Axway

API Gateway v7.4.1 with SP2

Security Target

January 2017

Document prepared by

Document History

Version	Date	Author	Description
0.1	August 31, 2015	John Daniels	Initial draft.
0.2	January 19, 2016	John Daniels	Updated per lab comments
0.3	February 29, 2016	John Daniels	Updated per validator comments
0.4	March 18, 2016	John Daniels	Updated per validator comments
0.5	July 8, 2016	John Daniels	Updated per test findings
0.6	August 19, 2016	John Daniels	Updated per ECR findings
0.7	September 8, 2016	John Daniels	Updated per discussions with validators
0.8	October 20, 2016	John Daniels	Updated to add CAVP certificate numbers
0.9	November 23, 2016	John Daniels	Updated per validator comments
1.0	December 2016	John Daniels	Updated for new VID #10778
1.1	January 2017	John Daniels	Updated per validator comments
1.2	January 10, 2017	John Daniels	Updated per validator comments
1.3	January 27, 2017	John Daniels	Updated per validator comments
1.4	January 31, 2017	John Daniels	Updated per validator comments

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Overview

The Axway API Gateway is an enterprise security management solution that provides management in a centralized location for access control over web services and related resources. The Axway API Gateway is a comprehensive platform for managing, delivering, and securing Web APIs. It provides integration, acceleration, governance, and security for API and SOA-based systems.

This Security Target (ST) defines the Axway API Gateway v7.4.1 Target of Evaluation (TOE) for the purposes of Common Criteria (CC) evaluation.

Whilst the Axway API Gateway offers a wide range of features, the TOE is constrained to the security features identified in section2.3:

1.2 Identification

Table 1: Evaluation identifiers

Target of Evaluation	Axway API Gateway v. 7.4.1 with SP2
Security Target	Axway API Gateway 7.4.1 with SP2 Security Target, v1.2

1.3 Conformance Claims

This ST supports the following conformance claims:

a. Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 3.1, Revision 4, CCMB-2012-09-001;

b. Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; Version 3.1, Revision 4, CCMB-2012-09-002; extended

c. Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components; Version 3.1, Revision 4, CCMB-2012-09-003 (conformant).

d. Protection profiles:

   i) Standard Protection Profile for Enterprise Security Management Policy Management v2.1, dated October 24, 2013 and

   ii) Standard Protection Profile for Enterprise Security Management Access Control v2.1, dated October 24, 2013.

e. Applicable Technical Decisions:

   i) TD 0079 1 RBG Cryptographic Transition per NIST SP 800-131A Revision 1

   ii) TD0071 Use of SHA-512 in ESM PPs

   iii) TD0066 Clarification of FAU_STG_EXT.1 Requirement in ESM PPs

   iv) TD0055 Move FTA_TAB.1 to selection-based requirement

   v) TD0042 Removal of low-level crypto failure audit in PPs

1.4 Terminology

Table 2: Terminology

Term	Definition
API	Application Programming Interface
CC	Common Criteria
EAL	Evaluation Assurance Level
ESM	Enterprise Security Management
PP	Protection Profile
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
ST	Security Target
TOE	Target of Evaluation

2 TOE Description

2.1 Type

The TOE is a comprehensive platform for managing, delivering, and securing APIs allowing for centralized enterprise security management solutions. The TOE controls how APIs and web services are exposed to and accessed by external client applications.

2.2 Usage

The TOE comprises the Axway API Gateway v7.4.1 software. The API Management architecture is as follows:

a. The API provider is the enterprise that makes the virtualized APIs for back-end applications available for API clients to consume. The API provider runs API Gateway and Policy Studio. For example, the API provider could be a credit card company that provides payment services to various customers.

b. The API clients are the end-user customer and partner organizations that consume the APIs made available by the API provider. For example, these could be specific hotel and retail organizations that enable their customers to make payments by credit card.

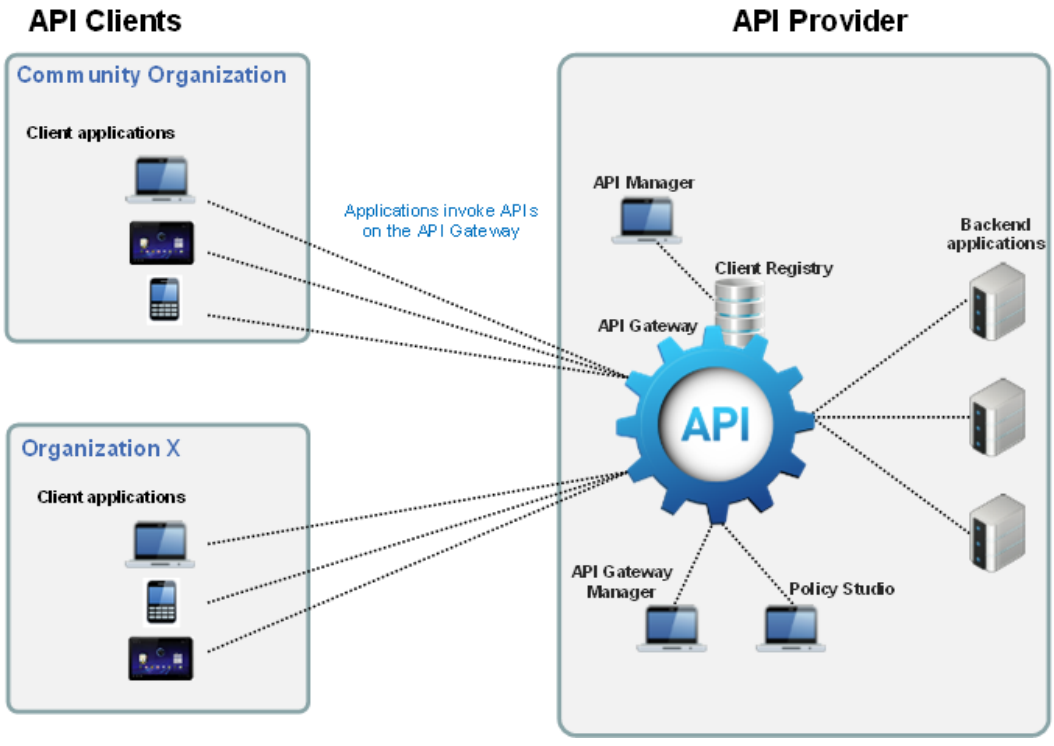

Figure 1: API Gateway Architecture

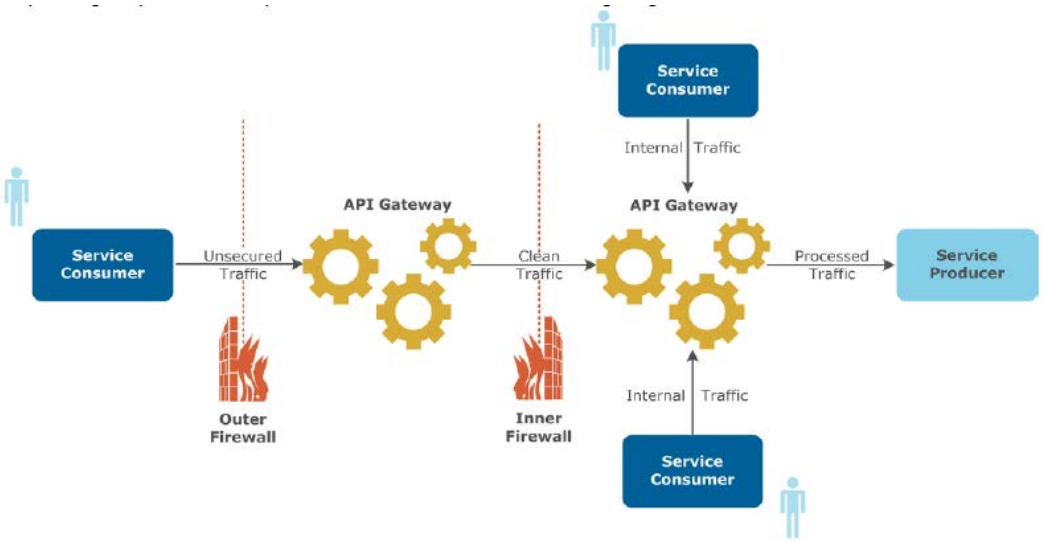

Figure 2: TOE usage scenario

Figure 2 shows the following non-TOE components:

a. Outer Firewall

b. Inner Firewall

c. Service Consumer

d. Service Producer

The TOE provides the following core functionality:

Identity Mediation. Through its support for a wide range of security standards, Axway API Gateway enables identity mediation between different identity schemes. For example, the API Gateway can authenticate external clients by username and password, but then issue SAML tokens that are used for identity propagation to application servers.

API Management. The API Gateway enables you to secure Web APIs against attack and abuse. It also enables you to govern and meter access to and usage of Web APIs. The API Gateway provides support for API management security standards such as OAuth. This enables you to share private resources with third-party websites without needing to provide credentials.

Application-level Networking. The API Gateway routes data based on sender identity, content, and type. This enables messages to be sent to the appropriate application in a secure manner. It also enables service virtualization, where services are exposed to clients with virtual addresses to mask their actual addresses for security and application delivery. In this way, the API Gateway acts as an important control point for network traffic by shielding endpoint services from direct access.

Audit Trail. The API Gateway satisfies audit requirements by enabling service transactions to be archived in a tamper-proof store for subsequent audit. Axway also facilitates privacy compliance support by allowing sensitive information, such as customer names, to be encrypted or stripped out of message traffic.

Policy Definition. Policy Studio provides a tool for developing policies that are enforced by one or more instances of API Gateway. When several instances of API Gateway are organized into groups they are managed via a Node Manager which ensures that the same policies are deployed on all the API Gateway instances in the

group, and all group members enforce the same policies and virtualize the same API and web services.

.

