BEA WebLogic SIP Server™

Configuring Security for WebLogic SIP Server

Version 2.2
Revised: May 16, 2006

Copyright

Restricted Rights Legend

This software is protected by copyright, and may be protected by patent laws. No copying or other use of this software is permitted unless you have entered into a license agreement with BEA authorizing such use. This document is protected by copyright and may not be copied photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form, in whole or in part, without prior consent, in writing, from BEA Systems, Inc.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE DOCUMENTATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA SYSTEMS DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE DOCUMENT IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks and Service Marks

Copyright © 1995-2006 BEA Systems, Inc. All Rights Reserved. BEA, BEA JRockit, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop, Built on BEA, Jolt, JoltBeans, SteelThread, Top End, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA AquaLogic, BEA AquaLogic Data Services Platform, BEA AquaLogic Enterprise Security, BEA AquaLogic Interaction, BEA AquaLogic Interaction Analytics, BEA AquaLogic Interaction Collaboration, BEA AquaLogic Interaction Content Services, BEA AquaLogic Interaction Data Services, BEA AquaLogic Interaction Integration Services, BEA AquaLogic Interaction Process, BEA AquaLogic Interaction Publisher, BEA AquaLogic Interaction Studio, BEA AquaLogic Service Bus, BEA AquaLogic Service Registry, BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Kodo, BEA Liquid Data for WebLogic, BEA Manager, BEA MessageQ, BEA SALT, BEA Service Architecture Leveraging Tuxedo, BEA WebLogic Commerce Server, BEA WebLogic Communications Platform, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Enterprise Security, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Java Adapter for Mainframe, BEA WebLogic JDriver, BEA WebLogic Log Central, BEA WebLogic Mobility Server, BEA WebLogic Network Gatekeeper, BEA WebLogic Personalization Server, BEA WebLogic Personal Messaging API, BEA WebLogic Platform, BEA WebLogic Portlets for Groupware Integration, BEA WebLogic Real Time, BEA WebLogic RFID Compliance Express, BEA WebLogic RFID Edge Server, BEA WebLogic RFID Enterprise Server, BEA WebLogic Server Process Edition, BEA WebLogic SIP Server, BEA WebLogic WorkGroup Edition, BEA Workshop for WebLogic Platform, BEA Workshop JSP, BEA Workshop JSP Editor, BEA Workshop Struts, BEA Workshop Studio, Dev2Dev, Liquid Computing, and Think Liquid are trademarks of BEA Systems, Inc. Accelerated Knowledge Transfer, AKT, BEA Mission Critical Support, BEA Mission Critical Support Continuum, and BEA SOA Self Assessment are service marks of BEA Systems, Inc.

All other names and marks are property of their respective owners.

Contents

1. Overview of WebLogic SIP Server Security Features

Authentication for SIP Servlets 1-1

Authentication Providers 1-2

Overriding Authentication with Trusted Hosts 1-3

P-Asserted-Identity Support 1-3

Role Assignment for SIP Servlet Declarative Security 1-3

Security Event Auditing 1-3

Common Security Configuration Tasks 1-4

2. Configuring Digest Authentication

Overview of Digest Authentication 2-1

What Is Digest Authentication? 2-1

Digest Authentication Support in WebLogic SIP Server 2.2 2-2

Prerequisites for Configuring LDAP Digest Authentication 2-6

Steps for Configuring Digest Authentication 2-7

Configure the LDAP Server or RDBMS 2-8

Using Unencrypted Passwords 2-8

Using Precalculated Hash Values 2-9

Using Reverse-Encrypted Passwords 2-9

Reconfigure the DefaultAuthenticator Provider 2-10

Configure an Authenticator Provider 2-10

Configure a New Digest Identity Asserter Provider 2-11

Configure an LDAP Digest Identity Asserter Provider 2-11

Configure an RDBMS Digest Identity Asserter Provider 2-14

Sample Digest Authentication Configurations 2-15

Oracle Internet Directory Server 2-16

WebLogic SIP Server Embedded LDAP 2-17

Store User Password Information in the Description Field 2-17

Set the Embedded LDAP Password 2-17

Configure the Digest Identity Asserter Provider 2-18

3. Configuring Client-Cert Authentication

Overview of Client-Cert Authentication 3-1

Configuring SSL and X509 for WebLogic SIP Server 3-2

Configuring the Default Identity Asserter 3-3

Configuring the LDAP X509 Identity Asserter 3-4

Configuring WebLogic SIP Server to Use WL-Proxy-Client-Cert 3-6

Supporting Perimeter Authentication with a Custom IA Provider 3-7

4. Configuring P-Asserted-Identity Assertion

Understanding Trusted Host Forwarding with P-Asserted-Identity 4-1

Overview of Strict and Non-Strict P-Asserted-Identity Asserter Providers 4-5

Configuring a P-Asserted-Identity Assertion Provider 4-6

5. Configuring 3GPP HTTP Authentication Providers

Overview 5-1

Configuring a X-3GPP-Asserted-Identity Provider 5-2

Overview of WebLogic SIP Server Security Features

The following sections provide an overview of WebLogic SIP Server security:

- “Authentication for SIP Servlets” on page 1-1
- “Overriding Authentication with Trusted Hosts” on page 1-3
- “P-Asserted-Identity Support” on page 1-3
- “Role Assignment for SIP Servlet Declarative Security” on page 1-3
- “Security Event Auditing” on page 1-3
- “Common Security Configuration Tasks” on page 1-4

Authentication for SIP Servlets

WebLogic SIP Server users must be authenticated whenever they request access to a protected resource, such as a protected method within a deployed SIP Servlet. WebLogic SIP Server enables you to implement user authentication for SIP Servlets using any of the following techniques:

- **DIGEST authentication** uses a simple challenge-response mechanism to verify the identity of a user over SIP or HTTP. This technique is described in “Configuring Digest Authentication” on page 2-1.
- **CLIENT-CERT authentication** uses an X509 certificate chain passed to the SIP application to authenticate a user. The X509 certificate chain can be provided in a number of different ways. In the most common case, two-way SSL handshake is performed before

transmitting the chain to ensure secure communication between the client and server. CLIENT-CERT authentication is described fully in "Configuring Client-Cert Authentication" on page 3-1.

- **BASIC authentication** uses the `Authorization` SIP header to transmit the username and password to SIP Servlets. BASIC authentication is not recommended for production systems unless you can somehow ensure that all connections between clients and the WebLogic SIP Server instance are secure. This document does not provide configuration instructions for using BASIC authentication.

Different SIP Servlets deployed on WebLogic SIP Server can use different authentication mechanisms as necessary. The required authentication mechanism is specified in the `auth-method` element of the SIP Servlet's `sip.xml` deployment descriptor. The deployment descriptor may also define which resources are to be protected, listing specific role names that are required for access. See Securing SIP Servlet Resources in *Developing Applications with WebLogic SIP Server* for information about securing resources and mapping roles in the SIP Servlet deployment descriptor.

Authentication Providers

WebLogic SIP Server authentication services are implemented using one or more authentication providers. An authentication provider performs the work of proving the identity of a user or system process, and then transmitting the identity information to other components of the system.

You can configure and use multiple authentication providers to uses different authentication methods, or to work together to provide authentication. For example, when using Digest authentication you typically configure both a Digest Identity Asserter provider to assert the validity of a digest, and a second LDAP or RDBMS authentication provider that determines the group membership of a validated user.

When linking multiple authentication providers, you must specify the order in which providers are used to evaluate a given user, and also specify how much control each provider has over the authentication process. Each provider can contribute a "vote" that specifies whether or not the provider feels a given user is valid. The provider's control flag indicates how the provider's vote is used in the authentication process.

For more information about configuring providers, see either "Configuring Digest Authentication" on page 2-1 or "Configuring Client-Cert Authentication" on page 3-1.

Overriding Authentication with Trusted Hosts

WebLogic SIP Server also enables you to designate trusted hosts for your system. Trusted hosts are hosts for which WebLogic SIP Server performs no authentication. If the server receives a SIP message having a destination address that matches a configured trusted hostname, the message is delivered without Authentication. See sip-security in *Configuring and Managing WebLogic SIP Server* for more information.

P-Asserted-Identity Support

WebLogic SIP Server supports the `P-Asserted-Identity` SIP header as described in RFC3325. This functionality automatically logs in using credentials specified in the `P-Asserted-Identity` header when they are received form a trusted host. When combined with the `privacy` header, `P-Asserted-Identity` also determines whether the message can be forwarded to trusted and non-trusted hosts. See "Configuring P-Asserted-Identity Assertion" on page 4-1 for more information.

