Oracle® Communications Converged Application Server

Security Guide

Release 8.0
F43739-02
June 2023

ORACLE

Oracle Communications Converged Application Server Security Guide, Release 8.0

F43739-02

Contents

Preface

This document describes security features and configuration for Oracle Communications Converged Application Server.

Audience

This document is intended for administrators who configure security for Converged Application Server.

My Oracle Support

My Oracle Support (https://support.oracle.com) is your initial point of contact for all product support and training needs. A representative at Customer Access Support (CAS) can assist you with My Oracle Support registration.

Call the CAS main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at http://www.oracle.com/us/support/contact/index.html. When calling, make the selections in the sequence shown below on the Support telephone menu:

1. Select `2` for New Service Request.
2. Select `3` for Hardware, Networking, and Solaris Operating System Support.
3. Select one of the following options:
   - For technical issues such as creating a new Service Request (SR), select `1`.
   - For non-technical issues such as registration or assistance with My Oracle Support, select `2`.

You are connected to a live agent who can assist you with My Oracle Support registration and opening a support ticket.

My Oracle Support is available 24 hours a day, 7 days a week, 365 days a year.

Emergency Response

In the event of a critical service situation, emergency response is offered by the Customer Access Support (CAS) main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at http://www.oracle.com/us/support/contact/index.html. The emergency response provides immediate coverage, automatic escalation, and other features to ensure that the critical situation is resolved as rapidly as possible.

A critical situation is defined as a problem with the installed equipment that severely affects service, traffic, or maintenance capabilities, and requires immediate corrective action. Critical situations affect service and/or system operation resulting in one or several of these situations:

- A total system failure that results in loss of all transaction processing capability
- Significant reduction in system capacity or traffic handling capability
- Loss of the system's ability to perform automatic system reconfiguration
- Inability to restart a processor or the system
- Corruption of system databases that requires service affecting corrective actions
- Loss of access for maintenance or recovery operations
- Loss of the system ability to provide any required critical or major trouble notification

Any other problem severely affecting service, capacity/traffic, billing, and maintenance capabilities may be defined as critical by prior discussion and agreement with Oracle.

Locate Product Documentation on the Oracle Help Center Site

Oracle Communications customer documentation is available on the web at the Oracle Help Center (OHC) site, http://docs.oracle.com. You do not have to register to access these documents. Viewing these files requires Adobe Acrobat Reader, which can be downloaded at http://www.adobe.com.

1. Access the Oracle Help Center site at http://docs.oracle.com.
2. Click **Industries**.
3. Under the Oracle Communications sub-header, click the **Oracle Communications documentation** link.
   The Communications Documentation page appears. Most products covered by these documentation sets appear under the headings "Network Session Delivery and Control Infrastructure" or "Platforms."
4. Click on your Product and then Release Number.
   A list of the entire documentation set for the selected product and release appears.
5. To download a file to your location, right-click the **PDF** link, select **Save target as** (or similar command based on your browser), and save to a local folder.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Revision History

Table 1 Revision History

Date	Revision
December 2021	• Initial release
June 2023	• Updates OCCAS variable

1

Converged Application Server Security Overview

This chapter describes the Converged Application Server security features.

Basic Security Considerations

The following principles are fundamental to using any application securely:

- **Keep software up to date**. This includes the latest product release and any patches that apply to it.
- **Limit privileges as much as possible**. Users should be given only the access necessary to perform their work. User privileges should be reviewed periodically to determine relevance to current work requirements.
- **Monitor system activity**. Establish who should access which system components, and how often, and monitor those components.
- **Install software securely**. For example, use firewalls, secure protocols such as SSL and secure passwords.
- **Learn about and use the Converged Application Server security features**. See Converged Application Server Security Concepts for additional overview information on Converged Application Server security features.
- **Use secure development practices**. For example, take advantage of existing database security functionality instead of creating your own application security.
- **Keep up to date on security information**. Oracle regularly issues security-related patch updates and security alerts. You must install all security patches as soon as possible.

> **Note:**
>
> The Converged Application Server is built upon both Oracle WebLogic Server and Oracle Coherence. Customers must stay up-to-date and apply all security patches for both Oracle WebLogic Server and Oracle Coherence.

See the “Critical Patch Updates and Security Alerts" Web site:
https://www.oracle.com/security-alerts/

Overview of Converged Application Server Security

Converged Application Server relies on the underlying security features of the Oracle WebLogic platform. As such, Converged Application Server benefits from the security features of the underlying WebLogic platform, including security realms, security monitoring features, and more.

See "Oracle Security Documentation" for information about securing the WebLogic platform.

Additional security features applicable to Converged Application Server include:

- Network channel-based security in the form of support for HTTPS and SIPS. See *Oracle Communications Converged Application Server Administrator's Guide* for more information on network channel security.
- Flexible client authentication mechanisms, including identity assertions by security providers, client certificate authentication, and digest-based authentication.

This document describes the security features specific for Converged Application Server. For WebLogic information, including information about performing a secure installation and implementing application security, see your Oracle WebLogic Server documentation.

Understanding the Converged Application Server Environment

When planning your Converged Application Server implementation, consider the following:

- Which resources need to be protected?
  - You need to protect customer data, such as credit-card numbers.
  - You need to protect internal data, such as proprietary source code.
  - You need to protect system components from being disabled by external attacks or intentional system overloads.
- Who are you protecting data from?

  For example, you need to protect your subscribers' data from other subscribers, but someone in your organization might need to access that data to manage it. You can analyze your workflows to determine who needs access to the data; for example, it is possible that a system administrator can manage your system components without needing to access the system data.
- What will happen if protections on a strategic resources fail?

  In some cases, a fault in your security scheme is nothing more than an inconvenience. In other cases, a fault might cause great damage to you or your customers. Understanding the security ramifications of each resource will help you protect it properly.

Oracle Security Documentation

To implement security, you configure Converged Application Server security features as well as those in the products on which it relies.

See the following documents for more information:

- *Oracle Fusion Middleware Securing Oracle WebLogic Server* in the Oracle WebLogic Server documentation.
- *Oracle Fusion Middleware Application Security Guide* in the Oracle WebLogic Server documentation
- *Oracle Communications Converged Application Server Administrator's Guide*.
- *Oracle Communications Converged Application Server Developer's Guide*.

Common Security Configuration Tasks

Table 1-1 lists Converged Application Server configuration tasks and provides links to additional information.

