Tivoli Access Manager for e-business 6.1.1 FP4
with
Tivoli Federated Identity Manager 6.2.1 FP2
Security Target

Document Version Number 1.30

Document Update Date: 2012-05-16

Authors: Scott Chapman, David Ochel, Moritz von Schwedler, Auston Holt, Clemens Wittinger, King Ables

Owner: Frank Marullo

Status: Release

Table of Contents

Trademarks

AIX® is a registered trademark of International Business Machines Corporation.

atsec is a trademark of atsec GmbH.

IBM® is a registered trademark of International Business Machines Corporation.

Java is a trademark of Oracle Corporation in the United States and other countries.

Linux® is registered trademark of Linus Torvalds in the U.S. and other countries.

Red Hat® and Red Hat® Enterprise Linux® are registered trademarks of Red Hat, Inc. in the U.S. and other countries.

Solaris is a trademark of Oracle Corporation.

SuSE is a trademark of Novell, Inc. in the U.S. and other countries.

Tivoli® is a registered of International Business Machines Corporation.

Windows® is a registered trademark of Microsoft Corporation in the United States and other countries.

Legal Notice

Document Control Information

History

This track of document changes is a part of the standard document tracking process.

Version	Date	Summary of Changes
1.30	2012-05-16	ST for TAMeb 6.1.1 FP4 with TFIM 6.2.1 FP2.

1. Introduction

1.1. Security Target Identification

Title:	Tivoli Access Manager for e-business Version 6.1.1 FP4 with Tivoli Federated Identity Manager 6.2.1 FP2 Security Target
Version:	1.30
Status:	Release
Date:	2012-05-16
Sponsor:	IBM Corporation
Developer:	IBM Corporation
Certification ID:	BSI-DSZ-CC-0636
Keywords:	Access Control, ISO 10181-3, aznAPI, TFIM, TAMeb

1.2. TOE Identification

The TOE is Tivoli Access Manager for e-business Version 6.1.1 FixPack 4 with Tivoli Federated Identity Manager Version 6.2.1 FixPack 2.

1.3. TOE Overview

The TOE consists of Tivoli Access Manager for e-business (TAMeb) and Tivoli Federated Identity Manager (TFIM). Since not all TOE usage scenarios make use of TFIM, this document distinguishes between the security functionality provided by the two products so that the reader understands the security functionality differences when TAMeb uses TFIM and when TAMeb does not use TFIM.

TAMeb is a complete authorization solution for corporate web, client/server, Tivoli Access Manager applications, and legacy (pre-existing) applications. TAMeb allows an organization to securely control user access to protected information and resources located within the organizations infrastructure. By providing a centralized, flexible, and scalable access control solution, TAMeb allows users to build highly secure and well-managed network-based applications and e-business infrastructure.

Tivoli Federated Identity Manager (TFIM) aids in mapping identities between disparate organizations allowing organizations to maintain their current identification mechanisms and control which identities have cross organizational access. TFIM is used as a federated single sign-on (F-SSO) solution allowing users to enter their authentication data once and be granted access to several systems across organizations. In the evaluated configuration, TFIM supports the Security Assertion Markup Language (SAML) 1.1 Browser/POST Profile [SAML1.1] for exchanging user identities between other federated identity managers.

When TAMeb and TFIM are used in combination in the evaluated configuration, authorized users of one organization can be granted access to selected information in another cooperating organization that supports the SAML 1.1 Browser/POST Profile for identity exchange.

1.3.1 TOE Type

The TOE follows the standardized access control framework (TAMeb) defined by [ISO 10181-3] and [AZNAPI] with the use of federated single sign-on (TFIM). The combined products allow web clients to access web-objects on disparate organizational systems by enforcing access to objects through TAMeb and federated identity mapping through TFIM. The cryptographic functions used by both TAMeb and TFIM are part of the operational environment. TAMeb uses GSKit for support of HTTPS and certificate authentication. TFIM uses Java cryptographic functions for signing and validating SAML responses.

1.3.2 Required Non-TOE Hardware and Software

The operational environment for the evaluated configuration consists of the following hardware platforms and operating systems:

- AIX 6.1 (64-bit)
- Windows Server 2008 Enterprise (32-bit)
- SuSE Linux Enterprise Server 10 SP1 on IBM xSeries (32-bit)
- Red Hat Enterprise Linux Version 5 on IBM xSeries (32-bit)

In addition, the operational environment also includes the following software products:

- IBM WebSphere Application Server (WAS) version 7.0.0.11 (includes Java)
- IBM WebSphere Application Server Deployment Manager version 7.0.0.11
- IBM HTTP Server (IHS) 6.1 WAS Plug-in
- IBM GSKit (Global Security Kit) version 7.0.4.33
- Directory Server (LDAP) version 6.1

The hardware platforms, operating systems, IBM WebSphere Application Server, Java, GSKit and Directory Server are not part of the TOE. The hardware platforms and operating systems are not shipped as part of the product. The TAMeb portion of the TOE uses GSKit, a library that implements the TLS v1.0 protocol, to secure its communication channels. The TFIM portion of the TOE does not use GSKit.

1.3.3 Intended Method of Use

The TOE is intended to be used in a web-based environment where access to objects is protected by independent systems with independent authentication mechanisms and user registries. It is intended to provide web single sign-on (SSO) to junctioned applications (a.k.a. target systems) and federated SSO to target systems hosted externally. The operational environment is expected to be a well managed user community in a non-hostile working environment such as a company intranet, well protected from external attacks.

1.3.4 Major Security Features

The TOE provides identification and authentication of users, policy-based access control for protected objects using access control lists (ACLs), auditing of security relevant actions, separation of

ordinary users from administrative users, management of security attributes and features, and, with TFIM, federated single sign-on (SSO).

1.3.5 Definition of Terms

Unauthenticated users	Individuals who are not known to the system but are part of the user community allowed to access resources available to unauthenticated users.
Authorized users	Individuals who have successfully authenticated themselves to the TOE and may access resources as defined by the access control policy of the TOE.
Authorized administrators	Individuals who have successfully authenticated themselves to the TOE as administrators and are allowed to perform administrative tasks within their administrative responsibilities via the TAMeb pdadmin command line interface or via the TFIM administrative command line interface.

1.4. TOE Description

The TOE consists of Tivoli Access Manager for e-business (TAMeb) and Tivoli Federated Identity Manager (TFIM). Since not all TOE usage scenarios make use of TFIM, this document distinguishes between the security functionality provided by the two products so that the reader understands the security functionality differences when the TOE uses TFIM and when the TOE does not use TFIM.

The TAMeb portion of the TOE follows the access control framework defined by the ISO 10181-3 [ISO 10181-3] standard and the Authorization API (aznAPI) [AZNAPI]. TFIM is an identity mapping application used to map identities between disparate organizations allowing TAMeb to connect to other organizations.

The TAMeb portion of the TOE is a complete authorization solution for corporate web, client/server, Tivoli Access Manager applications, and legacy (pre-existing) applications. TAMeb authorization allows an organization to securely control user access to protected information and resources. By providing a centralized, flexible, and scalable access control solution, TAMeb allows users to build highly secure and well-managed network-based applications and e-business infrastructure.

In addition to its state-of-the-art security policy management feature, the TAMeb portion of the TOE supports authentication, authorization, data security, secure communication and resource management capabilities. TAMeb is used in conjunction with standard Internet-based applications to build highly secure and well-managed intranets.

At its core, the TAMeb portion of the TOE provides:

- **Authentication Service** - with TAMeb's range of built-in authenticators.
- **Authorization Service** - The TAMeb authorization service, accessed through a standard authorization API, provides permit and deny decisions on access requests for native Tivoli Access Manager servers and third-party applications.

The TAMeb portion of the TOE contains a component called WebSEAL. WebSEAL is the Resource Manager/Authorization Evaluator responsible for managing and protecting web-based information and resources.

WebSEAL acts as a reverse web proxy by receiving HTTP/HTTPS requests from a web browser and delivering content from its own web server or from junctioned back-end web application servers

(a.k.a. target systems). Requests passing through WebSEAL are evaluated by the TAMeb authorization service to determine whether the user is authorized to access the requested resource.

WebSEAL provides the following features:

- Supports multiple authentication methods - Built-in authentication methods allow flexibility in supporting a variety of authentication mechanisms.
- Accepts HTTP and HTTPS requests.
- Integrates and protects back-end server web resources.
- Provides fine-grained access control for the back-end server web space - Supported resources include URLs, URL-based regular expressions, CGI programs, HTML files, Java servlets, and Java class files.
- Performs as a reverse web proxy - WebSEAL appears as a web server to clients and appears as a web browser to the back-end servers it is protecting.

The TAMeb portion of the TOE uses a Directory Server (i.e. LDAP server) for maintaining user credentials and other TAMeb data. The Directory Server is part of the operational environment.

For secure communication support, TAMeb uses the IBM Global Security Kit (GSKit) library. GSKit provides TLS support for all TAMeb TLS connections and is considered part of the operational environment.

1.4.1 The TAMeb without TFIM Scenario

The evaluated configuration supports the following scenario of TAMeb without TFIM.

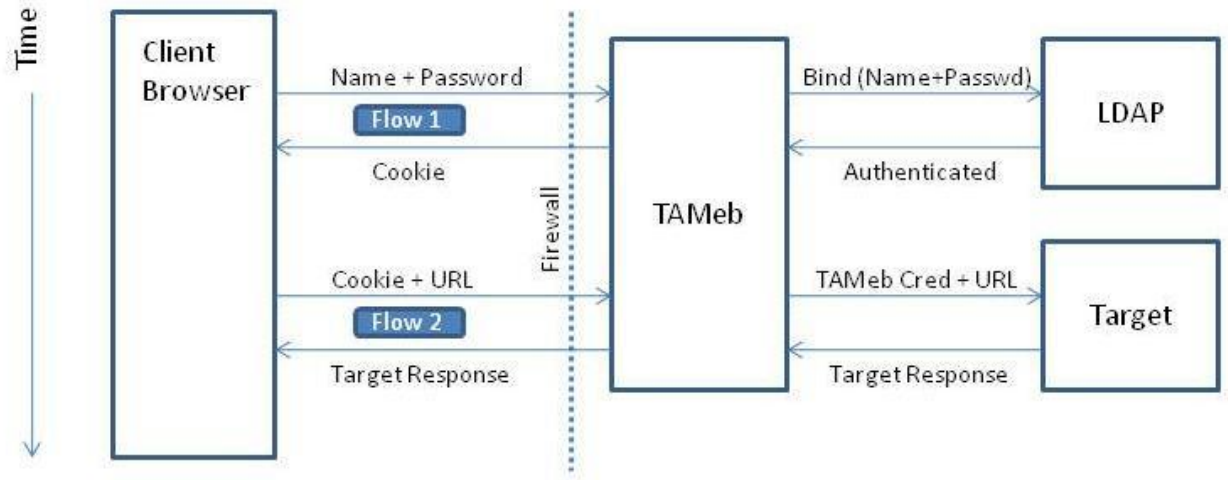

Figure 1-1: Data flow when Target is local to TAMeb

Figure 1-1 shows the data flow of a client web browser session when the target system is local to the TAMeb organization. Starting with Flow 1, TAMeb requires the user of the browser to authenticate to TAMeb, in this case using a user name and password as authentication information. TAMeb passes this information to the directory server (LDAP) in the form of an LDAP bind request. Upon success, the Directory Server returns a successful response and TAMeb creates an internal TAMeb credential

that represents the user which is valid for the length of the session. TAMeb then sends a TAMeb credential reference number to the browser in the form of a browser cookie along with a web page that is displayed to the user.

Figure 1-1 Flow 2 shows the user/browser requesting data in the form of a URL. When the browser requests data, the request is passed to TAMeb in the form of the cookie and the URL. TAMeb maps the cookie value to the TAMeb credential and sends the TAMeb credential and URL to the appropriate target system. The target system interprets the TAMeb credential, checks if the user specified in the TAMeb credential is allowed access to the URL information, and sends its response to TAMeb. TAMeb forwards the target system's response to the browser. Flow 2 can be repeated multiple times as long as the cookie remains valid.

In Figure 1-1, TFIM is not a necessary component because TAMeb knows how to access the local target system(s). In this case, TAMeb acts as both the Identity Provider and the Service Provider.

1.4.2 The TAMeb with TFIM Scenario

The evaluated configuration supports the following scenario of TAMeb with TFIM.

The TFIM portion of the TOE is used when two or more independent organizations having different user registries want to allow access to one or more target systems within one or more organizations. In this case, the organization local to the user contains the Identity Provider and the other organizations contain the Service Providers.

The TFIM portion of the TOE aids in mapping identities between disparate organizations allowing organizations to maintain their current identification mechanisms and control which identities have cross organizational access. TFIM is used as a federated single sign-on (F-SSO) solution allowing users to enter their authentication data once and be granted access to several systems across organizations.

Figure 1-2 shows the data flow of a client web browser session when the target system resides inside another organization's complex. Flow 1 works the same way as in Figure 1-1, where the user sends authentication data to their local $TAMeb_I$ (subscript 'I' for "Identity Provider"), is authenticated locally through the Directory Server, and receives $cookie_I$ referencing a local TAMeb credential.

Figure 1-2 Flow 2 shows the user/browser requesting data by sending $cookie_I$ and the URL, but unlike Figure 1-1, the URL resides in another organization's complex. In order to access the URL, $TAMeb_I$ needs a Security Assertion Markup Language (SAML) [SAML1.1] response to send to the other organization. A SAML response contains the browser user's identity information in a standardized format for use by the Service Provider organization. $TAMeb_I$, acting as an Identity Provider, sends the user's TAMeb credential to $TFIM_I$ and $TFIM_I$ returns a SAML response to $TAMeb_I$ containing the user's identity. The SAML response is also signed by $TFIM_I$ (using the cryptographic functionality provided by the IBM SDK for Java in the operational environment). Then, $TAMeb_I$ sends the SAML response to the browser.

The browser forwards the SAML response to the Service Provider's point-of-contact, $TAMeb_S$. $TAMeb_S$ sends the SAML response to $TFIM_S$. $TFIM_S$ verifies the $TFIM_I$ signature, converts the SAML response into a local TAMeb credential, and sends the TAMeb credential to $TAMeb_S$. $TAMeb_S$ sends a TAMeb credential reference number to the browser in the form of $cookie_S$. From this point on, Figure 1-2 Flow 4 is the same as Figure 1-1 Flow 2 because the user's identity is now temporarily defined on the Service Provider's side for the life of $cookie_S$. The browser sends $cookie_S$ and the URL to $TAMeb_S$. $TAMeb_S$ maps the cookie value to the TAMeb credential and sends the credential and URL to the appropriate target system. The target system interprets the TAMeb

credential, checks if the user specified in the TAMeb credential is allowed access to the URL information, and sends its response to $TAMeb_S$. $TAMeb_S$ forwards the target system's response to the browser. Flow 4 can be repeated multiple times as long as the cookie remains valid.

So TFIM plays two different roles: Identity Provider and Service Provider. As an Identity Provider it receives TAMeb credentials and converts them into SAML responses. As a Service Provider, it receives SAML responses and converts them into TAMeb credentials.

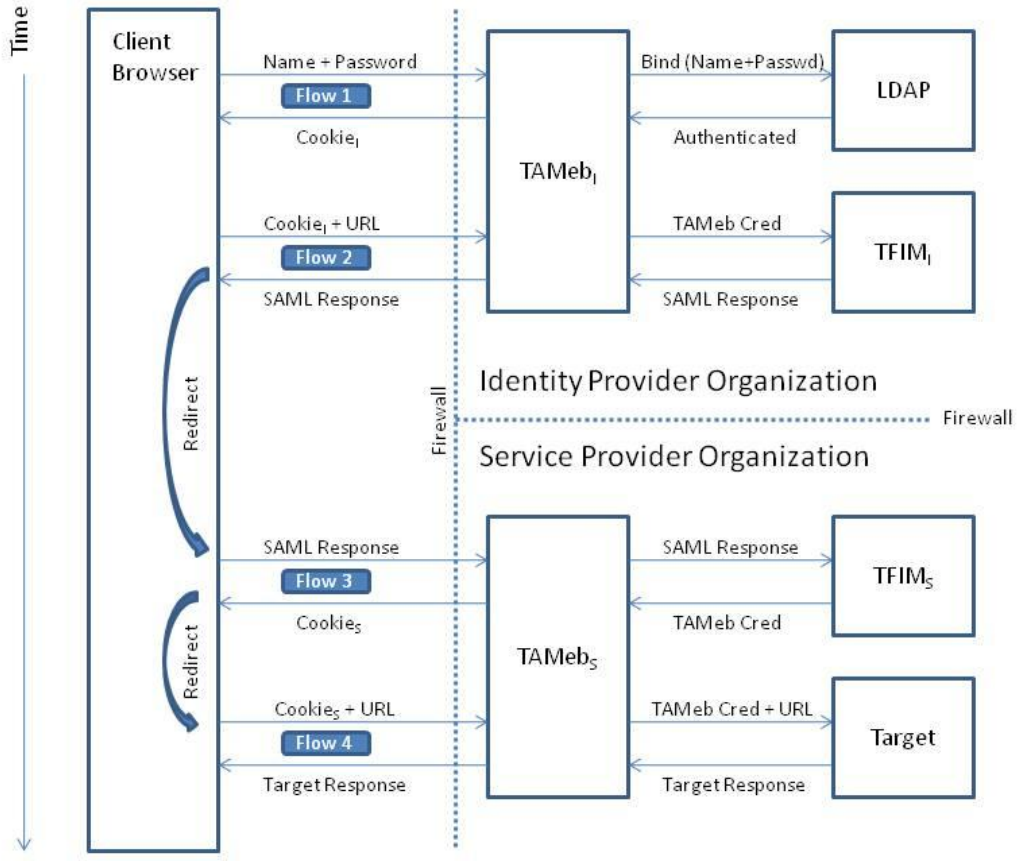

Figure 1-2: Data flow when Target is remote to TAMeb

TFIM uses the IBM WebSphere Application Server (WAS) Deployment Manager server. WAS is a Java-based web application server. WAS, WAS Deployment Manager, and Java are part of the operational environment.

1.4.3 Access Control Framework according to ISO 10181-3 and the Open Group Authorization API

The TAMeb portion of the TOE follows the access control framework defined by the ISO 10181-3

[ISO 10181-3] standard and the Authorization API (aznAPI) [AZNAPI]. The TOE uses the overall access control model and the interface described in those two standards. To explain those ideas we provide a short summary of them.

