ORACLE
INSURANCE

Oracle® Insurance Policy Administration

Security Guide

Version 10.0.0.0

Documentation Part Number: E40981_01

October, 2013

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

License Restrictions

Warranty/Consequential Damages Disclaimer

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

Warranty Disclaimer

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Restricted Rights Notice

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Hazardous Applications Notice

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Third Party Content, Products, and Services Disclaimer

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

TABLE OF CONTENTS

OVERVIEW

Security planning is a critical step to help protect your company's valuable data and ensure that information is not compromised. Established security policies and goals should guide the security plan your organization executes to secure its systems.

The Oracle Insurance Policy Administration (OIPA) system stores sensitive data and requires security measures to be taken. Security policies should align with those already established at your organization, or new ones should be established if they are not already defined.

This document provides guidelines for securing an OIPA installation, including the configuration and installation steps needed to meet security goals. Details on the types of security features and services that are available to detect and prevent a potential security breach are provided. These details encompass secure system deployment, protection of sensitive data, reliability and availability of the application, authentication and authorization mechanisms.

You may use this document to develop your organization's security policies and practices in the context of OIPA. It is critical that an organization set security standards and properly implement them. The development and review of security documentation, an evaluation of business requirements, and the configuration and validation of available security measures and services should all be performed.

Customer Support

If you have any questions about the installation or use of our products, please visit the My Oracle Support website: https://support.oracle.com, or call (800) 223-1711.

Oracle customers 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.

SYSTEM DEPLOYMENT

Network Security in OIPA Environment

When deploying OIPA on a network there are many security issues to take into consideration, especially the use of firewall and VPN technologies. A firewall will permit or deny network permissions based on configured rules, to protect the internal network from unauthorized access while permitting legitimate communications. Firewalls perform the following functions in a typical OIPA environment:

- Guard the company Intranet from unauthorized outside access.
- Separate Intranet users accessing the OIPA system from internal subnetworks where critical corporate information and services reside.
- Protect from IP spoofing and routing threats.
- Prohibit unauthorized users from accessing protected networks and control access to restricted services.

The OIPA user interface is browser-based and allows home-office users to access the application services. It is recommended that the users access the application from within the company network, secured behind the outside firewall. Virtual Private Network (VPN) technology should be used to allow employees working remotely to access the OIPA application. A VPN tunnels outside traffic through the firewall, placing outside clients virtually inside the firewall.

It may be required to provide access to the OIPA web services for external clients that are not allowed inside the company firewall. In that case, the web services must only be accessed through HTTP secured with SSL. OIPA web services support WS-Security standards, enabling web service user authentication using OIPA user accounts.

Please make sure that the firewalls used to secure an OIPA environment support the HTTP 1.1 protocol. This enables browser cookies and inline data compression for improved performance.

