Oracle® Communications ASAP

Security Guide

Release 7.2

E28042-01

April 2012

ORACLE®

Oracle Communications ASAP Security Guide, Release 7.2

E28042-01

Contents

A ASAP Secure Deployment Checklist

Preface

This guide provides guidelines and recommendations for setting up Oracle Communications ASAP in a secure configuration.

Audience

This guide is intended for system administrators, database administrators, developers, and integrators.

Related Documents

For more information, see the following documents in the Oracle Communications ASAP 7.2 documentation set:

- *Oracle Communications ASAP Release Notes*
- *Oracle Communications ASAP Concepts*
- *Oracle Communications ASAP Installation Guide*
- *Oracle Communications ASAP Service Request Translator User's Guide*
- *Oracle Communications ASAP Order Control Application User's Guide*
- *Oracle Communications ASAP Server Configuration Guide*
- *Oracle Communications ASAP System Administrator's Guide*
- *Oracle Communications ASAP Cartridge Development Guide*
- *Oracle Communications ASAP Developer's Guide*

Note: To download the *ASAP Developer's Guide* from the Oracle software delivery Web site, you must select **Oracle Communications Service Activation Developer Documentation Pack**. You can visit the Oracle software delivery Web site at:

http://edelivery.oracle.com

1

ASAP Security Overview

This chapter provides an overview of Oracle Communications ASAP security.

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, how often they should be accessed, and who should monitor those components.
- **Install software securely.** For example, use firewalls, secure protocols (such as SSL), and secure passwords. See "Performing a Secure ASAP Installation" for more information.
- **Learn about and use ASAP security features.** See "Implementing ASAP Security" for more information.
- **Use secure development practices.** For example, take advantage of existing database security functionality instead of creating your own application security. See "Security Considerations for Developers" for more information.
- **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. See "Critical Patch Updates and Security Alerts" on the Oracle Web site:

  `http://www.oracle.com/technetwork/topics/security/alerts-086861.html`

Understanding the ASAP Environment

When planning your ASAP implementation, consider the following:

- **Which resources must be protected?** For example:
  - You must protect customer data.
  - You must protect internal data, such as proprietary source code.
  - You must protect system components from being disabled by external attacks or intentional system overloads.
- **Who are you protecting data from?**

For example, if your business has service subscribers, you must protect their data from other subscribers, but someone in your organization might have to access that data to manage it. You can analyze your workflows to determine who needs access to the data; for example, a system administrator could manage your system components without needing to access the system data.

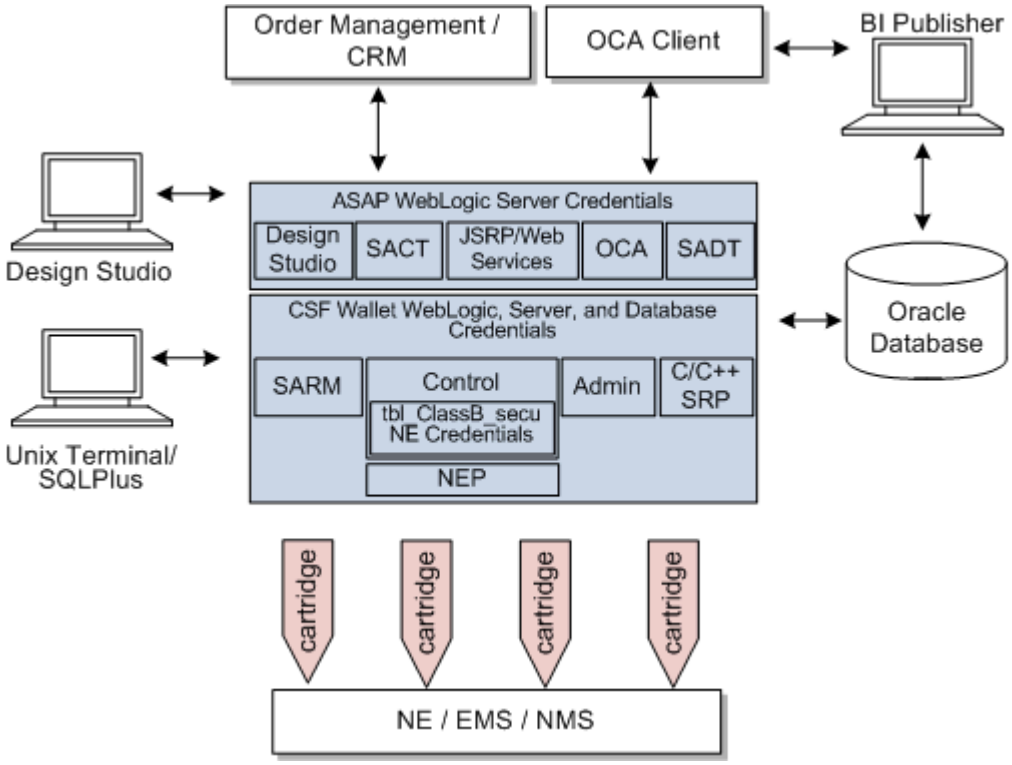
- **What happens if protections on strategic resources fail?**