ISO 10181-3 contains the following diagram to explain the general access control model:

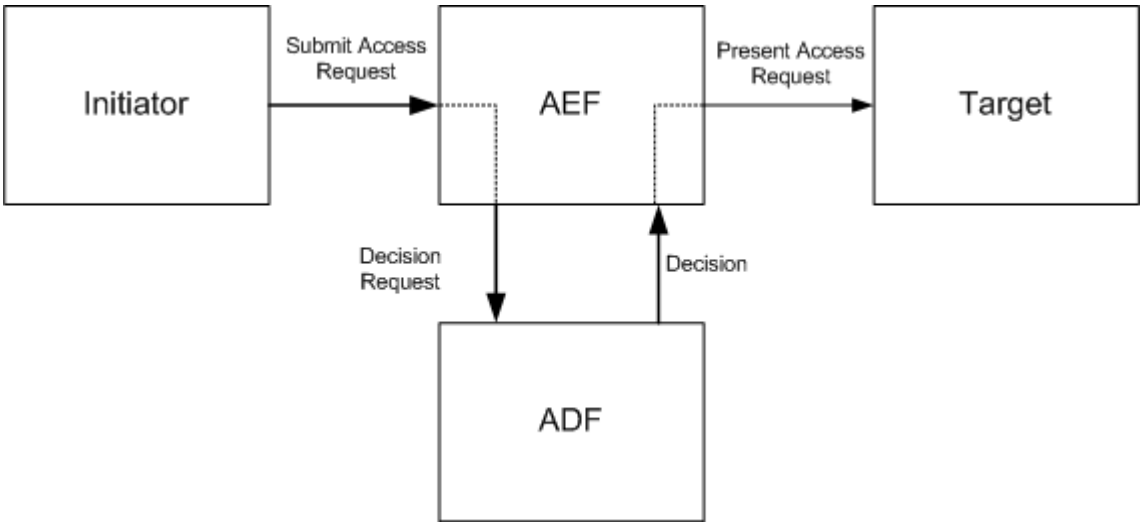

Diagram 1-1: ISO 10181-3 fundamental access control functions

In this model an initiator submits an access request to the "Access Enforcement Function" (AEF) of a system. This function passes the request to an "Access Decision Function" (ADF), which makes the decision based on the rules of the access control system which may be based on:

- The identity and attributes of the initiator
- The identity and attributes of the resource being requested
- Contextual information (e. g. time and date, number of request from the same initiator, information from other systems)

Separating the access enforcement from the access decision function as well as separating the access enforcement function from the actual target allows to implement a highly flexible access control and management systems in distributed environments. ISO 10181-3 actually is a general framework for such kind of access control and management system.

The Open Group now defined a standard for an application programming interface (API) for the interface between the Access Enforcement Function (AEF) and the Access Decision Function (ADF) which allows AEF and ADF components from different vendors to co-operate. The following diagram shows the aznAPI system structure as defined in [AZNAPI]

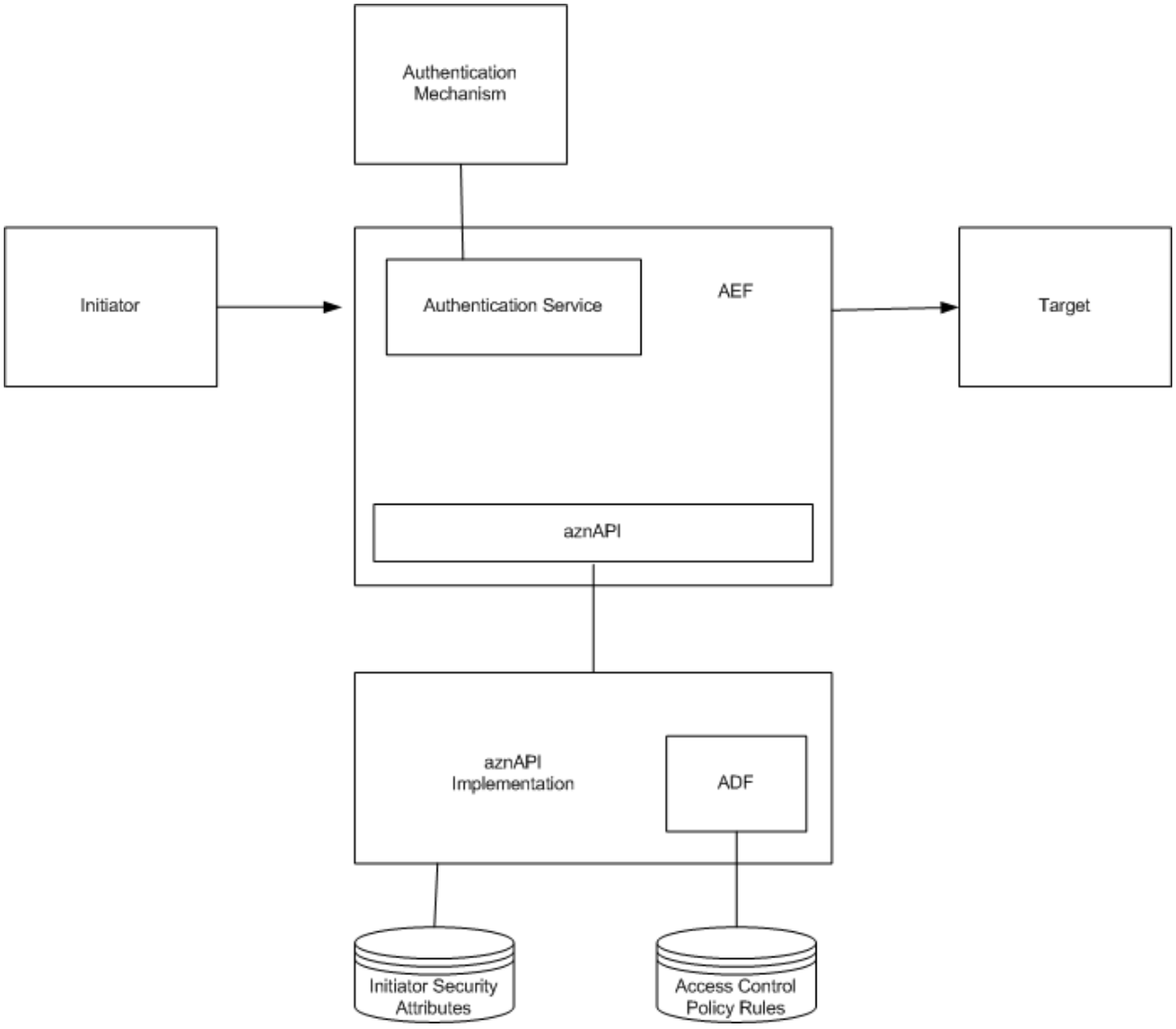

Diagram 1-2: aznAPI System Structure as defined in the Open Group Standard

In this model the initiator submits his access request to the AEF, which then (if required by the policy) authenticates the identity of the initiator using the authentication service within the AEF. This authentication service may use an external authentication mechanism (e. g. a Directory Server storing user attributes and credentials).

The request together with the initiator attributes is then passed via the aznAPI interface to the implementation of the aznAPI, which in turn may itself store or request from an external entity additional security attributes of the initiator of the request. Those together with the information passed via the aznAPI about the request (including the target of the request) as well as the information about the initiator of the request is passed to the ADF component, which uses the Access Control Policy Rules stored in some kind of database.

1.4.4 Mapping the TOE to the aznAPI System Structure

The TAMeb portion of the TOE follows the access control model defined in [ISO 10181-3] and [AZNAPI]. The overall TOE architecture is illustrated in Figure 1-3 and Figure 1-4 where the dotted line indicates the TOE boundary. With relation to the model defined in Diagram 1-2 the TOE includes the Access Enforcement Function (AEF) and the Access Decision Function (ADF) together with the Access Control Policy rules. The Authentication Mechanism is implemented with GSKit, which itself is not part of the TOE. The "Initiator Security Attributes" Database is implemented using a Directory Server, which itself is not part of the TOE. Also the Target system which has the actual resource to be protected is not part of the TOE.

In this model, a user on a client submits a request for a resource (e. g. accessing a URL on a network protected by the TOE). This request is intercepted by the TOE (much in the same way as an application gateway firewall system intercepts network requests). The TOE performs the following actions:

- Checking if the requested resource is protected but accessible to unauthenticated users and accessible without requiring SSO (i. e., the user and resource are located in the same TAMeb organization). If this is true, the request is passed through.
- Checking if the user has already been authenticated (i. e. there is a protected session where the user has been authenticated). If not, the user is required to authenticate (this is the case for password based authentication. Certificate based authentication will always take place when the session is established. Please read section 7.1.2 on authentication for details). This authentication makes use of an external Directory Server which stores user attributes and user credentials.
- Checking if the user has the right to access the requested resource in the requested mode. If not, the request is rejected. If yes, the request is passed through to the server holding the resource (the TOE works like a reverse proxy here).

To explain how the access rights are checked, an overview on the TAMeb components is provided first (please see Figure 1-3 and Figure 1-4 for an architectural overview of the TOE).

The "Resource Manager" is implemented within the TOE by the WebSEAL component. This component includes also the "Authorization Evaluator" as a subsystem. The Resource Manager communicates with the "Authorization Evaluator" via the aznAPI.

The "Policy Server" is responsible to define and maintain the access control policy. It uses the "Master Authorization Policy" database to store the access control policy rules.

To improve performance, the "Authorization Evaluator" manages a replica of the "Master Authorization Policy". The Policy Server informs all Authorization Evaluators about modifications to the "Master Authorization Policy" (actually what it does is to use a standard compression utility to compress the whole database and the transfer the whole new database). An Authorization Evaluator can also request the Policy Server to submit a new copy of the Master Authorization Policy (which it does upon startup, since there may be updates it did not get during a down-time). Also the Policy Server can demand an Authorization Evaluator to update the replica of the Master Authorization Policy in cases it is not sure that it has the latest version.

Administration of the TAMeb portion of the TOE is done via a workstation or terminal directly connected to the Policy Manager component. Only the command line interface and C language API for administration are part of the evaluated configuration. The C language API may be used by an organization to define its own tools to automate some of the administration tasks. But of course such

tools are not part of the evaluated configuration and it is up to the organization to ensure that those tools perform their task correctly.

Administration includes the management of the Master Authorization Policy (defining access rules for protected objects) as well as management of the TOE. It should be noted that access rights of administrators to administrative objects of the TOE are also stored and maintained in the Master Authorization Policy.

To perform authentication, TAMeb uses an external Directory Server supporting the LDAP protocol. The Directory Server is used as a repository for user and administrator attributes and credentials. Authentication of users is done by the Resource Manager, authentication of administrators is performed by the Policy Server (in the sense of the authentication service in Diagram 1-2) and both use the external Directory Server as the authentication mechanism.

The TAMeb communication channels are protected from modification and disclosure using the Transport Layer Security (TLS) v1.0 protocol. TLS is implemented by the operational environment and, therefore, is not part of the TOE. The following list specifies the protected TAMeb channels in the evaluated configuration that are protected by the operational environment:

- TAMeb and the client systems
- TAMeb and the target systems
- TAMeb and TFIM
- TAMeb and the LDAP server
- Policy Server and WebSEAL(s)

The TAMeb portion of the TOE uses the GSKit library for the implementation of the TLS protocol and its underlying cryptographic functions. The GSKit library is, therefore, part of the Policy Server and part of WebSEAL, but is considered part of the operational environment.

The Master Authorization Policy as well as the Replica Authorization Policy are databases. The Master Authorization Policy is a database held by the Policy Server and the Replica Authorization Policy is a database held by each Authorization Evaluator.

This TOE architecture (showing also the Directory Server and the servers holding the resources, although they are not part of the TOE) is shown in Figure 1-3 and Figure 1-4. Note that in Figure 1-3 and Figure 1-4, only the components shown in white are in the TOE. The components in blue are in the operational environment. The WAS Nodes, WAS Deployment Manager, and WAS Cell are part of the operational environment.

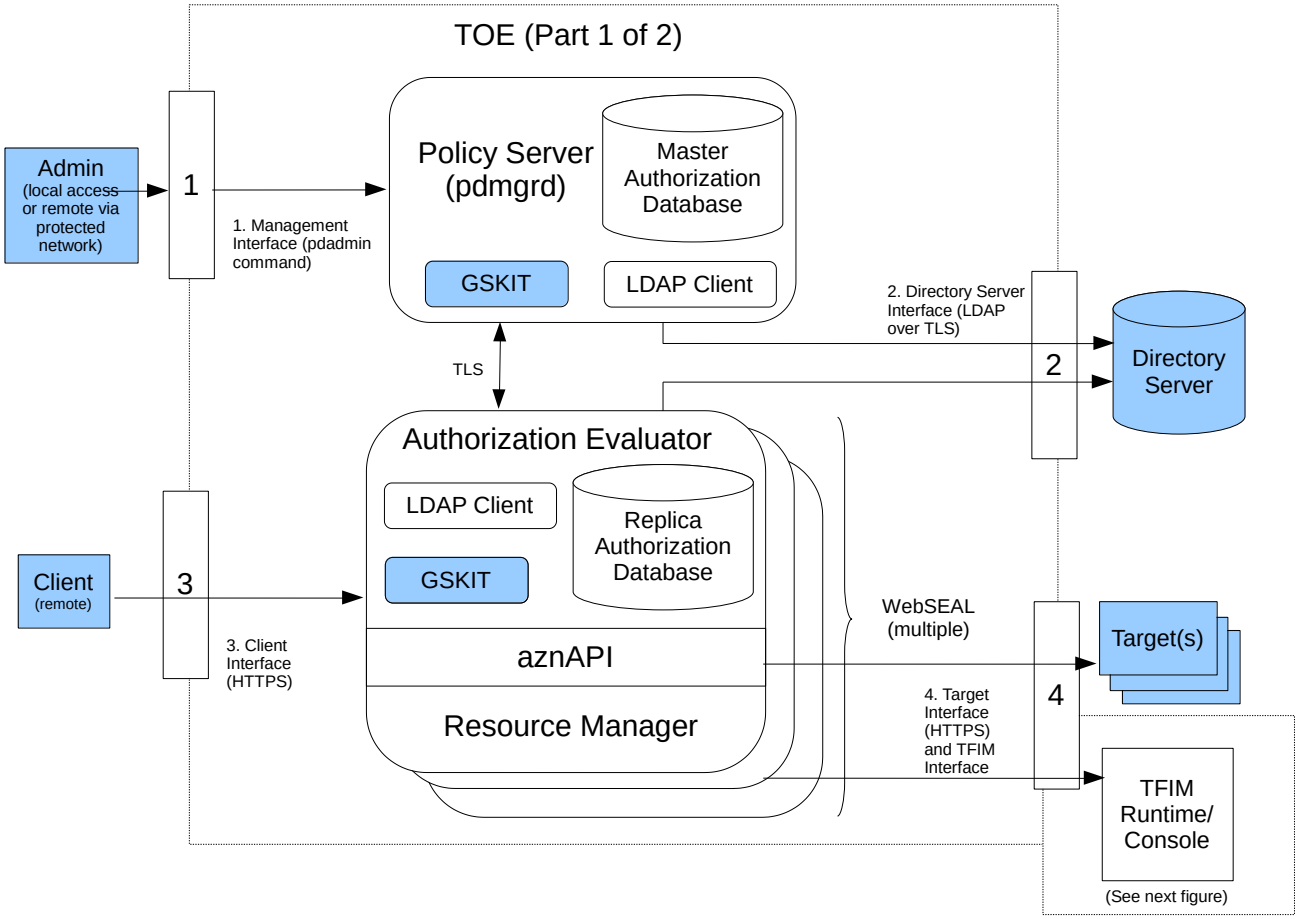

Figure 1-3: TOE logical boundary part 1

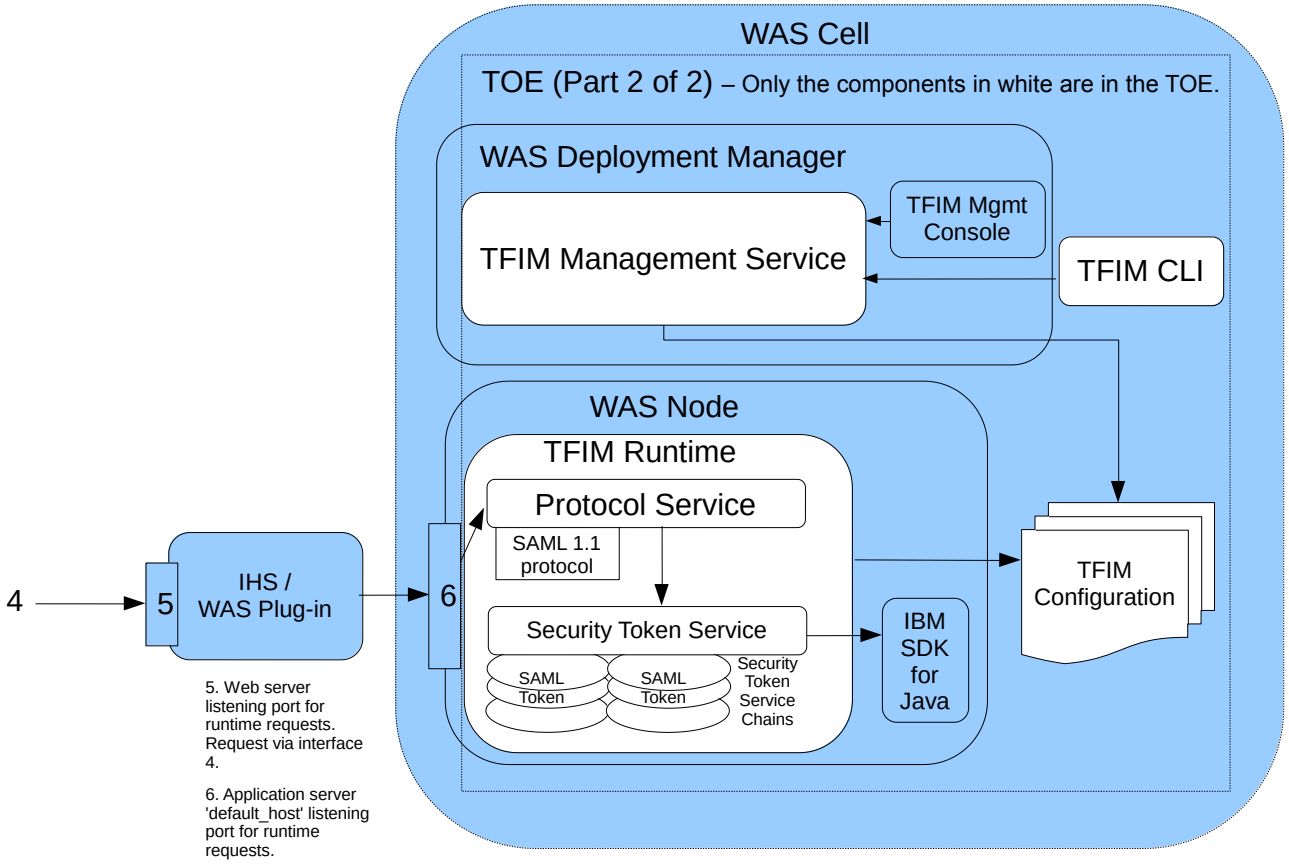

Figure 1-4: TOE logical boundary part 2

The TOE now maps in the following way to the system structure shown in Diagram 1-2 as defined in [AZNAPI]:

- The “Initiator” maps to the client.
- The “Access Enforcement Function” (AEF) **and** the “Authentication Service” map to the Resource Manager (part of WebSEAL). In addition, the “Access Enforcement Function” maps to TFIM.
- The “Access Decision Function” (ADF) maps to the “Authorization Evaluator” (part of WebSEAL).
- The “aznAPI” maps to the “aznAPI”.
- The “aznAPI” implementation maps to the “Authorization Evaluator” **and** the “Policy Server”.
- The “Authentication Mechanism” **and** the “Initiator Security Attributes” map to the “Directory Server” (the Directory Server is part of the operational environment).
- The “Access Control Policy Rules” map to the “Master Authorization Policy” **and** the “Replica Authorization Policy”.
- The “Target” maps to the “Target”.