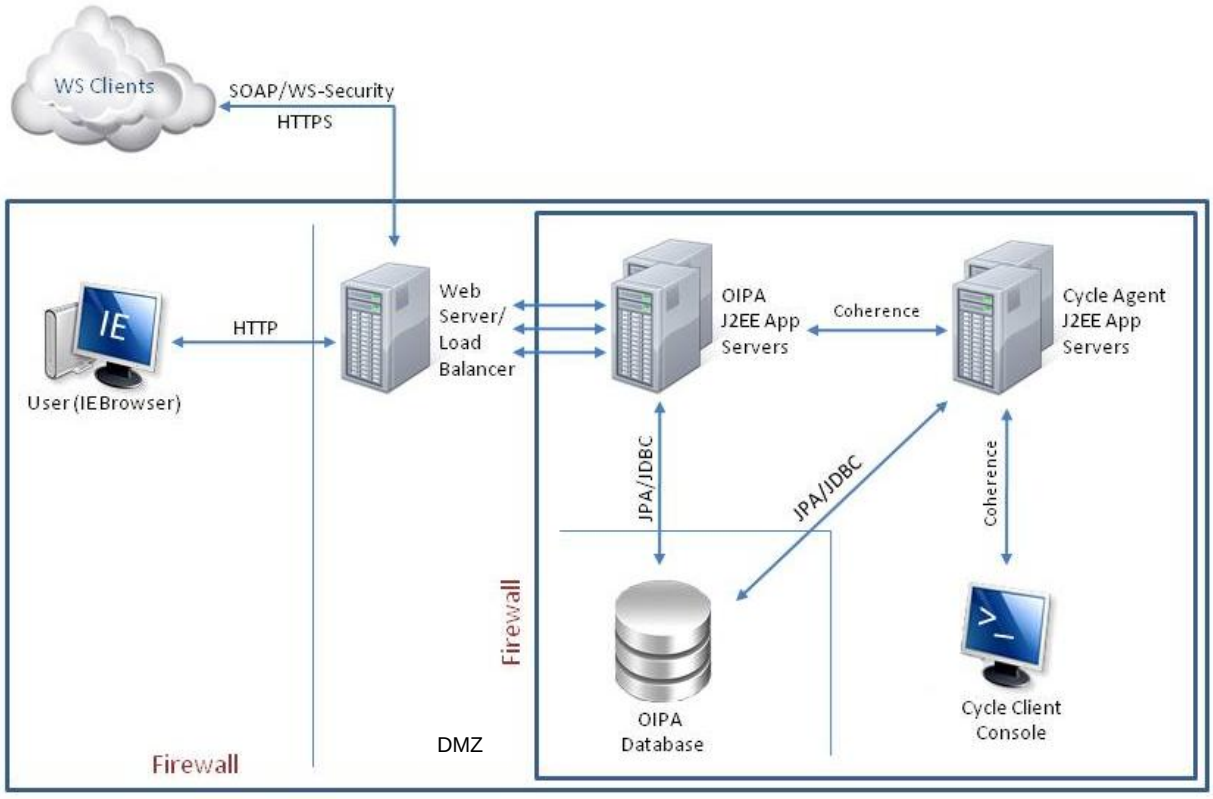

Figure 1. Firewalls in the OIPA environment

A typical OIPA environment usually has the following security zones:

- **Internet** - External web service clients may come from outside of the company network.
- **Intranet** - A company network separated by the external firewall that gives home users access to the OIPA user interface. This is also where OIPA web servers and load balancers may be placed. Alternatively, for additional protection, web and load balancing servers may be placed in a separate demilitarized zone (DMZ) where external and internal clients first interact with the OIPA environment.
- **OIPA application server and database zone** - OIPA application servers, including Cycle Web servers, database servers and possibly authentication servers (for example, if a customer chooses to implement a single sign-on using LDAP servers) reside in this zone. Access to the database that holds critical client information must be secured, with access restricted to system and database administrators only.

Database

Appropriate database users for the OIPA application should be set up as per the instructions in the associated version of the OIPA Database Install Instructions, which is located on the Oracle Technology Network. OIPA expects a Read-Only database user to be set up so that additional restrictions can be enforced on certain operations.

OIPA Use of Coherence

The OIPA application uses the Oracle Coherence distributed cache solution to minimize database traffic. In addition to using the cache, OIPA Cycle uses the Coherence Processing Pattern as a computing grid to allow task distribution among all OIPA Cycle Web. Batch processing on the grid is initiated through the Coherence communication protocol by the Cycle Client. Even though all parties involved in Coherence communications are located behind the firewall in the OIPA application server and database zone, it is important, nevertheless to secure Coherence according to the Coherence User Guide.

Oracle Coherence also provides workload management to distribute tasks across a computer cluster or other resources. This enables Cycle to achieve optimal resource utilization, maximize throughput, minimize response time and avoid overload, as well as avoid having a single point of failure for tasks processed in the grid. Along with the security provided by the firewalls, Coherence workload management provides these additional security features:

- TCP port exposure is limited to a single port that allows easier port security and firewall configuration.
- A virtual IP address hides actual physical IP addresses of the OIPA application servers.
- The suspect protocol protects against Denial of Service (DoS) attacks by detecting and barring “rogue” clients that attempt to overuse server resources.

Configuring SSL

The Secure Sockets Layer (SSL) protocol provides communication security by encrypting traffic across a network in a way designed to prevent eavesdropping and tampering. It uses asymmetric cryptography for privacy and a keyed message authentication code for message reliability. Setting up an SSL-secured connection requires a digital certificate issued by a trusted certificate authority. Self-signed digital certificates should only be used for internal testing.