Table 1-1 Security Configuration Tasks

Task	Document Reference
Configure a DNS resolver that supports DNSSEC. Converged Application Server supports a number of SIP RFCs that use DNS, and Converged Application Server accesses DNS a lot. DNSSEC is important to prevent malicious entities from spoofing DNS entries and cause issues to the deployment.	See the IETF specifications dealing with DNS security.
Understanding the Digest identity assertion providers Configuring LDAP Digest authentication Configuring Digest authentication with an RDBMS	See "Configuring Digest Authentication".
Understanding client-cert authentication solutions Delivering X509 certificates over 2-way SSL Developing a Perimeter authentication solution Using the Converged Application Server `WL_Client_Cert` header to deliver X509 certificates	See "Configuring Client-Cert Authentication".
Understand forwarding rules for SIP messages having the `P-Asserted-Identity` header Configuring `P-Asserted-Identity` providers	See "Overview of SIP Servlet Identity Assertion Mechanisms".
Defining security constraints for a SIP Servlet Mapping SIP Servlet roles to Converged Application Server roles and principals Debugging SIP Servlet security constraints	See "Securing SIP Servlet Resources" in *Converged Application Server Developer's Guide*
Configuring trusted hosts	See information on the sip-security setting in sipserver.xml, as described in *Oracle Communications Converged Application Server Administrator's Guide*

2

Converged Application Server Security Concepts

This chapter describes the Converged Application Serveri security features.

About Application Security

The SIP Servlet Specification (JSR 359) describes programmatic security considerations applicable to SIP Servlets. SIP Servlet security features are similar to those applicable to HTTP Servlets. Security features provided by the underlying WebLogic server platform can be applied to both types of servlets. You can find additional information about HTTP Servlet security by referring to the Oracle WebLogic Server documentation.

For SIP servlet security programming considerations specific for Converged Application Server development, see information about securing SIP servlet resources in the *Oracle Communications Converged Application Server Developer's Guide*.

Authentication for SIP Servlets

Converged Application Server users must be authenticated when they request access to a protected resource, such as a protected method within a deployed SIP Servlet. Converged Application 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. This technique is described in "Configuring Digest Authentication". To authenticate over HTTP, application developers must provide their own implementations.
- **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".
- **BASIC authentication** uses the `Authorization` SIP header to transmit the username and password to SIP Servlets. BASIC authentication is deprecated in RFC 3261 and is not recommended for production systems. This document does not provide configuration instructions for using BASIC authentication.

Different SIP Servlets deployed on Converged Application 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 *Converged Application Server Developer's Guide* for information about securing resources and mapping roles in the SIP Servlet deployment descriptor.

Authentication Providers

Converged Application 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 use 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.

See "Configuring Digest Authentication" or "Configuring Client-Cert Authentication" for more information about configuring providers.

Overriding Authentication with Trusted Hosts

Converged Application Server also enables you to designate trusted hosts for your system. Trusted hosts are hosts for which Converged Application Server performs no authentication. If the server receives a SIP message having a destination address that matches a configured trusted host name, the message is delivered without Authentication. See engine tier configuration reference information (sipserver.xml) in the *Oracle Communications Converged Application Server Administrator's Guide* for more information.

Identity Assertion Support

Converged Application Server supports the `P-Asserted-Identity` SIP header as described in RFC 3325. This functionality automatically logs in using credentials specified in the `P-Asserted-Identity` header when they are received from 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.

Converged Application Server also supports identity assertion using the `Identity` and `Identity-Info` headers as described in RFC 4474.

Both identity assertion mechanisms require that you configure an appropriate security provider with Converged Application Server. See "Overview of SIP Servlet Identity Assertion Mechanisms" 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. Converged Application 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 *Converged Application Server Developer's Guide* for more information.

Security Event Auditing

Converged Application Server includes an auditing provider that you can configure to monitor authentication events in the security realm. See *Securing Oracle WebLogic Server* in your Oracle WebLogic Server documentation for more information.

3
Configuring Digest Authentication

This chapter describes how to configure the Converged Application Server to use Digest authentication with a supported LDAP server or RDBMS.

Overview of Digest Authentication

The following sections provide a basic overview of Digest authentication, and describe Digest authentication support and configuration in Converged Application Server.

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, realm, 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 3-1 for a more detailed explanation of the challenge-response mechanism for a typical request.

Digest Authentication Support in Converged Application Server

Converged Application Server 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").

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`.

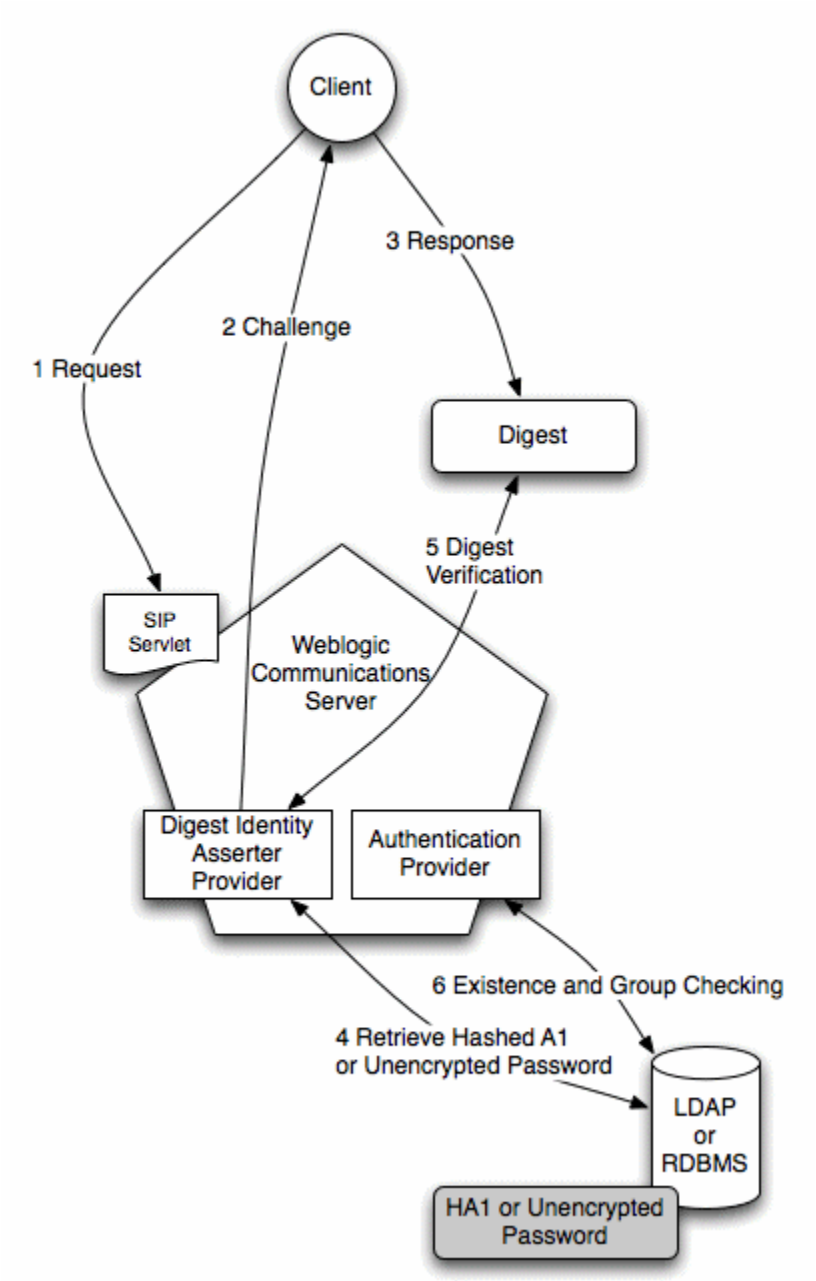
The Digest Identity Asserter provider requires that user credentials be stored in an LDAP server or RDBMS in one of the following ways:

