CA Siteminder Web Access Manager R12 SP1-CR3 Security Target

Version 0.8
May 29, 2009

Prepared for:
CA
100 Staples Drive
Framingham, MA 01702

Prepared by:
Booz Allen Hamilton
Common Criteria Testing Laboratory
900 Elkridge Landing Road, Suite 100
Linthicum, MD 21090-2950

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Security Target Introduction

This chapter presents the Security Target (ST) identification information and an overview. An ST contains the Information Technology (IT) security requirements of an identified Target of Evaluation (TOE) and specifies the functional and assurance security measures offered by the TOE.

1.1 ST Reference

This section provides information needed to identify and control this ST and its Target of Evaluation. This ST targets Evaluation Assurance Level 3 (EAL3).

1.1.1 ST Identification

ST Title:	CA SiteMinder ® Web Access Manager r12 SP1-CR3 Security Target
ST Version:	0.8
ST Publication Date:	May 29, 2009
ST Author:	Booz Allen Hamilton

1.1.2 Document Organization

Chapter 1 of this ST provides identifying information for the CA SiteMinder r12 SP1. It includes an ST Introduction, ST Reference, ST Identification, TOE Reference, TOE Overview, and TOE Type. Chapter 2 discusses the TOE Description, which consists of the physical and logical boundaries. Chapter 3 describes the conformance claims made by this ST. Chapter 4 describes the Security Problem Definition as it relates to threats, Operational Security Policies, and Assumptions met by the TOE. Chapter 5 identifies the Security Objectives of the TOE and of the operational environment. Chapter 6 describes the Extended Security Functional Requirements. Chapter 7 describes the Security Functional Requirements (SFRs). Chapter 8 describes the Security Assurance Requirements (SARs). Chapter 9 is the TOE Summary Specification (TSS), a description of the functions provided by the CA SiteMinder r12 SP1 to satisfy the security functional and assurance requirements. Chapter 10 provides a rationale, or pointers to a rationale, for security objectives, assumptions, threats, requirements, dependencies, and PP claims.

1.1.3 Terminology

Term	Definition
Administrator	A trusted user who has privileges to administer the TOE. The privileges and activities of an administrator account vary by administrative scope and tasks. Types of administrators include domain, system, and super user admins.
Administrator	A policy that sets restrictions on an administrator's ability to manage

Term	Definition
Policy	the TOE.
Agent	An Agent is installed on Web servers, or application servers to secure access to resources.
Agent Group	An Agent group is a Policy Server object that points to a group of Agents. The Agents in the group can be installed on different servers, but all of the Agents protect the same resources. Typically Agent groups are configured in SiteMinder for groups of servers that distribute the workload for access to a popular set of resources.
Agent Configuration Object	An Agent Configuration Object holds configuration parameters for one or more Web Agents.
Agent Key	Used by Web Agents to encrypt cookies.
Authentication Level	Each authentication method is associated with a particular level, ranging from a top priority of 1 to a lowest priority of 1000. End users that authenticate with a low level must re-authenticate when trying to access a resource with a higher authentication level.
Authentication Scheme	An authentication scheme is a Policy Server object that determines the credentials an end user will need to access a protected resource. Authentication schemes are assigned to realms. When an end user tries to access a resource in a realm, the authentication scheme of the realm determines the credentials that an end user must supply in order to access the resource.
Authorized	An administrator or workstation end user that has been identified and authenticated by the TOE.
Cluster	A set of Policy Servers that are grouped to improve system availability and response time by dynamically balancing load among the servers in the cluster and failing over to other clusters based on customer-defined failover thresholds. Clusters are typically grouped by data centers located in different geographic locations.
Cluster failover	Switching to another cluster when the number of servers available in a cluster falls below a configurable threshold. The priority of the clusters is defined in the Host Configuration Object. Requests will fail-back to a higher-priority cluster as soon as the threshold requirement for that cluster is met.
End User	An authorized user of the TOE without privileges who tries to gain access to a protected resource.
Get/Put/Post	An HTTP operation known as an end user's request. It is received by the Web Agent and forwarded to the Policy Server.
Global Objects	Objects that apply to all resources (global rules, global responses, global policies).
Global Rules	A global rule is a Policy Server object that specifies a filter used to apply a global policy to a large group of resources.

Term	Definition
Global Responses	A global response is a Policy Server object that determines a reaction to a global rule. Global responses are included in global policies, and take place when a global rule is triggered.
Global Policies	A global policy is a Policy Server object that binds end users, global rules, global responses, and optionally, time restrictions and IP address restrictions together.
Groups	A group (agent group, rule group, response group) can contain individual items or groups of its own type. For example, a rule group can contain rules and/or groups of rules.
Host Configuration Objects	A Host Configuration Object holds configuration parameters for the Trusted host.
Key Store	Entity used by the SiteMinder Policy Server to store encryption keys used by the Policy Server when communicating with SiteMinder Web Agents.
Nested Groups	Groups that contain other groups. Also known as sub-groups. When nested groups are allowed, each user group and each sub-group is searched when the policy is processed. When they are not allowed, each user group is searched, but sub-groups are not searched, when the policy is processed.
Nested Realms	Realms created within other realms to better represent the grouping of resources in a corporate network. Deeper levels of nested realms typically correspond to heightened security requirements in a directory tree. An administrator can achieve this by assigning a stronger authentication scheme with a higher protection level to the nested realm.
Policy	A policy is a Policy Server object that binds users, rules, responses, and optionally, time restrictions and IP address restrictions together. Policies establish entitlements for a SiteMinder protected entity. When a user attempts to access a resource, the policy is what SiteMinder ultimately uses to resolve the request.
Policy Domains	A policy domain is a logical grouping of one or more user directories, administrators, and realms. This Policy Server object is the basis for entitlement data. By creating policy domains, an administrator creates a container for entitlements that surround a particular groups of resources (realm), as well as the end users who may access the resources, and the administrator who sets up entitlements.
Policy Domain Object	A SiteMinder object within a domain (policy, realm, response, response groups, response attributes, rules and rule groups, rule policies).
Policy Server	CA SiteMinder software component that provides a platform for managed key operations, authentication, authorization, and security management.

Term	Definition
Policy Server Object	An object that the Policy Server uses (System objects, Policy Domain objects, Global objects).
Policy Store	Collection of CA SiteMinder Policy Server objects. Policy stores can reside in an ODBC (see page 19)-enabled database or an LDAP (see page 17) directory.
Policy Store Key	A key used to encrypt data that is sent between the Policy Server and the Policy Store. The key can be from 6 to 24 characters in length. All Policy Servers that share a SiteMinder Policy Store (a database containing policy information) must be configured using the same Policy Store Key.
Protected Resource	Any URL under SiteMinder protection.
Protection Level	A number between 0 and 1000 that is given to authentication schemes. A higher number indicates a higher level of protection.
Realm	A realm is a Policy Server object that identifies a group of resources. Realms typically define a directory or folder and possibly its subdirectories.
Realm Resource Filter	A string, such as a relative path to a directory that specifies the resources covered by the realm. If the realm is a top-level realm, specify the resources relative to the server that serves up the files or application. If the realm is nested, specify the resources relative to the parent realm.
Remote Server	The workstation used by the end user to gain access to the TOE.
Resource	Any URL to which an end user attempts to gain access.
Response	A response is a Policy Server object that determines a reaction to a rule. Responses are included in policies, and take place when a rule is triggered.
Response Groups	A response group is a Policy Server object that contains a logical grouping of responses. Response groups are most often used when many responses will be included in a policy.
Rules	A rule is a Policy Server object that identifies a resource and the actions that will be allowed or denied for the resource. Rules can also include actions associated with specific events, such as what to do if a end user fails to authenticate correctly when asked for their credentials.
Rule Groups	A rule group is a Policy Server object that contains multiple rules. Rule groups are used to tie together different rules that will be used in a single policy.
Rule Resource	A string or regular expression that specifies the resources to which the rule applies. Specify the resources relative to the realm containing the resource.
Scope	Indicates whether the administrator's privileges extend to all domains and applications or to only specific domains and applications. Included in the Administrator Policy.

Term	Definition
Session Key	The Policy Server creates Session Keys using AES with HMAC-SHA256. The Session Keys are utilized by the Policy Server and SiteMinder agents for protecting the TCP/IP message exchange between these components.
Session Ticket	Also known as session specification. Session tickets contain credentials and other information relating to an end user's session.
Session Ticket Key	The Policy Server utilizes the Session Ticket Key to encrypt Password Services data in the user stores, and sensitive data (keys, shared secrets, passwords) in the Policy Stores.
Super User Administrator	The default administrator account with full privileges that is set up during installation of the TOE. There are two Super User accounts created during the installation of the TOE. A local Super User account which is used during installation and configuration and a remote Super User account which is used in the evaluated configuration of the TOE. The remote Super User is the administrator that can access the TOE via the WAM Administrative UI.
System Objects	Objects used throughout a SiteMinder deployment (agents, agent groups, agent configuration objects, host configuration objects, user directories, policy domains, administrators, authentication schemes, registration schemes, agent types, password policies, trusted hosts).
Target network	The domain of workstations that have the TOE installed on them.
Task	Determines the privileges an administrator is allowed to perform within their scope. Included in the Administrator Policy.
Time Restrictions	A time restriction indicates when a rule fires. For example, if an administrator creates an Allow Access rule with a time restriction that limits access to a resource to 9am - 5 pm, Monday - Friday, the rule will only fire and allow end users to access the resource during the specified time. The resource will not be available outside the times indicated.
Trusted Hosts	A Trusted Host object represents the client component that connects to the Policy Server.
User	Defined as an administrator, domain administrator, system administrator, or end user of the TOE.
User Authorization Cache (memory)	A configurable cache inside the Policy Server that stores user information after the login step.
User Directories/ User Store	A user directory in SiteMinder is an object that contains details for connecting to an existing user directory that resides outside of SiteMinder. This allows an administrator to configure a simple connection to an existing user directory, instead of replicating end user information within SiteMinder. The username space is an LDAP directory server.

Table 1-1 Customer Specific Terminology

Term	Definition
Authorized user	A user who may, in accordance with the TSP, perform an operation. This is an end user or an administrator.
External IT entity	Any IT product or system, untrusted or trusted, outside of the TOE that interacts with the TOE.
Role	A predefined set of rules establishing the allowed interactions between an end user and the TOE.
TOE Security Functions (TSF)	A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.
User	Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.

Table 1-2 CC Specific Terminology

1.1.4 Acronyms

Acronym	Definition
ACM	Configuration Management
ADO	Delivery and Operation
ADV	Development
AGD	Guidance Documents
ALC	Life cycle support
ATE	Tests
AVA	Vulnerability assessment
CC	Common Criteria [for IT Security Evaluation]
EAL	Evaluation Assurance Level
EIS	Enterprise Information Systems
FAU	Security Audit
FCO	Communication
FCS	Cryptographic Support
FDP	User Data Protection
FIA	Identification and Authentication
FMT	Security Management
FPT	Protection of the TSF
FTA	TOE Access
FTP	Trusted Channels/Path
GUI	Graphical User Interface
ICMP	Internet Control Message Protocol
ID	Identifier
IP	Internet Protocol
IPX	Internetwork Packet Exchange
IT	Information Technology

Acronym	Definition
MAU	Media Access Unit
MIB	Management Information Base
NTLM	Windows NT LAN Manager
SF	Security Function
SFP	Security Function Policy
ST	Security Target
TCP	Transmission Control Protocol
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSFI	TOE Security Functions Interface
TSP	TOE Security Policy
UDP	User Datagram Protocol

Table 1-3 Acronyms

1.1.5 References

Reference Title	ID
Common Criteria for Information Technology Security Evaluation, CCMB-2007-09-004, Version 3.1 Revision 2, September 2007.	[CC]
CA™ SiteMinder Web Access Manager Web Agent Configuration Guide r12	[Web Agent]
CA™ SiteMinder Web Access Manager Policy Server Administration Guide r12	[Policy Server]
auth2_SiteMinder 6[1].0 Platform Support.112307.pdf	[Platform]
SiteMinder_r6_tech_whitepaper.pdf	[Whitepaper]

Table 1-4 References

1.1.6 CC Concepts

The following are CC concepts as used in this document. A Subject is any user of the TOE (administrator, end user, system, domain, or super user). An Object can either be a System Object (agent, policy domain, authentication scheme, etc), a Policy Domain Object (realm, rule, response, etc), or a Global Object (global rule, global response, global policy, etc). An Operation is any action on a resource (e.g. Get, Put, Post), or any policy server operation (authentication, authorization, administration, accounting). A Security Attribute is information such as a username and password that is kept in the user store. A Session is the timeframe from an initial end user's successful logon through that same end user's logout. A Resource is any URL that a user attempts to access. Information is any data used by the TOE. An External Entity is anything outside of the TOE that affects the TOE.

1.2 TOE Reference

1.2.1 TOE Identification

CA SiteMinder ® Web Access Manager r12 SP1-CR3

1.2.2 TOE Overview

The TOE allows administrators to control end user access to protected resources such as web server content and application server content. End user access is controlled via Web Agents.
The TOE:

- Provides a platform for access control to protected web resources;
- Integrates with and operates as a component of the web server to securely store authorization and authentication processes on the web server;
- Records end user actions on the web resources the TOE protects

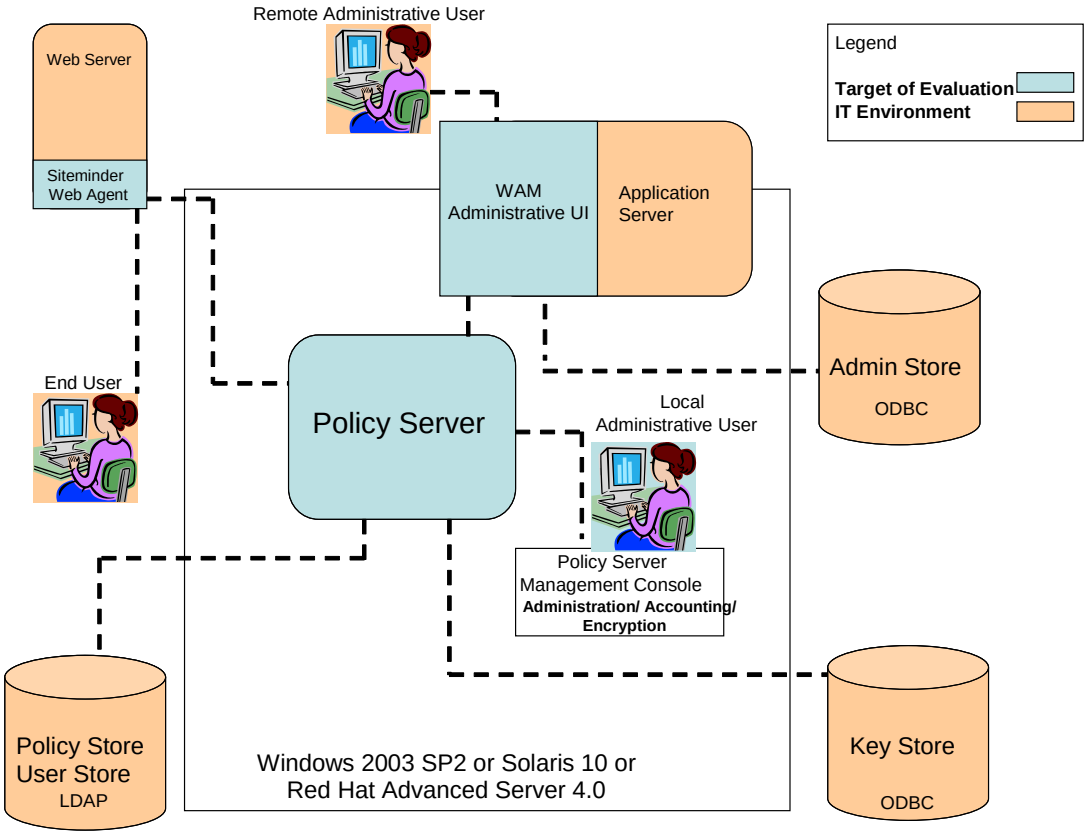

Figure 1-1 TOE Boundary

As illustrated in Figure 1-1, there are remote administrators and local administrators of the TOE. Remote administrators authenticate through the WAM Administrative UI and local administrators administer through the Policy Server Management Console.

Remote administrators have to identify and authenticate themselves before being allowed access to the Policy Server. Once authenticated, the administrators are only allowed to manage policies according to the task and scope of the domain(s) to which they are assigned. Authentication schemes for end users are configured using the WAM Administrative UI, which also authenticates administrator accounts using the Administrator Store. During authentication, the Web Agents communicate with the Policy Server to determine the proper credentials that must be retrieved from an end user who is requesting resources.

When an end user attempts to access a protected network resource, the Policy Server uses the authentication scheme associated with the resource's realm and protection level to determine how to identify the user. The end user is required to re-authenticate when attempting to access a resource with a higher protection level than the one he is currently accessing. A user in SiteMinder is the Super User, an administrator, a domain administrator, a system administrator, or an end user. An end user in SiteMinder is an authorized user without administrative privileges who tries to gain access to a protected resource. The authentication scheme specifies the credentials that the end user must supply for authentication, as well as the method used by the Policy Server to validate the end user's identity. If an end user tries unsuccessfully to authenticate after a number of attempts determined by the administrator, his account will be locked out. If an account is locked out, the TOE invokes a delay or timeframe which prevents the end user from attempting to gain access until the timeframe is exceeded or a system administrator resets the account. Components of the TOE interact to enforce access control. The Web Agent intercepts end user requests for resources and checks with the Policy Server to see if the requested resource is protected. If the resource is unprotected, the access request proceeds directly to the web server. If the resource is protected, the following series of events occur:

1. The Policy Server checks which authentication method is required for this resource. Typical credentials are a username and password, but other credentials, such as a certificate, may be required.

2. The Web Agent challenges the end user for credentials. The user responds with the appropriate credentials.

3. The Web Agent passes the credentials to the Policy Server, which determines if the credentials are correct.

4. If the end user passes the authentication phase, the Policy Server determines if the end user is authorized to access the resource. Once the Policy Server grants access, the Web Agent allows the request to proceed to the web server.

The TOE manages and enforces access control rules established by administrators. These rules define the operations that are allowed for each protected resource and include the end user's identity (username) of those that are allowed to perform each operation. If an end user is not included in a rule that allows access to a protected resource (either by username, rule, or group), he is denied access to that resource. On the other hand, an administrator cannot be explicitly denied access to a WAM Administrative UI webpage. The TOE tracks and logs successful authorizations utilizing the information stored in the user session cache, allowing the administrator to track end user activity and measure how

often applications on the Web site are used. The Web Agent sends a message to the Policy Server each time an end user is authorized from cache to access resources.

The TOE maintains default figures that can be overwritten by the Administrator. The administration service of the TOE is what enables the User Interface (UI) to record configuration information. The Policy Store is the database that contains entitlement information. There are two Policy Administration/Management interfaces. The Policy Server Management Console is used by the local TOE administrator to gain access to the Policy Server and is used for Polcy Server configuration and system management operations during initial setup and configuration of the TOE, this interface is non-SFR interfering in the evaluated configuration. The WAM administrative User Interface (WAM Admin UI) is used by the remote TOE administrator to gain access. This interface is used to manage Policy Server objects, which are the objects located in the different Stores..

The TOE generates log files that contain auditing information about the events that occur within the system, including the startup and shutdown of audit functions. For each of the audit events, the corresponding log includes the date and time of the event, the type of event, subject identity, and the outcome of the event. Based on the content of these logs, the TOE is able to associate the event with the end user or administrator that caused the event. These logs can be viewed through the local OS, so that security events or anomalies can be analyzed.

The Policy Server and Web Agent audit end user activity. The Web Agent sends a message to the accounting service each time an end user is authorized from cache to access resources. This action ensures that the accounting service is tracking successful authorizations for the Web Agent and the Policy Server. If an audit message is not successfully sent by the Web Agent to the accounting service for an authorization, access to the resource is denied.

The TOE uses encryption keys to encrypt and decrypt sensitive data passed between TOE components. Agent Keys are used to encrypt TOE cookies that may be read by all agents in a single sign-on environment, and are shared by all agents in a single sign-on environment, since each agent must be able to decrypt cookies encrypted by the other agents. Agent Keys are managed by the TOE, and distributed to Web Agents periodically. Session Ticket Keys are used by the TOE to encrypt Session Tickets. Session Tickets contain the end user's distinguished name and other information relating to a session (including the authentication method). Web Agents embed Session Tickets in TOE cookies, but cannot access the contents since they do not have access to Session Ticket Keys which never leave the Policy Server. Both types of keys are kept in the Policy Server's Key Store. The Policy Server performs key rollovers, which include the generation and encryption of new keys, in order to ensure the security of the keys.

Dynamic software-based load-balancing provides a high level of system availability and improves response time by distributing requests from SiteMinder Agents based on the computing power of the Policy Servers in a cluster. Cluster-to-cluster failover based on configurable failover thresholds further enhances the level of system availability and system response time. The following figure illustrates a deployment using two clusters:

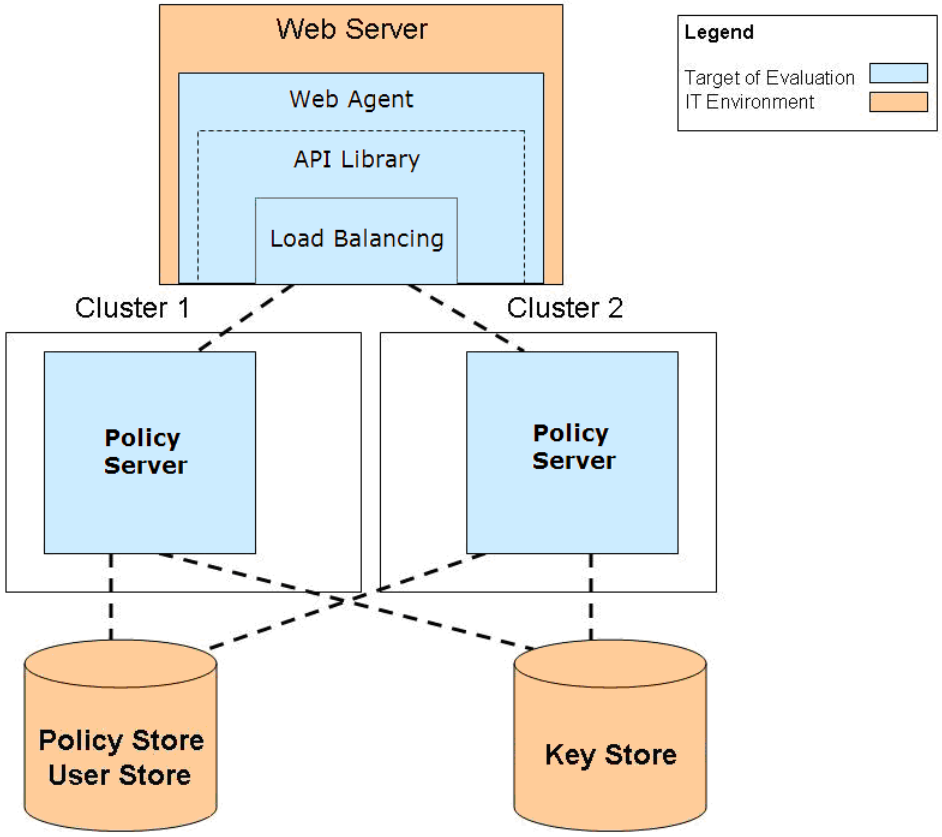

Figure 1-2 Policy Server Clusters

The diagram above shows two Policy Servers 1 and 2 with the corresponding Web Agent. For each Web Agent there is a primary and a secondary cluster. The load-balancer switches to the secondary cluster when the primary cluster fails. The load-balancer is part of the Web Agent API library (Web Agent API Layer).

A threshold parameter is used to speedup the cluster-to-cluster failover. The higher the threshold value is the less time it takes for a failover to occur. The reasoning behind having a higher threshold value is to react faster to the cluster overload and subsequently to the cluster failure.