1.4.5 Resource Manager

The Resource Manager shown in the previous Figure 1-3 as part of WebSEAL is responsible to protect the external resources on the target system (web servers).

The WebSEAL Resource Manager acts as a reverse web proxy by receiving HTTP/HTTPS requests from a web browser and delivering content from junctioned back-end web application servers (a.k.a. target systems). Requests passing through the WebSEAL Resource Manager are evaluated by the Authorization Evaluator (see Figure 1-3) to determine whether the user is authorized to access the requested resource.

The WebSEAL Resource Manager provides the following features:

- Supports multiple built-in authentication services (only password-based and client certificate-based authentication are supported in the evaluated configuration). Certificate-based authentication supports three separate modes of operation determining when a certificate is requested and how access is granted. It uses an external Directory Server storing user attributes and credentials as supporting mechanism for the authentication services. Support of certificates is implemented via GSKit which is external to the TOE.
- Accepts HTTPS requests (Only TLS v1.0 secured connections are accepted whenever a client establishes a connection to a WebSEAL component. Attempts to establish a connection not secured by TLS v1.0 will be rejected.)
- Integrates and protects back-end server resources through WebSEAL junction technology

  Supported resources include (but is not restricted to) URLs, URL-based regular expressions, CGI programs, HTML files, Java servlets, Java class files, Active Server Pages and other scripts.
- Performs as a reverse web proxy

  The WebSEAL Resource Manager appears as a web server to clients and appears as a web browser to the junctioned back-end servers it is protecting. It is in the junctioned part of TAMeb where TFIM connects and is used to aid in communicating with back-end servers that exist in other organizations.

1.4.6 Authorization Evaluator

The Authorization Evaluator component is running as a part of a WebSEAL system. When the Resource Manager component calls functions of the aznAPI to check if the client has the right to access the resource in the intended way, the Authorization Evaluator will check the local replica of the Master Authorization Policy to decide if the request can be granted or not.

The Authorization Evaluator is also responsible for synchronizing the local replica of the Master Authorization Policy with the Policy Server. The Authorization Evaluator may ask the Policy Server if his version is still up-to-date (which it always does on start-up and continuously during operation) and will store a new version of the replica database on demand of the Policy Server. In addition the Authorization Evaluator will serve additional system management commands coming from the Policy Server.

The Authorization Evaluator sets up a secured communication channel with the Policy Server using the TLS v1.0 protocol with client and server authentication. The Policy Server will take the role of the server while the Authorization Evaluator will take the role of a client.

1.4.7 Policy Server

The Policy Server component is responsible for the management of the Master Authorization Policy. For this management, it provides a separate interface for administrators (the TAMeb pdadmin interface) as a command line interface as well as a C API on the Policy Server. To perform administrative actions an administrator has to identify and authenticate himself via these interfaces. Only password based authentication is possible at this interface.

An administrator using a remote terminal or remote workstation to connect to the Policy Server for administration needs to ensure that the remote terminal or workstation are in a secured environment and managed securely. Management is performed by setting up a secured connection to communicate with a shell of the operating system on the Policy Server. There it calls the pdadmin command line interface and authenticates himself. Note that the security measures to protect the remote terminal or remote workstation as well as the security measures used to protect the communication link between the remote terminal or workstation are not part of the TOE but have to be assured in the operational environment.

Administrators can now define and/or modify rules in the Master Authorization Policy as well as perform administrative actions for the Authorization Evaluator or Resource Manager components. Whenever they modify the Master Authorization Policy a request for synchronization of their replica database is sent to all Authorization Evaluator components.

The Policy Server uses the Master Authorization Policy to store access control rules to system management objects and uses the Directory Server to store attributes and credentials of administrators. Management of the TOE can only be performed via the pdadmin interface of the Policy Server, since the "management objects" are not known to the Resource Manager or the Authorization Evaluator and therefore not accessible via a client interface to the TOE.

1.4.8 LDAP Client

The LDAP Client is a C API library that implements the LDAP client-side protocol. It uses the GSKit library to provide TLS v1.0 protected communications to the Directory Server. This library is used by both the Resource Manager and the Policy Server. GSKit is part of the operational environment.

1.4.9 TFIM

TFIM is a separately installable product responsible for providing federated single sign-on (F-SSO) capability to TAMeb. The TFIM portion of the TOE contains the following major component(s) (see Figure 1-3 and Figure 1-4):

- TFIM Runtime
- TFIM Management Service

The TFIM Runtime runs on an IBM WebSphere Application Server (WAS). It interfaces with TAMeb via the WebSEAL External Authentication Interface (EAI) over a TLS connection. The TFIM TLS implementation is part of the Java Virtual Machine (JVM) which resides in the operational environment.

The TFIM portion of the TOE can act as both an Identity Provider and a Service Provider in a federated system. As an Identity Provider, TFIM receives a TAMeb credential from TAMeb, converts it into a SAML response, and sends the SAML response to TAMeb. When creating a SAML response, TFIM signs the response with its X.509v3 credential's private key. The cryptographic code used by TFIM for signing the SAML response is part of the operational environment, not of TFIM.

As a Service Provider, TFIM receives a SAML response from TAMeb, validates the signature of the SAML response, converts the response into a TAMeb credential, and sends the TAMeb credential to TAMeb. The cryptographic code used by TFIM to validate the SAML response signature is part of the operational environment, not of TFIM.

In the evaluated configuration, TFIM supports only the SAML 1.1 Browser/POST Profile scenario defined in [SAML1.1] section 4.2. Also, administrators must restrict identity mappings to one-to-one mappings of Identity Provider credentials to Service Provider credentials in the evaluated configuration.

In addition, the TFIM Runtime also performs auditing and maintains an audit log separate from the TAMeb audit log. Although the TFIM Runtime performs auditing in the evaluated configuration, the auditing subsystem must only be managed while the TFIM Runtime is offline (through the use of the TFIM Management Console which is not part of the evaluated configuration).

The TFIM Runtime is managed using a set of administrative command line interfaces local to the TFIM Management Service. These commands can only be executed by users who have administrative access to the TFIM Management Service. (The command line interface does not support the management of audit.) Administrator user account names and passwords are created and managed by WAS in the operational environment.

The following TFIM components are not part of the evaluated configuration:

- TFIM Management Console

The TFIM command line interface (CLI) provides administrators with an interface to manage F-SSO data through the TFIM Management Service component. Access is managed by WebSphere in the operational environment. The configuration data managed by the CLI resides with each TFIM instance. All requests to modify the configuration data are sent to and handled by the TFIM Management Service.

TFIM uses the Java runtime TLS, which is part of the operational environment, for all secure communications.

1.4.10 TOE Configuration

The following describes the specifics of the configuration of TAMeb and TFIM that conforms to the description in this Security Target and is, henceforth, called the evaluated configuration:

- The Policy Server component of the TOE is installed and operated on a dedicated system that communicates via a network connection to WebSEAL (the Resource Manager/Authorization Evaluator).
- The Resource Manager and Authorization Evaluator are installed and operated on the same system. They communicate with each other via a library interface (the aznAPI). They communicate with the Policy Server via a network connection with a dedicated application layer protocol running over TLS v1.0. The TLS protocol implementation is part of the operational environment.

  Note that the evaluated configuration does not include Authorization Evaluator components running on a machine separate from the Resource Manager that uses them.
- The evaluated configuration has one Policy Server and one or more Resource Manager/Authorization Evaluator systems. All Resource Manager/Authorization Evaluator systems operate independent from each other and are only connected to the central Policy

Server. Load balancing and failover configurations of Resource Manager/Authorization Evaluator systems are therefore not supported in the evaluated configuration.

- The Policy Server and all the Resource Manager/Authorization Evaluator use the same operating system as a basis. Configurations using different operating system platforms for different components of the TOE are not part of the evaluated configuration.
- Communication between client systems and the TOE, the web server systems and the TOE, the LDAP server and the TOE, TAMeb and TFIM, and the Policy Server and the Resource Manager/Authorization Evaluator is protected using the TLS v1.0 protocol. The use of unencrypted communication is disabled in the TOE. Also the use of version 2 and version 3 of the SSL protocol is disabled. All components are configured to use TLS v1.0. The external LDAP server also needs to support TLS v1.0 and be configured to use TLS v1.0. All TLS implementations are part of the operational environment.
- No hardware encryption device is used. The TAMeb cryptographic services are provided by the software implementation of the GSKit component. GSKit is part of the operational environment.
- FIPS mode (for GSKit) must be turned on in the evaluated configuration.
- TFIM must have Java Security enabled along with FIPS enabled in the evaluated configuration.
- In the evaluated configuration, only the following ciphers must be used with WebSEAL:
    - Triple DES (TDEA), AES (128 bit and 256 bit keys)
- The TOE is configured to use password based authentication and TLS client certificate based authentication for the authentication of users. In addition, the WebSEAL External Authentication Interface (EAI) must be enabled.
- The TOE is configured to use password based authentication for administrators that request access to the TOE via the TAMeb pdadmin command line interface or the TAMeb C API.
- The use of the Web Portal Manager component for the administration of the TOE is **not** supported. Instead only the TAMeb pdadmin command line interface, the TAMeb C API, and the TFIM administrative command line interface are supported in the evaluated configuration.
- Only LDAP is supported for the access to the Directory Server in the evaluated configuration. Active Directory or other protocols are not supported. LDAP Replicas are also not supported.
- The TOE uses only the English language pack.
- Multiple TAMeb registry domains are not supported by the TOE and only the default domain is used.
    - Thus, export and import of security policies (Protected Object Policies (POPs), Access Control Lists (ACLs), authorization rules, and objects) to other Tivoli Access Manager domains are not supported in the evaluated configuration.
- TAMeb Authorization rules are not supported in the evaluated configuration.
- 'TAMeb registry attribute entitlements service' is not supported in the evaluated configuration.

- The Policy Proxy Server is supported by the TOE, but is considered part of the operational environment, not the TOE. The Policy Proxy Server must have security policy caching disabled.
- The integration of the IBM Tivoli Identity Manager is not supported in the evaluated configuration.
- The TOE supports the usage of IPv6 except IPv6 POP based network authentication is not supported in the evaluated configuration.
- The use of Access Manager Session Management Server (SMS) is not supported in the evaluated configuration.
- The use of the Common Auditing and Reporting Service (CARS), also known as the Common Audit Service (CAS), is not supported in the evaluated configuration.
- The transparent path junction option for WebSEAL is not supported in the evaluated configuration.
- WebSEAL's support for maintaining session state is limited to TLS IDs and session cookies in the evaluated configuration.
- Only the "Minimal" LDAP data format (selected during the installation of the Policy Server) is supported in the evaluated configuration.
- Only SAML 1.1 Browser/POST Profile is supported in the evaluated configuration.
- The TFIM alias service is not supported in the evaluated configuration.
- WebSEAL's switch user functionality must be disabled (i.e., user impersonation is disallowed).
- TFIM must be configured to support only one-to-one mappings of Identity Provider credentials/accounts to Service Provider credentials/accounts.
- TFIM must have "WebSEAL No ACLD" selected as the point of contact server.
- TFIM clusters are not supported in the evaluated configuration.
- The TFIM Management Console is not to be used in the evaluated configuration.

To set up the evaluated configuration compliant with the description above the user needs to follow the guidance documentation for the evaluated configuration as found in the [CCGuide].

The TAMeb system components to be installed are:

1. Policy Server:
   - IBM Global Security Toolkit (GSKit) 7.0.4.33 – (operational environment)
   - IBM Directory Client 6.1.0.6 – (TOE – a.k.a. LDAP Client)
   - Tivoli Access Manager runtime 6.1.1 FP4 – (TOE)
   - Tivoli Access Manager policy server 6.1.1 FP4 – (TOE)
2. Resource Manager/Authorization Evaluator (WebSEAL)
   - IBM Global Security Toolkit (GSKit) 7.0.4.33 – (operational environment)
   - IBM Directory Client 6.1.0.6 – (TOE – a.k.a. LDAP Client)

- Tivoli Access Manager runtime (including the Authorization Evaluator) 6.1.1 FP4 – (TOE)
- Tivoli Access Manager for e-business WebSEAL server 6.1.1 FP4 – (TOE)

The TFIM system components to be installed are:

1. Tivoli Federated Identity Manager Runtime / Management Service 6.2.1 FP2 – (TOE)
2. Tivoli Federated Identity Manager Management Console 6.2.1 FP2 – (Used only to configure TFIM's auditing when TFIM is offline)

TFIM, Policy Server, and all Resource Manager/Authorization Evaluator within an evaluated configuration use the same operating system platform (but run on different machines). Those platforms are defined in section 1.3.2. The TAMeb and TFIM component version numbers provided above are valid for all operating system platforms defined in section 1.3.2.

1.4.11 TOE Boundary

Figure 1-3 and Figure 1-4 show the logical boundary of the TOE. They show that the TFIM Runtime (a.k.a. TFIM server), TFIM Management Service, Policy Server, the Resource Manager/Authorization Evaluator, the Master Authorization Policy database as well as the Replica Authorization Policy database are part of the TOE. They also shows that the WebSphere Application Server (for TFIM), WAS Deployment Manager, Directory Server, the Client system as well as the web application servers (a.k.a. target systems) are all part of the operational environment. The TFIM Management Console is to be used only to configure TFIM's auditing when TFIM is offline.

The TAMeb component Policy Proxy Server that is mentioned in section 1.4.10 is not part of the TOE.

1.4.12 Security Model

The Security Model has the following components:

1. A User Registry (LDAP). (The LDAP Server is part of the operational environment.) The user registry contains all users and groups allowed to participate in the TAMeb secure domain.
2. A Master Authorization Policy Database

   This database contains a representation of all resources in the domain (= protected object space). The security administrator can define Access Control Lists (ACLs) and Protected Object Policies (POPs) for those resources that require protection
3. An Authentication Service

   This service verifies the claimed identity of a user. All users that are going to be authenticated must have an entry in the User Registry. Users that are not authenticated can only access resources where the resource and the type of access are allowed for unauthenticated users.

   When a user is successfully authenticated a set of identification information (credentials, which include user identity, group membership and security attributes) is extracted from the information stored in the User Registry and maintained for the user within the TOE.
4. An Authorization Service

For each attempted access this service verifies if the user attempting access has the right to access the resource in the intended way. This is done by comparing the user's credentials with the rules defined for the resource in the Authorization Policy Database. The Authorization Service is called by the Resource Manager and return "yes" or "no" depending on the evaluation of the rules from the database.

5. An Audit Service

   A configurable number of events will generate an audit record that allows to trace when the event has happened and which user caused the event.

6. An Administration Interface

   This interface is used to administer the TOE (including the WebSEAL systems and TFIM).

7. Configuration Files

   A number of configuration files are used by the components of the TOE. The settings of those files define the behavior of the security functions of the TOE. Configuration files need to be defined correctly at the installation time of the TOE to ensure a secure initial configuration. The administrator will maintain configuration files either by using the administration commands of the TAMeb pdadmin interface or directly by editing the files.

8. Secure Communication

   Communication channels to the TOE and between the components of the TOE are protected using TLS v1.0 communication security mechanisms. In the case of TAMeb, GSKit is used to provide TLS protected channels. (See section 1.4.4 for a list of protected channels and their supported protection protocols.) In the case of TFIM, Java cryptographic functions provide the communication security.

9. Security Token Service

   The Security Token Service (STS) is the backbone subcomponent of TFIM, providing an implementation of WS-Trust, an open standard specifying a protocol for exchanging a security token of an arbitrary incoming type for a token of arbitrary outgoing type. STS provides a framework for adding (plugging in) varies security token modules, such as SAML 1.1, to support the issuing, renewing, and validating of security tokens from different domains. The underlying cryptographic functionality of the STS is provided by the IBM SDK for Java, part of the operational environment.

The TOE has the following security functionality:

1. Authentication of administrators

   Administrators allowed administering the TOE via the TAMeb pdadmin interface or the C API are identified and authenticated. (LDAP is involved in the authentication process. LDAP is part of the operational environment.)

2. Authentication of users

   Users are identified and authenticated. The TOE requires the authentication of users either by a userid/password combination verified by LDAP (LDAP is part of the operational environment) or by authenticating the users with X.509v3 certificates using GSKit (GSKit is part of the operational environment). **Note: This implies some trust of the client system to protect the user's authentication data. This is expressed in the assumptions and requirements for client**

systems.

3. Assigning user credentials to authenticated users

   The credentials of authenticated users are created from the information stored in the user's record in the user registry within the Directory Server.

4. Access Control to protected objects of the web application servers (a.k.a. target systems)

   Those objects are protected as defined in the policy defined by an administrator on the Policy Manager.

5. Access Control to TOE management objects

   A flexible management model allows limit the administration capabilities of administrative users to defined sections of the protected object space

6. Auditing of activities

   The TOE is capable of auditing defined events.

7. Security management

   The TOE provides security management for the security functionality of the TOE and security management for most of the security attributes of the TOE.

2. CC Conformance Claim

This ST is CC Part 2 conformant and CC Part 3 conformant, with a claimed Evaluation Assurance Level of EAL4, augmented by ALC_FLR.3.

This ST does not claim conformance to any Protection Profile.

Common Criteria [CC] and Common Evaluation Methodology [CEM] version 3.1, revision 3 have been taken as the basis for this conformance claim.

3. Security Problem Definition

3.1. Threat Environment

The assumed security threats in the operational environment are listed below.

The **assets** to be protected by the TOE comprise the information stored, processed or transmitted by the TOE. The term "information" is used here to refer to all data held within the TOE or parts of the TOE, including data in transit between parts of the TOE, if appropriate. This does not include resources managed in the operational environment of the TOE representing the target of access requests, since the TOE is only responsible for the access decision making but not the enforcement of the access control in the operational environment (the access control decision is in turn achieved by relying on the access control policy rules, which again is information held within the TOE and therefore has to be protected by the TOE). The TOE counters the general threat of unauthorized access to information, where "access" includes disclosure, modification and destruction.

The **threat agents** can be categorized as either

- unauthenticated users of the TOE, i.e. individuals who are not known to the system but may access resources available to unauthenticated users
- authorized users of the TOE, i.e. individuals who have successfully authenticated themselves to the TOE and may access resources as defined by the access control policy via the Resource Manager component

The threat agents are assumed to originate from a well managed user community in a non-hostile working environment, and hence the product protects against threats of inadvertent or casual attempts to breach the system security. The TOE is not intended to be applicable to circumstances in which protection is required against determined attempts by hostile and well funded attackers to breach system security. An example of such an environment is a company intranet well protected from external attacks and with an overall user community (including unauthenticated users) that can be assumed to be non-hostile. System administrators of the TOE as well as those for the underlying operating system and the Directory Server used are assumed to be trustworthy and follow the instructions provided to them with respect to the secure configuration and operation of the systems under their responsibility.

3.1.1 Threats Countered by the TOE

T.UAACTION	An undetected violation of the TSP may be caused as a result of an attacker (either an unauthenticated user or an authorized user) attempting to perform actions that the individual is not authorized to do, thus, allowing the attacker to gain unauthorized access to TSF data and/or user data.
T.UAUSER	An attacker (either an unauthenticated user or an authorized user) may impersonate an authorized user of the TOE including the threat of an authorized user that tries to impersonate as another authorized user without knowing the authentication credentials and gain unauthorized access to TSF data and/or user data.