- **Unencrypted (clear text) passwords.** The simplest configuration stores users' unencrypted passwords in a store. If you choose this method, Oracle 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 for more information.
- **Reverse-Encrypted Passwords.** Converged Application 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.
- **A pre-calculated hash of each password, username, and realm.** If storing unencrypted or reverse-encrypted 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.

> **Note:**
>
> 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 3-1 Digest Authentication in Converged Application Server



1. The client makes an unauthorized request for a protected application resource. (You protect SIP Servlet resources by specifying the `@SipSecurity` annotation. See RFC 359, sections 22.3.10.)
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.

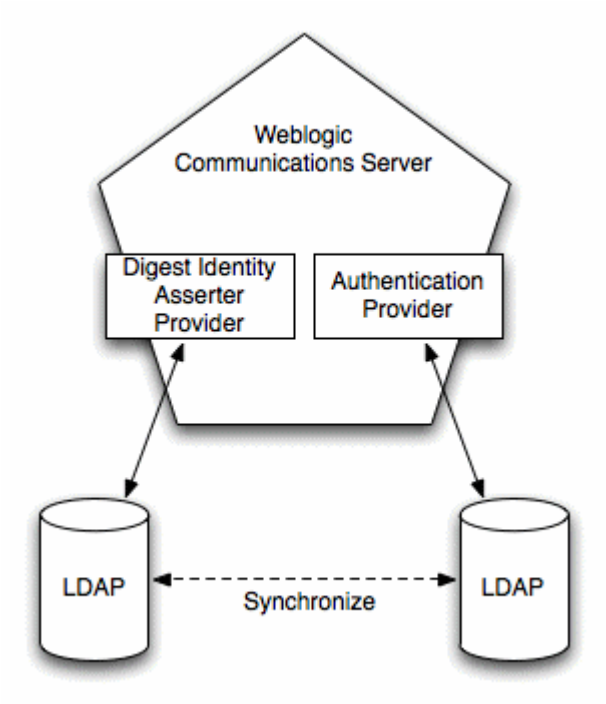
> **Note:**
>
> If you do not require user existence checking or group population, you can use the special "no-op" Identity Assertion Authenticator to avoid an extra connection to the LDAP Server; see "Configure an Authenticator Provider" for more information.

After authentication is complete, the SIP Servlet container performs an authorization check for the logged in `javax.security.auth.Subject` against the `@SipConstraint` annotation inside the `@SipSecurity` annotation. See JSR 359, sections 20.8, 22.3.10.1 and 22.3.10.2.

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. Maintaining LDAP stores in this manner is beyond the scope of this documentation.

Figure 3-2 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 Converged Application Server configuration.

Table 3-1 summarizes all of the information you will need in order to fully configure your LDAP server for Digest authentication with Converged Application Server.

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".

Table 3-1 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
Principal	A Distinguished Name (DN) that Converged Application Server can use to connect to the LDAP Server.	cn=ldapadminuser
Credential	A credential for the above principal name (generally a password).	ldapadminuserpassword

Table 3-1 (Cont.) Digest Identity Asserter Checklist

Item	Description	Sample Value
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 Converged Application 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=person))
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".	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. Converged Application 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 2 minutes.	2

Steps for Configuring Digest Authentication

Follow these steps to configure Digest authentication with Converged Application Server:

1. Configure the LDAP Server or RDBMS.
2. Reconfigure the DefaultAuthenticator Provider.

> **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.
4. Configure a New Digest Identity Asserter Provider.

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. Be aware 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.

Converged Application Server provides a simple utility (PreCalculatedHash) to generate a hash of the A1 value from a given username, realm name, and unencrypted password. The utility is packaged as `com.bea.wcp.sip.security.utils.PreCalculatedHash`. Use the syntax:

```
java com.bea.wcp.sip.security.utils.PreCalculatedHash user_name realm_name password
```

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

Converged Application 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.JSafeEncryptionUtil` and is packaged in the `wlss.jar` file in the `WebLogic_Home/sip/server/lib` directory, where `WebLogic_Home` is the directory where the WebLogic Server component of Converged Application Server is installed.

To view usage instructions and syntax:

1. Add `wlss.jar` to your classpath. The default path is:

   ```
   export CLASSPATH=$CLASSPATH:~/oracle/Middleware/Oracle_Home/wlserver/sip/
   server/lib/wlss.jar
   ```

2. Execute the utility without specifying options:

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

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 Converged Application Server domain you want to configure.
2. In the left pane of the Console, select the **Security Realms** node.
3. Select the name of your security realm in the Realms table in the right pane of the Console.
4. Select the **Providers** tab, then select the **Authentication** subtab.
5. Select the **DefaultAuthenticator** provider.
6. Select the **Configuration** tab and then select the **Common** subtab.
7. Choose SUFFICIENT from the **Control Flag** drop down list.