1.3 Components of the TOE and Operational Environment (in evaluation)

CA SiteMinder r12 SP1 is made of three components; a Policy Server, a Web Agent, and the WAM Administrative UI. The Policy server allows an authorized administrator to configure policies that govern the level of access a particular end user can gain to protected resources. The Web Agent resides on the Web Server and is used to manage end user sessions and enforce the policies defined by the Policy Server. The WAM

Administrative UI is used for remote administration, which includes end user management and resource policy definition.

1.3.1 Policy Server

The Policy Server provides access control and single sign-on. It allows an authorized administrator to configure policies via the Siteminder WAM Administrative UI. It also allows administrators to override the default values provided for the TOE. The policies govern the level of access to a resource granted to an end user. It typically runs on a separate Windows or UNIX system and offers the following security operations:

- Authentication — Authenticates end users via a range of authentication methods including usernames and passwords, and public key certificates.
- Authorization — Manages and enforces access control rules established by administrators.
- Administration — Enables the Siteminder WAM Administrative UI to record configuration information in the Policy Store.
- Accounting service — Generates log files that contain auditing information.
- Trusted path — Uses an encrypted tunnel and encrypted keys to pass sensitive data between separate parts of the TOE.

1.3.2 Database Stores

The Policy Store, User Store, Administrator Store, and Key Store are database stores which are configured during the installation and configuration of SiteMinder.
The following table lists the TOE database types for the evaluated configuration:

Database	Database Description	Available Storage
Policy Store	The database that stores all categories (system, policy domain and global) of Policy Server objects, including rules and realms.	LDAP
User Store	The database that stores user data, including organizational information, user and group attributes, and credentials such as passwords.	LDAP
Administrator Store	The database that stores administrator data, including their login credentials.	ODBC
Key Store	The database that contains keys used to encrypt cookies created by the TOE Agent.	ODBC

Table 1-5 TOE Database Types and Storage Options

LDAP/ODBC drivers are used to access the various databases used by the TOE (i.e., User Store, Policy Store, Administrator Store, and Key Store). The TOE uses the database administrator account of the respective database for read, write, and execution privileges.

1.3.2.1 External User Directories

The SiteMinder User Store can be configured to pull information from a newly configured database or pre-existing external directory, The external directory will store the end user data, including organizational information, user and group attributes, and credentials such as passwords. The Policy Server uses the connection to this external directory to verify user identities and retrieve user attributes contained in the external directory. The connection to this external directory is configured through the WAM Administrative UI.

Instead of replicating the external user information within SiteMinder, SiteMinder pulls the information from the external directory into the Policy Server, which uses the information to authenticate the end user. This information is stored in the Policy Server's authorization cache (memory), which is a configurable cache that stores user information after the login step.
The Policy Server can be configured to connect to any number of supported user directories, including:

- LDAP
- ODBC
- Oracle
- Windows NT
- Custom

In the evaluated configuration, SiteMinder is configured to connect to an external LDAP user directory, and it will be called the User Store.

1.3.3 User Authorization Cache

The Policy Server maintains the User Authorization Cache. The User Authorization Cache stores end user distinguished names (DNs) based on the end user portion of policies and includes the end users' group membership. The User Authorization Cache is configured via the Policy Server Management Console and is shared amongst the Web Agent and the Policy Server. User authorization cache is included in the evaluated configuration.

1.3.4 **Policy Server Management Console** and Administrative UI

Most fundamental system configuration tasks are performed using the Policy Server Management Console, which are needed to place the TOE in the evaluated configuration, while tasks for establishing policies for end users and resources are performed using the WAM Administrative UI.

The Policy Server Management Console (or Management Console) provides a range of Policy Server configuration and system management options. Policy Server management tasks include but are not limited to access configuration, audit log configuration, and encryption setup. The Management Console has a tab-based user interface in which information and controls are grouped together by function and presented together on tabs in a single window. The Policy Server Management Console can only be run by an administrator as defined in the Operational Environment.
The Policy Server Management Console contains the following tabs:

- Status tab - Allows an administrator to View the status of and start and stop the Policy Server.
- Settings tab - Configure TCP and UDP port settings for Policy Server administration and connection and thread settings for performance.
- Data tab - Configure policy store, key store, and session store databases and audit log and token data locations.
- Super user tab - Change the password of the local Super user.
- Keys tab - Configure key management policy.
- Logs tab - Configure policy server and audit logging.
- Profiler tab - Enable and configure output for the profiler, which can be used for debugging Policy Server issues.
- Advanced tab - Adjust the settings for the Policy Server administration journal and optional Event Handler libraries.

The WAM Administrative UI lets administrators view, modify, and delete Policy Server objects. WAM Admin UI component provides a web accessable GUI (also known as the WEB Admin GUI) which allows an adminstrator to manage the TOE. Although the details of each task differ by object, the general methods are similar. For example, the procedure for deleting an Agent is similar to the procedure for deleting a response. Policy Server objects include but are not limited to end users, policies, rules, realms and agents. The ability for administrators to perform actions on Policy Server objects is what allows them to define what resources are protected by the TOE from end user access.

Note: The WAM ADMIN UI can be installed (1) on the Policy Server or on (2) a separate system. In the latter case, it communicates to the policy server in the same manner as a WEB Agemt.

1.3.5 Web Agents

A SiteMinder Web Agent is a software component that controls end user access to a protected resource (any URL protected by the TOE). The Web Agent grants or denies access by enforcing policies defined through the Policy Server. Web Agents work with the Policy Server to authenticate and authorize end users for access to web server resources. The Web Agent enables Web applications to personalize content. The network path between the Web Agent and the Policy Server is secured by AES encryption over a standard TCP/IP connection. The Web Agent is integrated with a Web

server. The Web Agent intercepts requests for a resource and determines whether or not the resource is protected by the TOE.

Web Agents perform the following tasks:

- Intercept access requests for protected resources and work with the Policy Server to determine whether or not an end user should have access.
- Provide information to a Web application that dictates how content is presented to the end user (policy-based personalization) and how to deliver access privileges.
- Ensure an end user's ability to securely access information. Web Agents store contextual information about end user access privileges in a session cache. Performance can be optimized by modifying the cache settings.
- Enable single sign-on across web servers in a single cookie domain or across multiple cookie domains without requiring end users to re-authenticate.

In the evaluated configuration Web Agents are installed on the following web servers:

- Windows: IIS 6.0, ASF Apache 2.2
- Solaris: SunOne 6.1 SP2, ASF Apache 2.2
- Linux: SunOne 6.1 SP2, ASF Apache 2.2

1.3.6 Agent Cache

The Web Agent maintains an Agent Cache on each Web Agent machine. In the evaluated configuration, the Agent Cache resides on physical memory (RAM or Hard drive). The Agent Cache has two components; The Agent Resource Cache and the Agent User Cache.

The Agent Resource Cache stores a record of accessed resources that are protected by various realms. This cache speeds up Agent to Policy Server communication, since the Agent knows about resources for which it has already processed requests. This cache is shared by the Web Agent and the Policy Server.

The Agent User Cache stores end users' encrypted Session Tickets, and acts as a session cache by storing user, realm, and resource information. Entries in this cache are invalidated based on timeouts established by the realms an end user accesses.

1.3.7 Web Agent API Layer

In the evaluated configuration, the Web Agent API Layer is a component of the Web Agent that performs as a part of the clustering feature. All load balancing is done via the Web Agent API Layer logic. The Web Agent API Layer is responsible for dynamically balancing the load between Policy Servers in a cluster based on server response time, and for failing-over to another cluster under the failover criteria. For example, if the response time for one Policy Server within a cluster is too slow, the Web Agent API Layer will defer the request to the another Policy Server within that cluster. If the primary cluster fails (cluster failover is defined by a configurable threshold of policy servers being down

within a cluster) and failover is enabled, a backup cluster takes over operations. Cluster configuration is specified through the Web Agent API Layer.

1.3.8 Policy Server Management Tasks

The TOE administrator, is responsible for system-level configuration and the tuning of the Operational Environment, monitoring and ensuring its performance, as well as management of end users and end user sessions as necessary. The administrator must perform most fundamental system configuration tasks using the Policy Server Management Console. However tasks related to the establishment of policies to protect resources are performed remotely using the Siteminder WAM Administrative UI.Policy Server. Some WAM Administrative UI management tasks include:

- Cache Management
- Configuring and Managing Encryption Keys
- End user Session and Account Management
- Policy Creation and Management

1.3.9 Policy Server Management Tools

The Policy Server provides a number of administrative tools to help manage the TOE environment. These tools may be used initially to configure the TOE, but are not used once the TOE is in an operational state. The following list describes the function of each tool:

- XPSImport and XPSExport – Imports and exports policy store data in XML format.
- Smobjimport – Imports policy data into the policy store.
- Smobjexport – Contains arguments that allows an export of an entire policy store; a specified policy domain; or a specified policy domain and all system objects used by the policy domain. System objects include but are not limited to administrators, Agents, authentication schemes and user directories.
- Smldapsetup – Manages the policy store in an LDAP directory.
- ODBC database SQL scripts –Removes the policy store, and log schema from ODBC databases.
- Smpatchcheck –Checks to make sure all of the required/recommended patches are installed on the Solaris machine.
- Smreg – Changes the Super user password.

1.4 TOE Security Environment

It is assumed that there will be no untrusted users or software on the Policy Server hosts. The Policy Server and Web Agent rely upon the underlying operating system and platform to provide reliable time stamps and to protect the Policy Server and Web Agent Client hosts from interference or tampering. Table 1-6 provides detailed information regarding the supported operating systems. The TOE environment is one where the potential attacker is unsophisticated, with access to only standard equipment and public information about the product.

The TOE security environment is categorized as follows:

- **Identification and authentication** – There are two types of users to the TOE; end users who access protected resources through the Web Agents and Administrative users (Super user, Domain Admin, System Admin) that configure the TOE through the Policy Server Management Console. The TOE relies on the Operational Environment to provide user identification and authentication at the Operating System level for administrators accessing the Policy Server Management Console. The root or Windows administrator on the Policy Server is automatically granted full control of the Policy Server Management Console. This account, therefore, has administrative control over the Policy Server computer and the configuration features available from the Policy Server Management Console. On the Web Agent client, operating system access is based on the user account that the system uses to run the web server, including Web Agent configuration files. The TOE administrators use username and passwords to gain access to the TOE on both the Policy Server and Web Agent machines. Since TOE administrator access to these machines is mapped to underlying OS accounts, the TOE itself does not enforce strong passwords. The TOE utilizes the password enforcing features of the underlying Operating system to enforce strong passwords for administrators. It is important to note that these local interfaces are only used for the installation and configuration of the TOE, once in the evaluated configuration all administration will be performed via the WAM Administrative UI.

- Audit Review – The TSF relies on the Operational Environment to protect its audit records which are stored in flat files on the local OS of the Policy Server and Web Agent. The TSF also relies on the Operational Environment to provide the ability to view the records and provide access control to the audit information, ensuring only users with equal or greater privileges than the user who installed the TOE can view the audit records.

- **Cryptographic support** - The TSF relies on the Operational Environment to provide a trusted communication path between remote end users and the Web Agent and remote administrators and the WAM Administrative UI via HTTP over SSL v3.0. This provides assured identification of its end points and protection of communication from modification and disclosure.

- **Partial protection of TSF** - The TOE relies on the underlying OS to provide security capabilities for the TOE's protection. The TSF relies on the host OS to prevent other applications from:
    - Interfering with an executing TSF
    - Bypassing the TOE security functions at the OS level, and
    - Modifying TSF configuration, audit data, and executable images on disk.

- **Reliable Time** – the Policy Server and Web Agent rely on the underlying OS for reliable time. TOE functions such as audit logging rely on reliable time stamps.

Component	TOE Version	Platforms
Policy Server WAM Administrative UI Web Agent	r12 SP1	Linux Red Hat Advanced Server 4.0
		Microsoft Windows 2003 SP2
		Solaris 10
Policy Store User Store	r12 SP1	SunOne LDAP 5.2
		Windows 2003 Active Directory
Key Store Administrator Store	r12 SP1	Oracle 10g R2
Application Server	r12 SP1	JBoss 4.0.5
		WebLogic 9.2
Web Servers	r12 SP1	SunOne 6.1 SP 2
		ASF Apache 2.2
		IIS 6.0

Table 1-6 Supported Operating Systems

In addition to the platforms listed in Table 1-6, the following non-TOE software is required to run the TOE:

- SSL v3.0 implementation
- Transport standards HTTP, and FTP implementations
- SMTP implementation
- Web browser software

1.4.1 End User Accounts

The TOE provides end user session and account management functionality, allowing the session cache to be flushed, enable and disable end users, and manage passwords for individual end users.

1.4.2 End User Authentication and Authorization

End users' access is deteremined by the user information located in the User Store, and the rules and policies in the Policy Store. In the evaluated configuration, the User Store is an LDAP server which contains the end user's information, which includes their authentication information. In the evaluated configuration, the TOE can also utilize a pre-existing External User Directory to retrieve end user information. The Policy Server uses the protected resources name and the end user's information to query the Policy Store to determine the end user's authentication requirements, and the privileges they have on the protected resources.

The end user is authenticated by one or more of the following two methods: Basic over SSL, Windows Authentication Scheme, and/or a x.509 certificate, which is determined by the realm the end user is attempting to access.

1.4.3 Administrator Accounts

There are two options for SiteMinder administrator accounts. The default is to use the SiteMinder administrator accounts (super user account created by the install and the domain administrator accounts) that are created and stored locally in the Administrator Store. The Super User account has the maximum system privileges. This account is used to create all other Administrator accounts and also assigns the categories, rights, and scope of those accounts. There are no password policies associated with these accounts - no maximum/minimum length and no limitations on what characters need to/cannot be used. The accounts do not expire and no password history is maintained.

1.5 Usage and major security features

CA SiteMinder controls end user access to protected resources such as web server content. SiteMinder grants or denies access by enforcing policies defined by SiteMinder Administrators. These policies govern the level of access and which resources are accessed by an end user. The TOE also provides the ability to set roles for security relevant authority as well as to restrict the ability to define and assign scope of tasks to authorized administrators.
CA SiteMinder extends the security and user management functions of the web server by controlling end user access to protected Web content and resources. SiteMinder also enables Web applications to configure unique access control policies.

1.6 Excluded From TOE

- Local configuration files
- Non-persistent sessions (see section 9.1.3.1 Session Ticket Management)
- Use of the Static Agent Key for cookie encryption
- The native Operating System
- Support for RADIUS
- User tokens (e.g. smartcards)
- Multiple Policy Stores
- Virtual Servers

- Proxy Servers (including Reverse Proxy Servers)
- Domino Application Servers
- Security Zones
- Administrative Journal and Event Handler
- Nested Security
- OneView Monitor
- Simple Network Management Protocol (SNMP) Module
- Event Manager Application
- Directory Mapping
- The following Authentication Schemes - CRYPTO Card RB-1, HTML forms, MS Passport, RADIUS CHAP/PAP, RADIUS Server, Safeword Server, Safeword Server and HTML Forms, SecurID, TeleID, Anonymous, Custom, Impersonation, Certificate Mapping
- Credentials Selector
- Variables (Static, Request Context, User Context, Form Post)
- Impersonation

1.7 TOE Type

SiteMinder r12 SP1 provides the following: Web Access Control.

2 TOE Description

This section provides a description of the TOE in its evaluated configuration. This includes the physical and logical boundaries of the TOE

2.1 Physical Boundary

The TOE includes the following CA SiteMinder components:

- Policy Server
- WAM Administrative UI
- Web Agent

The following table illustrates the minimum requirements needed to install the SiteMinder WAM Administrative UI on a Windows and UNIX system.

Component	Windows or Linux	Solaris Unix
CPU	Single or Dual-processor, Intel Pentium III (or compatible), 700-900 MHZ	Sparc Workstation 440 MHz
Memory	512 MB system RAM. 1 GB is recommended	512 MB system RAM. 1 GB is recommended
Available Disk Space	540 MB	540 MB
Temp Directory Space	450 MB	450 MB
JDK	The required JDK version is installed on the same system as the WAM Administrative UI	The required JDK version is installed on the same system as the WAM Administrative UI
Screen Resolution	1024 x 768 or higher resolution with 256 colors or better to properly view the Administrative UI	1024 x 768 or higher resolution with 256 colors or better to properly view the Administrative UI
Web Browser	A supported Web browser to view the WAM Administrative UI	A supported Web browser to view the WAM Administrative UI

Table 1-7 Minimum requirements for installation of WAM Administrative UI

The following table illustrates the minimum requirements needed to install the SiteMinder Policy Server on a Windows and UNIX system.

Component	Windows or Linux	Solaris Unix
CPU	Intel Pentium III or better	Sparc Workstation 440 MHz
Memory	512 MB system RAM	512 MB RAM
Available Disk Space	270 MB	300 MB

Temp Directory Space	180 MB	200 MB (10 MB is required for daily operation)
JRE	The required JRE version is installed on the same system as the Policy Server	The required JRE version is installed on the same system as the Policy Server
LDAP Directory Server or relational database	Ensure that LDAP directory server or relational database being used as a policy store is supported	Ensure that LDAP directory server or relational database being used as a policy store is supported

Table 1-8 Minimum requirements for installation of Policy Server

2.2 Logical Boundary

The logical boundaries of the TOE are described in the terms of the security functionalities that the TOE provides to the systems that utilize this product for end user access control to protected resources.

The logical boundary of the TOE will be broken down into five security class features. The TOE provides the following security features:

Identification, authentication and authorization- The TOE provides user identification, authentication and authorization through the use of user accounts and passwords for Administrators and end users. End users have to identify and authenticate themselves before being allowed access to protected resources. This is based on username, group, and rule group. While administrators have to identify and authenticate themselves to the TOE via the WAM Aministrative UI before gaining access to the management features of the TOE.

End user authentication schemes must be configured using the Siteminder WAM Administrative UI. During authentication, SiteMinder Web Agents communicate with the Policy Server to determine the proper credentials that must be retrieved from an end user who is requesting resources.

- Resources in each policy domain are grouped into realms. In the evaluated configuration, all realms will be created in a "Protected" state (i.e., access to all the resources in the realm are governed via SiteMinder policies and assigned a protection level). When an end user attempts to access a protected network resource, the Policy Server uses the authentication scheme associated with the resource's realm to determine how to authenticate and authorize the end user. The authentication scheme specifies the credentials that the end user must supply for authentication, as well as the method used by the Policy Server to authenticate the end user's identity.

- Authenticated end users of one realm can access a resource in another realm without re-authenticating as long as the second realm is protected by an authentication scheme with an equal or lower protection level. If an end user tries to access a resource protected by an authentication scheme with a higher protection level, SiteMinder prompts the end user to reauthenticate by entering the credentials required by the authentication scheme.

- Three types of authentication schemes are supported by the evaluated configuration: Basic Authentication, Windows Authentication Scheme, and X.509 certificates.

    - Basic Authentication Over SSL v3.0 -- Basic authentication identifies an end user based on a user name and password. The end user's identity is stored in a User Store. With a basic authentication scheme, the Policy Server locates the end user's information in the User Store based on the user name, then verifies that the password matches the one saved in the User Store. The end user will supply his or her credentials via an SSL enabled browser. If the user name and password supplied by the end user match the data in the User Store, SiteMinder authenticates the end user.

    - Windows Authentication Scheme -- The Windows authentication scheme allows SiteMinder to provide access control in deployments with Active Directories running in native mode, as well as Active Directories configured to support NTLM authentication. The Windows Authentication scheme replaces SiteMinder's previous NTLM authentication scheme. Existing NTLM authentication schemes continue to be supported and is configured using the new Windows Authentication scheme. The NTLM authentication scheme is used for resources that are protected by Web Agents on IIS Web servers, and whose end users access resources via Web browsers. This scheme relies on a properly-configured IIS Web server to acquire and verify an end user's credentials. The Policy Server bases authorization decisions on the end user's identity as asserted by the IIS server. The Windows Authentication Scheme relies on the underlying operating system to provide end user authentication along with the TOE.

    - X.509 Client Certificate Authentication -- SiteMinder supports the use of X.509 V3 client certificates. Digital certificates act as cryptographic proof of an end user's identity. Once a certificate is installed on a client, that certificate is used to verify the identity of an end user who is accessing a resource. Certificate authentication is combined with basic authentication to provide an even higher level of access security.

Security Management - The TOE provides management capabilities through the WAM Administrative UI provides account management functionality, allowing an administrator to enable and disable end users, manage passwords for individual end users, and create rules/policies which determine an end user's access rights and privileges on protected resources. The Policy Server Management Console is used to manage the configuration functions of the Policy Server such as logging and starting and stopping the Policy Server processes, these functions are only used to place the TOE into the evaluated configuration. End users have the management ability to change their own passwords.

Security audit – The TOE provides security auditing capabilities. Both the Web Agent and Policy Server provide logging capabilities. The Web Agent logs all successful authorizations of end users and the accesses to protected resources via the Policy Server. The Web Agent logs are stored locally and sent to the Policy Server to store in its local audit logs. Error messages of the Web Agent itself are also logged. The Policy Server records information about the Policy Server itself and logs authentication, authorization, administrator accesses, and administrator changes to policy store objects.

Partial protection of TSF – The Policy Server and Web Agent provide partial protection of TSF data. The TOE presents limited access to end users. It maintains and controls individual sessions for end users.

Resource Utilization- The Web Agent (Web Agent API Layer) is responsible for dynamically balancing the load between Policy Servers in a cluster based on server response time. The Web Agent API Layer will also fail-back to the original cluster when the threshold requirements of available policy servers in that cluster are met.

2.2.1 Cryptographic Support

The TOE provides cryptographic support. The cryptography used in this product has not been FIPS certified nor has it been analyzed or tested to conform to cryptographic standards during this evaluation. All cryptography has only been asserted as tested by the vendor. The Policy Server communicates with the Web Agent and WAM Administrative UI via an encrypted channel, which protects the data being transferred from disclosure and modification. The TSF provides encryption of TOE information stored in its different stores and information sent among the different components of the TOE. SiteMinder r12 SP1 should be installed in "FIPS 140 mode" wherein the AES algorithm is used.

3 Conformance Claims

3.1 CC Version

This ST is CC v3.1.

3.2 CC Part 2 extended

This ST and Target of Evaluation (TOE) is Part 2 extended for EAL3 to include all applicable NIAP and International interpretations through 13 May 2008.

3.3 CC Part 3 conformant plus flaw remediation

This ST and Target of Evaluation (TOE) is Part 3 conformant plus flaw remediation for EAL3 to include all applicable NIAP and International interpretations through 13 May 2008.

3.4 PP Claims

This ST does not claim Protection Profile (PP) conformance.

3.5 Package Claims

This ST claims a package for EAL3.

3.6 Package Name conformant or Package Name Augmented

This ST and Target of Evaluation (TOE) is conformant to EAL package claims augmented with ALC_FLR.1 and ASE_TSS.2.

3.7 Conformance Claim Rationale

There is no Conformance Claim rationale for this ST.

4 Security Problem Definition

4.1 Threats

The TOE itself has threats and the TOE is also responsible for addressing threats to the environment in which it resides. The assumed level of expertise of the attacker for all the threats is unsophisticated. The following are threats addressed by the TOE.

T.ACCESS	Authorized users could gain electronic access to protected network resources by attempting to establish a connection that they are not permitted to perform.
T.ADMIN_ERROR	An administrator may incorrectly install or configure the TOE, or install a corrupted TOE resulting in ineffective security mechanisms.
T.AUDIT_COMPROMISE	A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.
T.EAVESDROPPING	A malicious user could eavesdrop on network traffic to gain unauthorized access to TOE data.
T.MASK	Users whether they be malicious or non-malicious, could gain unauthorised access to the TOE by bypassing identification and authentication countermeasures.
T.MASQUERADE	A user may masquerade as an authorized user or an authorized IT entity to gain access to data or TOE resources.
T.UNAUTH	Users could gain unauthorised access to the web resources by bypassing identification and authentication requirements.