2.3 Physical Scope

The TOE comprises the Axway API Gateway v7.4.1 software which includes Axway API Gateway v7.4.1 core service pack (SP2). The TOE is deployed as a software component comprised of three main components for policy definition and policy consumption as follows:

a) **Policy Studio.** A GUI application that provides the user with the primary administrative interface to the Gateway. Policy Studio is used to construct policies and administer the TOE. Policy Studio pushes policies to multiple gateway instances; it submit the policies to the admin node manager which propagate new policies to all the node managers in the enterprise.

b) **API Gateway.** One or more instances of the API Gateway software that enforce policies to control web services. Basic configuration is performed using the Policy Studio to virtualize APIs and develop policies (for example, to enforce security, compliance, and operational requirements). Each Gateway instance has a corresponding node manager on the same host; it is part of the API Gateway server. One Node Manager is designated as the admin node manager. A simple TOE deployment is depicted in Figure 2.

c) **API Gateway Manager**. A web-based interface for monitoring Gateway traffic in real-time and for configuring global password policy, audit events, audit offload, and other such events

2.3.1 Guidance Documents

The TOE includes the following guidance documents:

a. API Gateway v7.4.1 Administrator's Guide

b. API Gateway v7.4.1 API Management Guide

c. API Gateway v7.4.1 Appliance User Guide

d. API Gateway v7.4.1 Concepts Guide

e. API Gateway v7.4.1 Developer Guide

f. API Gateway v7.4.1 Installation Guide

g. API Gateway v7.4.1 OAuth User Guide

h. API Gateway v7.4.1 Promotion and Deployment Guide and

i. API Gateway v7.4.1 User's Guide

j. API Gateway Security Guide

2.3.2 Non-TOE Components

The TOE operates with the following components in the environment:

a. **OpenSSL.** Cryptography of the TLS is provided by OpenSSL FIPS Object Module Version 2.0.10.

b. **DHCP Server.** The TOE can utilize a Dynamic Host Configuration Protocol (DHCP) server to acquire automatically assign an IP address.

c. **Time Server.** The TOE can utilize a Network Time Protocol (NTP) server to synchronize its system clock with a central time source.

d. **Web Browser.** The remote administrator can use a web browser to access the Web GUI interface (API Gateway Manager). See below for supported browsers.

e. **LDAP Server –** Used for external Identification and Authentication for administrators and client service users.

f. **Audit Server –** Used for external audit storage.

The **API Gateway** TOE component operates on the following operating systems:

a. Windows Server 2012 R2

b. Redhat Enterprise Linux 6.6

The **Policy Studio** TOE component operates on the same operating systems as the API Gateway and Linux and Solaris it requires also xWindows environment and GTK+2.

The **API Gateway Manager** TOE component runs on the following web browsers:

a. Internet Explorer 8, 9, 10, 11

b. Chrome 19 or higher

The installation and guidance documentation specifies any specific security settings for the web browsers.

2.4 Logical Scope

The logical scope of the TOE comprises the following security functions:

Access Control Policy Definition - This security function refers to the access control policy definition capabilities of the API Gateway. Policy Studio and API Gateway Manager are the Policy Management tools that are used to configure and define access control policies for Axway API Gateway, which is the compatible Access Control product.

Access Control Policy Enforcement - The API Gateway enforces polices defined by the Policy Studio (see section 6.1 for policy types). In the evaluated configuration, the Gateway may only consume policies created and deployed from the Axway Policy Studio.

Policy Security - Policy Studio transmits policies to the Gateway when they are explicitly deployed by the policy developer. A trusted channel (TLS) is established between Policy Studio and the Gateway to protect the transmission of policy data.

Security Audit - The TOE generates the audit events identified in Table 16. The TOE may store logs locally on the file system or remotely on an external audit server. Communication with the external audit server is secured using TLS (refer to section 6.7 for detail).

Robust Administrative Access - Access to the TOE can be achieved via the Policy Studio application and the web-based API Gateway Manager interface. Users must authenticate prior to being granted access. Users may authenticate via username and password.

Continuity of Enforcement - The Gateway continues policy enforcement in the event of a loss of connectivity with Policy Studio by enforcing the last policy received. Continuous connectivity with the Policy Studio is not expected or required.

Protected Communication: The TOE uses TLS to provide trusted channels for communication between its separate components; between itself and an external LDAP server and between itself and an external HTTP-based audit server. It provides a trusted path via HTTPS for remote administrators to access the TOE external interfaces.

2.5 Policy Filters

The core functionality of the Axway API Gateway is its ability to define and enforce policies to protect APIs and web services. To achieve this, the Axway API Gateway utilizes security policies comprising message filters where each filter processes the message in a certain way. For example, authentication filters extract user credentials from the message in order to authenticate the sender. Similarly, authorization filters use the extracted credentials to authorize the user against a number of 3rd party Identity Management servers to ensure that the user has permissions to access the requested resource.

The API Gateway also ships with a whole range of other content-based, routing, conversion and other types of filters that are not directly related to access control or security. In order to clarify the relationship between policy filters and the scope of evaluation, the following table classifies each policy filter as one of the following:

a) **Enforcing.** Filters that enforce the TOE security policy and are the focus of this evaluation.

b) **Unevaluated Functional.** Filters that facilitate product functionality and may be present in the evaluated configuration but that do not interfere with the security functions of the TOE. Such filters have not been evaluated.

c) **Unevaluated Security.** Filters that are security related but have not been evaluated.

Table 3: Policies (ESM Policy Manager PP)

Filter	Enforcing	Unevaluated Functional	Unevaluated Security
Authentication Filters			
HTTP Basic Authentication	X		
HTTP Digest Authentication			X
SSL (HTTPS Interface with mutual authentication)	X		
Attribute Authentication			X
Authenticate API Key			X
CA SOA Security Manager			X
Check Session			X
Create Session			X

Filter	Enforcing	Unevaluated Functional	Unevaluated Security
End Session			X
HTML Form-based Authentication	X		
HTTP Header	X		
IP Address	X		
Insert SAML Authentication Assertion			X
Insert Timestamp			X
Insert WS-Security Username Token			X
Kerberos Client			X
Kerberos Service			X
SAML Authentication	X		
SAML PDP Authentication			X
Security Token Service Client			X
WS-Security Username Token Authentication			X
Authorization Filters			
LDAP RBAC	X		
SAML Authorization			X
RSA Access Manager			X
Attribute Authorization			X
Axway PassPort Authorization			X
CA SOA Security Manager			X
Certificate Attributes	X		
Entrust GetAccess			X
Insert SAML Authorization Assertion			X
SAML PDP Authorization			X
Tivoli			X
XACML PEP			X
Content Filtering Filters			
Content Type		X	
Content Validation		X	
JSON Schema Validation		X	
XML Schema Validation		X	

Filter	Enforcing	Unevaluated Functional	Unevaluated Security
Message Size		X	
Threatening Content			X
Validate Timestamp		X	
XML Complexity		X	
ClamAV Anti-virus			X
McAfee Anti-virus			X
Sophos Anti-virus			X
ICAP		X	
Throttling		X	
Validate Selector Expression		X	
Validate HTTP Headers		X	
Validate Query String		X	
Validate REST Filter		X	
WS-Security Policy Layout		X	
Integrity Filters			
XML Signature Generation			X
XML Signature Verification	X		
SMIME Sign			X
SMIME Verify			X
Encryption Filters			
XML Encryption Settings			X
XML Encryption			X
XML Decryption Settings			X
XML Decryption			X
SMIME Encrypt			X
SMIME Decrypt			X
PGP Encrypt and Sign			X
PGP Decrypt and Verify			X
Generate Key			X
Certificate Filters			
CRL (Dynamic)			X

Filter	Enforcing	Unevaluated Functional	Unevaluated Security
CRL (in LDAP)			X
CRL (static)			X
CRL Responder			X
Certificate Chain			X
Certificate Validity (i.e. Expired)	X		
Create Thumbprint			X
Extract Certificate Attributes			X
Find Certificate			X
OCSP Client			X
Validate Server's Certificate Store			X
XKMS			X
Cache Filters			
Cache Attribute		X	
Create Key		X	
Is Cached?		X	
Remove Cached Attribute		X	
Monitoring Filters			
Alert		X	
Log Message Payload	X		
SLA Filter		X	
Axway Sentinel Event		X	
Axway Sentinel Link Event		X	
Attribute Filters			
Compare Attributes		X	
Extract REST Request Attributes		X	
Extract WSS Header Block		X	
Extract WSS Timestamp		X	
Extract WSS Username Token		X	
Get Cookie		X	
Insert SAML Attribute Assertion			X
JSON Path		X	

Filter	Enforcing	Unevaluated Functional	Unevaluated Security
Retrieve from Directory Server		X	
Retrieve from HTTP Header		X	
Retrieve from SAML Attribute Assertion		X	
Retrieve from SAML PDP		X	
Retrieve from Tivoli		X	
Retrieve from Message		X	
Retrieve from or Write to Database		X	
Retrieve from User Store		X	
Routing Filters			
Connect to URL		X	
Connection		X	
Dynamic Router		X	
Static Router		X	
Extract Path Parameters		X	
File Upload		X	
File Download		X	
HTTP Redirect		X	
HTTP Status Code		X	
Insert WS-Addressing		X	
Read WS-Addressing		X	
Read from JMS		X	
Rewrite URL		X	
SMTP		X	
Save to File		X	
Send to JMS		X	
TIBCO Rendezvous		X	
Utility Filters			
Time Filter	X		
Check Group Membership			X
Management Services RBAC			X
Scripting Language Filter		X	

Filter	Enforcing	Unevaluated Functional	Unevaluated Security
Copy/Modify Attributes		X	
CA SiteMinder Filters			
Authentication			X
Authorization			X
Certificate Authentication			X
Logout			X
Session Validation			X
Fault Filters			
SOAP Fault	X		
Generic Error		X	
JSON Error		X	
Oracle Access Manager Filters			
Authentication			X
Authorization			X
Log in with Certificate			X
Log out Session			X
SSO Token Validation			X
Oracle Entitlements Server Filters			
10g Authorization			X
10g Get Roles			X
11g Authorization			X
Sun Access Manager Filters			
Authentication			X
Authorization			X
Log Out Session			X
Retrieve Attributes			X
SSO Token Validation			X
X.509 Certificate Authentication			X
WS-Trust Filters			
Create WS-Trust			X
Consume WS-Trust			X

Filter	Enforcing	Unevaluated Functional	Unevaluated Security
Web Service Filters		X	
Security Services Filters			X
Resolver Filters		X	
OpenID Connect Filters			X
OAuth 2.0 Filters		X	
OAuth 2.0 Client Filters		X	
Conversion Filters		X	
Amazon Web Services Filters		X	

3 Security Problem Definition

3.1 Threats

Table 44 identifies the threats drawn from the ESM Policy Manager PP.