8. Click **Save** to save your changes.
9. Restart the server.

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 provided in *Oracle Fusion Middleware Securing Oracle WebLogic Server* to create an LDAP authentication provider for your LDAP server. Use the information from Table 3-1 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 Converged Application Server domain you want to configure.
2. In the left pane of the Console, select the **Security Realms** node.
3. Select the name of your security realm in the Realms table in the right pane of the Console.
4. Select **Providers**, then select the **Authentication** subtab.
5. Click **New**.
6. Enter a name for the new provider, and select *IdentityAssertionAuthenticator* from the **Type** drop down list.
7. Click **OK**.
8. Select the name of the new provider from the Authentication Providers table.
9. Select the **Configuration** tab and then select the **Common** subtab.
10. Choose SUFFICIENT from the **Control Flag** drop down list.
11. Click **Save** to save your changes.
12. Restart the server.

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
- Configure an RDBMS Digest Identity Asserter Provider

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 Converged Application Server domain you want to configure.
2. In the left pane of the Console, select the **Security Realms** node.
3. Select the name of your security realm in the Realms table in the right pane of the Console.
4. Select **Providers**, then select the **Authentication** subtab.
5. Click **New**.
6. Enter a name for the new provider, and select **LdapDigestIdentityAsserter** from the **Type** drop down list.
7. Click **OK**.
8. Select the name of the new provider from the list of providers.
9. Select **Configuration**, then select the **Provider Specific** subtab in the right pane.
10. On the configuration page, enter LDAP server and Digest authentication information into the fields as follows (use the information from Table 3-1):
    - **User From Name Filter**: Enter an LDAP search filter that Converged Application 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 (for example, cn=Users,dc=example,dc=com).
    - **Credential Attribute Name**: Enter the credential attribute in the LDAP directory that stores either the pre-calculated hash value or the unencrypted password (for example, authpassword;wlss). By default Converged Application 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 **Please type again to confirm**: If you have stored encrypted passwords, enter the base-64 encrypted key used as part of the reverse encryption algorithm.
    - **Encryption Init Vector** and **Please type again to confirm**: 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 (for example, example.com).
    - **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.
- **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 *host_name:port* value for each server separated by spaces. For example:
  `ldap1.mycompany.com:1050 ldap2.mycompany.com:1050`

  See *Oracle Fusion Middleware Securing Oracle WebLogic Server* 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 Converged Application Server and the LDAP Server.
- **Principal**: Enter the name of a principal that Converged Application Server uses to access the LDAP server (for example, orclApplicationCommonName=WLSSInstance1,cn=WLSS,cn=Products,cn=OracleContext,dc=example,dc=com).
- **Credential** and **Please type again to confirm**: Enter the credential for the above principal name (generally a password).
- **OIDSupportEnabled**: Select this checkbox if you are using Oracle Internet Directory as your LDAP provider. This checkbox is necessary when using a precalculated hash value because Oracle Internet Directory prefixes the hash value with {SASL/MD5} as described in RFC 2307. Other LDAP providers may omit the prefix.

11. Click **Save** to save your changes.
12. Select the Performance tab in the right pane.
13. On the Performance page, enter the caching and connection information into the fields as follows:
    - **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 Converged Application Server. To view the current number of configured threads, right-click on the Converged Application 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.
    - **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.

14. Click **Save** to save your changes.
15. Restart the server.

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 Converged Application Server domain you want to configure.
2. In the left pane of the Console, select the **Security Realms** node.
3. Select the name of your security realm in the Realms table in the right pane of the Console.
4. Select **Providers**, then select the **Authentication** subtab.
5. Click **New**.
6. Enter a name for the new provider, and select *DBMSDigestIdentityAsserter* as the type.
7. Click **OK**.
8. Select the name of the new provider from the Authentication Providers table.
9. Select **Configuration**, then select the **Provider Specific** subtab in the right pane.
10. In the configuration 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 **Please type again to confirm**: If you have stored encrypted passwords, enter the base-64 encrypted key used as part of the reverse encryption algorithm.
- **Encryption Init Vector** and **Please type again to confirm**: 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.

11. Click **Save** to save your changes.
12. Restart the server.

Sample Digest Authentication Configuration Using Embedded LDAP

You can use Converged Application Server's embedded LDAP implementation for 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 Converged Application Server domain you want to configure.
2. In the left pane of the Console, select the **Security Realms** node.
3. Select the name of your security realm in the Realms table in the right pane of the Console.
4. Select the **Users and Groups** tab, then select the **Users** subtab.
5. Click **New**.
6. Enter a name for the new user in the **Name** field.
7. 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.

8. Enter an 8-character password in the **Password** and **Confirm Password** fields. You cannot proceed without adding a standard password entry.
9. Click **OK**.

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":

1. Log in to the Administration Console for the Converged Application Server domain you want to configure.
2. In the left pane, click the name of the domain you are configuring.
3. Select the **Security** tab in the right pane, then select the **Embedded LDAP** subtab.
4. Enter the password you would like to use in the **Credential** and **Confirm Credential** fields.
5. Click **Save**.
6. Reboot the server.

Configure the Digest Identity Asserter Provider

Example 3-1 shows the security provider configuration in `config.xml` for a domain that uses LDAP implementation embedded in Converged Application Server. Note that such a configuration is recommended only for testing or development purposes. Example 3-1 highlights values that you must define when configuring the provider using the instructions in "Configure an LDAP Digest Identity Asserter Provider".

Example 3-1 Sample Security Provider Configuration with Embedded LDAP

```
<sec:authentication-provider xmlns:ext="http://www.bea.com/ns/weblogic/90/
security/extension" xsi:type="ext:ldap-digest-identity-asserterType">
        <sec:name>myrealmLdapDigestIdentityAsserter</sec:name>
        <ext:user-base-dn>ou=people, ou=myrealm, dc=mydomain</ext:user-base-dn>
        <ext:credential-attribute-name>description</ext:credential-attribute-
name>
        <ext:digest-realm-name>wlss.oracle.com</ext:digest-realm-name>
        <ext:host>server.example.com</ext:host>
        <ext:port>7001</ext:port>
        <ext:principal>cn=Admin</ext:principal>
      </sec:authentication-provider>
```

4

Configuring Client-Cert Authentication

This chapter describes how to configure Oracle Communications Converged Application Server to use Client-Cert authentication.

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 Converged Application 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). Converged Application Server provides two security providers that can be used to validate the X509 certificate; see "Configuring SSL and X509 for Converged Application Server".
- **WL-Proxy-Client-Cert Header**: Converged Application 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 Converged Application 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 Converged Application Server to enable use of the `WL-Proxy-Client-Cert` header; see "Configuring Converged Application Server to Use WL-Proxy-Client-Cert". You must also configure an X509 Identity Asserter provider as described in "Configuring SSL and X509 for Converged Application Server".

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" for a summary of steps required to implement perimeter authentication.