3.2. Assumptions

3.2.1 Environment of Use of the TOE

A.NOBYPASS	It has to be ensured that protected resources cannot be accessed in a way that bypasses the TOE and that all internal and external access attempts to protected resources have to be channeled through the TOE.
A.CLIENT_KEYMAN	Users have to administer and protect private keys of their client system used for authentication and key exchange with the TOE in a secure way. This includes the secure generation of strong keys as well as the protection of private keys against any kind of unauthorized access and use.
A.CLIENT_PWMAN	Users have to protect their passwords used for authentication to the TOE such that no unauthorized access to them is possible.
A.ADM_PWMAN	Administrators have to protect their passwords used for authentication to the TOE such that no unauthorized access to them is possible.
A.PHYS_PROT	The machines running the TOE software need to be protected against unauthorized physical access and modification. All machines running parts of the TOE software require this protection.
A.SINGLE_APP	Any machine used to run all or a part of the TOE software are assumed to be used solely for this purpose and are not used to run other application software except those required for the management and maintenance of the underlying operating system and hardware.
A.OS_CONF_MGMT	The operating system of the machines running the TOE are assumed to be configured and maintained by trained and trustworthy personnel such that the operating system provides a reliable basis for the operation of the TOE software. Especially it is assumed that the operating system is configured such that no unauthorized access to functions provided by the operating system software (including network daemons) is possible either locally or via any network connection.
A.ADMIN	The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation. They will perform administration activities from a secure environment using terminals and/or workstations they trust via secured connections to the Policy Server.

A.DIR_PROT	The Directory Server used by the TOE provides protection mechanism against unauthorized access to TSF data stored in the directory. This includes the requirement for authentication when accessing user entries and the configuration to use TLS v1.0 as the preferred protocol to protect the communication links.
A.CRED_PROT	The operational environment protects credentials against unauthorized access in order to prevent attackers from impersonating an authorized TOE user and to prevent attackers from getting unauthorized access to the directory information.
A.PROTOCOL_SEC	The operational environment components that implement TLS used by the TOE and the operational environment components generating and interpreting SAML responses implement their security protocols and cryptographic functions correctly.
A.PROTECTED_NET	The TOE components reside within a protected network where WebSEAL's HTTP/HTTPS ports are the only visible external interfaces. In addition, WebSEAL's HTTP/HTTPS ports are protected by a firewall that allows only inbound HTTP/HTTPS requests via the visible external interfaces.
A.AUDIT_CONFIG	The TFIM audit mechanism will only be configured during TFIM installation or when TFIM is in an offline mode of operation.
A.SEC_COM	Communication of TOE external entities with the TOE as well as communication between physically distributed parts of the TOE are secured using TLS to ensure the integrity and confidentiality of the communication.

3.3. Organizational Security Policies

P.AUTHORIZED_USERS	Only those users who have been authorized to access web resources protected by the TOE may access those resources after they have been successfully authenticated (unless a protected web resource is defined to be accessible by unauthenticated users, in which case no prior authentication is required).
P.AUTHORIZED_SERVER	Only TAMeb servers and TFIM servers authorized for access to TSF data may access this data after the paired servers have been successfully authenticated.
P.AUTHORIZED_ADMIN	Only administrators authorized for access to defined management resources of the TOE may access those resources after they have been successfully authenticated.

P.NEED_TO_KNOW	The system must allow to limit the access to, modification of, and destruction of the information in protected web resources to those authorized users which have a "need to know" for that information.
P.ACCOUNTABILITY	The administrators of the system shall be held accountable for their actions within the system.
P.ADM_DELEGATION	Specific administration tasks as well as management operations to defined subsets of the web resources protected by the TOE may be delegated to administrators that are only allowed to perform the management tasks within their defined area of responsibility and are not able to extend this area themselves.
P.PWD_STRENGTH	Passwords for both administrative accounts and user accounts should have sufficient strength as commensurate with the importance of the information protected by the accounts.

4. Security Objectives

4.1. Objectives for the TOE

O.ACC_ADM	The TSF must control the definition and management of access control rules and policies and restrict those activities to authorized administrators. The TSF must allow to restrict the rights of some administrators to define access control rules for a subset of the protected object space only.
O.ACCESS_DECISION	The TSF must base its access decision on defined access control rules and policies defined by administrators.
O.AUDITING	The TSF must record the security relevant actions of users (including administrators) of the TOE. The TSF must present this information to authorized administrators.
O.AUTHENT_ADMIN	The TAMeb must authenticate administrators which request access to the TOE and its resources. **Note:** The access policy rules may define some resources that can be accessed by everybody including unauthenticated users. Those resources are not seen as part of the resources protected by the TOE.
O.AUTHENT_SERVER	Paired TAMeb and TFIM servers must authenticate each other prior to transferring TSF data.
O.AUTHENT_USER	The TSF must authenticate users which request access to resources protected by the TOE unless the resource is allowed to be accessed by unauthenticated users.
O.AUTHORIZATION	The TSF must ensure that only authorized administrators, users, and TOE servers gain access to the TOE and the resources it protects. **Note:** The access control rules may also allow unauthenticated users to access resources explicitly defined to be accessible to unauthenticated users.
O.MANAGE	The TSF must provide all the functions and facilities necessary to support the authorized administrators that are responsible for the management of TOE security.
O.PWD_STRENGTH	The TOE must enforce a minimum password policy for accounts created and maintained by the TOE.
O.TIME	The TOE must provide reliable time stamps.

4.2. Objectives for the Operational Environment

4.2.1 Objectives for the underlying operating system

OE.OS_CONFFILE_PROT	The operating system within the operational environment must provide protection of the configuration files against unauthorized access.

4.2.2 Objectives for the Directory Server

OE.DS_ACCESS_CNTRL	The Directory Server must provide access control mechanisms to prohibit unauthorized access to directory entries. This access control must also be enforced when importing and exporting data.
OE.DS_AUTHENT	The Directory Server must identify and authenticate users that request access to directory entries.

4.2.3 Objectives for the general Operational Environment

OE.PROTOCOL_SEC	The operational environment components that implement TLS used by the TOE and the operational environment components generating and interpreting SAML responses must implement their security protocols and cryptographic functions correctly.
OE.PWD_STRENGTH	The operational environment must provide sufficient password strength for both administrative accounts and user accounts as commensurate with the importance of the information protected by the accounts.
OE.SEC_COM	Communication of TOE external entities with the TOE as well as communication between physically distributed parts of the TOE must be secured via TLS to ensure the integrity and confidentiality of the communication.

4.3. Non-IT Objectives for the Environment

OE.INSTALL	Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner which maintains IT security objectives.
OE.PHYSICAL	Those responsible for the TOE must ensure that those parts of the TOE critical to security policy are protected from physical attack, which might compromise IT security objectives.
OE.CREDEN	Those responsible for the TOE must ensure that all access credentials, such as passwords or other authentication

	information, are protected by the users in a manner which maintains IT security objectives.
OE.OS_OPERATE	The operating system of the machines running the TOE is assumed to be configured and maintained such that it provides a reliable basis for the operation of the TOE software. The operating system is configured such that no unauthorized access to functions provided by the operating system software (including network daemons) is possible either locally or via any network connection. Any machine used to run all or a part of the TOE software is used solely for this purpose and is not used to run other application software except those required for the management and maintenance of the underlying operating system and hardware.
OE.SEC_INTEGRATE	Those responsible for the TOE must ensure that the TOE is integrated into the overall system in a way that prohibits direct access to resources to be protected by the TOE in a way that bypasses the TOE and its security functions. This includes following the guidance documentation for securely configuring the TOE and implementing a secure network policy, specifically placing the TOE components in a protected network where only WebSEAL is externally visible.
OE.AUDIT_CONFIG	The TFIM audit mechanism must only be configured during TFIM installation or when TFIM is in an offline mode of operation.

4.4. Security Objectives Rationale

4.4.1 Security Objectives Coverage

The following table provides a mapping of TOE objectives to threats and policies, showing that each objective counters or enforces at least one threat or policy, respectively.

Objective	**Threats / OSPs**
O.ACC_ADM	P.ADM_DELEGATION P.NEED_TO_KNOW
O.ACCESS_DECISION	P.AUTHORIZED_ADMIN P.AUTHORIZED_USERS
O.AUDITING	T.UAUSER T.UAACTION P.ACCOUNTABILITY
O.AUTHENT_ADMIN	P.AUTHORIZED_ADMIN T.UAUSER

O.AUTHENT_SERVER	T.UAUSER P.AUTHORIZED_SERVER
O.AUTHENT_USER	T.UAUSER P.AUTHORIZED_USERS
O.AUTHORIZATION	P.AUTHORIZED_USERS P.AUTHORIZED_SERVER P.AUTHORIZED_ADMIN
O.MANAGE	P.ADM_DELEGATION
O.PWD_STRENGTH	P.PWD_STRENGTH
O.TIME	P.ACCOUNTABILITY

Table 4-1: Mapping of security objectives to threats and policies

The following table provides a mapping of the objectives for the operational environment to assumptions, threats and policies, showing that each objective holds, counters or enforces at least one assumption, threat or policy, respectively.

Objective	**Assumptions / Threats / OSPs**
OE.OS_CONFFILE_PROT	P.AUTHORIZED_ADMIN
OE.DS_ACCESS_CNTRL	P.AUTHORIZED_USERS P.AUTHORIZED_ADMIN A.CRED_PROT A.DIR_PROT
OE.DS_AUTHENT	P.AUTHORIZED_USERS P.AUTHORIZED_ADMIN A.CRED_PROT A.DIR_PROT
OE.INSTALL	A.NOBYPASS A.ADMIN A.SINGLE_APP A.OS_CONF_MGMT A.PROTECTED_NET
OE.CREDEN	A.CRED_PROT A.ADM_PWMAN A.CLIENT_PWMAN A.CLIENT_KEYMAN
OE.PHYSICAL	A.NOBYPASS A.PHYS_PROT

OE.OS_OPERATE	A.NOBYPASS A.SINGLE_APP A.ADMIN A.OS_CONF_MGMT
OE.PROTOCOL_SEC	A.PROTOCOL_SEC
OE.PWD_STRENGTH	P.PWD_STRENGTH
OE.SEC_COM	A.SEC_COM
OE.SEC_INTEGRATE	A.NOBYPASS
OE.AUDIT_CONFIG	A.AUDIT_CONFIG

Table 4-2: Mapping of security objectives for the operational environment to assumptions, threats, and policies

4.4.2 Security Objectives Sufficiency

The following rationale provides justification that the security objectives are suitable to counter each individual threat and that each security objective tracing back to a threat, when achieved, actually contributes to the removal, diminishing or mitigation of that threat:

Threat	**Rationale for security objectives**
T.UAACTION	The threat that • an undetected violation of the TSP may be caused as a result of an attacker (possibly, but not necessarily, a person allowed to use the TOE) attempting to perform actions that the individual is not authorized to do, thus, allowing the attacker to gain unauthorized access to TSF data and/or user data is removed by: • O.AUDITING requiring the TSF to log security relevant actions.
T.UAUSER	The threat that • an attacker (possibly, but not necessarily, a person allowed to use the TOE) may impersonate an authorized user of the TOE including the threat of an authorized user that tries to impersonate as another authorized user without knowing the authentication credentials and gain unauthorized access to TSF data and/or user data is diminished by: • O.AUTHENT_ADMIN requiring authentication[1] for TAMeb administrators

[1] The TAMeb authentication process uses the external LDAP server to check the credentials.

	• O.AUTHENT_SERVER requiring authentication between paired TAMeb and TFIM servers • O.AUTHENT_USER requiring authentication[2] for TOE users • O.AUDITING implementing audit records for security relevant actions

Table 4-3: Sufficiency of objectives countering threats

The following rationale provides justification that the security objectives are suitable to cover each individual organizational security policy (OSP), that each security objective that traces back to an OSP, when achieved, actually contributes to the implementation of the OSP, and that if all security objectives that trace back to an OSP are achieved, the OSP is implemented.

OSP	**Rationale for security objectives**
P.ACCOUNTABILITY	The OSP that • the administrators of the system shall be held accountable for their actions within the system is covered by: • O.AUDITING for containing the requirement of audit records for those very actions • O.TIME for providing an accurate time source to be included in the audit records
P.ADM_DELEGATION	The OSP that • specific administration tasks as well as management operations to defined subsets of the web resources protected by the TOE may be delegated to administrators that are only allowed to perform the management tasks within their defined area of responsibility and are not able to extend this area themselves is covered by: • O.ACC_ADM which provides the means to control the (management) access to certain access control policy rules by, again, access control policy rules (in this case, the targets of an access control decision request initiated by an administrator are access control policy rules related to a certain object space in the operational environment) • O.MANAGE which requires the existence of functions to

[2] The TAMeb authentication process uses the external LDAP server to check the credentials.

	perform those management activities
P.AUTHORIZED_ADMIN	The OSP that • only administrators authorized for access to defined management resources of the TOE may access those resources after they have been successfully authenticated is covered by: • O.ACCESS_DECISION which requires an access control decision by the TOE • O.AUTHORIZATION which requires an authorization decision by the TOE • O.AUTHENT_ADMIN which requires the TAMeb decisions be based on the authentication of administrators • OE.DS_ACCESS_CNTRL which requires the Directory Server to have and use an access control policy for authorized administrators • OE.DS_AUTHENT which requires the Directory Server to have and use an authentication policy for authorized administrators • OE.OS_CONFFILE_PROT for requiring that the audit configuration file is protected against unauthorized access
P.AUTHORIZED_SERVER	The OSP that • only TAMeb servers and TFIM servers authorized for access to TSF data may access this data after they have been successfully authenticated is covered by: • O.AUTHORIZATION which requires an authorization decision by the TOE • O.AUTHENT_SERVER which requires the authentication of paired TAMeb and TFIM servers to each other
P.AUTHORIZED_USERS	The OSP that • only those users who have been authorized to access web resources protected by the TOE may access those resources after they have been successfully authenticated (unless a protected web resource is defined to be accessible by unauthenticated users, in which case no prior authentication is required) is covered by: • O.ACCESS_DECISION which requires an access control decision by the TOE

	• O.AUTHORIZATION which requires an authorization decision by the TOE • O.AUTHENT_USERS which requires the authentication of users • OE.DS_ACCESS_CNTRL which requires the Directory Server to have and use an access control policy for authorized users • OE.DS_AUTHENT which requires the Directory Server to have and use an authentication policy for authorized users
P.NEED_TO_KNOW	The OSP that • the system must allow to limit the access to, modification of, and destruction of the information in protected web resources to those authorized users which have a "need to know" for that information is covered by: • O.ACC_ADM allowing specifying which subjects may access which objects using access control policy rules
P.PWD_STRENGTH	The OSP that • passwords for both administrative accounts and user accounts should have sufficient strength as commensurate with the importance of the information protected by the accounts is covered by: • O.PWD_STRENGTH requiring the TOE to enforce a minimum password policy for accounts created and maintained by the TOE • OE.PWD_STRENGTH requiring the operational environment to provide sufficient password strength for both administrative accounts and user accounts as commensurate with the importance of the information protected by the accounts

Table 4-4: Sufficiency of objectives covering OSPs

The following rationale provide justification that the security objectives for the environment are suitable to cover each individual assumption, that each security objective for the environment that traces back to an assumption about the environment of use of the TOE, when achieved, actually contributes to the environment achieving consistency with the assumption, and that if all security objectives for the environment that trace back to an assumption are achieved, the intended usage is supported.

Assumption	Rationale for security objectives
A.ADM_PWMAN	The assumption that • administrators have to protect their passwords used for authentication to the TOE such that no unauthorized access to them is possible is covered by: • OE.CREDEN requiring that appropriate measures for the protection of access credentials are ensured by the responsible personnel
A.ADMIN	The assumption that • The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation. They will perform administration activities from a secure environment using terminals and/or workstations they trust via secured connections to the Policy Server. is covered by: • OE.INSTALL requiring competent and trustworthy administrators that deliver, install, manage and operate the TOE in a manner which maintains the operational environment security objectives • OE.OS_OPERATE which makes dedicated requirements on the operation and configuration of the underlying machines hosting the TOE application
A.CLIENT_KEYMAN	The assumption that • Users have to administer and protect private keys of their client system used for authentication and key exchange with the TOE in a secure way. This includes the secure generation of strong keys as well as the protection of private keys against any kind of unauthorized access and use. is covered by: • OE.CREDEN requiring that appropriate measures for the protection of access credentials are ensured by the responsible personnel
A.CLIENT_PWMAN	The assumption that • users have to protect their passwords used for authentication to the TOE such that no unauthorized access to them is possible is covered by: • OE.CREDEN requiring that appropriate measures for the

	protection of access credentials are ensured by the responsible personnel
A.CRED_PROT	The assumption that • the operational environment protects credentials against unauthorized access in order to prevent attackers from impersonating an authorized TOE user and to prevent attackers from getting unauthorized access to the directory information is covered by: • OE_CREDEN which requires that access to credentials is prohibited in the operational environment • OE.DS_ACCESS_CNTRL which requires access control for the Directory Server used by the TOE • OE.DS_AUTHENT which requires authentication for the Directory Server used by the TOE
A.DIR_PROT	The assumption that • The Directory Server used by the TOE provides protection mechanism against unauthorized access to TSF data stored in the directory. This includes the requirement for authentication when accessing user entries and the configuration to use TLS v1.0 as the preferred protocol to protect the communication links. is covered by: • OE.DS_AUTHENT requiring the Directory Server to perform user identification and authentication • OE.DS_ACCESS_CNTRL requiring the Directory Server to control the access to directory entries
A.NOBYPASS	The assumption that • it has to be ensured that protected resources cannot be accessed in a way that bypasses the TOE and that all internal and external access attempts to protected resources have to be channeled through the TOE is covered by: • OE.SEC_INTEGRATE which requires that the TOE is integrated into the overall system in a way that prohibits any bypass of the TOE access control functions and includes following the guidance documentation for securely configuring the TOE and implementing a secure network policy, specifically placing the TOE components in a protected network where only WebSEAL is externally visible • OE.INSTALL which requires the correct installation and secure operation of the TOE

	• OE.PHYSICAL which requires physical protection for all parts of the TOE • OE.OS_OPERATE which requires proper configuration of the underlying operating system for the machines the TOE is running on
A.OS_CONF_MGMT	The assumption that • The operating system of the machines running the TOE are assumed to be configured and maintained by trained and trustworthy personnel such that the operating system provides a reliable basis for the operation of the TOE software. Especially it is assumed that the operating system is configured such that no unauthorized access to functions provided by the operating system software (including network daemons) is possible either locally or via any network connection. is covered by: • OE.INSTALL which includes the demand for an appropriate installation and configuration of the underlying operating system • OE.OS_OPERATE which includes the demand for an appropriate maintenance of the underlying operating system
A.PHYS_PROT	The assumption that • The machines running the TOE software need to be protected against unauthorized physical access and modification. All machines running parts of the TOE software require this protection. is covered by: • OE.PHYSICAL requiring protection of those machine(s) from unauthorized physical access
A.PROTOCOL_SEC	The assumption that • The operational environment components that implement TLS used by the TOE and the operational environment components generating and interpreting SAML responses implement their security protocols and cryptographic functions correctly. is covered by: • OE.PROTOCOL_SEC which requires the operational environment components that implement TLS used by the TOE and the operational environment components generating and interpreting SAML responses to implement their security protocols and cryptographic functions correctly.
A.SEC_COM	The assumption that • Communication of TOE external entities with the TOE as well as communication between physically distributed parts of the TOE

	are secured via TLS to ensure the integrity and confidentiality of the communication. is covered by: • OE.SEC_COM which requires the communication of TOE external entities with the TOE as well as communication between physically distributed parts of the TOE must be secured via TLS to ensure the integrity and confidentiality of the communication.
A.SINGLE_APP	The assumption that • any machine used to run all or a part of the TOE software are assumed to be used solely for this purpose and are not used to run other application software except those required for the management and maintenance of the underlying operating system and hardware is covered by: • OE.INSTALL which requires the installation and maintenance of the TOE in accordance with its operational environment security objectives • OE.OS_OPERATE which demands that all machines running TOE software are solely used for this purpose
A.PROTECTED_NET	The assumption that • The TOE components reside within a protected network where WebSEAL's HTTP/HTTPS ports are the only visible external interfaces. In addition, WebSEAL's HTTP/HTTPS ports are protected by a firewall that allows only inbound HTTP/HTTPS requests via the visible external interfaces. is covered by: • OE.INSTALL which requires the installation and maintenance of the TOE in accordance with its operational environment security objectives
A.AUDIT_CONFIG	The assumption that • The TFIM audit mechanism will only be configured during TFIM installation or when TFIM is in an offline mode of operation. is covered by: • OE.AUDIT_CONFIG which requires the TFIM audit mechanism to only be configured during TFIM installation or when TFIM is in an offline mode of operation

Table 4-5: Sufficiency of objectives covering assumptions

5. Extended Components Definition

This Security Target does not extend the security components provided by the Common Criteria.