Role Assignment for SIP Servlet Declarative Security

The SIP Servlet API specification defines a set of deployment descriptor elements that can be used for providing declarative and programmatic security for SIP Servlets. The primary method for declaring security constraints is to define one or more `security-constraint` elements and role definitions in the `sip.xml` deployment descriptor. WebLogic SIP Server adds additional deployment descriptor elements to help developers easily map SIP Servlet roles to actual principals and/or roles configured in the SIP Servlet container. See Securing SIP Servlet Resources in *Developing Applications with WebLogic SIP Server* for more information.

Security Event Auditing

WebLogic SIP Server includes an auditing provider that you can configure to monitor authentication events in the security realm. See Configuring a WebLogic Auditing Provider in the WebLogic Server 8.1 documentation for more information.

Common Security Configuration Tasks

Table 1-1 lists WebLogic SIP Server configuration tasks and provides links to additional information.

Table 1-1 Security Configuration Tasks

Task	Description
"Configuring Digest Authentication" on page 2-1	• Understanding the Digest identity assertion providers • Configuring LDAP Digest authentication • Configuring Digest authentication with an RDBMS
"Configuring Client-Cert Authentication" on page 3-1	• Understanding client-cert authentication solutions • Delivering X509 certificates over 2-way SSL • Developing a Perimeter authentication solution • Using the WebLogic SIP Server `WL_Client_Cert` header to deliver X509 certificates
"Configuring P-Asserted-Identity Assertion" on page 4-1	• Understand forwarding rules for SIP messages having the `P-Asserted-Identity` header • Configuring `P-Asserted-Identity` providers
Securing SIP Servlet Resources in *Developing Applications with WebLogic SIP Server*	• Defining security constraints for a SIP Servlet • Mapping SIP Servlet roles to WebLogic SIP Server roles and principals • Debugging SIP Servlet security constraints
sip-security in *Configuring and Managing WebLogic SIP Server*	• Configuring trusted hosts

Configuring Digest Authentication

The following sections describe how to configure WebLogic SIP Server to use Digest authentication with a supported LDAP server or RDBMS:

- "Overview of Digest Authentication" on page 2-1
- "Prerequisites for Configuring LDAP Digest Authentication" on page 2-6
- "Steps for Configuring Digest Authentication" on page 2-7
- "Sample Digest Authentication Configurations" on page 2-15

Overview of Digest Authentication

The following sections provide a basic overview of Digest authentication, and describe Digest authentication support and configuration in WebLogic SIP Server 2.2.

What Is Digest Authentication?

Digest authentication is a simple challenge-response mechanism used to authenticate a user over SIP or HTTP. Digest authentication is fully described in RFC 2617.

When using Digest authentication, if a client makes an un-authenticated request for a protected server resource, the server challenges the client using a nonce value. The client uses a requested algorithm (MD5 by default) to generate an encrypted response—a Digest—that includes a username, password, the nonce value from the challenge, the SIP method, and the requested URI.

The server verifies the client Digest by recreating the Digest value and comparing it with the client's Digest. To recreate the Digest value the server requires a hash of the "A1" value (see RFC 2617) that includes, at minimum, the nonce, username, password and realm name. The server either recreates the hash of the A1 value using a stored clear-text password for the user, or by obtaining a precalculated hash value. Either the clear-text password or precalculated hash value can be stored in an LDAP directory or accessed from an RDBMS using JDBC. The server then uses the hash of the A1 value to recreate the Digest and compare it to the client's Digest to verify the user's identity.

Digest authentication provides secure authorization over HTTP because the clear text password is never transmitted between the client and server. The use of nonce values in the client challenge also ensures that Digest authentication is resistant to replay attacks. See Figure 2-1 for a more detailed explanation of the challenge-response mechanism for a typical request.

Digest Authentication Support in WebLogic SIP Server 2.2

WebLogic SIP Server 2.2 includes LDAP Digest Identity Asserter security providers for asserting the validity of a client's Digest using LDAP or an RDBMS. A separate authorization provider is required to complete the authentication process (see "Configure an Authenticator Provider" on page 2-10).

The Digest Identity Asserter only verifies a user's credentials using the client Digest. After the Digest is verified, the configured authorization provider completes the authentication process by checking for the existence of the user (by username) and also populating group membership for the resulting `javax.security.auth.Subject`.

The Digest Identity Asserter provider requires that user credentials be stored in an LDAP server or RDBMS in one of the following ways:

- **Unencrypted (clear text) passwords.** The simplest configuration stores users' unencrypted passwords in a store. If you choose this method, BEA recommends using an SSL connection to the LDAP store or database to reduce the risk of exposing clear text passwords in server-side network traffic. Some LDAP stores do not support storing unencrypted passwords by default; in this case you must create or use a dedicated credential attribute on the LDAP server for storing the password. See "Configure the LDAP Server or RDBMS" on page 2-8.
- **A pre-calculated hash of each password, username, and realm.** If storing unencrypted passwords is unacceptable, you can instead store a pre-calculated hash value of the username, security-realm, and password in a new or existing attribute in LDAP or an RDBMS. The Digest Identity Asserter then retrieves only the hash value for comparison to

the client-generated hash in the Digest. Storing pre-calculated hash values provides additional security.

The LDAP Digest Identity Asserter is compatible with any LDAP provider that permits storage of a clear text password or pre-calculated hash value.

Notes: You cannot change the schema for the built-in LDAP store to add a dedicated field for storing clear text passwords or pre-calculated hash values. However, you can use the predefined "description" field to store password information for testing or demonstration purposes.

If you do not use the DefaultAuthenticator provider for authentication decisions, you must make DefaultAuthenticator an optional provider (ControlFlag="SUFFICIENT" or lower) before you can use Digest authentication. This will generally be the required configuration in production installations where a separate LDAP store is used to maintain clear text or hashed password information.

Figure 2-1 Digest Authentication in WebLogic SIP Server 2.2

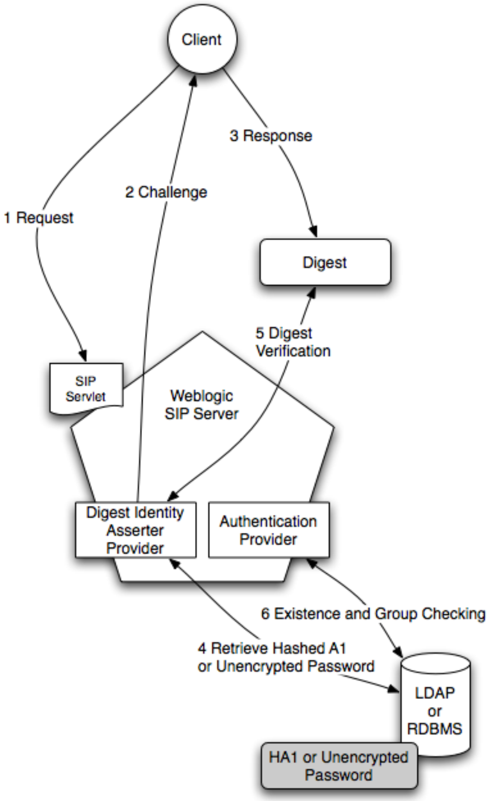

Client
3 Response
2 Challenge
1 Request
Digest
5 Digest Verification
SIP Servlet
Weblogic SIP Server
Digest Identity Asserter Provider
Authentication Provider
6 Existence and Group Checking
4 Retrieve Hashed A1 or Unencrypted Password
LDAP or RDBMS
HA1 or Unencrypted Password

Figure 2-1 shows the basic architecture and use of an Identity Asserter provider for a typical client request:

1. The client makes an unauthorized request for a protected application resource. (SIP Servlet resources can be protected by specifying security constraints in the `sip-xml` deployment descriptor. See Controlling Access to SIP Servlet Resources.)

2. The Digest Identity Asserter provider generates a challenge string consisting of the nonce value, realm name, and encryption algorithm (either MD5 or MD5-sess). The SIP container delivers the challenge string to the client.

Note: The Digest Identity Asserter maintains a cache of used nonces and timestamps for a specified period of time. All requests with a timestamp older than the specified timestamp are rejected, as well as any requests that use the same timestamp/nonce pair as the most recent timestamp/nonce pair still in the cache.

3. The client uses the encryption algorithm to create a Digest consisting of the username, password, real name, nonce, SIP method, request URI, and other information described in RFC 2617.
4. The Digest Identity Asserter verifies the client Digest by recreating the Digest value using a hash of the A1 value, nonce, SIP method, and other information. To obtain a hash of the A1 value, the Identity Asserter either generates HA1 by retrieving a clear-text password from the store, or the Identity Asserter retrieves the pre-calculated HA1 from the store.
5. The generated Digest string is compared to the client's Digest to verify the user's identity.
6. If the user's identity is verified, an authentication provider then determines if the user exists and if it does, the authentication provider populates the `javax.security.auth.Subject` with the configured group information. This step completes the authentication process.