4.2 Organizational Security Policies

There are no Organizational Security Policies that apply to the TOE.

4.3 Assumptions

The specific conditions listed in this section are assumed to exist in the environment in which the TOE is deployed. These assumptions are necessary as a result of practical realities in the development of the TOE security requirements and the essential environmental conditions on the use of the TOE.

4.3.1 Personnel Assumptions

A.ADMIN One or more authorised administrators will be assigned to install, configure and manage the TOE and the security of the information it contains.

A.PATCHES System Administrators exercise due diligence to update the TOE with the latest patches and patch the Operational Environment (e.g., OS and database) so they are not susceptible to network attacks.

A.NOEVIL Users and administrators of the TOE are not careless, wilfully negligent, or hostile and will follow and abide by the instructions provided by the organization's guidance documentation.

4.3.2 Physical Assumptions

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

5 Security Objectives

5.1 Security Objectives for the TOE

The following security objectives are to be satisfied by the TOE.

O.ACCESS The TOE will provide measures to authorize users to access specified TOE resources once the user has been authenticated. User authorization is based on access rights configured by the authorized users of the TOE.

O.AUDIT The TOE will provide measures for recording security relevant events that will assist the authorized users in detecting misuse of the TOE and/or its security features that would compromise the integrity of the TOE and violate the security objectives of the TOE.

O.AUTH The TOE will provide measures to uniquely identify all users and will authenticate the claimed identity prior to granting a user access to the resources protected by the TOE. The TOE will provide measures to uniquely identify all administrators and will authenticate the claimed identity prior to granting an administrator access to the TOE.

O.MANAGE The TOE will provide authorized administrators with the resources to manage and monitor user accounts, resources, and security information relative to the TOE.

O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management.

O.EAVESDROPPING The TOE will encrypt TSF data that traverses the network to prevent malicious users from gaining unauthorized access to TOE data.

O.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. The TSF will provide load balancing within clusters and failover between clusters, which allows for continued operation of the TOE in the event of a failure within or between clusters.

O.ROBUST_TOE_ACCESS	The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate.

5.2 Security Objectives for the operational environment of the TOE

The following security objectives for the Operational environment of the TOE must be satisfied in order for the TOE to fulfill its security objectives.

OE. ADMIN	One or more authorized administrators will be assigned to install, configure and manage the TOE and the security of the information it contains.
OE.FILESYS	The security features offered by the underlying Operating System and Database protect the files used by the TOE.
OE.ROBUST_ACCESS	The Operational environment will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate.
OE.EAVESDROPPING	The Operational Environemnt will encrypt TSF data that traverses the network to prevent malicious users from gaining unauthorized access to TOE data.
OE.NOEVIL	All end users and administrators of the TOE are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided by the organization's guidance documentation.
OE.LOCATE	The TOE will be located within controlled access facilities that will prevent unauthorized physical access.

6 Extended Security Functional Requirements

Extended Security Functional Requirements for the TOE
The Table below contains the extended security functional requirements for the TOE:

Security Function	Security Functional Components
Identification and Authentication	FIA_UAU_EXT.5(1) Multiple authentication schemes

Table 1-9 Extended Security Functional Requirements for the TOE

6.1.1 Class FIA: Identification and Authentication

Identification and Authentication is required to ensure that users are associated with the proper security attributes (e.g. identity, groups, roles, security or integrity levels). The following extended requirements for the FIA class have been included in this ST because the TOE is capable of performing I&A schemes that are not covered by CC Part 2. Refer to the Extended Security Requirements Rationale in Section 8 for more information.

6.1.1.1 FIA_UAU_EXT.5(1) Multiple authentication schemes

Hierarchical to:	No other components.
FIA_UAU_EXT.5.1(1)	The TSF shall provide [*x.509, Windows Authentication Scheme, and Basic over SSL*] to support end user authentication.
FIA_UAU_EXT.5.2(1)	The TSF shall authenticate any user's claimed identity according to the [*the realm and its associated protection level*].
Dependencies:	No dependencies.
Application Note:	*Windows Authentication Scheme relies on the underlying OS, this SFR has been extended. This SFR is implemented partially by the TOE and Operational environment .*

6.2 Extended Security Assurance Requirements

There are no extended Security Assurance Requirements in this ST.

6.3 Extended Security Requirements for the Operational environment

The Table below lists the extended security functional requirements for the Operational environment

Security Function	Security Functional Components
Cryptographic Support	FCS_CKM_EXT.1 Cryptographic key generation
	FCS_CKM_EXT.4 Cryptographic key destruction
	FCS_COP_EXT.1 Cryptographic operation
Identification and Authentication	FIA_UAU_EXT.5 (2) Windows Authentication Schemes
Protection of the TSF	FPT_STM_EXT.1 Reliable Time Stamps
Trusted Path/Channels	FTP_TRP_EXT.1 Trusted Path
Security Audit	FAU_SAR_EXT.1 Audit Review
	FAU_STG_EXT.1 Protected Audit Trail Storage

Table 1-10 Extended Security Functional Requirements for the Operational environment

6.3.1 Class FCS: Cryptographic Support

The TOE relies on the Operation Environment to protect the path from the remote end user and the remote administrator to their respective TOE interface. The TOE's administrative guidance includes information on the necessity of having this path protected by SSL encryption. This will ensure that a user's information (particularly authentication information) cannot be read by another user which is sniffing the packets which are being sent over that path.

6.3.1.1 FCS_CKM_EXT.1 Cryptographic key generation

Hierarchical to: No other components.

FCS_CKM_EXT.1.1 The Operational Environment shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [RSA] and specified cryptographic key sizes [1024 bits] that meet the following: [RFC 2313].

Dependencies: FCS_COP_EXT.1 Cryptographic operation
FCS_CKM_EXT.4 Cryptographic key destruction

Application note: *This SFR supports key generation for SSL.*

6.3.1.2 FCS_CKM_EXT.4 Cryptographic key destruction

Hierarchical to: No other components.

FCS_CKM_EXT.4.1 The Operational Environment shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [overwrite method] that meets the following: [key zeroization].

Dependencies: FCS_CKM_EXT.1 Cryptographic key generation

Application note: *This SFR supports key destruction for SSL.*

6.3.1.3 FCS_COP_EXT.1 Cryptographic Operation

Hierarchical to: No other components.

FCS_COP_EXT.1.1 The Operational Environment shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm [AES] and cryptographic key sizes [128 bits] that meet the following: [RFC 3268].

Dependencies: FCS_CKM_EXT.1 Cryptographic key generation
FCS_CKM_EXT.4 Cryptographic key destruction

Application Note: *This SFR supports the symmetric key usage for SSL.*

6.3.2 Class FIA: Identification and Authentication

Identification and Authentication is required to ensure that end users are associated with the proper security attributes (e.g. identity, groups, roles, security or integrity levels). The following extended requirements for the FIA class have been included in this ST because the Operational environment is capable of performing I&A schemes that are not covered by CC Part 2. Refer to the Extended Security Requirements Rationale in Section 8 for more information.

6.3.2.1 FIA_UAU_EXT.5 (2) Windows Authentication Schemes

Hierarchical to: No other components.

FIA_UAU_EXT.5.1 (2)	The Operational environment shall provide: [*Gathers Windows credentials for resources stored on an IIS 6.0 Web or ASF Apache 2.2 server or Solaris 10 server and accessed by an Internet browser. IIS 6.0 Web Agent supports the Windows authentication scheme Windows Password*] to support end user authentication.
FIA_UAU_EXT.5.2 (2)	The Operational environment shall authenticate any end user's claimed identity according to the [*following rules: CA SiteMinder end users must use one of the Windows or Solaris authentication scheme listed in FIA_UAU_EXT.5.1(2) before accessing a protected resource*].
Dependencies:	No dependencies.
Application Note:	*SiteMinder's browser support is independent of the type of agent used.*

6.3.2.2 FIA_UID_EXT.2 User identification before any action

Hierarchical to:	No other components.
FIA_UID_EXT.2.1	The Operational environment shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.
Dependencies:	No dependencies.

6.3.3 Class FAU: Security Audit

Security auditing involves recognising, recording, storing, and analysing information related to security relevant activities (i.e. activities controlled by the TSF). The following extended requirements for the FAU class have been included in this ST because the Operational Environment is capable of performing auditing events that are not covered by CC Part 2. Refer to the Extended Security Requirements Rationale in Section 8 for more information.

6.3.3.1 FAU_SAR_EXT.1 Audit review

Hierarchical to: No other components.

FAU_SAR_EXT.1.1 The Operational Environment shall provide [*an authorized administrator*] with the capability to read [*role requesting access to objects/resources,end user role assignment, role based access authorization result, object/resource for which access is being requested,*] from the audit records.

FAU_SAR_EXT.1.2 The Operational Environment shall provide the audit records in a manner suitable for the administrator to interpret the information.

Dependencies: FAU_GEN.1 Audit data generation

Application Note: *The TOE writes its audit records to local files on the OS which it is installed upon. The TOE relies on the OS's authentication and authorization mechanisms to determine who can have access to these audit files. In the evaluated configuration the TOE has been installed with the OS's root or admin account; which means that all audit files created by the TOE will have the permission of that root or admin account.*

6.3.3.2 FAU_STG_EXT.1 Protected audit trail storage

Hierarchical to: No other components.

FAU_STG_EXT.1.1 The Operational Environment shall protect the stored audit records from unauthorized deletion.

FAU_STG_EXT.1.2 The Operational Environment shall be able to [*prevent*] unauthorized modifications to the audit records in the audit trail.

Dependencies: FAU_GEN_1 Audit data generation

Application Note: *This requirement represents the data stored in the local audit files.*

6.3.4 Class FTP: Trusted Path/Channels

The TOE relies on the Operation Environment to protect the path from the remote end user and the remote administrator to their respective TOE interface. The TOE's administrative guidance includes information on the necessity of having this path protected by SSL encryption. This will ensure that a user's information (particularly authentication information) cannot be read by another user which is sniffing the packets which are being sent over that path.

6.3.4.1 FTP_TRP_EXT.1 Trusted Path

Hierarchical to: No other components.

FTP_TRP_EXT.1.1 The Operational Environment shall provide a communication path between the TSF and [*remote*] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [*modification, disclosure*].

FTP_TRP_EXT.1.2 The Operational Environment shall permit [*remote users*] to initiate communication via the trusted path.

FTP_TRP_EXT.1.3 The TSF shall require the use of the trusted path for [*initial user authentication, all other TSF mediated actions]*].

Dependencies: No dependencies

6.3.5 Class FPT: Protection of the TSF

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 following extended requirements for the FPT class have been included in this ST because the Operational Environment is capable of performing protection of the TSF events that are not covered by CC Part 2. Refer to the Extended Security Requirements Rationale in Section 8 for more information.

6.3.5.1 FPT_STM_EXT.1 Reliable time stamps

Hierarchical to:	No other components.
FPT_STM_EXT.1.1	The Operational Environment shall be able to provide reliable time-stamps for use by the TOE.
Dependencies:	No dependencies
Application Note:	*The Underlying OS needs to provide reliable time stamps from the system clock that is used for inclusion in the audit records generated by the TOE.*

6.4 Proper dependencies

All dependencies for the extended security functional requirements were pulled from CC Part 2.

7 Security Functional Requirements

The following table provides a summary of the Security Functional Requirements implemented by the TOE.

Security Function	Security Functional Components
Security Audit	FAU_GEN.1 Audit data generation
	FAU_GEN.2 User identity association
Cryptographic Support	FCS_CKM.1(1) Cryptographic key generation
	FCS_CKM.1(2) Cryptographic key generation
	FCS_CKM.1(3) Cryptographic key generation
	FCS_CKM.4 Cryptographic key destruction
	FCS_COP.1(1) Cryptographic operation
	FCS_COP.1(2) Cryptographic operation
	FCS_COP.1(3) Cryptographic operation
	FCS_COP.1(4) Cryptographic operation
User Data Protection	FDP_ACC.1(1) Subset access control
	FDP_ACC.1(2) Subset access control
	FDP_ACF.1(1) Security attribute based access control
	FDP_ACF.1(2) Security attribute based access control
Identification and Authentication	FIA_AFL.1 Authentication and failure handling
	FIA_ATD.1 User attribute definition
	FIA_SOS.1 Verification of Secrets
	FIA_UAU.1

Security Function	Security Functional Components
	Timing of authentication
	FIA_UAU.6 Re-authenticating
	FIA_UID.2 User identification before any action
Security Management	FMT_MSA.1(1) Management of security attributes
	FMT_MSA.1(2) Management of security attributes
	FMT_MSA.1 (3) Management of security attributes
	FMT_MSA.2 Secure security attributes
	FMT_MSA.3 Static attribute initialization
	FMT_SMR.1 Security management roles
	FMT_SMF.1 Specification of management functions
Protection of the TSF	FPT_FLS.1 Failure with preservation of secure state
Resource Utilization	FRU_FLT.1 Degraded fault tolerance
Trusted Path/Channels	FTP_ITC.1 Inter-TSF trusted channel

Table 1-11 Functional Components

7.1.1 Class FAU: Security Audit

7.1.1.1 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:

a. Start-up and shutdown of the audit functions;

b. All auditable events for the [*not specified*] level of audit; and

c. [all operations listed in Table 1-12, whether allowed or denied by the TOE.]

Application Note: *An audit record is the same as an audit log.*

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a. Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b. For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [remote server host name, remote server host ID].

Dependencies: FPT_STM.1 Reliable time stamps

Application Note: *The remote server account name responsible for the record's creation is the identity of the end user or administrator.*

Application Note: *An audit record is the same as an audit log.*

7.1.1.2 FAU_GEN.2 User identity association

Hierarchical to: No other components.

FAU_GEN.2.1 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

7.1.2 Class FCS: Cryptographic Support

The Cryptography used in this product has not been FIPS certified nor has it been analyzed or tested to conform to cryptographic standards during this evaluation. All cryptography has only been asserted as tested by the vendor.

7.1.2.1 FCS_CKM.1(1) Cryptographic key generation

Hierarchical to: No other components.

FCS_CKM.1.1(1) The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [AES] and specified cryptographic key sizes [128 bits] that meet the following: [none].

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

Application note: *This SFR supports key generation of the Session Key for channel encryption.*

7.1.2.2 FCS_CKM.1(2) Cryptographic key generation

Hierarchical to: No other components.

FCS_CKM.1.1(2) The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [AES Key Wrap] and specified cryptographic key sizes [192 bits] that meet the following: [none].

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

Application note: *This SFR supports key generation of the Agent Keys and Session Ticket Keys by the Policy Server.*

7.1.2.3 FCS_CKM.1(3) Cryptographic key generation

Hierarchical to: No other components.

FCS_CKM.1.1(3) The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [FIPS-140 Key Expansion Algorithm] and specified cryptographic key sizes [128 bits] that meet the following: [none].

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

Application note: *This SFR supports key generation of an AES key which is derived from the Agent Keys or Session Ticket Keys, and is used for the encryption of cookies by the Web Agent.*

7.1.2.4 FCS_CKM.4 Cryptographic key destruction

Hierarchical to: No other components.

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [overwrite method] that meets the following: [key zeroization].

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]
FMT_MSA.2 Secure security attributes

Application note: *This SFR supports key destruction of the Agent Keys and Session Ticket Keys during key rollover.*

7.1.2.5 FCS_COP.1(1) Cryptographic Operation

Hierarchical to: No other components.

FCS_COP.1.1(1) The TSF shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm [AES in OFB mode] and cryptographic key sizes [128 bits] that meet the following: [none].

Dependencies: [FDP_ITC.1 Import of user data without security attributes
Or FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

Application Note: *This SFR supports the Session Key usage for encryption between the Policy Server component and the Web Agent/WAM Administrative UI components.*

7.1.2.6 FCS_COP.1(2) Cryptographic Operation

Hierarchical to: No other components.

FCS_COP.1.1(2) The TSF shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm [AES in CBC mode] and cryptographic key sizes [128 bits] that meet the following: [none].

Dependencies: [FDP_ITC.1 Import of user data without security attributes Or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

Application Note: *This SFR supports the AES key derived from the Session Ticket Key used for encrypting/decrypting Session Tickets, administrator tokens, and Password Services state data.*

7.1.2.7 FCS_COP.1(3) Cryptographic Operation

Hierarchical to: No other components.

FCS_COP.1.1(3) The TSF shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm [AES in CBC mode] and cryptographic key sizes [128 bits] that meet the following: [none].

Dependencies: [FDP_ITC.1 Import of user data without security attributes Or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

Application Note: *This SFR supports the AES key derived from an Agent Key used for the encryption of all cookies created by the Web Agent and sent to an end user.*

7.1.2.8 FCS_COP.1(4) Cryptographic Operation

Hierarchical to: No other components.

FCS_COP.1.1(4) The TSF shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm [AES Key Wrap] and cryptographic key sizes [128 bits] that meet the following: [none].

Dependencies: [FDP_ITC.1 Import of user data without security attributes Or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

Application Note: *This SFR supports the Policy Store Key usage for encrypting/decrypting information within the Policy Store and Key Store.*

7.1.3 Class FDP: User Data Protection

7.1.3.1 FDP_ACC.1(1) Subset access control

Hierarchical to: No other components.

FDP_ACC.1.1(1) The TSF shall enforce the [*User Policy*] on [*end users to access protected resources via the web agent using the operations GET, PUT, POST.*]

Dependencies: FDP_ACF.1 Security attribute based access control

Application Note: *A policy is a Policy Server object that binds end users, rules, responses, and optionally, time restrictions and IP address restrictions together. Policies establish entitlements for a SiteMinder protected entity. When an end user attempts to access a resource, the policy is what SiteMinder ultimately uses to resolve the request.*

7.1.3.2 FDP_ACC.1(2) Subset access control

Hierarchical to: No other components.

FDP_ACC.1.1(2) The TSF shall enforce the [*Administrator Policy*] on [*Administrators performing operations as defined in Table 1-12 to policy server objects*]

Dependencies: FDP_ACF.1 Security attribute based access control

7.1.3.3 FDP_ACF.1(1) Security attribute based access control

Hierarchical to: No other components.

FDP_ACF.1.1(1) The TSF shall enforce the [*Domain Policy*] to objects based on the following: [*Username, Groups, Resources, Realm*].

Application Note: *The above list of subject security attributes is enforced for the CA Siteminder end users.*

Application Note: *The above list of security attributes is enforced for access to protected resources.*

FDP_ACF.1.2(1) The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [

- *An end user is explicitly granted access to a resource by association of the username.*
- *An end user is implicitly granted access to a resource if he/she belongs to a group which has been granted access.*
- *A Rule or Rule Group assigned to the Policy specifies the resources protected by the policy*].

FDP_ACF.1.3(1) The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [*An end user is explicitly granted access to a resource by association of the username*.]

FDP_ACF.1.4(1) TSF shall explicitly deny access of subjects to objects based on the [*on a rule denying access to an explicit resource by a specific end user*].

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

Application Note: *If any matching rules deny access to a resource, processing stops, and the Policy Server returns any responses associated with the deny access rule to the SiteMinder Agent. For the FDP_ACF.1(1) set of requirements, subjects are end users.*

Roles		**Operations**	**Policy Server Objects**	**Interface**
Administrator/ Super User				
Scope	**Tasks**			
System	Manage System and Domain Objects	Create/edit/delete	Agents	WAM ADMIN UI
		Create/edit/delete	Agent Configuration Objects	WAM ADMIN UI
		Create/edit/delete	Agent groups	WAM ADMIN UI
		Create/edit/delete	Host Configuration Objects	WAM ADMIN UI
		Create/edit/delete	policy domains	WAM ADMIN UI
		Create/edit/delete	authentication schemes,	WAM ADMIN UI
		Create/edit/delete	certificate mappings	WAM ADMIN UI
		Create/delete	parent realms in all domains.	WAM ADMIN UI

Roles		Operations	Policy Server Objects	Interface
Administrator/ Super User				
Scope	Tasks			
		Create/edit/delete	administrators.	WAM ADMIN UI
		Flush	all caches, including cached resources.	WAM ADMIN UI
		Change	global settings.	WAM ADMIN UI
		All privileges for Manage Domain Objects listed below.		See below
Domains	Manage Domain Objects	create/edit/delete	rules (in managed domains)	WAM ADMIN UI
		create/edit/delete	rule groups (in managed domains)	WAM ADMIN UI
		create/edit/delete	responses (in managed domains)	WAM ADMIN UI
		create/edit/delete	response groups (in managed domains)	WAM ADMIN UI
		create/edit/delete	policies (in managed domains)	WAM ADMIN UI
		Edit	top-level realms (in managed domains (not resource filters)).	WAM ADMIN UI
		Create/edit/delete	nested realms (in managed domains)	WAM ADMIN UI
		Flush	specific realms from the resource cache, and flush all resources (in privileged domains) from the cache.	WAM ADMIN UI
System	Manage Keys and Password	Create/edit/delete	password policies	WAM ADMIN UI
		Rollover	Agent Keys	WAM ADMIN UI
			Session Ticket Keys	WAM ADMIN UI
Domains	Manage Password Policies	Create/edit/delete	password policies for end users in directories attached to managed domains.	WAM ADMIN UI
System	Manage Users	Flush	all end user session caches, or flush theend user session cache of any individual end user cache from any directory.	WAM ADMIN UI
		Enable/disable	End users in any directory.	WAM ADMIN UI
		Force password change	on any end user in any directory.	WAM ADMIN UI

Roles		Operations	Policy Server Objects	Interface
Administrator/ Super User				
Scope	Tasks			
		Change password	On any end user in any directory	WAM ADMIN UI
Domain	Manage end users	Flush session caches	for individual end users in directories attached to managed domains.	WAM ADMIN UI
		Enable/disable	End users in directories attached to managed domains.	WAM ADMIN UI
		Force password change	on end users in directories attached to managed domains.	WAM ADMIN UI
		Change password	On any end user in any directory	WAM ADMIN UI

Table 1-12 Management of TSF data

7.1.3.4 FDP_ACF.1(2) Security attribute based access control

Hierarchical to: No other components.

FDP_ACF.1.1(2) The TSF shall enforce the [*Administrator Policy*] to objects based on the following: [*task and scope*].

Application Note: *Administrators who have privileges to Manage Domain Objects can create objects in a domain within the administrator's scope.*

FDP_ACF.1.2(2) The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [*An administrator is granted access to a policy server object by association to scope and task*]

FDP_ACF.1.3(2) The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [*The default SiteMinder Super User has full control with no exceptions*]

FDP_ACF.1.4(2) TSF shall explicitly deny access of subjects to objects based on the [*none*].

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

Application Note *For the FDP_ACF.1(2) set of requirements, subjects are administrators.*

7.1.4 Class FIA: Identification and Authentication

7.1.4.1 FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components.

FIA_AFL.1.1 The TSF shall detect when [*an administrator configurable positive integer within [0-2147483647*] unsuccessful authentication attempts occur related to [*end user login attempts*].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been [*met*], the TSF shall : [*lockout the End User account(s) login for the administrative configured number of minutes*].

Dependency: FIA_UAU.1 Timing of authentication

7.1.4.2 FIA_ATD.1 User attribute definition

Hierarchical to: No other components.

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [*see Table 1-13 below*]

Role	Attribute	Specific Attribute
End User	User name	SM_USER SM_USERNAME SM_USERLOGINNAME
	Password	SM_USERPASSWORD
	Session Ticket	SM_USERSESSIONSPEC
	Groups	SM_USERGROUPS
	Nested Groups	SM_USERNESTEDGROUPS
	Policies	SM_USERPOLICIES
	Responses for all Policies	SM_USERPRIVS
	Responses for all Rules under a Realm	SM_USERREALMPRIVS
	Authentication Level	SM_AUTHENTICATIONLEVEL

	Disabled User	SM_USERDISABLEDSTATE
Administrator/Super User	Scope	
	Task	

Table 1-13 SiteMinder Generated User Security Attributes

Dependencies: No dependencies

7.1.4.3 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 [*complexity requirements as defined in the organization's administrator guidance*].