Configuring SSL and X509 for Converged Application Server

Converged Application 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, Converged Application 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 information on configuring secure transport in the *Oracle Communications Converged Application Server Administrator's Guide*.

See "Configuring the Default Identity Asserter" 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".

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 Converged Application Server, or you can create a custom user name mapper class. See the chapters on configuring a WebLogic credential mapping provider in *Securing Oracle WebLogic Server* the Oracle WebLogic Server Documentation for information on creating a custom user name mapper class.

Follow these instructions to configure the Default Identity Asserter:

1. Log in to the Administration Console for the Converged Application Server domain you want to configure.
2. In the left pane of the Console, select the **Security Realms** node.
3. Select the name of your security realm in the right pane from the Realms table.
4. Select the **Providers** tab and then select the **Authentication** subtab.
5. In the right pane of the Console, select *DefaultIdentityAsserter* from the Authentication Providers table.
6. Select the **Configuration** tab and then select the **Common** subtab.
7. Select **X.509** in the Available column of the Active Types table and use the arrow to move it to the Chosen column.
8. Click **Save** to apply the change.
9. You can use either a custom Java class to map names in the X.509 certificate to user names 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. Select the **Provider Specific** subtab.

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

c. Click **Save**.

To use the default user name mapper:

a. Select the **Provider Specific** subtab.

b. Select **Use Default User Name Mapper**.

c. In the **Default User Name Mapper Attribute Type** list, select either **CN** (for Common Name) or **E** (for 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 **Save**.

10. Restart the server.

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 Converged Application Server domain you want to configure.
2. In the left pane of the Console, select the **Security Realms** node.
3. Select the name of your security realm in the right pane from the Realms table.
4. Select the **Providers** tab and then select the **Authentication** subtab.
5. Click **New**.
6. Enter a name for the new provider, and select *LDAPX509IdentityAsserter* as the type.
7. Click **OK**.
8. In the list of providers, select the name of the provider you just created.
9. In the **Configuration > Provider Specific** tab, enter LDAP server information into the fields as follows:
   - **User Field Attributes**: Enter an LDAP search filter that Converged Application 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.
   - **User Name 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.
   - **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 *host name:port* value for each server separated by spaces. For example:
     `ldap1.mycompany.com:1050 ldap2.mycompany.com:1050`

See *Securing Oracle WebLogic Server* in the Oracle WebLogic Server 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 Converged Application Server and the LDAP Server.
- **Principal**: Enter the name of a principal that Converged Application 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 Save to save your changes.
11. Restart the server.

Configuring Converged Application Server to Use WL-Proxy-Client-Cert

In order for Converged Application Server to use the `WL-Proxy-Client-Cert` header, a proxy server or load balancer must first transmit the X.509 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 Converged Application 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 Converged Application Server domain you want to configure.
2. In the left pane, expand **Environment**, then select the **Servers** node.
3. Select the name of a server from the Servers table.
4. Select **Configuration**, then select the **General** subtab in the right pane.
5. Select **Client Cert Proxy Enabled**.
6. Click **Save** to save your changes.
7. Follow the instructions under "Configuring SSL and X509 for Converged Application Server" to configure either the default identity asserter or the LDAP Identity Asserter provider to manage X509 certificates.
8. Restart the server.

To enable the `WL-Proxy-Client-Cert` header for an individual Web Application, set the `com.bea.wcp.clientCertProxyEnabled` context parameter to true in the application's `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).

Converged Application 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 Converged Application 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 X.509 or

Default Identity Asserter providers described above. See *Securing Oracle WebLogic Server* in the Oracle WebLogic Server documentation for information about creating providers for handling tokens passed with perimeter authentication.

5

Configuring SIP Servlet Identity Assertion Mechanisms

This chapter describes how to configure and use Oracle Communications Converged Application Server Identity Asserter providers.

Overview of SIP Servlet Identity Assertion Mechanisms

A SIP Servlet can be configured to use one of the following identity assertion mechanisms:

- `P-Asserted-Identity`: With this mechanism, identity must be asserted using the `P-Asserted-Identity` header in a SIP message that originates from a trusted domain. This identity assertion mechanism is described in RFC 3325.
- `Identity`: With this mechanism, identity must be asserted using the `Identity` and `Identity-Info` headers in SIP messages, which can originate from other domains. This identity assertion mechanism is described in RFC 4474.

Converged Application Server does not support the `WebSocket` identity assertion mechanism.

You specify the identity assertion mechanism in the `@SipLogin` annotation inside an `@SipApplication` annotation. The `@SipLogin` annotation element determines which identity assertion mechanism is required for the Servlet. See the JSR 359, section 22.3.3 for information `@SipApplication` annotation, and information on configuring the identity assertion for a Servlet. See JSR 359, section 22.3.3.1 for information on `@SipLogin` annotation.

Converged Application Server supports identity assertion mechanisms using security providers. The sections that follow describe how Converged Application Server handles messages with each identity assertion mechanism, and how to configure the required security providers.

