**Security Layer – Technical Manual**

****

****



|  |  |
| --- | --- |
| Document Owners | Liudmila Dobriakova [RPLY] |
| Contributors | Lorenzo Bracco [RPLY] |
| Dissemination | Public |
| Date | 30/05/2016 |
| Version | 1.0 |

Table of Contents

1 Authentication and Authorization Introduction 3

1.1 Overview 3

1.2 Authentication and Authorization Architecture 4

1.3 Licensing 5

1.4 Deployment procedure 6

# Authentication and Authorization Introduction

The goal of the security authentication layer is to create a trusted access to the Osmose platform and a correct granting of the resources based on the client that requests the access.

This layer will manage authentication and authorizazion to the Osmose platform and, eventually, among some of its sub-systems.

Security is a one of the most important aspects of any IoT platform and it has several key concepts, such as:

* Confidentiality
* Integrity
* Availability
* Authentication
* Authorization

## Overview

Authentication is the establishment and verification of user identity.

Authorization is the granting of permission to access specific information and carry out specific actions.

For the Osmose platform, we are going to take into account the following important aspects of the security:

• Authentication

• Authorization

Authorization cannot occur without authentication, so authentication is a fundamental prerequisite to start the authorization process; the authentication credentials are used for the proper authorization.

An entity, typically called Authrization Server, use the credentials obtained during the authentication in order to grant or deny the access to the resource requested by a Client Application; there are different protocols to obtain this flow, each of them with a different specification of the involved parties.

The common practices used for the authentication flow are:

• Username and password

• Email-based identification and authentication

• Security tokens in combination with passwords

• Digital certificates

## Authentication and Authorization Architecture

Different tools combination have been tested to find suitable solutions.

To implement Osmose Security Authentication Layer we desided to stop on the ForgeRock solution including OpenAM (downloadable .war file should be deployed on App server in our case we use Apache Tomcat 7) is a ForgeRock open source access management server platform, which use for the correct running the following components:

* OpenDJ is a LDAP server developed by ForgeRock (but other servers could be used, we use this since the same provider and major probability of compatibility and good documented) - this is to store all information about the users, and its main duty is to authenticate users, and decide what kind of information it shares about them with other parties. (Orange blocks with the Identity Provider IdP label)
* Service Provider Policy Agent is another component necessary to protect the resources based on the policies defined using Identity Provider (IdP) and this is a Web Polies Agent provided by ForgeRock and running on Apache HTTP Server - the orange block with the Service Provider (SP) label on the slide.
* And the reverse proxy is an Apache HTTP Server where Policy Agent is running to expose the services to be protected by OpenAM infrastructure.



Figure 1 – The basic IdP-SP setup for Osmose platform

## Licensing

All the software used is open-source and free to download.

OpenAM project is provided under multiple licenses (http://openam.forgerock.org/license.html):

* [CDDL-1.0](http://openam.forgerock.org/license.html#CDDL-1.0)
* [CC BY-NC-ND 3.0](http://openam.forgerock.org/license.html#CC_BY-NC-ND_3.0)

All the components of the Forgerock family (OpenAM, the Policy Agent and OpenDJ) are available also in an enhanced version, which involves a paid subscription with extended support and mid-version updates.

## Deployment procedure

A test deployment has been configured using the following components:

* Ubuntu 15.10
* Apache Tomcat 8
* Apache Web Server 2.4
* OpenAM 12
* OpenDJ 2.6
* Web Policy Agent 4.0

The first step is to download the ZIP package for OpenDJ from <https://forgerock.org/opendj/> and proceed with installation following the guide at <https://backstage.forgerock.com/#!/docs/opendj/2.6/install-guide>:

* When asked for the FQDN the domain of the server must be entered
* The "Only create the base entry" option must be chosen during the procedure in order to allow OpenAM to populate the data store during its configuration

The second step is to download the OpenAM WAR package from <https://forgerock.org/openam/>, copy it to the Apache Tomcat webapps folder and follow [https://backstage.forgerock.com/#!/docs/openam/12.0.0/install-guide#chap-install-core](https://backstage.forgerock.com/#!/docs/openam/12.0.0/install-guide) for the installation:

* Apache Tomcat must be HTTPS enabled
* Configuration is started by going on the web page https://<hostname>:<tomcatport>/openam
* During the procedure OpenAM must provide with the details to connect to OpenDJ

OpenDJ and OpenAM together allow to handle users, groups and policies to control access to the system resources.

The OSMOSE services which need to be protected are exposed locally and run on the WSO2 Enterprise Service Bus. That is why a proxy is used to expose the services on the server public interface using the OpenAM Web Policy Agent, which performs authorization on all incoming requests.

