FMT_MSA.1(1) Management of Security Attributes - Attribute Management	FMT_MSA.1(1), which provides the Security Administrator with the capability to manage the security attributes of both principals and protected resources.
	FMT_MSA.2 Secure Security Attributes	FMT_MSA.2 ensures that only specific secure values are accepted for security attributes. This requirement is designed meet the ID requirement to prevent user authentication password reuse. A history of static authenticator changes will be maintained with assurance of non-replication of individual authenticators. When a user changing their password submits a previously used password, the system will consider that an "insecure" value for that security attribute and reject it.
	FMT_MSA.3 Static Attribute Initialization	FMT_MSA.3 requires that by default, the TOE does not allow an access to a protected resource until an access policy rule allows it.
	FMT_MTD.1	FMT_MTD.1 is used by the Security Administrator to manage TSF data

Objective	Requirements Addressing the Objective	Rationale
	Management of TSF Data	and configuration.
	FMT_SMF.1 Specification of Management Functions	FMT_SMF.1 requires that the TSF shall be capable of performing specified security management functions.
	FMT_SMR.1 Security Management Roles	FMT_SMR.1 requires that roles exist for administrative actions: the Security Administrator, who is responsible for configuring the TOE's security policies, including the management of the security data that is critical to the cryptographic operations; the Audit Administrator, who is restricted to reading and deleting the audit trail; and Authorized Applications which are permitted to query the TOE. The TSF is able to associate a human user with one or more roles.
O.MEDIATE The TOE must protect user data in accordance with its security policy.	FDP_ACC.1 Access Control Policy	FDP_ACC.1 defines that an Authorization Server Access Control policy will be enforced on principals attempting to gain access to a list of named objects. All the operations among subject and object covered are by the Authorization Server policy. The "subjects" are generally the principals. The "named objects" are the designated web based resources (web server, directories, files, or objects) that the Authorization Server is protecting.
	FDP_ACF_(EXT).1 Access Control Functions	FDP_ACF_(EXT).1 defines the Security Attribute used to provide Access Control to objects based on the following Authorization Server Access Control policy.
	FIA_ATD.1(2) User Attribute Definition - Principal	FIA_ATD.1(2) and FIA_ATD.1(3) define the Security Attributes associated with the principals and authorized applications.
	FIA_ATD.1(3) User Attribute Definition - Authorized Application	FIA_ATD.1(2) and FIA_ATD.1(3) define the Security Attributes associated with the principals and authorized applications.

Objective	Requirements Addressing the Objective	Rationale
	FMT_MSA.1(2) Management of Security Attributes - Attribute Authority	FMT_MSA.1(2) restricts disclose of user security attributes to authorized applications.
O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.	FDP_RIP.2 Full Residual Information Protection	FDP_RIP.2 is used to ensure the contents of resources are not available to subjects other than those explicitly granted access to the data. For this TOE it is critical that the memory used to make authorization decisions is either cleared or that some buffer management scheme be employed to prevent the authorization decision of one user's request to be used in a subsequent authorization decision.
O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE.	FIA_AFL.1 Authentication Failure Handling	FIA_AFL.1 provides a detection mechanism for unsuccessful authentication attempts by remote administrators. The requirement enables a Security Administrator settable threshold that prevents unauthorized users from gaining access to authorized administrator's account by guessing authentication data by locking the targeted account until the Security Administrator takes some action (e.g., re-enables the account) or for some Security Administrator defined time period. Thus, limiting an unauthorized user's ability to gain unauthorized access to the TOE.
	FIA_ATD.1(1) User Attribute Definition - Administrator	FIA_ATD.1 defines the attributes for administrators, principals, and authorized applications that shall be used to determine identity and enforce what type of access each entity has to the TOE or to another protected resource based on the access control policy.
	FIA_SOS.1 Verification of Secrets	FIA_SOS.1.1 ensures that a mechanism is in place to verify that user's passwords must contain a minimum of 8 alphanumeric charters with at least one numeric charter. This type of password cannot be easily be broken with a dictionary search or elementary password cracking

Objective	Requirements Addressing the Objective	Rationale
		software.
	FIA_UAU.2 Timing of Authentication	FIA_UAU.2 contributes to this objective by preventing services from being provided by the TOE to unauthenticated users.
	FIA_UID.2 Timing of Identification	FIA_UID.2 contributes to this objective by preventing services from being provided by the TOE to unidentified users.

8.5.2 Security Assurance Requirements Rationale

EAL3+ was chosen to provide a moderate level of assurance that is consistent with good commercial practices. As such, additional tasks are placed upon the vendor assuming the vendor follows reasonable software engineering practices and can provide support to the evaluation for design and testing efforts. The chosen assurance level is appropriate with the threats defined for the environment. While the System may monitor a hostile environment, it is expected to be in a non-hostile position and embedded in or protected by other products designed to address threats that correspond with the intended environment. At EAL3+, the System will have incurred a search for obvious flaws to support its introduction into the non-hostile environment.

The augmentation of ALC_FLR.2 was chosen to give greater assurance of the developer's on-going flaw remediation processes.

8.5.3 Dependency Rationale

This ST does satisfy all the requirement dependencies of the Common Criteria. Table 22 lists each requirement to which the TOE claims conformance with a dependency and indicates whether the dependent requirement was included. As the table indicates, all dependencies have been met.