Understanding Trusted Host Forwarding with P-Asserted-Identity

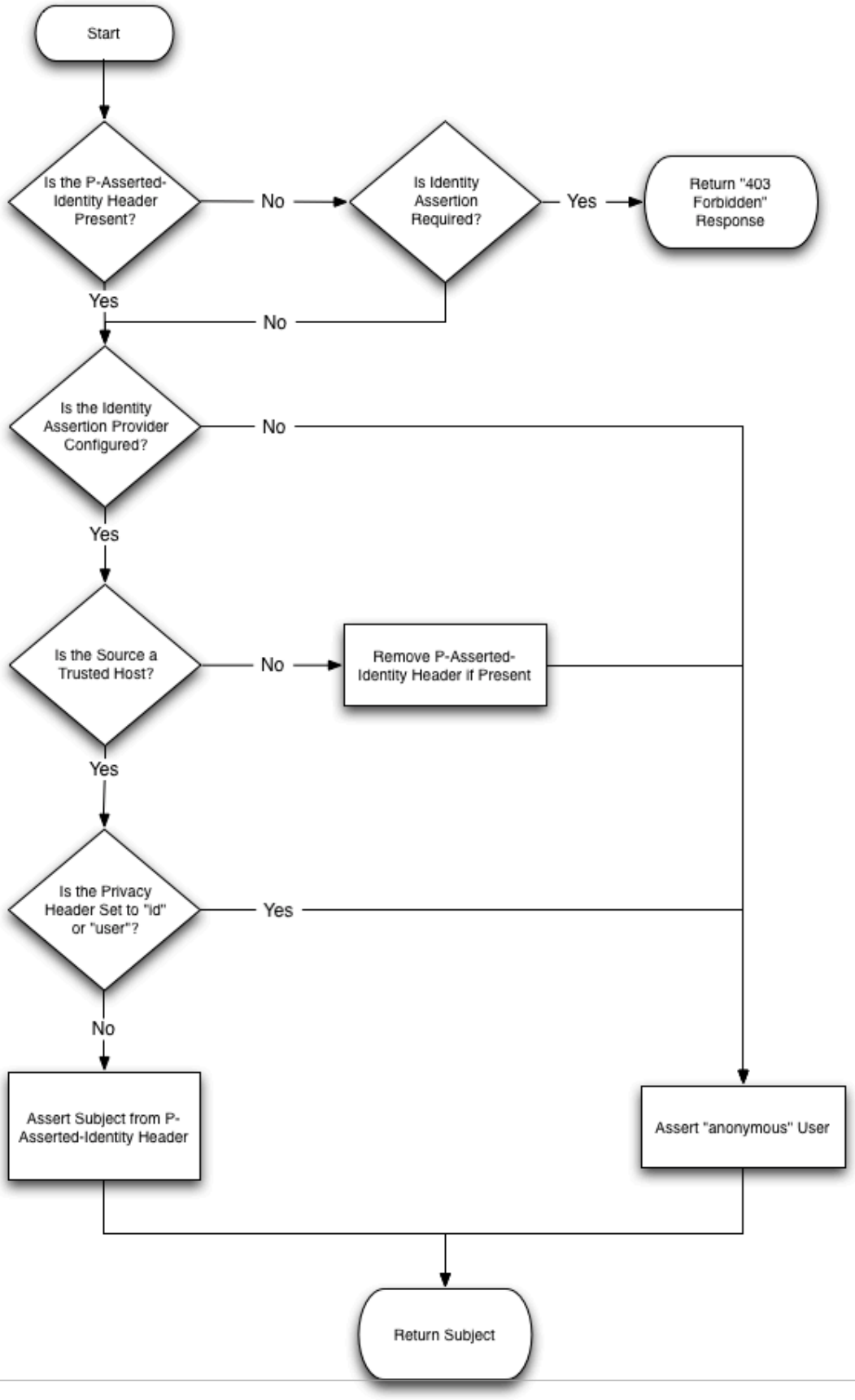
The `P-Asserted-Identity` header is honored only within a trusted domain. In a Converged Application Server system, trusted domains are purely configuration-based. To enable use of the header, you must configure one of two available P-Asserted Identity Assertion providers as described in "Configuring a P-Asserted-Identity Assertion Provider". 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 Converged Application 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@oracle.com`). By default, Converged Application Server removes the domain portion of the address

(`@oracle.com`) and uses the remainder as the user name. If you must support overlapping usernames from different names (for example, `sipuser@oracle.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 `@oracle.com`.

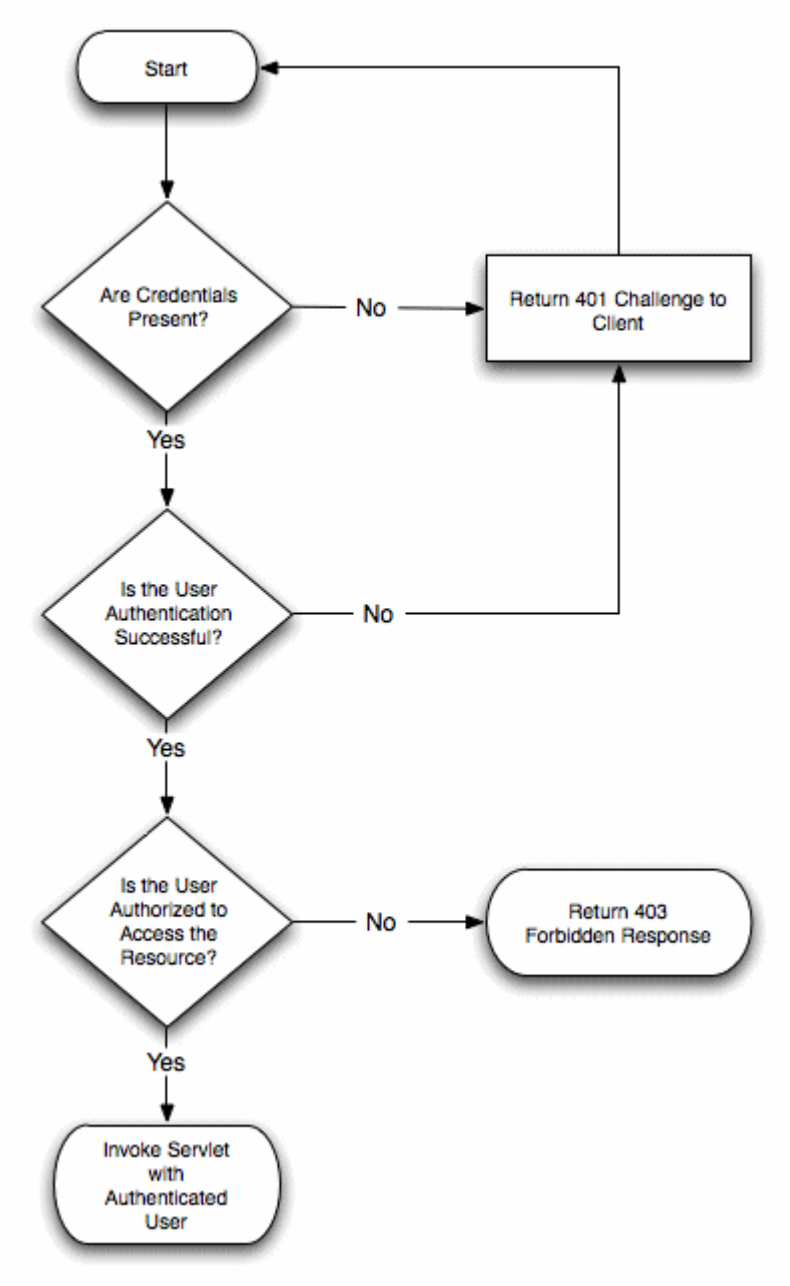
The presence of a `P-Asserted-Identity` header combined with the `Privacy` header also determines the way in which Converged Application Server proxies incoming requests. The value of the `@SipLogin` annotation is also considered.