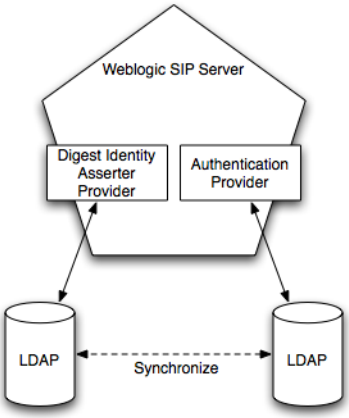
   Note: If you do not require user existence checking or group population, you can use the special "no-op" Identity Assertion Authenticator to avoid an extra connection to the LDAP Server; see "Configure an Authenticator Provider" on page 2-10 for more information.

   After authentication is complete, the SIP Servlet container performs an authorization check for the logged in `javax.security.auth.Subject` against the declarative security-constraints defined in the Servlet's `sip.xml` deployment descriptor.

The LDAP Digest Identity Asserter and the configured Authentication provider can either use the same LDAP store or different stores.

Note: If you use multiple LDAP stores, you must also create some infrastructure to keep both stores synchronized in response to adding, removing, or changing user credential changes, as shown in Figure 2-2. Maintaining LDAP stores in this manner is beyond the scope of this documentation.

Figure 2-2 Multiple LDAP Servers



Prerequisites for Configuring LDAP Digest Authentication

In order to configure Digest authentication you must understand the basics of LDAP servers and LDAP administration. You must also understand the requirements and restrictions of your selected LDAP server implementation, and have privileges to modify the LDAP configuration as well as the WebLogic SIP Server configuration.

Table 2-1 summarizes all of the information you will need in order to fully configure your LDAP server for Digest authentication with WebLogic SIP Server 2.2.

Note that the LDAP authentication provider and the Digest Authentication Identity Asserter provider can be configured with multiple LDAP servers to provide failover capabilities. If you want to use more than one LDAP server for failover, you will need to have connection information for each server when you configure Digest Authentication. See "Steps for Configuring Digest Authentication" on page 2-7.

Table 2-1 Digest Identity Asserter Checklist

Item	Description	Sample Value
Host	The host name of the LDAP server.	MyLDAPServer
Port	The port number of the LDAP server. Port 389 is used by default.	389

Table 2-1 Digest Identity Asserter Checklist

Item	Description	Sample Value
Principal	A Distinguished Name (DN) that WebLogic SIP Server can use to connect to the LDAP Server.	cn=ldapadminuser
Credential	A credential for the above principal name (generally a password).	ldapadminuserpassword
LDAP Connection Timeout	The configured timeout value for connections to the LDAP server (in seconds). For best performance, there should be no timeout value configured for the LDAP server. If a timeout value is specified for the LDAP server, you should configure the Digest Identity Asserter provider timeout to a value equal to or less than the LDAP server's timeout.	30 seconds
User From Name Filter	An LDAP search filter that WebLogic SIP Server will use to locate a given username. If you do not specify a value for this attribute, the server uses a default search filter based on the user schema.	(&(cn=%u)(objectclass=pe rson))
User Base DN	The base Distinguished Name (DN) of the tree in the LDAP directory that contains users.	cn=users,dc=mycompany, dc=com
Credential Attribute Name	The credential attribute name used for Digest calculation. This corresponds to the attribute name used to store unencrypted passwords or pre-calculated hash values. See "Configure the LDAP Server or RDBMS" on page 2-8.	hashvalue
Digest Realm Name	The realm name to use for Digest authentication.	mycompany.com
Digest Algorithm	The algorithm that clients will use to create encrypted Digests. WebLogic SIP Server supports both MD5 and MD5-sess algorithms. MD5 is used by default.	MD5
Digest Timeout	The Digest authentication timeout setting. By default this value is set to 120 seconds.	120

Steps for Configuring Digest Authentication

Follow these steps to configure Digest authentication with WebLogic SIP Server 2.2:

1. "Configure the LDAP Server or RDBMS" on page 2-8.

2. "Reconfigure the DefaultAuthenticator Provider" on page 2-10.

   Note: DefaultAuthenticator is set up as a required authentication provider by default. If the DefaultAuthentication provider, which works against the embedded LDAP store, is not used for authentication decisions, you must change the Control Flag to "SUFFICIENT".

3. "Configure an Authenticator Provider" on page 2-10.
4. "Configure a New Digest Identity Asserter Provider" on page 2-11.

The sections that follow describe each step in detail.

Configure the LDAP Server or RDBMS

The LDAP server or RDBMS used for Digest verification must store either unencrypted, clear text passwords, pre-calculated hash values, or passwords encrypted by a standard encryption algorithm (3DES_EDE/CBC/PKCS5Padding by default). The sections below provide general information about setting up your LDAP server or RDBMS to store the required information. Keep in mind that LDAP server uses different schemas and different administration tools, and you may need to refer to your LDAP server documentation for information about how to perform the steps below.

If you are using multiple LDAP servers to enable failover capabilities for the security providers, you must configure each LDAP server as described below.

Using Unencrypted Passwords

If you are using an RDBMS, or if your LDAP server's schema allows storing unencrypted passwords in the user's password attribute, no additional configuration is needed. The Digest Identity Asserter provider looks for unencrypted passwords in the password field by default.

If the schema does not allow unencrypted passwords in the password attribute, you have two options:

- Store the unencrypted password in an existing, unused credential attribute in the LDAP directory.
- Create a new credential attribute to store the unencrypted password.

See your LDAP server documentation for more information about credential attributes available in the schema. Regardless of which method you use, record the exact attribute name used to store unencrypted passwords. You must enter the name of this attribute when configuring the LDAP Digest Identity Asserter provider.

Using Precalculated Hash Values

If you want to use precalculated hash values, rather than unencrypted passwords, you can store the hash values in one of two places in your LDAP directory:

- In an existing, unused credential attribute.
- In a new credential attribute that you create for the hash value.

See your LDAP server documentation for more information using or creating new credential attributes.

For RDBMS stores, you can place the hash values in any column in your schema; you will define the SQL command used to obtain the hash values when configuring the RDBMS Identity Assertion Provider.

WebLogic SIP Server provides a simple method call to generate a hash of the A1 value from a given username, realm name, and unencrypted password. The built-in method is available at `com.bea.wcp.sip.util.DigestUtils.getHA1("username", "password", "realm-name")`, and is packaged in the *`WLSS_HOME`*`\telco\lib\wlss.jar` file. You can use also use 3rd-party utilities for generating the hash value, or create your own method using information from RFC 2617.

Note that you must also create the necessary infrastructure to update the stored hash value automatically when the user name, password, or realm name values change. Maintaining the password information in this manner is beyond the scope of this documentation.

Using Reverse-Encrypted Passwords

WebLogic SIP Server provides a utility to help you compute the Encryption Key, Encryption Init Vector, and Encrypted Passwords values used when you configure the Digest Authorization Identity Asserter provider. The utility is named `com.bea.wcp.sip.security.utils.JSafeEncryptionServiceImpl` and is packaged in the `wlss.jar` file in the *`WLSS_HOME`*`/telco/lib` directory.

To view usage instructions and syntax:

1. Add `wlss.jar` to your classpath:

   ```
   set CLASSPATH=%CLASSPATH%;c:\bea\wlss220\telco\lib\wlss.jar
   ```

2. Execute the utility without specifying options:

   ```
   java com.bea.wcp.sip.security.utils.JSafeEncryptionServiceImpl
   ```

Reconfigure the DefaultAuthenticator Provider

In most production environments you will use a separate LDAP provider for storing password information, and therefore the DefaultAuthenticator, which works against the embedded LDAP store, must not be required for authentication. Follow the instructions in this section to change the provider's control flag to "sufficient".

> **Note:** DefaultAuthenticator is set up as a required authentication provider by default. If the DefaultAuthentication provider, which works against the embedded LDAP store, is not used for authentication decisions, you must change the Control Flag to "SUFFICIENT".

To reconfigure the DefaultAuthenticator provider:

1. Log in to the Administration Console for the WebLogic SIP Server domain you want to configure.
2. In the left pane of the Console, expand the Security->Realms->myrealm->Providers->Authentication node.
3. Select the DefaultAuthenticator node in the left pane.
4. In the General tab on the right pane, change the Control Flag value to SUFFICIENT.
5. Click Apply to apply your changes.
6. Reboot the server to realize the changed security configuration.

Configure an Authenticator Provider

In addition to the Digest Identity Asserter providers, which only validate the client digest, you must configure an "authentication" provider, which checks for a user's existence and populates the user's group information. Follow the instructions in Configuring an LDAP Authentication Provider in the WebLogic Server 8.1 SP5 documentation set to create an LDAP authentication provider for your LDAP server. Use the information from Table 2-1, "Digest Identity Asserter Checklist," on page 2-6 to configure the provider.