Dependencies: No dependencies

7.1.4.4 FIA_UAU.1 Timing of authentication

Hierarchical to: No other components.

FIA_UAU.1.1 The TSF shall allow [*access unprotected resources*] on behalf of the user to be performed before the user is authenticated.

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

Dependencies: FIA_UID.1 Timing of identification

7.1.4.5 FIA_UAU.6 Re-authenticating

Hierarchical to: No other components.

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions [*if the realm is protected by an authentication scheme with a higher protection level*].

Dependencies: No dependencies

7.1.4.6 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 identify itself before allowing any other TSF-mediated actions on behalf of that user.

Dependencies: No dependencies

7.1.5 Class FMT: Security Management

7.1.5.1 FMT_MSA.1(1) Management of security attributes

Hierarchical to: No other components.

FMT_MSA.1.1(1) The TSF shall enforce the [*Administrator Policy*] to restrict the ability to [*view or perform operations specified in the Management of Security Attributes column in Table 1-14 below*] the security attributes [*as specified in Table 1-14 below*] to [*the roles as specified in Table 1-14 below*]

.

Dependencies: FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions

Application Note: *In the table below, the interface used for managing security attributes is the WAM ADMIN UI.*

Roles		**Security Attribute**	**Management of Security Attributes**
Administrator/Super User			
Scope	**Tasks**		
System	Manage System and Domain Objects	Session Ticket	Flush all caches, including cached resources; Revalidation of the session ticket.
		Session Ticket Key	Rollover of the Session Ticket Key
		Agent Key	Rollover of the agent key.
		Scope	Create/edit/delete administrators.
		Tasks	

Roles		Security Attribute	Management of Security Attributes
Administrator/Super User			
Scope	**Tasks**		
Domains	Manage Domain Objects	Session Ticket	Flush specific realms from the resource cache, and flush all resources (in privileged domains) from the cache; Revalidation of the session ticket.
		Session Ticket Key	Rollover of the Session Ticket Key
		Agent Key	Rollover of the Agent Key.
Domains	Manage Password Policies	Password	Create/edit/delete password policies for end users in directories attached to managed domains.

Roles		Security Attribute	Management of Security Attributes
Administrator/Super User			
Scope	**Tasks**		
System	Manage End Users	Session Ticket	Flush all end user session caches, or flush the end user session cache of any individual end user cache from any directory; Revalidation of the session ticket.
		Session Ticket Key	Rollover of the Session Ticket Key
		Agent Key	Rollover of the agent key.
		User name	Create/edit/delete user name
		Disabled User	Enable/disable end users in any directory.
		Password	Force password change on any end user in any directory.
		Password	Change password of any user
		Groups, Nested Groups	Add Users to Groups and Nested Groups
		Responses for all Policies	Add Users to Policies
		Responses for all Rules under a Realm	Associate a Rule with a Response or Response Group
Domain	Manage End Users	Session Ticket	Flush end user session caches for individual end users in directories attached to managed domains; Revalidation of the session ticket.
		Session Ticket Key	Rollover of the Session Ticket Key
		Agent Key	Rollover of the agent key.
		User name	Create/edit/delete user name
		Disabled User	Enable/disable end users in directories attached to managed domains.
		Password	Force password change on end users in directories attached to managed domains.
		Password	Change password

Roles		Security Attribute	Management of Security Attributes
Administrator/Super User			
Scope	**Tasks**		
		Groups, Nested Groups	Assign Users to Groups and Nested Groups
		Responses for all Policies	Add Users to Policies
		Responses for all Rules under a Realm	Associate a Rule with a Response or Response Group

Table 1-14 Management of Security Attributes

7.1.5.2 FMT_MSA.1(2) Management of security attributes

Hierarchical to:	No other components.
FMT_MSA.1.1(2)	The TSF shall enforce the [*User Policy*] to restrict the ability to [*change*] the security attributes [*password*] to [*end users*].
Dependencies:	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions

7.1.5.3 FMT_MSA.1(3) Management of security attributes

Hierarchical to:	No other components.
FMT_MSA.1.1(3)	The TSF shall enforce the [*Administrator Policy*] to restrict the ability to [*query, modify, or delete*] the security attributes [*as listed in Table 1-16 External LDAP Directory Attributes*] to [*Administrators*].
Dependencies:	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions

7.1.5.4 FMT_MSA.2 Secure security attributes

Hierarchical to:	No other components.
FMT_MSA.2	The TSF shall ensure that only secure values are accepted for [Session Keys, Agent Keys, Session Ticket Keys, AES Keys].
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

7.1.5.5 FMT_MSA.3 Static attribute initialization

Hierarchical to:	No other components.
FMT_MSA.3.1	The TSF shall enforce the [*Administrator 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 [*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

7.1.5.6 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 security management functions: [*as specified in Table 1-12*]

Dependencies: No dependencies

Application Note: *Modify end users includes changing user passwords.*

7.1.5.7 FMT_SMR.1 Security roles

Hierarchical to: No other components.

FMT_SMR.1.1 The TSF shall maintain the roles [*Administrators, End Users, Super User*].

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

Dependencies: FIA_UID.1 Timing of identification

Application Note: *The above security functional requirement component has three roles: Administrators (Domain Admin, System Admin), End Users, and Super User, which are default roles that can be renamed, but not deleted by anyone.*

7.1.6 Class FPT: Protection of the TSF

7.1.6.1 FPT_FLS.1 Failure with preservation of secure state

Hierarchical to: No other components.

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: [*failure of a Policy Server in a SiteMinder cluster is detected*].

Dependencies: No dependencies

7.1.7 Class FRU: Resource Utilization

7.1.7.1 FRU_FLT.1 Degraded fault tolerance

Hierarchical to: No other components.

FRU_FLT.1.1 The TSF shall ensure the operation of [*access control to protected resources*] when the following failures occur: [*failure of a Policy Server in a SiteMinder cluster is detected*].

Dependencies: No dependencies

7.1.8 Class FTP: Trusted Path/Channels

7.1.8.1 FTP_ITC.1 Inter-TSF trusted channel

Hierarchical to: No other components.

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit [*the TSF*] to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [*the transfer of data between the Policy Server component, and the Web Server and WAM Administrative UI components*].

Dependencies: No dependencies

7.2 Security Requirements for the Operational Environment

There are no security functional requirements for the operational Environment in this ST.

7.3 Operations defined

The notation, formatting, and conventions used in this security target (ST) are consistent with version 3.1 of the Common Criteria for Information Technology Security Evaluation. All of the components in this ST are taken directly from Part 2 of the CC except the ones noted with "_EXT" in the component name. Font style and clarifying information conventions were developed to aid the reader.

The CC permits four functional component operations:, iteration, assignment, selection, and refinement to be performed on functional requirements. These operations are defined in Common Criteria, Part 1 as:

7.3.1 Assignments made

Assignments allow the specification of parameters and are specified by the ST author are in ***[italicized bold text].***

7.3.2 Iterations made

Iterations allow a component to be used more than once with varying operations and are identified with a dash and a pound sign "-#". These follow the short family name and allow components to be used more than once with varying operations. An asterisk "*" refers to all iterations of a component.

7.3.3 Selections made

Selections allow the specification of one or more items from a list and are specified by the ST author are in ***[italicized bold text].***

7.3.4 Refinements made

Refinements allow the addition of details and are identified with "Refinement:" right after the short name. Additions to the CC text are specified in **italicized bold and underlined text.**

8 Security Assurance Requirements

This section identifies the Security Assurance Requirement components met by the TOE. These assurance components meet the requirements for EAL3 augmented with ALC_FLR.1 and ASE_TSS.2.

8.1 Security Architecture

8.1.1 Security architecture description (ADV_ARC.1)

ADV_ARC.1.1D	The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.
ADV_ARC.1.2D	The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.
ADV_ARC.1.3D	The developer shall provide a security architecture description of the TSF.
ADV_ARC.1.1C	The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.
ADV_ARC.1.2C	The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.
ADV_ARC.1.3C	The security architecture description shall describe how the TSF initialisation process is secure.
ADV_ARC.1.4C	The security architecture description shall demonstrate that the TSF protects itself from tampering.
ADV_ARC.1.5C	The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.
ADV_ARC.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.1.2 Functional specification with complete summary (ADV_FSP.3)

ADV_FSP.3.1D	The developer shall provide a functional specification.
ADV_FSP.3.2D	The developer shall provide a tracing from the functional specification to the SFRs.
ADV_FSP.3.1C	The functional specification shall completely represent the TSF.
ADV_FSP.3.2C	The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.3.3C	The functional specification shall identify and describe all parameters associated with each TSFI.
ADV_FSP.3.4C	For each SFR-enforcing TSFI, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.
ADV_FSP.3.5C	For each SFR-enforcing TSFI, the functional specification shall describe direct error messages resulting from security enforcing effects and exceptions associated with invocation of the TSFI.
ADV_FSP.3.6C	The functional specification shall summarize the SFR-supporting and SFR-non-interfering actions associated with each TSFI.
ADV_FSP.3.7C	The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.
ADV_FSP.3.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.3.2E	The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

8.1.3 Architectural design (ADV_TDS.2)

ADV_TDS.2.1D ADV_TDS.2.2D	The developer shall provide the design of the TOE. The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.
ADV_TDS.2.1C	The design shall describe the structure of the TOE in terms of subsystems.
ADV_TDS.2.2C	The design shall identify all subsystems of the TSF.
ADV_TDS.2.3C	The design shall describe the behaviour of each SFR non-interfering subsystem of the TSF in detail sufficient to determine that it is SFR non-interfering.
ADV_TDS.2.4C	The design shall describe the SFR-enforcing behaviour of the SFR-enforcing subsystems.
ADV_TDS.2.5C	The design shall summarize the SFR-supporting and SFR-non-interfering behaviour of the SFR-enforcing subsystems.
ADV_TDS.2.6C	The design shall summarize the behavior of the SFR-supporting subsystems.
ADV_TDS.2.7C	The design shall provide a description of the interactions among all subsystems of the TSF.

ADV_TDS.2.8C The mapping shall demonstrate that all behaviour described in the TOE design is mapped to the TSFIs that invoke it.

ADV_TDS.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_TDS.2.2E The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

8.2 Guidance Documents

8.2.1 Operational user guidance (AGD_OPE.1)

AGD_OPE.1.1D The developer shall provide operational user guidance.

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

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

8.2.2 Preparative Procedures (AGD_PRE.1)

AGD_PRE.1.1D	The developer shall provide the TOE including its preparative procedures.
AGD_PRE.1.1C	The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.
AGD_PRE.1.2C	The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.
AGD_PRE.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AGD_PRE.1.2E	The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

8.3 Life Cycle Support

8.3.1 Authorization Controls (ALC_CMC.3)

ALC_CMC.3.1D	The developer shall provide the TOE and a reference for the TOE.
ALC_CMC.3.2D	The developer shall provide the CM documentation.
ALC_CMC.3.3D	The developer shall use a CM system.
ALC_CMC.3.1C	The TOE shall be labeled with its unique reference.
ALC_CMC.3.2C	The CM documentation shall describe the method used to uniquely identify the configuration items.
ALC_CMC.3.3C	The CM system shall uniquely identify all configuration items.
ALC_CMC.3.4C	The CM system shall provide measures such that only authorized changes are made to the configuration items.
ALC_CMC.3.5C	The CM documentation shall include a CM plan.
ALC_CMC.3.6C	The CM plan shall describe how the CM system is used for the development of the TOE.
ALC_CMC.3.7C	The evidence shall demonstrate that all configuration items are being maintained under the CM system.
ALC_CMC.3.8C	The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.

ALC_CMC.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.3.2 CM Scope (ALC_CMS.3)

ALC_CMS.3.1D The developer shall provide a configuration list for the TOE.

ALC_CMS.3.1C The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; and the implementation representation.

ALC_CMS.3.2C The configuration list shall uniquely identify the configuration items.

ALC_CMS.3.3C For each TSF relevant configuration item, the configuration list shall indicate the developer of the item. Evaluator action elements:

ALC_CMS.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.3.3 Delivery Procedures (ALC_DEL.1)

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

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

ALC_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

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

8.3.4 Identification of security measures (ALC_DVS.1)

ALC_DVS.1.1D The developer shall produce development security documentation.

ALC_DVS.1.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

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

ALC_DVS.1.2E The evaluator shall confirm that the security measures are being applied.

8.3.5 Life-cycle definition (ALC_LCD.1)

ALC_LCD.1.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC_LCD.1.2D The developer shall provide life-cycle definition documentation.

ALC_LCD.1.1C The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

ALC_LCD.1.2C The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE. Evaluator action elements:

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

8.3.6 Basic flaw remediation (ALC_FLR.1)

ALC_FLR.1.1D The developer shall document flaw remediation procedures addressed to TOE developers. Content and presentation elements:

ALC_FLR.1.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.1.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.1.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.1.4C The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users. Evaluator action elements:

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

8.4 Security Target Evaluation

8.4.1 Conformance Claims (ASE_CCL.1)

ASE_CCL.1.1D The developer shall provide a conformance claim.

ASE_CCL.1.2D	The developer shall provide a conformance claim rationale.
ASE_CCL.1.1C	The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.
ASE_CCL.1.2C	The CC conformance claim shall describe the conformance of the ST to CCPart 2 as either CC Part 2 conformant or CC Part 2 extended.
ASE_CCL.1.3C	The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.
ASE_CCL.1.4C	The CC conformance claim shall be consistent with the extended components definition.
ASE_CCL.1.5C	The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.
ASE_CCL.1.6C	The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.
ASE_CCL.1.7C	The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.
ASE_CCL.1.8C	The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.

8.4.2 Extended components definition (ASE_ECD.1)

ASE_ECD.1.1D	The developer shall provide a statement of security requirements.
ASE_ECD.1.2D	The developer shall provide an extended components definition.
ASE_ECD.1.1C	The statement of security requirements shall identify all extended security requirements.
ASE_ECD.1.2C	The extended components definition shall define an extended component for each extended security requirement.
ASE_ECD.1.3C	The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.

ASE_ECD.1.4C	The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.
ASE_ECD.1.5C	The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.
ASE_ECD.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ASE_ECD.1.2E	The evaluator shall confirm that no extended component can be clearly expressed using existing components.

8.4.3 ST Introduction (ASE_INT.1)

ASE_INT.1.1D	The developer shall provide an ST introduction.
ASE_INT.1.1C	The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.
ASE_INT.1.2C	The ST reference shall uniquely identify the ST.
ASE_INT.1.3C	The TOE reference shall identify the TOE.
ASE_INT.1.4C	The TOE overview shall summarize the usage and major security features of the TOE.
ASE_INT.1.5C	The TOE overview shall identify the TOE type.
ASE_INT.1.6C	The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.
ASE_INT.1.7C	The TOE description shall describe the physical scope of the TOE.
ASE_INT.1.8C	The TOE description shall describe the logical scope of the TOE.
ASE_INT.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ASE_INT.1.2E	The evaluator shall confirm that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

8.4.4 Security Objectives (ASE_OBJ.2)

ASE_OBJ.2.1D	The developer shall provide a statement of security objectives.
ASE_OBJ.2.2D	The developer shall provide a security objectives rationale.

ASE_OBJ.2.1C	The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.
ASE_OBJ.2.2C	The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.
ASE_OBJ.2.3C	The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.
ASE_OBJ.2.4C	The security objectives rationale shall demonstrate that the security objectives counter all threats.
ASE_OBJ.2.5C	The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.
ASE_OBJ.2.6C	The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.
ASE_OBJ.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.4.5 Security Requirements (ASE_REQ.2)

ASE_REQ.1.1D	The developer shall provide a statement of security requirements.
ASE_REQ.1.2D	The developer shall provide a security requirements rationale.
ASE_REQ.1.1C	The statement of security requirements shall describe the SFRs and the SARs.
ASE_REQ.1.2C	All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.
ASE_REQ.1.3C	The statement of security requirements shall identify all operations on the security requirements.
ASE_REQ.1.4C	All operations shall be performed correctly.
ASE_REQ.1.5C	Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.
ASE_REQ.1.6C	The statement of security requirements shall be internally consistent.

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

8.4.6 Security Problem Definition (ASE_SPD.1)

ASE_SPD.1.1D The developer shall provide a security problem definition.

ASE_SPD.1.1C The security problem definition shall describe the threats.

ASE_SPD.1.2C All threats shall be described in terms of a threat agent, an asset, and an adverse action.

ASE_SPD.1.3C The security problem definition shall describe the OSPs.

ASE_SPD.1.4C The security problem definition shall describe the assumptions about the operational environment of the TOE.

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

8.4.7 TOE Summary Specification (ASE_TSS.2)

ASE_TSS.2.1D The developer shall provide a TOE summary specification.

ASE_TSS.2.1C The TOE summary specification shall describe how the TOE meets each SFR.

ASE_TSS.2.2C The TOE summary specification shall describe how the TOE protects itself against interference and logical tampering.

ASE_TSS.2.3C The TOE summary specification shall describe how the TOE protects itself against bypass.

ASE_TSS.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_TSS.2.2E The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

8.5 Tests

8.5.1 Analysis of Coverage (ATE_COV.2)

ATE_COV.2.1D The developer shall provide an analysis of the test coverage.

ATE_COV.2.1C The analysis of the test coverage shall demonstrate the correspondence between the tests in the test documentation and the TSFIs in the functional specification.

ATE_COV.2.2C The analysis of the test coverage shall demonstrate that all TSFIs in the functional specification have been tested.

ATE_COV.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.5.2 Basic Design (ATE_DPT.1)

ATE_DPT.1.1D The developer shall provide the analysis of the depth of testing.

ATE_DPT.1.1C The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems in the TOE design.

ATE_DPT.1.2C The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.

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

8.5.3 Functional Tests (ATE_FUN.1)

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

ATE_FUN.1.2D The developer shall provide test documentation

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

ATE_FUN.1.2C The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.3C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4C The actual test results shall be consistent with the expected test results.

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

8.5.4 Independent Testing (ATE_IND.2)

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

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

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

ATE_IND.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

ATE_IND.2.3E The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

8.6 Vulnerability Assessment

8.6.1 Vulnerability Analysis (AVA_VAN.2)

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

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

AVA_VAN.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.2.2E The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.2.3E The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.

AVA_VAN.2.4E The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

9 TOE Summary Specification

9.1 TOE Security Functions

This section describes the security functions provided by the TOE.

9.1.1 Access Control

There are two types of access controls within the TOE; Access Control for administrators and Access Control for end users.

9.1.1.1 Administrator Access Control

The Siteminder WAM Administrative UI authenticates SiteMinder administrator accounts using the Administrator Store. WAM Admin UI component provides a web accessable GUI (also known as the WEB Admin GUI) which allows an adminstrator to manage the TOE. Administrators accessing the Policy Server Management Console are required by the underlying OS to enter a username and password, but they are not required to authenticate to the TOE via one of the authentication schemes mentioned in this document (see section 2.2 of the INT for more information).

Depending on one's role in an organization, SiteMinder administrators have access to different resources and features, and are responsible for different tasks, based on policies created via the Administrative UI. These policies define rules and the tasks which are performed by the domain administrator in the policy domain. At least one administrator must be assigned to each domain. An administrator can be assigned to more than one domain; however, he can only perform the duties within the scope of the domain(s) to which he's assigned. The SiteMinder administrative model implements fine-grained administrative privileges, so the management of Policy Server objects and SiteMinder tools across a few or many individuals in an organization are organized accordingly.

The remote Super User account is included in the TOE and has all of the same attributes as an administrator. This account is the default account that is set up at installation and is used to create all other Administrator accounts and also assigns the categories, rights, and scope of those accounts. It is not recommended that the Super User account be used for day-to-day administration. An administrator can only create another administrator with the same or lesser privileges. For example, an administrator with GUI and reports privileges cannot create an administrator with GUI, reports privileges and local API privileges. Administrator privileges are determined by the tasks that are enabled for the administrator. These privileges allow administrators to use a set of Policy Server features.

9.1.1.2 End User Access Control to Protected Resources

This section refers only to end user access to protected resources. By default, if a resource is unprotected, access to that resource is allowed.

Administrators have the ability to create end users and to assign all end users to groups as needed. These end users and/or groups are then added to policies in order to gain access to protected resources. A policy exists as part of a Policy Domain. The policy defines the type of access for an end user. A policy binding is created between the selected end users and the policy when an administrator adds an end user or group to the policy. When an end user tries to access a protected resource, the policy verifies that the end user is part of its policy binding, and then enforces the rules included in the policy to see if the end user is allowed to access the resource, if any authentication and authorization events must be processed, and if any responses should be generated and returned to SiteMinder Web Agents. When the Policy Server processes rules, it looks for the longest matching string that consists of a resource filter specified in the realm and a resource specified in the rule. The following figure illustrates all of the possible components of a policy.

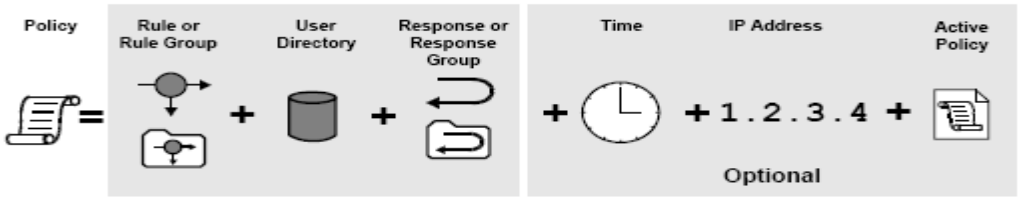

Figure 1-3 Components of a Policy

- **Rules/Rule Groups -** A policy must contain at least one rule or rule group. A rule identifies a specific resource or resources that are included in the policy.
- **Users -** A policy must specify the end users (via username) or groups of end users (via group name) that are affected by the policy. Connections to these end users or groups of end users must be configured in the SiteMinder User Store dialog. Only end users or end user groups for User Stores that are included in the policy domain in which the policy is located may be associated with a policy.
- **Responses -** A response defines the action that is triggered when an end user accesses a resource specified in a rule. Responses can return attributes from a User Store for use by other applications or to customize the appearance of a resource. Responses can also trigger actions based on authentication and authorization events.
- **IP Addresses** (Optional) - A policy may be limited to specific end user IP addresses. Once an administrator adds an IP address restriction to a policy, if an end user attempts to access a resource from an IP address not specified in the policy, the policy will not apply for the end user, and therefore will not allow/deny access or process any responses.
- **Time Restrictions** (Optional) - A policy may be limited to specific days or a ranges of hours. A policy with a time restriction will not be applied outside specified times, and therefore will not allow/deny access to protected resources or process any responses.

By default, when a policy is created, the policy is enabled. The WAM Administrative UI allows an administrator to enable and disable policies. When a policy is enabled, rules contained in the policy apply when end users attempt to access the resources specified in the rules. If an administrator disables a policy, the rules contained in the policy still apply, but no end user will be authorized by the policy. Any resources specified in rules contained in the policy are still protected. Until an administrator enables the policy, no end users may access resources associated with the rules specified in the policy. However, if another enabled policy allows access to a resource in the disabled policy, end users associated with the enabled policy may access the resource. In addition an administrator can delete a policy. When an administrator deletes a policy, all of the elements in the policy still exist. It is only the grouping (or binding) of those elements that are removed.

In a deployment of nested realms, the Policy Server keeps a ranked list of matching realms for use during processing. If any matching rules deny access to a resource, processing stops, and the Policy Server returns any responses associated with the deny access rule to the SiteMinder Agent. The Policy Server collects responses from all matching rules enforced. When the Policy Server finishes collecting responses based on rules, it deletes any duplicate responses. In a deployment that uses nested realms, the Policy Server collects the entire list of accumulated responses for all matching rules.
A rule with an authorization event action may be coupled with a SiteMinder response in a policy. When an end user is authorized (or rejected), the Policy Server passes any responses associated with the applicable On-Access rule back to the requesting Agent.
Rules with a Web Agent action (Get, Put, Post) either allow or deny access to the resource(s) specified by a rule when one of the HTTP actions specified in the rule occurs. When a rule that specifies Allow Access applies and the end user has authenticated successfully, SiteMinder allows the end user to access the specified resource. There is no Deny Access for protected resources (urls) "out of the box". Rules must be created for this to happen. If a rule specifies Deny Access, SiteMinder denies access to the successfully authenticated end user. Deny Access rules may be added to policies to provide an additional layer of security by rejecting specific individuals or groups who should not have access to a resource. Allow Access is the default. Deny Access rules take precedence over allow access rules. If a Deny Access rule and an Allow Access rule applies when an end user attempts to access a resource, the presence of the deny access rule overrides all Allow Access rules.

