CA SiteMinder® Federation Security Services r12 SP1 CR3 Security Target

Version 1.0
April 5, 2010

Prepared for:
CA
100 Staples Drive
Framingham, MA 01702

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

Table of Contents

List of Figures

List of Tables

1 Security Target Introduction

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

1.1 ST Reference

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

1.1.1 ST Identification

ST Title:	CA SiteMinder® Federation Security Services r12 SP1 CR3
ST Version:	1.0
ST Publication Date:	April 5, 2010
ST Author:	Booz Allen Hamilton

1.1.2 Document Organization

Chapter 1 of this ST provides identifying information for the CA SiteMinder Federation Security Services r12 SP1 CR3 ST. It includes an ST Introduction, ST Reference, TOE Reference, TOE Overview, and TOE Type.

Chapter 2 describes the TOE Description, which includes the Evaluated Components of the TOE, Excluded Components, Physical Boundary and Logical Boundary.

Chapter 3 describes the Conformance Claims made by this ST. This chapter provides information on CC Version, CC Part2, CC Part3, PP Claims, Package Name, and Conformance Claims Rationale.

Chapter 4 describes the Security Problem Definition as it relates to Threats, Operational Security Policies, Assumptions and Objectives met by the TOE and Operational Environment.

Chapter 5 identifies the Extended Security Functional Requirements (SFRs).

Chapter 6 describes the Extended Security Assurance Requirements (SARs).

Chapter 7 describes the Security Functional Requirements (SFRs) for the TOE.

Chapter 8 describes the Security Assurance Requirements (SARs) for EAL3.

Chapter 9 is the TOE Summary Specification (TSS), a description of the functions provided by CA SiteMinder Federation Security Services r12 SP1 CR3 to satisfy the Security Functional Requirements (SFRs) and the Security Assurance Requirements (SARs). This chapter also includes the TSS Rationale.

Chapter 10 provides a rationale, or pointers to a rationale, for the Security Problem Definition as defined in Chapter 4. This chapter also includes the EAL Justification, Extended SFR Rationale, SFR Rationale, and SFR Dependency Rationale.

Chapter 11 provides a table for the Assurance Requirements Evidence.

1.1.3 Terminology

This section defines the customer and CC terminology used throughout this ST. These tables are to be used by the reader as a quick reference guide for terminology definitions.

Terminology	Definition
Account Linking	The process by which a user's identification information is used to bridge two distinct accounts.
Administrator	A trusted user who has privileges to administer the TOE.
Affiliate Domain	A logical grouping of federated entities associated with one or more user stores. The affiliate domain not only contains federated entities but it also defines which user stores are associated with the domain. To authenticate a user, SiteMinder must have access to the user store where a user record is defined. The Policy Server locates a user record by querying the user stores specified in the affiliate domain's search order. The search order is defined when adding user store connections to an affiliate domain. The order of directories can be shifted.
Artifact	A reference to a SAML assertion.
Artifact Resolution Service	Provides a mechanism by which SAML protocol messages may be passed by reference using a small, fixed-length value called an artifact. The artifact receiver uses the Artifact Resolution Service to ask the message creator to dereference the artifact and return the actual protocol message. The artifact is passed to a message recipient using one SAML binding (e.g. HTTP Redirect) while the resolution request and response take place over a synchronous binding, such as SOAP.
Asserting Party	A SAML authority that generates an assertion for use by a Relying Party. The Asserting Party creates, maintains, and manages identity information for users and provides user authentication to other relying parties. In SAML 2.0, an Asserting Party is the Identity Provider. In SAML 1.1, an Asserting Party is the producer.
Assertion	An assertion contains several different internal statements about authentication, authorization, and attributes. The valid structure and contents of an assertion are defined by the SAML assertion XML schema. Assertions are created by an Asserting Party based on a request of some sort from a Relying Party, although under certain circumstances, the assertions are delivered to a Relying Party in an unsolicited manner. SAML defines two browser-based protocols that specify how SAML assertions are passed between partners to facilitate single sign-on. The profiles are: • Browser/artifact profile—defines a SAML artifact as a reference to a SAML assertion. • Browser/POST profile—returns a response that contains an

	assertion. Note: For SAML 2.0, the artifact and POST profiles are referred to as HTTP bindings.
Assertion Query/Request Profile	The SAML Attribute Authority adheres to the SAML 2.0 Assertion Query/Request profile. It relies on the Attribute Service to process a query message and create attribute assertions. The SAML Requester is a SAML entity that uses the SAML 2.0 Assertion Query/Request profile to request attributes for a user
Attribute Authority	The SAML Attribute Authority adheres to the SAML 2.0 Assertion Query/Request profile. It relies on the Attribute Service to process a query message and create attribute assertions. These assertions contain user attributes that a SAML Requester uses for SiteMinder to authorize access to protected resources. The Attribute Service is part of the Federation Web Services application.
Attribute Service	The Attribute Service uses the NameID to disambiguate the user so it knows what values to return for the requested attributes. The Attribute Service returns a response message that includes an attribute assertion wrapped in a SOAP message. This response includes the user attributes. When an attribute is configured, Administrators indicate whether the attribute is used as part of a single sign-on request, or to satisfy an attribute query request. The attributes function is determined by the Retrieval Method field in the SAML Service Provider Attribute dialog.
Attribute Statement	Specific identifying attributes about the subject
Authentication Scheme	An authentication scheme is a Policy Server object that determines the credentials a user will need to access a protected resource. Authentication schemes are assigned to realms. When a user tries to access a resource in a realm, the authentication scheme of the realm determines the credentials that a user must supply in order to access the resource.
AuthnRequest Service (SAML 2.0)	This service enables a Service Provider to generate an AuthnRequest message for cross-domain single sign-on. This message contains information that enables Federation to redirect the user's browser to the Single Sign-on Service at the Identity Provider. The AuthnRequest service is used for single sign-on using POST and artifact binding. *Note: The format of the AuthnRequest message issued by this service is specified in the Profiles for the OASIS Security Assertion Markup Language (SAML) v2.0.*
Authorization	The process of identifying and authenticating an administrator user by the TOE.
Backchannel	Used for secure communications directly with remote partner (i.e. not through user browser); Federated Web Server(s) in communication with a Web Agent
Binding	SAML Binding refers to how the various SAML protocol messages are carried over underlying transport protocols. Note: For SAML 2.0, the artifact and POST profiles are referred to as HTTP bindings.
Consumer	The Relying Party (SAML 1.1). A consumer is the entity that uses the SAML assertions to authenticate a user and to establish a session for the user.
Disambiguation	The method by which the TOE locates a user in the user store.

Enhanced Client and Proxy (ECP) Profile	Defines a specialized SSO profile where enhanced clients or proxies use the Reverse-SOAP (PAOS) and SOAP bindings.
Entity Role	An Asserting or Relying Party.
Entity Type	A local or remote entity.
Federated Network	In a federated network, there is an entity that generates SAML assertions (Asserting Party). Assertions contain information about a user whose identity is maintained locally at the federated entity that generates them. There is another entity that uses the SAML assertions (Relying Party) to authenticate a user and to establish a session for the user. Depending on the protocol, these two entities are named differently, but the functions they serve are the same. In SAML 1.1, the Asserting Party is known as a producer, while the Relying Party is known as a consumer. In SAML 2.0, the Asserting Party is known as an Identity Provider (IdP), while the Relying Party is known as a Service Provider (SP). A federated entity may be both a producing authority (Identity Provider/IdP) and a consuming authority (Service Provider/SP).
Federation	A federation consists of one Asserting Party (Identity Provider/IdP) and one or more relying parties (Service Provider/SP). A federation provides a means for these partner services to agree on and establish a common, shared name identifier to refer to the user in order to share information about the user across the organizational boundaries.
Federated Entity	A partner in a federated network.
Federation Web Services	Also referred to as Web Agent Option Pack. Consists of the following: • Single Sign On (SSO) • Single Log Out (SLO) • Artifact Resolution • Assertion Consumer • Inter-site Transfer • SAML Credential Collector • Assertion Retriever • Agent API • Attribute Service • Auth URL JSP FWS provides the SAML credential collector servlet, which consumes assertions and other services for federated network configurations.
Get/Put/POST	An HTTP operation known as a user's request. It is received by the Web Agent and forwarded to the Policy Server.
Groups	A group (agent group, rule group, response group) contains individual items or groups of its own type. For example, a rule group can contain rules and/or groups of rules.
HTTP Artifact Binding	Defines that an artifact (described above in the Artifact Resolution Protocol) needs to be transported from a message sender to a message receiver using HTTP. Two mechanisms are provided: either an HTML form control or a query string in the URL."
HTTP Redirect Binding	Defines how SAML protocol messages are transported using HTTP redirect messages (302 status code responses)
HTTP POST Binding	Defines how SAML protocol messages are transported within the base64-encoded content of an HTML form control.
Identity Mapping	The method of user identification; the user identification decision determines what information (one or more user attributes) is sent as the user identity in the assertion.

Identity Provider (IdP)	The Asserting Party (SAML 2.0). The IdP generates SAML assertions to be used by the Service Provider.
Key Store	Entity used by the SiteMinder Policy Server to store encryption keys used by the Policy Server when communicating with SiteMinder Web Agents.
Option Pack	The Policy Server Option Pack is an add-on to the SiteMinder Policy Server. It contains the central processing of the TOE, which includes the operations to create and extract data from SAML assertions, and query and modification of SiteMinder data stores. This add-on is not a separate installer; instead, it is a selectable option during the installation of the Policy Server.
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 stores, 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 group of resources (realm), as well as the users who may access the resources, and the administrator who sets up entitlements.
Policy Server	CA SiteMinder software component that provides a platform for managed key operations, authentication, authorization, and security management. The Policy Server provides the SAML authentication scheme at the Relying Party. It also provides the SAML assertion generator used by a producing federated entity.
Policy Server Option Pack	See Option Pack.
Policy Store	Collection of CA SiteMinder Policy Server objects. Policy stores reside in an ODBC (see page 19)-enabled database or an LDAP (see page 17) directory.
Producer	The Asserting Party (SAML 1.1). A producer is the entity that generates the SAML assertions.
Profile	SAML profiles define how the SAML assertions, protocols, and bindings are combined and constrained to provide greater interoperability in particular usage scenarios. Some of these profiles are examined in detail later in this document.
Protocol Message	SAML protocol messages are used to make the SAML-defined requests and return appropriate responses. The structure and contents of these messages are defined by the SAML-defined protocol XML schema.
Protected Resource	Any set of data or applications that require authorization and authentication in order to gain access.
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 define a directory or folder and possibly its subdirectories.
Relying Party	A SAML entity that uses information from a SAML authority to provide access to services. The Relying Party uses assertions it receives from an Asserting Party to authenticate a user. In SAML 2.0, the Relying Party is the Service Provider. In SAML 1.1, the Relying Party is the consumer.
Reverse SOAP (PAOS) Binding	Defines a multi-stage SOAP/HTTP message exchange that permits an HTTP client to be a SOAP responder. Used in the Enhanced Client or Proxy Profile and particularly designed to support WAP gateways.
Rule	A Policy Server object that identifies a resource and the actions that will be

	allowed or denied access to the resource. Rules also include actions associated with specific events, such as what to do if a user fails to authenticate correctly when asked for their credentials.
SAML Attribute	A component of a user's Distinguished Name(DN) required by a Relying Party in a federation to disambiguate the user during Single Sign-On.
Scope	Indicates whether the administrator's privileges extend to all domains and applications or to only specific domains and applications.
Secure Proxy Engine	Forwards traffic to backend servers; employs web server, servlet engine, proxy server and Federation Web Services features. This engine consists of two components – Apache Web Server and Tomcat server.
Security Assertion Markup Language (SAML)	This standard defines an XML-based framework for describing and exchanging security information between on-line business partners. In the evaluated configuration, SAML v1.1 and v2.0 are used.
Security Zone	A security zone is a segment of a single cookie domain, used as a method of partitioning applications to permit different security requirements for resource access.
Service Provider (SP)	The Relying Party (SAML 2.0)
Single Logout Profile	Defines how the SAML Single Logout Protocol is used with SOAP, HTTP Redirect, HTTP POST, and HTTP Artifact bindings.
Single Logout Protocol	Defines a mechanism to allow near-simultaneous logout of active sessions associated with a principal. The logout is directly initiated by the user, or initiated by an IdP or SP because of a session timeout, administrator command, etc.
Single Sign-on Service (SAML 2.0)	This service enables an Identity Provider to process IdP-or SP-initiated requests for federated resources. The Identity Provider gathers the necessary Service Provider configuration information to generate an assertion that it passes back to the Service Provider. The Service Provider then uses the assertion for authentication purposes.
SiteMinder Object	Rules, realms, domains, and other components of SiteMinder which can be managed by the TOE. Refer to Table 7-2 for applicable functions.
SLO Service	This service allows a user to log out of all applications in the federation simultaneously, with a single logout event. Single logout is initiated by an Identity Provider or a Service Provider.
Smkeydatabase	The smkeydatabase is a key and certificate database used for signing, verification, encryption, and decryption between a SiteMinder consuming authority and a SiteMinder producing authority. The database is made up of multiple files. Administrators manage and retrieve keys and certificates in this database using the SiteMinder tool called smkeytool.
Tunnel Services	Tunnel Services provides an API which is used to facilitate trusted channels for communications between distributed parts of the TOE
User	An authorized user of the TOE without administrative privileges.
User Store	A user store in SiteMinder is an object that contains details for connecting to an existing user store that resides outside of SiteMinder. This allows an administrator to configure a simple connection to an existing user store, instead of replicating user information within SiteMinder. The username space is an LDAP directory server.
User Session	An instance of a user requesting a federated resource or an Administrator managing the TOE. Once granted access to the federated resource by SiteMinder, the session is established across the federation and becomes a global session.
Web Agent	A Web Agent is installed on a Web server to secure access to resources.
Web Agent Configuration	An Agent Configuration Object holds configuration parameters for one or

Object	more Web Agents.
Web Agent Group	A Web 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. Agent groups are configured in SiteMinder for groups of servers that distribute the workload for access to a popular set of resources.
Web Agent Option Pack	See Federation Web Services.
Xpath Query	Xpath is how an Administrator specifies a path to a specific component of an XML file. Xpath is used to define where to look up user information in the XML file.

Table 1-1: Customer Terminology Definitions

Term	Definition
Authorized user	A user who may, in accordance with the TSP, perform an operation.
Base Component	The entity in a composed TOE, which has itself been the subject of an evaluation, providing services and resources to a dependent component (SiteMinder Web Access Manager r12 SP1 CR3).
Composed Assurance Package (CAP)	A CAP is applied to a composed TOE, which is comprised of components that have been (are going through) component TOE evaluation.
Composed TOE	A TOE comprised solely of two or more components that have been successfully evaluated.
Dependent Component	An entity in a composed TOE, which is itself the subject of an evaluation, relying on the provision on services by a base component (Federation Security Services).
External IT entity	Any IT product or system, un-trusted or trusted, outside of the TOE that interacts with the TOE.
TOE Security Functionality (TSF)	A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.

Table 1-2: CC Specific Terminology

1.1.4 Acronyms

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

Acronym	Definition
AES	Advanced Encryption Standard
API	Application Programming Interface
CA (not vendor)	Certificate Authority
CAP	Composed Assurance Package
CC	Common Criteria
CRL	Certificate Revocation List
DB	Database
DER	Distinguished Encoding Rules
DNS	Domain Name Service
EAL	Evaluation Assurance Level
ECP	Enhanced Client or Proxy
FIPS	Federal Information Processing Standards
FSS	Federation Security Services

FTP	File Transfer Protocol
FWS	Federation Web Services
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol over SSL
ID	Identification
IDP	Identity Provider
IETF	Internet Engineering Task Force
IT	Information Technology
J2EE	Java 2 Platform, Enterprise Edition
JSP	Java Server Pages
LDAP	Lightweight Directory Access Protocol
MDSSO	Multi-domain Single Sign-On
OASIS	Organization for the Advancement of Structured Information Standards
OCSP	Online Certificate Status Protocol
ODBC	Open Database Connectivity
OS	Operating System
PKCS	Public Key Cryptography Services
PEM	Privacy-Enhanced Electronic Mail
PS	Policy Server
RP	Relying Party
SAML	Security Assertion Markup Language
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SLO	Single Logout
SM	SiteMinder
SMTP	Simple Message Transfer Protocol
SP	Service Provider
SPS	Secure Proxy Server
SQL	Structured Query Language
SSL	Secure Socket Layer
SSO	Single Sign-on
ST	Security Target
TCP/IP	Transmission Control Protocol/Internet Protocol
TOE	Target of Evaluation
TSF	TOE Security Functionality
UI	User Interface
URL	Uniform Resource Locator
WA	Web Agent
WAM	Web Access Management
WS	Web Server

Table 1-3: Acronym Definitions

1.1.5 References

- CA SiteMinder Federation Security Services Guide r12 SP1
- CA SiteMinder Web Access Manager r12 SP1 CR3 Security Target v1.0
- CA SiteMinder Web Access Manager Policy Server Configuration Guide r12 SP1

- CA SiteMinder Web Access Manager Policy Server Administration Guide r12 SP1
- CA SiteMinder Web Access Manager Web Agent Configuration Guide r12 SP1
- W3C Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V1.1
- W3C Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0
- W3C Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0
- W3C Encryption Syntax and Processing

1.2 TOE Reference

CA SiteMinder® Federation Security Services r12 SP1 CR3

1.3 TOE Overview

This Security Target (ST) defines the Information Technology (IT) security requirements for the CA SiteMinder Federation Security Services r12 SP1 CR3. The TOE is an identification and access management application consisting of CA's Federation Security Services built on top of CA SiteMinder Web Access Manager r12 SP1 CR3. The TOE allows partnerships to be established between two organizations in order to share user identification information and facilitate single sign-on (SSO) and single logout (SLO) across multiple domains, where each domain has its own Policy Server/Web Agent. SiteMinder provides users the ability to easily and securely access the data and applications of these federated entities once they have been authenticated by Federation.

Note: For more information on SiteMinder Web Access Manager r12 SP1 CR3, see the CA SiteMinder Web Access Manager r12 SP1 CR3 Security Target v1.0.

The TOE:

- Has the ability to generate SAML assertions, which includes identity information and attributes from a user store.
- Possesses the capability to send the assertion to the relevant federated partner(s).
- Has the ability to store a SAML assertion until it is retrieved by the Relying Party (applies to the artifact profile).
- Possesses an intuitive user interface.
- Is able to pass identity data to a target application as an encrypted cookie or header.

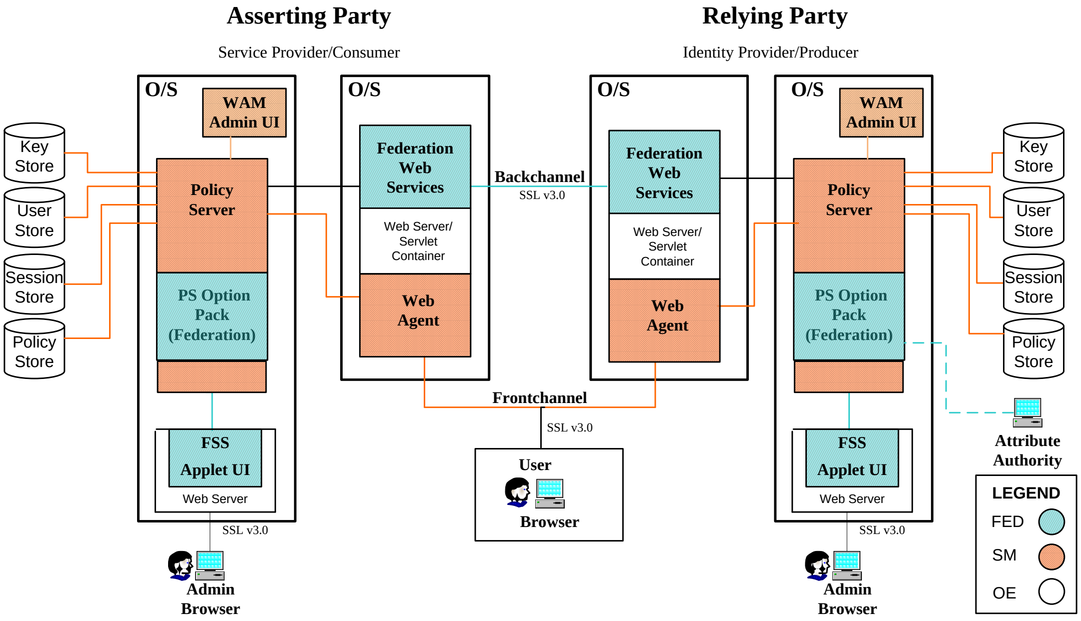

Figure 1-1: TOE Boundary

Note: In SAML 1.1, entities that consume assertions are referred to as consumers (Relying Party) and entities that generate assertions are referred to as producers (Asserting Party). In SAML 2.0, entities that consume assertions are referred to as Service Providers (Relying Party) and entities that generate assertions are referred to as Identity Providers (Asserting Party). The general terms of Asserting Party and Relying Party are used throughout this ST where applicable.