If you do not require user existence checking or group population, then, in addition to a Digest Identity Asserter provider, you can configure and use the special "no-op" authentication provider, packaged by the name "IdentityAssertionAuthenticator." This provider is helpful to avoid an extra round-trip connection to the LDAP server. Note that the provider performs no user validation and should be used when group information is not required for users.

To configure the "no-op" authorization provider:

1. Log in to the Administration Console for the WebLogic SIP Server domain you want to configure.
2. In the left pane of the Console, expand the Security->Realms->myrealm->Providers->Authentication node.
3. Select the Authentication node in the left pane.
4. Click Configure a new Identity Assertion Authenticator...
5. Enter a name for the new provider, and set the Control Flag to SUFFICIENT.
6. Click Create to create the new provider.

Configure a New Digest Identity Asserter Provider

Follow these instructions in one of the sections below to create the Digest Identity Asserter provider and associate it with your LDAP server or RDBMS store:

- "Configure an LDAP Digest Identity Asserter Provider" on page 2-11
- "Configure an RDBMS Digest Identity Asserter Provider" on page 2-14

Configure an LDAP Digest Identity Asserter Provider

Follow these instructions to create a new LDAP Digest Identity Asserter Provider:

1. Log in to the Administration Console for the WebLogic SIP Server domain you want to configure.
2. In the left pane of the Console, expand the Security->Realms->myrealm->Providers->Authentication node.
3. Select the Authentication node in the left pane.
4. In the right pane of the Console, select Configure a new LDAP Digest Identity Asserter...
5. Enter a name for the new provider in the Name field, or accept the default, and click Create.
6. In the Active Types Chooser area, select both of the available types (WWW-Authenticate.DIGEST and Authorization.DIGEST) and use the arrow to move them to the Chosen column.
7. Click Apply to create the new provider.
8. Select the Details tab in the right pane to further configure the new provider.

9. In the Details tab, enter LDAP server and Digest authentication information into the fields as follows (use the information from Table 2-1):

   – **User From Name Filter**: Enter an LDAP search filter that WebLogic SIP Server will use to locate a given username. If you do not specify a value for this attribute, the server uses a default search filter based on the user schema.

   – **User Base DN**: Enter the base Distinguished Name (DN) of the tree in the LDAP directory that contains users.

   – **Credential Attribute Name**: Enter the credential attribute in the LDAP directory that stores either the pre-calculated hash value or the unencrypted password. By default WebLogic SIP Server uses the password attribute of the user entry. If you use a pre-calculated has value instead of an unencrypted password, or if the unencrypted password is stored in a different attribute, you must specify the correct attribute name here.

   – **Group Attribute Name**: Enter the group attribute in the LDAP directory that stores a the set of group names to which the user belongs.

   – **Password Encryption Type**: Select the format in which the password is stored: `PLAINTEXT`, `PRECALCULATEDHASH`, or `REVERSIBLEENCRYPTED`.

   – **Encryption Algorithm**: If you have stored encrypted passwords, enter the encryption algorithm that the Digest identity assertion provider will use for reverse encryption.

   – **Encryption Key** and **Confirm Encryption Key**: If you have stored encrypted passwords, enter the base-64 encrypted key used as part of the reverse encryption algorithm.

   – **Encryption Init Vector** and **Confirm Encryption Init Vector**: If you have stored encrypted passwords, enter the base-64 encrypted init vector string used as part of the reverse encryption algorithm.

   – **Digest Realm Name**: Enter the realm name to use for Digest authentication.

   – **Digest Algorithm**: Select either MD5 or MD5-sess as the algorithm to use for encrypting Digests.

   – **Digest Timeout**: This value defines the nonce timeout value for the digest challenge. If the nonce timeout is reached before the client responds, the client is re-challenged with a new nonce. By default, the Digest Timeout is set to 120 seconds.

   – **LDAP Connection Pool Size**: Enter the number of connections to use for connecting to the LDAP Server. This value should be equal to or less than the total number of execute threads configured for WebLogic SIP Server. To view the current number of configured threads, right-click on the WebLogic SIP Server name in the left pane of the

Administration Console and select View Execute Queues; the SIP Container uses the Thread Count value of the queue named sip.transport.Default. The default value of LDAP Connection Pool Size is 10.

Note that stale connections (for example, LDAP connections that are timed out by a load balancer) are automatically removed from the connection pool.

- **Host**: Enter the host name of the LDAP server to use for Digest verification. If you are using multiple LDAP servers for failover capabilities, enter the *hostname:port* value for each server separated by spaces. For example: `ldap1.mycompany.com:1050 ldap2.mycompany.com:1050`

  See Configuring Failover for LDAP Authentication Providers in the WebLogic Server 8.1 SP5 documentation for more information about configuring failover.
- **Port**: Enter the port number of the LDAP server.
- **SSL Enabled**: Select this option if you are using SSL to communicate unencrypted passwords between WebLogic SIP Server and the LDAP Server.
- **Principal**: Enter the name of a principal that WebLogic SIP Server uses to access the LDAP server.
- **Credential**: Enter the credential for the above principal name (generally a password).
- **Confirm Credential**: Re-enter the principal's credential.
- **Cache Enabled**: Specifies whether a cache should be used with the associated LDAP server.
- **Cache Size**: Specifies the size of the cache, in Kilobytes, used to store results from the LDAP server. By default the cache size is 32K.
- **Cache TTL**: Specifies the time-to-live (TTL) value, in seconds, for the LDAP cache. By default the TTL value is 60 seconds.
- **Results Time Limit**: Specifies the number of milliseconds to wait for LDAP results before timing out. Accept the default value of 0 to specify no time limit.
- **Connect Timeout**: Specifies the number of milliseconds to wait for an LDAP connection to be established. If the time is exceeded, the connection times out. The default value of 0 specifies no timeout value.
- **Parallel Connect Delay**: Specifies the number of seconds to delay before making concurrent connections to multiple, configured LDAP servers. If this value is set to 0, the provider connects to multiple servers in a serial fashion. The provider first tries to connect to the first configured LDAP server in the Host list. If that connection attempt fails, the provider tries the next configured server, and so on.

If this value is set to a non-zero value, the provider waits the specified number of seconds before spawning a new thread for an additional connection attempt. For example, if the value is set to 2, the provider first tries to connect to the first configured LDAP server in the Host list. After 2 seconds, if the connection has not yet been established, the provider spawns a new thread and tries to connect to the second server configured in the Host list, and so on for each configured LDAP server.

– **Connection Retry Limit**: Specifies the number of times the provider tries to reestablish a connection to an LDAP server if the LDAP server throws an exception while creating a connection.

– **Base64 Decoding Required**: This field is not applicable to the LDAP Digest Identity Asserter provider.

10. Click Apply to apply your changes.
11. Reboot the server to realize the changed security configuration.

Configure an RDBMS Digest Identity Asserter Provider

Follow these instructions to create a new RDBMS Digest Identity Asserter Provider:

1. Log in to the Administration Console for the WebLogic SIP Server domain you want to configure.
2. In the left pane of the Console, expand the Security->Realms->myrealm->Providers->Authentication node.
3. Select the Authentication node in the left pane.
4. In the right pane of the Console, select Configure a new DBMS Digest Identity Asserter...
5. Enter a name for the new provider in the Name field, or accept the default, and click Create.
6. In the Active Types Chooser area, select both of the available types (WWW-Authenticate.DIGEST and Authorization.DIGEST) and use the arrow to move them to the Chosen column.
7. Click Apply to create the new provider.
8. Select the Details tab in the right pane to further configure the new provider.
9. In the Details tab, enter RDBMS server and Digest authentication information into the fields as follows:

- **Data Source Name**: Enter the name of the JDBC DataSource used to access the password information.
- **SQLGet Users Password**: Enter the SQL statement used to obtain the password or hash value from the database. The SQL statement must return a single record result set.
- **SQLList Member Groups**: Enter a SQL statement to obtain the group information from a specified username. The username is supplied as a variable to the SQL statement, as in `SELECT G_NAME FROM groupmembers WHERE G_MEMBER = ?`.
- **Password Encryption Type**: Select the format in which the password is stored: `PLAINTEXT`, `PRECALCULATEDHASH`, or `REVERSIBLEENCRYPTED`.
- **Encryption Algorithm**: If you have stored encrypted passwords, enter the encryption algorithm that the Digest identity assertion provider will use for reverse encryption.
- **Encryption Key** and **Confirm Encryption Key**: If you have stored encrypted passwords, enter the base-64 encrypted key used as part of the reverse encryption algorithm.
- **Encryption Init Vector** and **Confirm Encryption Init Vector**: If you have stored encrypted passwords, enter the base-64 encrypted init vector string used as part of the reverse encryption algorithm.
- **Digest Realm Name**: Enter the realm name to use for Digest authentication.
- **Digest Algorithm**: Select either MD5 or MD5-sess as the algorithm to use for encrypting Digests.
- **Digest Timeout**: This value defines the nonce timeout value for the digest challenge. If the nonce timeout is reached before the client responds, the client is re-challenged with a new nonce. By default, the Digest Timeout is set to 120 seconds.
- **Base64 Decoding Required**: This field is not applicable to the RDBMS Digest Identity Asserter provider.