The Web Agent rule actions are:

- Get - Retrieves a resource for viewing via HTTP.
- Put - Supports legacy HTTP actions.
- Post - Posts information supplied by a user via HTTP.

- Authentication events occur as SiteMinder tries to establish an end user's identity. As a rule action, an authentication event causes the Policy Server to enforce a rule at a particular point in the authentication process. Authentication events occur when an end user accesses a resource protected by a rule that includes an On-Auth event. Unlike Web Agent actions or authorization events, authentication events always apply to the entire realm and the protection level associated with the realm. An administrator can't create an On-Auth rule that applies to a portion of a realm. The following is a list of possible On-Auth events:
    - OnAuthAccept - Occurs if authentication was successful. This event may be used to redirect an end user after a successful authentication.
    - OnAuthReject - Occurs if authentication failed for an end user that is bound to a policy containing an On-Auth-Reject rule. This event may be used to redirect the end user after a failed authentication.
        - *Note: OnAuthAccept and OnAuthReject events are enforced both at authentication time (when the end user enters his / her username and password) and at validation time (when the end user's cookie is read for end user information).*
    - OnAuthAttempt - Occurs if the end user was rejected because SiteMinder does not know this end user (an unregistered end user, for example, can be redirected to register first).
    - OnAuthChallenge - Occurs when custom challenge-response authentication schemes are activated (for example, a token code).
    - OnAuthUserNotFound - This event is only used to trigger Active Responses.

Authorization events occur as SiteMinder verifies whether or not an end user is authorized to access a resource. As a rule action, an authorization event causes the Policy Server to enforce a rule at a particular point in the authorization process. The following is a list of possible authorization events:

- OnAccessAccept - Occurs as the result of successful authorization. This event may be used to redirect end users who are authorized to access a resource.
- OnAccessReject - Occurs as the result of failed authorization. This event may be used to redirect end users who are not authorized to access a resource.

User-initiated password changes allow end users to change their passwords without any intervention from an administrator. End users can elect to change their passwords by clicking a link to access the Password Change Request form. To enable user-initiated password changes, the Policy Server administrator must add a Change Password link to an HTML page. For example, administrators might add this link to a login page so end users can opt to change their password at login.

9.1.2 Identification and Authentication

Each end user must be successfully identified and authenticated before being allowed access to protected resources. The Policy Server verifies an end user's identity by retrieving user attributes contained in the User Store. The TOE employs authentication schemes associated with the resource's realm and protection level. Authentication schemes provide a way to collect credentials and determine the identity of an end user. If an end user tries to access a resource with a higher protection level than the one he is currently accessing, he will be required to re-authenticate. If an end user attempts to access a resource with an equal or lower protection level within the same session, he is not required to re-authentiate. The TOE supports a variety of authentication schemes. In the evaluated configuration the following schemes are used: basic username/password authentication, Windows authentication schemes, to digital certificate authentication (x.509). Simple schemes can be used for low risk network resources, while complex schemes may be employed to ensure added security for critical network resources. Authentication schemes must be configured using the WAM Administrative UI. During authentication, SiteMinder Web Agents communicate with the Policy Server to determine the proper credentials that must be retrieved from an end user who is requesting resources.

The following diagram outlines how the system authenticates end users to protected resources:

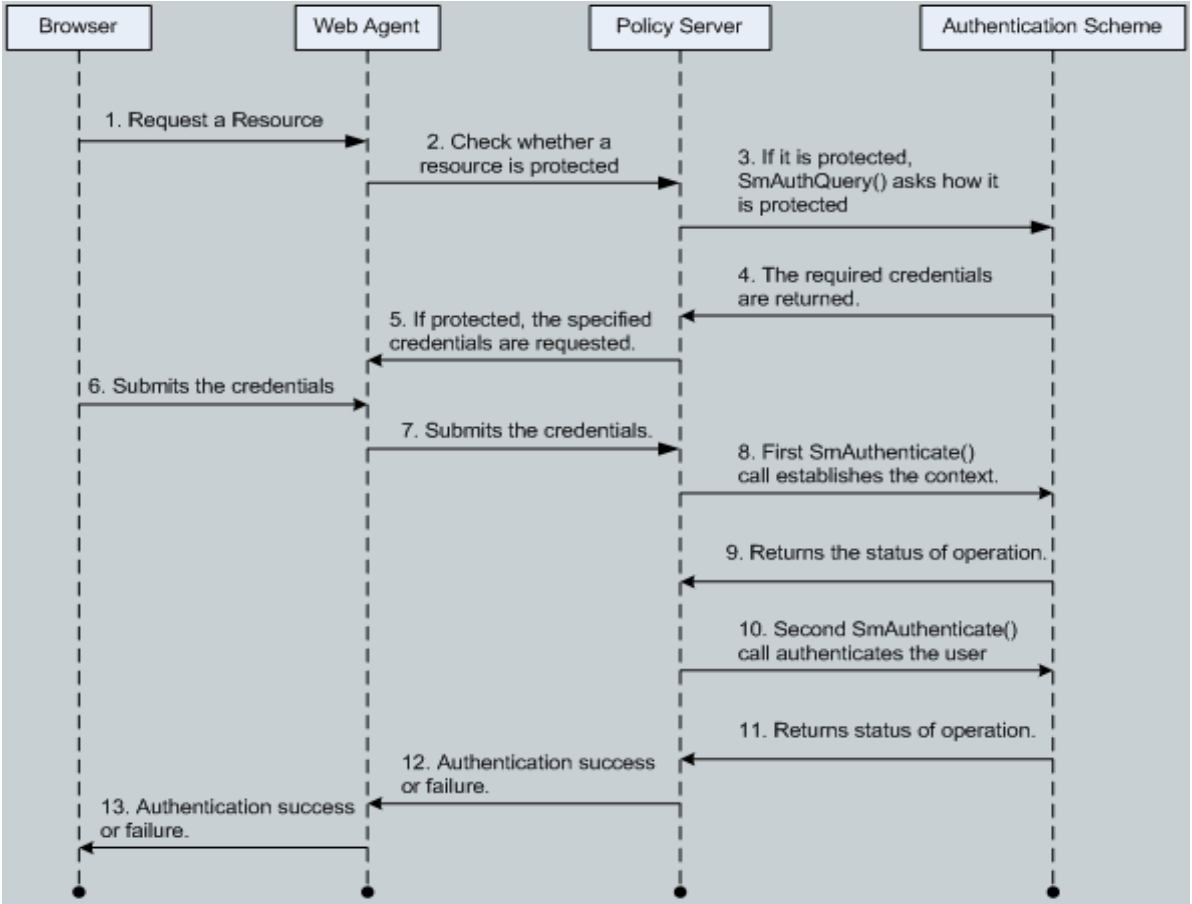

Figure 1-4 Authentication to Protected Resources

9.1.2.1 End User Authentication

End Users are authenticated to the TOE via basic HTTP over SSL, X.509 certificates and the Windows authentication scheme. Authentication is configured to run over a Secure Sockets Layer (SSL) connection. The TOE uses HTTP over SSL as a method of establishing an encrypted connection (using port 4443) between a client and a server using digital certificates to initiate the connection, and to establish a proof of identity. The following authentication scheme types are configured to use an SSL connection:

- Basic HTTPS
- X.509 Client Certificates

9.1.2.1.1 Basic Authentication Schemes

Basic authentication scheme is used to verify end user identities against the usernames and passwords that exist in a User Store. Basic authentication to a web resource identifies an end user based on a user name and password. The end user's identity is stored in the User Store. With a basic authentication scheme, the Policy Server locates the user's information based on the user name, then verifies that the password matches the one saved in the User Store. If the user name and password supplied by the end user match the data in the directory, SiteMinder authenticates the end user.

9.1.2.1.2 X.509 Certificates

SiteMinder supports the use of X.509 V3 client certificates with RSA signatures using SHA-256 encryption. Digital certificates act as cryptographic proof of an end user's identity. Once a certificate is installed on a client, that certificate is used to verify the identity of an end user who is accessing a resource. Certificate authentication uses SSL communication and is combined with basic authentication to provide an even higher level of access security.

9.1.2.1.3 Windows Authentications

Windows Authentication is a proprietary mechanism developed by Microsoft to validate end users in pure Windows environments. Windows Authentication enforces Single Sign-On by allowing Windows to gather end user credentials during the initial interactive desktop login process and subsequently transmitting that information to the security layer. SiteMinder, using the Windows Authentication scheme, secures resources by processing end user credentials obtained by the Microsoft Integrated Windows Authentication infrastructure. The Windows Authentication scheme allows SiteMinder to provide access control in deployments with Active Directories running in native mode. This scheme relies on a properly-configured IIS Web server to acquire and verify an end user's credentials. The Policy Server bases authorization decisions on the end user's identity on the response received by the IIS Web Server.

9.1.2.2 Administrator Authentication

Traffic between the WAM Administrative UI applet and the Policy Server is protected by exactly the same mechanism as is used for protecting the Web Agent-to-Policy Server traffic (see section 9.1.6 Encrypted Communications for more information). The only difference is that there is no mutual authentication between the Applet and the Policy Server. Instead, after the secured connection is established, an administrator has to send his username/password over this protected connection. Only after credentials of the administrator are verified, can he execute administrative actions.
Local administrators can launch the Policy Server Management Console locally but first need to authenticate to the account that was used to install the Policy Server. The account that was used for installation of the Policy Server is the account that will have access to the Policy Server (an OS account with administrative privilege on Windows or root account on Solaris OS). In this case the underlying OS provides identification and authentication of administrators who are able to login directly to the server OS hosting the Policy Server software. The Policy Server Management Console is a non-security relevant interface that will only be used during the installation and configuration of the TOE.

9.1.3 Security Management

9.1.3.1 Session Ticket Management

SiteMinder implements session management using Session Tickets. A Session Ticket contains basic information about an end user and that end user's authentication information; it is used to identify the end user's session across all sites in a single sign-on SiteMinder environment. Session Tickets are encrypted and can only be read or validated by the Policy Server. SiteMinder Web Agents use Session Tickets to identify end users and provide the Session Ticket to the Policy Server for authorizations. The Session Ticket is handled differently depending upon whether the session is persistent or non-persistent, however, only persistent sessions are included in this evaluation.
For a persistent session, the Web Agent places the Session Ticket in its Agent User Cache and, if possible, in an optional cookie on the client. If a cookie is written, no end user-specific data is kept in the cookie itself. The Session Ticket located in the cookie is used as an index into the Web Agent's cache to find the end user session data which is also stored there. This allows a Web Agent to validate an end user's session, enforcing the session timeouts, and enforce policy decisions which have previously been made by the Policy Server.

Management of the Session Ticket consists of revalidating a Session Ticket. An administrator can explicitly configure the Web Agent, working through the Policy Server. A session cookie stored in the end user's browser contains the session ID, which the Web Agent uses to reacquire the end user's information from the Agent User Cache. An administrator can also flush the Agent User Cache which would force an end user to re-authenticate.

9.1.3.2 Management of Security Attributes

SiteMinder provides authorized administrators with the management interfaces to manage the following security attributes.

Role	Attribute	Specific Attribute
End User	User name	SM_USER SM_USERNAME SM_USERLOGINNAME
	Password	SM_USERPASSWORD
	Session Ticket	SM_USERSESSIONSPEC
	Groups	SM_USERGROUPS
	Nested Groups	SM_USERNESTEDGROUPS
	Policies	SM_USERPOLICIES
	Responses for all Policies	SM_USERPRIVS
	Responses for all Rules under a Realm	SM_USERREALMPRIVS
	Authentication Level	SM_AUTHENTICATIONLEVEL
	Disabled User	SM_USERDISABLEDSTATE

Table 1-15 Security Attributes

- SM_USER - If the authentication method associated with a realm accessed by a user requires that the user provide a username, the Web Agent places the username in an SM_USER http header variable. In cases where the user does not provide a username, such as certificate-based authentication, the value of the SM_USER header variable is not set.
- SM_USERNAME - For an authenticated user, this attribute holds the user DN as disambiguated by SiteMinder. For an unauthenticated user, this attribute holds the user ID as specified by the user in the login attempt.
- SM_USERLOGINNAME - This attribute holds the user ID as specified by the user in the login attempt.
- SM_USERPASSWORD - This attribute holds the password as specified by the user in the login attempt. This attribute is only available after a successful authentication through the OnAuthAccept event. The value is returned only on authentication, not on authorization.
- SM_USERSESSIONSPEC - This attribute holds the user's session ticket. When user credentials are authenticated, the Policy Server compares the credentials to entries in a user directory. If the credentials match an entry, the Policy Server creates a session ticket and authenticates the user. SiteMinder confirms that a user's session ticket is valid instead of rechecking the user's credentials against a directory when an authenticated user makes additional requests.
- SM_USERGROUPS - This attribute holds the groups to which the user belongs. If the user belongs to a nested group, this attribute contains the group lowest in the hierarchy.
- SM_USERNESTEDGROUPS - This attribute holds the nested groups to which the user belongs. For only the group furthest down in the hierarchy, use SM_USERGROUPS. For example, if user JSmith belongs to the group Accounts Payable, which is contained in group Accounting, SM_USERNESTEDGROUPS contains Accounting and Accounts Payable. If the administrator wants only Accounting, use SM_USERGROUPS.
- SM_USERPOLICIES - When a user is authorized for a resource, this attribute holds the names of the policies that give the user authorization. For example, suppose that to purchase an item, the purchaser must be one of the users associated with the Buyer policy. If the Policy Server authorizes the purchaser to buy an item, then SM_USERPOLICIES will contain Buyer.
- SM_USERPRIVS - When a user is authenticated and is authorized for a resource, SM_USERPRIVS holds all of the response attributes for all policies that apply to that user, in all policy domains.

- SM_USERREALMPRIVS - When a user is authenticated or is authorized for a resource under a realm, SM_USERREALMPRIVS holds all the response attributes for all rules under that realm. For example, suppose that there is a realm called Equipment Purchasing. Under that realm, there is a CheckCredit rule. Associated with the CheckCredit rule is a response that returns the buyer's credit limit, such as limit = $15000, as a response attribute. If the buyer attempts to purchase equipment worth $5000, the CheckCredit rule applies. SM_USERREALMPRIVS would contain all of the response attributes for all of the rules under the Equipment Purchasing realm.
- SM_AUTHENTICATIONLEVEL - When a user is authenticated for a resource, this attribute holds an integer number (of 0 to 1000) that represents the protection level of the authentication scheme under which the user was authenticated.
- SM_USERDISABLEDSTATE - This attribute holds a decimal number that represents a bit mask of reasons that a user is disabled. The bits are defined in SmApi.h under the Sm_Api_DisabledReason_t data structure, which is part of the CA SiteMinder SDK. For example, a user may be disabled as a result of inactivity, Sm_Api_Disabled_Inactivity. In Sm_Api_DisabledReason_t, the reason Sm_Api_Disabled_Inactivity, corresponds to the value 0x00000004. So, in this case, SM_USERDISABLEDSTATE is 4. A user can be disabled for multiple reasons.

9.1.3.3 Managing the Password Policy

Within the Password Policy dialog, the number of failed logins for an end user is set. Once the end user has met the configured number of failed logins, the following will occur depending on how SiteMinder is configured:

- disable the End User account(s) until the account is reactivated by an authorized Administrator
- allow the End User one login attempt after a specified number of minutes
- re-enable the End User account after a specified number of minutes

The Incorrect Password group box is where an administrator can specify how many failed logins are allowed before an end user account is disabled. Additionally, an administrator can specify how long the account is disabled before an end user can attempt to log in again.

The "Account will be disabled after" field specifies the number of consecutive failed login attempts an end user can make before the end user account is disabled. Limiting the number of unsuccessful attempts protects against programs designed to access a resource by repeatedly trying passwords until the correct one is found. If an end user fails to login correctly after the specified number of attempts, the Policy Server disables the end user's account. The end user's account must be re-enabled by an administrator.

End Users are required to create and use strong/complex passwords as instructed by their organization's user guidance in accordance with the documented password policy. The organization's password policy can be configured by an administrator, and will be enforced by CA SiteMinder.

In the WAM Administrative UI, under the password policy properties dialog the password policy is configured. The Password Policy Dialog Fields and Controls are as follows:

- Name field - The name of the password policy.
- Description field - Optionally, a brief description of the password policy.
- The Password Policy dialog also contains the following tabs:
- General - Allows an administrator to bind the password policy to all or part of a user and or administrator directory, and enable the password policy.
- Expiration - Lets an administrator configure criteria for password expiration such as the maximum number of failed login attempts or the amount of time an account can be inactive before it is disabled.
- Composition - Enables an administrator to specify rules governing the composition of valid passwords such as minimum length and the minimum numbers that comprise a password.
- Regular Expressions - Allows an administrator to specify regular expressions that valid passwords must match or not match.
- Restrictions - Allows an administrator to set restrictions for reusing passwords, creating similar passwords, and using specific words.
- Advanced - Allows an administrator to specify the order in which password policies that apply to the same user directory or namespace should be evaluated and set password preprocessing options (for example, removing white space).

9.1.3.4 Static attribute initialization

CA SiteMinder enforces the CA SiteMinder Access Control Policy restrictive default values for security attributes that are used to enforce the SFP. CA SiteMinder allows per the CA SiteMinder Access Control Policy the Administrator to specify alternative initial values.

9.1.3.5 Management of TSF Data

9.1.3.5.1 Administrators

Through the WAM Administrative UI, the following must be done to create an administrator:

- Identify the administrator
- Identify the scope of an administrator's privileges
- Select the individual tasks the administrator is allowed to complete.

9.1.3.6 Web Agent Configuration

The Web Agent can be configured either centrally from the Policy Server or locally on the workstation where the Web Agent resides. In this evaluation, all agents will be centrally configured, which uses an Agent Configuration Object at the Policy Server to define the Agent's configuration.

9.1.3.7 Register Trusted Hosts

A registered host is a machine which has resources protected by Siteminder, or the machine that has the WAM Administrative UI component. You register a trusted host from the system where you install a Web Agent or the WAM Administrative UI; the host registration process is part of the installation and configuration process. After registration is complete, the registration tool creates the SmHost.conf file for the Web Agent or populates a local database with the Shared Secret. After this file is created or the database is populated successfully, the client computer becomes a trusted host. A trusted host must be registered to communicate with the Policy Server.

HostConfigFile (SmHost.conf) is a file that results from a successful registration of a trusted host computer with the Policy Server. Once a host is successfully registered, it is considered a trusted host. By default, all Web Agents on the computer share this file, and the trusted host registration must occur before an Web Agent can operate.

9.1.3.8 Policy Domains

A policy domain is a logical grouping of resources associated with one or more user directories. In addition, policy domains require one or more administrator accounts that can make changes to the objects within the policy domain. Policy domains contain realms, rules, responses, and policies (and optionally, rule groups and response groups). An administrator with the Manage System and Domain Objects privilege can grant control over a policy domain to other administrators.

In addition to acting as a container for domain objects, policy domains also connect to user directories. The Policy Server authenticates end users based on the requirements of the realm in which the target resource resides. In order to authenticate an end user, the Policy Server must find the User Store where an end user is defined. The Policy Server does this by locating the policy domain to which a realm belongs. From the policy domain, the Policy Server queries the User Store specified in the policy domain's search order.

9.1.3.9 Policies

A policy is created through the WAM Administrative UI and defines the rules and tasks which are performed by the domain administrator in the policy domain. Once a policy is created, end users or groups of end users are added to the policy, creating a policy binding between the selected end users and the policy. When an end user tries to access a protected resource, the policy verifies that the end user is part of its

policy binding, and then applies the rules included in the policy to see if the end user is allowed to access the resource.

9.1.3.10 Realms

In order to manage access control for a complex set of resources that are available on the Internet or corporate intranet, that set of resources must be logically grouped so that security policies are created. SiteMinder uses several object types to allow an administrator to group resources. When grouping resources in SiteMinder, the basic groupings for resources are realms.

A realm is a cluster of resources that reside in a common location on the network within a policy domain and are grouped together according to security requirements. A realm is configured by an administrator through the WAM Administrative UI using the Realm Dialog. Only administrators with the Manage System and Domain Objects privilege may create, edit, and delete realms. The contents of a realm are protected by SiteMinder Web Agents. When end users request resources within a realm, the associated Web Agent handles authentication and authorization of the end user. Each realm determines the authentication scheme required to access the resources within that realm.

9.1.3.11 Nested Realms

Nested realms are realms created within an existing realm. A nested realm has a parent, or top level realm, and is considered a child of the parent realm. Administrators with the Manage Domain Objects privilege may create, edit, and delete nested realms underneath existing realms in their policy domains.

Nested realms allow an administrator to increase the protection level of resources that are lower in a directory tree. The administrator can then assign an authentication scheme with a higher protection level to the nested realm. By default, to access resources in the child realm, an end user must be authorized for resources in the parent realm and for resources in the child realm. The administrator can globally change the default behavior of the Policy Server and always allow access to the resources in the child realm for end users who are authorized either for the parent realm or the child realm.

When an administrator creates nested realms, he can also create separate rules to protect the resources in the child realms. Administrators may also copy an existing rule, attach the rule to another realm, and rename the rule.

Responses can also be associated with rules. An administrator can associate a response or response group with a rule in a policy. When the rule is enforced, the associated response executes also.

9.1.3.12 Rules

The WAM Administrative UI allows an administrator to enable and disable rules. If a rule is enabled, SiteMinder automatically protects the resource(s) specified in the rule. If a rule is disabled, SiteMinder does not automatically protect the resource(s) specified in the rule. By default, when a rule is created, it is enabled. If a rule is enabled, no end user may access the protected resource(s) unless a policy that contains the rule has been created, and the end user attempting to access the rule is part of a group specified in the policy. To allow access to resources before a policy is put into place, an administrator can disable the rule.

If a rule is deleted, the rule is automatically removed from any policies that include the rule. The policies remain on the system. An administrator should verify that the policies function without the deleted rule.

9.1.3.13 Rule Groups

A rule group is a set of rules that are bound to SiteMinder policies. An administrator can use a rule group to combine groups of rules that apply to the same policy. For example, if an administrator has a number of rules that allow a GET action for different resources of a Web site, they could then create a rule group that contains all of the resources. When an administrator configures the policy that will include the rules, he can add a single rule group to the policy, rather than add all of the rules individually.

Through the WAM Administrative UI, an administrator can create, edit, and delete rule groups. In addition, the administrator can add a rule to or remove a rule from a rule group.

9.1.3.14 Response

A response is included in a policy and is an action that should take place when an end user accesses a specified resource within a realm. In creating a response, an administrator will select the Agent Type (Web, Affiliate, etc) as SiteMinder Web Agent indicating that the response passes information to a SiteMinder Web Agent. Each SiteMinder Response may contain one or more response attributes, which identify the pieces of information that the Policy Server passes to a SiteMinder Agent. Each SiteMinder Agent type can accept different response attributes.

Policies can include responses that allow or deny access to a resource, customize an end user's session time, redirect the end user to other resources, or customize the content the end user receives based on attributes contained in a User Store. In a policy, responses are bound to specific rules or rule groups. When a rule is enforced, the associated response returns information to a SiteMinder Web Agent. Responses take the form of name/value pairs. When a rule is triggered, the Policy Server returns the paired response to the SiteMinder Agent. For example, if an end user attempts to access a protected Web page, but is not authorized to view the contents of the page, a response can redirect the end user to an HTML page that indicates the end user doesn't have access, and provide details for contacting a system administrator. For Web Agents, SiteMinder adds response attributes to HTTP header variables or HTTP cookie variables so that the responses are available to the Web resource or application named in the rule.