As shown in Figure 1-1, there are two users of the TOE: users and Administrators. Users do not have administrative privileges and use a web browser to authenticate to the TOE using SSL v3.0. Administrators use a browser to launch the FSS Applet UI, which then connects to the TOE using SSL v3.0. The FSS Applet UI runs on top of an environmental web server, which mediates all communications between the remote administrator and the TOE. Administrators use the FSS Applet UI to manage the TOE once they have been authenticated. Administrators manage multiple aspects of the TOE such as agents and their configurations, groups, policy and affiliate domains, authentication schemes, policies and rules See Section 9.1.4 for more information on the management functions that are performed from the FSS Applet UI. All data in the environmental data stores is encrypted. The TOE does not access these data stores directly; instead, requests to query or modify these data stores are made on behalf of SiteMinder, which performs the relevant cryptographic and logical operations.

There must be at least two instances of Federation installed – one Asserting Party and one or more Relying Parties. Figure 1-2 demonstrates that there can be multiple Relying Parties per Asserting Party. Each instance consists of Federation Web Services, Policy

Server Option Pack (AKA Federation) and the FSS Applet UI. These components work together to establish a user's identification and authentication to the TOE. Once established, SiteMinder's Policy Server and Web Agent are used in conjunction with the Federation components in order to provide access to protected resources. The Composed TOE relies on the Operational Environment to provide the Key Store, User Store, Session Store and Policy Store. To gain a better understanding of how SiteMinder provides access to protected resources through the use of rules and policies, see the CA SiteMinder Web Access Manager r12 SP1 CR3 Security Target v1.0. The two instances of Federation are identical, but their provider role determines the functions they perform.

The Asserting Party is the central repository for maintaining user information. One of the critical features of the SAML authentication schemes is to map remote users at a producing authority to local users at the consuming authority. The mapping is defined as part of the authentication scheme configuration. User mapping information enables the authentication scheme to locate the correct user record for authentication. For more information on user mapping, refer to Section 1.4.3. In the evaluated configuration, the specific repository is an LDAP directory that is used as a SiteMinder User Store.

A Relying Party is simply a destination for the user to access (though an Asserting Party can be a destination as well). By authenticating through the Asserting Party via username, password, and certificate (depending on the authentication scheme used), Federation establishes sessions with all Relying Parties defined by the federation. As a result, the user always authenticates through the Asserting Party. The user either authenticates directly through the Asserting Party, or the Relying Party provides a redirect to the Asserting Party in order to provide initial authentication. The TOE facilitates the authentication process by allowing the Policy Server to determine the authentication scheme being used to identify the user. Once the user has been authenticated, SiteMinder determines whether or not the requested operations will be allowed or denied. (see Section 9.1.2 for more information on authentication).

The TOE uses two separate means of communications in order to establish a connection between the Asserting Party and the Relying Party – frontchannel and backchannel. As shown in Figure 1-1, frontchannel refers to the user's web browser being redirected between the asserting and the Relying Party. Alternatively, backchannel refers to the Service Provider directly "reaching back" to the Identity Provider in order to get information from it. HTTP-POST uses the frontchannel, while HTTP-artifact uses the backchannel.

Single Sign-on (SSO) is employed to allow transactions across partner websites in multiple domains. The TOE is capable of handling multiple user sessions between partner federated entities. It does so by controlling access to resources based on user information passed from a federated partner. Single logout (SLO) is also used which allows for users to log out from all federated entities by logging out at a single location. The session's cookies are destroyed at all associated partner federated entities once logged out – this creates a secure means to protect user's data. A federation is not limited to a single Asserting Party (Identity Provider/Producer) and Relying Party (Service

Provider/Consumer). As shown in Figure 1-2, multiple Relying Parties can be configured so that single sign-on can be established between more than two devices. In addition, one or more Attribute Authorities may be used to provide elements of the user DN which are not stored on the Asserting Party itself.

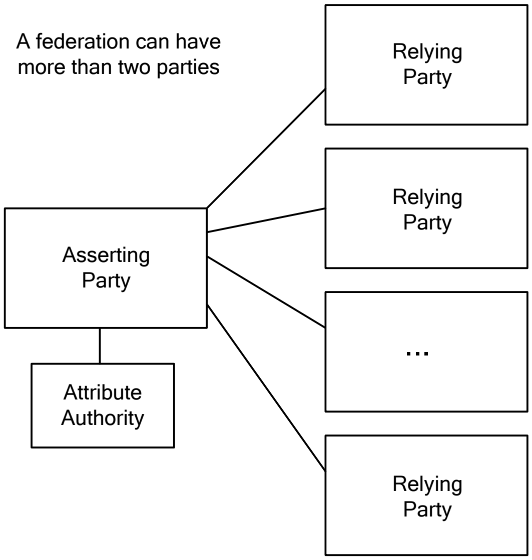

Figure 1-2: Possible Configuration of Federation Security Services

1.4 Federation Security Services Concepts

1.4.1 SAML

The Organization for the Advancement of Structured Information Standards (OASIS) Security Assertion Markup Language (SAML) standard defines an XML-based framework for describing and exchanging security information between on-line business partners. This security information is expressed in the form of portable SAML assertions that applications working across security domain boundaries can trust. The OASIS SAML standard defines precise syntax and rules for requesting, creating, communicating, and using these SAML assertions.

An assertion contains several different internal statements about authentication, authorization, and attributes. SAML defines two browser-based protocols that specify how SAML assertions are passed between partners to facilitate single sign-on. The profiles are:

- Browser/artifact profile—defines a SAML artifact as a reference to a SAML assertion.
- Browser/POST profile—returns a response that contains an assertion.

Note: For SAML 2.0, the artifact and POST profiles are referred to as HTTP bindings.

There are several drivers behind the adoption of the SAML standard, including:

- Single Sign-On: Over the years, various products have been marketed with the claim of providing support for web-based SSO. These products have relied on browser cookies to maintain user authentication state information so that re-authentication is not required each time the web user accesses the system. However, since browser cookies are never transmitted between DNS domains, the authentication state information in the cookies from one domain is never available to another domain. Therefore, these products have supported multi-domain SSO (MDSSO) through the use of proprietary mechanisms to pass the authentication state information between the domains. Business partners have heterogeneous environments that make the use of proprietary protocols impractical for MDSSO. SAML solves the MDSSO problem by providing a standard vendor-independent grammar and protocol for transferring information about a user from one web server to another independent of the server DNS domains.

- Federated identity: When online services wish to establish a collaborative application environment for their mutual users, not only must the systems be able to understand the protocol syntax and semantics involved in the exchange of information; they must also have a common understanding of who the user is that is referred to in the exchange. Users often have individual local user identities within the security domains of each partner with which they interact. Identity federation provides a means for these partner services to agree on and establish a common, shared name identifier to refer to the user in order to share information about the user across the organizational boundaries. The user is said to have a federated identity when partners have established such an agreement on how to refer to the user. From an administrative perspective, this type of sharing helps to reduce identity management costs as multiple services do not need to independently collect and maintain identity-related data (e.g. passwords, identity attributes). In addition, administrators of these services do not have to manually establish and maintain the shared identifiers; rather control for this resides with the user.

- Web services and other industry standards: SAML allows for its security assertion format to be used outside of a “native” SAML-based protocol context. This modularity has proved useful to other industry efforts addressing authorization services (IETF, OASIS), identity frameworks, web services (OASIS, Liberty Alliance), etc. The OASIS WS-Security Technical Committee has defined a profile for how to use SAML’s rich assertion constructs within a WS-Security security token that is used, for example, to secure web service SOAP message exchanges. In particular, the advantage offered by the use of a SAML assertion is that it provides a standards-based approach to the exchange of information, including attributes that are not easily conveyed using other WS-Security token formats.

The TOE supports both SAML 1.1 and 2.0. See Table 1-4 for the differences between SAML 1.1 and 2.0. For more information on SAML, see the OASIS Security Assertion Markup Language (SAML) V2.0 Technical Overview v14.

SAML 1.1	SAML 2.0
Uses the term “Producer”	Uses the term “Identity Provider”
Uses the term “Consumer”	Uses the term “Service Provider”
Request must be initiated by Producer end	Request may be initiated by either end
Does not support single logout	Does support single logout
Cannot use attribute authority	Can use attribute authority
Protects assertion with signature	Protects assertion with encryption and signature

Table 1-4: SAML Properties

1.4.2 Entities

In a federated network, there is an entity that generates assertions (Asserting Party). Assertion information is gathered both from the Asserting Party itself as well as one or more potential third party attribute authorities. Assertions contain information about a user whose identity is maintained locally at the federated entity that generates them. There is another entity that uses the assertions (Relying Party) to authenticate a user and to establish a session for the user. Depending on the protocol, these two entities are named differently, but the functions they serve are the same.

Protocol	Generates Assertions (Entity)	Consumes Assertions (Entity)
SAML 1.1	Producer	Consumer
SAML 2.0	Identity Provider (IdP)	Service Provider (SP)

Table 1-5: Entities

A federated entity may be both an Asserting Party (producer/IdP) and a Relying Party (consumer/SP).

Figure 1-3 below illustrates the message flow for an SP-initiated SSO exchange. Note that this example is based on SAML 2.0 so the terminology is specific to that standard. However, the general flow of information is the same for the other standards. In such an exchange, the user attempts to access a resource on the SP, sp.example.com. However, he does not have a current logon session on this federated entity and his federated identity is managed by his IdP, idp.example.org. The user is sent to the IdP to log on and the IdP provides a SAML web SSO assertion for the user's federated identity back to the SP. For this specific use case, the HTTP Redirect Binding is used to deliver the SAML <AuthnRequest> message to the IdP and the HTTP POST Binding is used to return the SAML <Response> message containing the assertion to the SP. For more information on SAML, refer to the OASIS Security Assertion Markup Language (SAML) V2.0 Technical Overview v14 guide.

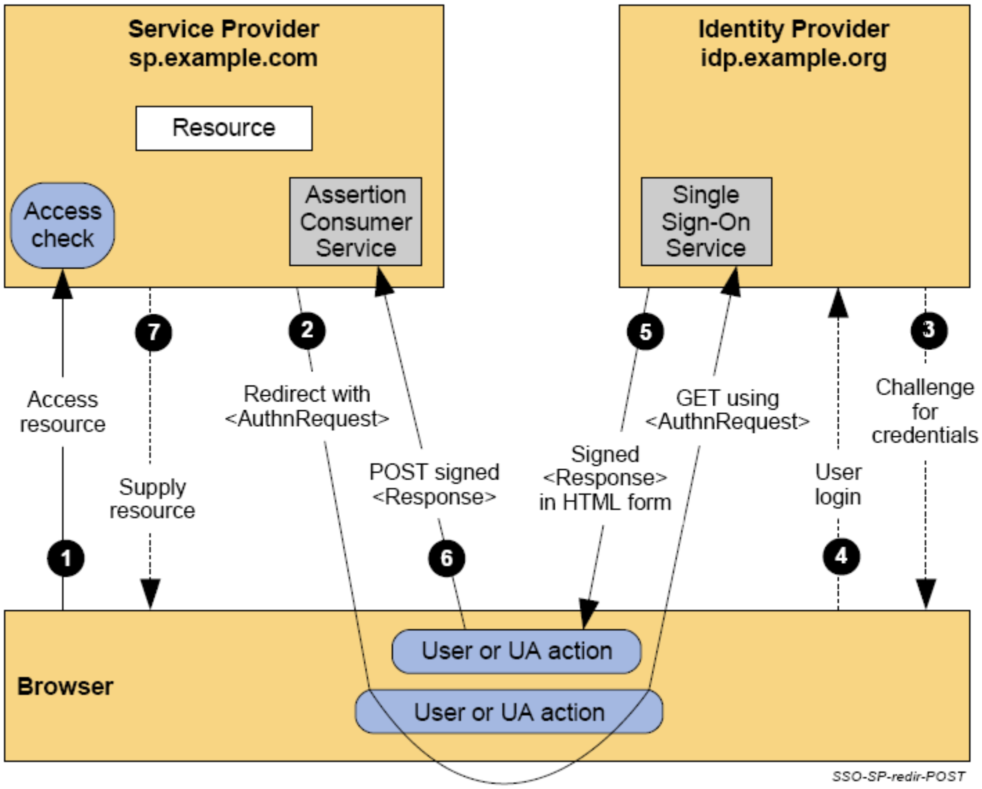

Figure 1-3: SP-Initiated SSO with Redirect and POST Bindings

1.4.3 User Mapping

User mapping is the ability to establish a relationship between a user identity at one business and a user identity at another business. This relationship is established by mapping remote users at an Asserting Party to local users at a Relying Party. There are two types of mapping:

- One-to-one mapping maps a unique remote user store entry at the producing authority to a unique user entry at the consuming authority. One-to-one mapping is often referred to as account linking, as it links an account at a producing authority federated entity to an account at a consuming authority.

- N-to-one mapping maps a group of remote user store entries to a single local profile entry. N-to-one mapping allows several user records at a producing authority to be mapped to one user record or profile at a consuming authority. An administrator at the consuming authority uses this type of mapping to define access control for a group of remote users, without having to maintain a record for each remote user.

1.4.4 Federated Single Sign-on with Security Zones

A SiteMinder environment can be set up to include a Web application environment for web service protection and a federation environment for federated resource protection. This method makes a SiteMinder deployment more efficient. Certain Federation Security

Services features require a persistent user session because the SAML assertion must be stored in the session store, which is connected to the Policy Server. These features include:

- Artifact Single sign-on - For SAML 1.1 and SAML 2.0, the SAML assertion can be stored in a persistent session that is later retrieved by the consuming federated entity.

Note that the ST defines authentication schemes as SAML 1.1 artifact, SAML 1.1 POST, and SAML 2.0 template. When the SAML 2.0 template is selected, the administrator has the ability to select either POST or artifact bindings. As a result, artifact single sign-on can be used regardless of SAML version.

- Federated Logout - For SAML 2.0 Single Logout at IdP and SP federated entity. Partner data is stored in a persistent user session to facilitate notification of partners during a federated logout.

Use of persistent user sessions slows down performance because of the calls to the session store to retrieve assertions or handle log-off requests. However, security zones eliminate the need for a persistent user session for requested Asserting Party-side applications protected by a Web Agent. A security zone is a segment of a single cookie domain, used as a method of partitioning applications to permit different security requirements for resource access. All applications in a single zone permit single sign-on to one another. If an application is in another zone, single sign-on is determined by the configured trust relationship. Security zones are a part of SiteMinder's single sign-on feature and are implemented by SiteMinder Web Agents.

1.4.5 Affiliate Domain

An affiliate domain is a logical grouping of federated entities associated with one or more user stores. The affiliate domain not only contains federated entities but it also defines which user stores are associated with the domain. To authenticate a user, SiteMinder must have access to the user store where a user record is defined. The Policy Server locates a user record by querying the user stores specified in the affiliate domain's search order. The search order is defined when adding user store connections to an affiliate domain. The option of shifting the order of directories exists.

1.5 TOE Type

CA SiteMinder Federation Security Services r12 SP1 CR3 provides the following: federated authentication. The TOE type is: Web Access Control. This was chosen because the base component of this composed TOE (CA SiteMinder Web Access Manager r12 SP1 CR3) was evaluated as a Web Access Control product.

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 Evaluated Components of the TOE

The evaluated components are identified as follows:

- Federation Web Services
- Policy Server Option Pack (AKA Federation)
- Federation Security Services Applet UI

2.1.1 Federation Web Services

The Federation Web Services (FWS) application is installed on a server that has a connection to a SiteMinder Policy Server. Federation Web Services consists of the following:

- Single Sign On (SSO)
- Single Log Out (SLO)
- Artifact Resolution
- Assertion Consumer
- Inter-site Transfer
- SAML Credential Collector
- Assertion Retriever
- Agent API
- Attribute Service
- Auth URL JSP

The Federation Web Services and the SiteMinder Web Agent support the following protocols:

- SAML browser artifact protocol
- SAML POST profile protocol

Note: To install Federation Web Services, a web or application server is needed. In the evaluated configuration, a web server running ServletExec 6.0 will be used. To ensure security, SSL must be enabled on both this web server and the web server where Federation Web Services is installed.

2.1.1.1 SAML Browser Artifact Protocol

For the SAML browser artifact protocol, the Federation Web Services application includes the following services:

- Assertion Retrieval Service (SAML 1.1)--A producer federated entity component. This service handles a SAML request for the assertion that corresponds to a

SAML artifact by retrieving the assertion from the SiteMinder session store. The assertion retrieval request and response behavior is defined by the SAML specification. Note: The assertion retrieval service is used only by the SAML artifact profile, not by the SAML POST profile.

- SAML Credential Collector (SAML 1.1)--A consumer federated entity component that receives a SAML artifact or an HTTP form with an embedded SAML response and obtains the corresponding SAML assertion. The credential collector issues SiteMinder cookies to a user's browser.

- Intersite Transfer Service (SAML 1.1)--For SAML POST profile, a producer federated entity component that transfers a user from the producer federated entity to a consumer federated entity. For SAML artifact profile, the same function is performed by the Web Agent, which acts as the Intersite Transfer Service.

- Single Sign-on Service (SAML 1.1)--This service implements processing for a Producer to process an AuthnRequest message and gather the necessary Consumer configuration information to authenticate the user, redirect the user to the Web Agent to authenticate, and invokes the assertion generator to obtain an assertion that is passed back to the Consumer.

2.1.1.2 SAML POST Profile Protocol

For SAML POST Profile protocol, the Federation Web Services application includes the following services:

- Artifact Resolution Service (SAML 2.0)--An Identity Provider-side service that corresponds to the SAML 2.0 authentication using the HTTP-artifact binding. This service retrieves the assertion stored in the SiteMinder session store at the Identity Provider. This is a Federation-specific service. Note: The artifact resolution service is used only by the HTTP-artifact binding.

- Assertion Consumer Service (SAML 2.0)--A Service Provider component that receives a SAML artifact or an HTTP form with an embedded SAML response and obtains the corresponding SAML assertion. The Assertion Consumer Service issues SiteMinder cookies to a user's browser. Note: The Assertion Consumer Service will accept an AuthnRequest with an AssertionConsumerServiceIndex value of 0. All other values for this setting will be denied.

- AuthnRequest Service (SAML 2.0)--This service, a Federation-specific service, is a servlet deployed as part of the Federation Web Services application for SAML 2.0. It implements processing for a Service Provider to generate an <AuthnRequest> message to authenticate a user for cross-domain single sign-on. This message contains information that enables the Federation Web Services application to redirect the user's browser to the single sign-on service at the Identity Provider. The AuthnRequest service is used for single sign-on using the POST or artifact binding. Note: The format of the AuthnRequest message issued by this service is specified in the Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0.

- Single Sign-on Service (SAML 2.0)--This service implements processing for an Identity Provider to process an AuthnRequest message and gather the necessary SP configuration information to authenticate the user, redirect the user to the Web Agent to authenticate, and invokes the assertion generator to obtain an assertion that is passed back to the Service Provider.
- Single Logout Service (SAML 2.0)--This service implements processing of single logout functionality, which is initiated by an Identity Provider or a Service Provider.

2.1.2 Policy Server Option Pack (AKA Federation)

The PS Option Pack enables user store connectivity, authentication functions, and session store abilities. In order to manage the settings of Federation, the Federation Security Services Applet UI must be used.
Consists of the following:

- SAML Assertion Generator
- Configuration Services
- SAML Auth Schemes
- Tunnel Services
- smkeydatabase

2.1.2.1 SAML Assertion Generator

The SAML assertion generator creates an assertion for a user who has a session at an Asserting Party. When a request for a SAML assertion is made, the Web Agent invokes the SAML assertion generator, which creates an assertion based on the user session and information configured in the policy store. The assertion is then handled according to the authentication profile or binding configured, as follows:

- SAML artifact profile/binding--assertion is placed in the SiteMinder session store and a reference to the assertion is returned to the Web Agent in the form of a SAML artifact.
- SAML POST profile/binding--assertion is returned via the user’s browser as a SAML response embedded in a HTTP form.

The Web Agent is responsible for sending the SAML artifact or SAML response to the federated entity that will consume the assertion accordance with the SAML profile or binding. At the Relying Party, the SAML 1.1 credential collector or the SAML 2.0

Assertion Consumer Service must be available to process the SAML artifact or response message.

2.1.2.2 SAML Authentication Schemes

SiteMinder supports the following authentication schemes:

- SAML 1.1 Artifact
- SAML 1.1 POST
- SAML 2.0 Template

Each authentication scheme enables a federated entity to consume SAML assertions. Upon receiving an assertion, the authentication scheme validates the assertion, maps assertion data to a local user, and establishes a SiteMinder session at the federated entity consuming the assertion. One of the critical features of the SAML authentication schemes is to map remote users at a Relying Party to local users at the Asserting Party. The mapping is defined as part of the authentication scheme configuration. User mapping information enables the authentication scheme to locate the correct user record for authentication.