10. Click Apply to apply your changes.
11. Reboot the server to realize the changed security configuration.

Sample Digest Authentication Configurations

After configuring Digest authentication using the preceding steps, you can verify the configuration by examining the `config.xml` file for your domain. The sections that follow show sample excerpts from `config.xml`.

The Administration Console automatically encrypts credential information stored in `config.xml`. If you are editing `config.xml` manually, you can use the `weblogic.management.EncryptionHelper` utility to encrypt the credentials as described in the WebLogic Server 8.1 documentation.

Oracle Internet Directory Server

Listing 2-1 shows the security provider configuration in `config.xml` for a domain that uses LDAP Digest authentication. Note that although the IPlanetAuthenticator provider was selected, the provider is configured to use an Oracle Internet Directory Server.

Listing 2-1 Sample Security Provider Configuration for Oracle

```
<weblogic.security.providers.authentication.IPlanetAuthenticator
   ControlFlag="SUFFICIENT"
   Credential="{3DES}uQtI9MFcc7yR9hqgx39J2g=="
   DisplayName="OIDAuthenticator"
   GroupBaseDN="cn=Groups,dc=bea,dc=com" Host="lcw2k18.bea.com"
   Name="Security:Name=myrealmOIDAuthenticator" Port="389"
   Principal="cn=orcladmin" Realm="Security:Name=myrealm"
   UserBaseDN="cn=users,dc=bea,dc=com"
   UserFromNameFilter="(&amp;(cn=%u)(objectclass=person))"/>

<com.bea.wcp.sip.security.authentication.LdapDigestIdentityAsserter
   ActiveTypes="Authorization.DIGEST|WWW-Authenticate.DIGEST"
   Credential="{3DES}uQtI9MFcc7yR9hqgx39J2g=="
   CredentialAttributeName="middlename"
   DigestRealmName="wcp.bea.com" Host="lcw2k18.bea.com"
   Name="Security:Name=myrealmLdapDigestIdentityAsserter"
   Principal="cn=orcladmin" Realm="Security:Name=myrealm"
   UserBaseDN="cn=users,dc=bea,dc=com"
```

```
    UserFromNameFilter="(&amp;(cn=%u)(objectclass=person))"/>
```

WebLogic SIP Server Embedded LDAP

You can use WebLogic SIP Server’s embedded LDAP implementation to use Digest authentication in a test or demo environment. Because you cannot change the schema of the embedded LDAP store, you must store password information in the existing “description” field.

To use the embedded LDAP store for Digest authentication, follow the instructions in the sections that follow.

Store User Password Information in the Description Field

To create new users with password information in the existing “description” field:

1. Log in to the Administration Console for the WebLogic SIP Server domain you want to configure.
2. In the left pane of the Console, select the Security->Realms->myrealm->Users node.
3. Click Configure a new User...
4. Enter a name for the new user in the Name field.
5. Enter the Digest password information for the user in the Description field. The password information can be either the clear-text password, a pre-calculated hash value, or a reverse-encrypted password.
6. Enter an 8-character password in the Password and Confirm Password fields. You cannot proceed without adding a standard password entry.
7. Click Apply.

Set the Embedded LDAP Password

Follow these instructions to set the password for the embedded LDAP store to a known password. You will use this password when configuring the Digest Identity Asserter provider as described in “Configure an LDAP Digest Identity Asserter Provider” on page 2-11:

1. Log in to the Administration Console for the WebLogic SIP Server domain you want to configure.
2. In the left pane of the Console, select the Security node.
3. In the right pane, select Configuration->Embedded LDAP.

4. Enter the password you would like to use in the Credential and Confirm Credential fields.
5. Click Apply.
6. Reboot the server.

Configure the Digest Identity Asserter Provider

Listing 2-2 shows the security provider configuration in `config.xml` for a domain that uses LDAP implementation embedded in WebLogic SIP Server. Note that such a configuration is recommended only for testing or development purposes. Listing 2-2 highlights values that you must define when configuring the provider using the instructions in "Configure an LDAP Digest Identity Asserter Provider" on page 2-11.

Listing 2-2 Sample Security Provider Configuration with Embedded LDAP

```
<com.bea.wcp.sip.security.authentication.LdapDigestIdentityAsserter
   ActiveTypes="Authorization.DIGEST|WWW-Authenticate.DIGEST"
   Credential="{3DES}449oKgbalpo65cVYXzKhBg=="
   CredentialAttributeName="description"
   DigestRealmName="wcp.bea.com" Host="myserver.mycompany.com"
   Name="Security:Name=myrealmLdapDigestIdentityAsserter"
   Port="7001" Principal="cn=Admin"
   Realm="Security:Name=myrealm"
   UserBaseDN="ou=people, ou=myrealm, dc=mydomain" />
```

Configuring Client-Cert Authentication

The following sections describe how to configure WebLogic SIP Server to use Client-Cert authentication:

- "Overview of Client-Cert Authentication" on page 3-1
- "Configuring SSL and X509 for WebLogic SIP Server" on page 3-2
- "Configuring WebLogic SIP Server to Use WL-Proxy-Client-Cert" on page 3-6
- "Supporting Perimeter Authentication with a Custom IA Provider" on page 3-7

Overview of Client-Cert Authentication

Client-Cert authentication uses a certificate or other custom tokens in order to authenticate a user. The token is "mapped" to a user present in the WebLogic SIP Server security realm in which the Servlet is deployed. SIP Servlets that want to use Client-Cert authentication must set the `auth-method` element to `CLIENT-CERT` in their `sip.xml` deployment descriptor.

The token used for Client-Cert authentication can be obtained in several different ways:

- **X509 Certificate from SSL**—In the most common case, an X509 certificate is derived from a client token during a two-way SSL handshake between the client and the server. The SIP Servlet can view the resulting certificate in the `javax.servlet.request.X509Certificate` request attribute. This method for performing Client-Cert authentication is the most common and is described in the SIP Servlet specification (JSR-116). WebLogic SIP Server provides two security providers that can be used to validate the X509 certificate; see "Configuring SSL and X509 for WebLogic SIP Server" on page 3-2.

- **WL-Proxy-Client-Cert Header**—WebLogic SIP Server provides an alternate method for supplying a Client-Cert token that does not require a two-way SSL handshake between the client and server. Instead, the SSL handshake can be performed between a client and a proxy server or load balancer before reaching the destination WebLogic SIP Server. The proxy generates the resulting X509 certificate chain and encrypts it using base-64 encoding, and finally adds it to a special `WL-Proxy-Client-Cert` header in the SIP message. The server hosting the destination SIP Servlet then uses the `WL-Proxy-Client-Cert` header to obtain the certificate. The certificate is also made available by the container to Servlets via the `javax.servlet.request.X509Certificate` request attribute.

  To use this alternate method of supplying client tokens, you must configure WebLogic SIP Server to enable use of the `WL-Proxy-Client-Cert` header; see "Configuring WebLogic SIP Server to Use WL-Proxy-Client-Cert" on page 3-6. You must also configure an X509 Identity Asserter provider as described in "Configuring SSL and X509 for WebLogic SIP Server" on page 3-2.

SIP Servlets can also use the `CLIENT-CERT auth-method` to implement perimeter authentication. Perimeter authentication uses custom token names and values, along with a custom security provider, to authenticate clients. See "Supporting Perimeter Authentication with a Custom IA Provider" on page 3-7 for a summary of steps required to implement perimeter authentication.

Configuring SSL and X509 for WebLogic SIP Server

WebLogic SIP Server includes two separate Identity Assertion providers that can be used with X509 certificates. The LDAP X509 Identity Asserter provider receives an X509 certificate, looks up the LDAP object for the user associated with that certificate in a separate LDAP store, ensures that the certificate in the LDAP object matches the presented certificate, and then retrieves the name of the user from the LDAP object. The Default Identity Asserter provider maps the user according to its configuration, but does not validate the certificate.

With either provider, WebLogic SIP Server uses two-way SSL to verify the digital certificate supplied by the client. You must ensure that a SIPS transport (SSL) has been configured in order to use Client-Cert authentication. See Managing WebLogic SIP Server Network Resources in *Configuring and Managing WebLogic SIP Server* if you have not yet configured a secure transport.