Table 4: Threats (ESM Policy Manager PP)

Identifier	Description
T.ADMIN_ERROR	An administrator may unintentionally install or configure the TOE incorrectly, resulting in ineffective security mechanisms.
T.CONDTRADICT	A careless administrator may create a policy that contains contradictory rules for access control enforcement.
T.EAVES	A malicious user could eavesdrop on network traffic to gain unauthorized access to TOE data.
T.FORGE	A malicious user may exploit a weak or nonexistent ability for the TOE to provide proof of its own identity in order to send forged policies to an Access Control product.
T.UNAUTH	A malicious user could bypass the TOE's identification, authentication, or authorization mechanisms in order to illicitly utilize the TOE's management functions.
T.WEAKPOL	A Policy Administrator may be incapable of using the TOE to define policies in sufficient detail to facilitate robust access control, causing an Access Control product to behave in a manner that allows illegitimate activity or prohibits legitimate activity.
T.WEAKIA	A malicious user could be illicitly authenticated by the TSF through brute-force guessing of authentication credentials.

Table 5 identifies the threats drawn from the ESM Access Control PP.

Table 5: Threats (ESM Access Control PP)

Identifier	Description
T.DISABLE	A malicious user or careless user may suspend or terminate the TOE's operation, thus making it unable to enforce its access controls upon the environment or TOE-protected data.
T.EAVES	A malicious user could eavesdrop on network traffic to gain unauthorized access to TOE data.
T.FALSIFY	A malicious user can falsify the TOE's identity, giving the Policy Management product false assurance that the TOE is enforcing a policy.
T.FORGE	A malicious user may attempt to mask their actions, causing audit data to be incorrectly recorded or never recorded.
T.MASK	A malicious user could bypass the TOE's identification, authentication, or authorization mechanisms in order to illicitly utilize the TOE's management functions.
T.NOROUTE	A malicious or careless user may cause the TOE to lose connection to the source of its enforcement policies, adversely affecting access control behaviors.

T.OFLOWS	A malicious user may attempt to provide incorrect Policy Management data to the TOE in order to alter its access control policy enforcement behavior.
T.UNAUTH	A malicious or careless user may access an object in the Operational Environment that causes disclosure of sensitive data or adversely affects the behavior of a system.

3.2 Organizational Security Policies

2) Table 6 identifies the Organizational Security Policies (OSPs) drawn from the ESM Policy Manager PP that are addressed by the TOE.

Table 6: OSPs from ESM Policy Manager PP

Identifier	Description
P.BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system.

3) Table 7 identifies the Organizational Security Policies (OSPs) drawn from the ESM Access Control PP that are addressed by the TOE.

Table 7: OSPs from ESM Access Control PP

Identifier	Description
P.UPDATEPOL	The organization will exercise due diligence to ensure that the TOE is updated with relevant policy data.

3.3 Assumptions

4) Table 8 identifies the assumptions drawn from the ESM Policy Manager PP.

Table 8: Assumptions (ESM Policy Manager PP)

Identifier	Description
A.CRYPTO	The TOE will use cryptographic primitives provided by the Operational Environment to perform cryptographic services.
A.ESM	The TOE will be able to establish connectivity to other ESM products in order to share security data
A.ROBUST	The Operational Environment will provide mechanisms to the TOE that reduce the ability for an attacker to impersonate a legitimate user during authentication.
A.SYSTIME	The TOE will receive reliable time data from the Operational Environment
A.USERID	The TOE will receive identity data from the Operational Environment.

Identifier	Description
A.MANAGE	There will be one or more competent individuals assigned to install, configure, and operate the TOE.

5) Table 9 identifies the assumptions drawn from the ESM Access Control PP.

Table 9: Assumptions (ESM Access Control PP)

Identifier	Description
A.CRYPTO	The TOE will use cryptographic primitives provided by the Operational Environment to perform cryptographic services.
A.ESM	The TOE will be able to establish connectivity to other ESM products in order to share security data
A.POLICY*	The TOE will receive policy data from the Operational Environment.
A.ROBUST	The Operational Environment will provide mechanisms to the TOE that reduce the ability for an attacker to impersonate a legitimate user during authentication
A.SYSTIME	The TOE will receive a reliable time data from the Operational Environment.
A.USERID	The TOW will receive identity data from the Operational environment.
A.INSTALL	There will be a competent and trusted administrator who will follow the guidance provided in order to install the TOE.

6) ***Note:** The assumption A.POLICY is included for PP conformance; however, it is addressed by the requirements of the ESM Policy Manager PP – see security objective O.POLICY. The Policy Manager provides policy data.

4 Security Objectives

4.1 Objectives for the Operational Environment

7) Table 10 identifies the objectives for the operational environment drawn from the ESM Policy Manager PP.

Table 10: Operational environment objectives (ESM Policy Manager PP)

Identifier	Description
OE.ADMIN	There will be one or more administrators of the Operational Environment that will be responsible for managing the TOE.
OE.CRYPTO	The Operational Environment will provide cryptographic primitives that can be used by the TOE to provide services such as ensuring the confidentiality and integrity of communications.
OE.INSTALL	Those responsible for the TOE shall ensure that the TOE is delivered, installed, managed, and operated in a secure manner.
OE.PERSON	Personnel working as TOE administrators shall be carefully selected and trained for proper operation of the TOE.
OE.PROTECT*	One or more ESM Access Control products will be deployed in the Operational Environment to protect organizational assets.
OE.ROBUST	The Operational Environment will provide mechanisms to reduce the ability for an attacker to impersonate a legitimate user during authentication.
OE.SYSTIME	The Operational Environment will provide reliable time data to the TOE.
OE.USERID	The Operational Environment shall be able to identify a user requesting access to the TOE.

8) ***Note:** OE.PROTECT is included for PP conformance; however, it is addressed by the requirements of the ESM Access Control PP. The Gateway performs the ESM Access Control functions.

9) Table 11 identifies the objectives for the operational environment drawn from the ESM Access Control PP.

Table 11: Operational environment objectives (ESM Access Control PP)

Identifier	Description
OE.CRYPTO	The Operational Environment will provide cryptographic primitives that can be used by the TOE to provide services such as ensuring the confidentiality and integrity of communications.
OE.INSTALL	Those responsible for the TOE shall ensure that the TOE is delivered, installed, managed, and operated in a secure manner.
OE.POLICY	The Operational Environment will provide a policy that the TOE will enforce.
OE.PROTECT	One or more ESM Access Control products will be deployed in the Operational Environment to protect organizational assets.

Identifier	Description
OE.SYSTIME	The operational Environment will provide reliable time data to the TOE.
OE.USERID	The Operational Environment shall be able to identify a user requesting access to the TOE.

4.2 Objectives for the TOE

10) Table 12 identifies the security objectives for the TOE drawn from the ESM Policy Manager PP.

Table 12: Security objectives (ESM Policy Manager PP)

Identifier	Description
O.ACCESSID	The TOE will contain the ability to validate the identity of other ESM products prior to distributing data to them.
O.AUDIT	The TOE will provide measures for generating and recording security relevant events that will detect access attempts to TOE-protected resources by users.
O.AUTH	The TOE will provide a mechanism to securely validate requested authentication attempts and to determine the extent to which any validated subject is able to interact with the TSF.
O.BANNER	The TOE will display an advisory warning regarding use of the TOE.
O.CONSISTENT	The TSF will provide a mechanism to identify and rectify contradictory policy data.
O.DISTRIB	The TOE will provide the ability to distribute policies to trusted IT products using secure channels.
O.INTEGRITY	The TOE will contain the ability to assert the integrity of policy data.
O.MANAGE	The TOE will provide the ability to manage the behavior of trusted IT products using secure channels.
O.POLICY	The TOE will provide the ability to generate policies that are sufficiently detailed to satisfy the Data Protection requirements for one or more technology types in the Standard Protection Profile for Enterprise Security Management Access Control.
O.PROTCOMMS	The TOE will provide protected communication channels or administrators, other parts of a distributed TOE, and authorized IT entities.
O.ROBUST	The TOE will provide mechanisms to reduce the ability for an attacker to impersonate a legitimate user during authentication.
O.SELFID	The TOE will be able to confirm its identity to the ESM deployment upon sending data to other processes within the ESM deployment.

11) Table 13 identifies the security objectives for the TOE drawn from the ESM Access Control PP.

Table 13: Security objectives (ESM Access Control PP)

Identifier	Description
O.DATAPROT	The TOE will protect data from unauthorized modification by enforcing an access control policy produced by a Policy Management product.
O.INTEGRITY	The TOE will contain the ability to verify the integrity of transferred data from Operational Environment components.
O.MAINTAIN	The TOE will be capable of maintaining access control policy enforcement if it is unable to communicate with the Policy Management product which provided it the policy.
O.MNGRID	The TOE will be able to identify and authorize a Policy Management product prior to accepting policy data from it.
O.MONITOR	The TOE will monitor the behavior of itself for anomalous activity (e.g., provide measures for generating and recording security relevant events that will detect access attempts to TOE-protected resources by users).
O.OFLOWS	The TOE will be able to recognize and discard invalid or malicious input provided by users.
O.PROTCOMMS	The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.
O.RESILIENT	If the TOE mediates actions performed by a user against resources on an operating system, the system administrator or user shall not be allowed to perform an operation in the Operational Environment that would disable or otherwise modify the behavior of the TOE.
O.SELFID	The TOE will be able to confirm its identity to the Policy Management product while sending receipt of a new policy arrival.

5 Security Requirements

5.1 Conventions

12) This document uses the following font conventions to identify the operations defined by the CC:

a. **Assignment.** Indicated with italicized text.

b. **Refinement.** Indicated with bold text and strikethroughs.

c. **Selection.** Indicated with underlined text.

d. **Assignment within a Selection:** Indicated with italicized and underlined text.

e. **Iteration.** Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3).

13) Explicitly stated SFRs are identified by having a label 'EXT' after the requirement name for TOE SFRs.

5.2 Extended Components Definition

14) Table 14 identifies the extended components which are incorporated into this ST. All extended components are reproduced directly from the ESM PM and the ESM AC protection profiles and therefore no further definition is provided in this document.

