User Registries and Authentication Mechanisms

Introduction

A User Registry is a repository of user ids and credentials for authentication purposes. An Authentication Mechanism provides support to use these User Registries for authenticating users. This document serves as a guide to understanding the different User Registry and Authentication Mechanism options available when enabling security within WebSphere Application Server..

The document also discusses the various WebSphere Application Server specific components that closely interact with the User registry and thereby enforce application security based on the user credentials embedded in the external client request for an application resource.

User Registry

WebSphere Application Server (WAS) provides a **UserRegistry** interface to support many different types of directory services including various OS platform user registries (NT registry, NT Domain Controller, Unix, iSeries) and LDAP. Other types can be supported by implementing the UserRegistry interface.

There are three categories of User Registries:

- **Local OS UserRegistry** – An implementation of the UserRegistry interface that interfaces with the local OS platform user registry. The local OS platform user registry contains the credentials of the operation system users that have a valid userid/password and permissions on the same machine as the WebSphere Application Server. On a Windows platform, the local OS registry can also be the Windows Domain.

- **LDAP UserRegistry** – An implementation of the UserRegistry interface that interfaces with a LDAP server to retrieve user information. The server can be local (on the same machine as the WebSphere Application Server) or remote. Further, the access to the LDAP server from the Application Server can be over Secure Socket Layer (SSL).

- **Custom UserRegistry** – Custom implementations that retrieve the users' information stored in a database or a file system or any other custom storage.

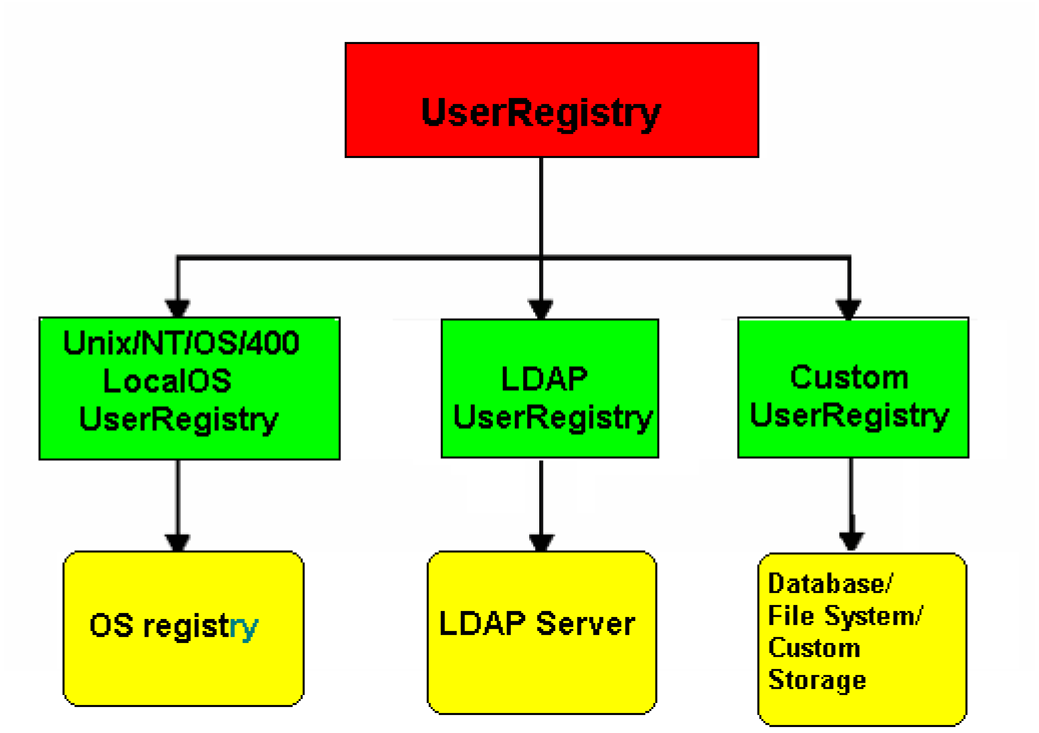

UserRegistry
Unix/NT/OS/400 LocalOS UserRegistry
LDAP UserRegistry
Custom UserRegistry
OS registry
LDAP Server
Database/ File System/ Custom Storage

Figure 1

Authentication Mechanisms

The WebSphere Application Server Authentication Mechanism is implemented as a Java Authentication and Authorization Service (JAAS) **LoginModule**.

The Java Authentication and Authorization Service (JAAS) is a set of APIs that can be used for two purposes:

- For authentication of users, to reliably and securely identify who is currently making the request within an application, an applet, a bean, or a servlet;
- For authorization of users, to insure the current execution request and the user associated with it have the proper credentials to execute Java code, regardless of whether the code is running as an application, an applet, a bean, or a servlet; and

JAAS provides a pluggable architecture where different login modules can be used for authentication of different clients and devices. These login modules are separate from the application and allow Java applications to remain independent from underlying authentication technologies. New or updated technologies can be plugged in without requiring modifications to the application itself. For example, one type of LoginModule may perform a username/password-based form of authentication. Other LoginModules may interface to hardware devices such as smart cards or biometric devices.