See "Configuring the Default Identity Asserter" on page 3-3 to configure the Default Identity Asserter provider. In most production installations you will have a separate LDAP store and will need to configure the LDAP X509 Identity Asserter provider to use client-cert authentication; see "Configuring the LDAP X509 Identity Asserter" on page 3-4.

Configuring the Default Identity Asserter

The Default Identity Asserter can be configured to verify an X509 certificate passed to it by a client over a secure (SSL) connection. The Default Identity Asserter requires a separate user name mapper to map the associated client "certificate" to a user configured in the default security realm. You can use the default user name mapper installed with WebLogic SIP Server, or you can create a custom user name mapper class as described in Configuring a User Name Mapper in the WebLogic Server 8.1 Documentation.

Follow these instructions to configure the Default Identity Asserter:

1. Log in to the Administration Console for the WebLogic SIP Server domain you want to configure.
2. In the left pane of the Console, expand the Security->Realms->myrealm->Providers->Authentication node.
3. Select the Authentication node in the left pane.
4. In the right pane of the Console, select DefaultIdentityAsserter from the table of configured providers.
5. In the Types table, select X.509 and use the arrow to move this type to the Chosen column.
6. Select Base64 Decoding Required if the client token is being passed via two-way SSL or a `WL-Proxy-Client-Cert` header.
7. Click Apply to apply the change.
8. You can use either a custom Java class to map names in the X509 certificate to usernames in the built-in LDAP store, or you can use the default user name mapper. To specify a custom Java class to perform user name mapping:

   a. Enter the name of the custom class in the User Name Mapper Class Name field.

   b. Click Apply.

   To use the default user name mapper:

   a. Click the Details tab.

b. Select Use Default User Name Mapper

c. In the Default User Name Mapper Attribute Type field, select either CN-Common Name or E-Email Address depending on the user name attribute you have stored in the security realm.

d. In the Default User Name Mapper Attribute Delimiter field, accept the default delimiter of "@". This delimiter is used with the E-Email Address attribute type to extract the email portion from the client token. For example, a token of "joe@mycompany.com" would be mapped to a username "joe" configured in the default security realm.

e. Click Apply.

Configuring the LDAP X509 Identity Asserter

Follow these steps to create and configure the X509 Authentication Provider.

1. Log in to the Administration Console for the WebLogic SIP Server domain you want to configure.
2. In the left pane of the Console, expand the Security->Realms->myrealm->Providers->Authentication node.
3. Select the Authentication node in the left pane.
4. In the right pane of the Console, select Configure a new LDAP Digest p Asserter...
5. Enter a name for the new provider in the Name field, or accept the default, and click Create.
6. In the Active Types Chooser area, select X.509 and use the arrow to move this type to the Chosen column.
7. Click Apply to create the new provider.
8. Select the Details tab in the right pane to further configure the new provider.
9. In the Details tab, enter LDAP server information into the fields as follows:
   - **User Field Attributes**: Enter an LDAP search filter that WebLogic SIP Server will use to locate a given username. The filter is applied to LDAP objects beneath the base DN defined in the **Certificate Mapping** attribute described below.
   - **Username Attribute**: Enter the LDAP attribute that stores the user's name.
   - **Certificate Attribute**: Enter the LDAP attribute that stores the certificate for the user name.

- **Certificate Mapping**: Specify how a query string to construct the base LDAP DN used to locate the LDAP object for the user.
- **Base64 Decoding Required**: Select this field if the client token is being passed via two-way SSL or a `WL-Proxy-Client-Cert` header.
- **Host**: Enter the host name of the LDAP server to verify the incoming certificate. If you are using multiple LDAP servers for failover capabilities, enter the *hostname:port* value for each server separated by spaces. For example: `ldap1.mycompany.com:1050 ldap2.mycompany.com:1050`

  See Configuring Failover for LDAP Authentication Providers in the WebLogic Server 8.1 SP5 documentation for more information about configuring failover.
- **Port**: Enter the port number of the LDAP server.
- **SSL Enabled**: Select this option if you are using SSL to communicate unencrypted passwords between WebLogic SIP Server and the LDAP Server.
- **Principal**: Enter the name of a principal that WebLogic SIP Server uses to access the LDAP server.
- **Credential**: Enter the credential for the above principal name (generally a password).
- **Confirm Credential**: Re-enter the principal's credential.
- **Cache Enabled**: Specifies whether a cache should be used with the associated LDAP server.
- **Cache Size**: Specifies the size of the cache, in Kilobytes, used to store results from the LDAP server. By default the cache size is 32K.
- **Cache TTL**: Specifies the time-to-live (TTL) value, in seconds, for the LDAP cache. By default the TTL value is 60 seconds.
- **Follow Referrals**: Select this to specify that a search for a user or group within the LDAP X509 Identity Assertion provider should follow referrals to other LDAP servers or branches within the LDAP directory.
- **Bind Anonymously On Referrals**: By default, the LDAP X509 Identity Assertion provider uses the same DN and password used to connect to the LDAP server when following referrals during a search. If you want to connect as an anonymous user, check this box.
- **Results Time Limit**: Specifies the number of milliseconds to wait for LDAP results before timing out. Accept the default value of 0 to specify no time limit.

- **Connect Timeout**: Specifies the number of milliseconds to wait for an LDAP connection to be established. If the time is exceeded, the connection times out. The default value of 0 specifies no timeout value.
- **Parallel Connect Delay**: Specifies the number of seconds to delay before making concurrent connections to multiple, configured LDAP servers. If this value is set to 0, the provider connects to multiple servers in a serial fashion. The provider first tries to connect to the first configured LDAP server in the Host list. If that connection attempt fails, the provider tries the next configured server, and so on.

  If this value is set to a non-zero value, the provider waits the specified number of seconds before spawning a new thread for an additional connection attempt. For example, if the value is set to 2, the provider first tries to connect to the first configured LDAP server in the Host list. After 2 seconds, if the connection has not yet been established, the provider spawns a new thread and tries to connect to the second server configured in the Host list, and so on for each configured LDAP server.
- **Connection Retry Limit**: Specifies the number of times the provider tries to reestablish a connection to an LDAP server if the LDAP server throws an exception while creating a connection.

10. Click Apply to apply your changes.
11. Reboot the server to realize the changed security configuration.

Configuring WebLogic SIP Server to Use WL-Proxy-Client-Cert

In order for WebLogic SIP Server to use the `WL-Proxy-Client-Cert` header, a proxy server or load balancer must first transmit the X509 certificate for a client request, encrypt it using base-64 encoding, and then add the resulting token `WL-Proxy-Client-Cert` header in the SIP message. If your system is configured in this way, you can enable the local WebLogic SIP Server instance (or individual SIP Servlet instances) to examine the `WL-Proxy-Client-Cert` header for client tokens.

To configure the server instance to use the `WL-Proxy-Client-Cert` header:

1. Log in to the Administration Console for the WebLogic SIP Server domain you want to configure.
2. In the left pane, expand the Servers node and select a server to configure. (Alternately, expand the Clusters node and select a cluster name to configure `WL-Proxy-Client-Cert` use for the entire cluster.)

3. Select the Configuration->General tab in the right pane.
4. Select Client Cert Proxy Enabled.
5. Click Apply to apply your changes.
6. Follow the instructions under "Configuring SSL and X509 for WebLogic SIP Server" on page 3-2 to configure either the default identity asserter or the LDAP Identity Asserter provider to manage X509 certificates. Select the Base64 Decoding Required option to decode the token passed in the WL-Proxy-Client-Cert header.
7. Reboot the server to realize the changed configuration.

To enable `WL-Proxy-Client-Cert` header for an individual Web Application, set the `com.bea.wcp.clientCertProxyEnabled` context parameter to true in the `sip.xml` deployment descriptor.

Supporting Perimeter Authentication with a Custom IA Provider

With perimeter authentication, a system outside of WebLogic Server establishes trust via tokens. The system is generally comprised of an authentication agent that creates an artifact or token that must be presented to determine information about the authenticated user at a later time. The actual format of the token varies from vendor to vendor (for example, SAML or SPNEGO).

WebLogic SIP Server supports perimeter authentication through the use of an Identity Assertion provider designed to recognize one or more token formats. When the authentication type of a SIP Servlet is set to `CLIENT-CERT`, the SIP container in WebLogic SIP Server performs identity assertion on values from the request headers. If the header name matches the active token type for a configured provider, the value is passed to the provider for identity assertion.

The provider can then use a user name mapper to resolve the certificate to a user available in the security realm. The user corresponding to the Subject's Distinguished Name (SubjectDN) attribute in the client's digital certificate must be defined in the server's security realm; otherwise the client will not be allowed to access a protected WebLogic resource.