The SAML authentication schemes are installed by the Policy Server. After installation, the administrator uses the FSS Applet UI to define and configure these schemes and use them to define authentication for federated resources.

2.1.3 Federation Security Services Applet UI

The FSS Applet UI is a web application that uses the HTTP protocol to administer and manage the configuration of entities and partnerships and various server settings. Many of the SiteMinder functions can also be accomplished using the WAM Admin UI, but the FSS Applet UI serves as the primary interface for administrators in the evaluated configuration. During configuration of the TOE, the terminal(s) used by the administrator to run the FSS Applet UI must be registered by using the WAM Admin UI.

2.2 Operational Environment Components

The TOE relies on the Operational Environment to provide the following:

- SiteMinder Policy Store
- SiteMinder Key Store
- SiteMinder User Store
- SiteMinder Session Store

The information held in these stores is needed for proper use of the TOE. However, since the stores are part of the Operational Environment, they are not included in the evaluation.

2.2.1 Policy Store

The Extensible Policy Store stores all TOE data objects. The TOE allows for either a SQL Server, LDAP, or Oracle database to be used for the policy store. Administrators of the TOE can access the Policy Store as well as create and manage tables in the database. It also stores authentication credentials for the FSS Applet UI. Access to the database is controlled by the usage of a username and password. The Policy Store is accessed by both Policy Server and PS Option Pack (Federation) components.

2.2.2 Key Store

The database that contains keys used to encrypt cookies created by the SiteMinder Web Agent. The Key Store is accessed by the Policy Server.

2.2.3 Session Store

The session store is the database that stores user sessions, SAML attributes and SAML assertions. It is accessed by the Policy Server.

2.2.4 User Store

The User Store is the database that stores user data, including organizational information, user and group attributes, and credentials such as passwords. The User Store is accessed by the Policy Server.

2.3 SiteMinder Components

The following components are SiteMinder components and have previously been evaluated at EAL3. These components are part of the base component of the Composed TOE and are relied upon by the dependent component in order for it to function properly:

- SiteMinder Web Agent
- SiteMinder Policy Server
- WAM Admin UI

2.3.1 SiteMinder Web Agent

A SiteMinder Web Agent is a software component that controls 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 authorize 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 a user should have access.

- Provide information to a Web application that dictates how content is presented to the user (policy-based personalization) and how to deliver access privileges.
- Ensure a user's ability to securely access information. Web Agents store contextual information about user access privileges in a session cache. Performance is 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 users to re-authenticate.

A value for the Web Agent configuration parameter DefaultAgentName must be configured for all Relying Party Web Agents. This value specifies a Web Agent identity. Additionally, the specified Agent identity must be included in the Resource Filter of the realm that protects the target resource. The DefaultAgentName parameter is configured in the Agent Configuration Object or the local Agent configuration file. Omitting the DefaultAgentName parameter or using the value specified in the AgentName parameter in the realm resource filter causes SAML 1.1 authentication to fail, regardless of the single sign-on profile.

The SiteMinder Web Agent has previously been evaluated and is not included in this evaluation. For more information, see the CA SiteMinder Web Access Manager r12 SP1 CR3 Security Target v1.0.

2.3.2 SiteMinder Policy Server

The SiteMinder Policy Server provides functions such as the authentication schemes (SAML 1.1 artifact, SAML 1.1 Post, SAML 2.0 Template) and the Assertion Generator. When a 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 Policy Server installed at the Asserting Party, includes the assertion generator component. The assertion generator creates SAML assertions, which are XML documents that contain authentication information about a user. For the SAML artifact profile, after an assertion is generated, it is stored by the session store until it is requested by the Relying Party. The AMAssertionGenerator.properties file is required for operation of the Assertion Generator. It contains parameters that the Assertion Generator uses to generate SAML assertions. If any changes are made to the AmAssertionGenerator.properties file, the changes will not be picked up by the Policy Server until it is restarted.

The Policy Server has previously been evaluated and is not included in this evaluation. For more information, see the CA SiteMinder Web Access Manager r12 SP1 CR3 Security Target v1.0.

2.3.3 SiteMinder WAM Administrative UI

The SiteMinder WAM Administrative UI is a web-based administration console for SiteMinder that is installed independent of the Policy Server. An administrator uses the

SiteMinder WAM Administrative UI to view, modify, and delete all Policy Server objects except those related to Federation Security Services. While the SiteMinder tasks that Federation Security Services builds upon can be configured via the SiteMinder WAM Administrative UI or the FSS Applet UI, those which apply specifically to Federation Security Services (such as configuring affiliates and SAML authentication schemes) must be handled using the FSS Applet UI.

The SiteMinder WAM Admin UI is used to set up the Policy Server and to get the base component up and running. However, the SiteMinder WAM Admin UI is not used in the evaluated configuration.

2.4 Excluded from the TOE

- SAML affiliate agents – A stand-alone component that provides authentication and session management capabilities to a consumer federated entity that does not use a SiteMinder Policy Server and Web Agent. The SAML Affiliate Agent only supports SAML 1.0 and it is not FIPS-compatible.

- Secure Proxy Server Federation Gateway - The SiteMinder Secure Proxy Server (SPS) Federation gateway offers a proxy-based solution to access control in a federated network. Unlike a traditional proxy, which serves a group of users requesting Internet resources, the SPS Federation gateway is a reverse proxy, meaning it acts on behalf of users requesting resources from an enterprise. The SPS Federation gateway is a self-contained system; it has its own servlet engine and web server built in to the system and relies on its proxy engine to handle access requests from federated partners to protected resources. Enhancing SPS to work as a federation gateway allows quick deployments. As a component of SiteMinder Federation security services, the SPS Federation gateway can replace the Web Agent and Federation Web Services to provide the services of the Federation Web Services application. A single SPS Federation gateway can limit the amount of configuration required for access to resources by limiting the need for many Web Agents. Note: The Secure Proxy Server is a separately-licensed product from SiteMinder.

- WS-Federation Authentication Scheme - Active Directory Federation Services (ADFS) is Microsoft’s Web Services-based solution for federation and single sign-on (SSO). ADFS runs on Windows Server 2003 R2 and accomplishes SSO by letting partners securely share a user's identity information and access rights across a secure network. This feature has been excluded from the evaluation because it is no longer supported by the development consortium.

These optional components provide no added security related functionality and are therefore not included in the evaluated configuration.

2.5 Physical Boundary

The FSS Applet UI will be deployed in an environmental web server on the same machine as the Policy Server. Supported web servers are the same as those which are supported for use with Web Agents. The TOE is installed on top of existing instances of SiteMinder. The Policy Server and Web Agent will be deployed on separate machines in each instance. The Web Agent and FWS will be deployed on the same machine in each instance.

The following table illustrates the minimum requirements needed to install FWS on a Windows or UNIX/Linux 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
Web Server	IIS 6.0 or ASF Apache 2.2 on Microsoft Windows 2003 SP2 SunOne Web Server 7.0 or ASF Apache 2.2 on Red Hat Advanced Server 4.0	SunOne Web Server 7.0 or ASF Apache 2.2 on Solaris 10
Servlet Container	Servlet Exec 6.0 on Microsoft Windows 2003 SP2 Servlet Exec 6.0 on Red Hat Advanced Server 4.0	Servlet Exec 6.0 on Solaris 10 Sparc

Table 2-1: Minimum Requirements for Installation of SiteMinder Web Agent with FWS (Web Agent Option Pack)

Table 2-2 lists the supported operating systems for the TOE.

Component	TOE Version	Platforms
Policy Server FSS Applet UI Web Agent PS Option Pack Federation Web Services (FWS)	r12 SP1 CR3	Linux Red Hat Advanced Server 4.0
		Microsoft Windows 2003 SP2
		Solaris 10
Policy Store User Store	r12 SP1 CR3	SunOne LDAP 5.2 on Red Had Advanced Server 4.0
		Windows 2003 Active Directory on Microsoft Windows 2003 SP2
		SunOne LDAP 5.2 on Solaris 10

Key Store Session Store	r12 SP1 CR3	Oracle 10g R2 on Red Hat Advanced Server 4.0
		Oracle 10g R2 on Microsoft Windows 2003 SP2
		Oracle 10g R2 on Solaris 10
Web Servers	r12 SP1 CR3	SunOne Web Server 7.0 on Red Hat Advanced Server 4.0
		ASF Apache 2.2 on Red Hat Advanced Server 4.0
		IIS 6.0 and ASF Apache 2.2 on Microsoft Windows 2003 SP2
		SunOne 6.1 SP2 and ASF Apache 2.2 on Solaris 10
Servlet Container	r12 SP1 CR3	ServletExec 6.0

Table 2-2: Supported Operational Environment Components for the TOE

In addition to the platforms listed in Table 2-2, the following non-TOE software is required to run the TOE:

- SSL v3.0 implementation
- Transport standards HTTP
- Web browser software

Table 2-3 illustrates the minimum requirements needed to install the Policy Server on a Windows, Linux, or 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	Ensure that LDAP directory server being used as a policy store is supported	Ensure that LDAP directory server being used as a policy store is supported
Web Server	IIS 6.0 or ASF Apache 2.2 on Microsoft Windows 2003 SP2 SunOne Web Server 7.0 or ASF Apache 2.2 on Red Hat Advanced Server 4.0	SunOne Web Server 7.0 or ASF Apache 2.2 on Solaris 10

Table 2-3: Specifications for SiteMinder Policy Server with PS Option Pack

2.6 Logical Boundary

The logical boundary of the TOE includes the CA SiteMinder r12 SP1 CR3 with Federation Security Services software. The TOE enforces the following security functions as described below: Security Audit, Encrypted Communications, Identification and Authentication, TOE Access, Security Management, and Protection of the TSF and Trusted Path/Channel.

2.6.1 Security Audit

The TOE generates data for log files that contain auditing information about the events that occur within the system, including the startup and shutdown of audit functions and all user and Administrator actions on the TOE. Based on the content of these logs, the TOE is able to associate the event with the user or administrator that caused the event. The audit data generated by the TOE is stored in SiteMinder log files, so audit storage and review is not the responsibility of the TOE.

The TOE employs trace logging in order to monitor the performance of the Web Agent and Policy Server. These logging mechanisms provide comprehensive information about the operation of SiteMinder processes so performance and troubleshooting issues can be analyzed.

The component that controls the trace messages for Federation services at the Policy Server is the Fed_Server component. This component monitors activity for the assertion generator and the SAML authentication schemes. FWS logging can be configured by modifying the parameters of the LoggerConfig.properties file.

The following subcomponents are available for the Fed_Server component:

- Configuration --monitors SAML 2.0 Relying Party configuration activity.
- Assertion_Generator--watches the activity for the SAML 1.1 and 2.0 assertion generators.
- Auth_Scheme--monitors the activity of the SAML 1.1 or SAML 2.0 authentication schemes.
- Saml_Requester--watches SAML Requester activity
- Attribute_Service--watches the Attribute Service activity

Note that authorization events generated by Federation are recorded using the same audit mechanisms used for SiteMinder.

2.6.2 Encrypted Communications

The TOE uses symmetric encryption keys generated by SiteMinder to encrypt and decrypt sensitive data passed between TOE components, between TOE and SiteMinder components, and between TOE and users/Administrators. The TOE uses imported public keys and digital signatures in order to protect and validate SAML assertions passed to Relying Parties. 128-bit AES is provided for symmetric key cryptography, and RSA and X509 are used for public keys and digital signing. Once keys are used by the TOE, they are destroyed by the key zeroization capabilities of SiteMinder.

Symmetric keys used by the TOE are stored in the SiteMinder Key Store. Public key and signature information used by the TOE is stored in a separate database called the smkeydatabase, which is installed during the initial setup of the TOE. Operations on this database such as importing certificates are performed using a tool called smkeytool.

Because the FSS Applet UI is accessed from an environmental web browser, encrypting communications between the administrator's browser and the TOE is the responsibility of the environment. However, once the applet has been launched, it uses 128-bit AES cryptography to communicate back to the Policy Server.

2.6.3 Identification & Authentication

Users and Administrators must be identified and authenticated to the TOE prior to being able to perform any action on the TOE. Users must re-authenticate when certain conditions are met. Administrators must choose a user authentication scheme supported by the TOE and configure the scheme to be used by the TOE. During configuration, Administrators need to define a method for the authentication scheme to look up a user in a user store, where security attributes are maintained for users. These attributes are associated with subjects acting on behalf of the user.

The TOE relies on the Asserting Party's SM User Store to identify and authenticate a user based on a pre-configured component of their DN, which is then passed along to all federated Relying Parties. Locating the user in the user store is the process of disambiguation. This is the user for which the system generates a session during the authentication process.

The TOE uses the rules enforced by SiteMinder for the realm containing the protected targeted resource. However, it uses its own authentication schemes based on SAML. The rule is triggered during the authorization process by SiteMinder to receive SAML attributes from the session store. The attributes are supplied as HTTP header variables and used by a client application. The headers are then returned to the customer's application.

2.6.4 TOE Access

The TOE enacts the process of single logout (SLO) (also known as cross-domain single signout) which results in the simultaneous end of all sessions for a particular user, thereby ensuring security. These sessions must be associated with the browser that initiated the logout. Single logout does not necessarily end all sessions for a user. For example, if the user has two browsers open, that user can establish two independent sessions. Only the session for the browser that initiates the single logout is terminated at all federated entities for that session. The session in the other browser will still be active. Single logout is triggered by a user-initiated logout.

Session establishment can also be denied by the TOE. When an assertion (SAML 2.0) is successfully validated, the SAML 2.0 authentication scheme writes assertion data in the expiry data table with a key of the assertion ID and an expiration time. If the scheme cannot write to the table in the session store, the SAML 2.0 authentication scheme denies the authentication in the same manner as an invalid assertion.

2.6.5 Security Management

The TOE provides for two distinct roles – Users and Administrators. Users are those who attempt to access federated resources. Once Federation successfully authenticates the user, SiteMinder enforces authorization to the protected federated resources via the user's web browser. Administrators are those who have full privileges to manage and maintain data as well as create, edit, and delete objects from the Federation Security Services (FSS) Applet UI. Administrators are the only users allowed to modify the following functions:

- SAML affiliations for SAML 2.0
- SAML authentication schemes
- Affiliate domains, which contain:
    - Affiliates (SAML 1.1)
    - Service Providers (SAML 2.0)
- SiteMinder objects and policies

For a complete description of administrative capabilities, please refer to Section 9.1.4

2.6.6 Protection of the TSF and Trusted Path/Channel

All communication between users/Administrators and the TOE are secured via an environmental trusted path using SSL v3.0. All communication between TOE components and, as well as communication between Federation Web Services and the Policy Server, utilize a proprietary algorithm from SiteMinder known as the TLI handshake. This is used by the Asserting Party to establish communications with its Relying Parties for single sign-on. For more information on encryption used by SiteMinder, see the CA SiteMinder Web Access Manager r12 SP1 CR3 Security Target v1.0.

Protecting the Federation Web Services application at the Asserting Party ensures that the services that make up the application are secure. The policies for the Federation Web Services application are created automatically. However, to enforce protection and to specify who can access Federation Web Services, Administrators must authenticate to the FSS Applet UI where they manage the TOE.

There is a pre-configured policy that uses the Basic over SSL authentication scheme to protect the Assertion Retrieval Service. When configuring the policy for the client certificate authentication scheme, this policy is created for a different realm than the realm that uses the Basic over SSL scheme. For protection of data transmitted between separate parts of the TOE, SSL v3.0 is used.

In order to establish single sign-on between the Asserting Party and Relying Party, the SSO bindings supported by the Relying Party need to be specified. In the FSS Applet UI, the SSO tab allows single sign-on to be configured using the artifact or POST binding. This enforces the single use assertion policy for POST binding to prevent the replaying of a valid assertion. When replay is detected, the TOE denies the request and returns an error to the user.

3 Conformance Claims

3.1 CC Version

This ST is compliant with *Common Criteria for Information Technology Security Evaluation*, CCMB-2009-07-004, Version 3.1 Revision 3, July 2009.

3.2 CC Part 2 Conformant

This ST and Target of Evaluation (TOE) is Part 2 conformant for EAL 3 to include all applicable NIAP and International interpretations through 15 December 2009.

3.3 CC Part 3 Augmented Plus Flaw Remediation

This ST and Target of Evaluation (TOE) is Part 3 augmented plus flaw remediation for EAL3 to include all applicable NIAP and International interpretations through 15 December 2009.

3.4 PP Claims

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

3.5 Package Claims

This TOE has a package claim of EAL3 augmented ALC_FLR.1 and ASE_TSS.2 with CAP-B for integration with validated product CA SiteMinder Web Access Manager r12 SP1 CR3.

3.6 Package Name Conformant or Package Name Augmented

This Target of Evaluation (TOE) has a package claim of EAL3 augmented, and CAP-B augmented for integration with validated product CA SiteMinder Web Access Manager r12 SP1 CR3. Both package claims have been augmented with ALC_FLR.1 and ASE_TSS.2, while CAP-B is also augmented with ALC_CMC.3, ALC_CMS.3.

3.7 Conformance Claim Rationale

Since this ST references the TOE as a Composed TOE, a conformance to CAP-B in addition to EAL3 is claimed. The TOE is comprised of two separate products: CA SiteMinder Web Access Manager r12 SP1 CR3 (SiteMinder) and CA Federation Security Services (Federation). Together, the Composed TOE is known as CA SiteMinder Federation Security Services r12 SP1 CR3. Federation is provided as an option to the SiteMinder product and although SiteMinder is able to run independently of Federation, Federation cannot run independently of SiteMinder. The definition of a Composed Assurance Package states that a CAP is applied to a composed TOE, which is comprised of components that have been (or are going through) component TOE evaluation. SiteMinder was previously evaluated at EAL3. Federation is being evaluated at the same EAL level. In addition, CAP-B Security Assurance Requirements (SARs) are satisfied in this evaluation as required for a Composed TOE. As stated in CC Part 3, "CAP-B permits a conscientious developer to gain maximum assurance from understanding, at a subsystem level, the affects of interactions between component TOEs integrated in the composed TOE, whilst minimizing the demand of involvement of the base component developer."

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 basic. The following are threats addressed 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.

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 unauthorized access to resources protected by the TOE by bypassing identification and authentication countermeasures.

T.UNAUTH Users or administrators could gain unauthorized 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 authorized administrators will be assigned to install, configure and manage the TOE and the security of the information it contains.

A.PATCHES 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 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.

A.PASSWORD It is assumed that users will select strong passwords to be enforced by SiteMinder and will protect their authentication data.

4.3.2 Physical Assumptions

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

4.3.3 Connectivity Assumptions

A.FILESYS The administrator will secure the underlying Operating System and data stores in order to protect the files used by the TOE.

4.4 Security Objectives

4.4.1 Security Objectives for the TOE

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

O.AUDIT The TOE will provide measures for recording security relevant events that will assist local OS administrators 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 their claimed identity prior to allowing SiteMinder the ability to enforce access to resources protected by SiteMinder. 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.

4.4.2 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 data stores protect the files used by the TOE.

OE.NOEVIL All Administrators 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 on an isolated network with no connections to other networks.

OE.PASSWORD Users shall ensure that they choose strong passwords to be enforced by SiteMinder and that they protect their authentication data.

5 Extended Security Functional Requirements

5.1 Extended Security Functional Requirements for the TOE

There are no extended Security Functional Requirements for the TOE included in this ST.

6 Extended Security Assurance Requirements

There are no extended Security Assurance Requirements in this ST.

7 Security Functional Requirements

7.1 Security Functional Requirements for the TOE

Security Function	Security Functional Components
Security Audit (FAU)	FAU_GEN.1 Audit data generation
	FAU_GEN.2 User identity association
Cryptographic Support (FCS)	FCS_COP.1(1) Cryptographic operation
	FCS_COP.1(2) Cryptographic operation
	FCS_COP.1(3) Cryptographic operation
	FCS_COP.1(4) Cryptographic operation
Identification and Authentication (FIA)	FIA_ATD.1 (1) User attribute definition
	FIA_ATD.1 (2) User attribute definition
	FIA_UAU.2 (1) User authentication before any action
	FIA_UAU.2 (2) User authentication before any action
	FIA_UAU.5 Multiple authentication methods
	FIA_UAU.6 Re-authenticating
	FIA_UID.2 (1) User identification before any action
	FIA_UID.2 (2) User identification before any action
	FIA_USB.1 User-subject binding
Security Management (FMT)	FMT_MOF.1 Management of security functions behavior
	FMT_MTD.1 Management of TSF data
	FMT_SMF.1 Specification of Management Functions
Protection of the TSF (FPT)	FPT_ITC.1 Inter-TSF trusted channel
	FPT_ITT.1 Basic internal TSF data transfer protection
	FPT_RPL.1 Replay detection
TOE Access (FTA)	FTA_SSL.4 User-initiated termination
	FTA_TSE.1 TOE session establishment

Table 7-1: Security Functional Requirements for the TOE

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) [***Authentication, authorization, access to URLs, management operations listed in Table 9-1, whether allowed or denied by the TOE***].