Figure 5-1 Managing Inbound Requests Having P-Asserted-Identity and Privacy Headers



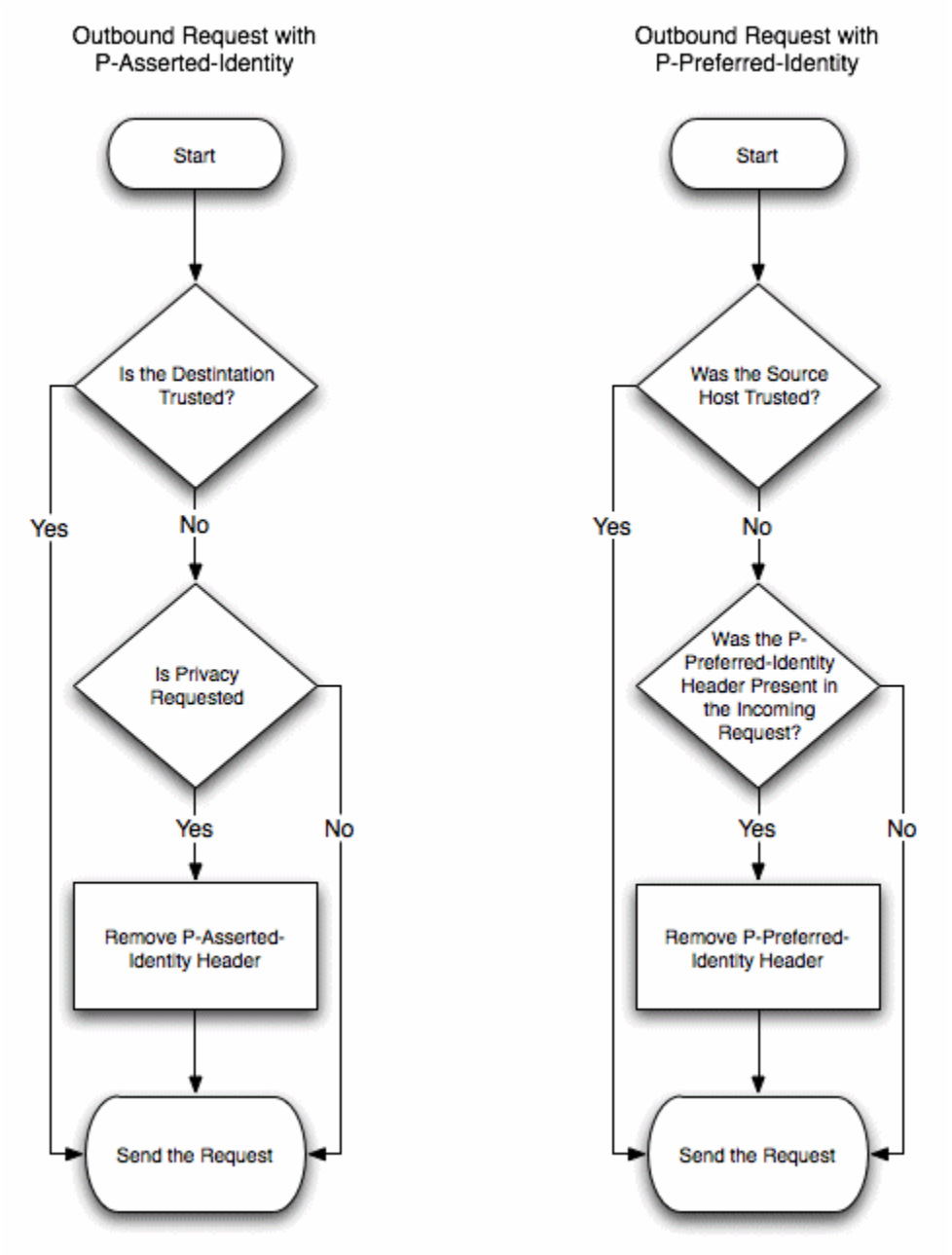
The following diagram describes the standard security check procedure that Converged Application Server uses when an asserted user name is not authorized to access a requested resource. The standard security check is performed according to the `auth-method` defined in the `login-config` element of the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

Figure 5-2 Standard Security Check Procedure



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

Figure 5-3 Managing Outbound Requests Having P-Asserted-Identity or P-Preferred Identity



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 *Converged Application Server Developer's Guide* for more information.

If authorization of the anonymous user fails, Converged Application 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".

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 Converged Application Server domain you want to configure.
2. In the left pane of the Console, select the **Security Realms** node.
3. Select the name of your security realm in Realms table in the right pane of the Console.
4. Select the **Providers** tab and select the **Authentication** subtab.
5. Click **New** in the Authentication Providers table.
6. Enter a name for the new provider, and select one of the following options for the Type:
   - *PAssertedIdentityAsserter*: 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.
   - *PAssertedIdentityStrictAsserter*: 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" for more information.
7. Click **OK**.
8. Select the name of the provider you just created.
9. Select the **Provider Specific** tab.

10. Fill in the fields of the configuration 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 information on `sip-security` in the *Oracle Communications Converged Application Server Administrator's Guide*.

    - **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 *Securing Oracle WebLogic Server* in the Oracle WebLogic Server documentation for more information.