Table 14: Extended Components

Component	Title	Source
ESM_ACD.1	Access Control Policy Definition	ESM Policy Management PP
ESM_ATD.1	Object Attribute Definition	ESM Policy Management PP
ESM_ATD.2	Subject Attribute Definition	ESM Policy Management PP
ESM_EAU.2	Reliance on Enterprise Authentication	ESM Policy Management PP
ESM_EID.1	Enterprise Identification	ESM Policy Management PP
ESM_EID.2	Reliance on Enterprise Identification	ESM Access Control PP
FAU_SEL_EXT.1	External Selective Audit	ESM Policy Management PP
FAU_STG_EXT.1	External Audit Trail Storage	ESM Policy Management PP ESM Access Control PP
FCS_HTTPS_EXT.1	HTTPS	ESM Policy Management PP ESM Access Control PP
FCS_TLS_EXT.1	TLS	ESM Policy Management PP ESM Access Control PP
FMT_MOF_EXT.1	External Management of Functions Behavior	ESM Policy Management PP
FMT_MSA_EXT.5	Management of Security Attributes	ESM Policy Management PP
FPT_APW_EXT.1	Protection of Stored Credentials	ESM Policy Management PP ESM Access Control PP
FPT_FLS_EXT.1	Failure of Communications	ESM Access Control PP
FPT_SKP_EXT.1	Protection of Secret Key Parameter	ESM Policy Management PP ESM Access Control PP

5.3 Functional Requirements

Table 15: Summary of SFRs

Component	Title	Source
ESM_ACD.1	Access Control Policy Definition	ESM Policy Management PP
ESM_ACT.1	Access Control Policy Enforcement	ESM Policy Management PP
ESM_ATD.1	Object Attribute Definition	ESM Policy Management PP
ESM_ATD.2	Subject Attribute Definition	ESM Policy Management PP
ESM_EAU.2	Reliance on Enterprise Authentication	ESM Policy Management PP
ESM_EID.1	Enterprise Identification	ESM Policy Management PP
ESM_EID.2	Reliance on Enterprise Identification	ESM Access Control PP
FAU_GEN.1	Audit Data Generation	ESM Policy Management PP ESM Access Control PP
FAU_SEL.1	Selective Audit	ESM Access Control PP
FAU_SEL_EXT.1	External Selective Audit	ESM Policy Management PP
FAU_STG.1	Protected Audit Trail Storage (Local Storage)	ESM Access Control
FAU_STG_EXT.1	External Audit Trail Storage	ESM Policy Management PP ESM Access Control PP
FCO_NRR.2	Enforced proof of receipt	ESM Access Control PP
FCS_HTTPS_EXT.1	HTTPS	ESM Policy Management PP ESM Access Control PP
FCS_TLS_EXT.1	TLS	ESM Policy Management PP ESM Access Control PP
FDP_ACC.1	Access Control Policy (Host Based)	ESM Access Control PP
FDP_ACF.1	Access Control Function	ESM Access Control PP
FIA_AFL.1	Authentication Failure Handling	ESM Policy Manager PP
FIA_SOS.1	Verification of Secrets	ESM Policy Manager PP
FIA_USB.1	User-Subject Binding	ESM Policy Management PP
FMT_MOF.1	Management of Functions Behavior	ESM Policy Management PP
FMT_MOF.1(1)	Management of Functions Behavior	ESM Access Control PP
FMT_MOF.1(2)	Management of Functions Behavior	ESM Access Control PP
FMT_MOF_EXT.1	External Management of Functions Behavior	ESM Policy Management PP
FMT_MSA.1	Management of Security Attributes	ESM Access Control PP
FMT_MSA.3	Static Attribute Initialization	ESM Access Control PP
FMT_MSA_EXT.5	Management of Security Attributes	ESM Policy Management PP
FMT_SMF.1	Specification of Management Functions	ESM Access Control PP ESM Policy Management PP
FMT_SMR.1	Security Roles	ESM Policy Management PP ESM Access Control PP
FPT_APW_EXT.1	Protection of Stored Credentials	ESM Policy Management PP ESM Access Control PP
FPT_FLS_EXT.1	Failure of Communications	ESM Access Control PP
FPT_RPL.1	Replay Detection	ESM Access Control PP
FPT_SKP_EXT.1	Protection of Secret Key Parameter	ESM Policy Management PP ESM Access Control PP
FRU_FLT.1	Degraded Fault Tolerance	ESM Access Control PP
FTA_SSL.3	TSF-initiated Termination	ESM Policy Management PP
FTA_SSL.4	User Initiated Termination	ESM Policy Management PP
FTA_TAB.1	TOE Access Banner	ESM Policy Management PP

Component	Title	Source
FTP_ITC.1	Inter-TSF Trusted Channel	ESM Policy Management PP ESM Access Control PP
FTP_TRP.1	Trusted Path	ESM Policy Management PP

5.3.1 Class ESM: Enterprise Security Management (ESM)

5.3.1.1 ESM_ACD.1 Access Control Policy Definition

Hierarchical to: No other components.

ESM_ACD.1.1 The TS shall provide the ability to define access control policies for consumption by one or more compatible Access Control products.

ESM_ACD.1.2 Access control policies defined by the TSF shall be capable of containing the following:

Subjects: [*API and/or Web Service Clients (Source: external users)*]; and

Objects: [*[APIs and/or Web Services (Source: TOE published APIs and services*]; and

Operations: [*Operations exposed by the web service/API*]; and

Attributes:[*Username, and PolicyName, path.)]*

ESM_ACD.1.3 The TSF shall associate unique identifying information with each policy.

Dependencies: No dependencies

5.3.1.2 ESM_ACT.1 Access Control Policy Transmission

ESM_ACT.1.1 The TSF shall transmit policies to compatible and authorized Access Control products under the following circumstances: [immediately following creation of a new or updated policy].

5.3.1.3 ESM_ATD.1.1 Object Attribute Definition

Hierarchical to: No other components

ESM_ATD.1.1 The TSF shall maintain the following list of security attributes

belonging to individual objects: [Object**:** APIs and *Web services: Attributes: Policy ID,* filter name.

ESM_ATD.1.2 The TSF shall be able to associate security attributes with individual objects.

Dependencies: No Dependencies.

5.3.1.4 ESM_ATD.2 Subject attribute definition

Hierarchical to: No other components.

ESM_ATD.2.1 The TSF shall maintain the following list of security attributes belonging to individual subjects:

[Subject: *API and Web Service Clients*

Attributes: Authentication credentials, Username/Group, Role]

ESM_ATD.2.2 The TSF shall be able to associate security attributes with individual subjects.

Dependencies: No dependencies

5.3.1.5 ESM_EAU.2 Reliance on Enterprise Authentication

Hierarchical to: No other components.

ESM_EAU.2.1 The TSF shall rely on [*external LDAP service and API Gateway*] for subject authentication.

ESM_EAU.2.2 The TSF shall require each subject to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that subject.

Dependencies: ESM_EID.2 Reliance on Enterprise Identification

5.3.1.6 ESM_EID.2 Reliance on Enterprise Identification

Hierarchical to: No other components.

ESM_EID.2.1 The TSF shall rely on [*external LDAP service and API Gateway] for subject identification.*

ESM_EID.2.2 The TSF shall require each subject to be successfully identified before allowing any other TSF-mediated actions on behalf of that subject.

Dependencies: No dependencies.

5.3.2 Class: Security Audit (FAU)

5.3.2.1 FAU_GEN.1 Audit Data Generation

Hierarchical to: No other components

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

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

b) All auditable events identified in table 16 for the [not specified] level of audit; and

c) *All administrative actions*;

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, [*information specified in column three of Table 16*].

Dependencies: FPT_STM.1 Reliable Time Stamps

Table 16 – Auditable Events

Component	Event	Additional Information
ESM_ACD.1	Creation or modification of policy	Unique policy identifier
ESM_ACT.1	Transmission of policy to Access Control products	Destination of policy
ESM_ATD.1	Definition of object attributes	Identification of the attribute defined
ESM_ATD.2	Association of attributes with objects	Identification of the object and the attribute
ESM_EAU.2	All use of the authentication mechanism	None
FAU_SEL.1	All modifications to audit configuration	None
FAU_SEL_EXT.1	All modifications to audit configuration	None
FAU_STG_EXT.1	Establishment and disestablishment of communications with audit server	Identification of audit server
FCO_NRR.2	The invocation of the non-repudiation service	Identification of the information, the destination, and a copy of the evidence provided
FCS_HTTPS_EXT.1	Failure to establish a session, establishment/termination of a session	Non-TOE endpoint of connection (IP address), reason for failure (if applicable)
FCS_TLS_EXT.1	Failure to establish a session, establishment/termination of a session	Non-TOE endpoint of connection (IP address), reason for failure (if applicable)
FDP_ACC.1	Any changes to the enforced policy or policies	Identification of Policy Management product making the change
FDP_ACF.1	All requests to perform an operation on an object covered by the SFP	Subject identity, object identity, requested operation
FIA_AFL.1	The reaching of an unsuccessful authentication attempt threshold, the actions taken when the threshold is reached, and any actions taken to restore the normal state.	Action taken when threshold is reached
FIA_SOS.1	Rejection or acceptance by the TSF of any tested secret	None
FIA_SOS.1	Identification of any changes to the defined quality metrics	The change made to the quality metric
FIA_USB.1	Successful and unsuccessful binding of user attributes to a subject	None
FMT_MOF.1	All modifications to TSF behavior	None
FMT_SMF.1	Use of the management functions	Management function performed
FMT_SMR.1	Modifications to the members of the management roles	None
FPT_FLS_EXT.1	Failure of communication between the TOE and Policy Management product	Identity of the Policy Management product, reason for the failure
FPT_RPL.1	Detection of replay	Action to be taken based on the specific actions
FTP_ITC.1	All use of trusted channel functions	Identity of the initiator and target of the trusted channel

Component	Event	Additional Information
FTP_TRP.1	All attempted uses of the trusted path functions	Identification of user associated with all trusted path functions, if available

5.3.2.2 FAU_SEL.1 Selective Audit (Access Control PP)

Hierarchical to: No other components.

FAU_SEL.1.1 The 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. [event type]; and

b. [*no additional attributes*].

Dependencies: FAU_GEN.1 Audit Data Generation

FMT_MTD.1 Management of TSF Data

5.3.2.3 FAU_SEL_EXT.1 External audit trail storage (Policy Manager PP)

Hierarchical to: No other components.

FAU_SEL_EXT.1.1 The TSF shall be able to select the set of events to be audited by an [an ESM Access Control product] from the set of all auditable events based on the following attributes:

a) event type; and

b) no additional attributes.