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 (if applicable), 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: *Federation start-up and shutdown are implicit in the SiteMinder start-up and shutdown. SiteMinder relies on the Operational Environment to provide an accurate timestamp. The timestamp is included in the audit records.*

7.1.1.2 FAU_GEN.2-Refinement User identity association

Hierarchical to: No other components

FAU_GEN.2.1 For audit events resulting from actions of identified users ***or administrators***, the TSF shall be able to associate each auditable event with the identity of the user ***or administrator*** that caused the event.

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

7.1.2 Class FCS: Cryptographic Support

All cryptography for this product has only been asserted as tested by the vendor. The testing of the specific cryptographic algorithms will not be tested as part of this evaluation.

7.1.2.1 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: [***FIPS Pub 197***].

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]
FCS_CKM.4 Cryptographic key destruction

Application note: *This operation is for the encryption of communications between Tunnel Services and Federation Web Services and between Tunnel Services and the FSS Applet UI.*

7.1.2.2 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: [***RFC 3602***].

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]
FCS_CKM.4 Cryptographic key destruction

Application Note: *This SFR supports the AES key derived from the Session Ticket Key used for defining administrator sessions*

7.1.2.3 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 **[*RSA*]** and cryptographic key sizes **[*1024 bits*]** that meet the following: **[*PKCS #1, PKCS #5*].**
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] FCS_CKM.4 Cryptographic key destruction
Application Note:	*This SFR supports the public key encryption of SAML 2.0 assertions.*

7.1.2.4 FCS_COP.1 (4) Cryptographic operation

Hierarchical to:	No other components.
FCS_COP.1.1 (4)	The TSF shall perform [***digital signing***] in accordance with a specified cryptographic algorithm [***X.509 V1, V2, and V3***] and cryptographic key sizes [***Base64, DER, PEM***] that meet the following: [***PKCS #8, PKCS #12***].
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] FCS_CKM.4 Cryptographic key destruction
Application Note:	*This SFR supports the signing of SAML assertions.*

7.1.3 Class FIA: Identification and Authentication

7.1.3.1 FIA_ATD.1 (1) User attribute definition

Hierarchical to:	No other components.
FIA_ATD.1.1 (1)	The TSF shall maintain the following list of security attributes belonging to individual users: [***SAML attributes***].

Dependencies: No dependencies.

7.1.3.2 FIA_ATD.1 (2)-Refinement User attribute definition

Hierarchical to: No other components.

FIA_ATD.1.1 (2) Refinement: The TSF shall maintain the following list of security attributes belonging to individual ~~users~~ ***Administrators***: [***username, password, hostname, pass phrase***].

Dependencies: No dependencies.

7.1.3.3 FIA_UAU.2 (1) User authentication before any action

Hierarchical to: FIA_UAU.1 Timing of authentication

FIA_UAU.2.1 (1) The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Dependencies: FIA_UID.1 Timing of identification

7.1.3.4 FIA_UAU.2 (2)-Refinement User authentication before any action

Hierarchical to: FIA_UAU.1 Timing of authentication

FIA_UAU.2.1 (2) Refinement: The TSF shall require each ~~user~~ Administrator to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that ~~user~~ Administrator.

Dependencies: FIA_UID.1 Timing of identification

7.1.3.5 FIA_UAU.5 Multiple authentication mechanisms

Hierarchical to: No other components.

FIA_UAU.5.1 The TSF shall provide: [***the following schemes: Basic Over SSL Template, X509 Client Cert Template over SSL using SAML 1.1 HTTP-artifact, SAML 1.1 HTTP-POST, or SAML 2.0 Template***] to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the [***rules describing how the multiple authentication mechanisms provide authentication***].

Dependencies: No dependencies.

Application Note: These SFRs refer to the ability for one of many authentication schemes to be specified, and to the ability for the TSF to authenticate a user based on the data passed through any of these schemes.

7.1.3.6 FIA_UAU.6 Re-authenticating

Hierarchical to: No other components.

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions [***the user has been authenticated to an Asserting Party, authentication for a Relying Party is being requested, and an affiliation between the Asserting Party and Relying Party has been established***].

Dependencies: No dependencies

7.1.3.7 FIA_UID.2 (1) User identification before any action

Hierarchical to: FIA_UID.1 Timing of identification

FIA_UID.2.1 (1) The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Dependencies: No dependencies

7.1.3.8 FIA_UID.2 (2)-Refinement User identification before any action

Hierarchical to: FIA_UID.1 Timing of identification

FIA_UID.2.1 (2) Refinement: The TSF shall require each ~~user~~ ***Administrator*** to be successfully identified before allowing any other TSF-mediated actions on behalf of that ~~user~~ ***Administrator***.

Dependencies: No dependencies

7.1.3.9 FIA_USB.1 User-subject binding

Hierarchical to: No other components

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on behalf of that user: [***SAML attributes***].

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on behalf of users: [***the Asserting Party's Relying Party object defines the attributes which are associated with SAML assertions and where they are derived (either from some component of the user DN or a static assignment)***].

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on behalf of users: [***no rules***].

Dependencies: FIA_ATD.1 User attribute definition

Application Note: *Once an assertion has been defined, it is static. If some attribute has changed, a new assertion must be generated in order to incorporate it.*

7.1.4 Class FMT: Security Management

7.1.4.1 FMT_MOF.1 Management of security functions behavior

Hierarchical to: No other components.

FMT_MOF.1.1 The TSF shall restrict the ability to [**determine the behaviour of, disable, enable, modify the behaviour of**] the functions [***the following functions: SAML affiliations, SAML authentication schemes, affiliate domains, SiteMinder objects and policies***] to [***Administrators***].

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

Application Note: *"SiteMinder Objects" refers to rules, realms, domains, and other components of SiteMinder which can be managed by the FSS Applet UI. Refer to Table 7-2 for applicable functions.*

7.1.4.2 FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

FMT_MTD.1.1 The TSF shall restrict the ability to [**create, view, modify, delete**] the [***Policy Server objects listed in Table 7-2 Management of TSF Data***] to [***Administrator***s].

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

7.1.4.3 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: [***the operations listed in Table 7-2 Management of TSF Data on the Policy Server objects listed in Table 7-2 Management of TSF Data***].

Dependencies: No dependencies.

Application Note: *The highlighted items in grey are objects which can be managed using the WAM Administrative UI as well if desired.*

Operations	**Policy Server Objects**	**Interface**
Create/view/modify/delete	Agents	FSS Applet Admin UI
Create/view/modify/delete	**Agent Configuration Objects**	FSS Applet Admin UI
Create/view/modify/delete	**Host Configuration Objects**	FSS Applet Admin UI
Create/view/modify/delete	Policy domains	FSS Applet Admin UI
Create/view/modify/delete	Affiliate domains	FSS Applet Admin UI
Create/view/modify/delete	Authentication Schemes	FSS Applet Admin UI
Create/view/modify/delete	SAML Affiliations	FSS Applet Admin UI
Create/view/modify/delete	rules (in managed domains)	FSS Applet Admin UI

Create/view/modify/delete	**policies (in managed domains)**	FSS Applet Admin UI
Create/view/modify/delete	Affiliates	FSS Applet Admin UI
Create/view/modify/delete	SAML Service providers	FSS Applet Admin UI

Table 7-2: Management of TSF Data

7.1.4.4 FMT_SMR.1 Security roles

Hierarchical to: No other components.

FMT_SMR.1.1 The TSF shall maintain the roles [***Administrator and user***].

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

Dependencies: FIA_UID.1 Timing of identification

7.1.5 Class FPT: Protection of the TSF

7.1.5.1 FPT_ITC.1 Inter-TSF trusted channel

Hierarchical to: No other components.

FPT_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 communicated data from modification or disclosure.

FPT_ITC.1.2 The TSF shall permit [**the TSF, another trusted IT product**] to initiate communication via the trusted channel.

FPT_ITC.1.3 The TSF shall initiate communication via the trusted channel for [***single sign-on***].

Dependencies: No dependencies

7.1.5.2 FPT_ITT.1 Basic internal TSF data transfer protection

Hierarchical to: No other components.

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

Dependencies: No dependencies.

Application Note: *"Separate parts of the TOE" refers to backchannel communication directly between multiple Federation Web Services instances.*

7.1.5.3 FPT_RPL.1 Replay detection

Hierarchical to: No other components.

FPT_RPL.1.1 The TSF shall detect replay for the following entities: [***HTTP POST bindings for SAML***].

FPT_RPL.1.2 The TSF shall perform [***denial of the request and return of HTTP 500 error***] when replay is detected.

Dependencies: No dependencies.

7.1.6 Class FTA: TOE Access

7.1.6.1 FTA_SSL.4 User-initiated termination

Hierarchical to: No other components.

FTA_SSL.4.1 The TSF shall allow user-initiated termination of the user's own interactive session.

Dependencies: No dependencies.

Application Note: The user's own interactive session refers to all of the federated partnerships established as a result of single sign-on. This requirement encapsulates single logout behaviour.

7.1.6.2 FTA_TSE.1 TOE session establishment

Hierarchical to: No other components.

FTA_TSE.1.1 The TSF shall be able to deny session establishment based on [***malformed XML, invalid SAML assertion***].

Dependencies: No dependencies.

7.2 Security Functional Requirements for the Base Component

Because the TOE is a composed TOE of multiple distinct products, not all necessary requirements are imposed on the dependent component. The following SFRs from the base component (CA SiteMinder Web Access Manager R12-SP3) apply to the evaluated composed TOE in order for it to accomplish its intended behavior of offering federated single sign-on to an environment that controls access to web-based resources:

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_ACF.1(1) 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.6 Re-authenticating
Security Management	FMT_MSA.1(2) Management of security attributes
	FMT_MSA.2 Secure security attributes

Table 7-3: SFRs for the Base Component

For information regarding these requirements, refer to section 7 of *CA SiteMinder Web Access Manager R12 SP1-CR3 Security Target, Version 1.0.* This document can be found on the VPL under VID 10317 or at the following URL: http://www.niap-ccevs.org//cc-scheme/st/st_vid10317-st.pdf.

The base component claims several SFRs which are not being considered as a part of this evaluation. Listed below are the SFRs which have been omitted and a rationale for their exclusion in the composed TOE deployment.

FDP_ACC.1(2): The WAM UI has been previously validated for management of the base component, but it is not used to manage the dependent component. Management functions of the composed TOE necessarily require usage of the FSS Applet UI at least in part. Usage of this interface is not governed by a policy since there is a single superuser account which is allowed access to it. WAM UI functions affect the behavior of the base component only.

FDP_ACF.1(2): Refer to FDP_ACC.1(2) above.

FIA_UAU.1: In the composed TOE deployment, authentication is required before making any action against federated resources. The reason for this is that single sign-on must be first established before any access control decisions can be made. Therefore, FIA_UAU.1 in the base component is superseded by FIA_UAU.2 in the dependent component.

FIA_UID.2: This requirement is claimed by both the base and dependent component. It is disregarded here to avoid redundancy.

FIA_UAU_EXT.5: The base component specifies a set of authentication schemes to be used when authenticating against a single Policy Server. This ST specifies the same set of authentication schemes but also defines the method used to establish single sign-on between federated entities. Because of this, FIA_UAU_EXT.5 in the base component is superseded by FIA_UAU.5 in the dependent component.

FMT_MSA.1(1): In the composed TOE, security attributes are managed by the FSS Applet UI instead of the WAM UI. While the previously-validated WAM UI can still be used to modify specific properties of the base component, the FSS Applet UI is able to manage the federation aspects of the composed TOE as well as a sufficient subset of the base component's behavior to demonstrate that access control to protected resources is enforced. The WAM Admin UI expresses no novel behavior in the composed TOE, so the focus of the evaluation is placed on the FSS Applet UI. As a result, the set of managed attributes and the policy which governs their use is superseded by FMT_SMF.1 in the dependent component.

FMT_MSA.1(3): Because the WAM Admin UI is not used to perform functions specific to the composed TOE, the Administrator Policy does not apply, and this requirement is not necessary.

FMT_MSA.3: Refer to FMT_MSA.1(3) above.

FMT_SMF.1: Refer to FMT_MSA.1(1) above.

FMT_SMR.1: Refer to FDP_ACC.1(2) above.

FPT_FLS.1: In the composed TOE deployment, two or more Policy Servers control access to separate enclaves in the environmental network. This is a contrast from a potential mode of operation for the base component which allows two Policy Servers to control access to the same resources in a high-availability configuration. As a result, high availability is not examined in the composed TOE.

FRU_FLT.1: Refer to FPT_FLS.1 above.

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

7.3.1 Assignments Made

An assignment allows the specification of parameters and is specified by the ST author in [***italicized bold text***].

7.3.2 Iterations Made

Iteration allows a component to be used more than once with varying operations and is identified with the iteration number within parentheses after the short family name.

7.3.3 Selections Made

A selection allows the specification of one or more items from a list and is specified by the ST author in [**bold text**].

7.3.4 Refinements Made

A refinement allows the addition of details and is identified with "Refinement:" right after the short name. ~~The old text is shown with a strikethrough~~ and ***the new text is specified by 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 and for CAP-B as required for a Composed TOE. In addition, CAP-B is augmented with ALC_CMC.3 and ALC_CMS.3 since these assurance components are required in order to satisfy EAL3.

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 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 initialization process is secure.

ADV_ARC.1.4C The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5C The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

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

8.1.2 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 SFR-enforcing actions 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 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 The developer shall provide the design of the TOE. ADV_TDS.2.2D The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

ADV_TDS.2.1C The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.2.2C The design shall identify all subsystems of the TSF.

ADV_TDS.2.3C The design shall describe the behavior of each SFR non-subsystem of the TSF in detail sufficient to determine that it is SFR non-interfering.

ADV_TDS.2.4C The design shall describe the SFR-enforcing behavior of the SFR-enforcing subsystems.

ADV_TDS.2.5C The design shall summarize the SFR-supporting and SFR-non-interfering behavior of the SFR-enforcing subsystems.

ADV_TDS.2.6C The design shall summarize the behavior of the SFR-supporting subsystems.

ADV_TDS.2.7C The design shall provide a description of the interactions among all subsystems of the TSF.

ADV_TDS.2.8C The mapping shall demonstrate that all TSFIs trace to the behavior described in the TOE design that they invoke.

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 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 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 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 and provide 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 and provide 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 and provide 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 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 objective 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.2.1D The developer shall provide a statement of security requirements.

ASE_REQ.2.2D The developer shall provide a security requirements rationale.

ASE_REQ.2.1C The statement of security requirements shall describe the SFRs and the SARs.

ASE_REQ.2.2C All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.

ASE_REQ.2.3C The statement of security requirements shall identify all operations on the security requirements.

ASE_REQ.2.4C All operations shall be performed correctly.

ASE_REQ.2.5C Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.

ASE_REQ.2.6C The security requirements rationale shall trace each SFR back to the security objectives for the TOE.

ASE_REQ.2.7C The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.

ASE_REQ.2.8C The security requirements rationale shall explain why the SARs were chosen.

ASE_REQ.2.9C The statement of security requirements shall be internally consistent.

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

8.4.6 Security Problem Definition (ASE_SPD.1)

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

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

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

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

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

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

8.4.7 TOE Summary Specification (ASE_TSS.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 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 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 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, and 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.

8.7 Composition

The SARs in this section satisfy CAP-B SARs for a Composed TOE. All additional security assurance requirements for CAP-B have been satisfied by the EAL3 security assurance requirements.

8.7.1 Composition Rationale (ACO_COR.1)

ACO_COR.1.1D The developer shall provide composition rationale for the base component.

ACO_COR.1.1C The composition rationale shall demonstrate that a level of assurance at least as high as that of the dependent component has been obtained for the support functionality of the base component, when the base component is configured as required to support the TSF of the dependent component.

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

8.7.2 Rigorous Interface Testing (ACO_CTT.2)

ACO_CTT.2.1D The developer shall provide composed TOE test documentation.

ACO_CTT.2.2D The developer shall provide base component interface test documentation.

ACO_CTT.2.3D The developer shall provide the composed TOE for testing.

ACO_CTT.2.4D The developer shall provide an equivalent set of resources to those that were used in the base component developer's functional testing of the base component.

ACO_CTT.2.1C The composed TOE and base component interface test documentation shall consist of test plans, expected test results and actual test results.

ACO_CTT.2.2C The test documentation from the developer execution of the composed TOE tests shall demonstrate that the TSF behaves as specified and is complete.

ACO_CTT.2.3C The test documentation from the developer execution of the base component interface tests shall demonstrate that the base component interface relied upon by the dependent component behaves as specified and is complete.

ACO_CTT.2.4C The base component shall be suitable for testing.

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

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

ACO_CTT.2.3E The evaluator shall test a subset of the TSF interfaces of the composed TOE to confirm that the composed TSF operates as specified.

8.7.3 Basic Evidence of Design (ACO_DEV.2)

ACO_DEV.2.1D The developer shall provide development information for the base component.

ACO_DEV.2.1C The development information shall describe the purpose and method of use of each interface of the base component used in the composed TOE.

ACO_DEV.2.2C The development information shall provide a high-level description of the behavior of the base component, which supports the enforcement of the dependent component SFRs.

ACO_DEV.2.3C The development information shall show correspondence between the interfaces, used in the composed TOE, of the base component and the dependent component to support the TSF of the dependent component.

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

ACO_DEV.2.2E The evaluator shall determine that the interface description provided is consistent with the reliance information provided for the dependent component.

8.7.4 Basic Reliance Information (ACO_REL.1)

ACO_REL.1.1D The developer shall provide reliance information of the dependent component.

ACO_REL.1.1C The reliance information shall describe the functionality of the base component hardware, firmware and/or software that is relied upon by the dependent component TSF.

ACO_REL.1.2C The reliance information shall describe all interactions through which the dependent component TSF requests services from the base component.

ACO_REL.1.3C The reliance information shall describe how the dependent TSF protects itself from interference and tampering by the base component.

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

8.7.5 Composition Vulnerability Analysis (ACO_VUL.2)

ACO_VUL.2.1D The developer shall provide the composed TOE for testing.

ACO_VUL.2.1C The composed TOE shall be suitable for testing.

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

ACO_VUL.2.2E The evaluator shall perform an analysis to determine that any residual vulnerability identified for the base and dependent components are not exploitable in the composed TOE in its operational environment.

ACO_VUL.2.3E The evaluator shall perform a search of public domain sources to identify possible vulnerabilities arising from use of the base and dependent components in the composed TOE operational environment.

ACO_VUL.2.4E The evaluator shall perform an independent vulnerability analysis of the composed TOE, using the guidance documentation, reliance information and composition rationale to identify potential vulnerabilities in the composed TOE.

ACO_VUL.2.5E The evaluator shall conduct penetration testing, based on the identified vulnerabilities, to demonstrate that the composed TOE is resistant to attacks by an attacker with basic attack potential.

9 TOE Summary Specification

9.1 TOE Security Functions

The following sections identify the security functions of the TOE. They include Security Audit, Identification and Authentication, TOE Access, Security Management, Encrypted Communications, and Protection of the TSF.

9.1.1 Security Audit

The TOE relies on SiteMinder to generate log files that contain auditing information about the events that occur within the system, including the startup and shutdown of audit functions, authentication, authorization, access to URLs, and management operations listed in Table 9-1. The success and failure of each of these events is audited by the TOE. The TOE is able to associate each event with the user or administrator that caused the event. Specifically, each audit log is recorded with the user ID of the user that caused the event as a parameter. Once a user is authenticated to Federation, the user has sessions on both the asserting party and relying party. Because of this, when the user accesses information on one domain, the audit record will include their identity on the corresponding domain.

Audit logs are stored in the operational environment and can be viewed locally on the system a Policy Server is installed on by using that system's root account. The audit logs are identical to the ones used by SiteMinder. This is because authentication events handled by the TOE are an extension of those made available by SiteMinder but are still fundamentally the same type of events.

The Federation Security Services components log specific events to monitor and debug activity across the federated network. The following types of logs are created by the TOE:

- Federation Web Services log--Error messages of Federation Web Services are logged in affwebserv.log. Trace messages of Federation Web Services are logged in FWSTrace.log. The LoggerConfig.properties file lets Administrators enable logging so the Federation Web Services application can record assertion retrieval, session management, notification alert information, and trace messages related to Federation Web Services.

- Policy Server log--Logs the results of calls from the SAML assertion generator and SAML authentication scheme activities in smtracedefault.log.