First of all the Apache Web Server must be installed and configured to redirect requests to the local services as needed. For example:

#START PROXY SETTINGG  
ProxyPreserveHost On  
ProxyPass /esb <https://localhost:9443>  
ProxyPassReverse /esb <https://localhost:9443>  
ProxyPass /carbon <https://localhost:9443/carbon>  
ProxyPassReverse /carbon <https://localhost:9443/carbon>  
#END PROXY

The above configuration lines allow to expose at appropriate paths the Enterprise Service Bus endpoints.

Then the Web Policy Agent must be installed (for the download refer to the URL above for OpenAM). The installation guide is at <https://backstage.forgerock.com/#!/docs/openam-policy-agents/3.3.0/web-install-guide>:

* First an agent profile must be created from the OpenAM console
* The installation tool of the web agent must be provided the details inserted for the profile
* In an environment based on Ubuntu and Apache 2.4 (like the one used for testing) you may need to add the following line to the Apache configuration file in order to make the automatic procedure work:

# LoadModule

What remains to be configured is users, groups and policies using the OpenAM console at https://<hostname>:<tomcatport>/openam to define the rules for data access control and allow only the intended users to access specific resources.

In the following figures you can find a few screenshots of the OpenAM console interface.

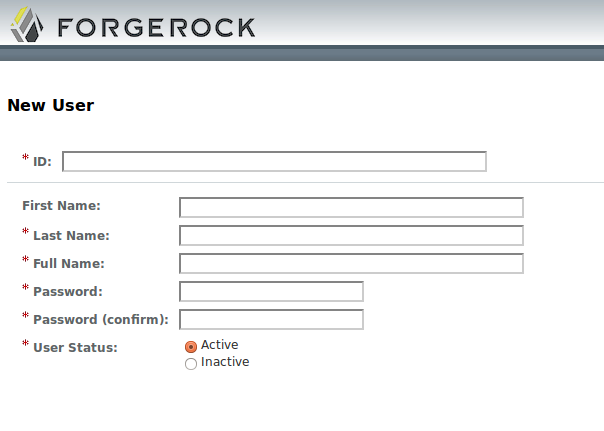


Figure 2 - Form to create a new user on OpenAM

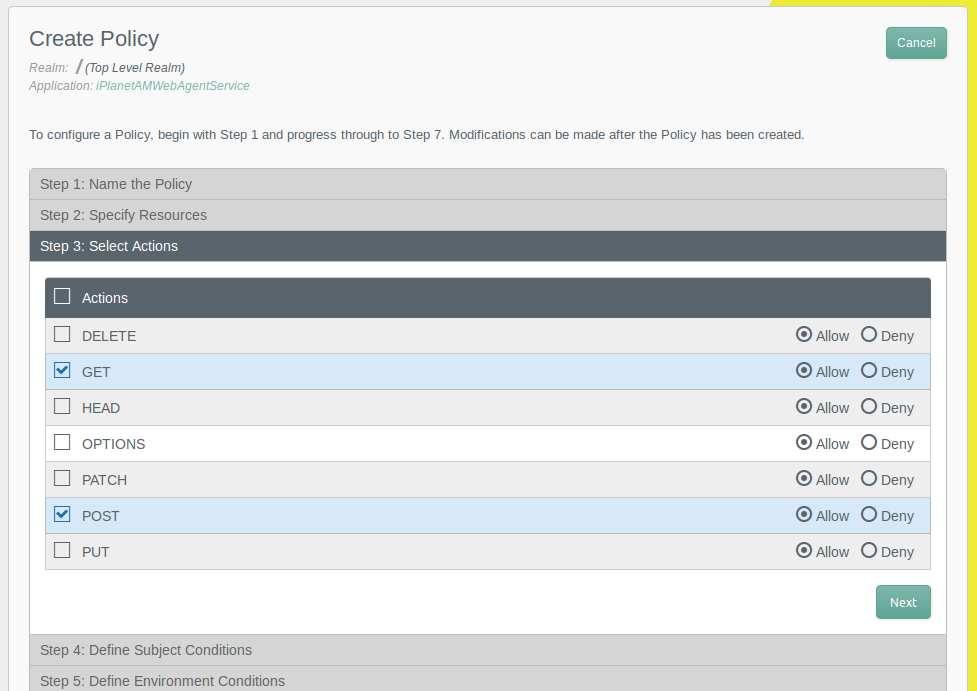


Figure 3 - OpenAM console interface for policy definition

The flow works as follows: the requests come to the Apache server, which thanks to the policy agent talks with OpenAM to authorize the users and, in case of access granted, forwards the requests to the local services of the OSMOSE Data Access Gateway.

Refer to the official guides of the used tools to retrieve the specific user manual:

* OpenAM: <http://openam.forgerock.org/openam-documentation/openam-doc-source/doc/admin-guide/index.html>