If you want to use custom tokens to pass client certificates for perimeter authentication, you must create and configure a custom Identity Assertion provider in place of the LDAP X509 or Default Identity Asserter providers described above. See Identity Assertion Providers in *Developing Security Providers for WebLogic Server* (WebLogic Server 8.1 Documentation) for information about creating providers for handling tokens passed with perimeter authentication.

Configuring P-Asserted-Identity Assertion

The following sections describe how the `P-Asserted-Identity` and `privacy` headers affect forwarding to trusted and non-trusted hosts, and how to configure a WebLogic SIP Server P-Asserted-Identity Asserter provider:

- "Understanding Trusted Host Forwarding with P-Asserted-Identity" on page 4-1
- "Overview of Strict and Non-Strict P-Asserted-Identity Asserter Providers" on page 4-5
- "Configuring a P-Asserted-Identity Assertion Provider" on page 4-6

Understanding Trusted Host Forwarding with P-Asserted-Identity

The `P-Asserted-Identity` header is honored only within a trusted domain. In a WebLogic SIP Server system, trusted domains are purely configuration-based. To enable use of the header, you must configure one of two available P-Asserted Identity Assertion providers as described in "Configuring a P-Asserted-Identity Assertion Provider" on page 4-6. The `P-Asserted-Identity` assertion providers expose the trusted domain configuration for `P-Asserted-Identity` headers. If you do not configure a provider, the header considers no IP addresses as being "trusted."

When WebLogic SIP Server receives a message having the `P-Asserted-Identity` header from a trusted host configured with the provider, it logs in the user specified in the header to determine group membership and other privileges. The value contained in the `P-Asserted-Identity` header must be a SIP address (for example, `sipuser@bea.com`). By default, WebLogic SIP

Server removes the domain portion of the address (@bea.com) and uses the remainder as the user name. If you must support overlapping usernames from different names (for example, sipuser@bea.com and sipuser@cea.com), you can create and use a custom user-name mapper to process the header contents into a unique username (for example, sipsuser_b and sipuser_c). Using a custom user name mapper also enables you to support WebLogic user names that contain an "@" character, such as @bea.com.

The presence of a P-Asserted-Identity header combined with the Privacy header also determines the way in which WebLogic SIP Server proxies incoming requests. Figure 4-1 describes how incoming SIP requests are managed in relation to the P-Asserted-Identity header.

Figure 4-1 Managing Inbound Requests Having P-Asserted-Identity and Privacy Headers

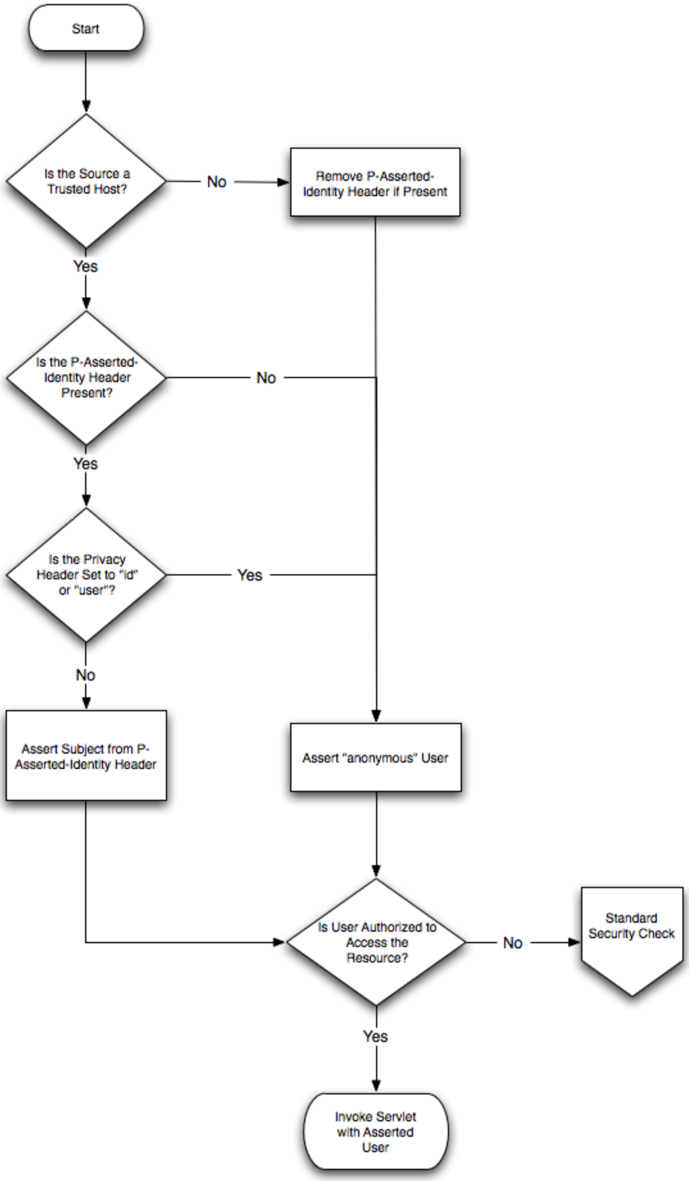
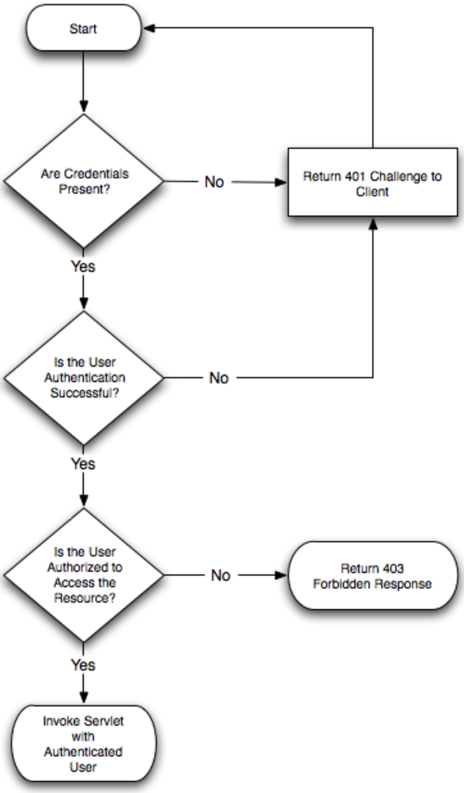

Start
Is the Source a Trusted Host?
No
Remove P-Asserted-Identity Header if Present
Yes
Is the P-Asserted-Identity Header Present?
No
Yes
Is the Privacy Header Set to "id" or "user"?
Yes
No
Assert Subject from P-Asserted-Identity Header
Assert "anonymous" User
Is User Authorized to Access the Resource?
No
Standard Security Check
Yes
Invoke Servlet with Asserted User

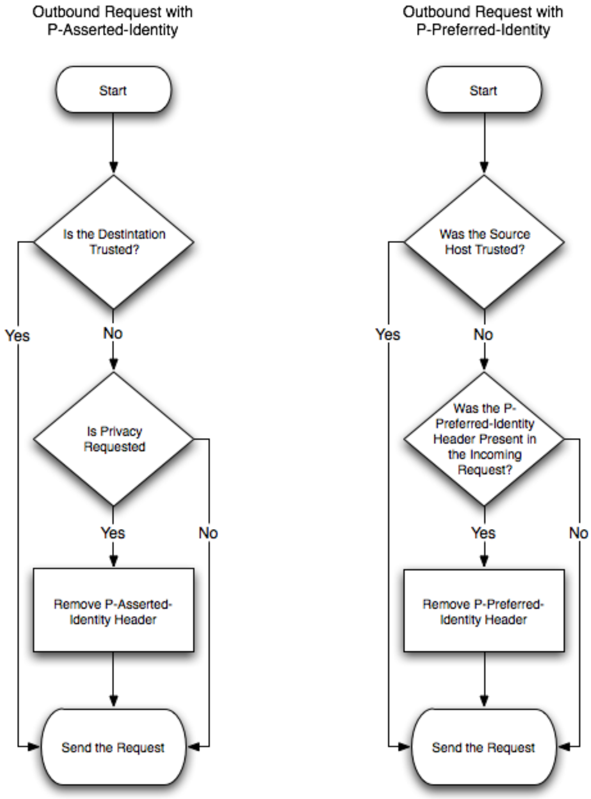
Figure 4-2 describes the standard security check procedure that WebLogic SIP Server uses when an asserted user name is not authorized to access a requested resource. The standard security check is performed according to the `auth-method` defined in the `login-config` element of the `sip.xml` descriptor for the current application.

Figure 4-2 Standard Security Check Procedure


Start
Are Credentials Present?
No
Return 401 Challenge to Client
Yes
Is the User Authentication Successful?
No
Yes
Is the User Authorized to Access the Resource?
No
Return 403 Forbidden Response
Yes
Invoke Servlet with Authenticated User

The presence of a `P-Asserted-Identity` header or a `P-Preferred-Identity` header also affects the processing of outbound SIP requests. Figure 4-3 describes the behavior.