Table 22 – Functional Requirements Dependencies

SFR ID	Dependencies	Dependency Met	Rationale
FAU_GEN.1	FPT_STM.1	✓	The dependency is met through the IT environment, via the CAPP. The Operating System on which the TOE is installed provides reliable time stamps for the TOE's use.
FAU_GEN.2	FAU_GEN.1	✓	

SFR ID	Dependencies	Dependency Met	Rationale
	FIA_UID.1	✓	FAU_GEN.2 meets its dependency FIA_UID.1 through the inclusion of FIA_UID.2. Since FIA_UID.2 is hierarchical to FIA_UID.1, FAU_GEN.2 successfully meets its dependency.
FDP_ACC.1	FDP_ACF.1	✓	FDP_ACC.1 meets its dependency FDP_ACF.1 through the inclusion of FDP_ACF_(EXT).1. FDP_ACF_(EXT).1 is based off of the SFR FDP_ACF.1 but has been modified to be better describe the functionality of the TOE. Therefore, FDP_ACC.1 successfully meets this dependency.
FDP_ACF_(EXT).1	FDP_ACC.1	✓	
	FMT_MSA.3	✓	
FDP_RIP.2	None		
FIA_AFL.1	FIA_UAU.1	✓	FIA_AFL.1 meets its dependency FIA_UAU.1 through the inclusion of FIA_UAU.2. Since FIA_UAU.2 is hierarchical to FIA_UAU.1, FIA_AFL.2 successfully meets its dependency.
FIA_ATD.1(1)	None		
FIA_ATD.1(2)	None		
FIA_ATD.1(3)	None		
FIA_SOS.1	None		
FIA_UAU.2	FIA_UID.1	✓	FAU_UAU.2 meets its dependency FIA_UID.1 through the inclusion of FIA_UID.2. Since FIA_UID.2 is hierarchical to FIA_UID.1, FIA_UAU.2 successfully meets its dependency.
FIA_UID.2	None		
FMT_MOF.1(1)	FMT_SMF.1	✓	
	FMT_SMR.1	✓	

SFR ID	Dependencies	Dependency Met	Rationale
FMT_MOF.1(2)	FMT_SMF.1	✓	
	FMT_SMR.1	✓	
FMT_MOF.1(3)	FMT_SMF.1	✓	
	FMT_SMR.1	✓	
FMT_MSA.1(1)	FDP_ACC.1	✓	
	FMT_SMF.1	✓	
	FMT_SMR.1	✓	
FMT_MSA.1(2)	FDP_ACC.1	✓	
	FMT_SMF.1	✓	
	FMT_SMR.1	✓	
FMT_MSA.2	FDP_ACC.1	✓	
	FMT_SMR.1	✓	
	FMT_MSA.1	✓	FMT_MSA.2 meets its dependency of FMT_MSA.1 through the inclusion of FMT_MSA.1(1) and FMT_MSA.1(2). The original SFR has been refined and iterated into two separate SFRs. FMT_MSA.1(1) and FMT_MSA.1(2) combine to deliver the same functionality as FMT_MSA.1. Therefore, the dependency has been successfully met.
FMT_MSA.3	FMT_MSA.1	✓	FMT_MSA.3 meets its dependency of FMT_MSA.1 through the inclusion of FMT_MSA.1(1) and FMT_MSA.1(2). The original SFR has been refined and iterated into two separate SFRs. FMT_MSA.1(1) and FMT_MSA.1(2) combine to deliver the same functionality as FMT_MSA.1. Therefore, the dependency has been successfully met.

SFR ID	Dependencies	Dependency Met	Rationale
	FMT_SMR.1	✓	
FMT_MTD.1	FMT_SMF.1	✓	
	FMT_SMR.1	✓	
FMT_SMF.1	None		
FMT_SMR.1	FIA_UID.1	✓	FMT_SMR.1 meets its dependency FIA_UID.1 through the inclusion of FIA_UID.2. Since FIA_UID.2 is hierarchical to FIA_UID.1, FMT_SMR.1 successfully meets its dependency.
FTA_TAB.1	None		
FPT_TST_(EXT)1.1	None		

9 Terminology and Acronyms

9.1.1 Terminology

Terms	Definition
Authorization Server Access Control Policy	The Authorization Server Access Control Policy is referred to in the TOE's documentation as its "Security Policy".
Objects	Protected Resources
Principals	Users or Applications

9.1.2 Acronyms

Acronym	Definition
API	Application Programming Interface
CAPP	Controlled Access Protection Profile
CC	The Common Criteria for Information Technology Security Evaluation
CEM	Common Evaluation Methodology
CM	Configuration Management
DAL	Data Abstraction Layer
DCOM	Distributed Component Object Model
DMZ	Demilitarized Zone
EAL	Evaluation Assurance Level
FIPS	Federal Information Processing Standard
FTP	File Transfer Protocol
HP	Hewlett-Packard
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IT	Information Technology

Acronym	Definition
IWA	Integrated Windows Authentication
JDK	Java Development Kit
JRE	Java Runtime Environment
JSP	Java Server Page
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
MB	Megabyte
MHz	Megahertz
NIST	National Institute of Standards and Technology
NMS	Network Management System
NTLM	NT LAN Manager
NTP	Network Time Protocol
OS	Operating System
OSP	Organizational Security Policy
PDC	Primary Domain Controller
PP	Protection Profile
SDK	Software Development Kit
SFR	Security Functional Requirement
SNMP	Simple Network Management Protocol
SPS	Secure Proxy Server
SQL	Structured Query Language
SSO	Single Sign On

Acronym	Definition
ST	Security Target
TOE	Target of Evaluation
TLS	Transport Layer Security
TSF	TOE Security Functionality
TSFI	TOE Security Functional Interface
TSP	TOE Security Policy
URL	Uniform Resource Locator
WAX	Web Agent Extension