- Web Agent Response Attributes
- WebAgent-OnAccept-Redirect
- WebAgent-OnAccept-Text
- WebAgent-OnAuthAccept-Session-Idle-Timeout
- WebAgent-OnAuthAccept-Session-Max-Timeout
- WebAgent-OnReject-Redirect
- WebAgent-OnReject-Text

An administrator can also delete a response from the WAM Administrative UI. When a response is deleted, it is removed from any policies that previously contained the response.

9.1.3.15 Response Groups

A response group is a collection of responses that are logically grouped so they are applied to a single rule within a policy. All relevant responses in a response group will apply when a rule paired with the response group applies. Response groups allow an administrator to combine multiple responses in a single object. When administrators creates policies, they can more easily associate multiple responses with a single rule within those policies.

An administrator can change all of the properties of a response group, except Agent Type. To change the Agent Type of a response group, an administrator must delete the response group from the WAM Administrative UI and create a new group of the appropriate Agent Type. However, once a response group is deleted, only the grouping is deleted, not the individual responses contained in the group.

9.1.3.16 Global Settings

Global policies allow an administrator to configure policies (and their associated rules and responses) as system level objects that are applied across all domains. Standard SiteMinder policies are created in the context of a single domain policy. However, large production environments may contain thousands of domains. In this type of environment it is useful to define types of behavior (represented by policies) that are common for many domains. Using standard policies, the same policy must be recreated for each domain that requires the same behavior.

9.1.3.17 Authentication Schemes

To identify an end user, SiteMinder employs authentication schemes. Authentication schemes provide a way to collect credentials and determine the identity of an end user. SiteMinder supports a variety of authentication schemes. These schemes range from basic user name/password authentication to digital certificate authentication. Simple schemes can be used for low risk web resources, while complex schemes may be employed to ensure added security for critical web resources. Authentication schemes for end users are configured using the WAM Administrative UI. During authentication, SiteMinder Web Agents communicate with the Policy Server to determine the proper credentials that must be retrieved from an end user who is requesting access to resources.

9.1.3.18 Password Policy

The TOE does not have a password policy associated with remote Super User or administrator accounts - no maximum/minimum length, no limitations on what characters need to/cannot be used, and no password history is maintained.

The TOE can enforce password policies (with complexity requirements) as configured by the respective authorized administrator of an organization for end users. The TOE by default does not enforce password policys or complexity requirements for end users until after a policy is created. To change passwords or force password changes, a password policy must be configured by an administrator through the WAM Administrative UI in the Password Policy Dialog Properties dialog. The User Store that contains end user information must be configured with the Password Data attribute.

The Manage User Accounts pane in the WAM Administrative UI enables the administrator to force password changes for end users . A password policy must exist before an administrator can force end users to change passwords. If no password policy exists, end users will not be able to change their passwords, and therefore will not be able to access protected resources. If an administrator forces an end user to change passwords, and the end user is accessing resources through a Web Agent that is not using a SSL connection, the end user's new password information will be received over the non-secure connection. To provide a secure change of passwords, the administrator should set up a password policy that redirects the end user over a SSL connection when changing passwords. In the evaluated configuration, all communication between the end user and the Web Agent will be encrypted with SSL 3.0. The TOE relies on the Operational Environment to supply the encryption over this path.

9.1.3.19 Advanced Password Options

Advanced password policy options allow an administrator to configure pre-processing of submitted passwords prior to validation and storage (for example, forcing all characters to be upper case) and to assign the policy a priority to allow predictable evaluation of multiple password policies that apply to the same User Store or namespace. An administrator uses the Advanced tab in the Password Policy Properties dialog to configure advanced password options through the WAM Administrative UI .
If the password policy is one of multiple policies that will apply to the same User Store or namespace, specify an Evaluation Priority (0-999; 999 is highest) for the password policy.

9.1.3.20 Force Password change

The User Management dialog box in the WAM Administrative UI enables an administrator to force password changes for end users, or change user passwords to new values. If the administrator is using the Password Must Change feature of SiteMinder's Registration Services, the password changes are forced from the User Management dialog box. However, a Password Policy must be defined.

9.1.3.21 Enabling and Disabling End Users

An administrator can enable and disable a SiteMinder end user account through the WAM Administrative UI. SiteMinder begins a user session after an end user logs in and is authenticated. SiteMinder stores end user attributes in its Agent User Cache located on the Web Agent machine's physical memory. When an administrator *disables* an end user, the Web Agent flushes the Agent User Cache, removing user identification and session information. When the end user attempts to access additional resources in the current session, the Web Agent will no longer has the user's data in its cache. The Agent contacts the Policy Server and attempts to re-authenticate the end user. The Policy Server determines that this end user is disabled in the User Store and rejects the Web Agent's request to authenticate, thereby ending the session.

9.1.3.22 Flush caches

The Cache Management options provide a method for administrators to flush the contents of all caches, or individual caches. Flushing all caches may adversely affect the performance of a Web site, since all requests immediately following the cache flush must retrieve information from the different User Stores and the Policy Store. However, this action may be necessary if critical user privileges and policy changes must go into effect immediately. Cache management features are only available to administrators who have either the Manage Users or Manage System and Domain Objects privileges. The Flush All button is only available for administrators with the Manage System and Domain Objects.

SiteMinder automatically flushes appropriate cache entries when administrators make changes to SiteMinder objects. The cache settings also specify a regular interval for applying administrative changes. However, if an administrator is making very sensitive changes (for example, changing the access rights to highly critical information), he may want to flush SiteMinder caches manually, so that unauthorized users will not be able to access protected resources based on information stored in the caches.

Cache Management features are accessible from the Policy Server Global Tools pane in the WAM Administrative UI. They let administrators force an update of SiteMinder data by manually flushing the following caches: All Caches, User Session Caches, and Resource Caches. All Caches enables the administrator to flush all caches, including user sessions, resource information, and user directory caches (including certificate CRLs). User Session Caches enables the administrator to force end users to reauthenticate when they try to access protected resources. When an administrator disables an end user, the Web Agent flushes the session cache, removing user identification and session information. When the end user attempts to access additional resources in the current session, the Web Agent no longer has the user's data in its cache. Resource Caches enables the administrator to flush cached information about resources.

9.1.3.23 Management of TSF Data for the Operational Environment

The management of TSF data is done either through the Policy Server Management Console or the WAM Administrative UI. The use of Policy Server Management Console method to provide this ability relies on the identification and authentication of users at the OS.

9.1.3.24 Super user Account

There are two Super User accounts created during the installation of the TOE. A local Super User account which is used during installation and configuration and a remote Super User account which is used in the evaluated configuration of the TOE. The local Super User is the Policy Server administrator "root" account on Unix or administrator account on Windows that is established automatically by the Policy Server installation process. This account has the same attributes as the administrator account, but with unrestricted access. The remote Super User is the administrator that can access the TOE via the WAM Administrative UI. The remote Super User account is part of the TOE and is used to create additional administrator accounts and also assigns the categories, rights, and scope of those accounts. It is also possible to setup separate super user accounts for each domain, or across domains, wherein that administrator has full privileges within the domain(s) to which he's assigned. The Super User password can be changed from the WAM Administrative UI interface. It is not recommended that the Super User account be used for day-to-day administration.

9.1.3.25 Security roles

The TOE maintains the roles of Super User, Administrator and End User. Everyone who has access to Policy Server objects and tools is considered an administrator of the Policy Server. The Administrators are configured to have access rights to different domains, depending on their responsibilities.

When the WAM Administrative UI is installed, the installation sets up a default administrator account automatically. This account called the remote Super User account has maximum privileges, and with it additional administrator accounts are created for those who need to add or make changes to Policy Server objects. When the Policy Server is installed, the installation sets up a default administrator account automatically. This account called the local Super User and is the administrator of the SiteMinder Management Console. The local Super User account is only used to install and configure the TOE and will not be used in the evaluated configuration.
The End User Role refers to the end users that are accessing protected resources via the Web Agents.

9.1.3.26 External LDAP User Directories

The Policy Server is configured to communicate with an external User Store that uses the Lightweight Data Access Protocol (LDAP). Through the WAM Administrative UI, Administrators can connect to an external LDAP User Store, view the contents of the directory, and search for a particular user. The Policy Server pulls in the external user information and stores it in its own configurable User Authorization Cache (memory). This cache stores external user information after the login step. The Policy Server uses this information, together with the objects stored in the Policy Store (realms, rules, etc) in order to allow or restrict an end user's access to a protected resource.

9.1.3.26.1 General Information about LDAP

LDAP user directories are created with an inverted tree structure. Due to this hierarchical structure, LDAP-enabled directories can contain multiple user namespaces. A namespace is a grouping of entities under a node in the LDAP Directory Information Tree (DIT). Any branch of an LDAP DIT can be defined in a user directory connection as a separate namespace. Typically, user directory connections are configured for DIT branches that represent an organization (o) or an organizational unit (ou). Users and user groups are located under an o= or ou= node in the directory structure. Any node in an LDAP tree is identified by its distinguished name (DN), which is made up of a comma separated list of its own name and the names of the nodes above it in the directory tree. This method of naming allows each point in the user directory to have a unique DN.

9.1.3.26.2 User Disambiguation in an External LDAP Directory

User disambiguation is the process of locating a unique user in a User Store. The Policy Server uses information supplied in the User Lookup group box of the User Directory pane, and a user-supplied value, such as login name, to locate a user.

There are two methods of locating users in a User Store. Users can be located by the following:

- DN - A user lookup by DN is constructed from the User Directory pane in the User Lookup group box of the LDAP Settings area. The value specified in the User Lookup Start field, the username as specified by the user during login, and the value specified in the User Lookup End field are concatenated.

- Search expression - An LDAP directory server may contain numerous users in complicated DITs, and it may not be practical to create a large number of user directory connections. Instead, one user directory connection pointing to a common root can be created with the User DN Lookup Start and User DN Lookup End properties defining an LDAP search expression. The result of the search expression is a list of user DNs for the Policy Server to try during authentication.

9.1.3.26.3 Directory Attributes Overview

Some SiteMinder features require read or read/write access to seven SiteMinder attributes whose values are stored in the external LDAP User Store connected to the Policy Server. When a connection from the Policy Server to a User Store is configured, the names of the user attributes in that User Store that correspond to the seven SiteMinder attributes must be specified (see Table 1-16 External LDAP Directory Attributes). This is done in the fields on the User Attributes group box. For example, the name of the Universal ID might be Student ID in one User Store, while in another User Store, the name of the Universal ID might be Social Security Number. Once the User Store connections are configured, SiteMinder can access the correct user attribute in the selected User Store each time that it encounters the Universal ID.

This capability of SiteMinder can be extended through user attribute mapping. User attribute mapping allows for the definition of common names, mapping each one to user attribute names in multiple user directories with different underlying schema.

Each SiteMinder attribute is associated with a data type and one or more directory types and is described in the table. (R) indicates that the attribute requires read access. (RW) indicates that the attribute requires read/write access.

SiteMinder Attribute	Data Type	Directory Type	Description
Universal ID (R)	string	LDAP	Specifies the universal ID or user identifier that SiteMinder passes to protected applications to maintain a user's identity. This feature is a bridge between SiteMinder and legacy applications, which often use attributes, such as Social Security Number, to identify a user. The universal ID is also used in configuring Directory mapping.
Disabled Flag (RW)	string	LDAP	Specifies the user's account status.
Password Attribute RW)	binary	LDAP	Specifies the user's password.
Password Data (RW)	binary	LDAP	Is used to track password policy information
Anonymous ID (RW)	string	LDAP	Stores the DN of users who are authenticated using an anonymous authentication scheme.
Challenge/Response (RW)	string	LDAP	Specifies the question and answer pair that is used by the Forgotten Password feature in Password Services and DMS. The Challenge string is the password hint that is passed to the user.

Table 1-16 External LDAP Directory Attributes

9.1.4 Audit

Auditing allows the administrators to track end user and administrative activity as well as analyze and correct security events and anomalies. Objects, events, and activities to be audited are defined by the administrator. At a minimum, the audit record for end user and administrator actions on the TOE stores the date and time the record was created, the remote server host name and ID, the remote server account name responsible for the report creation, the object/resource scanned, status, and the total number of objects/resources scanned. In the evaluated configuration, the date, time, type of event, subject identity, and outcome (success or failure) is logged for each audit event, which includes the startup and shutdown of audit functions. Based on the content of these logs, the TOE is able to associate the event with the user that caused the event.

The audit logs are stored in local audit files in the Operational Environment separate from the Policy Server. By storing the logs separately from the Policy Server, the Operational Environment is able to protect the audit records from unauthorized deletion and protect unauthorized modifications to the records in the audit trail. Both the Policy Server and Web Agent provide separate audit logging. All audit logging relies on the underlying operating system in the Operational Environment to provide reliable time stamps.

The Web Agent uses an auditing feature which allows the administrator to track and log successful authorizations of an end user, these audited events are stored in a local audit log file called the Trace Log. This allows the administrator to track user and role activity, and to measure how often applications on a particular web site are being used. For an administrator to view these logs, they must authenticate to the OS which the Web Agent and Web Server are installed on and have equal or higher privileges then the account used to install the TOE components.

Table 1-18 describes the administrator events that are audited by SiteMinder.

Operations	Policy Server Objects
Create/edit/delete	Agents
Create/edit/delete	Agent Configuration Objects
Create/edit/delete	Agent groups
Create/edit/delete	Host Configuration Objects
Create/edit/delete	policy domains
Create/edit/delete	authentication schemes,
Create/edit/delete	certificate mappings
Create/delete	parent realms in all domains.
Create/edit/delete	administrators.
Flush	all caches, including cached resources.
Change	global settings.
All privileges for Manage Domain Objects listed below.	

Operations	Policy Server Objects
create/edit/delete	rules (in managed domains)
create/edit/delete	rule groups (in managed domains)
create/edit/delete	responses (in managed domains)
create/edit/delete	response groups (in managed domains)
create/edit/delete	policies (in managed domains)
Edit	top-level realms (in managed domains (not resource filters)).
Create/edit/delete	nested realms (in managed domains)
Flush	specific realms from the resource cache, and flush all resources (in privileged domains) from the cache.
Create/edit/delete	password policies
Manage	keys
Create/edit/delete	password policies for end users in directories attached to managed domains.
Flush	all end user session caches, or flush the end user session cache of any individual end user cache from any directory.
Enable/disable	End users in any directory.
Force password change	on any end user in any directory.
Change password	On any end user in any directory
Flush session caches	for individual end users in directories attached to managed domains.
Enable/disable	End users in directories attached to managed domains.
Force password change	on end users in directories attached to managed domains.
Change password	On any end user in any directory

Table 1-18 SiteMinder Auditable Events

9.1.4.1 EnableAuditing Parameter

The administrator can measure how often applications on a website are used or track user activities with the EnableAuditing parameter. This parameter specifies whether the Web Agent logs all successful authorizations. User authorizations are logged even when the Web Agent uses information from its cache instead of contacting the Policy Server. Web Agents log user names and access information in a local file (called the Trace Log), and sends the information to the Policy Server to log as well when users access resources.

9.1.4.2 Transaction ID

The Web Agent generates a unique transaction ID for each successful user authorization request. The Web Agent adds the Transaction ID to the HTTP header. The Transaction ID is recorded in the Audit Log, and the Policy Server Log. Activities for a particular application are tracked using the Transaction ID.

9.1.4.3 Web Agent Error Logging

The Web Agent logging function is used to monitor the performance of the Web Agent and its communication with the Policy Server. The logging feature provides accurate and comprehensive information about the operation of SiteMinder processes to analyze performance and troubleshoot issues, which includes the startup and shutdown of the Web Agent and its auditing functions. IIS 6.0 Web Agents create log files only after the first user request is submitted. ASF Apache 2.2 Web Agents create log files when the Apache server starts.

The administrator uses these logs to help solve any issues that may prevent the Web Agent from operating properly. For Windows platforms, the EnableWebAgent parameter must be set to yes to ensure that the Web Agent log gets created. If the EnableWebAgent is set to no (the default) and the logging parameters are set, the Web Agent log gets created only for Agents on UNIX platforms.

9.1.4.4 Policy Server Logging

The Policy Server generates log files that contain information about the status of the Policy server and configurable levels of auditing information about authentication, authorization, and other events in the Policy Server log file. Policy Server logging is configured under the Logs tab of the Policy Server Management Console. The Policy Server Log group box is used to specify where the settings for the Policy Server log are specified. This log records information about the status of the Policy Server and its processes. The Policy Server Audit Log Group Box is used to specify the types of auditing events that should be included in the Policy Server log:

- **Authentication Events drop down list** - Specifies which client authentication events the Policy Server should log (Log All Events, Log Rejection Events only, or Log No Events).
- **Authorization Events drop down list** - Specifies which client authorization events the Policy Server should log (Log All Events, Log Rejection Events only, or Log No Events).
- **Affiliate Events drop down list**—Specifies which affiliate events the Policy Server should log (Log All Events, or Log No Events).
- **Administrator Access Events drop down list** -Specifies which administrator access events the Policy Server should log (Log All Events, Log Rejection Events only, or Log No Events).
- **Administrator Changes to Policy Store Objects drop down list** - Specifies which administrator changes to Policy Store objects the Policy Server should log (Log All Events, or Log No Events).

The Policy Server can be configured to log information about exceptions that might occur while preparing or executing audit logs to the Windows event log viewer or UNIX syslog file.

The TOE calls object events when objects are created, updated, or deleted. Any exceptions occurred while preparing/executing the product are logged to the Windows event viewer and UNIX syslogs under the 'ObjAuditLog' category. Access events result from end user and administrator-related activities which are called in the context of authentication, authorization, administration, and affiliate activity. Any exceptions that occur while preparing/executing SiteMinder access audit logs will be logged to the Windows event viewer and UNIX syslogs under the 'AccessAuditLog' category.

9.1.5 Load Balancing and Failover

Load balancing and failover in a SiteMinder deployment provide a high level of system availability and improve response time by distributing requests from SiteMinder Agents to Policy Servers. Defining clusters in combination with load balancing and failover further enhances the level of system availability and system response time. A cluster is defined as a set of one or more Policy Servers grouped according to customer-defined criteria, and with load-balancing between the servers. Each Policy Server is completely independent of the other Policy Servers within its cluster and within other clusters.
All load balancing is done via the Web Agent API Layer logic. The Web Agent API Layer is responsible for dynamically balancing the load between Policy Servers in a cluster based on server response time, and for failing-over to another cluster under the failover criteria. For example, if the response time for one Policy Server within a cluster is too slow, the Web Agent API Layer will defer the request to the another Policy Server within that cluster. If the entire primary cluster fails and failover is enabled, a backup cluster takes over policy operations.
In any clustered SiteMinder environment, the default failover threshold is 0 (i.e. if the threshold is not defined, all of the Policy Servers in a cluster must fail before requests are redirected to the next cluster). When the number of available Policy Servers falls below the specified threshold, all requests that would otherwise be serviced by the failed Policy Server cluster are forwarded to another cluster. The failover threshold is represented by a percentage of the Policy Servers in a cluster. For example, if a cluster consists of four Policy Servers and the failover threshold for the cluster is set at 50%, when three of the four Policy Servers in the cluster fail, the cluster fails, and all requests fail-over to the next cluster. In the evaluated configuration, the default failover threshold is zero, which means that all servers in a cluster must fail before failover occurs.
For further understanding of failover, refer to the figure in the TOE description which illustrates a deployment using two clusters.

9.1.6 Encrypted Communications

A trusted path is established for all communications and interactions with the TOE to ensure that all traffic communications to and from the TOE are protected from unauthorized disclosure. The TOE relies on the Operational Environment to provide all protected communications from the TOE to a user of the TOE. In the evaluated configuration the Operational Environment will be configured to have SSL 3.0 encryption over these paths.

Communications between the end user and the SiteMinder Web Agent are done through HTTP over SSL v3.0 with SHA-256 on an ASF Apache 2.2 or SunONE 6.0 SP2 web server, or SHA-1 on an IIS 6.0 server.

Communications between the administrator and the SiteMinder Policy Server WAM Admin UI are also done through HTTP over SSL v3.0 with SHA-256 on an ASF Apache 2.2 or SunONE 6.0 SP2 web server, or SHA-1 on an IIS 6.0 server.

In the case of the Apache web server interface, the end user initiates authentication to the web server components of the TOE using digest authentication from Apache, specifically the Apache module mod_auth_digest controls the encryption of the passwords, and protects the TOE from replay attacks. During the I&A process, the end user performs the SSL protocol handshake, is prompted with a login pop up window, and is allowed to enter I&A credentials. Note that through this interface an end user is authenticated as part of the SSL protocol handshake using an RSA key pair.

The TOE also establishes a trusted channel for all communications and interactions between TOE components. Communications between the Policy Server and the Web Agent/WAM Administrative UI use AES in OFB mode with HMAC-SHA-256 with key sizes of 128 bits. The process of establishing this channel is a proprietary method known as the TLI handshake.

- AES (Advanced Encryption Standard) is an encryption algorithm approved by FIPS 140 and defined in FIPS 197. AES is a secret-key cipher that enciphers data in blocks of 128 bits and whose key size may be 128, 192, or 256 bits. In the evaluated configuration, AES is used by SiteMinder with 128-bit HMAC keys for the AES-encrypted TCP/IP connections, for message integrity and for key agreement of the session keys' key-wrapping keys.

9.1.7 Encrypted Data

AES is used with HMAC-SHA256 to encrypt Single-sign on (SSO) tokens, to protect generated URLs from user modification, and to protect Password Services data in user stores and sensitive data (keys, shared secrets, passwords) in Policy Stores. AES is also used to protect the session keys which the Policy Server sends to SiteMinder agents for their protected TCP/IP message exchange. The aforementioned encrypted data is Base64-encoded after encryption.

9.1.7.1 Session Ticket Key

Session tickets, administrator tokens, and Password Services state data are encrypted using AES in CBC mode encryption with an HMAC-SHA224 digest with 128-bit keys. This encryption is done through the use of an AES key derived from the Session Ticket Key and can only be read or validated by the Policy Server.

Session Ticket Keys are created by the Policy Server using the AES Key Wrap algorithm and generate 192 bit keys. When the Policy Server needs to encrypt or decrypt Session Tickets, administrator tokens, or Password Services state data, the Policy Server will take the Session Ticket Key and place it through the FIPS-140 Key Expansion Algorithm generating 128-key which will be used to encrypt or decrypt the data.

A session ticket contains basic information about a user and that user's authentication information; it is used to identify the user's session across all sites in a single sign-on SiteMinder environment. The session ticket is what the Policy Server uses to determine how long a user's authentication remains valid. SiteMinder Web Agents use session tickets to identify users and provide session information to the Policy Server. An administrator token contains the administrator's distinguished name which is used by the Policy Server to determine the administrator's scope and tasks that they are authorized to perform actions in. The Password Services state data is subset of the information that is maintained in the User Store.

9.1.7.2 Cookies

Cookies are encrypted by the Web Agent using an AES key derived from an Agent Key that is retrieved from the Policy Server. Agent Keys are created by the Policy Server using the AES Key Wrap algorithm and generate 192 bit keys, which are then distributed to the Web Agents. When the Web Agent needs to encrypt or decrypt cookies the Web Agent will take the Agent Key and place it through the FIPS-140 Key Expansion Algorithm generating 128-key which will be used to encrypt or decrypt the data.

There are three major types of cookies that may be sent with every request, including SMSESSION, SMIDENTITY, and SMDATA (there are other cookies that are used only during authentication, like FORMCRED cookies). All three types are encrypted using AES and HMAC-SHA256. In FIPS mode, SiteMinder uses AES encryption in CBC mode with 128-bit keys.