JAAS authorization can be used for programmatic credential validation within an application.

WebSphere Application Server implements two JAAS LoginModule Authentication Mechanisms:

1. Simple WebSphere Authentication Mechanism (**SWAM**) Login Module, and
2. Light Weight Third Party (**LTPA**) Authentication Mechanism Login Module.

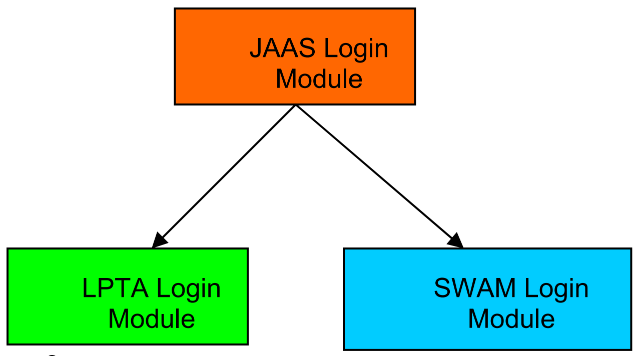


Figure 2

The difference between the two login modules is in regards to the token generation. After verifying the userid and password, the LTPA Authentication Mechanism will generate an LTPA security token which is encrypted by a LTPA key. The SWAM Authentication Mechanism does not generate a security token.

LTPA security tokens can be forwarded and Single Sign-on can be supported within a Network Deployment environment. LTPA is available in a single server installation, however it is the only Authentication Mechanism supported in a Network Deployment environment. The SWAM Authentication Mechanism is provided only on the standalone Base package (not available in Network Deployment version) and is designed for standalone server environment and for testing environment. Single Sign-on is not provided when using SWAM.

WebSphere Security using User Registry and Authentication Mechanism

Now that you have a basic understanding of User Registry, we will get into the details of how WebSphere Application Server provides support for User Registry to enforce application security.

WebSphere Application Server(s) may receive client requests from many different sources and transports. Client requests may be coming from a from a browser client via HTTP/HTTPS protocol, from a Java client via the RMI/IIOP protocol or from a Web Service client via SOAP over HTTP/HTTPS protocol. Aside from the difference in transports, security protocols, and containers, the request flow in all those cases is pretty much the same from the security perspective.

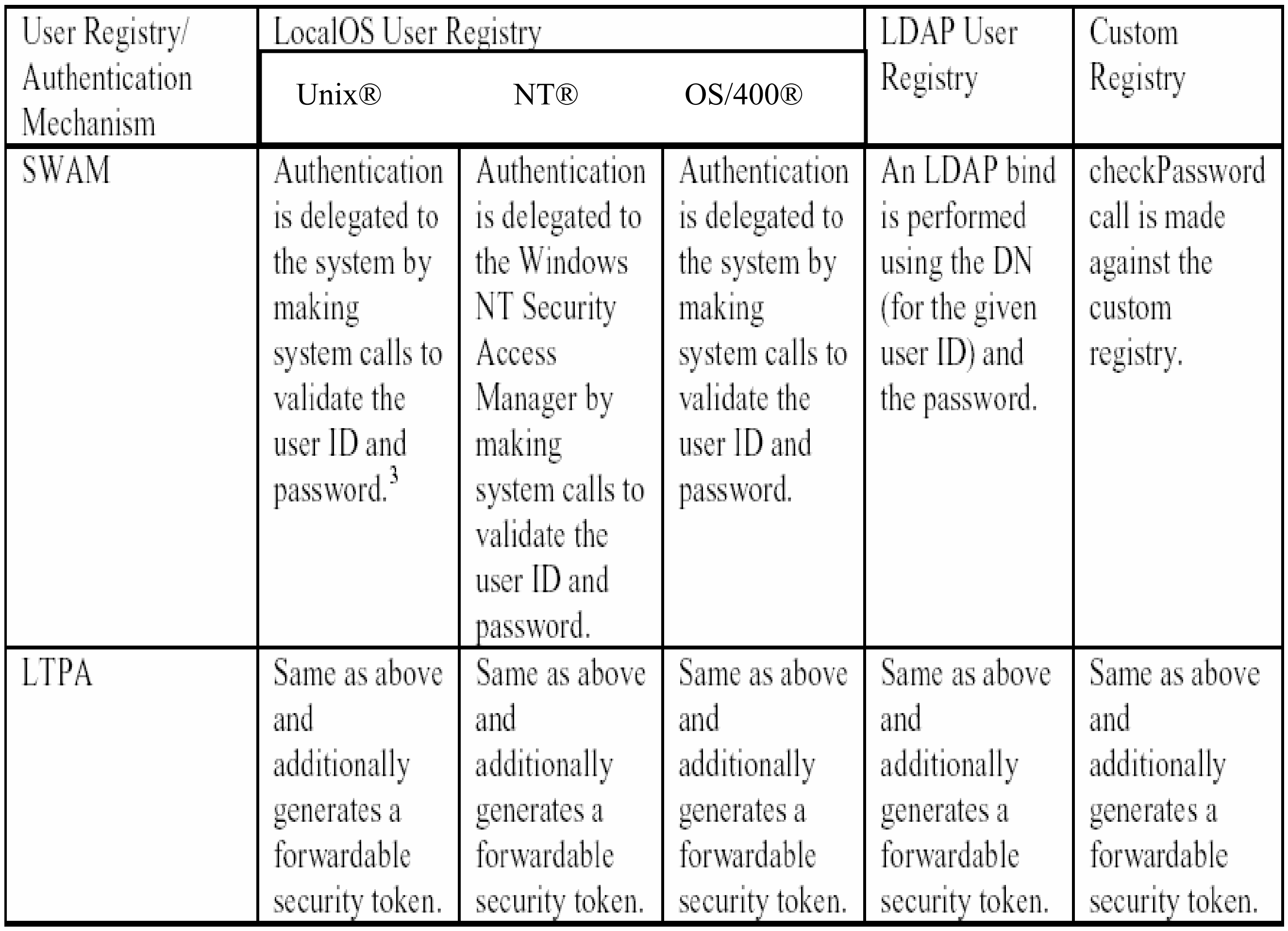
A security interceptor abstraction represents the module that retrieves the client Authentication data from the client request to perform Authentication. The security interceptor may be realized by the Web and EJB security collaborators in the case of JSP/Servlet and EJB,