Any entry points for OIPA web services that are consumed by external third party clients should be secured with SSL. Also, organization standards may require securing communication between browser-based clients and web servers in the demilitarized zone that host the front end of the OIPA system.

Setting up a web server to use SSL-secured HTTP protocol (HTTPS) instead of unsecure HTTP is server-specific. The information below should help locate information to navigate through the configuration process.

SSL in WebLogic 10.3.6.0

WebLogic Application Server supports SSL 3.0 and Transport Layer Security (TLS) 1.0 specifications. WebLogic does not support SSL version 2.0 and below.

For information on how to configure SSL in WebLogic please refer to the following websites or follow the steps below:

http://docs.oracle.com/cd/E23943_01/web.1111/e13707/ssl.htm#SECMG384

http://download.oracle.com/javase/6/docs/technotes/guides/security/jsse/JSSERefGuide.html

Steps to Configure SSL/https:

1. Login to the WebLogic console.
2. In the Domain Structure box, expand **Environment** and click **Servers**.
3. Click on the server that you created. Example: OIPA_SERVER.
4. Select the SSL Listen Port Enabled checkbox. Example: 7002 is port number
6. Click **Save**.
7. Restart the server.
8. Navigate to `https://machinename:7002/PASJava` in your Internet Explorer browser to access the login page of OIPA.

http://docs.oracle.com/cd/E23943_01/apirefs.1111/e13952/taskhelp/security/ConfigureKeystoresAndSSL.html

Steps to Configure Certificates:

The steps listed below are based on the default JDK certificate.

```
WEBLOGIC_JAVA_SECLIB = Specify the location of JDK 1.7.x. /jre/lib/security. For
Example: /opt/oracle/jdk1.7.0_25/jre/lib/security
WEBLOGIC_JAVA_HOME = Specify the location of JDK 1.7.x. For Example:
/opt/oracle/jdk1.7.0_25/
```

Note: If JDK is not installed on your machine, then download and install latest update of Oracle 1.7 JDK

1. Install the Oracle WebLogic 10.3.6.0 application server.
2. Go to `WEBLOGIC_JAVA_HOME\bin` and run the commands listed below.
   - `keytool -genkey -keystore jre/lib/security/wsse.keystore -storepass jbossws -keyalg RSA -keysize 1024 -validity 1000 -alias localhost -dname "CN=localhost"`
   - `keytool -export -keystore jre/lib/security/wsse.keystore -storepass jbossws -alias localhost -file server/default/conf/localhost.cer`
   - `keytool -import -keystore jre/lib/security/wsse.truststore -storepass jbossws -trustcacerts -alias localhost -file jre/lib/security/localhost.cer`
3. The above step will create two files within `WEBLOGIC_JAVA_SECLIB`.
   - wsse.keystore
   - wsse.truststore
4. Move wsse.keystore and wsse.truststore to the **conf** folder where all properties files reside. Example: `C:\OIPA\conf`.

5. Log in to the Oracle Weblogic console and go to **Environment >Server > OIPA > Server Start** and add the details listed below to Arguments.

    - `-Duser.language=en -Duser.region=US -Djava.net.preferIPv4Stack=true -Djava.net.preferPv6Addresses=false -javaagent:C:\OIPA\lib\spring-instrument-3.1.0.RELEASE.jar -Dtangosol.coherence.override=C:\OIPA\conf\coherence-config.xml -Dtangosol.coherence.cacheconfig=C:\OIPA\conf\coherence-cache-config.xml -Dtangosol.pof.config=com-adminserver-pas-web-pof-config.xml -Djavax.net.ssl.trustStore=C:\OIPA\conf\wsse.truststore -Djavax.net.ssl.trustStorePassword=jbossws -Djavax.net.ssl.keyStore=C:\OIPA\conf\wsse.keystore -Djavax.net.ssl.keyStorePassword=jbossws`

6. Go to `WEBLOGIC_JAVA_SECLIB` and create a back-up of the **cacerts** file.

7. Create a new certification (cacerts) file by following the steps below.

    - Copy InstallCert.class and InstallCert$SavingTrustManager.class in `WEBLOGIC_JAVA_HOME\bin`.