In SAML 1.1, entities that consume assertions are referred to as consumers (Relying Party) and entities that generate assertions are referred to as producers (Asserting Party). In SAML 2.0, entities that consume assertions are referred to as Service Providers (Relying Party) and entities that generate assertions are referred to as Identity Providers (Asserting Party). Administrators have the ability to track user activity at the Relying Party and administrator activity at the Asserting Party. The Asserting Party receives notifications from the Relying Party about which resources that user has accessed. When the user accesses specific URLs at the Relying Party, the Relying Party has the ability to notify the Asserting Party. The Asserting Party logs this activity and uses the information for auditing or reporting purposes.

The TOE is equipped with the ability to monitor the performance of the Web Agent and Policy Server. These logging mechanisms provide comprehensive information about the operation of SiteMinder processes so performance and troubleshooting issues can be analyzed.

The Federation Web Services (FWS) application represents the federation client. The component that controls the trace messages and monitors FWS activity is the Fed_Client component.

Within the Fed_Client component, the following sub components are included:

- single sign-on--monitors single sign-on activity
- single logout--monitors requests for single logout.
- administration--watches administration-related messages
- request--monitors request and authentication activity.
- general--monitors activity not covered by the other subcomponents.
- configuration--monitors SAML 2.0 Relying Party configuration messages

FWS uses the common tracing facility used by the Web Agent to log trace messages. The following files are used to set up trace logging:

- Trace Configuration File - the configuration file that determines which components and events FWS monitors. The default file is FWSTrace.conf.
- Trace Log File – the output file for all the logged messages. A name and the location must be provided for this file in the Web Agent configuration file.
- Web Agent configuration file or Agent Configuration Object--contains the logging parameters that enable logging and format the log. It does not define message content.

The component that controls the trace messages for Federation services at the Policy Server is the Fed_Server component. This component monitors activity for the assertion generator and the SAML authentication scheme and can be configured during initial setup of the TOE. The following subcomponents are available for the Fed_Server component:

- Configuration --monitors SAML 2.0 Relying Party configuration activity.
- Assertion_Generator--watches the activity for the SAML 1.1 and 2.0 assertion generators.

- Auth_Scheme--monitors the activity of the SAML 1.1 or SAML 2.0 authentication schemes.
- Saml_Requester--watches SAML Requester activity.
- Attribute_Service--watches the Attribute Service activity.

In order to have changes that were made to the Federation configuration at the Asserting Party/IdP or Relying Party/SP appear in the trace logs, the Federation Web Services cache must first be flushed.

Trace Logging templates also come pre-installed with Federation Web Services that allow administrators to use templates to collect the data being written to log instead of creating an original trace configuration file. The templates have the ability to collect the following information (dependent upon which template is chosen):

- Default – collects only specified data
- Collects SSO messages
- Collects SLO messages
- Collects Asserting Party Profile messages

In SAML 1.1, entities that consume assertions are referred to as consumers. Administrators have the ability to track user activity at the consumer. Producers receive notifications from the consumers about which resources that user has accessed. When the user accesses specific URLs at the consumer, the consumer has the ability to notify the producer. The producer logs this activity and uses the information for auditing or reporting purposes.

9.1.2 Identification and Authentication

Users and Administrators must be identified and authenticated to the TOE prior to being able to perform any action on a resource protected by the TOE. By authenticating through the Asserting Party via username, password, and certificate (depending on authentication scheme used), Federation establishes sessions with all Relying Parties defined by the federation.

If a user's initial request is to the Asserting Party, they are prompted to authenticate. If their initial request is to the Relying Party, the Relying Party determines they do not yet have a session and redirects them to the Asserting Party, where they are prompted to authenticate. As a result, the user always authenticates through the Asserting Party. The TOE facilitates the authentication process by allowing the Policy Server to determine the authentication scheme being used to identify the user. Once the user has been authenticated, SiteMinder determines whether or not the requested operations will be allowed or denied. The TOE automatically re-authenticates users when they have been authenticated to an Asserting Party, authentication for a Relying Party is being requested, and an affiliation between the Asserting Party and Relying Party has been established.

9.1.2.1 Disambiguation

The TOE maintains SAML attributes for users, and username, password, hostname and pass phrase for Administrators. The SAML attributes which are maintained for users are components of the user DN required by the Relying Party to disambiguate the user. These attributes are associated with subjects acting on behalf of the user after they authenticate. For the initial association of security attributes with subjects acting on behalf of users, the Asserting Party's Relying Party object defines the attributes which are associated with SAML assertions and if they are derived from a component of the user DN or from a static assignment.

When configuring an authentication scheme, a method for the authentication scheme to look up a user in a user store is defined. This is defined by mapping the elements of the user DN which are used to identify the user on both the Asserting Party and Relying Party. Locating the user in the user store is the process of disambiguation. This is the user for which the system generates a session during the authentication process.

There are two ways of configuring user disambiguation:

- Locally, as part of the authentication scheme
- By selecting a configured SAML affiliation

The TOE uses one of Basic over SSL, Windows Authentication , or X.509 certificates to establish an initial connection to the Asserting Party. The mechanism for this is identical to that of the base component, SiteMinder Web Access Manager R12 SP1-CR3. The user provides their credentials to the Asserting Party and is authenticated. Once this has happened, the Asserting Party composes different SAML assertions using different DN attributes based on its defined affiliations and sends them to all necessary Relying Parties.

9.1.2.2 Affiliate Domains

An affiliate domain is a logical grouping of federated entities (Asserting Party or Relying Party) associated with one or more user stores. The affiliate domain not only contains federated entities but it also defines which user stores are associated with the domain. In order for Federation to authenticate a user, SiteMinder must have access to the user store where a user record is defined. The Policy Server locates a user record by querying the user stores specified in the affiliate domain's search order. The search order is defined when adding user store connections to an affiliate domain. The order of directories can be shifted. Affiliate domains are configured by using the FSS Applet UI.

9.1.2.3 Authentication Schemes

The TOE supports multiple authentication schemes in order for SiteMinder to prevent unauthorized access to protected services and resources. The following authentication methods are included in the evaluated configuration:

- Basic Over SSL Template
- X509 Client Cert Template
- Windows Authentication Template

Note that Basic over SSL or X509 Client Cert authentication is performed in conjunction with one of the SAML templates. The first category (Basic Over SSL or X509 Client Cert) is the mechanism by which a user establishes their session at the Asserting Party, and the second category (SAML Artifact or POST) is the mechanism by which the Asserting Party propagates this session to the Relying Party.

Each authentication scheme enables a federated entity to consume SAML assertions. Upon receiving an assertion, the authentication scheme validates the SAML assertion, maps assertion data to a local user, and establishes a SiteMinder session at the federated entity consuming the assertion. One of the critical features of the SAML authentication schemes is to map remote users at an Asserting Party to local users at the Relying Party. The mapping is defined as part of the authentication scheme configuration. User mapping information enables the authentication scheme to locate the correct user record for authentication.

The SAML authentication schemes are installed by the Policy Server. After installation, the administrator uses the FSS Applet UI to define and configure these schemes and uses them to protect specific resources.

9.1.2.4 SAML 1.1 Authentication Schemes

There are two SAML 1.1 authentication methods available for configuration with SiteMinder:

- SAML Artifact profile
- SAML POST profile

The SAML-based authentication schemes let a consumer in a federated network authenticate a user. When SAML 1.1 is used, the authentication must be initiated at the consumer. It enables cross-domain single sign-on by consuming a SAML assertion and establishing a SiteMinder session. After the user is identified, SiteMinder authorizes the user for specific resources. A consumer is a federated entity that uses a SAML 1.1 assertion to authenticate a user. A producer is a federated entity that generates SAML 1.1 assertions.

Note: A federated entity may be both a SAML producer and a SAML consumer.

The following illustration shows the major components for authentication at the consumer federated entity.

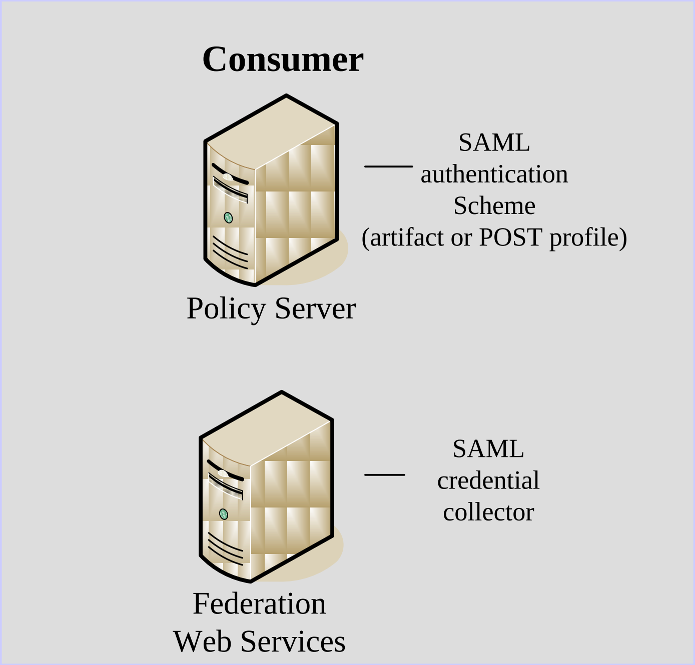

Figure 9-1: Authenticating at a Consumer Federated Entity

The SAML 1.1 authentication scheme is configured at the consumer-side Policy Server and is invoked by the SAML credential collector. The SAML credential collector is a component of the Federation Web Services application and is installed on the consumer-side Web Agent or SPS Federation gateway. The credential collector obtains information from the SAML authentication scheme at the Policy Server, then uses that information to access a SAML assertion. The SAML assertion becomes the user's credentials to log into the Policy Server at the consumer federated entity. The user is authenticated by Federation. SiteMinder then determines if authorization is allowed. If so, the browser is then redirected to the target resource.

9.1.2.5 SAML 1.1 POST Profile Authentication Scheme

The following illustration shows how the SAML POST profile authentication scheme processes requests.

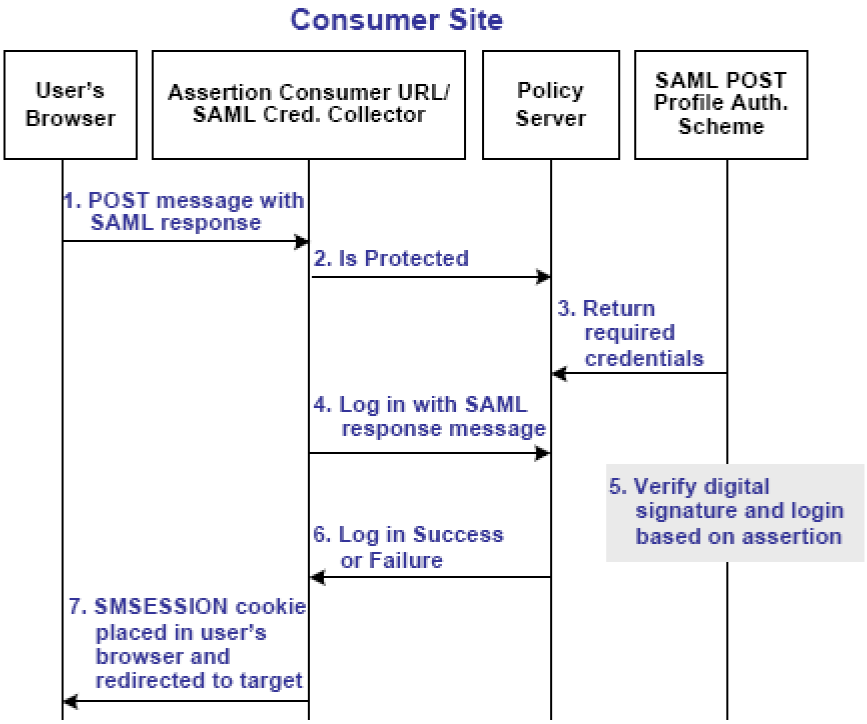
Consumer Site
User's Browser
Assertion Consumer URL/ SAML Cred. Collector
Policy Server
SAML POST Profile Auth. Scheme
1. POST message with SAML response
2. Is Protected
3. Return required credentials
4. Log in with SAML response message
5. Verify digital signature and login based on assertion
6. Log in Success or Failure
7. SMSESSION cookie placed in user's browser and redirected to target

Figure 9-2: SAML 1.1 POST profile authentication scheme

Unless otherwise stated, the following process takes place at the consumer federated entity:

- A user's browser POSTs an HTML form to the Assertion Consumer URL (which is the URL for the SAML credential collector). This form contains a SAML response message and target URL originally generated at the producer.

- The SAML credential collector makes a call to the Policy Server to determine if the target resource is protected.

- The Policy Server replies that the target URL is protected by the SAML POST profile authentication scheme. This indicates to Federation Web Services application that a signed response from the POSTed form is the expected credential for the login call.

- The SAML credential collector makes a login call to the Policy Server, passing the digitally signed SAML response as credentials.

- The SAML POST profile authentication scheme verifies the signature and other fields of the response and the assertion.

- If the checks succeed and the user is found in the directory, then authentication succeeds. If any of the checks fail, authentication fails.

- Assuming login succeeds, the SAML credential collector sends information to the Web Agent to create an SMSESSION cookie. This is put in the user's browser, and then redirects the user to the target resource, which is protected by the Relying Party's Web Agent. If the login fails, the credential collector redirects the user to the configured No Access URL.

9.1.2.6 SAML 1.1 Artifact Authentication Scheme

The following illustration shows how the SAML 1.1 artifact authentication scheme processes requests. This illustration shows the SAML 1.1 artifact authentication functional model.

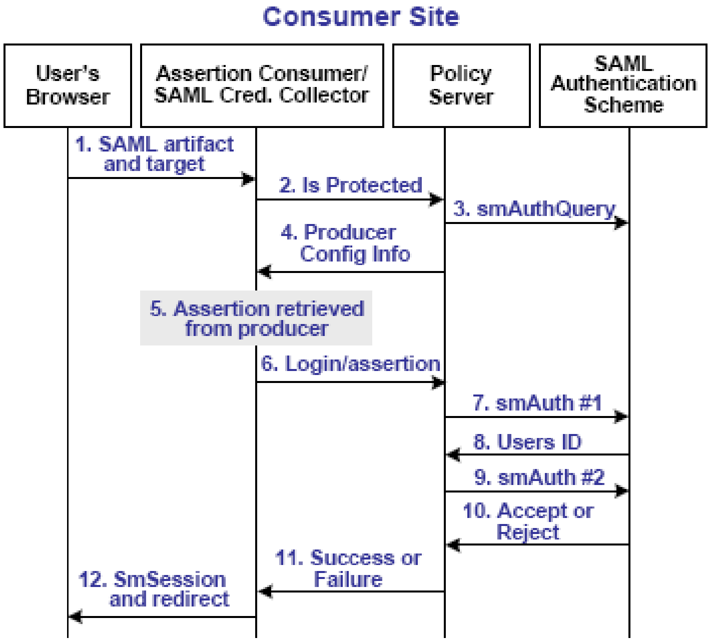

Figure 9-3: SAML 1.1 artifact profile authentication scheme

Unless otherwise stated, all activity in this process occurs at the consumer federated entity:

- A user is redirected to the SAML credential collector with a SAML artifact and a target URL. The artifact and target is originally generated from the SiteMinder Web Agent at the producer federated entity.

- The SAML credential collector calls the Policy Server to check if the requested resource is protected. This resource is protected by the SAML artifact authentication scheme.

- The Policy Server passes the necessary data to the SAML artifact authentication scheme, which extracts the producer configuration information, such as the affiliate name and password.

- The Policy Server returns the producer configuration information to the SAML credential collector. This information enables the credential collector servlet to call a producer federated entity and retrieve a SAML assertion.

- The SAML credential collector takes the data from the Policy Server and uses it to retrieve the SAML assertion stored at the producer Policy Server’s Session Store.
- Once an assertion is returned, the credential collector uses the assertion as credentials to establish a session.
- The Policy Server makes the initial user disambiguation call to the SAML authentication scheme.
- Using the authentication scheme data and the assertion, the scheme locates the user and returns a unique identifier for the user to the credential collector.
- The Policy Server makes the second user authentication call to the authentication scheme.
- The scheme validates the SAML assertion and returns an accept or reject message to the Policy Server.
- The Policy Server sends the success or failure message to the credential collector.
- If the login succeeds, the SAML credential collector creates a session cookie and places it in the user’s browser then redirects the user to the target resource. If the login fails, the credential collector servlet redirects the user to a No Access URL.

9.1.2.7 Protecting a Resource with SAML 1.1 Authentication Scheme

At the consumer, a SAML 1.1 artifact or POST profile authentication scheme must be configured for each producer that generates assertions. After that authentication scheme is created, it is used to protect Federation resources. To protect a Federation resource with a SAML authentication scheme:

- Create a realm that uses the SAML authentication scheme. The realm is the collection of target resources being requested by users. There are two ways to set-up a realm that includes a SAML authentication scheme:
    - A unique realm can be created for each authentication scheme already configured.
    - A single target realm can be configured that uses a custom authentication scheme to dispatch requests to the corresponding SAML authentication schemes. Configuring one realm with a single target for all producers simplifies configuration of realms for SAML authentication.
- After configuring a realm, configure an associated rule and optionally, a response.
- Group the realm, rule, and response into a policy that protects the target resource. Each target URL in the realm is also identified in an intersite transfer URL. An inter-site transfer URL redirects a user from the producer to the consumer, and the target URL is specified in the URL’s TARGET variable. At the producer

federated entity, an administrator needs to include this URL in a link so that this link the user gets redirected to the consumer.

9.1.2.8 SAML 2.0 Authentication Schemes

Any federated entity consumes assertions to authorize users. The SAML 2.0 authentication scheme lets a Service Provider in a federated network authenticate a user. When SAML 2.0 is used, the authentication is initiated at the Identity Provider or the Service Provider. SAML 2.0 enables cross-domain single sign-on by consuming a SAML assertion from an Identity Provider, identifying a user, and establishing a SiteMinder session. After a SiteMinder session is established, SiteMinder authorizes the user for specific resources.

The SAML 2.0 authentication scheme first determines a LoginID from the assertion. The LoginID is a SiteMinder-specific term that identifies the user. By default, the LoginID is extracted from the Name ID value in the assertion. The LoginID is also obtained from elsewhere in the assertion by specifying an Xpath query.

After the authentication scheme determines the LoginID, it uses the LoginID to locate a user in the user store. By default, the LoginID is passed back to the Policy Server to locate the user in the user store. For example, if an LDAP user store is configured to search for users based on the UID attribute the Policy Server searches for the user based on the UID. Also, a search specification is configured to locate a user in the user store. The search specification controls how the LoginID is used in the query to locate a user.

Note: A federated entity may be both an Identity Provider and a Service Provider.

The major components for SAML 2.0 authentication are shown in the following illustration:

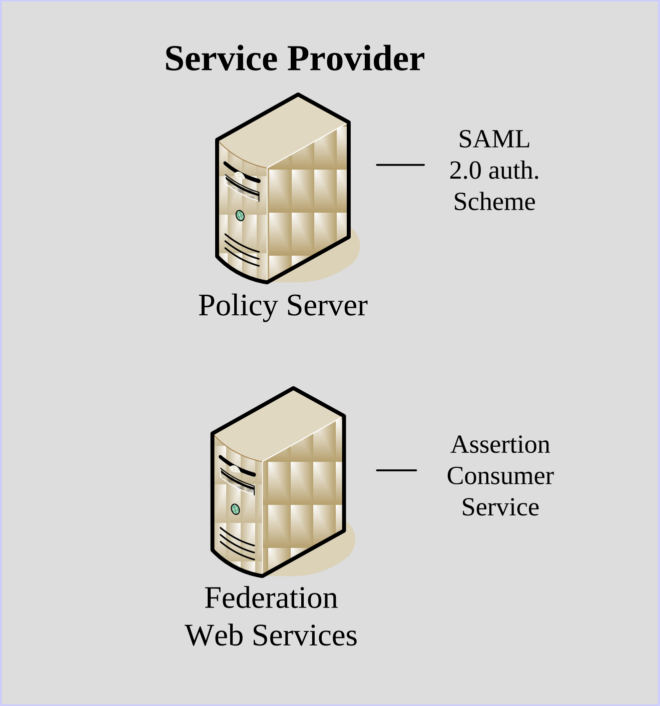

Figure 9-4: SAML 2.0 authentication components

The SAML 2.0 authentication scheme is configured at the FSS Applet UI, and is invoked by the Assertion Consumer Service. This service is a component of the Federation Web Services application and is installed on the Service Provider's Web Agent or SPS Federation gateway. The Assertion Consumer Service obtains information from the SAML authentication scheme and then uses that information to extract the necessary information from a SAML assertion. An authentication attempt is made on the user's behalf using the SAML assertion. If the authentication is successful, a session is established for the user. All authorization decisions are then responsibility of the SiteMinder Policy Server and Web Agent.

Note: The Assertion Consumer Service accepts an AuthnRequest that includes an AssertionConsumerServiceIndex value of 0. All other values for this setting are denied.

The following illustration shows how the SAML 2.0 authentication scheme processes requests.

