webMethods Suite Security Infrastructure Guide

Introduction to SIN

Version 9.5 SP1

November 2013

Security Infrastructure

This document applies to SIN Version 9.5 SP1.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2008-2013 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, United States of America, and/or their licensors.

The name Software AG, webMethods and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at http://documentation.softwareag.com/legal/.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at http://documentation.softwareag.com/legal/ and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices and license terms, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". This document is part of the product documentation, located at http://documentation.softwareag.com/legal/ and/or in the root installation directory of the licensed product(s).

Document ID: SINEXT-INSTALL-95SP1-20130929

Table of Contents

Preface

This document provides a general overview of what Security Infrastructure (SIN) is and how it works. It addresses those readers who want to understand the general concepts underlying the security mechanisms in the webMethods product suite.

Although webMethods products themselves can be configured in a variety of ways, SIN is not product-specific, has no specific hardware requirements and applies under any operating system.

SIN is installed automatically with products installed with the Software AG Installer.

This document introduces SIN under the following headings:

The following table explains the abbreviations and terminology used in this documentation:

Abbreviation	Description
SIN	Security Infrastructure, the name of the product
SSX	Software AG Security eXtensions, an interface for user authentication
IAF	Integrated Authentication Framework, Software AG product that provides a service for authentication and issuance of authentication token or artifact
IAF token	Authentication token that contains all the information about the authenticated user
IAF artifact	Short index to the information provided by the IAF service

1 Overview of SIN

Software AG's webMethods suite of products has a common authentication infrastructure called SIN. It provides the products with security components for authentication of users, management of roles, and query of user, role, and group information. It works both on client-side applications and on server-side applications.

SIN's basic advantage is the re-use of existing security components. For example, SIN supports the same security mechanism for an application that uses Tamino and another one that uses LDAP directory without any change of code on the application level.

SIN is based on the Java Authentication and Authorization Service (JAAS). The JAAS framework allows you to define stacks of `LoginModules` that can be defined without code changes.

The existence of more than one login module is caused by the need to accommodate different authentication methods. Most modules depend on third-party libraries and are kept in separate Java packages and JAR files. For this reason, SIN is distributed as four jar files:

- *sin-common.jar*
- *sin-ssx.jar*
- *sin-xmlserver.jar*
- *sin-misc.jar*

All interfaces and common classes that are to be used by an application programmer are contained in *sin-common.jar*. The other JAR files contain `LoginModules` that you can configure according to your environment and the desired authentication process.

Functionality

SIN's functionality enables you to have the following set of capabilities:

- Provide JAAS support
- Authenticate users

- Retrieve group and role information for the authenticated user
- Retrieve roles for arbitrary users from a role repository
- Manage roles in a role repository

Why Use JAAS in SIN?

The authentication "tool" of SIN is the `LoginModules`.

They are based on the Oracle JAAS framework, which is a security framework for authenticating users. JAAS accommodates the information for groups and roles in classes derived from `java.security.Principal`. The API is integrated in JDK since version 1.4.

With JAAS, you have the following benefits:

- Authentication is independent from applications
- Professional services do not need special know-how to customize and re-use standard components for different authentication schemes
- Products have a high level of integration and can accommodate customer environments and requirements at install time

How Does SIN Work?