6. Security Requirements

6.1. TOE Security Functional Requirements

The Security Functional Requirement (SFR) elements in this section conform to the following formatting rules:

- Assignments – ***bold/italic***
- Refinements – ***bold/italic/underlined***
- Selections – ***bold/italic***

6.1.1 Security Audit (FAU)

FAU_GEN.1 Audit data generation

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

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

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

c) ***The following defined events:***

WebSEAL:

- ***successful and unsuccessful authentication attempts with a userid / password combination***
- ***failed authorization for access to a protected resource***
- ***user change of password***
- ***user locking***

Policy Server:

- ***creation of user by administrator***
- ***user locked by administrator***
- ***user unlocked by administrator***
- ***all commands of administrators that result in a modification of the policy database***

TFIM:

- ***Federated profiles (single sign-on)***
- ***Federated profiles (single logout)***
- ***Federated profiles (name identifier management)***
- ***Trusted service module (SSO operations supporting issuing, mapping, and validating credentials)***
- ***Message security (signing and validating signatures)***

- ***Audit provisioning.***

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, ***for auditing administrator commands: parameters passed to the command***

FAU_GEN.2 User identity association

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

FAU_SAR.1 Audit review

FAU_SAR.1.1 The TSF shall provide ***authorized administrators*** with the capability to read ***all information*** from the audit records.

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

Application Note: The TOE does not provide a direct interface to read the audit trail. Instead the administrator has to use a tool outside of the TOE to read the audit records. The responsibility of the TOE with respect to the requirements of FAU_SAR.1 are related to set the access rights to the audit files and to store the information in the audit files in a way suitable for interpretation even when read with a program like an editor.

FAU_SEL.1 Selective audit

FAU_SEL.1.1 The ***TAMeb*** TSF shall be able to select the set of events to be audited from the set of all auditable events based on the following attributes:

a) object identity, host identity
b) audit event category (authn, azn, mgmt)

Application Note: The audit categories can be defined on a per-server basis, which satisfies the selection of "host identity" above. POP can be used to define auditing on a per object basis.

FAU_STG.1 Protected audit trail storage

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

FAU_STG.1.2 The TSF shall be able to ***prevent*** unauthorised modifications to the stored audit records in the audit trail.

Application Note: The protection from unauthorized deletion is achieved with the TOE setting the access permissions to the audit files appropriately. Prevention of modifications also is based on the access rights to the audit files and by the fact that the TOE itself does not

provide any function that could be used to modify the audit records once stored in the audit file. This security functional requirement of course also relies on the appropriate protection of the TOE itself against unauthorized access in the operational environment.

FAU_STG.3 Action in case of possible audit data loss

FAU_STG.3.1 The TSF shall ***rollover to a new audit log file*** if the audit ***log file*** exceeds ***an authorized administrator set limit***.

Application Note: By specifying a positive value for the "rollover_size" parameter in the audit configuration the administrator can define that the TOE saves the current log file under a defined name that includes a timestamp and rolls over to a new audit file when the defined file size limit for the log file is reached. When a log file with the standard name already exists when the TOE is started, the TOE will append audit records to the end of the existing file until the defined file size limit is reached. This feature does not switch to a different file system if the current file system is full.

6.1.2 User data protection (FDP)

FDP_ACC.2(1) *Web-Space* complete access control

FDP_ACC.2.1 The ***TAMeb*** TSF shall enforce the ***Web-Space access control policy*** on ***users as subjects and objects in the WebSEAL protected object space*** and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2 The ***TAMeb*** TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

FDP_ACC.2(2) *Management* complete access control

FDP_ACC.2.1 The ***TAMeb*** TSF shall enforce the ***management access control policy*** on ***administrators as subjects and objects in the management protected object space*** and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2 The ***TAMeb*** TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

FDP_ACF.1(1) *Web-Space* security attribute based access control

FDP_ACF.1.1 The ***TAMeb*** TSF shall enforce the ***Web-Space access control policy*** to objects based on the following: ***subjects and objects defined in FDP_ACC.2(1) and access control lists (ACLs) and protected object policies (POPs) as object security attributes***.

FDP_ACF.1.2 The ***TAMeb*** TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: ***users have the requested type of access to a protected object in the Web-Space under the following conditions:***

Condition A:

- ***the user has been successfully authenticated and***

- ***if the object resides on another federated service provider, the TFIM Identity Provider's SAML 1.1 response is accepted by the TFIM Service Provider and mapped to an appropriate local service provider user identity and***
- ***the user has the "traverse" right for all objects from the root object down the path to the requested object and***
- ***the following rules in the specified order:***
  - ***if the user has an entry in the ACL associated with the object, then:***
    - ***if the entry contains the requested type of access, then access is granted; otherwise, access is denied***
  - ***if the user is member of one or more groups that have an entry in the ACL associated with the object:***
    - ***if any of the matching group entries contain the requested type of access, then access is granted; otherwise, access is denied***
  - ***if the ACL associated with the object has an entry of type "any-other", then:***
    - ***if the entry contains the requested type of access, then access is granted; otherwise, access is denied***
  - ***otherwise, access is denied***

Condition B:

- ***the user has not been authenticated and***
- ***the object resides in the same TAMeb organization (i. e., accessible without SSO) and***
- ***a "traverse" right exists for all objects from the root object down the path to the requested object for unauthenticated users and***
- ***the ACL associated with the object has both an entry of type "any-other" and an entry of type "unauthenticated" where the requested access right is contained in both entries, then access is granted; otherwise, access is denied.***

Application Note: The "ACL associated with the object" is the ACL of the object if the object has an explicit ACL or the ACL inherited from the next object up on the path to the root that has an explicit ACL.

FDP_ACF.1.3 The ***TAMeb*** TSF shall explicitly authorise access of subjects to objects based on the following additional rules: ***none***.

FDP_ACF.1.4 The ***TAMeb*** TSF shall explicitly deny access of subjects to objects based on the following additional rules: ***rules defined by the Protected Object Policy, if such a Protected Object Policy has been defined for the requested object. Protected Object Policies can deny access based on:***

- ***the time-of-day***

- ***the strength of the authentication mechanism***
- ***the IP address (IPv4 addresses only, IPv6 addresses are not supported)***
- ***the Quality of Protection***.

FDP_ACF.1(2) *Management* security attribute based access control

FDP_ACF.1.1 The ***TAMeb*** TSF shall enforce the ***management access control policy*** to objects based on the following: ***subjects and objects defined in FDP_ACC.2(2) and access control lists (ACLs) and protected object policies (POPs) as object security attributes***.

FDP_ACF.1.2 The ***TAMeb*** TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: ***users have the requested type of access to a protected object in the management protected object space under the following conditions:***

a)

- ***the user has been successfully authenticated and***
- ***the user has the "traverse" right for all objects from the root object down the path to the requested object and***
- ***the following rules in the specified order:***
  - ***if the user has an entry in the ACL associated with the object, then:***
    - ***if the entry contains the requested type of access, then access is granted; otherwise, access is denied***
  - ***if the user is member of a group that has an entry in the ACL associated with the object, then:***
    - ***if the entry contains the requested type of access, then access is granted; otherwise, access is denied***
  - ***the ACL associated with the object has an entry of type "any-other", then:***
    - ***if the entry contains the requested type of access, then access is granted; otherwise, access is denied***
  - ***otherwise, access is denied***

b)

- ***the user has not been authenticated and***
- ***the object resides in the same TAMeb organization (i. e., accessible without SSO) and***
- ***a "traverse" right exists for all objects from the root object down the path to the requested object for unauthenticated users and***
- ***the ACL associated with the object has both an entry of type "any-other" and an entry of type "unauthenticated" where the requested access right is***

contained in both entries, then access is granted; otherwise, access is denied.

Application Note: The "ACL associated with the object" is the ACL of the object if the object has an explicit ACL or the ACL inherited from the next object up on the path to the root that has an explicit ACL.

FDP_ACF.1.3 The ***TAMeb*** TSF shall explicitly authorise access of subjects to objects based on the following additional rules: ***none***.

FDP_ACF.1.4 The ***TAMeb*** TSF shall explicitly deny access of subjects to objects based on the following additional rules: ***rules defined by the Protected Object Policy, if such a Protected Object Policy has been defined for the requested object. Protected Object Policies can deny access based on:***

- ***the time-of-day***
- ***the strength of the authentication mechanism***
- ***the IP address (IPv4 addresses only, IPv6 addresses are not supported)***
- ***the Quality of Protection***.

6.1.3 Identification and authentication (FIA)

FIA_AFL.1 Authentication failure handling

FIA_AFL.1.1 The ***TAMeb*** TSF shall detect when ***three*** unsuccessful authentication attempts occur related to ***password based authentication attempts of users***.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been ***met or surpassed***, the ***TAMeb*** TSF shall ***disable further login attempts of that user for the time defined by the administrator in the disable-time-interval configuration parameter***.

Application Note: The number of unsuccessful authentication attempts allowed before the action defined in FIA_AFL.1.2 is taken, can be configured by the administrator in the set-max-login-failures configuration parameter. The administrator guidance defines the value of 3 for the evaluated configuration.

FIA_ATD.1 User attribute definition

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users:

TAMeb:

- ***Administrators & users:***
  - ***user name***
  - ***registry identifier (distinguished name)***
  - ***password***
  - ***list of groups to which the user belongs***

- ***TAMeb server:***
    - ***X.509v3 certificate (for server authentication)***

TFIM:

- ***TFIM server:***
    - ***X.509v3 certificate (for signing SAML responses)***

Application Note: The TAMeb user's common name and surname are not seen as security attributes.

Application Note: The TAMeb user attributes are stored outside the TSF in the external LDAP server.

FIA_SOS.1 Verification of secrets

FIA_SOS.1.1 The ***TAMeb*** TSF shall provide a mechanism to verify that secrets meet ***the following conditions:***

Minimum password length is 8 characters

Minimum number of alphabetic characters is 4

Minimum number of non-alphabetic character is 1

Maximum number of repeated characters is 2

No space character is allowed within the password.

Application Note: The conditions defined are the default settings of the parameter in the WebSEAL configuration. Those parameters are configurable by the administrator.

FIA_UAU.1 Timing of authentication

FIA_UAU.1.1 The ***TAMeb*** TSF shall allow ***access as defined in the 'any-other' entry bitwise 'and' masked with the 'unauthenticated' entry in an object's ACL*** on behalf of the user to be performed before the user is authenticated.

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

Application Note: An ACL may contain an entry that defines the access modes allowed for anonymous users, i. e. users that are not identified and authenticated. As defined above, access is granted only if both the 'any-other' and the 'unauthenticated' entry allow the mode of access and if the resource can be accessed without SSO.

Application Note: Both certificate-based and LDAP-based authentication methods are supported. Certificates are authenticated via GSKit, part of the operating environment.When an external LDAP server performs the authentication of a user on behalf of TAMeb, the LDAP result is enforced by TAMeb.

FIA_UAU.2(1) *Administrator* authentication before any action

FIA_UAU.2.1 The ***TAMeb*** TSF shall require each ***administrator*** to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that ***administrator***.

Application Note: The term "user" in the CC SFR FIA_UAU.2 has been refined by "administrator" to differentiate the authentication policy of users and administrators. While there is a possibility of unauthenticated users, all administrators (which use a different interface for authentication) are required to authenticate successfully before performing any administrative action on the TOE.

Application Note: An external LDAP server performs the authentication of an administrator on behalf of TAMeb and TAMeb enforces the result.

FIA_UAU.2(2) *TFIM Service Provider* authentication before any action

FIA_UAU.2.1 The ***TFIM Service Provider*** TSF shall require each ***SAML 1.1 response from an Identity Provider*** to be successfully authenticated before allowing any other TSF-mediated actions on behalf of ***the requesting TAMeb Service Provider***.

Application Note: There are three parties involved in this identification and authentication process. The TFIM Service Provider identifies and authenticates the SAML 1.1 response from an Identity Provider on behalf of the TAMeb Service Provider. The TAMeb Service Provider effectively delegates the identification and authentication of the SAML 1.1 response to the TFIM Service Provider. TFIM relies on the Java runtime cryptographic functionality contained in the operational environment to aid in the SAML 1.1 identification and authentication.

FIA_UAU.2(3) *TAMeb server* authentication before any action

FIA_UAU.2.1 The ***TAMeb server*** TSF shall require each ***TFIM server*** to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that ***TFIM server***.

Application Note: The TAMeb server uses the TLS certificate-based authentication to identify and authenticate the TFIM server. The TAMeb server uses the GSKit TLS which is part of the operational environment.

FIA_UAU.2(4) *TFIM server* authentication before any action

FIA_UAU.2.1 The ***TFIM server*** TSF shall require each ***TAMeb server*** to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that ***TAMeb server***.

Application Note: The TFIM server uses the TLS certificate-based authentication to identify and authenticate the TAMeb server. The TFIM server uses the Java runtime TLS which is part of the operational environment.

FIA_UAU.5 Multiple authentication mechanisms

FIA_UAU.5.1 The ***TAMeb*** TSF shall provide ***the following authentication methods for users:***

- ***Combination of user ID and password***
- ***TLS client certificate***

and the combination of user ID and password for administrators

to support user authentication.

FIA_UAU.5.2 The ***TAMeb*** TSF shall authenticate any user's claimed identity according to the ***following rules:***

administrators are authenticated by a user ID / password combination only

users are authenticated according to the following rules:

- ***The TOE first checks if the client can present a TLS client certificate. If yes, this is used for authentication of the client.***
- ***If no client certificate is presented, the server can request user ID/password authentication or a form to enter the user ID and the password is presented to the client the first time it tries to access an object in an access mode not allowed for unauthenticated users***.
- ***If the user authenticates using user ID/password but the object's POP requires client certificate authentication, TAMeb will attempt to authenticate the user using a client certificate (i.e., step-up authentication) and not allow access to the object unless client certificate authentication is successful.***

Application Note: "Client-side Certificate", "Forms Authentication" and "Basic Authentication" are the only authentication methods for users used in the TOE configuration of TAMeb. TLS client certificates are verified by GSKit, part of the operational environment.

Application Note: This function uses data stored in the external LDAP server.

Application Note: The TOE supports three certificate authentication modes of operation when configured for client-side certificate authentication: [1] Required: always request a certificate, refuse the connection if not provided; [2] Optional: always request the certificate, use it if provided, but allow an unauthenticated connection if not; and [3] Delayed: allow the unauthenticated connection and only request the certificate when required. TAMeb uses GSKit for the TLS implementation. GSKit is part of the operational environment.

FIA_UAU.6 Re-authenticating

FIA_UAU.6.1 The ***TAMeb*** TSF shall re-authenticate the user under the conditions ***the user has connected to WebSEAL and the client browser terminates the TLS session or the values for the session lifetime timeout or the inactive-timeout for the session are exceeded***.

Application Note: This condition for re-authentication applies for users only, not for administrators that have connected via the TAMeb pdadmin interface.

Application Note: This SFR does not apply to authentication methods in which the client resubmits the original authentication data. For example, it does not apply to Basic Authentication.

FIA_UID.1 Timing of identification

FIA_UID.1.1 The ***TAMeb*** TSF shall allow ***access as defined in the bitwise 'and' of the 'any-other' and 'unauthenticated' entry in an object's ACL*** on behalf of the user to be performed before the user is identified.

FIA_UID.1.2 The ***TAMeb*** TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Application Note: An ACL may contain an entry that defines the access modes allowed for anonymous users, i. e. users that are not identified and authenticated.

Application Note: Both certificate-based and LDAP-based identification methods are supported. Certificates are authenticated via GSKit, part of the operating environment When an external LDAP server performs the identification of a user on behalf of TAMeb, the LDAP result is enforced by TAMeb.

FIA_UID.2(1) *Administrator* identification before any action

FIA_UID.2.1 The ***TAMeb*** TSF shall require each ***administrator*** to be successfully identified before allowing any other TSF mediated actions on behalf of that ***administrator***.

Application Note: The term "user" in the CC SFR FIA_UID.2 has been refined by "administrator" to differentiate the authentication policy of users and administrators. While there is a possibility of unauthenticated users, all administrators (which use a different interface for authentication) are required to authenticate successfully before performing any administrative action on the TOE.

Application Note: An external LDAP server performs the identification of an administrator on behalf of TAMeb and TAMeb enforces the result.

FIA_UID.2(2) *TFIM Service Provider* identification before any action

FIA_UID.2.1 The ***TFIM Service Provider*** TSF shall require each ***SAML 1.1 response from an Identity Provider*** to be successfully identified before allowing any other TSF mediated actions on behalf of ***the requesting TAMeb Service Provider***.

Application Note: See the application note for FIA_UAU.2(2).

FIA_UID.2(3) *TAMeb server* identification before any action

FIA_UID.2.1 The ***TAMeb server*** TSF shall require each ***TFIM server*** to be successfully identified before allowing any other TSF mediated actions on behalf of that ***TFIM server***.

FIA_UID.2(4) *TFIM server* identification before any action

FIA_UID.2.1 The ***TFIM server*** TSF shall require each ***TAMeb server*** to be successfully identified before allowing any other TSF mediated actions on behalf of that ***TAMeb server***.

Application Note: See the application note for FIA_UAU.2(4).

FIA_USB.1 User-subject binding

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

- ***user ID (TAMeb and TFIM)***

- ***user role (TAMeb and TFIM)***
- ***list of groups to which the user belongs (TAMeb only)***.

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: ***none***.

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: ***none***.

6.1.4 Security management (FMT)

FMT_MOF.1 Management of security functions behaviour

FMT_MOF.1.1 The ***TAMeb*** TSF shall restrict the ability to ***modify the behaviour of*** the functions

- ***the audit function***
- ***the authentication function***
- ***the ACL access control policy***

to ***authorized administrators***.