- **SMSESSION** - Session ticket cookie, always present.
- **SMIDENTITY** - identity cookie. Present only if the "User Tracking" option is configured through the Policy Server Global Tools task.
- **SMDATA** - cookie that keeps user credentials. Present only if the "Allow Form Authentication Scheme to Save Credentials" option is configured through the Policy Server's Infrastructure Authentication task.

The SMSESSION cookie is created for each successful authentication. By default the SMSESSION cookie is transient. However, it is configured to be persistent. Here is the format of the SMSESISON cookie:

Field Name	Meaning
random	First field is a form of "random1=random2" where random 1 and random2 are random numbers converted into 8 bytes strings.
VER	Version of the SMSESSION cookie.
NAME	The username from the user's credentials
DN	The Distinguished Name of the User
IPADDRESS	The IP address of the remote client
SRVSESSIONID	The session ID as received from the Server via the Login call
SAVEID	Used to preserve SessionID while crossing realms that require re-authentication
SRVSESSIONSPEC	The session spec as received from the Policy Server
AUTHTIME	Time of successful login
LASTACCESSTIME	Last time the WebAgent saw the cookie
MAXTIMEOUT	Session timeout
IDLETIMEOUT	Session idle timeout
TARGETH	Target Host, prevents cookies from being used in a different cookie domain
IMPERSONATE	Use for WINNT impersonation

Table 1-169 SMSESSION Cookie

The SMIDENTITY cookie is created for each successful authentication if the user-tracking option is configured through the Policy Server Global Tools task. By default SMIDENTITY cookie is persistent. However, it can be configured to be transient. Here is the format of the SMIDENTITY cookie:

Field Name	Meaning
random	First field is a form of "random1=random2" where random 1 and random2 are random numbers converted into 8 bytes strings.
VER	Version of the SMIDENTITY cookie.
NAME	The username from the user's credentials
DN	The Distinguished Name of the User
ISGUID	Set to 1 if user DN is a GUID (used for tracking anonymous users)
IDTICKET	The identity spec as received from the Policy Server

Table 1-20 SMIDENTITY Cookie

SMDATA is a persistent cookie that is created once after the first successful form-based authentication if "allow to save credentials" option is configured through the forms authentication scheme.

Field Name	Meaning
NAME	Username
PASSWORD	Concatenation of the Password field and other fields used by the authentication form

Table 1-21 SMDATA Cookie

9.1.7.3 Agent Keys

SiteMinder Web Agents use a key to encrypt and decrypt any cookies that pass between Web Agents in a SiteMinder environment. All keys must be set to the same value for all Web Agents communicating with a Policy Server.

Agent keys are AES keys and are encrypted using AES Key Wrap algorithm. They are stored in the Key Store in encrypted form. They are encrypted using the 128-bit Policy Store/Key Store Key with the AES Key Wrap algorithm. The Policy Store/Key Store Key is created during installation of the TOE and is itself stored in encrypted form – the Policy Store Key in a system local file, the Key Store Key encrypted in the Policy Store. Agent Shared Secrets (used for Agent authentication and in the TLI Handshake), along with other sensitive data, are also encrypted with the Policy Store Key and stored in the Policy Store.

Agent Keys are used to encrypt SiteMinder cookies that may be read by all agents in a single sign-on environment, and are shared by all Web Agents in a single sign-on environment, since each Web Agent must be able to decrypt cookies encrypted by the other agents. Agent Keys are managed by the Policy Server, and distributed to Web Agents periodically.

The Web Agent protects the following data:

- Data it sends to a Policy Server – uses AES encryption with 128-bit keys.
- Session and Identity cookies it sends to a user browser – uses AES and HMAC-SHA256 with 128-bit keys.
- The shared secret used to identify the Agent/host – uses AES encryption with 128-bit keys.

9.1.8 TOE Protection

9.1.8.1 TP-1 TSF domain separation

The Policy Server maintains individual sessions associated with administrators. The TSF maintains a session ID as part of a session to prevent interference between administrator actions.

The administrator sessions are not locked. CA SiteMinder allows multiple administrators to make changes simultaneously. The configuration changes last saved by any session are enforced.

The TSF when invoked by the underlying host OS maintains a security domain that protects it from interference and tampering by untrusted subjects in the TOE's Scope of Control.

The Policy Server and Web Agents are passive devices in that they indirectly connect to networks via other devices e.g. network interface. The Policy Server and Web Agent's protected domain includes the Policy Server and Web Agent software.

In addition to the Policy Server and Web Agent specific software, other software files such as configuration files and audit data are also stored on disk. These files are modified by an authorized Administrator accessing them through the Policy Server Management Console, and directly through the OS. In addition, the Policy Server and Web Agent rely on the OS for file process separation. These files are modified by an authorized user accessing them through the Policy Server Management Console, and directly through the OS. Access Permissions are enforced to provide protection from unauthorized users accessing these files. The underlying assumption regarding the operation of the Policy Server and Web Agent are that they are maintained in a physically secure environment.

9.1.8.2 TOE Protection by the Operational Environment

The encryption keys are considered inside the TOE Boundary. The Policy Server and Agents use encryption keys to encrypt and decrypt sensitive data passed between Policy Servers and Agents in a SiteMinder environment.
Agent Keys are managed by the Policy Server and are distributed to agents periodically. Since each Web Agent must be able to decrypt cookies that are encrypted by the other agents, the cookies may be shared by all agents in a single sign-on environment.
AES keys derived from session ticket keys are used by the Policy Server to encrypt session tickets, administrator tokens, and Password Services state data. Session tickets contain credentials and other information relating to a session (including user credentials). Agents embed session tickets in SiteMinder cookies, but cannot access the contents since they do not have access to session ticket keys which never leave the Policy Server. An administrator token contains the administrator's distinguished name which is used by the Policy Server to determine the administrator's scope and tasks that they are authorized to perform actions in.
Both types of keys are kept in the Policy Server's Key Store.

9.1.8.3 Storage of Keys

The TOE protects all keys by storing them in a Key Store. The Key Store is a repository from which all Policy Servers retrieve the most current keys and their respective information and distributes them to the TOE as needed. By default, the Key Store is part of the Policy Store but a separate key store database is created if desired, as done in the evaluated configuration.

9.1.8.4 Agent Key

Web Agents use an AES key derived from Agent Keys to encrypt and decrypt SiteMinder cookies so the data contained in the cookies is only readable to Web Agents. The Web Agents use the keys to encrypt cookies before sending them to an end user's browser and to decrypt cookies received from the user's browser, even when they were encrypted by other Web Agents. All Web Agents need to be aware of the same keys, and the keys must be set to the same value for all Web Agents communicating with a Policy Server. This rule is particularly important for Web Agents in a single sign-on environment. When a Web Agent starts up and makes a management call request, the Policy Server supplies the current set of keys. A Web Agent makes a management call every time it polls the Policy Server. The Web Agent receives the updated keys.
The Policy Server provides two types of keys:
Dynamic Keys - A dynamic key (Agent Key) is generated by a Policy Server algorithm and distributed to other connected Policy Servers and their associated Web Agents. Dynamic keys can be rolled over at a regular interval, or by using the Key Management dialog box of the WAM Administrative UI.
Static Keys - A static key remains the same indefinitely, and is generated by a Policy Server algorithm or configured manually. SiteMinder uses this type of key for a subset of features that require information to be stored in cookies over extended periods of time.

9.1.8.5 Key Rollover

To ensure that the keys remain secure, the Policy Server performs a key rollover for both Agent Keys and Session Ticket Keys. A key rollover is the process of generating new keys, encrypting them, and in the case of Agent Keys distributing them to all Web Agents within a SiteMinder environment. Automated key rollover enforces TOE protection by ensuring the integrity of keys and simplifying key management particularly in large SiteMinder installations that share a single key store. Dynamic key rollover is configured by using the SiteMinder Key Management dialog box of the Policy Server Management Console, while manual Key rollover is performed through the WAM Administrative UI.

If Agent Keys have been updated, Web Agents pick up the changes during polling. The default polling time is 30 seconds, but are configured by changing the PSPollInterval parameter of a Web Agent. When the Policy Server processes a dynamic Agent Key rollover, the value of the current key replaces the value of the old key. The value of the future key replaces the value of the current key, and the Policy Server generates a new value for the future key.

When a Web Agent detects that a key rollover has occurred, the Agent retrieves new values for the following Agent keys:
Old Key - Last value used for the dynamic Agent key before the current value.
Current Key - Value of the current dynamic Agent key.
Future Key - Next value that will be used as the *current key* in a dynamic Agent key rollover.
Static Key - A long-term key that the Web Agent can use for SiteMinder features that need to identify an end user and maintain this information for long periods of time. Static keys also support cookie encryption for single sign-on when dynamic keys are not enabled. Web Agents require multiple keys to preserve cookie data and ensure a smooth transition between old keys and new keys.

During key rollover both the Policy Server (both types of keys) and Web Server (Agent Key only) store the new key(s) through the process of zeroization of the previous key(s) and then writing the new key(s) over the same location.

9.2 TOE Summary Specification Rationale

This section identifies the security functions provided by the TOE mapped to the security functional requirement components contained in this ST. This mapping is provided in the following table.

Security Function	Security Functional Components
Security Audit	FAU_GEN.1 Audit data generation
	FAU_GEN.2 User identity association
	FAU_SAR_EXT.1 Audit review
	FAU_STG_EXT.1 Protected Audit Trail Storage
Cryptographic Support	FCS_CKM.1(1) Cryptographic key generation
	FCS_CKM.1(2) Cryptographic key generation
	FCS_CKM.1(3) Cryptographic key generation
	FCS_CKM_EXT.1 Cryptographic key generation
	FCS_CKM.4 Cryptographic key destruction
	FCS_CKM_EXT.4 Cryptographic key destruction
	FCS_COP.1(1) Cryptographic operation
	FCS_COP.1(2) Cryptographic operation
	FCS_COP.1(3) Cryptographic operation
	FCS_COP.1(4) Cryptographic operation
	FCS_COP_EXT.1 Cryptographic operation
User Data Protection	FDP_ACC.1(1) Subset access control
	FDP_ACC.1(2) Subset access control
	FDP_ACF.1(1) Security attribute based access control
	FDP_ACF.1(2) Security attribute based access control
Identification and Authentication	FIA_AFL.1 Authentication and failure handling
	FIA_ATD.1 User attribute definition

Security Function	Security Functional Components
	FIA_SOS.1 Verification of Secret
	FIA_UAU.1 Timing of authentication
	FIA_UAU_EXT.5(1) Multiple authentication schemes
	FIA_UAU_EXT.5(2) Windows authentication scheme
	FIA_UAU.6 Re-authenticating
	FIA_UID.2 User identification before any action
	FIA_UID_EXT.2 User identification before any action
Security Management	FMT_MSA.1(1) Management of security attributes
	FMT_MSA.1(2) Management of security attributes
	FMT_MSA.1(3) Management of security attributes
	FMT_MSA.2 Secure security attributes
	FMT_MSA.3 Static attribute initialization
	FMT_SMR.1 Security management roles
	FMT_SMF.1 Specification of management functions
Protection of the TSF	FPT_FLS.1 Failure with preservation of secure state
	FPT_STM_EXT.1 Reliable Time Stamps
Resource Utilization	FRU_FLT.1 Degraded fault tolerance
Trusted Path/Channels	FTP_TRP_EXT.1 Trusted path
	FTP_ITC.1 Inter-TSF trusted channel

Table 1-172 Security Functional Components

9.2.1 User Data Protection

The User Data Protection function of the TOE enforces the FDP_ACC.1(1), FDP_ACC.1(2), FDP_ACF.1(1), and FDP_ACF.1(2) requirements.

When an end user attempts to access a protected resource within a realm, SiteMinder's Policy Server uses the authentication scheme associated with the resource's realm to determine how to identify the end user. The Administrator Policy is enforced to manage administrators within a domain. The Policy is enforced by the TSF on end users who attempt to access protected resources via the Web Agent (where GET, PUT, POST are explicitly defined during the creation of the policy). The TSF enforces the Domain Policy to objects based on username, groups, resources, and realm when an end user attempts to access a protected resource within a realm. If an end user supplies the correct username or is on the group list for access to the resource/realm, than he is granted access to the resource. Components of the TOE interact to enforce access control. The Web Agent intercepts user requests for resources and checks with the Policy Server to see if the requested resource is protected. If the resource is unprotected, the access request proceeds directly to the web server. If the resource is protected, the Policy Server checks which authentication method is required for that particular resource, the Web Agent then challenges the end user for those credentials, and passes the credentials to the Policy Server for verification. Once the Policy Server authenticates the end user, it grants access to the resource and the Web Agent allows the request to proceed to the web server.

9.2.2 Identification and Authentication

The identification and authentication function of the TOE enforces the FIA_ AFL.1, FIA_ATD.1, FIA_UAU.1, FIA_UAU_EXP.5(1), FIA_UAU_EXP.5(2), FIA_UAU.6, FIA_UID.2, FIA_SOS.1 and FIA_UID_EXP.2 requirements.

The TOE provides user identification, authentication and authorization through the use of user names and passwords for Administrators and end users. Endusers have to identify and authenticate themselves before being allowed access to protected resources. Administrators have to identify and authenticate themselves before gaining access to the WAM Administrative UI. Authentication schemes for end users must be configured using the WAM Administrative UI. During end user authentication, SiteMinder Web Agents communicate with the Policy Server to determine the proper credentials that must be retrieved from an end user who is requesting access to protected resources within a realm. When an end user attempts to access a protected web resource, the Policy Server uses the authentication scheme associated with the resource's realm to determine how to authenticate and authorize the end user. The authentication scheme specifies the credentials that the end user must supply for authentication, as well as the method used by the Policy Server to authenticate the end user's identity. Authenticated end users of one realm can access a resource in another realm without re-authenticating as long as the second realm is protected by an authentication scheme with an equal or lower protection level. If an end user tries to access a resource protected by an authentication scheme with a higher protection level, SiteMinder prompts the user to reauthenticate by entering the credentials required by the authentication scheme. Three types of authentication schemes are supported by the evaluated configuration: Basic Authentication, Windows Authentication Scheme, and X.509 certificates.

The administrator defines, through the password policy, the number of failed login attempts which can not be exceeded before an end user's account is locked out. If an account is locked out, the TOE invokes a delay or timeframe which prevents the end user from attempting to gain access until the timeframe is exceeded or a system administrator resets the account.

9.2.3 Security Audit

The security audit function of the TOE enforces the FAU_GEN.1, and FAU_GEN.2 requirements. FAU_GEN.1 requires a reliable time-stamp, which is provided by FPT_STM_EXT.1.1 (provided by the Operational environment). Also provided by the Operational Environment is FAU_SAR_EXP.1, which allows local users to read the TOE's audit logs.

In the evaluated configuration the Windows operating system's respective syslog file stores the record of the startup and shutdown of the TOE's audit functions; while the other Unix and Solaris operating system has this logged in the Error Log. The TOE also provides security auditing capabilities. Both the Web Agent and Policy Server provide logging capabilities. The audit records together include activity by end user, administrative operations by administrator, authorized applications per end user, denied authorizations, end users per application, denied resources, policies per specific role in an application, protected resources, authentication and authorization activity by resource, resource per end user, roles defined per resource, and end users by role. The miniumum contents of each entry in the audit record include the following: Date and time of the report creation, remote server host name, remote server host ID, and remote server account name responsible for the report creation. The Operational Environment shall protect the stored audit records from unauthorized deletion, and shall be able to prevent unauthorized modifications to the audit records in the audit trail.
The Web Agent logs all successful authorizations of end users and the accesses to protected resources via the Policy Server, as well as in the local Trace log file. If an audit message is not successfully sent to the accounting service for an authorization, access to the resource is denied.
The Policy Server records information about the Policy Server itself and logs authentication, authorization, administrator accesses, and administrator changes to policy store objects.

9.2.4 Security Management

The security management function of the TOE enforces the FMT_MSA.1(1), FMT_MSA.1(2), FMT_MSA.1(3), FMT_MSA.2, FMT_MSA.3, FMT_SMR.1, and FMT_SMF.1 requirements.

The TOE provides management capabilities through the Policy Server's WAM Administrative UI. The TSF shall provide the ability to manage its security functions including the management of end user accounts and end user access rights, and TOE resources. The TOE shall provide the ability to set roles for security relevant authority as well as to restrict the ability to define and assign tasks to authorized administrators. The TSF shall allow the Administrator to specify alternative initial values to override the default values set by the TOE when an object or information is created.(e.g. maximum timeout and idle timeout) The TSF also ensures that only secure values are accepted for security attributes.

Through the Administrator Policy, the WAM Administrative UI provides account management functionality, allowing an administrator to enable and disable end users, and manage passwords for individual end users. The Policy Server Management Console is used to manage the functions of the Policy Server such as logging and starting and stopping the Policy Server processes, these features are only used to place the TOE in the evaluated configuration.

The TSF shall enforce the User Policy to restrict the ability of end users to change passwords. The TSF shall enforce the Administrator Policy to query, modify, or delete the external user attributes of the external LDAP directory.

9.2.5 Cryptographic Support

The Cryptographic Support function of the TOE enforces the FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM.1(3), FCS_CKM_EXT.1, FCS_CKM.4, FCS_CKM_EXT.4, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), and FCS_COP_EXT.1 requirements.

The TOE uses encryption keys to encrypt and decrypt sensitive data passed between the TOE's components and stored in the Operational Environment. AES keys derived from Agent keys are used to encrypt TOE cookies that may be read by all agents in a single sign-on environment, and are shared by all agents in a single sign-on environment, since each agent must be able to decrypt cookies encrypted by the other agents. Agent keys are managed by the TOE, and distributed to agents periodically. AES keys derived from session ticket keys are used by the TOE to encrypt session tickets, administrator tokens, and Password Services state data. Session tickets contain credentials and other information relating to a session (including user credentials). Agents embed session tickets in TOE cookies, but cannot access the contents since they do not have access to session ticket keys which never leave the Policy Server.

Both types of keys are kept in the Policy Server's key store..

The following cryptographic keys are generated by the TSF in SiteMinder: Session Keys: AES with 128-bit keys, Agent Keys/Session Ticket Keys: AES Key Wrap with 192-bit keys, and AES Key (derived from Agent Keys/Session Ticket Keys): FIPS-140 Key Expansion Algorithm with 128-bit keys. The TSF shall destroy cryptographic keys in accordance with the zeroization and overwriting method as previously discussed in section 9.1.8.5.

The TOE also relies upon the Operating Environment to provide key generation, key destruction, and cryptographic operation for SSL v3.0 communication between the TOE and its users.

9.2.6 Protection of the TSF

The Protection of the TSF function of the TOE enforces the FPT_FLS.1, and FPT_STM_EXT.1 requirements.
The Policy Server and Web Agent provide partial protection of TSF data. The TOE maintains and controls individual sessions for Administrators and End Users. The TSF, when invoked by the underlying host OS, ensures that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed. The TSF maintains a security domain for its own execution that protects it from interference and tampering by untrusted subjects initiating actions through its own TSFI.

9.2.7 Resource Utilization

The Protection of the TSF function of the TOE enforces the FRU_FLT.1 requirement. The TSF shall ensure the operation of access control to protected resources when a failure of a Policy Server in a SiteMinder cluster is detected.

9.2.8 Trusted Path/Channels

The TOE relies on the Operational Environment will provide SSL v3.0 encryption to enforce the FTP_TRP_EXP.1 requirement. The Operational Environment shall provide a path for communication between the TOE and remote users of the TOE that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification and disclosure. The TSF shall allow initial communication to the trusted path by remote end users and Administrators, and it shall require the use of the trusted path for initial end user and Administrator authentication and all other TSF mediated actions.

The Trusted Channel function of the TOE enforces the FTP_ITC.1 requirement. The TSF shall provide a channel for communication between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure. The TSF shall allow the TSF to initiate communication via the trusted channel, and shall use the trusted channel for the transfer of data between the Policy Server component, and the Web Server and the WAM Administrative UI components.

10 Rationale

10.1 Security Objectives Rationale

The following table provides a mapping with rationale to identify the security objectives that address the stated assumptions and threats.

Assumption	Objective	Rationale
A. ADMIN There will be one or more authorized administrators assigned to install, configure, and manage the TOE and the security information it contains.	OE. ADMIN One or more authorized administrators will be assigned to install, configure and manage the TOE and the security of the information it contains.	OE. ADMIN maps to A. ADMIN in order to ensure that authorized administrators install, manage and operate the TOE in a manner that maintains its security objectives.
A. PATCHES System Administrators exercise due diligence to update the TOE with the latest patches and patch the Operational environment (e.g. OS and database) so they are not susceptible to network attacks.	OE. ADMIN One or more authorized administrators will be assigned to install, configure and manage the TOE and the security of the information it contains.	OE. ADMIN maps to A. PATCHES in order to ensure that the authorized administrators properly patch the Operational environment in a manner that maintains its security objectives.
A.NOEVIL End users and administrators of the TOE are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided by the organization's guidance documentation.	OE.NOEVIL All end users and administrators of the TOE are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided by the organization's guidance documentation.	OE.NOEVIL maps to A.NOEVIL in order to ensure that there are no careless, willfully negligent, or hostile end users or administrators of the TOE.
A.LOCATE The TOE will be located within controlled access facilities that will prevent unauthorized physical access.	OE.LOCATE The TOE will be located within controlled access facilities that will prevent unauthorized physical access.	OE.LOCATE maps to A.LOCATE in order to ensure the physical security in which the TOE operates.