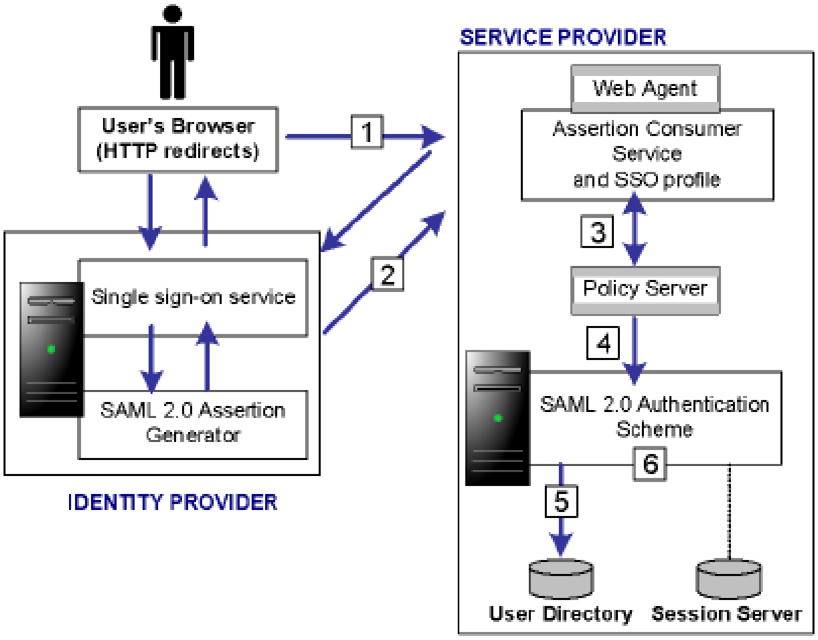
SERVICE PROVIDER
Web Agent
Assertion Consumer Service and SSO profile
User's Browser (HTTP redirects)
1
2
3
4
5
6
Single sign-on service
SAML 2.0 Assertion Generator
IDENTITY PROVIDER
Policy Server
SAML 2.0 Authentication Scheme
User Directory
Session Server

Figure 9-5: SAML 2.0 authentication flow

The functional flow for authentication is as follows:

- A user's browser makes a request for a Service Provider resource. This request goes to the AuthnRequest service at the Service Provider. The request is then redirected to the Identity Provider to obtain a SAML assertion.
- The Identity Provider returns a response to the Service Provider. In the case of the HTTP-POST binding, the response contains the assertion. For the HTTP-Artifact binding, the response contains a SAML artifact.
- The Assertion Consumer Service at the Service Provider receives the response message and determines whether the POST or Artifact binding is being used. If the artifact binding is being used, the Assertion Consumer Service sends the artifact to the Identity Provider to obtain a response that contains the actual assertion. The Assertion Consumer Service sends the response with the assertion as credentials to the Policy Server.
- The Policy Server invokes the SAML 2.0 authentication scheme by passing the response message with the user credentials to the scheme to be authenticated.
- The user disambiguation process begins.
- After the disambiguation phase is complete, the SAML 2.0 authentication scheme validates the credentials in the assertion, validates the assertion itself for time validity, and, if applicable, verifies if the assertion was signed by a trusted Identity Provider.

Note: For the POST binding, a signature is required and there must be certificate lookup information supplied. If a signature is not present, authentication fails. However, for the Artifact binding, a signed assertion is optional because the assertion response is obtained over a secure channel between the Service Provider and Identity Provider. If Single Logout is enabled, the user is redirected by the SLO servlet to a No Access URL.

When a user attempts to gain access to a protected (and federated) resource by authenticating, the SAML assertion becomes the user's credentials to log into the Policy Server at the consumer federated entity. By this process, the user is authenticated and is redirected to the targeted resource.

For the realm containing the protected target resource, a rule is created that is triggered during the authorization process to retrieve the SAML attributes from the session store. The rule is based on an authorization event (onAccessAccept) because the user has already been authenticated by the FWS application, therefore the Web Agent cannot re-authenticate the user and pass on the HTTP headers. The retrieval of the attributes must happen during the authorization stage.

9.1.2.9 SAML Attributes as HTTP Headers

An assertion response includes attributes in the assertion. These attributes are supplied as HTTP header variables and used by a client application; these headers are used for finer grained access control. The benefit of including attributes in HTTP headers is as follows:

- HTTP headers are not persistent. They are present only within the request or response that contains them.
- HTTP headers, as supplied by the SiteMinder Web Agent, are not seen by the user's browser, which reduces security concerns.

During authentication, a series of SAML attributes are extracted from an assertion and supplied as HTTP headers. During the authorization process, these headers are returned to the customer's application.

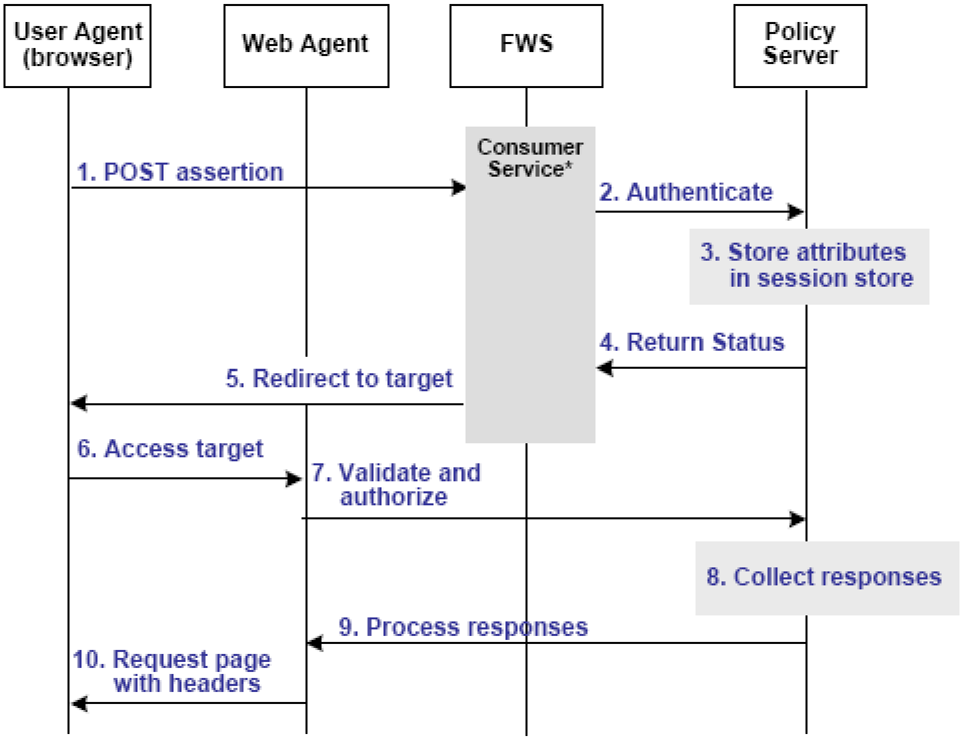

Figure 9-6: Consuming Side of Federated Network

A consumer service can be one of the following:

- SAML Credential Collector (SAML 1.1)
- Assertion Consumer Service (SAML 2.0)

After the consuming partner authenticates the user with the SAML assertion, the SAML attributes are written to the session store. SiteMinder then determines if authorization is allowed. If so, the browser is then redirected to the target resource.

For the realm containing the protected target resource, a rule needs to be created that is triggered during the authorization process to retrieve the SAML attributes from the session store. The rule is based on an authorization event (onAccessAccept) because the user has already been authenticated by the FWS application, therefore the Web Agent cannot re-authenticate the user and pass on the HTTP headers. So, the retrieval of the attributes must happen during the authorization stage.

A response must be configured that sends the SAML attributes as HTTP headers to the Web Agent. The Web Agent will process the response and make the header variables available to the client application.

To implement the use of SAML attributes as HTTP headers, an Administrator must group together the authorization event rule and active response in a policy.

9.1.3 TOE Access

The TOE enacts the process of single logout (SLO) (also known as cross-domain single sign-out) which results in the simultaneous end of all sessions for a particular user, thereby ensuring security. These sessions must be associated with the browser that initiated the logout. Single logout does not necessarily end all sessions for a user. For example, if the user has two browsers open, that user can establish two independent sessions. Only the session for the browser that initiates the single logout is terminated at all federated entities for that session. The session in the other browser will still be active. Single logout is triggered by a user-initiated logout at either the Asserting Party or the Relying Party.

With single logout enabled, information about the user's session is stored in the session store by the Assertion Generator. When a single logout request is completed, the user's session information is removed from the session store. This allows the Asserting Party to keep track of which sessions still need to be terminated as the process iterates.

By configuring the settings on the SLO tab an Administrator is informing the Asserting Party whether the Relying Party supports the single logout protocol, and if so, how single logout is handled. If single logout is enabled, an Administrator must also:

- Enable the session store at the Asserting Party.
- Configure persistent sessions for the realm containing the protected resources at the Relying Party.

Note: SiteMinder only supports the HTTP-Redirect binding for the single logout protocol.

Federation Web Services redirects the user to the logout confirm page after the user's session is completely removed at the Asserting Party and all Relying Party federated entities.

The illustration that follows shows the detailed flow for a single logout request between a user's browser and the Federation Security Service components deployed at an Asserting Party and Relying Party federated entities. This set-up enables single logout for all entities that have a session with a particular user. The following diagram, which uses SAML 2.0 naming conventions, assumes that the SP initiates the logout request.

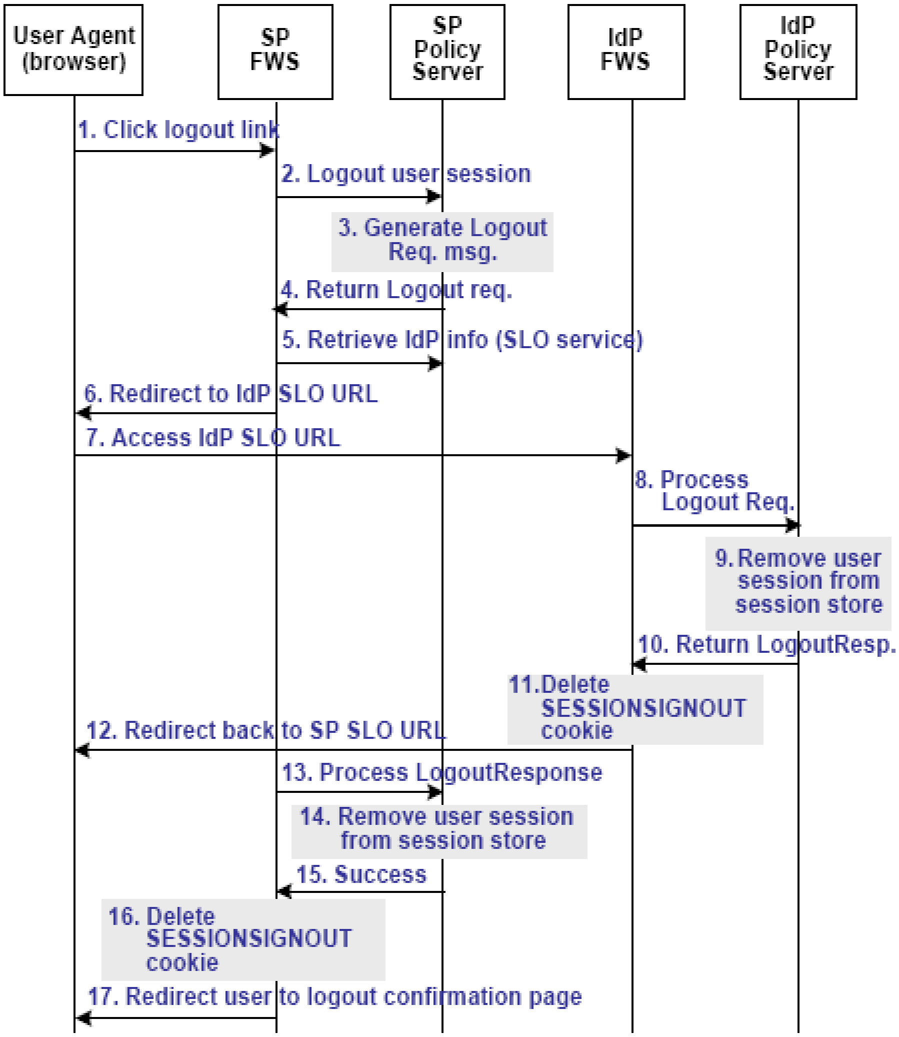
User Agent (browser)
SP FWS
SP Policy Server
IdP FWS
IdP Policy Server
1. Click logout link
2. Logout user session
3. Generate Logout Req. msg.
4. Return Logout req.
5. Retrieve IdP info (SLO service)
6. Redirect to IdP SLO URL
7. Access IdP SLO URL
8. Process Logout Req.
9. Remove user session from session store
10. Return LogoutResp.
11.Delete SESSIONSIGNOUT cookie
12. Redirect back to SP SLO URL
13. Process LogoutResponse
14. Remove user session from session store
15. Success
16. Delete SESSIONSIGNOUT cookie
17. Redirect user to logout confirmation page

Figure 9-7: SAML 2.0 Single Logout

The sequence of events is as follows:

- The user clicks a link at SP to end his global session. The user's browser accesses the Single Logout servlet at the SP. SP FWS renames the SMSESSION cookie to SESSIONSIGNOUT to invalidate the user's current session.

- FWS reads the SessionId value from the SESSIONSIGNOUT cookie and asks the Policy Server to terminate the user session.

- Based on the session store information, the user session status is changed to a 'LogoutInProgress' state in the session store. The Policy Server determines that the user session was created based on the SAML assertion received from an IdP. It generates a LogoutRequest request to invalidate the user's session at the IdP.

- The Policy Server returns a LogoutRequest request to SP FWS. It also returns the IdP's Provider ID and provider type.

- SP FWS retrieves the IdP's provider configuration data, which includes the SLO service URL, from the Policy Server.

- SP FWS redirects the user to the SLO service at the IdP with the SAML LogoutRequest message added as query parameter.

- User's browser accesses SLO service at the IdP. When the IdP FWS receives a LogoutRequest message, it renames the SMSESSION cookie to SESSIONSIGNOUT.

- The IdP processes the signed LogoutRequest message then tries to invalidate the user's session at all SPs specified in the session store for that user session, with the exception of the SP that sent the original LogoutRequest. Note: The process for logging the user out at each SP is similar to Step 2 through Step 7.

- After terminating the user's session from all relevant SPs, the IdP removes the user session from the session store.

- The IdP Policy Server returns a signed LogoutResponse message to the IdP FWS, containing the SP's provider ID and provider type. It also informs FWS that user session is removed from session store.

- After learning that the user session is removed from the session store, IdP FWS deletes the SESSIONSIGNOUT cookie.

- The IdP FWS redirects the user to the single logout service at the SP with the SAML LogoutResponse message added as query parameter. The single logout service is part of the SP FWS application.

- The user's browser accesses SP's SLO service, which processes the signed LogoutResponse message. If the LogoutResponse message contains non-SUCCESS return code, FWS issues a SIGNOUTFAILURE cookie, and a base 64-encoded Partner ID is appended to the cookie value. If there are multiple IDs in the cookie, they are separated by a space character. The SP Policy Server receives the LogoutResponse message from FWS and processes it.

- The SP Policy Server removes the user session from the session store.

- After the session is removed from the session store, the Policy Server sends a SUCCESS return code to FWS along with the SP provider ID in the final LogoutResponse message.

- If there are no more LogoutRequest or LogoutResponse messages to process, SP FWS deletes the SESSIONSIGNOUT cookie.

- FWS redirects the user to the Logout Confirmation page at the SP.

Session establishment can also be denied by the TOE based on a malformed XML or invalid SAML assertion. When an assertion (SAML 2.0) is successfully validated, the SAML 2.0 authentication scheme writes assertion data in the expiry data table with a key of the assertion ID and an expiration time. The Session Store Management thread in the Policy Server deletes expired data from the expiry data table. If single policy use is enforced, writing assertion data will fail if an entry already exists in the expiry data table with the primary key of the assertion ID because the assertion has already been used to establish a session. If the scheme cannot write to the table in the session store, the SAML 2.0 authentication scheme denies the authentication in the same manner as an invalid assertion.

Writing assertion data may fail for other reasons; however, if the single use of the assertion cannot be enforced because the database is unavailable for any reason, then the authentication scheme will deny the request to ensure that assertions cannot be re-used.

9.1.4 Security Management

The TOE provides for two distinct roles –Users and Administrators. Users are those who attempt to access protected and federated resources. Once successfully authenticated, they receive authorization from SiteMinder to view the federated resources via their web browser. Administrators are those who have full privileges to manage and maintain data as well as create, view, modify, and delete objects. For a list of capabilities that only Administrators can perform on the TOE, refer to Table 9-1. Only Administrators are allowed to perform view, create, enable/disable, modify, and delete operations for the following functions:

- SAML affiliations for SAML 2.0
- SAML authentication schemes
- Affiliate domains, which contain:
    - Affiliates (SAML 1.1)
    - Service Providers (SAML 2.0)
- SiteMinder objects and policies (except for application objects for EPM)

Beginning with SiteMinder r12 SP1 CR3, there are two graphical user interfaces (UIs) that configure SiteMinder policy objects: the WAM Administrative UI and the Federation Security Services (FSS) Applet UI. However, only the FSS Applet UI is used in the evaluated configuration. The WAM Administrative UI was previously validated during the certification of the base component. The FSS Applet UI is capable of performing the required base component management operations to set up the Composed TOE, so in the Composed TOE, the WAM Administrative UI does not need to be used.

The FSS Applet UI is an applet-based application that is installed with the Policy Server. The Federation-specific UI objects consist of affiliates (consumers, service providers, resource partners) and SAML authentication schemes that are configured to support federated communication between two partners. The intent of the FSS Applet UI is to let Administrators manage SiteMinder Federation Security Services. Many of the FSS operations overlap with the operations required to manage SiteMinder. As a result, these operations will be identical to how they are performed using the SiteMinder WAM UI. In order to operate the TOE, a certain minimum amount of configuration of SiteMinder is required. All of these operations are performed either using the SiteMinder WAM UI or the FSS Applet UI.

Administrators use the FSS Applet UI to manage all of the Policy Server objects listed in Table 9-1. Note that the highlighted items in grey are objects which can be managed using the WAM Administrative UI as well if desired.

Operations	**Policy Server Objects**	**Interface**
Create/view/modify/delete	Agents	**FSS Applet Admin UI**
Create/view/modify/delete	**Agent Configuration Objects**	**FSS Applet Admin UI**
Create/view/modify/delete	**Host Configuration Objects**	**FSS Applet Admin UI**
Create/view/modify/delete	Policy domains	**FSS Applet Admin UI**
Create/view/modify/delete	Affiliate domains	**FSS Applet Admin UI**
Create/view/modify/delete	Authentication Schemes	**FSS Applet Admin UI**
Create/view/modify/delete	SAML Affiliations	**FSS Applet Admin UI**
Create/view/modify/delete	rules (in managed domains)	**FSS Applet Admin UI**
Create/view/modify/delete	**policies (in managed domains)**	**FSS Applet Admin UI**
Create/view/modify/delete	Affiliates	FSS Applet Admin UI
Create/view/modify/delete	SAML Service providers	FSS Applet Admin UI

Table 9-1: Management of TSF Data

In order to be granted access and have a session established, administrators must provide the following credentials: username, password, host name, and pass phrase. Once granted access, administrators have the ability to manage Agents, specifically create new Agents, modify existing Agents, or delete existing Agents. These management operations are performed by selecting the "Agents" tab from within the FSS Applet UI. Administrators also have the ability to manage Agent Configurations.

In order to manage Policy Domains, administrators have the ability to create, modify existing, or delete existing Policy Domains, which serve to challenge users when attempting to access a Relying Party resource. Administrators that grant control over a policy domain to other administrators have the privilege to manage system and domain objects. Within the scope of managed domains, administrators also manage policies and rules, and create top-level realms.

Exclusive to the FSS Applet UI are the following administrative abilities: create, edit, and delete Affiliate Domains; create, edit, and delete SAML affiliations, create, edit, and delete Affiliates; and, create, edit, and delete SAML Service Providers.

Affiliate domains require one or more administrator accounts that can modify the objects in the domain. Administrators manage all objects in any domain; they have the privilege Manage Affiliates. Additionally, through the FSS Applet UI, entities are added to the affiliate domain. The following consuming authorities can be added to the affiliate domain:

- SAML 1.1 Affiliates
- SAML 2.0 Service Providers

These entities must be given permission to access Federation Web Services at the producing authority when protecting the Federation Web Services application. When the Policy Server is installed, policies for each service that comprises the Federation Web Services application are automatically created. The protection of the Federation Web Services application using SiteMinder policies must be enforced.