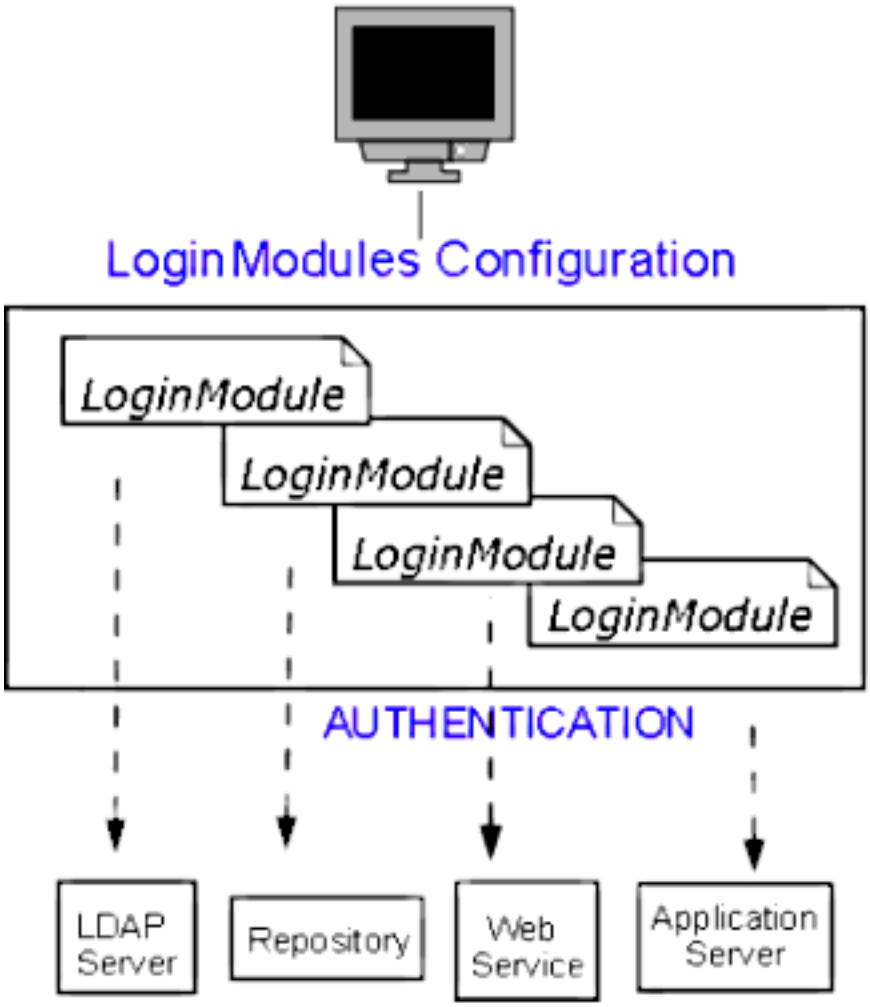
The process of authentication includes the successful calling of a `LoginModule`.

`LoginModules` can prompt for and verify a user name and a password. If authentication is successful, JAAS creates a `Subject` that contains one or more `Principals` with security-related attributes like passwords and cryptographic keys.

A particular application defines a `LoginModules` configuration that is instantiated from the application. The configuration specifies the `LoginModule` that is to be used with a particular application.

What is a `LoginContext`?

The `LoginContext` is a grouping of `LoginModules`. It provides the basic methods for user authentication. The stack of login modules allows you to configure applications to use more than one `LoginModule`. For example, you can configure both a `KerberosLoginModule` and an `X500LoginModule`.



How Does a System Authenticate a User?

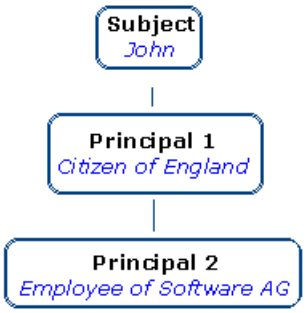
1. The `LoginContext` locates the JAAS configuration file of the appropriate `LoginModule`.
2. The application uses the configuration file to authenticate the user.
3. After the application has authenticated the user, the `LoginContext` creates `Principals` and adds them to the `Subject`.

What is the Difference Between a `Principal` and a `Subject`?

The `Principal` interface represents the abstract notion of a principal that can be any entity, such as an individual, a corporation, and a logon ID, while the `Subject` class represents a grouping of related information for a single entity. Such information includes the `Subject`'s identities, as well as its security-related attributes (passwords and cryptographic keys).

For example, if a `Subject` is a person named "John", he may have two `Principals`:

- **Principal 1**
  It represents "John" as the citizen of a particular country.
- **Principal 2**
  It represents "John" as the employee of a particular company.



Both Principals refer to the same `Subject` even though each has a different name.

2 Related Resources

Following are links to external resources related to SIN:

- **Version 6 of the Java™ Platform, Standard Edition Documentation** - *http://docs.oracle.com/javase/6/docs/api/*
- **Security** - *http://docs.oracle.com/javase/6/docs/technotes/guides/security/index.html*
- **JAAS Reference Guide** - *http://docs.oracle.com/javase/6/docs/technotes/guides/security/jaas/JAASRefGuide.html*
- **JAAS Tutorials** - *http://docs.oracle.com/javase/6/docs/technotes/guides/security/jaas/tutorials/index.html*
- **Introduction to JAAS and Java GSS-API Tutorials** - *http://docs.oracle.com/javase/6/docs/technotes/guides/security/jgss/tutorials/index.html*