  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 helps you protect it properly

Overview of ASAP Security

Figure 1–1 shows all the various components that can comprise ASAP, including the components to which it connects.

Figure 1–1 ASAP Components



ASAP security is designed for three essential functions: managing ASAP WebLogic-based users, securing data, and protecting diagnostics files. ASAP provides these security functions in the following locations:

- **ASAP WebLogic server security**: An ASAP WebLogic server instance contains default users, groups, and roles that support the various WebLogic-based ASAP functionality, like the order control application (OCA) client, the Service activation configuration tool (SACT), the service activation deployment tool (SADT) and the Java service request processor (JSRP).
- **ASAP server and database credential security**: The ASAP environment contains a credential store factory (CSF) wallet that stores the ASAP schema user names and

passwords, and the ASAP WebLogic server user name and password. These credentials are called class A secure data. Each ASAP server can use this wallet to obtain these credentials. The CSF wallet provides both secure storage and encryption for these credentials.

- **Network Element (NE) credential security for Network Element Processor (NEP) to NE communication**: The ASAP Control database stores credentials required for NEPs to access NEs. These credentials are called class B secure data. You can use the ASAP security tool to add, change, or delete credential information, and also to enable encryption.
- **ASAP system configuration parameters**: Some ASAP system configuration parameters can have a significant impact on security. Parameters settings, such as diagnostic levels and server security attributes should be configured to ensure data security.

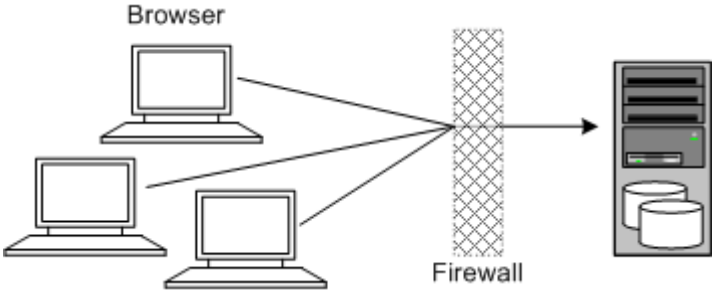
Recommended Deployment Topologies

This section describes recommended deployment topologies for ASAP.

Single-Computer Installation Topology

Figure 1–2 shows a single-computer installation topology.

Figure 1–2 Single-Computer Deployment



In this topology, all the application components and data are kept on a single system, protected from external attacks by a firewall. The firewall can be configured to block known illegal traffic types. There are fewer resources to secure because all the components are on a single system and all the communication is local. Fewer ports have to be opened through the firewall.

Conversely, there are fewer points of attack, and if security is compromised, an attacker would have access to the entire system and data.