    - From `WEBLOGIC_JAVA_HOME\bin`, run InstallCert through a command prompt **like java InstallCert localhost:7002.** The KeyStore jssecacerts will load and a connection will be opened. Messages will then be presented regarding the certificates.

    - When the process is complete, the following message will appear: **Enter certificate to add to trusted keystore or 'q' to quit**. Type **1** to continue.

    - When the process is complete, another message will appear: **Added certificate to keystore 'jssecacerts' using 'jssecacers' using alias 'localhost-1**'. Run java InstallCert localhost:7002 one more time, then enter **q** to exit. This will create a new **jssecacerts** keystore file in `WEBLOGIC_JAVA_SECLIB` and rename it to **cacerts**.

    Note: Repeat step 7 to enable SSL for different port numbers.

8. Stop the WebLogic application server (JVM, Node, Manager).

9. Restart the machine.

10. Start the WebLogic application server (JVM, Node, Manager).

11. Enter `https://machinename:7002/PASJava` in your Internet Explorer browser to access the login page of OIPA.

SSL in JBoss 5.1.0

Steps to Configure SSL/https:

1. Go to `JBOSS_HOME \server\default\deploy\jbossweb.sar` and edit **server.xml**. Add the details below by commenting the 8080 port number.

```
<!-- A HTTP/1.1 Connector on port 8080
      <Connector protocol="HTTP/1.1" port="8080" address="${jboss.bind.address}"
               connectionTimeout="20000" redirectPort="8443" />
-->
       <Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
               maxThreads="150" scheme="https" secure="true"
               clientAuth="false" sslProtocol="TLS" />
```

2. Restart the application server.

Steps to Configure Certificates:

```
32 bit JAVA_SECLIB = C:\Program Files\Java\jdk1.7.x\jre\lib\security
64 bit JAVA_SECLIB = C:\Program Files (x86) \Java\jdk1.7.x\jre\lib\security
JAVA_HOME =  C:\Program Files\Java\jdk1.7.x
```

Community Edition:

```
JBOSS_HOME = C:\jboss-5.1.0.GA
```

Enterprise Edition:

```
JBOSS_HOME = C:\jboss-eap-5.1\jboss-as
```

1. Install JBoss-5.1.0.GA or Redhat JBoss-eap-5.1.0.
2. Find the location of the JDK/JRE.
   - Go to an environment set up and find out the value of `JAVA_HOME`.
   - Go to run.bat.conf of the JBoss.
3. If JDK is not installed on your machine, then download and install latest update of 1.7 Oracle JDK.
   - Set up JDK in the Environment Variable. `JAVA_HOME= JDK location`.
4. Go to `JBOSS_HOME` and run the commands listed below.
   - `keytool -genkey -keystore server/default/conf/wsse.keystore -storepass jbossws -keyalg RSA -keysize 1024 -validity 1000 -alias localhost -dname "CN=localhost"`
   - `keytool -export -keystore server/default/conf/wsse.keystore -storepass jbossws -alias localhost -file server/default/conf/localhost.cer`
   - `keytool -import -keystore server/default/conf/wsse.truststore -storepass jbossws -trustcacerts -alias localhost -file server/default/conf/localhost.cer`

5. The above step will create two files within `JBOSS_HOME/server/default/conf`.
   - wsse.keystore
   - wsse.truststore
6. Edit run.bat.conf or run.sh and add the details listed below.
   - `set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStore= %JBOSS_HOME %\server\default\conf\wsse.truststore -Djavax.net.ssl.trustStorePassword=jbossws -Djavax.net.ssl.keyStore=%JBOSS_HOME % \server\default\conf\wsse.keystore -Djavax.net.ssl.keyStorePassword=jbossws"`
7. Go to `JBOSS_HOME \server\default\deploy\jbossweb.sar` and edit **server.xml**. Add the details listed below by commenting the 8080 port number.

```
<Connector protocol="HTTP/1.1" SSLEnabled="true"
 port="8443" address="${jboss.bind.address}"
 scheme="https" secure="true" clientAuth="false"
 truststoreFile="${jboss.server.home.dir}/conf/wsse.truststore"
 truststorePass="jbossws"
 keystoreFile="${jboss.server.home.dir}/conf/wsse.keystore"
 keystorePass="jbossws"
 sslProtocol = "TLS" />
```