The Security interceptor uses the **WebSphere Application Server - Security Server** to perform Authentication. The Security Server consists of two sub components:

1. **Authentication Mechanism**
2. **User Registry**.

The WebSphere Application Server Authentication process is based on the type of Authentication Mechanism, Authentication data, and the user registry that contains the user data. In cases when a user ID and password are supplied for Authentication, the task of checking the password is delegated to a user registry. In those cases where a digital certificate is used to prove the user's identity, the certificate credentials are mapped to an associated user registry entry.

To accommodate different types of Authentication mechanisms, Authentication methods, and user registries, WebSphere Application Server V5 allows the Authentication Mechanism to be configured independently of the user registries. Both SWAM and LTPA Authentication mechanisms may be used with any type of user registries. The following table summarizes the Authentication Mechanism and user registry configurations.



User Registry/ Authentication Mechanism	LocalOS User Registry			LDAP User Registry	Custom Registry
	Unix®	NT®	OS/400®		
SWAM	Authentication is delegated to the system by making system calls to validate the user ID and password.[3]	Authentication is delegated to the Windows NT Security Access Manager by making system calls to validate the user ID and password.	Authentication is delegated to the system by making system calls to validate the user ID and password.	An LDAP bind is performed using the DN (for the given user ID) and the password.	checkPassword call is made against the custom registry.
LTPA	Same as above and additionally generates a forwardable security token.	Same as above and additionally generates a forwardable security token.	Same as above and additionally generates a forwardable security token.	Same as above and additionally generates a forwardable security token.	Same as above and additionally generates a forwardable security token.

Table 1

Note: When running as non-root user (on the machine WebSphere Application Server is installed), you must use either LDAP Registry or Custom Registry to for the user registry. Local OS Registry cannot be used in such a case. There are no such restrictions, however, when running as a root user.

How the Authentication Mechanism and User Registry implementation(s) work together:

Figure 4 below outlines the sequence of steps involved in authenticating a user request coming into the WebSphere Application Server. Authentication is required for enterprise beans clients and Web clients when they are accessing protected resources. Web clients use the HTTP or HTTPS protocol and Enterprise beans clients send the Authentication information using CSIV2 or SAS protocol to send the Authentication information.