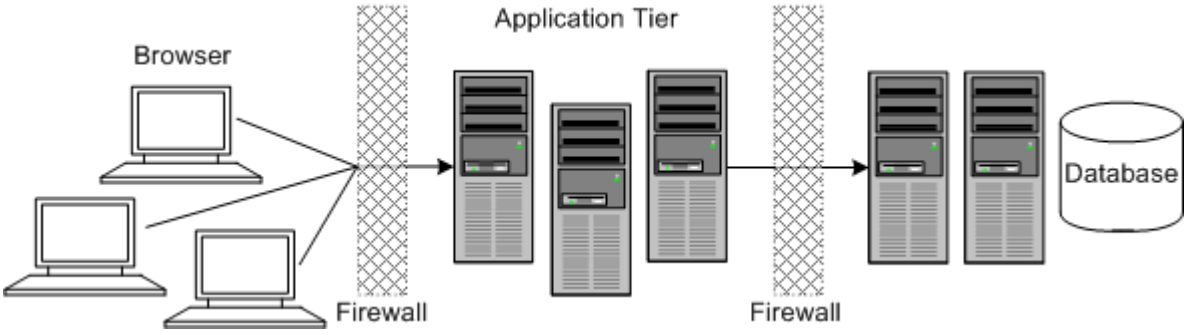
A single-computer installation topology is best suited for test and lab environments:

A single-computer deployment is cost effective for small organizations but does not provide high availability because all components are stored on a single system.

Tiered Deployment

Figure 1–3 shows a tiered installation deployment: a scalable ASAP deployment offering greater security and high availability.

Figure 1–3 Tiered Deployment



In this topology, the application tier is isolated by firewalls from both the Internet and the intranet. The database and servers are protected from potential attacks by two layers of firewall. Both firewalls can be configured to block known illegal traffic types. The two layers of firewall provide intrusion containment. Although there are a greater number of components to secure, and more ports have to be opened to allow secure communication between the tiers, the attack surface is spread out.

ASAP Port Requirements

Table 1–1 lists and describes ASAP ports.

Table 1–1 ASAP Ports

Port	Description
SARM server	The SARM server port for sending and receiving.
Control server	The Control server port for sending and receiving.
NEP server JNEP listener	The NEP server port for sending and receiving. The JNEP listener port for sending and receiving.
Admin server	The Admin server port for sending and receiving.
Daemon server	The Daemon server port for sending and receiving.
OCA server	The OCA server port for sending and receiving.
JSRP sending WO JSRP receiving WO	The JSRP port for sending work orders. The JSRP port for receiving work orders.
Database connection	The port in the Oracle database connection string. There may be multiple ports if an Oracle Real Application Clusters (RAC) database is used.
WebLogic connection	The port for the ASAP WebLogic server and optional managed server. In addition, if the ASAP WebLogic server is installed on a different machine, you must also open the ports to the Oracle database from there.
Telnet for remote servers	If ASAP is deployed on multiple servers in a distributed configuration, the telnet port for rsh connectivity must be open.

Operating System Security

See the following documents:

- *Guide to the Secure Configuration of Red Hat Enterprise Linux 5*
- *Hardening Tips for the Red Hat Enterprise Linux 5*

Oracle Database Security

For more information about securing an Oracle Database, see *Oracle Database Security Guide* and *Oracle Database Advanced Security Administrator's Guide*.

WebLogic Server Security

For information about securing an ASAP WebLogic server, see *Oracle Fusion Middleware Securing a Production Environment for Oracle WebLogic Server*.

LDAP Security

Oracle recommends that you use Oracle Internet Directory for identity management (for example, users, roles, certificates). You can also use an external LDAP, which you must integrate with ASAP through the ASAP WebLogic server.

For information about setting up Oracle Internet Directory, see *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.

For information about setting up an external LDAP, see the LDAP application documentation. For information about security realms and setting up ASAP with an external LDAP, see *ASAP System Administrator's Guide*.

Oracle Security Documentation

ASAP uses other Oracle products, such as Oracle Database and Oracle WebLogic server. See the following documents, as they apply to ASAP:

- *Oracle Database Security Guide*
- *Oracle Fusion Middleware Securing a Production Environment for Oracle WebLogic Server*
- *Oracle Fusion Middleware Securing Resources Using Roles and Policies for Oracle WebLogic Server*
- *Oracle Application Server Security Guide*
- *Oracle Application Server Administrator's Guide*

2

Performing a Secure ASAP Installation

This chapter presents planning information for your Oracle Communications ASAP system and describes recommended deployment topologies that enhance security.

For more information about installing ASAP, see *ASAP Installation Guide*.