9.1.5 Encrypted Communications

The TOE uses encryption keys and digital signatures to encrypt, decrypt, and validate sensitive data passed to and from the TOE. Communications between the Policy Server and the Federation Web Services (FWS) use symmetric keys generated by SiteMinder (AES with key sizes of 128 bits, in CBC or OFB mode depending on the use of the key) in order to provide for a trusted channel for all communications and interactions between TOE components. This same channel is used for communications to the FSS Applet UI as well. SAML assertions are signed and optionally encrypted in order to protect user data from disclosure.

All communication between users/Administrators and the TOE are secured via a trusted path using SSL v3.0. Similarly, all communication between TOE components via the frontchannel and backchannel are secured via a trusted channel using SSL v3.0. The trusted path and trusted channel are established by utilizing keys which are generated by the base component (SiteMinder).

Communications between logically separate components of the TOE such as the Policy Server and Federation Web Services are secured using 128-bit AES keys in OFB mode. These keys are generated by SiteMinder for use by the TOE in accordance with FIPS Pub 197. Administrator data when communicating to the TOE such as their authentication credentials and management commands are secured using 128-bit AES keys in CBC mode. These keys are generated by SiteMinder for use by the TOE in accordance with RFC 3602.

Encrypting the Name ID in an assertion and/or the assertion itself is done through the FSS Applet UI. Encryption adds another level of protection when transmitting the assertion. When configuring encryption, the partner certificate must be specified, which is included in the assertion. When the assertion arrives at the Relying Party, the Relying Party decrypts the encrypted data using the associated private key.

SAML assertions are digitally signed when distributed to Relying Parties so that their integrity can be verified. In addition, SAML 2.0 assertions utilize 1024-bit RSA encryption to encrypt their contents. Table 9-2 lists all mechanisms which can be used by the TOE to protect assertions:

Private Keys	**Encodings Supported**	**Public Certificates**
Formats Supported		X.509 Certificate Formats Supported
PKCS#1 PKCS#5 PKCS#8 PKCS#12	Base64 DER PEM	V1 V2 V3

Table 9-2: SAML Formats Supported by the TOE

PKCS #1 defines the RSA standard used to encrypt assertions. PKCS #5 defines the password-based key derivation function to strengthen the RSA encryption. PKCS #8 and PKCS #12 apply to the transfer and storage of key pairs for public certificates.

In addition, SiteMinder encryption is used for the following:

- All sensitive data stored in the Policy Store and in the Key Store are encrypted using the AES Key Wrap algorithm.
- Encrypted keys (and Shared Secrets) that are stored elsewhere (Registry, files) are encrypted with the AES Key Wrap Algorithm.
- Sensitive data exported from the Policy and/or Key stores are re-encrypted using the AES Key Wrap algorithm with a key derived from the pass phrase given in the export utility's command line.
- In the User Store, each user's Password Blob attribute is a collection of password state data encrypted using AES-128 in CBC mode ("cookie encryption").
- The Session Spec, Identity Spec, Session Cookie, Identity Cookie, Data Cookie (and any other cookies) are encrypted using AES-128 in CBC mode ("cookie encryption").

Note that all symmetric keys used by the TOE are generated by SiteMinder and stored in the Key Store. All public keys and certificates are imported into the smkeydatabase using the smkeytool application. This allows a deployment to use certificates issued by a CA.

Once a single sign-on network has been configured, encryption and decryption can take place. The IdP encrypts the assertion with the public key, which corresponds to the private key and certificate that the SP uses to decrypt the assertion. Unlike SiteMinder, which only uses symmetric keys, the TOE also uses public key encryption.

9.1.5.1 Smkeydatabase

The smkeydatabase is a key and certificate database used for signing, verification, encryption, and decryption of SAML assertions between a Relying Party and an Asserting Party. The database is made up of multiple files. Keys and certificates in this database are managed and retrieved using the SiteMinder tool called smkeytool. Once the keys are used by the TOE, they are destroyed by SiteMinder’s zero overwrite process. The smkeydatabase holds the certificate authority certificate that establishes an SSL connection between the Relying Party and the Asserting Party. The certificate secures the backchannel that the assertion is sent across. The Artifact Resolution Service needs to be protected and the backchannel need to be secure so the Relying Party knows the SSL connection is secured by a trusted authority. RSA keys used for encrypting and decrypting SAML 2.0 assertions are imported using smkeytool as well.

The sensitive data stored in smkeydatabase is encrypted by SiteMinder using the AES Key Wrap algorithm (“AESKW”) with a 128-bit key generated by passing the user-provided database passphrase and some fixed constants through the FIPS-140 Key Expansion algorithm based on SHA-256.

For POST single sign-on, the Identity Provider digitally signs the SAML assertion. Consequently, the Service Provider must validate the signature. To validate a digital signature, a public key needs to be added to the Relying Party's smkeydatabase file. When the SAML authentication scheme is configured, the issuer's DN and serial number of the corresponding partner certificate must be specified.

Private keys and digital certificates are used by the TOE to perform cryptography and digital signing of SAML assertions in order to protect user attribute data from disclosure and so that this data can be validated. Multiple private keys are stored in the smkeydatabase. If multiple federated partners exist, a different private key is used for each partner. The smkeydatabase is installed as part of the TOE and is considered an internal component. The Policy Server uses certified Federal Information Processing Standard (FIPS) 140-2 compliant cryptographic libraries, which enables a SiteMinder environment to use FIPS-compliant algorithms to encrypt sensitive data. As a result, all data in the smkeydatabase is encrypted using these FIPS-compliant algorithms. The smkeydatabase supports the same keys listed above in Table 9-2. The smkeydatabase stores the following keys and certificates:

- Signer's private key and the corresponding certificate, which is the public key that is signed by a certificate authority
- Public-key certificates that correspond to the private keys

When SAML 2.0 is used, the Identity Provider requires that the Service Provider sign AuthnRequest messages. The Identity Provider must provide the public key certificate of the Service Provider for encrypting the data, while the Service Provider uses a private key to decrypt the data.

Table 9-3 lists the certificates stored in smkeydatabase and why they are used.

Type of Certificate	Reason Being Used
Certificate Authority (CA)	Used for establishing an SSL connection from a consuming authority to the web server at a producing authority. A set of common root CA certificates are shipped with the default smkeydatabase. To use a certificate for a CA that are not already in the key store, the certificate must be imported into the database.
Client Certificate	Used for sending a certificate from a consuming authority to a producing authority. The certificate serves as credentials when the consumer must authenticate using a client certificate authentication scheme to access the Assertion Retrieval or Artifact Resolution Service.
Partner Certificate	Used for performing digital signature verification at the consuming authority federated entity to ensure the authority issuing the assertion is a trusted site. At a SAML 2.0 Identity

	Provider, the partner certificate is used to verify the signed messages from the Service Provider during single logout. The Service Provider's certificate must exist at Identity Provider's machine.

Table 9-3: Certificates used by smkeydatabase

When the Web Agent initializes, it gets the entire client and server certificates, but the keys remain at the Policy Server.

9.1.5.2 Agent Keys and Session Ticket Keys

The TOE relies upon SiteMinder in order to generate Agent Keys which assist in maintaining user sessions. When the Web Agent initializes, it gets the entire client and server certificates, but the keys remain at the Policy Server.

After initial configuration of SiteMinder, the Key Store will already have a set of four Agent Keys, one static and three dynamic. Either the Static Agent Key or the set of three dynamic keys (Current, Old, and Future Agent Keys) are used to encrypt the SMSESSION cookie. In the evaluated configuration the set of three dynamic keys will be used. During the operation of the TOE, SiteMinder will perform key rollover and create new 192-bit Agent Keys using the AES Key Wrap algorithm. Once this occurs, SiteMinder will query the Key Store for the Agent Keys which are stored as the Current Key, and the Future Key. Once the keys are used by the TOE, they are destroyed by key zeroization.

When a user establishes a session with the TOE, they are given an encrypted SiteMinder SMSESSION cookie. Agent Keys are used by the SiteMinder Web Agents which run underneath Federation Web Services to encrypt/decrypt the SMSESSION cookie which is provided to a user when a session is established with the TOE. Once a new Agent Key is created by SiteMinder, the set of Agent Keys is distributed to all connected Web Agents when each Web Agent polls the SiteMinder Policy Server to determine if there is an updated set of Agent Keys. The set of Agent Keys are also distributed once a secure connection has been established between a Policy Server and Web Agent.

SiteMinder manages Session Ticket Keys on behalf of the TOE in a similar manner to the Agent Keys. New Session Ticket Keys are created using the AES Key Wrap algorithm with 128-bit keys. Once this occurs, the newly generated Session Ticket Key is stored in the Key Store. Since all previously encrypted Session Tickets and tokens cannot be unencrypted after the rollover, the Policy Server cannot verify the user’s session and determine the requester’s distinguished name. This requires the user or administrator to re-authenticate to the TOE and receive a Session Ticket or token which is encrypted with the new Session Ticket Key.

When a user has successfully authenticated to the TOE, the SAML Auth Scheme process instructs the Policy Server to create a Session Ticket to be sent to FWS for inclusion in the SMSESSION cookie. Before a Session Ticket or token leaves the Policy Server the

Cryptography Processes will use the 192-bit Session Ticket Key to encrypt it with AES-128 in CBC mode. The Session Ticket Key is passed through the FIPS-140 Key Expansion Algorithm to provide a 128-bit AES Key, a 128-bit AES Initialization Vector, and a 128-bit HMAC-SHA256 key. The 128-bit AES key with AES in CBC mode is then used to encrypt the Session Tickets or tokens before they are sent to the Web Agent. Session Tickets and tokens are decrypted when the TOE receives a request from an authenticated user on a protected resource or an authenticated administrator on a FSS Applet UI webpage. Encryption/decryption of the Session Ticket and the token is performed by retrieving the Session Ticket Key from the Key Store and repeating the same process with the Key Expansion Algorithm.

In the evaluated configuration the TOE will encrypt/decrypt all information stored in the Policy Store and the Key Store, which includes FWS Shared Secrets, the Session Ticket Key, and the four types of Agent Keys. This encryption/decryption is performed by the TOE with the 128-bit Policy Store Key to encrypt the information in the Stores with AES Key Wrap Algorithm. This requires the Policy Server to retrieve the Policy Store Key from local memory.

For more information on Agent Keys and Session Ticket Keys, see the CA SiteMinder Web Access Manager r12 SP1 CR3 Security Target v1.0.

9.1.6 Protection of the TSF

Protecting the Federation Web Services application at the Asserting Party ensures that the services that make up the application are secure. For protection of data transmitted between separate parts of the TOE and between the TOE and SiteMinder components, the proprietary TLI handshake is used to establish a trusted channel between them. The policies for protecting the Federation Web Services application are created automatically by the installation of the PS Option Pack. These policies can be modified by an authorized administrator by using the FSS Applet UI to manage the TOE.

There is a pre-configured policy that uses the Basic over SSL authentication scheme to protect the Assertion Retrieval Service. When configuring the policy for the client certificate authentication scheme, this policy is created for a different realm than the realm that uses the Basic over SSL scheme. To protect the Assertion Retrieval Service using a client certificate authentication scheme, the Administrator must do the following:

- Create a policy at the Asserting Party that uses an X.509 client certificate authentication scheme.
- Enable client certificate authentication at the Relying Party.

Further protection is done by enabling artifact binding for artifact single sign-on or POST binding for POST single sign-on. Similar to the process of creating web agents, artifact binding and POST binding is enabled through the FSS Applet UI.

For artifact single sign-on, if Basic over SSL is the authentication scheme protecting the Artifact Resolution Service, a certificate must be added to the Relying Party's smkeydatabase. The smkeydatabase holds the certificate authority certificate that establishes an SSL connection between the Relying Party and the Asserting Party. The certificate secures the backchannel that the assertion is sent across. The Artifact Resolution Service needs to be protected and the backchannel need to be secure so the Relying Party knows the SSL connection is secured by a trusted authority.

For POST single sign-on, the SAML response must be signed. There are configuration tasks at the Identity Provider and Service Provider to enable digital signing. The Identity Provider digitally signs the SAML assertion, and the Service Provider must validate the signature. To validate a digital signature, an Administrator needs to add a public key to Service Provider's smkeydatabase file. When the SAML authentication scheme is configured, the issuer's relevant DN attributes and serial number of the corresponding partner certificate are specified.

In the FSS Applet UI, the SSO tab allows single sign-on to be configured using the artifact or POST binding. This enforces the single use assertion policy for POST binding to prevent the replaying of a valid assertion. When replay is detected, the TOE denies the request and returns an HTTP 500 error to the user.

9.1.7 Self-Protection (ADV_ARC.1)

The TOE and the Operational Environment mutually protect the TOE and its security mechanisms from being circumvented. This is accomplished through the separation of roles, the tracking and auditing of user sessions, no remote direct connection to the Federation Web Services, interactions with a trusted product, the TOE's processes running at root privileges, and the protection of the paths/channels.

9.1.7.1 Separation of Roles

The TOE separates the roles of users on the TOE by having two separate interfaces for interaction with the TOE. In the evaluated configuration, only two remote interfaces will be used. One is the user connecting to the TOE via SiteMinder's Web Agent, and the other is the interface that connects the remote administrator to the TOE via the FSS Applet UI.

The TOE is able to determine if it is an Asserting Party or a Relying Party, which determines the code paths available to it. By designating different behaviors for the different providers, assurance is given that each will only perform its appropriate duties (for example, a Relying Party will not be able to erroneously determine a user's identity). In addition, since the FSS Applet UI only applies individually to single instances of the TOE, an administrator cannot make changes to multiple servers using the same UI.

9.1.7.2 User Sessions

Once a user authenticates from their respective external interface, the TOE and its Operational Environment implement a method of tracking their sessions and actions on the TOE. When a user attempts to authenticate to the Asserting Party, they provide their credentials, and if valid, a SAML assertion is generated and an SMESSION cookie is issued to the user's browser for the session. This assertion is then transmitted to any federated Relying Parties, which then use this assertion as a user authentication request. If successful, each Relying Party will issue SMSESSION cookies to the user's browser, so that the user has a session for all federated entities. The user credentials required by the Relying Parties as defined in the affiliation are embedded inside the SAML assertion and encrypted using the appropriate methods. When the assertion is presented to a Relying Party, it is decrypted and disambiguated, and the user credentials are verified by SiteMinder so that the user is appropriately identified.

All user and administrator requests are also audited by the PS Option Pack. The user's and administrator's timestamps, distinguished names, requests, and responses are recorded by different processes within the Policy Server Option Pack. These processes communicate with SiteMinder to generate audit data. In order to view the audit data, there must be a local Administrator on the system where Federation is installed.

When an administrator authenticates to the TOE, the access request is forwarded to SiteMinder, which then enforces authorization to use the management functions of the

FSS Applet UI. Administrators must take care to ensure that the only trusted individuals are given access to this interface, as there is no granularity of privileges in the FSS Applet UI. All administrator operations using this interface are audited.

If the TOE does not receive a well-formed and valid SAML assertion for a user, Federation will not be able to authenticate them, which is a prerequisite for SiteMinder to authorize actions against protected resources. As a result, the TOE cannot enable any TSF-mediated actions unless a user session has been created.

9.1.7.3 Connection to the PS Option Pack

The TOE protects the PS Option Pack by having another TOE component interpret a user's request before forwarding that request to the PS Option Pack. All requests by remote administrators must be parsed by the launched FSS Applet UI before being sent to the PS Option Pack. Once sent, they must be associated with a validated administrator session .All requests by remote users must first be parsed by the SiteMinder Web Agent and Federation Web Services. If these components are unable to parse the request, then an error will occur at the externally viewable subsystem and the request will never reach the PS Option Pack. The only information passed from the user to the TOE is authentication information, which is encoded as a SAML assertion by Federation Web Services, or a logout request, which is interpreted as a call to SiteMinder's Session Store. Assertions are cryptographically signed, and SAML 2.0 assertions can be encrypted as well. The user cannot perform any management functions without administrative privilege , which greatly restricts the potential data they can pass to the TOE.

9.1.7.4 Interactions with a Trusted Product

The TOE relies on communication with SiteMinder via internal interfaces in order to function properly. A SiteMinder deployment is needed to determine if resources are protected and to forward authentication requests to FWS when necessary. The TOE also relies on SiteMinder to retrieve information from the environmental data stores in order to perform its own processes. The mechanisms SiteMinder uses to do this are the same ones which were validated as part of SiteMinder's EAL3 evaluation. Administrator management of TOE data uses the SiteMinder object layer, which is the same backend used by the SiteMinder WAM UI to manage SiteMinder data. As a result, the TOE has assurance that calls made to SiteMinder components for data transmission, storage, retrieval, and management will be executed in a functionally appropriate and secure manner.

9.1.7.5 TOE Processes Perform with Root Privileges

Any actions on TOE files or processes are determined by the Operating System (OS) which the TOE has been installed on. This is because the TOE's processes run with root rwx permissions and the files are protected by the OS authorization mechanisms. In the evaluated configuration, the TOE will be installed by an administrator utilizing the Unix "root" or Windows Administrator user provided by the respective OS. A user that gains local access to a machine with the TOE components installed must first authenticate to the OS with the same permissions provided by these accounts to affect the TOE's processes or files. This is because any user with lesser privileges which tries to perform an action on the TOE's processes or files on the OS will be sent to the OS kernel for authorization. Once the kernel's authorization mechanisms recognize that the local user does not have root permission, the kernel will reject the request.

When the TOE is initialized its processes also run with root permissions, and therefore the TOE relies on the OS to determine what users have access locally to initialize these processes. The external interfaces visible to the users and administrators of the TOE are not accessible until the TOE is fully initialized and makes authentication and authorization decisions. The TOE is designed as an augmentation to SiteMinder and so single sign-on to protected web resources across multiple domains cannot be established until SiteMinder has been fully initialized in addition to the TOE.

9.1.7.6 Protection of Paths/Channels

The TOE provides protection of its internal channels between the TOE components and between TOE and SiteMinder components. The paths between these components are secured using the proprietary TLI handshake process. This process relies on keys generated by SiteMinder, but both SiteMinder and the TOE can perform operations using these keys. The protected channel allows single sign-on data to be passed between federated entities without risk of disclosure.

The TOE relies on the Operational Environment to protect the path from the remote administrator's FSS Applet UI instance to the Policy Server Option Pack's web server. The TOE's administrative guidance includes information on the necessity of having this path protected by SSL v3.0 encryption. This will ensure that administrator authentication information and management operations cannot be read by another user which is sniffing the packets which are being sent over that path.

The TOE also relies on the Operational Environment to protect data that is written to the environmental data stores. The security features offered by the underlying OS and data stores protect the files used by the TOE.

9.1.8 Base Component Dependencies

As previously stated, the TOE is not capable of running as a standalone product; instead, it functions as a composed TOE with the previously validated SiteMinder Web Access Manager R12 SP1-CR3 product operating as the base component. The base component provides the following behavior that is required for the proper function of the composed TOE:

- Cryptographic Key Generation – SiteMinder generates symmetric keys that are used by Federation to establish trusted communication between separate components of the composed TOE. In addition, the smkeytool application is used to generate public/private key pairs for cryptography and signing related to SAML assertions.

- User Data Protection – SiteMinder data stores are used to store data about users, policies, and sessions internal to the TOE. Once authenticated sessions have been established, SiteMinder is responsible for determining if the authenticated user is authorized to access protected resources based on a set of configured rules.

- Identification and Authentication – SiteMinder authentication schemes are used to facilitate initial authentication to the Asserting Party. Once accepted, Federation is responsible for propagating the authentication requests to each Relying Party.

- Strength of function – SiteMinder provides the composed TOE the ability to mandate password complexity requirements for users.

10 TOE Summary Specification Rationale

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

Security Function	Security Functional Components
Security Audit (FAU)	FAU_GEN.1 Audit data generation
	FAU_GEN.2 User identity association
Cryptographic Support (FCS)	FCS_COP.1(1) Cryptographic operation
	FCS_COP.1(2) Cryptographic operation
	FCS_COP.1(3) Cryptographic operation
	FCS_COP.1(4) Cryptographic operation
Identification and Authentication (FIA)	FIA_ATD.1 (1) User attribute definition
	FIA_ATD.1 (2) User attribute definition
	FIA_UAU.2 (1) User authentication before any action
	FIA_UAU.2 (2) User authentication before any action
	FIA_UAU.5 Multiple authentication methods
	FIA_UAU.6 Re-authenticating
	FIA_USB.1 User-subject binding
	FIA_UID.2 (1) User identification before any action
	FIA_UID.2 (2) User identification before any action
Security Management (FMT)	FMT_MOF.1 Management of security functions behavior
	FMT_MTD.1 Management of TSF data
	FMT_SMF.1 Specification of Management Functions
Protection of the TSF (FPT)	FPT_ITC.1 Inter-TSF trusted channel
	FPT_ITT.1 Basic internal TSF data transfer protection
	FPT_RPL.1 Replay detection
TOE Access (FTA)	FTA_SSL.4 User-initiated termination
	FTA_TSE.1 TOE session establishment

Table 10-1: Security Functional Components for Federation

10.1.1 Security Audit

The TOE relies on SiteMinder to generate log files that contain auditing information about the events that occur within the system, including the startup and shutdown of audit functions and all user/Admin actions on the TOE. For each event caused by a user/Admin, the TOE is able to associate each event with the user or administrator that caused the event. Each record captures the date, time, and type of event, subject identity, success or failure of the event, remote server host name, and remote server host ID. Additionally, Federation Web Services and the Policy Server create logs for error messages, trace messages and results of the SAML assertion generator and SAML authentication scheme activities.