Table 1-183 Assumption to Objective Mapping

Threat	Objective	Rationale
T.ACCESS Authorized users could gain electronic access to protected network	O.ACCESS The TOE will provide measures to authorize users to access specified TOE resources	O.ACCESS (FDP_ACC.1(1), FDP_ACC.1(2), FDP_ACF.1(1),

Threat	Objective	Rationale
resources by attempting to establish a connection that they are not permitted to perform.	once the user has been authenticated. User authorization is based on access rights configured by the authorized users of the TOE.	FDP_ACF.1(2)) addresses T.ACCESS by providing the authorized users with the capability to specify access restrictions on the protected TOE resources to authenticated users.
	OE. FILESYS The Security features offered by the underlying Operating system protect the files used by the TOE.	OE.FILESYS (FAU_STG_EXT.1 addresses T.ACCESS by ensuring that the underlying Operating System provides the capability to store and protect the files used by the TOE.
T.ADMIN_ERROR An administrator may incorrectly install or configure the TOE, or install a corrupted TOE resulting in ineffective security mechanisms.	O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management.	O.ROBUST_ADMIN_GUIDANCE (ALC_DEL.1, AGD_PRE.1, AGD_OPE.1) helps to mitigate T.ADMIN_ERROR 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 unsecure.
	O.MANAGE The TOE will provide authorized users with the resources to manage and monitor user accounts, TOE resources and security information relative to the	O.MANAGE (FMT_MSA.1(1), FMT_MSA.1(2), FMT_MSA.1(3), FMT_MSA.2, FMT_MSA.3,

Threat	Objective	Rationale
	TOE.	FMT_SMR.1, FMT_SMF.1) addresses T.ADMIN_ERROR by ensuring only authorized administrators can use the provided resources for managing and monitoring user accounts, TOE resources and security information relative to the TOE.
T.AUDIT_COMPROMISE A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.	O.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. The TSF will provide load balancing within clusters and failover between clusters, which allows for continued operation of the TOE in the event of a failure within or between clusters.	O.SELF_PROTECTION (FPT_FLS.1, FRU_FLT.1) addresses T.AUDIT_COMPROMISE by ensuring that the TSF can protect itself from users. If the TSF could not maintain and control its domain of execution, it could not be trusted to control access to the resources under its control, which includes the audit trail. Likewise, ensuring that the functions that protect the audit trail are always invoked is also critical to the mitigation of this threat.
	OE. FILESYS The Security features offered by the underlying Operating system protect the files used by the TOE.	OE.FILESYS (FAU_STG_EXT.1 addresses T.AUDIT_COMPROMISE by ensuring that the underlying Operating System provides the capability to store and protect the audit files used by the TOE.
T.EAVESDROPPING A malicious user could eavesdrop on network traffic to gain unauthorized access to TOE data.	O.EAVESDROPPING The TOE will encrypt TSF data that traverses the network to prevent malicious users from gaining unauthorized access	O.EAVESDROPPING (FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM.1(3), FCS_CKM.4,

Threat	Objective	Rationale
	to TOE data.	FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FTP_ITC.1) mitigates T.EAVESDROPPING by ensuring that all communication between components of the TOE are not sent unless they are encrypted.
	OE.EAVESDROPPING The Operational Environemnt will encrypt TSF data that traverses the network to prevent malicious users from gaining unauthorized access to TOE data.	OE.EAVESDROPPING (FCS_CKM_EXT.1, FCS_CKM_EXT.4, FCS_COP_EXT.1, FTP_TRP_EXT.1,) mitigates T.EAVESDROPPING by ensuring that all communication to and from the TOE are not sent unless they are encrypted.
T.MASK Users whether they be malicious or non-malicious, could gain unauthorised access to the TOE by bypassing identification and authentication countermeasures.	O.AUDIT The TOE will provide measures for recording security relevant events that will assist the authorized users in detecting misuse of the TOE and/or its security features that would compromise the integrity of the TOE and violate the security objectives of the TOE.	O.AUDIT (FAU_GEN.1, FAU_GEN.2, FAU_SAR_EXT.1, FPT_STM_EXT.1) addresses T.MASK by providing the authorized users with tools necessary to monitor user activity to ensure that misuse of the TOE does not occur.
T.MASQUERADE A user may masquerade as an authorized user or an authorized IT entity to gain access to data or TOE resources.	O.ROBUST_TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate.	O.ROBUST_TOE_ACCESS (FIA_UAU.1, FIA_UAU_EXT.5(1)), addresses T.MASQUERADE by controlling the logical access to the TOE and its resources. By constraining how and when authorized users can access the TOE, and by mandating the type and strength of the

Threat	Objective	Rationale
		authentication scheme, this objective helps mitigate the possibility of a user attempting to login and masquerade as an authorized user. 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.
	OE.ROBUST_ACCESS The Operational environment will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate.	OE.ROBUST_ACCESS (FIA_UAU_EXT.5(2) addresses T.MASQUERADE by controlling the logical access to the Operational environment and its resources.
T.UNAUTH Users could gain unauthorised access to the network resources by bypassing identification and authentication requirements.	O.AUTH The TOE will provide measures to uniquely identify all users and will authenticate the claimed identity prior to granting a user access to the resources protected by the TOE. The TOE will provide measures to uniquely identify all administrators and will authenticate the claimed identity prior to granting an administrator access to the TOE.	O.AUTH (FIA_ATD.1, FIA_UID.2, FIA_UID_EXT.2, FIA_AFL.1, FIA_SOS.1, FIA_UAU.6) addresses T.UNAUTH by providing measures to uniquely identify and authenticate users through User Name/Password, Basic over SSL, or x.509 certificates.

Table 1-194 Threat to Objective Mapping

10.2 Assurance Measures

This section identifies the assurance measures provided by the developer in order to meet the security assurance requirement components for EAL3 augmented with ALC.FLR.1. A description of each of the TOE assurance measures follows in Table 1-23.

Component	Document(s)	Rationale
ADV_ARC.1 Security Architecture Design	TOE Design Specification v1.7	This document describes the security architecture of the TOE.
ADV_FSP.3 Functional Specification with complete summary	Functional Specification v2.2	This document describes the functional specification of the TOE with complete summary.
ADV_TDS.2 Architectural Design	TOE Design Specification v1.7	This document describes the architectural design of the TOE.
AGD_OPE.1 Operational User Guidance	**1. CA SiteMinder Web Access Manager R12 SP1-CR3 Admin Supplemental Guidance, 1.1, May 29 2009.** **2. SM r12 SP1 Configuration Guide, December 2008** 3. Detailed Design Specifications, FIPS 140-2 compliance, October 2006 **4. CA SiteMinder Web Access Manager Policy Server Administration Guide r12** **5. CA SiteMinder Web Access Manager Policy Server Configuration Guide r12** **6. CA SiteMinder Web Access Manager Web Agent Configuration Guide r12**	These documents describe the operational user guidance for CA SiteMinder r12 SP1-CR3.
AGD_PRE.1 Preparative Procedures	Evaluated Configuration for CA Siteminder Web Access Manager R12 SP1, March 2009	This document describes the preparative procedures that need to be done prior to installing CA SiteMinder r12 SP1-CR3.
ALC_CMC.3 Authorizations Controls	Code Review and Submission Process, January 2009 Project Branch Configuration Management, January 2009 SCM Release and Engineering plan for SiteMinder r12 SP1, January 2009	These documents describe the authorization controls for the TOE.

Component	Document(s)	Rationale
ALC_CMS.3 CM Scope	Code Review and Submission Process, January 2009 Project Branch Configuration Management, January 2009 SCM Release and Engineering plan for SiteMinder r12 SP1, July 2008	These documents describe the CM scope of the TOE.
ALC_DEL.1 Delivery Procedures	**SiteMinder r12 SP1 – NIAP Download/Installation Instruction, October 2008**	This document describes product delivery for CA SiteMinder r12 SP1-CR3 and a description of all procedures used to ensure objectives are not compromised in the delivery process.

Component	Document(s)	Rationale
ALC_DVS.1 Identification of Security Measures	1. GIS – Backup Procedure, May 2008 2. GRC – Global Security – Pre-employment Screening (PES), April 2007 3. Global Risk and Compliance / Global Safety and Asset Protection (GSAP) – Security Operations, July 2007 4. Access Procedure, June 2007 5. GRC – BP&C – RIM – Records Security and Confidentiality Policy, May 2008 6. RIM – Records Disposal Procedure, May 2008 7. Privileged Access Procedure, June 2008 8. Control of Source Code and Design Documents Policy, February 2008 9. Global Risk & Compliance – Business Practices & Compliance – Privacy and Data Protection, March 2007 10. Inactive User Account Procedure, June 2007 11. Server Security Procedure, June 2008 12. US Employee Handbook, July 2008 13. GSC – Supporting CA Online, October 2008 14. CA SiteMinder Common Criteria Assurance Lifecycle Onsite Assessment, November 2008 15. SCM and Release Engineering Plan for SiteMinder r12 SP1, July 2008	These documents provide an identification of security measures for the TOE.
ALC_FLR.1 Basic Flaw Remediation	SiteMinder L2 SiteMinder Sustaining Cycle Process, October 2008	This document describes the ongoing flaw remediation process of the TOE following its initial release.
ALC_LCD.1 Life-Cycle Definition	Project 360 Reference Guide, July 2008	This document provides the life-cycle definition of the TOE.
ASE_CCL.1 Conformance Claims	**Security Target v0.8 (29 May 2009)**	This document describes the CC conformance claims made by the TOE.

Component	Document(s)	Rationale
ASE_ECD.1 Extended Components Definition	**Security Target v0.8 (29 May 2009)**	This document provides a definition for all extended components in the TOE.
ASE_INT.1 Security Target Introduction	**Security Target v0.8 (29 May 2009)**	This document describes the Introduction of the Security Target.
ASE_OBJ.2 Security Objectives	**Security Target v0.8 (29 May 2009)**	This document describes all of the security objectives for the TOE.
ASE_REQ.2 Security Requirements	**Security Target v0.8 (29 May 2009)**	This document describes all of the security requirements for the TOE.
ASE_SPD.1 Security Problem Definition	**Security Target v0.8 (29 May 2009)**	This document describes the security problem definition of the Security Target.
ASE_TSS.1 TOE Summary Specification	**Security Target v0.8 (29 May 2009)**	This document describes the TSS section of the Security Target.
ATE_COV.2 Analysis of Coverage	SiteMinder Functional Test Plan, March 2009	This document provides an analysis of coverage for the TOE.
ATE_DPT.1 Basic Design	SiteMinder Functional Test Plan, March 2009	This document describes the basic design of the TOE.
ATE_FUN.1 Functional Tests	SiteMinder Functional Test Plan, March 2009	This document describes the functional tests for the TOE.
ATE_IND.2 Independent Testing	Evaluator Independent Test Plan and Report, v1.0, 30 January 2009	This document describes the independent testing for the TOE.
AVA_VAN.2 Vulnerability Analysis	Evaluator Vulnerability Analysis, v1.1, 29 May 2009	This document describes the vulnerability analysis of the TOE.

Table 1-205 Assurance Requirements Evidence

10.3 EAL 3 Justification

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

10.4 Requirement Dependency Rationale

All Security Functional Requirement component dependencies have been met by the TOE, and the support of the Operational Environment with the exception of FCS_CKM.1, FCS_CKM.4, and FPT_STM.1.

FCS_CKM.1 Cryptographic key generation is a dependency of FCS_COP.1(4). However, the Policy Store Key that is used by FCS_COP.1(4) is created during the installation of the TOE and is therefore outside the scope of the evaluation, since it is before the TOE is in the evaluated configuration.

FCS_CKM.4 Cryptographic key destruction is a dependency for FCS_COP.1(4) However, the Policy Store Key that is used by FCS_COP.1(4) is created during the installation of the TOE and is therefore outside the scope of the evaluation, since it is before the TOE is in the evaluated configuration.

FCS_CKM.4 Cryptographic key destruction is a dependency for FCS_CKM.1(1) and FCS_COP.1(1) requirements. The Session Key is generated by the Policy Server and then placed in the local memory of the Policy Server and Web Agent, and is only kept in memory as long as a session exists between the two components. Once the connection is terminated a new Session Key is generated for use by both ends, and the old key cannot be used.

FCS_CKM.4 Cryptographic key destruction is a dependency for FCS_CKM.1(3) and FCS_COP.1(3) requirements. The AES Key is a key that is generated through the use of an Agent Key. The Agent Key is what is maintained in the local memory of the Web Agent and when replaced the set of Agent Keys are zeroed and overwritten.

FPT_STM.1, Reliable Time Stamps is a dependency of FAU_GEN.1, which is met by the IT environment. The underlying Operating System will be available to the TOE for use in determining the timestamp for the audit trail.

10.5 Security Functional Requirements Rationale

The following table provides a mapping with rationale to identify the security functional requirement components that address the stated TOE and environment objectives.

Objective	Security Functional Components	Rationale
O.ACCESS The TOE will provide measures to authorize users to access specified TOE resources once the user has been authenticated. User authorization is based on access rights configured by the authorized users of the TOE.	FDP_ACC.1(1) Subset access control	FDP_ACC.1(1) states the TSF shall enforce the Policy on Users to access protected resources via the web agent.
	FDP_ACC.1(2) Subset access control	FDP_ACC.1(2) states the TSF shall enforce the Administrator Policy on Administrators performing operations as defined in the Table 1-12 to Policy Server objects.
	FDP_ACF.1(1) Security attribute based access control	FDP_ACF.1(1) states the TSF shall enforce the Domain Policy to objects based on username, groups, resources, and realm.
	FDP_ACF.1(2) Security attribute based	FDP_ACF.1(2) states the TSF shall enforce the Administrator

Objective	Security Functional Components	Rationale
	access control	Policy to objects based on task and scope.
O.AUDIT The TOE will provide measures for recording security relevant events that will assist the authorized users in detecting misuse of the TOE and/or its security features that would compromise the integrity of the TOE and violate the security objectives of the TOE.	FAU_GEN.1 Audit data generation	FAU_GEN.1 states that the TSF shall be able to generate an audit record for the start-up and shutdown of the audit functions and all auditable events for the level of audit. For each record, the TSF shall record the date/time/type/outcome of the event and subject identity.Also, the TSF shall generate an audit report based on user activity, administrator operations, authorized applications, denied authorizations and resources, policies per role, protected resources, authentication and authorization, and roles. The TSF shall also record the date/time, remote server host name and ID, account name and errors.
	FAU_GEN.2 User identity association	FAU_GEN.2 states the TSF shall be able to associate each auditable event with the identity of the user that caused the event.
	FAU_SAR_EXT.1 Audit Review	FAU_SAR_EXT.1 states the TSF shall provide the Administrator of the Operational Environment with the capability to read role requesting access to objects/resources, user role assignment, role based access authorization result, object/resource for which access is being requested, from the audit records.
	FPT_STM_EXT.1 Reliable Time Stamps	FPT_STM_EXT.1 states that the Operational environment shall be able to provide reliable time-stamps for use by the TOE. These time stamps are used for audit records created by the TOE.
O.AUTH	FIA_ATD.1 User attribute definition	FIA_ATD.1 specifies the security attributes that should be

Objective	Security Functional Components	Rationale
The TOE will provide measures to uniquely identify all users and will authenticate the claimed identity prior to granting a user access to the resources protected by the TOE. The TOE will provide measures to uniquely identify all administrators and will authenticate the claimed identity prior to granting an administrator access to the TOE.		maintained at the level of the user. This means that the security attributes listed are assigned to and are changed at the level of the user. In other words, changing a security attribute in this list associated with a user should have no impact on the security attributes of any other user.
	FIA_UID.2 User identification before any action	FIA_UID.2 requires a user be identified before any access to the TOE and resources protected by the TOE is allowed.
	FIA_UID_EXT.2 User identification before any action	FIA_UID_EXT.2 requires each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.
	FIA_AFL.1 Authentication Failure Handling	FIA_AFL.1 ensures unauthorized users cannot endlessly attempt to authenticate. After some number of failed attempts defined by an authorized administrator, the user becomes locked out.
	FIA_SOS.1 Verification of Secret	FIA_SOS.1 The strength of the authentication scheme is sufficient to ensure that unauthorized users cannot easily impersonate an authorized user.
	FIA_UAU.6 Re-authentication	FIA_UAU.6 requires the user to re-authenticate if the realm is protected by an authentication scheme with a higher protection level.
O.MANAGE The TOE will provide authorized administrators with the resources to manage and monitor user accounts, resources, and security information relative to the TOE.	FMT_MSA.1(1) Management of security attributes	FMT_MSA.1(1) states the TSF shall enforce the Administrator Policy to restrict the ability to view or perform operations specified in Table 1-14 to roles as specified in the same table.
	FMT_MSA.1(2) Management of security attributes	FMT_MSA.1(2) states the TSF shall enforce the User Policy to restrict the ability to change a user's password.

Objective	Security Functional Components	Rationale
	FMT_MSA.1(3) Management of security attributes	FMT_MSA.1(3) states the TSF shall enforce the Administrator Policy to query, modify, or delete the external user attributes as listed in Table 1-16 External LDAP Directory Attributes.
	FMT_MSA.3 Static attribute initialization	FMT_MSA.3 states the TSF shall enforce the Administrator Policy to provide restrictive default values for security attributes that are used to enforce the SFP. It allows the authorized administrators to override the default values set for security attributes when creating user accounts.
	FMT_SMF.1 Specification of management functions	FMT_SMF.1 requires that the TOE provide the ability to manage its security functions including the management of user accounts and user access rights, TOE resources and security information recorded in the audit logs.
	FMT_SMR.1 Security Roles	FMT_SMR.1 requires the TOE to provide the ability to set roles for security relevant authority as well as to restrict the ability to define and assign roles to authorized administrators.
O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management.	ALC_DEL.1 Delivery Procedures	ALC_DEL.1 describes product delivery and a description of all procedures used to ensure objectives are not compromised in the delivery process.
	AGD_PRE.1 Preparative Procedures	AGD_PRE.1 documents the procedures necessary and describes the steps required for the secure installation, generation, and start-up of the TOE.
	AGD_OPE.1 Operational user guidance	AGD_OPE.1 describes the proper use of the TOE from a user standpoint.
O.EAVESDROPPING The TOE will encrypt TSF data that traverses the network to prevent malicious	FCS_CKM.1(1) Cryptographic key generation	FCS_CKM.1(1) states the TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [AES] and specified

Objective	Security Functional Components	Rationale
users from gaining unauthorized access to TOE data.		cryptographic key sizes [128 bits]
	FCS_CKM.1(2) Cryptographic key generation	FCS_CKM.1(2) states the TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [AES Key Wrap] and specified cryptographic key sizes [192 bits]
	FCS_CKM.1(3) Cryptographic key generation	FCS_CKM.1(3) states the TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [FIPS-140 Key Expansion Algorithm] and specified cryptographic key sizes [128 bits]
	FCS_CKM.4 Cryptographic key destruction	FCS_CKM.4 states the TSF shall destroy cryptographic keys with the overwrite method that meets key zeroization.
	FCS_COP.1(1) Cryptographic operation	FCS_COP.1(1) states the TSF shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm [AES in OFB mode] and cryptographic key sizes [128 bits]
	FCS_COP.1(2) Cryptographic operation	FCS_COP.1(2) states the TSF shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm [AES in CBC mode] and cryptographic key sizes [128 bits]
	FCS_COP.1(3) Cryptographic operation	FCS_COP.1(3) states the TSF shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm [AES in CBC mode] and cryptographic key sizes [128 bits]
	FCS_COP.1(4) Cryptographic operation	FCS_COP.1(4) states the TSF shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm [AES Key Wrap] and cryptographic key sizes [128 bits]
	FMT_MSA.2 Secure security attributes	FMT_MSA.2 states that the TSF shall ensure that only secure values are accepted for security attributes.

Objective	Security Functional Components	Rationale
	FTP_ITC.1 Inter-TSF trusted channel	FTP_ITC.1 states the TSF shall provide a communication channel between itself that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure. The TSF shall allow the TSF to initiate communication via the trusted channel, and it shall require the use of the trusted channel for the transfer of data between TOE subsystems.
OE.EAVESDROPPING The Operational Environemnt will encrypt TSF data that traverses the network to prevent malicious users from gaining unauthorized access to TOE data.	FTP_TRP_EXT.1 Trusted Path	FTP_TRP_EXT1 states the Operational Environment shall provide a communication path between the TSF and remote users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification or disclosure. The TSF shall allow remote users to initiate communication via the trusted path, and it shall require the use of the trusted path for initial user authentication and all other TSF mediated actions.
	FCS_CKM_EXT.1Cryptographic key generation	FCS_CKM_EXT.1 states the TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [RSA] and specified cryptographic key sizes [1024 bits]
	FCS_CKM_EXT.4 Cryptographic key destruction	FCS_CKM_EXT.4 states the TSF shall destroy cryptographic keys with the overwrite method that meets key zeroization.
	FCS_COP_EXT.1 Cryptographic operation	FCS_COP_EXT.1 states the TSF shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm

Objective	Security Functional Components	Rationale
		[AES] and cryptographic key sizes [128 bits]
OE.FILESYS The security features offered by the underlying Operating System and Database protect the files used by the TOE. .	FAU_STG_EXT.1 Protected audit trail storage	FAU_STG_EXT.1 requires that the underlying Operating System protect the audit records generated by the TOE that are stored in the SiteMinder Audit Store.
O.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.	FPT_FLS.1 Failure with preservation of secure state	FPT_FLS.1 requires that the TSF shall preserve a secure state when a failure of a Policy Server within a SiteMinder cluster is detected.
	FRU_FLT.1 Degraded Fault Tolerance	FRU_FLT.1 ensures the operation of access control to protected resources when the failure of a Policy Server in a SiteMinder cluster is detected.
O.ROBUST_TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate	FIA_UAU.1 Timing of authentication	FIA_UAU.1 states the TSF shall allow access to unprotected resources on behalf of the user to be performed before the user is authenticated, and shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Objective	Security Functional Components	Rationale
	FIA_UAU_EXT.5(1) Multiple authentication schemes.	FIA_UAU_EXT.5(1) states the TSF shall provide x.509 certificates, Windows Authentication Scheme, and Basic over SSL to support user authentication, and the TSF shall authenticate any user's claimed identity according to the realm and its associated protection level.
OE.ROBUST_ACCESS The Operational environment will provide mechanisms that control a user's physical access to the TOE and to explicitly deny access to specific users when appropriate.	FIA_UAU_EXT.5(2) Windows Authentication Scheme.	FIA_UAU_EXT.5(2) states the Operational environment shall authenticate any user's claimed identity through the use of Windows credentials or Windows password before accessing a protected resource.

Table 1-216 Security Functional Requirements Rationale

10.6 Extended Requirements Rationale

10.6.1 FAU_STG

This family defines the requirements for the TSF to be able to create and maintain a secure audit trail. Stored audit records refers to those records within the audit trail, and not the audit records that have been retrieved (to temporary storage) through selection. The following requirements were made extended requirements of FAU_STG:

FAU_SAR_EXT.1.1 and FAU_SAR_EXT.1.2 were made extended requirements because they refer to the operational environment as opposed to the TSF as stated in FAU_SAR.1. They ensure that the operational environment shall allow an administrator of the operational environment to view the audit records.

FAU_STG_EXT.1.1 and FAU_STG_EXT.1.2 were made extended requirements because they refer to the operational environment as opposed to the TSF as stated in FAU_STG.1.

They ensure that the operational environment shall protect the unauthorized deletion of audit records and prevent modifications to that data in the audit trail.

10.6.2 FIA_UID

This family defines the conditions under which users shall be required to identify themselves before performing any other actions that are to be mediated by the TSF and which require user identification. The following is an extended requirement of FIA_UID:

FIA_UID_EXT.2 was made an extended requirement of FIA_UID.2 because it states that the operational environment instead of the TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

10.6.3 FIA_UAU

This family defines the types of user authentication mechanisms supported by the TSF. This family defines the required attributes on which the user authentication mechanisms must be based. The following are extended requirements of FIA_UAU:

FIA_UAU_EXT.5(1) and FIA_UAU_EXT.5(2) were created to explain the authentication schemes. FIA_UAU_EXT.5(1) was made an extended requirement because Windows Authentication Scheme relies on the underlying operating system, which is in the operational environment. It states the TSF shall provide a x.509 certificate, Windows Authentication Scheme, and/or Basic over SSL to support user authentication, and the user's claimed identity shall be authenticated according to the realm and its associated protection level. FIA_UAU_EXT.5(2) was made an extended requirement because it refers to the operational environment as opposed to the TSF as stated in FIA_UAU.5. It states the operational environment shall authenticate any user's claimed identity through the use of Windows credentials or Windows password before accessing a protected resource.

10.6.4 FPT_STM

This family addresses requirements for a reliable time stamp function within a TOE. The following is an extended requirement for FPT_STM:

FPT_STM_EXT.1.1 was made an extended requirement because it refers to the OS in the operational environment as opposed to the TOE as stated in FPT_STM. It states that the underlying operating system in the operational environment shall provide a reliable time stamp from its system clock for use by the TOE.

10.6.5 FCS_CKM

This family addresses requirements for generating and destroying cryptographic keys by the TOE. The following is an extended requirement for FCS_CKM.1 and FCS_CKM.4:

FPT_CKM_EXT.1.1 was made an extended requirement because it refers to the operational environment as opposed to the TOE as stated in FCS_CKM.1. It states that the TOE relies on the operational environment to generate keys for SSL encryption between the TOE and its users.

FPT_CKM_EXT.4.1 was made an extended requirement because it refers to the operational environment as opposed to the TOE as stated in FCS_CKM.4. It states that the TOE relies on the operational environment to destroy keys that are used for SSL encryption between the TOE and its users.

10.6.6 FCS_COP

This family addresses requirements for conducting cryptographic operation by the TOE. The following is an extended requirement for FCS_COP.1:

FPT_COP_EXT.1.1 was made an extended requirement because it refers to the operational environment as opposed to the TOE as stated in FCS_COP.1. It states that the TOE relies on the operational environment to perform encryption and decryption for the SSL encryption between the TOE and its users.

10.6.7 FTP_TRP

This family addresses requirements for providing a protected communication path between the TOE and its remote users. The following is an extended requirement for FTP_TRP.1:

FTP_TRP_EXT.1 was made an extended requirement because it refers to the operational environment as opposed to the TOE as stated in FTP_TRP.1. It states that the TOE relies on the operational environment to provide a SSL protected communication path between the TOE and its users.

This Security Target does not include any extended Security Assurance Requirements.

10.7 PP Claims Rationale

This Security Target does not claim Protection Profile conformance.