Dependencies: FAU_GEN.1 Audit data generation

FMT_MTD.1 Management of TSF data

Application Note: The ESM Access Control product in the SFR refers to API Gateway component.

5.3.2.4 FAU_STG.1 Protected Audit Trail Storage (Local Storage)

Hierarchical to: No other components.

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

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

Dependencies: FAU_GEN.1 Audit data generation

5.3.2.5 FAU_STG_EXT.1 External audit trail storage

Hierarchical to: No other components

FAU_STG_EXT.1.1 The TSF shall be able to transmit the generated audit data to *a HTTP-based audit server and TOE-internal storage*.

FAU_STG_EXT.1.2 The TSF shall ensure that transmission of generated audit data to any external IT entity uses a trusted channel defined in FTP_ITC.1.

FAU_STG_EXT.1.3 The TSF shall ensure that any TOE-internal storage of generated audit data:

a) protects the stored audit records in the TOE-internal audit trail from unauthorised deletion; and

b) prevents unauthorised modifications to the stored audit records in the TOE-internal audit trail.

Dependencies: FAU_GEN.1 Audit data generation FTP_ITC.1 Inter-TSF Trusted Channel.

5.3.3 Class: Communication (FCO)

5.3.3.1 FCO_NRR.2 Enforced proof of receipt

Hierarchical to: FCO_NRR.1 Selective proof of receipt

FCO_NRR.2.1 The TSF shall enforce the generation of evidence of receipt for received [policies] at all times.

FCO_NRR.2.2 The TSF shall be able to relate the [*stored authentication credentials*] of the recipient of the information, and the [*policy ID*] of the information to which the evidence applies.

FCO_NRR.2.3 The TSF shall provide a capability to verify the evidence of receipt of information to [originator] given [*within 30 seconds*].

Dependencies: FIA_UID.1 Timing of identification

5.3.4 Class: Cryptographic Support (FCS)

5.3.4.1 FCS_HTTPS_EXT.1 HTTPS

Hierarchical to: No other components.

FCS_HTTPS_EXT.1.1 The TSF shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS as specified in FCS_TLS_EXT.1.

Dependencies: FCS_TLS_EXT.1 TLS

Application Note: The TOE utilizes cryptographic services from the operational environment and therefore does not claim any of the optional cryptographic SFRs per guidance at Annex C.8 of the ESM Policy Manager PP and Annex C.5 of the Access Control PP.

5.3.4.2 FCS_TLS_EXT.1 TLS

Hierarchical to: No other components.

FCS_TLS_EXT.1.1 The TSF shall implement one or more of the following protocols [TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)] supporting the following ciphersuites:

Mandatory Ciphersuites:

TLS_RSA_WITH_AES_128_CBC_SHA

Optional Ciphersuites:

[

TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA256

TLS_RSA_WITH_AES_256_CBC_ SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384]

5.3.5 Class: User Data Protection (FDP)

5.3.5.1 FDP_ACC.1 Access Control Policy (Web Based Access Control)

Hierarchical to: No other components.

FDP_ACC.1.1 The TSF shall enforce the [access control Security Function Policy (SFP)] on [

- Subjects: API and Web Service clients; and
- Objects: APIs, Web Services, data stores, and
- Operations: all permitted operations on the API/Web Service.

Dependencies: FDP_ACF.1 Security Attribute Based Access Control

5.3.5.2 FDP_ACF.1 Access Control Functions (Web Based Access Control)

Hierarchical to: No other components.

FDP_ACF.1.1 The TSF shall enforce the [access control SFP] to objects based on the following: [all operations between users and objects based upon the attributes defined in Table 17 below].

Table 17 FDP Requirement Table for Web-Based Access Control

Subject	Object	Operation
User	URLs	Access via HTTP operations
	Files	Open | Download
	CGI Scripts	Execute
		Enable | Disable
	Forms	HTTP GET | HTTP POST

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [rules received from an authorized and compatible Policy Management product].

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [*no additional rules*].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [*no additional rules*].

Dependencies: FDP_ACC.1 Subset Access Control

FMT_MSA.3 Static Attribute Initialization

5.3.6 Class: Identification and Authentication (FIA)

5.3.6.1 FIA_AFL.1 Authentication Failure Handling

Hierarchical to: No other components

FIA_AFL.1.1 The TSF shall detect when [*6*] unsuccessful authentication attempts occur related to [*login to Policy Studio and API Gateway Manager*].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been [surpassed], the TSF shall [*lock user account for 30 minutes*].

5.3.6.2 FIA_SOS.1 Verification of Secrets

Hierarchical to: No other components.

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

a) For environmental password-based authentication, the following rules apply:

1. Passwords shall be able to be composed of a subset of the following character sets:* [*printable ASCII character set*] *that include the following values* [*All printable ASCII characters, including 26 uppercase letters, 26 lowercase letters, 10 numbers, and 32 special characters "~", "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", "-", "_", "=", "+", "\", "|", "[", "{", "}", "]", ";", ":", "'", "", ",", "<", ".", ">", "/", "?"*]*; and

2. Minimum password length shall settable by an administrator, and support passwords of 16 characters or greater; and

3. Password composition rules specifying the types and numbers of required characters that comprise the password shall be settable by an administrator; and

4. Passwords shall have a maximum lifetime, configurable by an administrator; and

5. New passwords shall contain a minimum of an administrator-specified number of character changes from the previous password; and

6. Passwords shall not be reused within the last administrator-settable number of passwords used by that user;

b) For non-password-based authentication, the following rules apply:

The probability that a secret can be obtained by an attacker during the lifetime of the secret is less than 2-20.

Dependencies: No dependencies.

5.3.6.3 FIA_USB.1 User-Subject Binding

Hierarchical to: No other components.

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

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: [*Username in the credentials is looked up against an internal file to determine the roles that have been assigned to that user*.].

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: [*Rules take effect immediately*].

Dependencies: FIA_ATD.1 User Attribute Definition

5.3.7 Class: Security Management (FMT)

5.3.7.1 FMT_MOF.1 (1) Management of Functions Behavior (Access Control PP)

Hierarchical to: No other components.

FMT_MOF.1.1 (1) The TSF shall restrict the ability to [determine the behavior of, modify the behavior of] the functions[: audited events, repository for trusted audit storage, access control SFP, policy being implemented by the TSF, access control SFP behavior to enforce in the event of communications outage, [*assignment: No other functions*]] to [an authorized and compatible Policy Management product].

Dependencies: FMT_SMF.1 Specification of Management Functions

FMT_SMR.1 Security Roles

5.3.7.2 FMT_MOF.1 (2) Management of Functions Behavior (Policy Manager PP)

FMT_MOF.1.1 (2) The TSF shall restrict the ability to [determine the behavior of] the functions: [*manage admin users, view real-time traffic data, view logs*] to [an authorized and compatible Enterprise Security Management product].

Dependencies: FMT_SMF.1 Specification of Management Functions

FMT_SMR.1 Security Roles

5.3.7.3 FMT_MOF_EXT.1 External Management of Functions Behavior

Hierarchical to: No other components.

FMT_MOF_EXT.1.1 The TSF shall restrict the ability to query the behavior of, modify the functions of Access Control products: audited events, repository for audit storage, Access Control SFP, policy version being implemented, Access Control SFP behavior to enforce in the event of communications outage, [*query the behavior of, modify the functions of the API Gateway Access Control product: audited events, repository for audit storage, Access Control SFP, policy version being implemented, addition of users and assigning roles to those users*] to [*Policy Developer can*

download, edit, deploy, version, and tag a configuration; an API Gateway administrator has read/write access to API Gateway Manager; an API Gateway Operator has read-only access to the API Gateway Manager].

Dependencies: FMT_SMF.1 Specification of Management Functions

FMT_SMR.1 Security Roles

5.3.7.4 FMT_MSA.1 Management of Security

Hierarchical to: No other components.

FMT_MSA.1.1 The TSF shall enforce the [access control SFP] to restrict the ability to [change_default, query, modify, delete, [*no other operations*]] the security attributes [access control policies, access control policy attributes, implementation status of access control policies] to [an authorized and compatible Policy Management product].

Dependencies: FDP_ACC.1 Subset Access Control

FMT_SMF.1 Specification of Management Functions

FMT_SMR.1 Security Roles

5.3.7.5 FMT_MSA.3 Static Attribute Initialization

Hierarchical to: No other components.

FMT_MSA.3.1 The TSF shall enforce the [access control SFP] to provide [restrictive] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [authorized and compatible Policy Management product] to specify alternative initial values to override the default values when an object or information is created.

Dependencies: FMT_MSA.1 Management of Security Attributes

FMT_SMR.1 Security Roles

5.3.7.6 FMT_MSA_EXT.5 Consistent Security Attributes

Hierarchical to: No other components.

FMT_MSA_EXT.5.1 The TSF shall [identify the following internal inconsistencies within a policy prior to distribution: [*circular dependencies, policies with no "Start" filter*].

FMT_MSA_EXT.5.2 The TSF shall take the following action when an inconsistency is detected: [issue a prompt for an administrator to manually resolve the inconsistency, *block a policy package deployment until the issue has been resolved*.]].

Dependencies: FMT_MOF_EXT.1 External Management of Functions Behavior

5.3.7.7 FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [configuration of audited events,

configuration of repository for trusted audit storage, configuration of Access Control SFP, querying of policy being implemented by the TSF, management of Access Control SFP behavior to enforce in the event of communications outage,[manage users]].

Dependencies: No dependencies.

5.3.7.8 FMT_SMR.1 Security Roles

Hierarchical to: No other components.

FMT_SMR.1.1 The TSF shall maintain the roles [*Policy Developer, API Gateway Administrator, and API Gateway Operator*].

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

Dependencies: FIA_UID.1 Timing of Identification

5.3.8 Class: Protection of the TSF (FPT)

5.3.8.1 FPT_APW_EXT.1 Protection of Stored Credentials

Hierarchical to: No other components.

FPT_APW_EXT.1.1 The TSF shall store credentials in non-plaintext form.

FPT_APW_EXT.1.2 The TSF shall prevent the reading of plaintext credentials.

Dependencies: No dependencies.

5.3.8.2 FPT_FLS_EXT.1 Failure of Communication

Hierarchical to: No other components.