Pre-Installation Configuration

This section explains the pre-requisites to install ASAP with security:

- You must have at least one dedicated UNIX group and one dedicated user account within that group for ASAP.
  - Create a group for ASAP that includes the ASAP user account and the **root** user.
- When creating the ASAP WebLogic server domain:
  - Make sure that the administration server and the optional managed server SSL ports are used.
  - After you have created the WebLogic Server domain for ASAP, start the WebLogic administration server. Then, use t3s to start the managed server:

    ```
    startManagedWebLogic.sh ManagedServer t3s://host_name:SSL_Port
    ```

    Where *ManagedServer* is the name of the WebLogic managed server, *host_name* and *SSL_Port* are the host name and the SSL port number of the WebLogic administration server.
- Using the WebLogic administration console, configure certificate identity and trust store to use SSL. Do not use the default, demonstration certificate that comes with WebLogic server. See the WebLogic documentation for more information.

Installing ASAP Securely

You can perform a custom installation or a typical installation. Perform a custom installation to avoid installing options you do not need. Unused options and sample files can contain security vulnerabilities if deployed in a production environment.

To deploy and configure ASAP resources securely in the ASAP WebLogic sever domain, do the following:

1. Follow the steps to install ASAP as described in the *ASAP Installation Guide*, selecting the following:
   a. In the WebLogic Configuration screen, enter the SSL port of WebLogic administration server.

b. Select the option **Use SSL**.

   The **Enter Keystore File** field is enabled.

c. In the **Enter Keystore File** field, enter the KeyStore file.

d. After installing ASAP, change the passwords for all default ASAP WebLogic user accounts.

Securely Integrating BI Publisher with ASAP

Oracle Business Intelligence Publisher (BI Publisher) is installed into a WebLogic server domain. When installing BI Publisher, configure it to communicate with the SARM and Admin server over an SSL-enabled channel, and disable all unused ports, especially unsecured ports. See the BI Publisher documentation for more information.

3

Implementing ASAP Security

This chapter explains the security features of Oracle Communications ASAP. See *ASAP System Administrator's Guide* for more information on the ASAP security functionality.

Configuring WebLogic Server Security

ASAP uses the LDAP server included with the WebLogic Server software to manage default ASAP users, groups, roles, and methods. For more information about this embedded LDAP server, see the WebLogic Server documentation.

Configuring Authentication Providers for ASAP

During the ASAP installation process, the ASAP installer creates default ASAP users, groups, roles, and methods in the embedded LDAP authentication provider included with the ASAP WebLogic server. You can use this authentication provider to configure the default ASAP users, groups, roles, and methods, or add, delete, or modify your own users, groups, roles, and methods.

ASAP also supports external LDAP provides, such as the Oracle Internet Directory.

Managing ASAP WebLogic Server User Security

ASAP supports only the default WebLogic server **myrealm** security realm. Using security realms other than **myrealm**, disabled all ASAP WebLogic-based features.

ASAP administrators can configure user password policies through the WebLogic Administration Console and the password policy utility page. For more information, see *ASAP System Administrator's Guide*.

Configuring ASAP Server and Database Credential Security

Secure data must be stored in a secure location and distributed to authorized users. The ASAP security system governs how secure data is managed and ASAP diagnostics files are secured. This security system includes:

- **Secure Data Storage**: The ASAP security administrator pre-defines the nature and accessibility of secure data for each ASAP server. Class A secure data is stored in the CSF wallet during the initial ASAP installation procedures. For more information, see *ASAP Installation Guide*.
- **Secure Data Encryption**: The CSF wallet encrypts all data contained in it and obtained from it. In addition, the CSF wallet file (**cwallet.sso**) has restricted access

permissions. Many ASAP utilities and scripts use the passwords contained in the CSF wallet.

Configuring Security for Network Elements Communication

NE credentials (also called custom secure class B data) used primarily by NEPs to establish network connections to NEs must be stored in a secure location and distributed to authorized users. An ASAP administrator can store NE credentials using ASAP APIs or the command line ASAP security tool (**asap_security_tool**).

The ASAP security tool supports the following features to protect NE credentials:

- **Secure Data Storage**: An administrator can use the ASAP security tool to create NE credentials and store these credentials in a central repository on the Control server. The Control server distributes these credentials to NEPs and Java-enabled NEPs (JNEPs).

  ASAP stores NE credentials in the Control server in the **tbl_classB_secu** database table.