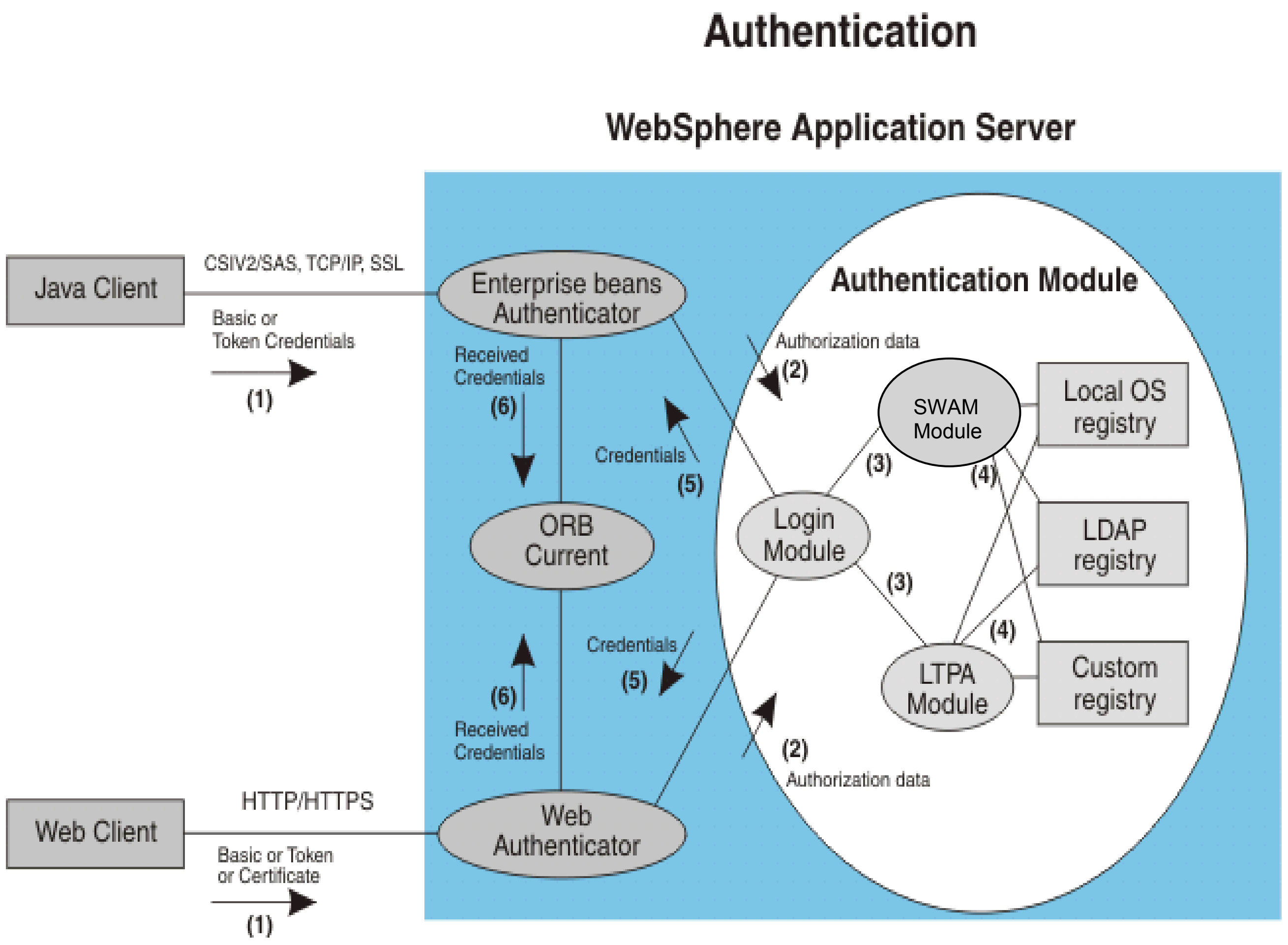


Figure 4

The Web Authentication is performed by the Web Authentication module and the EJB Authentication is performed by the EJB Authentication module. The WebAuthenticator and EJBAuthenticator pass the Authentication data to the configured Login Module (step 2 in the picture) which can be either SWAM or LPTA.

The Login Module uses the configured User Registry implementation to perform the Authentication. The User Registry implementation, in turn, interfaces with the actual repository to authenticate the user. If Authentication is successful, the Login Module creates a JAAS subject and stores the CORBA credentials derived from the Authentication data in the public credentials list of the subject

What has been covered:

1. The concept of User Registry and Authentication Mechanism within WebSphere Application Server.
2. Categories of User Registry.
3. Authentication Mechanisms provided by WebSphere Application Server.
4. Usage of User Registry and default Authentication Mechanism(s) to configure Security.

Relevant Resources:

1. http://publib.boulder.ibm.com/infocenter/ws51help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/csec_registries.html -- User Registries in WebSphere Application Server
2. http://publib.boulder.ibm.com/infocenter/ws51help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rsec_customauth.html -- Custom User Registries in WebSphere Application Server
3. http://publib.boulder.ibm.com/infocenter/wasinfo/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rsec_jaasauthor.html -- JAAS and WebSphere Application Server
4. http://java.sun.com/products/jaas/overview.html -- JAAS overview
5. http://publib7b.boulder.ibm.com/wasinfo1/en/info/aes/ae/csec_aumech.html -- Authentication mechanisms: WebSphere Application Server.

Trademarks and Disclaimers

This page is left intentionally blank.