FMT_MSA.1(1) Management of security attributes

FMT_MSA.1.1 The ***TAMeb*** TSF shall enforce the ***management access control policy*** to restrict the ability to ***modify, delete*** the security attributes ***ACL entries*** to ***users or groups having 'control' access for the ACL***.

FMT_MSA.1(2) Management of security attributes

FMT_MSA.1.1 The ***TAMeb*** TSF shall enforce the ***management access control policy*** to restrict the ability to ***change_default, modify, delete*** the security attributes ***ACL and POP*** to ***authorized administrators***.

FMT_MSA.3 Static attribute initialisation

FMT_MSA.3.1 The ***TAMeb*** TSF shall enforce the ***management access control policy and Web-Space access control policy*** to provide ***inherited*** default values for security attributes that are used to enforce the *SFP*.

FMT_MSA.3.2 The ***TAMeb*** TSF shall allow the ***administrator authorized to modify the ACL and POP of the container object*** to specify alternative initial values to override the default values when an object or information is created.

Application Note: If no ACL is attached to an object, this object inherits the ACL attached to container object that contains the object. This inheritance rule goes 'upward' in the protect object space tree until a container object with an ACL is reached. This rule is expressed with this requirement.

FMT_MTD.1(1) Management of TSF data

FMT_MTD.1.1 The ***TAMeb*** TSF shall restrict the ability to ***modify and delete*** the ***user attribute data except for user passwords*** to ***authorized administrators***.

FMT_MTD.1(2) Management of TSF data

FMT_MTD.1.1 The ***TAMeb*** TSF shall restrict the ability to ***modify*** the ***user passwords*** to ***authorized administrators and to users modifying their own passwords***.

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 The ***TAMeb*** TSF shall be capable of performing the following management functions:

- ***audit management***
- ***user and group management***
- ***ACL and POP management***.

FMT_SMR.1 Security roles

FMT_SMR.1.1 The TSF shall maintain the roles ***users, administrators, and servers***.

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

Application Note: Administration tasks can be delegated by the initially defined administrator to other administrators that he has created. The tasks a specific administrator is allowed to perform can be defined on a fine-grained basis as described in chapter 6. The term 'administrator' is used in this Security Target for any administrator that has been defined to perform administrative actions via the TAMeb pdadmin interface or via the TFIM administrative command line interface. Servers are the daemon processes that communicate with each other and authenticate themselves or their requests via SAML response using X.509v3 certificates and TLS.

6.1.5 Protection of the TSF (FPT)

FPT_STM.1 Reliable time stamps

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

6.2. Security Functional Requirements Rationale

6.2.1 Security Requirements Coverage

The following table provides a mapping of SFR to the security objectives, showing that each security functional requirement addresses at least one security objective.

Security Functional Requirements	Objective
FAU_GEN.1	O.AUDITING
FAU_GEN.2	O.AUDITING
FAU_SAR.1	O.AUDITING O.MANAGE
FAU_SEL.1	O.AUDITING O.MANAGE
FAU_STG.1	O.AUDITING
FAU_STG.3	O.AUDITING
FDP_ACC.2(1)	O.ACCESS_DECISION
FDP_ACC.2(2)	O.ACC_ADM O.ACCESS_DECISION
FDP_ACF.1(1)	O.ACCESS_DECISION
FDP_ACF.1(2)	O.ACC_ADM O.ACCESS_DECISION
FIA_AFL.1	O.AUTHENT_ADMIN
FIA_ATD.1	O.ACCESS_DECISION O.AUTHENT_ADMIN O.AUTHENT_SERVER O.AUTHORIZATION
FIA_SOS.1	O.AUTHENT_ADMIN O.AUTHENT_USER O.PWD_STRENGTH
FIA_UAU.1	O.AUTHENT_USER O.AUTHORIZATION
FIA_UAU.2(1)	O.AUTHENT_ADMIN O.AUTHORIZATION
FIA_UAU.2(2)	O.AUTHENT_SERVER O.AUTHORIZATION
FIA_UAU.2(3)	O.AUTHENT_SERVER O.AUTHORIZATION
FIA_UAU.2(4)	O.AUTHENT_SERVER O.AUTHORIZATION
FIA_UAU.5	O.AUTHENT_USER
FIA_UAU.6	O.AUTHENT_USER
FIA_UID.1	O.AUTHENT_USER O.AUTHORIZATION

Security Functional Requirements	Objective
FIA_UID.2(1)	O.AUTHENT_ADMIN O.AUTHORIZATION
FIA_UID.2(2)	O.AUTHENT_SERVER O.AUTHORIZATION
FIA_UID.2(3)	O.AUTHENT_SERVER O.AUTHORIZATION
FIA_UID.2(4)	O.AUTHENT_SERVER O.AUTHORIZATION
FIA_USB.1	O.ACCESS_DECISION O.AUTHENT_SERVER O.AUTHORIZATION
FMT_MOF.1	O.ACCESS_DECISION O.AUDITING O.AUTHENT_ADMIN O.ACC_ADM O.MANAGE
FMT_MSA.1(1)	O.ACCESS_DECISION O.ACC_ADM O.MANAGE
FMT_MSA.1(2)	O.ACCESS_DECISION O.ACC_ADM O.MANAGE
FMT_MSA.3	O.ACCESS_DECISION O.ACC_ADM O.MANAGE
FMT_MTD.1(1)	O.MANAGE
FMT_MTD.1(2)	O.MANAGE
FMT_SMF.1	O.ACCESS_DECISION O.AUDITING O.AUTHENT_ADMIN O.ACC_ADM O.MANAGE
FMT_SMR.1	O.ACCESS_DECISION
FPT_STM.1	O.AUDITING O.TIME

Table 6-1: Mapping of security functional requirements to security objectives

6.2.2 Security Requirements Sufficiency

The following arguments provide justification for each security objective for the TOE that the TOE security requirements are suitable to meet and achieve that security objective.

Security objectives	Rationale
O.ACC_ADM	This objective requires that administrators must be able to specify which objects may be accessed by which administrators or users (i.e. to manage according access control policy rules). This is implemented by requiring appropriate access control policy rules in *FDP_ACF.1(2)* and *FDP_ACC.2(2)*, which in turn allow administrators to access information – including access control policy rules – that is maintained by the TOE. *FMT_SMF.1* introduces a security function to manage the access control policy rules, which is restricted to be accessible only by administrators by *FMT_MOF.1*. *FMT_MSA.1(1)*, *FMT_MSA.1(2)* and *FMT_MSA.3* refine the management of those rules. Note: while the selection of functional requirements to achieve this objective may be (validly) considered as a sub-set of the management of access control policy rules already covered by the requirements implementing O.ACCESS_DECISION, O.ACC_ADM emphasizes on the explicit possibility to delegate the management of parts of the access control policy (i.e. only rules related to a dedicated object space) to certain administrators and, therefore, has been included as a separate aspect in this ST.
O.ACCESS_DECISION	This objective requires that access control decisions regarding access of authenticated administrators and users to objects are based on the identity of those administrators and users and made in accordance with the access control policy rules held by the TOE. *FMT_SMR.1* establishes the roles administrator and user. Access decisions are made according to two access control policies: the Web-Space access control policy as defined by *FDP_ACC.2(1)* and the management access control policy as defined by *FDP_ACC.2(2)*. The requirements *FDP_ACF.1(1)* and *FDP_ACF.1(2)* define the access control SFPs that lay out the base rules for access control decisions related to administrators who want to access objects managed by the TOE and users who want to access objects managed in the operational environment. Those rules relate to a data base of initiator security attributes, which is reflected by the requirement *FIA_ATD.1* demanding the maintenance of security attributes for users. *FIA_USB.1* guarantees that those security attributes can be associated with the subjects acting on behalf of those users. *FMT_SMF.1* introduces a security function to manage the access control policy rules, which is restricted to be accessible only by administrators by *FMT_MOF.1*. *FMT_MSA.1(1)*, *FMT_MSA.1(2)* and *FMT_MSA.3* refine the management of those rules.
O.AUDITING	This objective requires audit records for all actions performed by administrators as well as for all access control decisions made by the TOE related to administrators or users as initiators and the ability of administrators to inspect those records. *FAU_GEN.1* specifies the kind of audit events that is to be recorded, *FAU_GEN.2* ensures that those records are associated with the originating user identity (as far as

	possible). *FAU_SEL.1* allows TAMeb selective auditing to the discretion of the administrator of the TOE. *FAU_SAR.1* implement the requirements for the later analysis of audit records by authorized administrators. *FAU_STG.1* protects the audit records against modification, and *FAU_STG.3* regulates the behavior of the TSF in case the audit trail exceeds its administrator defined limit. *FMT_SMF.1* enables the management of the audit functionality and *FMT_MOF.1* restricts it to authorized administrators.
O.AUTHENT_ADMIN	This objective requires that TAMeb administrators, being initiators of access control decision requests, must be authenticated[3] by the TOE. This is implemented by *FIA_UID.2(1)* requiring identification by TAMeb and *FIA_UAU.2(1)* requiring authentication by TAMeb before any action other than authentication can be performed on behalf of an administrator. For TAMeb, passwords are stored in the initiator security attribute database for each administrator (*FIA_ATD.1*). For TAMeb, *FIA_SOS.1* imposes a "password policy" for secrets used to prove authenticity, while *FIA_AFL.1* limits the attempts of unsuccessful authentication attempts to prevent password guessing. For TAMeb, *FMT_SMF.1* enables the management of the authentication function and *FMT_MOF.1* restricts it to authorized administrators.
O.AUTHENT_SERVER	This objective requires the paired TAMeb and TFIM servers to authenticate each other to help ensure that the TSF data transferred between them is transferred to legitimate TOE components. This is implemented by FIA_UAU.2(3) and FIA_UID.2(3) which requires the TAMeb server to identify and authenticate the TFIM server, by FIA_UAU.2(4) and FIA_UID.2(4) which requires the TAMeb server to identify and authenticate the TFIM server, by FIA_ATD.1 for X.509v3 service certificates, and by FIA_USB.1 for identify and role binding.
O.AUTHENT_USER	This objective requires the ability of the TOE to authenticate[4] users trying to access protected objects. For TAMeb, this is implemented by *FIA_UID.1* requiring identification. For TAMeb, *FIA_SOS.1* defines the minimum requirements for the strength of password based authentication while *FIA_UAU.1* defines what unauthenticated users can do. For TAMeb, *FIA_UAU.5* defines the two different authentication mechanisms users may use to authenticate themselves. For TAMeb, *FIA_UAU.6* defines the situations where a re-authentication of users is required.
O.AUTHORIZATION	This objective requires that only authorized administrators, users, and TOE servers gain access to the TOE and the resources it protects. This is achieved by the authentication of TAMeb administrators as defined

[3] The TAMeb authentication process uses the external LDAP server to check the administrator credentials.

[4] The TAMeb authentication process uses the external LDAP server to check the user credentials.

	in *FIA_UAU.2(1)* and the identification of TAMeb administrators as defined in *FIA_UID.2(1)*. It includes the authentication and identification of users as defined in *FIA_UAU.1*, *FIA_UID.1* and servers as defined by *FIA_UAU(2), FIA_UAU.2(3), FIA_UID.2(2), FIA_UID.2(3), FIA_UAU.2(4),* and *FIA_UID.2(4)* and supported by *FIA_ATD.1* which defines user attributes used in the access control decisions and *FIA_USB.1* which binds subjects to users.
O.MANAGE	This objective requires the TSF to provide all required functions to support administrators to manage the TOE. The areas of management are defined in *FMT_SMF*.1. This includes management of the audit functions as defined by *FMT_MOF.1*, *FAU_SEL.1* and *FAU_SAR.1*, user and group management as defined by *FMT_MTD.1(1)* and *FMT_MTD.1(2)*, ACL and POP management as defined by *FMT_MOF.1*, *FMT_MSA.1(1), FMT_MSA.1(2)* and *FMT_MSA.3*.
O.PWD_STRENGH	The objective: • The TOE must enforce a minimum password policy for accounts created and maintained by the TOE. is met by: • FIA_SOS.1 which specifies the minimum password requirements for TAMeb.
O.TIME	This objective requires the TOE to provide reliable time stamps. This objective is met by *FPT_STM.1* which specifies that the TSF shall be able to provide reliable time stamps.

Table 6-2: Security objectives for the TOE rationale

6.2.3 Security Requirements Dependency Analysis

The following table demonstrates the dependencies of SFRs modeled in CC Part 2 and how the SFRs for the TOE resolve those dependencies:

SFR	**Dependencies**	**Fulfillment of dependencies**
FAU_GEN.1	FPT_STM.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1 FIA_UID.1	FAU_GEN.1 FIA_UID.1
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1
FAU_SEL.1	FAU_GEN.1 FMT_MTD.1	FAU_GEN.1 (see note 1 below)
FAU_STG.1	FAU_GEN.1	FAU_GEN.1
FAU_STG.3	FAU_STG.1	FAU_STG.1

SFR	Dependencies	Fulfillment of dependencies
FDP_ACC.2(1)	FDP_ACF.1	FDP_ACF.1(1)
FDP_ACC.2(2)	FDP_ACF.1	FDP_ACF.1(2)
FDP_ACF.1(1)	FDP_ACC.1 FMT_MSA.3	FDP_ACC.2(1) FMT_MSA.3
FDP_ACF.1(2)	FDP_ACC.1 FMT_MSA.3	FDP_ACC.2(2) FMT_MSA.3
FIA_AFL.1	FIA_UAU.1	FIA_UAU.1
FIA_ATD.1	no dependency	no dependency
FIA_SOS.1	no dependency	no dependency
FIA_UAU.1	FIA_UID.1	FIA_UID.1
FIA_UAU.2(1)	FIA_UID.1	FIA_UID.2(1)
FIA_UAU.2(2)	FIA_UID.1	FIA_UID.2(2)
FIA_UAU.2(3)	FIA_UID.1	FIA_UID.2(3)
FIA_UAU.2(4)	FIA_UID.1	FIA_UID.2(4)
FIA_UAU.5	no dependency	no dependency
FIA_UAU.6	no dependency	no dependency
FIA_UID.1	no dependency	no dependency
FIA_UID.2(1)	no dependency	no dependency
FIA_UID.2(2)	no dependency	no dependency
FIA_UID.2(3)	no dependency	no dependency
FIA_UID.2(4)	no dependency	no dependency
FIA_USB.1	FIA_ATD.1	FIA_ATD.1
FMT_MOF.1	FMT_SMF.1 FMT_SMR.1	FMT_SMF.1 FMT_SMR.1
FMT_MSA.1(1)	[FDP_ACC.1 or FDP_IFC.1] FMT_SMF.1 FMT_SMR.1	FDP_ACC.2(2) FMT_SMF.1 FMT_SMR.1
FMT_MSA.1(2)	[FDP_ACC.1 or FDP_IFC.1] FMT_SMF.1 FMT_SMR.1	FDP_ACC.2(2) FMT_SMF.1 FMT_SMR.1
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1 FMT_SMR.1
FMT_MTD.1(1)	FMT_SMF.1 FMT_SMR.1	FMT_SMF.1 FMT_SMR.1

SFR	Dependencies	Fulfillment of dependencies
FMT_MTD.1(2)	FMT_SMF.1 FMT_SMR.1	FMT_SMF.1 FMT_SMR.1
FMT_SMF.1	no dependency	no dependency
FMT_SMR.1	FIA_UID.1	FIA_UID.1
FPT_STM.1	no dependency	no dependency

Table 6-3: TOE SFR dependency analysis

Note 1: The dependency of FAU_SEL.1 on FMT_MTD.1 for the management of events to be audited is not fulfilled by the TOE because the TOE does not provide a management interface for this. Instead, an authorized administrator must manually edit a file to manage the audit events (OE.INSTALL).

6.3. Security Assurance Requirements

The security assurance requirements for the TOE are the Evaluation Assurance Level 4 components, augmented by ALC_FLR.3, as specified in [CC] part 3. No operations are applied to the assurance components.

Security assurance class	Security assurance components	Source
ADV: Development	ADV_ARC.1 Security architecture description	CC Part 3
	ADV_FSP.4 Complete functional specification	CC Part 3
	ADV_IMP.1 Implementation representation of the TSF	CC Part 3
	ADV_TDS.3 Basic modular design	CC Part 3
AGD: Guidance Documents	AGD_OPE.1 Operational user guidance	CC Part 3
	AGD_PRE.1 Preparative procedures	CC Part 3
ALC: Life-cycle support	ALC_CMC.4 Production support, acceptance procedures and automation	CC Part 3
	ALC_CMS.4. Problem tracking CM coverage	CC Part 3
	ALC_DEL.1 Delivery procedures	CC Part 3
	ALC_DVS.1 Identification of security measures	CC Part 3
	ALC_FLR.3 Systematic flaw remediation	CC Part 3
	ALC_LCD.1 Developer defined life-cycle model	CC Part 3
	ALC_TAT.1 Well-defined development tools	CC Part 3
ASE: Security Target evaluation	ASE_CCL.1 Conformance claims	CC Part 3
	ASE_ECD.1 Extended components definition	CC Part 3

Security assurance class	Security assurance components	Source
	ASE_INT.1 ST introduction	CC Part 3
	ASE_OBJ.2 Security objectives	CC Part 3
	ASE_REQ.2 Derived security requirements	CC Part 3
	ASE_SPD.1 Security problem definition	CC Part 3
	ASE_TSS.1 TOE summary specification	CC Part 3
ATE: Tests	ATE_COV.2 Analysis of coverage	CC Part 3
	ATE_DPT.1 Testing: basic design	CC Part 3
	ATE_FUN.1 Functional testing	CC Part 3
	ATE_IND.2 Independent testing - sample	CC Part 3
AVA: Vulnerability assessment	AVA_VAN.3 Focused vulnerability analysis	CC Part 3

Table 6-4: TOE EAL4 Security Assurance Requirements

6.4. Security Assurance Requirements Rationale

The evaluation assurance level has been chosen to match an Enhanced-Basic attack potential reflecting the expected assurance requirements of commercial customers using the TOE for the protection of data with a low or medium level of sensitivity. The TOE is intended to provide a reasonable level of protection for this data comparable to the protection provided by most commercial-off-the-shelf operating system products. In addition, the evaluation assurance level has been augmented with ALC_FLR.3 commensurate with the augmented flaw remediation capabilities offered by the developer beyond those required by the evaluation assurance level.

7. TOE Summary Specification

7.1. Statement of TOE Security Functions

7.1.1 F.Audit

The TOE (both TAMeb and TFIM) subsystems that perform auditing (as listed below) can be individually configured with respect to the audit functions they perform. In TAMeb, this is done using a defined configuration file, which defines the type of events to be collected. Only authorized TAMeb administrators can modify this file (FMT_MOF.1, FMT_SMF.1). TAMeb events can be selected based on object identity, host identity, and audit event category (FAU_SEL.1).