FPT_FLS_EXT.1.1 The TSF shall maintain policy enforcement in the following manner when the communication between the TSF and the Policy Management product encounters a failure state: [enforce the last policy received, [*failure policy*]].

Dependencies: No dependencies

5.3.8.3 FPT_RPL.1 Replay Detection

Hierarchical to: No other components.

FPT_RPL.1.1 The TSF shall detect replay for the following entities: [*TSF data*].

FPT_RPL.1.2 The TSF shall perform [r*eject the data*] when replay is detected.

Dependencies: No dependencies.

5.3.8.4 FPT_SKP_EXT.1 Protection of Secret Key Parameters

Hierarchical to: No other components.

FPT_SKP_EXT.1.1 The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

Dependencies: No dependencies.

5.3.9 Class Resource Utilization (FRU)

5.3.9.1 FRU_FLT.1 Degraded Fault Tolerance

Hierarchical to: No other components.

FRU_FLT.1.1 The TSF shall ensure the operation of [enforcing the most recent policy] when the following failures occur: [restoration of communications with the Policy Management product after an outage].

Dependencies: FPT_FLS.1 Failure with Preservation of Secure State

5.3.10 Class TOE Access (FTA)

5.3.10.1 FTA_SSL.3 TSF-initiated Termination

Hierarchical to: No other components.

FTA_SSL.3.1 Refinement: The TSF shall terminate a remote interactive session after an [Authorized Administrator-configurable time interval of session inactivity].

Dependencies: No dependencies.

5.3.10.2 FTA_SSL.4 User-initiated Termination

Hierarchical to: No other components.

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

Dependencies: No dependencies.

5.3.10.3 FTA_TAB.1 TOE Access Banner

Hierarchical to: No other components.

FTA_TAB.1.1 Refinement: Before establishing a user session, the TSF shall display a configurable advisory warning message regarding unauthorized use of the TOE.

Dependencies: No dependencies

5.3.11 Class Trusted Paths/Channels

5.3.11.1 FTP_ITC.1 Inter-TSF Trusted Channel

Hierarchical to: No other components.

FTP_ITC.1.1 Refinement: The TSF shall use [TLS/HTTPS] to provide a trusted communication channel between itself and authorized IT entities that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification and disclosure.

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

FTP_ITC.1.3 Refinement: The TSF shall initiate communication via the trusted channel for transfer of policy data, *[Policy Studio initiates*

communication via the trusted channel for the transfer of policy data. API Gateway Manager initiates communication via a trusted channel to manage users, view audit/domain logs, view system performance metrics, and manage server instances (i.e. start/stop/etc)].

Dependencies: No dependencies.

5.3.11.2 FTP_TRP.1 Trusted Path

Hierarchical to: No other components.

FTP_TRP.1.1 Refinement: The TSF shall use [TLS/HTTPS] to provide a trusted communication path between itself and [remote] users that is logically distinct from other communication channels and provides assured identification of its end points and protection of the communicated data from [modification, disclosure].

FTP_TRP.1.2 The TSF shall permit [remote users] to initiate communication via the trusted path.

FTP_TRP.1.3 Refinement: The TSF shall require the use of the trusted path for initial user authentication, execution of management functions.

5.4 Assurance Requirements

The TOE security assurance requirements, summarized in Table 18, are drawn from the Standard Protection Profile for Enterprise Security Management Policy Management, October 24, 2013, v.2.1 and the Standard Protection Profile for Enterprise Security Management Access Control, October 24, 2013, v.2.1.

Table 18: Assurance Requirements

Assurance Class	Components	Description
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative User Guidance
Tests	ATE_IND.1	Independent Testing - Conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability Survey
Life Cycle Support	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM Coverage

6 TOE Summary Specification

6.1 Access Control Policy Definition

Related SFRs:	ESM_ACD.1, ESM_ATD.1, ESM_ATD.2, ESM_ACT.1, FMT_MSA.1, FMT_MSA.3, FMT_MSA_EXT.5, FMT_SMF.1

This security function refers to the access control policy definition capabilities of the API Gateway. Policy Studio and API Gateway Manager are the Policy Management tools that are used to configure and define access control policies for Axway API Gateway, which is the compatible Access Control product. A summary of the policy definition capability is provided below, however an entire manual – *Axway API Gateway Policy Developer's Guide* – is dedicated to this topic and should be referenced for detailed information.

A policy defines restrictions for the consumption of a published Gateway-protected service. Policies are identified using a Policy Name and Policy ID. At the highest layer of abstraction, the attributes used in policy definition are as defined in ESM_ACD.1. Details for included policy filters are provided in sections 6.1.1, 6.1.2 and 6.1.3 below. Policies are transmitted to the API Gateway immediately after they are created or when a service client is newly registered with the API Gateway.

In Policy Studio, a service policy includes message filters that determine the authentication method, identity credentials, transport method, and routing method for the protected API or web service. The specific types of filters, their relative location, and the other filters determine the properties and validity of a policy. During processing, the Gateway executes each message filter sequentially according to its position in the policy, assigning a 'success', 'failure', or 'abort' outcome to each.

API Gateway Manager is a web-based interface for configuring global password policy, administrator users and their corresponding roles, audit events, audit offload and other global configuration. Furthermore, it can be used to provide a real-time, graphical view of API Gateway transactions. It can also be used to view various audit logs and trace files in order to diagnose run-time problems.

6.1.1 Access Control

The following subsets of filters are evaluated:

a. **HTTP Basic Authentication.** The API Gateway authenticates clients using HTTP basic authentication against an LDAP directory. This filter is used in conjunction with a HTTPS Interface to ensure that the client username and password are always passed over a TLS 1.2 encrypted channel.

   Direct authentication is supported, where the client submits the "Authorization" header on the first request. Furthermore, a challenge-response mechanism is supported, where the client does not submit the "Authorization" header in the first request, which forces the API Gateway to return a "challenge" to the client in the form of a HTTP 401 response code. The client must then submit the "Authorization" header on the subsequent request.

   Retries are supported in cases where the user enters incorrect (or no) credentials in the browser. The browser allows the user to "retry" their credentials for a number of times, which is configurable in the browser.

   If the user supplies invalid credentials more than 6 times in a 5 minute time period, the user will be locked out for 30 minutes.

   The format of the credentials can be configured to be one of the following, depending on what is configured in the LDAP directory:

i) **Username:** applicable

ii) **X.509 Distinguished Name:** Not applicable

There is an option to remove the “Authorization” header in a post processing step, but is not relevant to access control decisions.

For more information on the HTTP Basic Authentication filter, please refer to the *HTTP basic authentication* section of the *Policy Developer Guide.*

b. **HTML Form-based Authentication.** User credentials are passed to the API Gateway in a HTML form and authenticated using HTML form-based authentication against an LDAP directory. The filter is used along with a HTTPS Interface that enforces the use of TLS 1.2 to secure the client credentials.

It is possible to configure the form fields used to contain the username and password.

The format of the credentials can be configured to be one of the following, depending on what is configured in the LDAP directory:

i) **Username:** applicable

ii) **X.509 Distinguished Name:** Not applicable

As with HTTP Basic authentication, if the user submits an incorrect username and/or password more than 6 times in a 5 minute interval, that user will be locked out for 30 minutes.

With form-based authentication, the following attributes are validated on the user’s session:

Session Expiry: By default the session expires after 60 seconds.

Secure Flag: The filter can be configured to restrict the use of the session to secure channels only.

HTTP Only: Sets the HttpOnly attribute on the cookie to restrict access to the cookie from client-side script.

In order for the session expiry flag to work, the Session Check filter must be included in the policy. The Secure Flag check also has a dependency on the Compare Attributes Filter

c. **HTTP Header Authentication:** This filter is used in cases where the API Gateway receives end-user authentication credentials in an HTTP header. When the API Gateway receives the message, it authenticates the sender of the message and extracts the end-user identity from the token in the HTTP header for use in subsequent authorization filters. This filter has the following configurable fields:

Name: appropriate name of the filter

HTTP Header Name: The name of the Http header that contains the end-user credentials.

HTTP Header type: the type of credentials that are passed in the named HTTP header. The following are supported: X.509 Distinguished name; certificate; username.

d. **Mutual TLS 1.2 Authentication.** A HTTPS Interface is configured to require clients to present their certificates during the TLS 1.2 handshake. The CA

cert that issued the client certificate must be explicitly trusted by the HTTPS Interface.

The HTTPS Interface is configured to *require* client certificates and to trust a certificate chain on the client certificate of up to 2 certificates.

The HTTPS Interface supports the cipher suites enabled by the following OpenSSL cipher string and blocks the SSLv2 and SSLv3 protocols:

A check is performed to ensure that the server SSL certificate's Common Name resolves to a network address. SSL Server Name Identifier (SNI) is not applicable for the evaluated configuration.

e. **LDAP Attribute Authorization.** This filter enables authorization of an authenticated client for a backend service based on user roles stored in an LDAP directory. User attributes are read from the selected LDAP directory, and compared against some known values.

The filter can be configured to succeed if only 1 comparison succeeds or if all comparisons succeed. There are various types of matching rules that can be used in the comparison:

i) *Applicable:* contains, doesn't contain, is and is not

ii) *Not Applicable:* matches regular expression, doesn't match regular expression, ends with, is, is not, starts with

The advanced settings on this filter include the ability to cache retrieved attributes for use by successive filters and how to process multi-valued attributes. However these settings are not relevant to the access control decision that the filter enforces.

f. **SAML Authentication.** The API Gateway extracts a SAML 2.0 authentication assertion from a WS-Security block in the SOAP Header. Once the assertion has been extracted, the following validation is performed:

i) Ensure that the assertion is using the SAML 2.0 namespace.

ii) Check the Created and Expires assertions to ensure that the assertion is still valid, taking into account the configured drift time to allow for discrepancies between the machine on which the assertion was generated and the Gateway's machine.

iii) Make sure that the Issuer of the assertion matches one of the Trusted Issuers configured in the filter.

For more information on this filter, please refer to the *SAML Authentication* section of the *Policy Developer Guide*.

g. **XML Signature Verification.** XML Signature Verification is used to verify the integrity of an XML Signature embedded within a WS-Security block with a specified SOAP actor/role.