8. Start the JBoss server and validate the steps listed below.
   - Navigate to https://localhost:8443/PASJava/FileReceived?wsdl browser should display WSDL.
   - Navigate to https://localhost:8443/PASJava to login.
9. Stop the JBoss server.
10. Go to `JAVA_SECLIB` and create a backup of the **cacerts** file.
11. Create a new certification (cacerts) file by following the steps listed below.
    - Copy InstallCert.class and InstallCert$SavingTrustManager.class in `JAVA_SECLIB`.
    - From `JAVA_SECLIB`, run InstallCert through a command prompt like **java InstallCert localhost:8443.** The KeyStore will load and a connection will be opened. Messages will then be presented regarding the certificates.
    - When the process is complete, the following message will appear: **Enter certificate to add to trusted keystore or 'q' to quit [1]**. Type **1** to continue.
      When the process is complete, another message will appear: **Added certificate to keystore 'jssecacerts' using alias 'localhost-1'**. Run java InstallCert localhost:8443 one more time, then enter **q** to exit. This will create a new jssecacerts keystore file in `JAVA_SECLIB` and rename it to **cacerts**.
12. Restart the machine.
13. Start the JBoss server.
14. Navigate to `https://machinename:8443/PASJava` in your Internet Explorer browser to access the login page of OIPA.

For more information please refer to the following websites:

http://docs.jboss.org/jbossweb/3.0.x/ssl-howto.html

http://docs.redhat.com/docs/en-US/JBoss_Enterprise_Portal_Platform/5.1/html/Installation_Guide/index.html

SSL in WebSphere 8.5.5.0

Version 8 of WebSphere Application Server, everything is done from the admin console, which includes a complete overview of the SSL management capabilities.

For more information about managing SSL in WebSphere please refer to the following website or follow the steps listed below.

http://pic.dhe.ibm.com/infocenter/wasinfo/v8r0/index.jsp

Note: Search for Overview and new features: Securing under Networkd Deployment

Steps to Configure SSL/https

1. Login to the WebSphere console.
2. Expand **Server Types** and click **WebSphere Application Servers**.
3. Click on the server that you created. Example: `OIPA_WILDCAT_10.0.0.0`
4. Expand Port and copy `WC_defaulthost_secure=`*`port number`*. This will be pasted in step 7.
5. From the left side menu expand **Environment** and click **Virtual Host**.
6. Click **default_host** and click **Host_Aliases**.
7. Click **New** and copy the port number from step 4, then click **OK**.
8. Restart the server/JVM.
9. Navigate to `https://machinename:9444/PASJava` in your Internet Explorer browser to access the login page of OIPA.

Steps to Configure Certificates

32 bit WebSphere Application Server

```
IBM_JAVA_SECLIB = C:\Program Files (x86)\ WebSphere\AppServer\java\jre\lib\security
IBM_JAVA_HOME = C:\Program Files (x86)\IBM\WebSphere\AppServer\java
```

64 bit WebSphere Application Server

```
IBM_JAVA_SECLIB = C:\Program Files\ WebSphere\AppServer\java\jre\lib\security
IBM_JAVA_HOME = C:\Program Files\IBM\WebSphere\AppServer\java
```

1. Download and install IBM JDK, if WebSphere is not installed on the machine.
   - URL to download http://www.ibm.com/developerworks/java/jdk/
2. Start the WebSphere application server
3. Enable SSL in WebSphere.
   - Log in to the WebSphere console.
   - Expand **Server Types** and click **WebSphere Application Servers**.
   - Click on the server that you created. Example: `OIPA_WILDCAT_10.0.0.0`
   - Expand Port and copy `WC_defaulthost_secure=`*`port number`*. This will be copied later in the process.

- From the left menu, expand **Environment** and click **Virtual Host**.
- Click **default_host** and then click **Host_Aliases**.
- Click **New** and copy the port number then click **OK**.
- Go to `IBM_JAVA_SECLIB\security` and comment the details below in the java.security file.