The TOE provides the capability to generate audit records for the following events (FAU_GEN.1):

1. WebSEAL (pdweb)
   a) Authentication attempts (successful and unsuccessful).

      Note: TLS handshake failures are not audited, but presenting a valid certificate that does not match a valid user in the directory is audited.

   b) Authorization failures
   c) locking of User ID (after three consecutive unsuccessful authentication attempts)
   d) User changing his password
2. Policy Server (pdmgrd):
   a) New user created
   b) User locked by administrator
   c) User unlocked by administrator
   d) All administrator actions that result in modifications to the policy database
3. TFIM
   a) Federated profiles (Single sign-on)
   b) Federated profiles (Single logout)
   c) Federated profiles (Name identifier management)
   d) Trusted service module (SSO operations supporting issuing, mapping, and validating credentials)
   e) Message security (signing and validating signatures)
   f) Audit provisioning

Each audit event is recorded with the date and time (FPT_STM.1), the identity of the user that caused the event (FAU_GEN.2), the type of event and the success or failure. In the case of the Policy Server also all parameters of commands issued by an administrator are audited together with the command (FAU_GEN.1).

The TOE (both TAMeb and TFIM) saves event records in a log file. It sets the permissions on the log

file such that the file is only accessible by authorized administrators. The events are stored in human readable format and require an authorized administrator to use an operational environment tool like a text editor to view the log file (FAU_SAR.1). Only authorized administrators can modify and/or delete audit records and log files (FAU_STG.1). The TOE supports a "rollover size" parameter that causes the TOE to create a new log file once the current log file reaches the "rollover size" (FAU_STG.3). Only authorized administrators can modify the "rollover size" parameter. The TOE renames the old log file to include a timestamp in the file name.

7.1.2 F.Authentication

TAMeb is capable of identifying and authenticating users as well as administrators. In the case of administrators, successful identification and authentication is required before they can perform any administrative action (FIA_UAU.2(1), FIA_UID.2(1)). Once authenticated, the administrators user ID is bound to the session (FIA_USB.1) In the case of users, defined access to defined resources may be possible for users that are not authenticated, while other resources are restricted only to authenticated users, and in some cases users that have authenticated using a particular mechanism (FIA_UAU.1, FIA_UID.1). The administrator defines which type of access to which resources is allowed for each type of user (FMT_MOF.1). The TAMeb authentication mechanism uses LDAP as the authentication database.

TAMeb offers the functionality to require a secondary, stronger level of authentication for configured resources (FIA_UAU.5). For example, if a user is already logged in and authenticated by password, the TOE offers the functionality to "step-up" the authentication of the existing session by demanding the user to additionally authenticate with a client certificate.

TFIM also identifies and authenticates users as described in the following subsections.

7.1.2.1 TAMeb User Authentication

"Users" are those entities that attempt to access TAMeb resources via a client application.

For the purposes of this document, users must perform such access using HTTPS.

The user authentication is performed by the WebSEAL systems with the support of the external LDAP server. Note: Only LDAP is supported for the access to the Directory Server in the evaluated configuration. Active Directory or other protocols are not supported. LDAP Replicas are also not supported.

Users operate on client systems. TAMeb supports a set of authentication mechanisms, not all of which are part of the TOE. The TAMeb authentication mechanisms that are part of the TOE are:

Password based authentication

In this case, the user has to provide a user ID and password to the TOE for identification and authentication (FIA_UAU.1, FIA_UID.1). The TOE supports Basic Authentication (as defined in the definition of the HTTP protocol) as well as forms based authentication where the TOE defines a form used by the client system, where the user can enter his user ID and password. Both methods require that a secured communication channel between the client system and the TOE has been set up using the TLS v1.0 protocol. This ensures that passwords are protected during their transfer from the client system to the TOE. It is the responsibility of the user at the client system to check the certificate provided by the server site to be a valid certificate of the intended access before accepting the connection and transferring the password. The implementation of TLS is part of the operational environmnent.

After having received the user ID and password from the client, the TOE will contact the Directory Server to check for a correct user ID – password combination. If the user has been successfully authenticated, the TOE will construct the user's credentials, using information from the Directory Server (including the list of groups to which the user belongs), and store them for the time of the session in a cache (since those credentials might be used in access decisions for the user) (FIA_ATD.1, FIA_USB.1).

TAMeb allows for a defined limit of successive failed login attempts when authenticating with passwords. This value is not maintained in the directory but locally on the component that performs the authentication process. The recommended value for this parameter is 3 (FIA_AFL.1).

TAMeb also has parameters an administrator can use to define the password policy (FIA_SOS.1). Those parameters are:

- The minimum length of a password (default: 8)
- The minimum number of alphabetic characters (default: 4)
- The minimum number of non-alphabetic characters (default: 1)
- The maximum number of repeated characters (default: 2)

Certificate-based Authentication

TAMeb may also authenticate users using client certificate when setting up the TLS v1.0 connection or at a later time in the case of step-up authentication. It is the responsibility of the user to ensure that this certificate is bound to a dedicated user. How this is done is left to the user.

With certificate-based authentication, TAMeb uses GSKit (part of the operational environment) to validate the client's certificate during the establishment of the TLS session, or at a later point in the case of step-up authentication. TAMeb authenticates the client by matching the Distinguished Name (DN) in the Subject field of the client-side certificate with an existing DN entry in the directory (FIA_ATD.1, FIA_USB.1).

The result of successful authentication is a user identity that is then used to build a credential for that user. It is the credential that is required for the client to participate in the TOE secure domain.

The TOE supports client-side certificate authentication allowing a user to request an authenticated identity by providing a digital certificate. Three modes of operation are supported: [1] Required: always request a certificate, refuse the connection if not provided; [2] Optional: always request the certificate, use it if provided, but allow an unauthenticated connection if not; and [3] Delayed: allow the unauthenticated connection and only request the certificate when required. The mode is specified in the WebSEAL configuration file.

7.1.2.2 TAMeb Re-authentication

When the client browser terminates the TLS session, when the session life time is exceeded, or when the user has been inactive for an administrator definable amount of time, the user must authenticate again to TAMeb in order to access resources not accessible to unauthenticated users (FIA_UAU.6).

7.1.2.3 TAMeb Administrator Authentication

TAMeb administrators are authenticated with user ID and password with the support of the external LDAP server (FIA_UAU.2(1), FIA_UID.2(1)). The TOE provides the pdadmin command line interface and the C API for administration tasks (FMT_MOF.1).

To execute a single administration command the administrator uses the following pdadmin command structure:

pdadmin [-a *admin_user*] [-p *password*] *command*

where *admin_user* is replaced by the administrator's userid and *password* is replaced by the password of the administrator. As an alternative the administrator can specify the command without the password, the system will prompt the administrator for the correct password.

To execute a set of commands the administrator can create a file containing all the commands and then issue the command

pdadmin [-a *admin_user*] [-p *password*] *filename*

where *filename* is replaced by the name of the file containing the set of commands the administrator wants to be executed. As above he may omit the password from the command line in which case he is prompted by the system for the correct password.

A third alternative is to start an interactive administrative session and using the **login** command in the form

login –a *admin_user* –p <*password*>

Again the administrator may omit the password in which case he is prompted to enter the correct password. The interactive session is terminated with the **logout** command.

Alternatively, the C API provides the functions *ivadmin_context_create** for authentication [AMCAPI].

The password policy defined in the section on user authentication also applies for the administrators.

7.1.2.4 TFIM User Authentication

When the TFIM Runtime acts as an Identity Provider, it creates and signs a SAML response (FIA_ATD.1) which is ultimately sent to a Service Provider (the TFIM cryptographic operation is performed by the operational environment). When the TFIM Runtime acts as a Service Provider, it receives a signed SAML response and validates that the response is from a trusted entity by identifying and authenticating the signature (FIA_UAU.2(2), FIA_UID.2(2), FIA_USB.1).

TFIM can construct a user identity two different ways. The first method is for TFIM to return the user ID and attributes in the EAI authentication headers. The second method is for TFIM to construct a credential and return it. Both are documented in the WebSEAL administration guidance. If the second method is used, TFIM must be configured to use the AZN API to generate a credential.

7.1.2.5 Server Authentication

Paired TAMeb and TFIM servers (i.e., runtimes) identify and mutually authenticate each other using TLS v1.0 and X.509v3 certificates (FIA_UAU.2(3), FIA_UID.2(3), FIA_UAU.2(4), and FIA_UID.2(4)). The term "paired" implies that the TAMeb and TFIM servers are within the same organization. Specifically, the TAMeb and TFIM servers on the Identity Provider's side are considered paired and the TAMeb and TFIM servers on the Service Provider's side are considered paired.

7.1.3 F.Authorization

This entire section and its subsections apply to TAMeb only.

The authorization model of TAMeb is based on Access Control Lists (ACLs) and "Protected Object Policies" (POPs). The objects that are protected (the protected object space) are defined in a tree structure that maintains three types of objects:

Web objects, which represent anything that can be addressed by an HTTP URL. This includes static web pages as well as dynamic URLs.

Tivoli Access Manager Management Objects, which represent the management objects that can be managed through the pdadmin interface.

User-defined Objects, which can be any resource a customer defines that he wants to be access control protected by the TOE. This requires that access to those objects is guarded by applications using the authorization service through the authorization API. In the case of the TOE only those user-defined objects are considered that are defined by the WebSEAL subsystem of the TOE.

Administrators can define Access Control List policies and "Protected Object Policies" that together build the set of rules the Authorization Evaluator subsystem checks to decide if a user can be given the requested type of access to an object within the protected object space (FDP_ACC.2(1), FDP_ ACC.2(2), FDP_ACF.1(1), FDP_ACF.1(2), FMT_SMF.1).

7.1.3.1 TAMeb Access Control Lists (ACL)

As mentioned before the protected object space is organized as a tree with a single root, addressed by a forward slash. The next level of hierarchy consists of the Web Objects (/WebSEAL), the Tivoli Access Manager Management Objects (/Management) and (eventually) the user-defined objects.

The leaves within the tree that defines the protected object space are actually the individual objects. All branches within the tree are called "container objects" since they represent the container for all the leaves within subtree defined by the "container object".

Within the Tivoli Access Manager Management Object space, the following categories exist in the next level of the tree (categories marked in *italics* are not used in the evaluated configuration):

- ACL policy objects (/Management/ACL)
- *Third-party authorization control objects (/Management/Action)*
- POP objects (/Management/POP)
- Server management objects (/Management/Server)
- Configuration authorization control objects (/Management/Config)
- Policy objects (/Management/Policy)

- Authorization database replication control objects (/Management/Replica)
- User management objects (/Management/Users)
- Group management objects (/Management/Groups)
- *Global Sign On (GSO) management objects (/Management/GSO)*
- Authorization rule policy management object (/Management/Rule)
- Domain management object (/Management/Domain)
- Proxy management object (/Management/Proxy)

An administrator can create a new object with the **pdadmin object** command by defining the fully qualified location within the protected object space (provided he has the required permission).

An administrator can define and modify Access Control Lists (ACL) for objects within the protected object space. An ACL consists of:

1. A Type, which can be either “user”, “group”, “any-other” or “unauthenticated”. The type identifies, if the ACL defines permissions for specific user(s), group(s), any authenticated user or unauthenticated users.
2. An ID, which defines the unique identifier for the user (if of type “user”) or group (if of type “group”). ACLs of the types “any-other” or “unauthenticated” do not have such an ID.
3. A set of permissions, that define the type of access (action) allowed with the ACL. The possible permissions are:

   a – Attach

   A – Add

   b – Browse

   B – Bypass POP

   c – Control

   d – Delete

   g – Delegation

   l – List Directory

   m – Modify

   N – Create

   R – Bypass rule

   r – Read

   s – Server Administration

   t – Trace

   T – Traverse

   v – View

   W – Password

x – Execute

For user objects, the semantics of those permissions is defined by the Resource Manager that uses the Authorization Evaluator and the TAMeb database for access decision. The semantics of those permissions within the WebSEAL TOE subsystem are defined later in this chapter.

The TOE also uses the ACLs to determine access to management objects within the protected object space. The semantics of those access modes for the different types of management objects are described in the next section. Please note that other access modes than the ones described with the individual types of management objects have no effect.

Additional permissions for user objects may be defined for third party applications, but this option is not used in the configuration of the TOE. (Note: the permissions l r x are used by third party applications only).

ACLs may be either explicit or inherited (FMT_MSA.3). Any object without an explicit ACL inherits the ACL of the container object above in the object space tree. Note that this container object may also just have an inherited ACL. The root object must always have an ACL. A default ACL for this object is set at the TOE installation and initial configuration.

The TOE uses the following rule to determine if an authenticated user has the permission for the action requested for a defined object within the protected object space

(Note: When checking for the existence of an ACL for an object, it always means checking for an explicit or inherited ACL):

1. Check that the user has the traverse permission for all container objects on the path from the root container object down to the actual object. To check this, use the steps 2 to 4 of this algorithm for all container objects on the path and the "Traverse" (T) permission. The check stops if the user does not have the necessary traverse permissions and access to the object is denied; otherwise, continue to the next step.
2. Check if an ACL entry of type "user" exists for the user and the object. If an ACL entry for the user exists, permission is granted if the requested action is defined in the ACL entry and the evaluation algorithm stops or permission is denied if the requested action is not defined in the ACL entry and the evaluation algorithm stops. If an ACL entry for the user does not exist, the ACL evaluation algorithm continues to the next step.
3. Check if ACL entries of type "group" exist for the groups the user belongs to and the object. If they exist, check if the requested permission is contained in at least one of those entries. If yes, the permission is granted and the evaluation algorithm stops. If no, the permission is denied and the evaluation algorithm stops. If none of the groups that a user belongs to matches a group in the ACL entries, the ACL evaluation algorithm continues to the next step.
4. Check if an ACL entry of type "any-other" exists for the object. If yes, check if the permission is granted within this ACL entry. If yes, permission is granted. Permission is denied, if it is not granted by the "any-other" ACL entry or if the ACL entry of type "any-other" does not exist for the object.

The TOE uses the following rule to determine if an unauthenticated user has the permission for the action requested for a defined object within the protected object space.

1. Check that the object is located in the same TAMeb organization where the request is made. If the request requires SSO access, authenticate the user.
2. For requests from unauthenticated users not requiring SSO, check that unauthenticated users have

the traverse permission for all container objects on the path from the root container object down to the actual object. To check this, use the steps 2 to 4 of this algorithm for all container objects on the path and the "Traverse" (T) permission. The check stops if the user does not have the necessary traverse permissions and access to the object is denied; otherwise, continue to the next step.

3. Check if an ACL entry of type "unauthenticated" exists for the object. If no such ACL entry exists, access is denied and the evaluation algorithm stops; otherwise, continue to the next step.
4. Check if an ACL entry of type "any-other" exists for the object. . If no such ACL entry exists, access is denied and the evaluation algorithm stops; otherwise, continue to the next step.
5. Check if the requested access is granted in both the ACL entries of type "unauthenticated" and in the ACL entries of type "any-other" for the object. Access is granted if the requested type of access is granted in both ACL entries; otherwise, access is denied.

As a result, a user has the requested access to an object if the two following conditions are satisfied:

1. The user has traverse permission for all container objects on the path from the root down to the object
2. The user has the requested permission being explicitly granted by the object's ACL, which may be an explicit ACL or an inherited ACL.

7.1.3.2 TAMeb Administration of the Object Space

As mentioned all objects (i. e. representation of objects) in the overall protected object space build a tree structure with a single root. The tree itself is structured into different "object spaces".

Objects within an object space can be created by an administrator that has the "m" (modify) permission for the object container where the object is created. Objects can be deleted by an administrator that has the "d" (delete) permission for the object container of the object.

The following access rights to objects are managed by pdmgrd, since they relate to object management activities that are not controlled by the Resource Manager (WebSEAL):

- **b (browse):** Permission to browse objects and object spaces using the following administration commands: *objectspace list*, *object list*, *object listandshow*. Note: The command *object listandshow* requires the permission "v" in addition to "b".
- **d (delete):** Permission to delete objects and object spaces using the following administration commands: *objectspace delete*, *object delete*, *object modify set name*. Note: The command *object modify set name* requires the permission "m" in addition to "d".
- **m (modify):** Permission to create and modify objects and object spaces using the following administration commands: *objectspace create*, *object create*, *object modify*. Note: The command *object modify set name* requires the permission "d" in addition to "m".
- **v (view):** Permission to show object values and attributes using the following administration commands: *object listandshow*, *object show*. Note: The command *object listandshow* requires the permission "b" in addition to "v".

7.1.3.3 TAMeb ACL Semantics for Management Objects

As mentioned above the TOE uses ACLs also to control access to its own management objects. The "container objects" (object spaces) that exist for TOE management objects have been identified in the

previous section.

ACLs for management objects can be used to define the commands an administrator is allowed to use with a defined management object. This allows for flexible delegation of specific administrative tasks to specific administrators or administrator groups.

The following semantics for permissions exist for TOE management objects:

Management/ACL Permissions

- **d (delete):** Permission to delete the ACL policy with the *acl delete* command. Requires "c" permission also be become effective.
- **m (modify):** Permission to create ACLs and modify ACL attributes using the *acl create* and *acl modify* commands. The *acl modify* command also requires the "c" permission.
- **v (view):** Permission to find, list and show ACLs using the *acl find*, *acl list* and *acl show* command

Management/POP Permissions

The object defines the permissions of administrators to manage protected object policies (POPs). Permissions are:

- **d (delete):** Permission to delete a POP using the *pop delete* command
- **m (modify):** Permission to create POPs and modify POP attributes using the *pop create* and *pop modify* commands.
- **v (view):** Permission to find and list POPs and show POP details using the *pop find*, *pop list* and *pop show* commands.
- **B (bypass POP):** Permission to override the time-of-day POP attribute on an object.

Management/Server Permissions

The object defines the permissions of administrators to perform server management tasks. Permissions are:

- **s (server):** Permission to replicate the authorization database using the *server replicate* command.
- **v (view):** Permission to list registered servers and display server properties using the *server list* and *server show* commands.
- **t (trace):** Permission to enable dynamic trace or statistics administration using the *server task server_name trace* and *server task server_name stats* command.

Management/Config Permissions

The object defines the permissions of administrators to perform configuration management tasks. Permissions are:

- **m (modify):** Permission to create and modify a Resource Manager configuration using the *svrsslcfg –config* and *svrslcfg –modify* commands.
- **d (delete):** Permission to delete (unconfigure) a Resource Manager configuration using the *svrsslcfg –unconfig* command.

Management/Policy Permissions

The object defines the permissions of administrators to perform *policy get* and *policy set* commands to define or retrieve the overall user related policy attributes (like password restrictions, etc). Permissions are:

- **v (view):** Permission to perform the *policy get* command.
- **m (modify):** Permission to perform the *policy set* command.

Management/Replica Permissions

The object defines the permission of Resource Managers to download a replica of the Master Authorization Policy database in order to create a Replica Authorization Policy database. Permissions are:

- **v (view):** Permission to read the Master Authorization Policy database

Management/Users Permissions

The object defines the permissions of administrators to manage user accounts. Permissions are:

- **d (delete):** Permission to delete a user account using the *user delete* command.
- **m (modify):** Permission to modify a user account using the *user modify* command.
- **N (create):** Permission to create a user account using the *user create* and *user import* commands.
- **v (view):** Permission to view a user account and user account details using the *user list, user list-dn, user list-gsouser, user show, user show-dn* and *user show-groups* command.
- **W (password):** Permission to reset and validate a user password using the *user modify password* and *user modify password-valid* command.