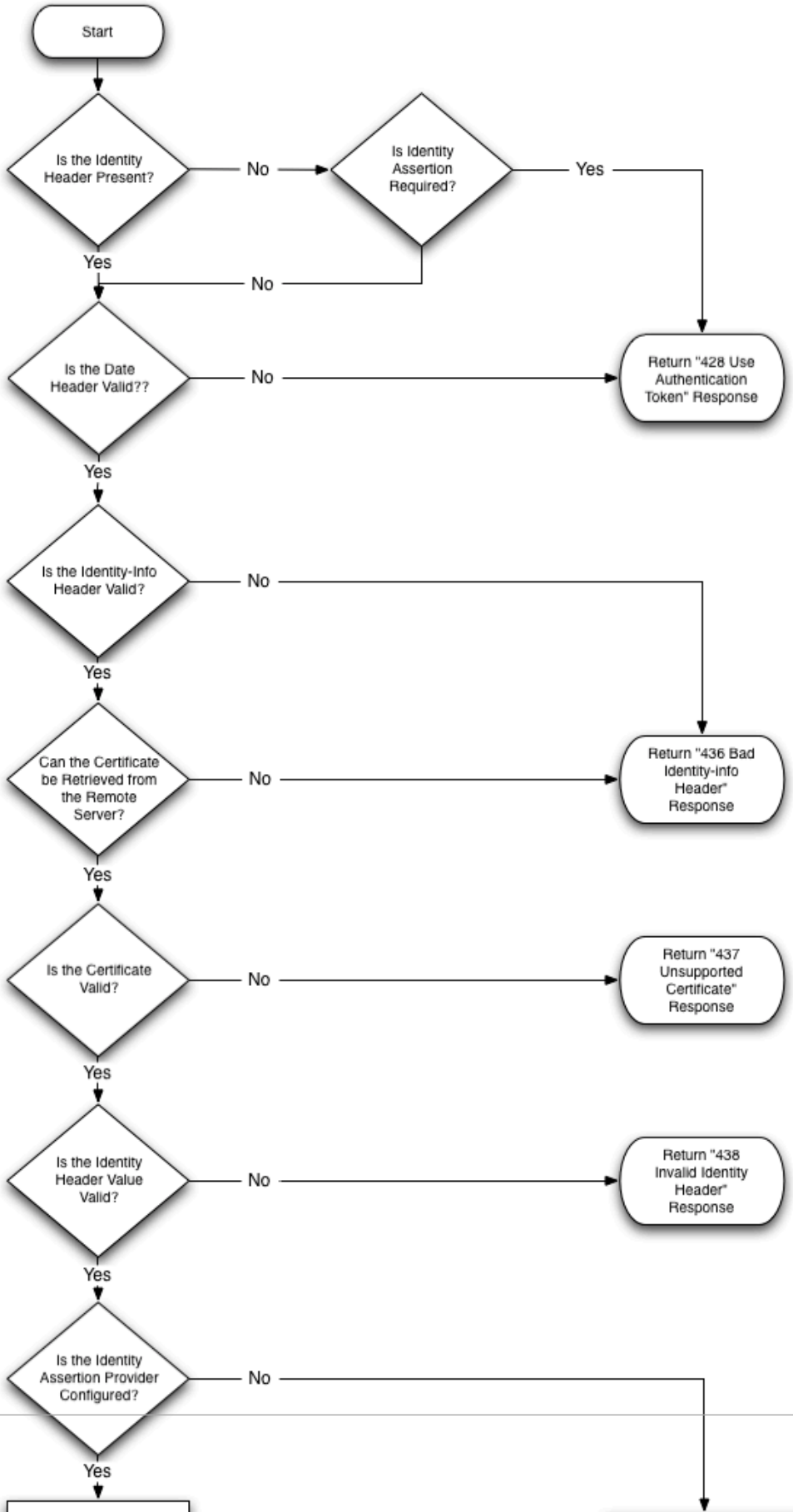
      Alternatively, leave this field blank to use the default user name mapper. The default mapper simply discards the domain name and takes the resulting user name without applying any additional logic.
11. Click **Save**.
12. Restart the server.

Understanding Identity Assertion with the Identity and Identity-Info Headers

Converged Application Server can also perform identity assertion using the `Identity` and `Identity-Info` headers, as described in RFC 4474. As with the `p-asserted-identity` assertion mechanism, `Identity` header assertion requires that you first configure the appropriate security provider (the `IdentityHeaderAsserter` provider) in Converged Application Server.

When asserting the identity of inbound requests having the `Identity` and `Identity-Info` headers, Converged Application Server considers the values of the `identity-assertion` and `identity-assertion-support` elements in `sip.xml` as well as the presence of a configured security provider.

Figure 5-4 Managing Inbound Requests Having Identity and Identity-Info Headers



Configuring the Identity Header Assertion Provider

Follow these steps to configure the security provider used to support the `Identity` header:

1. Log in to the Administration Console for the Converged Application Server domain you want to configure.
2. In the left pane of the Console, select the **Security Realms** node.
3. Select the name of your security realm from the Realms table in the right pane of the Console.
4. Select the **Providers** tab, and then select the **Authentication** subtab in the right pane.
5. Click **New** in the Authentication Providers table.
6. Enter a name for the new provider, and select *IdentityHeaderAsserter* for the Type.
7. Click **OK**.
8. Select the name of the provider you just created in the Authentication Providers table.
9. Select the **Configuration** tab and then select the **Provider Specific** subtab.
10. Fill in the fields of the Provider Specific subtab as follows:
    - **Date Period**: Enter the valid period for Date header, in seconds.
    - **Https Channel Name**: Enter the name of an HTTPS channel the provider should use to initialize an HTTPS client. An HTTPS channel is required (and must be configured separately) if a remote certificate must be retrieved via HTTPS.
    - **User Name Mapper Class Name** (optional): Enter the name of a custom Java class used to map user names in the `Identity` header to user names in the default security realm. A custom user name mapper class is required if you want to map usernames in the `Identity` header to WebLogic usernames. See *Securing Oracle WebLogic Server* in the Oracle WebLogic Server documentation for more information.
11. Click **Save**.
12. Restart the server.

6

Configuring 3GPP HTTP Authentication Assertion Providers

This chapter describes how to configure Oracle Communications Converged Application Server to handle the `X-3GPP-Asserted-Identity` header for HTTP authentication.

Overview

In order to function as an Application Server in an IMS network, Converged Application 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) (http://www.3gpp.org/ftp/Specs/html-info/33222.htm). Converged Application 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@oracle.com, user2@oracle.com). When configured with the default user name mapper class, the Converged Application Server providers remove the domain portion of the addresses (`@oracle.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@oracle.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 `@oracle.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 *Oracle Fusion Middleware Securing Oracle WebLogic Server* 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":

1. Log in to the Administration Console for the Converged Application Server domain you want to configure.
2. In the left pane of the Console, select the **Security Realms** node.
3. Select the name of your security realm from the Realm table in the right pane of the Console.
4. Select the **Providers** tab and then select the **Authentication** subtab.
5. In the Authentication Providers table, select **New**.
6. Enter a name for the new provider, and select one of the following options from the **Type** drop down list:
   - *X3gppAssertedIdentityAsserter*: 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.
   - *X3gppAssertedIdentityStrictAsserter*: 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" for more information.
7. Click **OK**.
8. Select the name of the new provider you just created from the Authentication Providers table.
9. In the Active Types chooser list, select the *X-3GPP-Asserted-Identity* type and use the arrow to move it to the Chosen column.
10. Click **Save**.
11. Select the **Provider Specific** tab.
12. Fill in the fields of the configuration page 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 information on `sip-security` in the *Oracle Communications Converged Application Server Administrator's Guide*). 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 *Oracle Fusion Middleware Securing Oracle WebLogic Server* for more information.

Alternatively, 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@oracle.com", "user2@oracle.com"
```

and asserts the identity "user1."

13. Click **Save**.
14. Restart the server.