The following types of XML Signatures are supported:

- Asymmetric
- Symmetric
- Enveloped
- Enveloping

The public key to use to validate the signature can be extracted from the KeyInfo section of the XML Signature using one of the following key referencing mechanisms:

i) *Not applicable:* Public key is embedded in the message.

ii) *Not applicable:* Public key included in a certificate that is contained within an attachment

iii) *Applicable:* Security Token Reference

For the purposes of this evaluation, the key will be referenced using one of the following applicable Security Token Reference methods:

i) *Applicable:* X509v3, EncryptedKey, EncryptedKeySHA1, Issuer DName and Serial Number, ThumbprintSHA1, X509SubjectKeyIndentifier

ii) *Not Applicable: GSS_Kerberosv5_AP_REQ, GSS_Kerberosv5APREQSHA1, Key Identifier with x509v3, PKCS7, SAMLAssertionID, SAMLID, SecurityContextToken, X509PKIPathv1, X509v1*

The nodes that must be signed by the XML Signature can be configured using the pre-configured *Node Locations* options (e.g. SOAP 1.2 Body).

The XML Signature must be signed with algorithms that comply with the WS-SecurityPolicy AlgorithmSuite of *Basic256*.

The filter will block messages that do not contain an XML Signature.

For more information on this filter, please refer to the *XML Signature Verification* section of the *Policy Developer Guide.*

h. **IP Address Authentication.** The API Gateway restricts access based on the client's IP address. Filtering can be done based on a specific IP address or on a range of IP addresses. Please refer to the *IP Address Authentication* section of the *Policy Developer Guide* for more information on this filter.

i. **Certificate Validity Filter:** This filter performs a simple check on a certificate to ensure that the validitiy period of an X.509 certificate has not expired. By default this filter searches for the X.509 certificate in the certificate message attribute which must be set by a predecessor filter in the policy (SSL Authentication filter). Configuration fields for this filter includes the 'Certificate Selector Expression' field which specifies where to obtain the certificate (for example, form a message attribute). The filter checks the validity of the specified certificate. If no certificate is found, the filter returns an error.

j. **Log Message Payload:** Logs the request and/or response message payload to the audit trail.

k. **SOAP Fault.** The SOAP Fault filter is used to return a SOAP 1.1 or 1.2 Fault to the client when an error occurs.

l. **Certificate Attributes Authorization:** This filter is used to authorize access to a web service based on the X.509 attributes of an authenticated client's certificate. This filter checks the attribute values in the Dname of the client to which the certificate belongs and succeed only if all the attribute values are matched to configured attribute values.

m. **Time Filter.** This filter enforces time-based access control to APIs and Web Services. The filter can be configured to either block or allow requests during a configurable time period. The time period is defined using one of the following options:

- User-defined period using configurable “from” and “to” times
- Days of the week
- Cron Expression

For more information on these configuration options, refer to the *Allow or block messages at certain times* section of the *Policy Developer Guide.*

6.1.2 Policy Logic

Policies implicitly support logical OR and AND operations in the way that they are composited. A succession of filters placed on *success paths* must all succeed in order for the policy to pass, while a succession of filters placed on *failure paths* can test for multiple conditions (i.e. logical OR) so that if any of them succeed, a common *success path* can be followed.

Please refer to the *Policy Development Guide* for explanations on the logic of sample policies.

Policy Studio implicitly guards against building faulty logic in policies. A warning is displayed if a policy developer attempts to build a circular dependency in the policy invocation path. A warning is also displayed if a policy developer attempts to deploy a policy with no start filter.

6.2 Access Control Policy Enforcement

Related SFRs:	FDP_ACC.1, FDP_ACF.1, FMT_MOF.1(1), FMT_MOF.1(2), FMT_MOF_EXT.1, FMT_MSA.1, FMT_MSA.3, FMT_MSA_EXT.5, FMT_SMF.1, FMT_SMR.1, ESM_EID.2

The Gateway enforces polices defined by the Policy Studio (see section 6.1 for policy types). In the evaluated configuration, the Gateway may only consume policies created and deployed from the Axway Policy Studio. The Gateway authenticates users logging in from Policy Studio using HTTP basic authentication over TLS (refer to section 6.7 for TLS details). Administrators logging in from the web-based API Gateway Manager are authenticated using Form-based authentication over TLS.

The Gateway performs the following message processing for a typical policy:

a) Service request arrives.

b) Request is resolved to a specific policy based on the incoming path

c) Request is run through the policy filters in order. Typically authentication filters are run first followed by authorization filters and then content-based filters

d) If all filters in the policy execute successfully, the request will be routed on to the protected API or service.

e) If the policy has been configured to process the response message, the response message filters are executed at this point.

f) If all of the response filters execute successfully, the response from the API or service is sent back to the client. If an error occurs at any stage, an appropriate SOAP (or other) fault can be returned to the client.

Policies comprise 1 or more message filters connected together using success and failure paths to form a *logical circuit.* If a filter executes successfully, the next filter on the *success path* from will be executed. If the filter fails, the next filter on the *failure path* will be invoked. If the filter aborts due to a condition that prevents the filter from running, the *Fault Handler* for that policy will be invoked.

By combining filters, success paths, failure paths, and fault handlers using simple drag-and-drop methods, it is possible to build up extremely powerful and flexible policies. When all filters on the success path execute successfully, the service requestor receives an appropriate response message. However, if a filter fails (and it has no success path filters configured), the service requestor receives an error message.

The TOE enforces a default policy of denying all access to protected services. When a new service is created an associated policy must be defined for the service with a relative path configured for the policy. If no policy is mapped to a service, the default policy is invoked for all message request to access the service.

Initial Gateway topology configuration is performed using the Gateway's *managedomain* script, described in Chapter 2 of the Axway API Gateway Administrator Guide. Subsequent to initial setup, configuration in performed by Policy Studio and API Gateway Manager.

The TOE restricts the ability to manage security attributes in accordance section 5.3.7. Policy values are restrictive by default – access to objects is denied unless the administrator defines a policy to enable access.

A user may terminate their interactive session at Policy Studio and API Gateway Manager using the logout functionality.

The following roles can be assigned to administrator users created in API Gateway Manager:

Role	Tool	Privileges
API Server Administrator	API Gateway Manager	Read/Write access to API Gateway Manager
API Server Operator	API Gateway Manager	Read-only access to API Gateway Manager
Policy Developer	Policy Studio	Download, edit, deploy, version, and tag a configuration

.

6.3 Policy Security

Related SFRs:	ESM_ATD.1, ESM_EAU.2, FCO_NRR.2, FIA_SOS.1, FPT_RPL.1, FTP_ITC.1, FTP_TRP.1

Policy Studio transmits policies to the Gateway when they are explicitly deployed by the policy developer. The policy developer admin selects the gateway servers to deploy on by selecting the gateway server name on the deployment screen. A trusted channel (TLS) is established between Policy Studio and the Gateway to protect the transmission of policy data. TLS provides replay detection and will reject the replayed packets and generate an audit event when detection occurs.

Access to Policy Studio and API Gateway Manager requires user identification and authentication (username & password) as described in section 6.5.

Policy Studio is a thick client Java application executed on a general purpose operating system. Remote access to Policy Studio is not supported. The API Gateway Manager is a web-based interface than can be accessed from the browser.

The TOE uses OpenSSL in the operational environment for TLS. Refer to section 6.7 for TLS details.

When a policy developer admin deploys a new policy to the admin node manager, the policy deployment windows shows that deployment is in process and when the deployment complete successfully. In addition, the API Gateway server generates an audit record when a policy is received from Policy Studio, providing proof of receipt. The API Gateway Manager is used to view generated audit records. As documented above, the channel between the API Gateway Manager and the Gateway is secured with HTTP basic authentication over TLS. The 'node' field of the receipt identifies the name of the Gateway to which the policy was applied.

6.4 Security Audit

Related SFRs: FAU_GEN.1, FAU_SEL.1, FAU_SEL_EXT.1, FAU_STG.1, FAU_STG_EXT.1

The TOE generates the audit events identified in Table 16. Audit records contains date and time of events, type of events, subject identity (where applicable) and outcome of events, audit records also contains the additional information identified on table 16. The TOE may store logs locally on the file system or remotely on an external audit server. Communication with the external audit server is secured using TLS (refer to section 6.7 for detail).

Authorized users may view all available audit events via the API Gateway Manager. All Audit events can be enabled and disabled on the API Gateway Manager interface.

The domain audit log captures management changes in the API Gateway domain that are written by the Admin Node Manager and by API Gateway instances. This includes details such as API Gateway configuration changes, log in/log out, deployments, user, or topology changes. For example, user Joe deployed a new configuration, admin user created a new group, or user Jane has read deployment data. The domain audit log is enabled by default. However, you can configure filtering options such as the number of events displayed, time interval, and event type. The domain log has the following defaults: 50 files, each 5Mb in size.

To view domain audit log events in the API Gateway Manager web console, perform the following steps:

1. In the API Gateway Manager, select Logs > Domain Audit.

2. Configure the number of events displayed in the Max results per server field on the left. Defaults to 1000.

3. Configure the Time Interval for events. Defaults to 1 day.

4. Click the Filter button to add more viewing options (Event Type or Groups and Servers).

5. Click Apply when finished.

The transaction log is used to stored audit records describing how the API Gateway processes business traffic. By default, the Gateway stores up to 20 transaction files, each of which is 1GB in size. The defaults can be configured in Policy Studio.

The audit files are only accessible to the admin that installed API Gateway. When the audit storage capacity has been reached, the TOE will overwrite the oldest audit files. In addition, the TOE offloads audit data files to an external audit server every 5 minutes; this is a TOE feature that is always present, it does not need to be configured.

The TOE relies on the underlying operating system to provide it with a reliable time stamp for use in the audit records. The TOE does not maintain its own time.

6.5 Robust Administrative Access

Related SFRs:	FIA_AFL.1, FIA_SOS.1, FTA_SSL.3 FTA_SSL.4, FIA_USB.1, FMT_MSA.1(1), FMT_SMR.1, FTP_TRP.1, FTA_TAB.1, FPT_SKP_EXT.1, FPT_APW_EXT.1; FMT_MOF_EXT.1

Access to the TOE can be achieved via the Policy Studio application and the web-based API Gateway Manager interface. Users must authenticate prior to being granted access. Users may authenticate via username and password. When authenticating with Policy Studio, HTTP basic authentication over TLS is used. When authenticating with API Gateway Manager, HTML form-based authentication over TLS is used.