- **Secure Data Encryptio**n: The Control server uses a symmetric secret key encryption method to achieve data confidentiality for custom secure data.

- **Key Distribution**: The Control server acts as a key distribution server, and distributes custom secure data to every ASAP server during provisioning. To acquire custom secure data, ASAP servers use a pre-defined key distribution protocol.

4

Security Considerations for Developers

This chapter provides information for developers about how to secure Oracle Communications ASAP work order messages from order management systems to and from the Java service request processor (JSRP) or the ASAP Web Service implementation, and provides information about developing secure ASAP cartridges.

Securing OSS/J Over JMS Connections

You can secure OSS/J messages over Java messaging service (JMS) connections using WebLogic security policies. These policies can be created to secure JMS destinations. These policies enable only authorized ASAP WebLogic users to send, receive, and browse JMS messages to and from a destination. For more information about configuring WebLogic Server JMS connection security, see the *Oracle Fusion Middleware Securing Resources Using Roles and Policies for Oracle WebLogic Server*.

Note: Web Service messages can also be sent over JMS.

Securing Web Services Connections

Web Services connections can be established over JMS, HTTP, and HTTPS. Oracle recommends using JMS or HTTPS to insure secure Web Service work order communication between and order management system and ASAP.

ASAP Web Service access control security determines the functionality that each user can access. ASAP uses policies and roles configured within the ASAP WebLogic server to secure Web Service work order messages. Clients that send Web Service work orders must provide an ASAO WebLogic user id that is a member of the ASAP WebLogic group **ASAP_WS_USERS_GROUP**. The **web.xml** file defines the security role **ASAP_WS_USERS** and **weblogic.xml** file defines the security principal name as **ASAP_WS_USERS_GROUP**.

Here is a sample security header in a SOAP request:

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<env:Header>
        <wsse:Security
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-sec
ext-1.0.xsd" env:mustUnderstand="1">
            <wsse:UsernameToken
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-util
ity-1.0.xsd" wsu:Id="unt_AF6po7ocfkMUDzde">
                <wsse:Username>username</wsse:Username>
```

```
                <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profi
le-1.0#PasswordText">password</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </env:Header>
    <env:Body>
        <m:order_type
xmlns:m="http://xmlns.oracle.com/communications/activation/asap/webservices">

OSS/J_work_order

        </m:order_type>
    </env:Body>
</env:Envelope>
```

Where:

- *username*: is the user name for the Web Service user-defined in the ASAP WebLogic server instance.
- *password*: is the password for the Web Service user-defined in the ASAP WebLogic server instance.
- *order_type*: is the type of work order sent.
- *OSS/J_work_order*: is the OSS/J work order information. When you add the work order information, do not include the XML header information (**<?xml version="1.0" encoding="UTF-8"?>**) since this has already been provided in the sample. Also, ensure that there are no namespace conflicts.

Cartridge Development

When developing an ASAP cartridge, store NE credentials in the Control server secure class B table. You must configure ASAP Java methods to access these credentials from the Control server when you establish connections from the NEP or JNEP to each NE. For more information, see *ASAP Cartridge Development Guide*.

A

ASAP Secure Deployment Checklist

The following security checklist lists guidelines to help you secure Oracle Communications ASAP and its components.

Secure Deployment Checklist

- Install only the components you require.
- Lock and expire default user accounts.
- Enforce strong password management.
- Restrict, control, and revisit user privileges:
  - Grant only the necessary privileges to each user.
  - Revoke unnecessary privileges from the PUBLIC user group.
  - Restrict permissions on run-time facilities
- Enforce the use of access controls.
- Require clients to authenticate.
- Restrict network access by doing the following:
  - Use firewalls.
  - Never leave an unnecessary hole in a firewall.
  - Password-protect the Oracle listener against remote access.
  - Monitor listener activity.
  - Monitor who accesses your systems.
  - Restrict system access by IP addresses
  - Encrypt network traffic.
  - Harden the operating system by installing it in a secure location where it would be difficult for a hacker to access, by ensuring that all null passwords have been changed, and by disabling remote root login.
- Apply all security patches and workarounds.
- Encrypt sensitive information.
- Contact Oracle Security Products if you discover a vulnerability in any Oracle product.