Figure 4-3 Managing Outbound Requests Having P-Asserted-Identity or P-Preferred Identity


Outbound Request with P-Asserted-Identity
Start
Is the Destintation Trusted?
Yes
No
Is Privacy Requested
Yes
No
Remove P-Asserted-Identity Header
Send the Request
Outbound Request with P-Preferred-Identity
Start
Was the Source Host Trusted?
Yes
No
Was the P-Preferred-Identity Header Present in the Incoming Request?
Yes
No
Remove P-Preferred-Identity Header
Send the Request

Overview of Strict and Non-Strict P-Asserted-Identity Asserter Providers

If the contents of a P-Asserted-Identity header are invalid, or if the header is received from a non-trusted host, then the security provider returns an “anonymous” user to the SIP Servlet container. If you configured the **PAsserted Identity Strict Asserter** provider, an exception is also thrown so that you can audit the substitution of the anonymous user. (If you configured the basic **PAsserted Identity Asserter** provider, no exception is thrown.)

With either provider, if identity assertion fails and the requested resource is protected (the request matches a `security-constraint` defined in `sip.xml`), the SIP container uses the `auth-method` defined in the `sip.xml` deployment descriptor to challenge the end user. For example, digest authentication may be used if the Servlet specifies the digest authentication method.

If the requested resource is not protected, the anonymous user is simply passed to the SIP Servlet without authorization. Because the 3GPP TS 24.229 specification recommends forced authorization even when a resource is unrestricted (and privacy is not requested), you should use declarative security to protect all of a SIP Servlet's resources to remain compliant with the specification. See Securing SIP Servlet Resources in Developing Applications with WebLogic SIP Server for more information.

If authorization of the anonymous user fails, WebLogic SIP Server then forces authentication by challenging the user.

Configuring a P-Asserted-Identity Assertion Provider

Follow these steps to configure a security provider used to support the `P-Asserted-Identity` header. Note that one of two providers can be selected, as described in "Overview of Strict and Non-Strict P-Asserted-Identity Asserter Providers" on page 4-5.

In addition to configuring one of the above providers, configure a secondary, "fallback" login method (for example, using DIGEST or CLIENT-CERT authentication).

To configure a `P-Asserted-Identity` provider:

1. Log in to the Administration Console for the WebLogic SIP Server domain you want to configure.
2. In the left pane of the Console, select the Security->Realms->myrealm->Providers->Authentication node.
3. In the right pane of the Console, select one of the following options:
   - Configure a new PAsserted Identity Asserter...—Select this option to configure a provider that does not throw an exception when the `P-Asserted-Identity` header is invalid or is received from a non-trusted host and an anonymous user is substituted.
   - Configure a new PAsserted Identity Strict Asserter...—Select this option to configure a provider that throws an exception when the `P-Asserted-Identity` header is invalid or is received from a non-trusted host and an anonymous user is substituted.

See "Overview of Strict and Non-Strict P-Asserted-Identity Asserter Providers" on page 4-5 for more information.

4. Enter a name for the new provider and click Create.
5. Select the Details tab to display the new provider's configuration.
6. Fill in the fields of the Details tab as follows:
   - **Trusted Hosts**: Enter one or more host names that the provider will treat as trusted hosts. You can enter a list of IP addresses or DNS names, and wildcards are supported.

     Note: The provider *does not use* trusted hosts configured in the `sipserver.xml` file (see sip-security in *Configuring and Managing WebLogic SIP Server*.)
   - **User Name Mapper Class Name**: Enter the name of a custom Java class used to map user names in the `P-Asserted-Identity` header to user names in the default security realm. A custom user name mapper is generally used if user names are received from two or more different domains. In this case additional logic may be required to map usernames received from each domain. A custom user name mapper class is required if you want to map usernames in the `P-Asserted-Identity` header to WebLogic usernames. See Configuring a User Name Mapper in the WebLogic Server 8.1 Documentation for more information.

     Alternately, leave this field blank to use the default user name mapper. The default mapper simply discards the domain name and takes the resulting user name without applying any additional logic.
   - **Base64Decoding Required**: This field is not used by the provider.
7. Click Apply.

Configuring 3GPP HTTP Authentication Providers

The following sections describe how to configure WebLogic SIP Server to handle the X-3GPP-Asserted-Identity header for HTTP authentication:

- "Overview" on page 5-1
- "Configuring a X-3GPP-Asserted-Identity Provider" on page 5-2

Overview

In order to function as an Application Server in an IMS network, WebLogic SIP Server supports handling the X-3GPP-Asserted-Identity header as specified in 3GPP TS 33.222 Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS). WebLogic SIP Server provides this support via a configured security provider, X3gppAssertedIdentityAsserter or X3gppAssertedIdentityStrictAsserter. The providers use the same authentication process, but the "strict" assertion provider also throws an exception when the header is received from a non-trusted host (which enables you to audit asserted identity requests from non-trusted hosts).

The X-3GPP-Asserted-Identity header functions for HTTP requests in the same manner that the P-Asserted-Identity header functions for SIP requests. When the container receives an incoming HTTP requesting having a X-3GPP-Asserted-Identity header, it first verifies that the request was received from a trusted host. If the host was trusted, the container asserts the user's identity using the information in the header, authenticates the user, and logs the user in if that user is authorized to access the requested resource. (If a request comes from a non-trusted host, the container simply ignores the header.)

The `X-3GPP-Asserted-Identity` header may contain multiple names in a list (for example, user1@bea.com, user2@bea.com). When configured with the default user name mapper class, the WebLogic SIP Server providers remove the domain portion of the addresses (`@bea.com`) and use the remainder as the user name. The default user name mapper always chooses the first username in the list and uses it for asserting the identity. This behavior can be changed by creating and configuring a custom user name mapper class. For example, if you must support overlapping usernames from different names (for example, `sipuser@bea.com` and `sipuser@cea.com`), a custom user-name mapper might process the header contents into a unique username (for example, `sipsuser_b` and `sipuser_c`). Using a custom user name mapper also enables you to support WebLogic user names that contain an "@" character, such as `@bea.com`.

In order for SIP Servlets to support authentication with the `X-3GPP-Asserted-Identity` header, the `auth-method` element must be set to `CLIENT-CERT` in the `web.xml` deployment descriptor. See web.xml Deployment Descriptor Elements in the WebLogic Server 8.1 Documentation for more information.

Configuring a X-3GPP-Asserted-Identity Provider

Follow these steps to configure a security provider used to support the `X-3GPP-Asserted-Identity` header in HTTP requests. Note that one of two providers can be selected, as described in the "Overview" on page 5-1:

1. Log in to the Administration Console for the WebLogic SIP Server domain you want to configure.
2. In the left pane of the Console, select the Security->Realms->myrealm->Providers->Authentication node.
3. In the right pane of the Console, select one of the following options:
   - Configure a new X3gpp Asserted Identity Asserter...—Select this option to configure a provider that does not throw an exception when the header is invalid or is received from a non-trusted host.
   - Configure a new X3gpp Asserted Identity Strict Asserter...—Select this option to configure a provider that throws an exception when the header is received from a non-trusted host and is therefore ignored.

   See "Overview" on page 5-1 for more information.

4. Enter a name for the new provider and click Create.
5. In the Active Types Chooser list, select the X-3GPP-Asserted-Identity type and use the arrow to move it to the Chosen column.
6. Click Apply.
7. Select the Details tab to display the new provider's configuration.
8. Fill in the fields of the Details tab as follows:
   - **Trusted Hosts**: Enter one or more host names that the provider will treat as trusted hosts. Note that the provider *does not use* trusted hosts configured in the `sipserver.xml` file (see sip-security in *Configuring and Managing WebLogic SIP Server*.) You can enter a list of IP addresses or DNS names, and wildcards are supported.
   - **User Name Mapper Class Name**: Enter the name of a custom Java class used to map user names in the `X-3GPP-Asserted-Identity` header to user names in the default security realm. A custom user name mapper is generally used if user names are received from two or more different domains. In this case additional logic may be required to map user names received from each domain. A custom user name mapper class is required if you want to map usernames to WebLogic usernames, or if you want to logically process multiple usernames specified in the `X-3GPP-Asserted-Identity` header (rather than using only the first username). See Configuring a User Name Mapper in the WebLogic Server 8.1 Documentation for more information.

     Alternately, leave this field blank to use the default user name mapper. The default mapper simply discards the domain name and takes the first resulting user name to assert the identity. For example, the default user name mapper takes the following header:

     ```
     X-3GPP-Asserted-Identity: "user1@bea.com", "user2@bea.com"
     ```

     and asserts the identity "user1."
   - **Base64Decoding Required**: This field is not used by the provider.
9. Click Apply.