Based on the above information, the TOE enforces the FAU_GEN.1 and FAU_GEN.2 requirements as discussed in Section 9.1.1.

10.1.2 Encrypted Communications

In the evaluated configuration, the TOE encrypts communications from itself to SiteMinder and between distributed instances of itself. It also performs encryption and signing of SAML assertions using public key information generated by a third party and imported into the smkeydatabase using smkeytool.

The TOE uses the following cryptographic algorithms:

- 128-bit AES in OFB mode to encrypt communications between Federation Web Services and the Policy Server and between the FSS Applet UI and Policy Server, conformant with FIPS Pub 197
- 128-bit AES in CBC mode to encrypt administrator session data, conformant with RFC 3602
- 1024-bit RSA to encrypt SAML assertions, conformant with PKCS #1 and PKCS #5
- X.509 V1, V2, or V3 signing using Base64, DER, or PEM encoding to sign and verify SAML assertions, conformant with PKCS #8 and PKCS #12.

While the smkeydatabase is used for storing public key and certificate data, the environmental Key Store is used for agent and session ticket keys from SiteMinder for encrypting communications.

The signer's private key and the corresponding certificate, which is the public key that is signed by a certificate authority, is stored in the smkeydatabase. The key and certificate are used to do the following:

1. Sign SAML responses for single sign-on requests
2. Sign SAML responses for AuthnRequests

3. Sign SAML responses for single logout requests
4. Decrypt assertions, Name IDs, and attributes

Based on the above information, the TOE enforces the FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), and FPT_ITC.1 requirements as discussed in Section 9.1.5.

10.1.3 TOE Access

The TOE enacts the process of single logout (SLO) which results in the simultaneous termination of all sessions for a particular browser by a single user. Only the session for the browser that initiates the single logout is terminated at all federated entities for that session. Single logout is triggered by a user-initiated logout at either the Asserting Party or the Relying Party. Figure 9-7 illustrates the flow for a Single Logout (SLO) for SAML 2.0.

Session establishment can also be denied by the TOE based on a malformed XML or invalid SAML assertion. When an assertion (SAML 2.0) is successfully validated, the SAML 2.0 authentication scheme writes assertion data in the expiry data table with a primary key of the assertion ID and an expiration time. If the scheme cannot write to the table in the session store, the SAML 2.0 authentication scheme denies the authentication in the same manner as an invalid assertion.

Based on the above information, the TOE enforces the FTA_SSL.4 and FTA_TSE.1 requirements as discussed in Section 9.1.3.

10.1.4 Identification and Authentication

Users and Administrators must be identified and authenticated to the TOE through the process of disambiguation prior to SiteMinder being able to determine access to the resources protected by the TOE. By authenticating through the Asserting Party via username, password, and certificate, Federation establishes sessions for users with all Relying Parties defined by the federation. As a result, the user always authenticates through the Asserting Party. The user either authenticates directly through the Asserting Party, or the Relying Party provides a redirect to the Asserting Party in order to provide initial authentication. The TOE facilitates the authentication process by allowing the Policy Server to determine the authentication scheme being used to identify the user. The authentication schemes supported by the TOE include Basic Over SSL Template or X509 Client Cert Template. The information is transmitted over SSL using either HTTP-artifact or HTTP-POST over SAML 1.1, SAML 2.0.

Based on the above information, the TOE enforces the FIA_ATD.1, FIA_UAU.2, FIA_UAU.5, FIA_UAU.6, and FIA_USB.1 requirements as discussed in Section 9.1.2.

10.1.5 Protection of the TSF

Protecting the Federation Web Services application at the Asserting Party ensures that the services that make up the application are secure. For protection of data transmitted between separate parts of the TOE, and to prevent the data from disclosure and modification, a proprietary algorithm is used. Policies to protect Federation Web Services are configured and applied by default, but they can be modified by an authorized administrator using the FSS Applet UI.

In order to establish single sign-on between the Asserting Party and Relying Party, the SSO bindings supported by the SP need to be specified. In the FSS Applet UI, the SSO tab allows single sign-on to be configured using the artifact or POST binding. This enforces the single use assertion policy for POST binding to prevent the replaying of a valid assertion. When replay is detected, the TOE denies the request and returns an HTTP 500 error to the user.

Based on the above information, the TOE enforces the FPT_ITT.1 and FPT_RPL.1 requirements as discussed in Section 9.1.6.

10.1.6 Security Management

The TOE provides for two distinct roles –Users and Administrators. Users are those who attempt to access federated resources. Once successfully authenticated, they receive authorization from SiteMinder to view the federated resources via their web browser. Administrators are those who have full privileges to manage and maintain data as well as create, edit, and delete objects.

There are two graphical user interfaces (UIs) that configure SiteMinder policy objects: the WAM Administrative UI and the Federation Security Services (FSS) Applet UI. Administrators use the FSS Applet UI to manage all of the Policy Server objects listed in Table 9-1. Note that the highlighted items in grey are objects which can be managed using the WAM Administrative UI as well if desired. However, only the FSS Applet UI is used in the evaluated configuration.

Based on the above information, the TOE enforces the FMT_MOF.1, FMT_MTD.1, and FMT_SMF.1 requirements as discussed in Section 9.1.4.

11 Rationale

11.1 Security Objectives Rationale

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

Assumption	Objective	Rationale
A.ADMIN 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 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 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 Administrators 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 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.
A.PASSWORD Users will select strong passwords to be enforced by SiteMinder and will protect their authentication data.	OE.PASSWORD Users shall ensure that they choose strong passwords to be enforced by SiteMinder and that they protect their authentication data.	OE.PASSWORD directly maps to A.PASSWORD to ensure that users will select strong passwords to be enforced by SiteMinder and will protect their authentication data.
A.FILESYS The underlying Operating System and data stores will protect the files used by the TOE.	OE.FILESYS The security features offered by the underlying Operating System and data stores protect the files used by the TOE.	OE.FILESYS maps to A.FILESYS to ensure the protection of files used by the TOE.

Table 11-1: Assumption to Objective Mapping

Threat	Objective	Rationale
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

Threat	Objective	Rationale
		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 administrators with the resources to manage and monitor user accounts, TOE resources and security information relative to the TOE.	O.MANAGE (FMT_MOF.1, FMT_MTD.1, 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.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 to TOE data.	O.EAVESDROPPING (FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FPT_ITT.1, FPT_ITC.1,) mitigates T.EAVESDROPPING by ensuring that all communications to and from the TOE are encrypted, ensuring that data gathered through eavesdropping cannot be used or interpreted.
T.MASK Users whether they be malicious or non-malicious, could gain unauthorized access to resources protected by the TOE by bypassing identification and authentication countermeasures.	O.AUDIT The TOE will provide measures for recording security relevant events that will assist local OS administrators 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) addresses T.MASK by providing local OS administrators with tools necessary to monitor user activity to ensure that misuse of the TOE does not occur.

Threat	Objective	Rationale
T.UNAUTH Users or administrators could gain unauthorized 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 their claimed identity prior to allowing SiteMinder the ability to enforce access resources protected by SiteMinder. 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(1), FIA_ATD.1(2), FIA_UID.2(1), FIA_UID.2(2), FIA_UAU.2(1), FIA_UAU.2(2), FIA_UAU.5, , FIA_UAU.6, FIA_USB.1, FTA_SSL.4, FTA_TSE.1, FPT_RPL.1) addresses T.UNAUTH by providing measures to uniquely identify and authenticate users through User Name/Password, host name, passphrase, Basic over SSL, or x.509 certificates and measures to uniquely identify and authenticate administrators through username, password, host name, and passphrase. The authorized users with the capability to specify access restrictions on the protected TOE resources to authenticated users.

Table 11-2: Threat to Objective Mapping

11.2 Security Functional Requirements Rationale

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

Objective	Security Functional Components	Rationale
O.AUDIT The TOE will provide measures for recording security relevant events that will assist local OS administrators 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.
O.AUTH The TOE will provide measures to uniquely identify all users and will authenticate their claimed identity prior to allowing SiteMinder the ability to enforce access resources protected by SiteMinder. 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.	FIA_ATD.1(1) User attribute definition	FIA_ATD.1(1) states that the TSF shall maintain the SAML attributes belonging to individual users.
	FIA_ATD.1(2) User attribute definition	FIA_ATD.1 (2) states that the TSF shall maintain the Username, Password, Host name, and pass phrase belonging to Administrators.
	FIA_UID.2 (1) User identification before any action	FIA_UID.2 (1) requires a user be identified before any access to the TOE and resources protected by the TOE is allowed.
	FIA_UID.2 (2) User identification before any action	FIA_UID.2 (2) requires an administrator be identified before any access to the TOE and resources protected by the TOE is allowed.
	FIA_UAU.2(1) User Authentication Before Any Action	FIA_UAU.2 (1) requires users to be successfully authenticated before any access to the TOE and its resources protected by the TOE is allowed.
	FIA_UAU.2(2) User Authentication Before Any Action	FIA_UAU.2 (2) requires administrators to be successfully authenticated before any access to the TOE and resources protected by the TOE.
	FIA_UAU.5 Multiple Authentication Schemes	FIA_UAU.5 states the TSF shall provide Basic Over SSL Template, X509 Client Cert Template, SAML Artifact Template, SAML 2.0 Template, and SAML POST Template authentication schemes to support user authentication.
	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.

	FIA_USB.1 User-Subject Binding	FIA_USB.1 states The TSF shall associate SAML attributes with subjects acting on behalf of that user. The TSF shall enforce the Asserting Party’s Relying Party object defines the attributes which are associated with SAML assertions and where they are derived (either from some component of the user DN or a static assignment) based on the initial association of user security attributes with subjects acting on behalf of The TSF shall enforce the no rules governing changes to the user security attributes associated with subjects acting on behalf of users.
	FTA_TSE.1 TOE Session Establishment	FTA_TSE.1 states that the TOE will deny session establishment based on a malformed XML or invalid SAML assertion.
	FTA_SSL.4 User-initiated Termination	FTA_SSL.4 states that the TOE allows a user to initiate termination of his own interactive session.

	FPT_RPL.1 Replay Detection	FPT_RPL.1 ensures that replaying will be detected for HTTP POST bindings for SAML. When replay is detected, the TSF will perform a denial of the request and return an HTTP 500 error to the user.
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_MOF.1 Management of Security Functions Behavior	FMT_MOF.1 states The TSF shall restrict the ability to determine the behaviour of, disable, enable, and modify the behaviour of the functions to Administrators. SAML affiliations for SAML 2.0, SAML authentication schemes, Affiliate domains, which can contain Relying Parties (SAML 2.0), and SiteMinder Objects and Policies (except for application objects for EPM).
	FMT_MTD.1	FMT_MTD.1 states The TSF shall restrict the ability to create, view, modify, and delete the Policy Server objects listed in Table 7-2 Management of TSF Data to Administrators.
	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 TSF data on Policy Server objects.
	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 associate the users with roles.
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 users from gaining unauthorized access to TOE data.	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 [RSA] and cryptographic key sizes [1024 bits]
	FCS_COP.1(4) Cryptographic operation	FCS_COP.1(4) states the TSF shall perform [digital signing] in accordance with a specified cryptographic algorithm [***X.509 V1, V2, and V3***] and cryptographic key sizes [***Base64, DER, PEM***]
	FPT_ITC.1 Inter-TSF Trusted Channel	FPT_ITC.1 states The TSF shall provide a communication channel between itself and another trusted IT product (SiteMinder) that is logically distinct from other communication channels and provides assured identification of its end points and protection of the communicated data from modification or disclosure. The TSF shall permit itself and another trusted IT product to initiate communication via the trusted channel. The TSF shall initiate communication via the trusted channel for single sign-on.
	FPT_ITT.1 Inter-TOE Transfer	FPT_ITT.1 states The TSF shall protect TSF data from disclosure and modification when it is transmitted between separate parts of the TOE.

Table 11-3: Security Functional Requirements Rationale

11.3 Requirement Dependency Rationale

All Security Functional Requirement component dependencies have been met by the TOE as defined by the CEM, with the exception of FPT_STM.1 and the FCS_COP.1 requirements.

FPT_STM.1 is a dependency of FAU_GEN.1. However, it is not included in this ST because the Operational Environment is responsible for providing accurate timestamps.

FCS_COP.1 (all iterations) is dependent on FCS_CKM.4 and FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2. None of these requirements are included in the ST because the encryption performed by the TOE is used for inter-TSF communication, inter-TOE transfer, and user authentication. In order to perform these operations, the TOE utilizes keys that are generated through other components, such as SiteMinder and either a Certificate Authority or self-signing process.

11.4 EAL Justification

The threats that were chosen are consistent with attacker of low attack potential; therefore, EAL3 was chosen for this ST. The base component of this Composed TOE was previously evaluated at EAL3. Refer to Section 3.7 for additional information on this Composed TOE.

11.5 PP Claims Rationale

This Security Target does not claim Protection Profile conformance.

12 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 ASE_TSS.2 and ALC_FLR.1. A description of each of the TOE assurance measures follows in Table 11-1.

Component	Document(s)	Rationale
ADV_ARC.1 Security Architecture Design	TOE Design Specification Document for CA Federation Security Services R12 SP1 CR3 V1.0	This document describes the security architecture of the TOE.
ADV_FSP.3 Functional Specification with complete summary	CA SiteMinder Federation Security Services r12 SP1 CR3 Functional Specification v1.0	This document describes the functional specification of the TOE with complete summary.
ADV_TDS.2 Architectural Design	[1] CA SiteMinder Federation Security Services r12 SP1 CR3 TOE Design Specification v1.0 [2] CA SiteMinder Federation Security Services r12 SP1 CR3 Functional Specification v1.0	This document describes the architectural design of the TOE.
AGD_OPE.1 Operational User Guidance	[1] CA SiteMinder® Federation Security Services - Federation Security Services Guide r12 SP1 [2] Evaluated Configuration for CA SiteMinder® Federation Security Services r12 SP1 CR3	This document describes the operational user guidance for CA SiteMinder Federation Security Services r12 SP1 CR3.
AGD_PRE.1 Preparative Procedures	Evaluated Configuration for CA SiteMinder® Federation Security Services r12 SP1 CR3	This document describes the preparative procedures that need to be done prior to installing CA SiteMinder Federation Security Services r12 SP1 CR3.

Component	Document(s)	Rationale
ALC_CMC.3 Authorizations Controls	[1] CA Clearcase Configuration Management Plan Version 1 [2] CA SiteMinder Federation Security Services - Configuration Management for Common Criteria r12sp1 [3] submission-approved_RE Mainline submission request to proj-hemlock-sp1 .txt [4] submission-request_Mainline submission request to proj-hemlock-sp1 for C65917 .txt [5] project-configuration-management.doc [6] FSS-12-SP1-Configuration-Item_List.zip [7] FSS-r12-SP1-CR3-Configuration-Item_List.zip	This document describes the authorization controls for the TOE.
ALC_CMS.3 CM Scope	[1] CA Clearcase Configuration Management Plan Version 1 [2] FSS-r12-SP1-CR3-Configuration-Item_List.zip	These documents describe the CM scope of the TOE.
ALC_DEL.1 Delivery Procedures	CA SiteMinder® Federation Security Services r12 SP1--- NIAP Download/Installation instruction	This document describes product delivery for CA SiteMinder Federation Security Services r12 SP1 CR3 and a description of all procedures used to ensure objectives are not compromised in the delivery process.
ALC_DVS.1 Identification of Security Measures	[1] 11-Backup_Procedure-GIS-2008Jun09.doc [2] 1619-GRC-Global_Security-Pre-employment_Screening-2008Apr05.doc [3] 1621 - GSAP.doc [4] 3649-Access_Procedure-2007Jun29.pdf [5] 5153-Project_360_Reference_Guide-2008Jul25.doc [6] 5725-GRC-BP-C-RIM-Records_Security_and_Confidentiality_Policy-2008May23.doc [7] 5727-GRC-BP-C-RIM-Records_Disposal_Procedure-2008May15.doc [8] 5804-Privileged_Access-2008Jun24.doc [9] 7417-Enterprise_Procedure-	This document provides an identification of security measures for the TOE.

Component	Document(s)	Rationale
	Privacy_and_Data_Protection-2007Mar06.doc [10] 7705-Inactive_User_Account_Procedure-2007Jun29.pdf [11] 7726-Server_Security_Procedure-2008Jun24.doc [12] 7978-US_Employee_Handbook-NorthAmerica-USA-2008Jul14.pdf	
ALC_LCD.1 Life-Cycle Definition	Project 360 Reference Guide revision 5.0	This document provides the life-cycle definition of the TOE.
ASE_CCL.1 Conformance Claims	CA SiteMinder Federation Security Services r12 SP1 CR3 Security Target v1.0	This document describes the CC conformance claims made by the TOE.
ASE_ECD.1 Extended Components Definition	CA SiteMinder Federation Security Services r12 SP1 CR3 Security Target v1.0	This document provides a definition for all extended components in the TOE.
ASE_INT.1 Security Target Introduction	CA SiteMinder Federation Security Services r12 SP1 CR3 Security Target v1.0	This document describes the Introduction of the Security Target.
ASE_OBJ.2 Security Objectives	CA SiteMinder Federation Security Services r12 SP1 CR3 Security Target v1.0	This document describes all of the security objectives for the TOE.
ASE_REQ.2 Security Requirements	CA SiteMinder Federation Security Services r12 SP1 CR3 Security Target v1.0	This document describes all of the security requirements for the TOE.
ASE_SPD.1 Security Problem Definition	CA SiteMinder Federation Security Services r12 SP1 CR3 Security Target v1.0	This document describes the security problem definition of the Security Target.
ASE_TSS.2 TOE Summary Specification	CA SiteMinder Federation Security Services r12 SP1 CR3 Security Target v1.0	This document describes the TSS section of the Security Target.
ATE_COV.2 Analysis of Coverage	Common-Criteria-Federation-MappingList_V5.0.xls	This document provides an analysis of coverage for the TOE.
ATE_DPT.1 Basic Design	Common-Criteria-Federation-MappingList_V5.0.xls	This document describes the basic design of the TOE.
ATE_FUN.1 Functional Tests	Common-Criteria-Federation-MappingList_V5.0.xls	This document describes the functional tests for the TOE.
ATE_IND.2 Independent Testing	Common-Criteria-Federation-MappingList_V5.0.xls	This document describes the independent testing for the TOE.
AVA_VAN.2 Vulnerability Analysis	CA SITEMINDER® FEDERATION SECURITY SERVICES R12 SP1 CR3 Version 0.5	This document describes the vulnerability analysis of the TOE.

Component	Document(s)	Rationale
ACO_COR.1 Composition Rationale	[1] CA SiteMinder Federation Security Services r12 SP1 CR3 TOE Design Specification v1.0 [2] CA SiteMinder Federation Security Services r12 SP1 CR3 Functional Specification v1.0 [3] TOE Design Specification Document for CA SiteMinder R12 SP1 v1.7 [4] Functional Specification Document for CA SiteMinder R12 SP1 v2.2	This document describes the composition rationale for the Composed TOE.
ACO_CTT.2 Rigorous Interface Testing	Common-Criteria-Federation-MappingList_V5.0.xls	This document describes the interface testing for the Composed TOE.
ACO_DEV.2 Basic Evidence of Design	[1] CA SiteMinder Federation Security Services r12 SP1 CR3 TOE Design Specification v1.0 [2] CA SiteMinder Federation Security Services r12 SP1 CR3 Functional Specification v1.0 [3] TOE Design Specification Document for CA SiteMinder R12 SP1 v1.7 [4] Functional Specification Document for CA SiteMinder R12 SP1 v2.2	This document describes the basic evidence of design for the Composed TOE.
ACO_REL.1 Basic Reliance Information	[1] CA SiteMinder Federation Security Services r12 SP1 CR3 TOE Design Specification v1.0 [2] CA SiteMinder Federation Security Services r12 SP1 CR3 Functional Specification v1.0 [3] TOE Design Specification Document for CA SiteMinder R12 SP1 v1.7 [4] Functional Specification Document for CA SiteMinder R12 SP1 v2.2	This document describes the basic reliance information for the Composed TOE.
ACO_VUL.2 Composition Vulnerability Analysis	CA SITEMINDER® FEDERATION SECURITY SERVICES R12 SP1 CR3 Version 0.5	This document describes the vulnerability analysis for the Composed TOE.

Table 12-1: Assurance Requirements Evidence