The TOE determines the username from the credentials presented at authentication and associates the defined role with the corresponding username. If the user role is changed while the user is logged in, the change takes effect immediately. For example: an admin user is logged in as 'operator' he is clicking through the pages on the interface to view traffic dashboard and other details viewable by that user. An API Gateway admin logs in to API Gateway Manager on another session and change the role of the 'Operator' user to Policy Developer role. This change will take effect immediately, when the user clicks on a page that is not viewable by the policy developer, the action will be blocked.

The TOE administrative user store is maintained internally. Client services user store is maintained on an external LDAP server. The TOE administrative users passwords are stored in a file as a base-64 encoded salted hash of the plaintext password. The salt is a 16-byte value generated using the SHA1PRNG pseudo-random number generator algorithm. A new salt is used for each password hash, which results in different password hashes for the same password. The algorithm used is provided by the JCE and is PBKDF2 with HMAC SHA1 using a key length of 256 bits. The algorithm repeats the digest of the password along with the salt for 1024 iterations.

All sensitive data (local user store, private keys and their passwords, and passwords required to connect to third-party services) are encrypted in the Entity Store using PBE with the entity store passphrase. The password and a random 8-byte salt are converted using the PKCS#12 mechanism to a key and IV for the encryption algorithm. The encryption algorithm used is EDES, EDE, 3 key, with the SHA1 digest used for generating the IV and key material. Pre-shared keys, symmetric keys, and private keys stored in the entity store cannot be viewed through an interface designed specifically for that purpose. The Entity Store Passphrase can be changed using Policy Studio.The TOE detects when a defined threshold of 6 unsuccessful authentication attempts has occurred in a 5 minutes time frame and will lock the associated account for 30 minutes. The TOE depends on its operational environment to provide cryptographic functionality that it uses.

The TOE allows specification of a password policy in accordance with FIA_SOS.1 and terminates inactive sessions at the web-based API Gateway Manager after an administrator defined period of inactivity. Users may also terminate their own session. For configuration details refer to the *API Gateway Settings Reference* section of the *Administrator Guide.*

The TOE displays an administrator defined banner at logon to the Policy Studio and API Gateway Manager interfaces. For configuration details, refer to the *Configure an advisory banner* section of the *Administrator Guide.*

6.6 Continuity of Enforcement

Related SFRs:	FPT_FLS_EXT.1, , FRU_FLT.1

The Gateway continues policy enforcement in the event of a loss of connectivity with Policy Studio by enforcing the last policy received. Continuous connectivity with the Policy Studio is not

expected or required. When policy is restored after a loss of connectivity, the Gateway will continue to enforce the last policy received until a new policy is deployed.

6.7 Protected Communication

Related SFRs: FTP_ITC.1, FTP_TRP.1, FCS_TLS_EXT.1, FCS_HTTPS_EXT.1

6.7.1 TLS Details

This section provides additional detail regarding the TOE's usage of TLS provided by the operational environment. All Gateway cryptographic operations for TLS use cases covered within the scope of this evaluation are performed by the OpenSSL FIPS Object Module. The API Gateway is configured to uses FIPS Approved algorithms (CAVP certificate numbers AES: #4127; RSA: #2237; ECDSA: #945; SHA: #3396; DRBG: #1247; HMAC: #2700; Component Test: #936).

6.7.1.1 TLS

The TOE makes use of TLS in the following ways:

a) Between service clients and the Gateway – in this case the TOE is a TLS server.

b) Between Policy Studio and the Gateway – in this case the TOE is both a TLS client (Policy Studio) and TLS server (Gateway).

c) Between API Gateway Manager and the Gateway – in this case the TOE is just a TLS server, because the client in this case is the browser, which is not part of the TOE.

d) Between the Gateway and the audit server – in this case the TOE is a TLS client.

e) Between the Gateway and the LDAP server – in this case the TOE is a TLS client.

The TLS implementation has the following characteristics when configured in accordance with the *Secure Installation Guide*:

a) TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346) and TLS 1.2 (RFC 5246) are supported without extensions.

b) Client authentication is supported (i.e. if configured the client must submit a trusted certificate to the server).

c) When acting as either client or server, the TOE is configured to negotiate the following cipher suit. If a listed cipher suite is not supported by the other party then the connection will be refused:

The TOE supports the following ciphersuites for communications with remote administrators and communications with remote audit and LDAP servers:

TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA 30
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_ SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384

7 Rationale

7.1 Conformance Claim Rationale

The following rationale is presented with regard to the PP conformance claims:

n. **TOE type.** As identified in section 2.1, the TOE is an enterprise security management solution that provides centralized management and access control over web services. The API Manager is consistent with the TOE type identified by the ESM Policy Manager PP and the API Gateway is consistent with the TOE type identified in the ESM Access Control PP.

o. **Security problem definition.** As shown in section 3, the threats, OSPs and assumptions are identical to those of the ESM Policy Manager PP and the ESM Access Control PP.

p. **Security objectives.** As shown in section 4, the security objectives are identical to those of the ESM Policy Manager PP and the ESM Access Control PP.

q. **Security requirements.** Section 5 of this ST defines the claimed security requirements. SARs have been reproduced directly from the claimed PPs. There were a number of duplicate SFRs included in both the ESM Policy Manager PP and the ESM Access Control PP. Table 19 below describes how this duplication has been addressed. In addition, the claimed PPs included a number of optional SFRs, Table 20 below describes how these have been addressed. No additional requirements have been specified.

The conformance of this ST to both the ESM Policy Manager PP and the ESM Access Control PP is consistent with the PP application notes presented in section 6.1.1 of each document, which states: 'The ESM PPs represent a family of related Protection Profiles written to encompass the variable capabilities of ESM products. For an ST that claims conformance to multiple PPs within the ESM PP family, it is recommended that the ST author clarify how the ESM components relate to one another through usage of application notes. This will assist the reader in determining how the parts of the product that are to be evaluated correspond with the CC's notion of different ESM capabilities.'

Table 19: Duplicate SFRs

Requirement	How the duplication of SFRs is handled in the ST
FAU_GEN.1	Same base requirement in both PPs. SFR specified once in the ST and events combined in Table 16.
FAU_SEL.1	Same base requirement in both PPS, however Only the API Gateway has an audit trail and so we are claiming the FAU_SEL SFR for the AC PP, but not the PM PP.
FAU_STG_EXT.1	Same base requirement in both PPS, however Only the API Gateway has an audit trail and so we are claiming the FAU_STG SFR for the AC PP, but not the PM PP.
FCS_CKM.1 (optional)	Same requirement in both PPs. Not claimed (optional).
FCS_CKM_EXT.4 (optional)	Same requirement in both PPs. Not claimed (optional).
FCS_COP.1(1) (optional)	Same requirement in both PPs. Not claimed (optional).
FCS_COP.1(2) (optional)	Same requirement in both PPs. Not claimed (optional).

Requirement	How the duplication of SFRs is handled in the ST
FCS_COP.1(3) (optional)	Same requirement in both PPs. Not claimed (optional).
FCS_COP.1(4) (optional)	Same requirement in both PPs. Not claimed (optional).
FCS_HTTPS_EXT.1	Same requirement in both PPs.
FCS_IPSEC_EXT.1	Same requirement in both PPs. Not claimed (optional)
FCS_SSH_EXT.1	Same requirement in both PPs. Not claimed (Optional)
FCS_TLS_EXT.1	Same requirement in both PPs.
FMT_MOF.1	Same requirement in both PPs.
FMT_SMF.1	Same base requirement in both PPs. SFR specified once in the ST and events combined in Table 16.
FMT_SMR.1	Same requirement in both PPs. SFR specified once in the ST.
FPT_APW_EXT.1	Same requirement in both PPs. SFR specified once in the ST.
FPT_SKP_EXT.1	Same requirement in both PPs. SFR specified once in the ST.
FPT_ITC.1	Same requirement in both PPs. SFR specified once in the ST.

Table 20: Optional SFRs

Requirement	Source	Rationale
FTA_TSE.1	Access Control PP	Not Included
ESM_ATD.1	Policy Management PP	Included

7.2 Security Objectives Rationale

All security objectives are drawn directly from the ESM Policy Manager PP and the ESM Access Control PP. A conformance rationale is presented at section 7.1.

7.3 Security Requirements Rationale

Security requirements are drawn directly from the ESM Policy Manager PP and the ESM Access Control PP. A conformance rationale is presented at section 7.1. An unfulfilled dependencies rationale is presented in section 6.1.10 of the ESM Policy Manager PP.

7.4 TOE Summary Specification Rationale

Table 22 provides a coverage mapping showing that all SFRs are mapped to the security functions described in the TSS.

Table 21: Map of SFRs to TSS Security Functions

SFR	Access Control Policy Definition	Access Control Policy Enforcement	Policy Security	Security Audit	Robust Administrative Access	Continuity of Enforcement	Protected Communication
ESM_ACD.1	X						
ESM_ACT.1	X						
ESM_ATD.1	X		X				
ESM_ATD.2	X						
ESM_EAU.2			X				
ESM_EID.2		X					
FAU_GEN.1				X			
FAU_SEL.1				X			
FAU_SEL_EXT.1				X			
FAU_STG.1				X			
FAU_STG_EXT.1				X			
FCO_NRR.2			X				
FCS_HTTPS_EXT.1							X
FCS_TLS_EXT.1							X
FDP_ACC.1		X					
FDP_ACF.1		X					
FIA_AFL.1					X		
FIA_SOS.1			X		X		
FIA_USB.1					X		
FMT_MOF.1		X					
FMT_MOF.1(1)		X					
FMT_MOF.1(2)		X					
FMT_MOF_EXT.1		X			X		
FMT_MSA.1	X	X			X		
FMT_MSA.3	X	X					
FMT_MSA_EXT.5	X	X					
FMT_SMF.1	X	X					
FMT_SMR.1		X			X		
FPT_APW_EXT.1			X		X		

SFR	Access Control Policy Definition	Access Control Policy Enforcement	Policy Security	Security Audit	Robust Administrative Access	Continuity of Enforcement	Protected Communication
FPT_FLS_EXT.1						X	
FPT_RPL.1			X				
FPT_SKP_EXT.1					X		
FRU_FLT.1						X	
FTA_SSL.3					X		
FTA_SSL.4					X		
FTA_TAB.1					X		
FTP_ITC.1			X				X
FTP_TRP.1			X		X		X