Management/Groups Permissions

The object defines the permissions of administrators to manage groups. Permissions are:

- **d (delete):** Permission to delete a group using the *group delete* command.
- **m (modify):** Permission to modify a group using the *group modify description* and *group modify remove* commands.
- **N (create):** Permission to create a group using the *group create* and *group import* commands.
- **v (view):** Permission to view a group definition using the *group list, group list-dn, user, group show, group show-dn* and *group show-members* command.
- **A (add):** Permission to a member to a group using the *group modify add* command.

Management/Rule Permissions

The object defines the permissions of administrators to manage authorization rule policies. Permissions are:

- **R (bypass rule):** Permission to override the authorization rule policy on an object.
- **d (delete):** Permission to delete an authorization rule.
- **m (modify):** Permission to create authorization rules and modify authorization rule attributes.
- **v (view):** Permission to find and list authorization rules and show authorization rule details.

Management/Domain Permissions

The object defines the permissions of administrators to manage domain tasks. Permissions are:

- **m (modify):** Permission to modify or create a domain.
- **v (view):** Permission to list and show domains.
- **d (delete):** Permission to delete a domain.

Further details on the management of the TOE are defined in the description of the function F.Management.

7.1.3.4 TAMeb Protected Object Policies (POPs)

Protected Object Policies (POPs) contain additional conditions on the request that are passed back to the Resource Manager (in the case of the TOE: WebSEAL) in the case the evaluation of the ACLs for the request was positive (i. e. according to the ACL policy request is granted). For those conditions of a POP it is the responsibility of the Resource Manager to enforce the conditions defined by the "Protected Object Policy". Like ACL attributes, POP attributes are inherited from parent objects (FMT_MSA.3).

The following attributes can be set in a "Protected Object Policy":

- **Warning Mode**. This attribute is used for debugging purpose mainly. Possible values are: "yes" and "no". If set to "yes", audit records are generated that capture the result of all ACL authorization decisions that would have been made if the warning mode would have been set to "no".
- **Audit Level**. This attribute defines the level of audit for the object. Possible values are: "permit", "deny" and "error". In the case of "permit", all requests on a protected object that result in successful access are audited. In the case of "deny", all requests on a protected object that result in denial of access are audited. In the case of "error", all internally generated error messages resulting from the denial of access to the protected object are audited.
- **Time-of-Day**. This attribute defines the day and time conditions on the access to a protected object.
- **Authentication Strength**. This attribute can be used to define restrictions on the authentication method required to gain access to the protected object. This is useful, if access to the object requires a high grade of confidence in the correct authentication of the user. It is the task of the Resource Manager (in the case of the TOE: WebSEAL) to ensure that the user has authenticated with required authentication method before granting access to the object.
- **Network-based Authentication**. This attribute allows to control access based on the IP address of the user. This can be used to prevent access to protected objects from specific IP addresses or range of IP addresses.

  Note: The evaluated configuration does not support IPv6 network-based authentication.
- **Quality of Protection**. This attribute allows to define the required level of protection for an object. Possible values are: "Privacy" and "Integrity". In the case of "Privacy", the Resource Manager has to ensure that the object is transferred over a TLS encrypted communication link. In the case of "Integrity", the Resource Manager has to ensure that a mechanism for the protection of the integrity of the object is used when transferred.
- **Re-Authentication**. Whenever this resource is accessed for the first time, an explicit

authentication is required.

- **Document Caching**. This attribute controls the caching of objects. When this attribute has the value of "no-cache" it means that the affected documents are not to be cached by the Resource Manager. The value of "public" tells the Resource Manager that this document can be cached.

Note: WebSEAL as the Resource Manager in the TOE does not support the "Integrity" attribute of a POP. Therefore, setting this attribute in the TOE has no effect.

Note: With the exception of the audit level, all attributes in the effective POP are ignored during an authorization check when the "B" (Bypass POP) in the effective ACL policy permission is set.

7.1.4 F.Management

The TOE as a whole supports the following roles: users, administrators, and servers (FMT_SMR.1).

7.1.4.1 TAMeb Administrators

At installation the TOE the group "iv-admin" is created with an initial administrator "sec_master". In addition a default ACL is defined for the "root" object in the protected object space. This default ACL for this object is:

Group iv-admin	TcmdbvaB
Any-other	T
Unauthenticated	T

which allows members of the group iv-admin (after installation only the user *sec_master* exists, which is a member of the *iv-admin* group) to create I, modify (m), delete (d), browse (b), view (v), attach (a) and define the bypass POP (B) attribute.

There are also default values for the different management object spaces, which are defined in the Base Administrator's Guide.

The mechanisms described in F.Authorization allow the initial administrator to define other administrators and/or administration groups and assign them the right to perform only specific administration tasks. This is achieved by assigning them the appropriate permissions for the individual management object spaces and objects within those object spaces as well as the appropriate permissions to individual objects or object spaces within the overall user object space.

7.1.4.2 TAMeb User and Group Management

TAMeb users are managed using an external LDAP server. An administrator with the appropriate permission in the /Management/User object space can perform user management operations like creating users, deleting users or reset a user's password. The commands to create and manage user accounts are defined in the Command Reference. The required access rights to perform the commands are defined in section 7.1.3.3 under "Management/Users" (FMT_MTD.1(1), FMT_SMF.1). In addition, users can modify their own passwords (FMT_MTD.1(2)).

Groups are managed using the TAMeb *pdadmin group* set of commands. The required access rights to perform the commands are defined in section 7.1.3.3 under "Management/Groups".

Users can be assigned to more than one group (FIA_ATD.1). Section 7.1.3.1 describes, how access rights to objects are evaluated which includes the evaluation of access rights in the case a user

belongs to more than one group.

7.1.4.3 TAMeb ACL and POP Management

TAMeb ACLs are managed using the TAMeb *pdadmin a*cl set of commands defined in the Command Reference. The required access rights to perform the commands are defined in section 7.1.3.3 under "Management/ACL" (FMT_MSA.1(1), FMT_MSA.1(2)).

TAMeb Protected Object Policies (POPs) are managed using the TAMeb *pdadmin pop* set of commands. The required access rights to perform the commands are defined in section 7.1.3.3 under "Management/POP" (FMT_MSA.1(2)).

7.1.4.4 Servers/Runtimes

The TAMeb and TFIM servers (i.e., runtimes) identify and mutually authenticate each other. As such, they are a type of role in the TOE defined as the "servers" role in FMT_SMR.1. The TAMeb server uses the TSF functionality of the TFIM Runtime to convert TAMeb user credentials into SAML responses and vice versa.

8. Abbreviations, Terminology and References

8.1. Abbreviations

Abbreviations used in this document.

ACI	Access Control Information
ACL	Access Control List
ADF	Access Control Decision Function
ADI	Access Control Decision Information
AEF	Access Control Enforcement Function
AES	Advanced Encryption Standard
API	Application Programming Interface
APP	Authorization Protection Profile
CARS	Common Auditing and Reporting Service
CAS	Common Audit Service
CC	Common Criteria, the name used historically for this multipart standard ISO/IEC 15408 in lieu of its official ISO name of "Evaluation criteria for information technology security"
DN	Distinguished Name
DRNG	Deterministic Random Number Generator
EAI	External Authentication Interface
EAL	Evaluation Assurance Level
F-SSO	Federated Single Sign-On
GSKit	Global Security Kit
HMAC	Hash-based Message Authentication Code
IHS	IBM HTTP Server
LDAP	Lightweight Directory Access Protocol
OSP	Organizational Security Policy
POP	Protected Object Policy
PP	Protection Profile
RNG	Random Number Generator
RSA	Rivest, Shamir, and Adleman (RSA) algorithm for public-key cryptography.
SAML	Security Assertion Markup Language
SF	Security Function

SFP	Security Function Policy
SFR	Security Functional Requirement
SHA (a.k.a. SHA-1)	Originally, Secure Hash Algorithm version 1, but now refers to a specific implementation of the Secure Hash Standard (SHS).
SSL	Secure Sockets Layer
SSO	Single Sign-On
ST	Security Target
TAMeb	Tivoli Access Manager for e-business
TDEA	Triple Data Encryption Algorithm
TFIM	Tivoli Federated Identity Manager
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functions
TSFI	TSF Interface
TSP	TOE Security Policy
WAS	WebSphere Application Server

8.2. Terminology

A glossary of terms used in this document.

Access Control Decision Function (ADF)	A specialized function that makes access control decisions by applying access control policy rules to an access request, ADI (of initiators, targets, access requests, or that retained from prior decision), and the context in which the access request is made.
Access Control Decision Information (ADI)	The portion (possibly all) of the ACI made available to the ADF in making a particular access control decision.
Access Control Enforcement Function (AEF)	A specialized function that is part of the access path between an initiator and a target on each access request and enforces the decision made by the ADF.
Access Control Information (ACI)	Any information used for access control purposes, including contextual information.
Access Control Policy	The set of rules that define the conditions under which an access may take place.
Access Control Policy Rules	Security policy rules concerning the provision of the access control service.
Access Request	The operations and operands that form part of an attempted access.

Assets	Information or resources to be protected by the countermeasures of a TOE.
Assignment	The specification of an identified parameter in a component.
Assurance	Grounds for confidence that an entity meets its security objectives.
Attack potential	The perceived potential for success of an attack, should an attack be launched, expressed in terms of an attacker's expertise, resources and motivation.
Augmentation	The addition of one or more assurance component(s) from Part 3 to an EAL or assurance package.
Authentication data	Information used to verify the claimed identity of a user.
Authorised user	A user who may, in accordance with the TSP, perform an operation.
Bitwise operations	Logical operations on binary data where the logical operation between two values each containing the same number of bits is applied to each aligned bit value.
Cipher suite	A group of cryptographic algorithms used to perform a function. For example, the TLS protocol's TLS_RSA_WITH_AES_128_CBC_SHA suite is a combination of RSA for key pair encryption with AES-128 for symmetric data encryption and SHA-1 for integrity protection.
Class	A grouping of families that share a common focus.
Component	The smallest selectable set of elements that may be included in a PP, an ST, or a package.
Connectivity	The property of the TOE which allows interaction with IT entities external to the TOE. This includes exchange of data by wire or by wireless means, over any distance in any environment or configuration.
Contextual Information	Information about or derived from the context in which an access request is made (e.g. time of day).
Custom chains	A chain of custom modules used by the TFIM Service Provider where each module includes one or more name-value pairs wherein a given name-value pair has a value that may be validated against a defined custom rule.
Dependency	A relationship between requirements such that the requirement that is depended upon must normally be satisfied for the other requirements to be able to meet their objectives.
Digital certificate	An electronic means of establishing an identity typically used in encrypting/decrypting data transferred between two entities and for digitally signing (digital signature) data including digital signature verification. Digital certificates contain a public key (typically the public half of an RSA key pair) and, under certain circumstances, a

	private key (the private half of an RSA key pair).
Element	An indivisible security requirement.
Evaluation	Assessment of a PP, an ST or a TOE, against defined criteria.
Evaluation Assurance Level (EAL)	A package consisting of assurance components from Part 3 that represents a point on the CC predefined assurance scale.
Evaluation authority	A body that implements the CC for a specific community by means of an evaluation scheme and thereby sets the standards and monitors the quality of evaluations conducted by bodies within that community.
Evaluation scheme	The administrative and regulatory framework under which the CC is applied by an evaluation authority within a specific community.
Extension	The addition to an ST or PP of functional requirements not contained in Part 2 and/ or assurance requirements not contained in Part 3 of the CC.
External IT entity	Any IT product or system, untrusted or trusted, outside of the TOE that interacts with the TOE.
Family	A grouping of components that share security objectives but may differ in emphasis or rigour.
Federated single sign-on chains	A chain of federated single sign-on modules used by the TFIM Service Provider where each module includes one or more name-value pairs wherein a given name-value pair has a value that may be validated against a specific federated single sign-on rule.
Formal	Expressed in a restricted syntax language with defined semantics based on well established mathematical concepts.
Human user	Any person who interacts with the TOE.
Identity	A representation (e.g. a string) uniquely identifying an authorised user, which can either be the full or abbreviated name of that user or a pseudonym.
Identity Provider	An entity that identifies and authenticates a user and provides a federated user identity for the user to be used by a Service Provider.
Informal	Expressed in natural language.
Initiator	An entity (e.g. human user or computer-based entity) that attempts to access other entities.
Internal communication channel	A communication channel between separated parts of TOE.
Internal TOE transfer	Communicating data between separated parts of the TOE.
Inter-TSF transfers	Communicating data between the TOE and the security functions of other trusted IT products.
IP address	An Internet Protocol (IP) address is a network address assigned to a network device (e.g. computer) and used to

	direct network information to a specific network device. Two major IP addressing schemes exist: IP version 4 (IPv4) addresses and IP version 6 (IPv6) addresses.
Iteration	The use of a component more than once with varying operations.
Object	A passive entity in the TOE, that contains or receives information, and upon which subjects perform operations.
Organisational security policies	One or more security rules, procedures, practices, or guidelines imposed by an organisation upon its operations.
Package	A reusable set of either functional or assurance components (e.g. an EAL), combined together to satisfy a set of identified security objectives.
Product	A package of IT software, firmware and/or hardware, providing functionality designed for use or incorporation within a multiplicity of systems.
Protection Profile (PP)	An implementation-independent set of security requirements for a category of TOEs that meet specific consumer needs.
Reference monitor	The concept of an abstract machine that enforces TOE access control policies.
Reference validation mechanism	An implementation of the reference monitor concept that possesses the following properties: it is tamperproof, always invoked, and simple enough to be subjected to thorough analysis and testing.
Refinement	The addition of details to a component.
Role	A predefined set of rules establishing the allowed interactions between a user and the TOE.
RSA key pair	The RSA algorithm requires two paired keys. Data encrypted with one key can be decrypted by the other key, but the same key used to encrypt the data cannot be used to decrypt the data.
Secret	Information that must be known only to authorised users and/or the TSF in order to enforce a specific SFP.
Security attribute	Information associated with subjects, users and/or objects that is used for the enforcement of the TSP.
Security Function (SF)	A part or parts of the TOE that have to be relied upon for enforcing a closely related subset of the rules from the TSP.
Security Function Policy (SFP)	The security policy enforced by an SF.
Security objective	A statement of intent to counter identified threats and/or satisfy identified organisation security policies and assumptions.
Security Target (ST)	A set of security requirements and specifications to be used as the basis for evaluation of an identified TOE.
Selection	The specification of one or more items from a list in a

	component.
Semiformal	Expressed in a restricted syntax language with defined semantics.
Service Provider	An entity that accepts a federated user identity from a trusted Identity Provider and then attempts to service the user's request on behalf of the user based on the user's federated user identity, typically converting the federated user identity into a local user identity within the servicing organization.
Single logout	When logging out of a single sign-on session, all other login sessions started during the single sign-on on behalf of the user are automatically logged out.
Single sign-on	The ability for a user to log into one system and for that system to automatically log the user into other user-requested applications and systems without the user supplying additional usernames and passwords during the session.
Subject	An active entity in the TOE that performs operations on objects.
System	A specific IT installation, with a particular purpose and operational environment.
Target	An entity to which access may be attempted.
Target of Evaluation (TOE)	An IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation.
TLS handshake	TLS handshake refers to the initial exchange of information and cipher suite negotiations between two TLS endpoints prior to transferring application level data.
TOE resource	Anything useable or consumable in the TOE.
TOE Security Functions (TSF)	A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.
TOE Security Functions Interface (TSFI)	A set of interfaces, whether interactive (man-machine interface) or programmatic (application programming interface), through which TOE resources are accessed, mediated by the TSF, or information is obtained from the TSF.
TOE Security Policy (TSP)	A set of rules that regulate how assets are managed, protected and distributed within a TOE.
TOE security policy model	A structured representation of the security policy to be enforced by the TOE.
Transfers outside TSF control	Communicating data to entities not under control of the TSF.
Trusted channel	A means by which a TSF and a remote trusted IT product can communicate with necessary confidence to support the TSP.

Trusted path	A means by which a user and a TSF can communicate with necessary confidence to support the TSP.
TSF data	Data created by and for the TOE, that might affect the operation of the TOE.
User	Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.
User data	Data created by and for the user, that does not affect the operation of the TSF.
Web services security chain	A chain of web services security modules used by the TFIM Service Provider where each module includes one or more name-value pairs wherein a given name-value pair has a value that may be validated against a specific web services security rule.

8.3. References

[AES]	Federal Information Processing Standards Publication 197, Advanced Encryption Standard (AES)
[AMBADM]	Tivoli Access Manager for e-business Base Installation Guide, Version 6.0.
[AMCAPI]	Tivoli Access Manager for e-business Administration C API Developer Reference, Version 6.0.
[AZNAPI]	Open Group Technical Standard: Authorization (AZN) API, The Open Group, January 2000.
[BSI-AIS20]	BSI Application Notes and Interpretation of the Scheme (AIS), AIS 20, Version 1, December 2, 1999
[CC]	Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model; Version 3.1, Revision 3, July 2009, CCMB-2009-07-001. Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components; Version 3.1, Revision 3 July 2009, CCMB-2009-07-002. Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components; Version 3.1, Revision 3, July 2009, CCMB-2009-07-003.
[CEM]	Common Methodology for Information Technology Security Evaluation, Evaluation methodology, Version 3.1, Revision 3, July 2009, CCMB-2009-07-004.
[CCGuide]	Tivoli Access Manager for e-business 6.1.1 and Federated Identity Manager 6.2.1 Common Criteria Guide, SC23-6138-01
[FIPS46-3]	FIPS PUB 46-3: DATA ENCRYPTION STANDARD (DES), October 25, 1999.

[FIPS81]	FIPS PUB 81: DES MODES OF OPERATION, Issued December 2, 1980, including CHANGE NOTICES 2 and 3.
[FIPS140-2]	FIPS PUB 140-2: SECURITY REQUIREMENTS FOR CRYPTOGRAPHIC MODULES, Issued May 25, 2001, including CHANGE NOTICES (12-03-2002)
[FIPS180-2]	FIPS PUB 180-2: Specification for the SECURE HASH STANDARD, including Change Notice to include SHA-224, August 1, 2002.
[FIPS186-2]	FIPS PUB 186-2: DIGITAL SIGNATURE STANDARD (DSS), including Change Notice, January 27, 2000.
[FIPS197]	FIPS PUB 197: Specification for the ADVANCED ENCRYPTION STANDARD (AES), November 26, 2001.
[ISO 10181-3]	ISO/IEC 10181-3: Information Technology – Open Systems Interconnection – Security frameworks for open systems: Access control framework, 1996.
[RFC2246]	The TLS Protocol, Version 1.0; T. Dierks, C. Allen, IETF RFC 2246.
[RFC2313]	PKCS #1: RSA Encryption Version 1.5, B. Kaliski, IETF RFC 2313.
[RFC3268]	Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS), P. Chown, IETF RFC 3268.
[SAML1.1]	Technical Overview of the OASIS Security Assertion Markup Language (SAML) V1.1, 11 May 2004, http://www.oasis-open.org/committees/download.php/6837/sstc-saml-tech-overview-1.1-cd.pdf
[TDEA]	NIST Special Publication 800-67 Version 1.1, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, Revised 19 May 2008
[X.509]	ITU-T RECOMMENDATION X.509 | ISO/IEC 9594-8: INFORMATION TECHNOLOGY - OPEN SYSTEMS INTERCONNECTION - THE DIRECTORY: PUBLIC-KEY AND ATTRIBUTE CERTIFICATE FRAMEWORKS.