Note: Make sure to uncomment Default JSSE socket factories and comment WebSphere socket factories (in cryptosf.jar).

```
# Default JSSE socket factories
ssl.SocketFactory.provider=com.ibm.jsse2.SSLSocketFactoryImpl
ssl.ServerSocketFactory.provider=com.ibm.jsse2.SSLServerSocketFactoryImpl

# WebSphere socket factories (in cryptosf.jar)
#ssl.SocketFactory.provider=com.ibm.websphere.ssl.protocol.SSLSocketFactory
#ssl.ServerSocketFactory.provider=com.ibm.websphere.ssl.protocol.SSLServerSo
cketFactory
```

- Stop the server, Node Agent and Deployment Manager.
- Start the Deployment Manager, Node Agent and server.

4. Navigate to https://localhost:9445/PASJava in your Internet Explorer browser to make sure SSL works as expected.
5. Log in to the application. If this action is successful, then SSL is set up correctly from the server side.
6. Go to `IBM_JAVA_HOME\bin` and run the commands listed below.
   - `keytool -genkey -keystore ../lib/security/wsse.keystore -storepass jbossws -keyalg RSA -keysize 1024 -validity 1000 -alias localhost -dname "CN=localhost"`
   - `keytool -export -keystore ../lib/security/wsse.keystore -storepass jbossws -alias localhost -file ./lib/security/localhost.cer`
   - `keytool -import -keystore ../lib/security/wsse.truststore -storepass jbossws -trustcacerts -alias localhost -file ../lib/security/localhost.cer`
7. The step above will create two files within `IBM_JAVA_SECLIB.`
   - wsse.keystore
   - wsse.truststore
8. Move wsse.keystore and wsse.truststore to the **conf** folder where all properties files reside. For example: `C:\OIPA\conf`
9. Login to the WebSphere console, and go to **Application servers > OIPA > Process definition > Java Virtual Machine.** Add the arguments listed below to JVM.
   - -Duser.language=en -Duser.region=US -Djava.net.preferIPv4Stack=true -Djava.net.preferPv6Addresses=false -javaagent:C:\OIPA\lib\spring-instrument-3.1.0.RELEASE.jar -Dtangosol.coherence.override=C:\OIPA\conf \coherence-config.xml -

Dtangosol.coherence.cacheconfig=C:\OIPA\conf \coherence-cache-config.xml -
Dtangosol.pof.config=com-adminserver-pas-web-pof-config.xml -
Djavax.net.ssl.trustStore=C:\OIPA\conf\wsse.truststore -
Djavax.net.ssl.trustStorePassword=jbossws -
Djavax.net.ssl.keyStore=C:\OIPA\conf\wsse.keystore -
Djavax.net.ssl.keyStorePassword=jbossws

10. Go to `IBM_JAVA_SECLIB` and take a backup of the **cacerts** file.
11. Create a new certification (cacerts) file by following the steps listed below.
    - Copy InstallCert.class and InstallCert$SavingTrustManager.class in `IBM_JAVA_HOME\bin`.
    - From `IBM_JAVA_HOME\bin`, run InstallCert through a command prompt **like java InstallCert localhost:9445.** The KeyStore jssecacerts will load and a connection will be opened. Then messages will be presented regarding the certificates.
    - When the process is complete, the following message will appear: **Enter certificate to add to trusted keystore or 'q' to quit**. Type **1** to continue.
      When the process is complete, another message will appear: **Added certificate to keystore 'jssecacerts' using 'jssecacers' using alias 'localhost-1'**. Run java InstallCert localhost:9445 one more time, then enter **q** to exit. This will create a new jssecacerts keystore.

Note: Repeat step 7 to enable SSL for different port numbers.

12. Stop the WebSphere application server (JVM, Node Agent, Deployment Manager).
13. Restart the machine.
14. Start the WebSphere application server (JVM, Node Agent, Deployment Manager).
15. Navigate to `https://machinename: 9445/PASJava` in your Internet Explorer browser to access the login page of OIPA

JMS

JMS set-up is optional. It is only required if the Data Intake feature is being utilized. See the Data Intake document on OTN for additional details.

Data Intake is the process of receiving files from Group Customers for the purpose of importing data into the Oracle Insurance Policy Administration system (OIPA). The data in the files may result in many changes, including but not limited to the following:

- Adding a new member to the system
- Changing an Employee's elected coverage
- Adding a dependent to a coverage
- Enrolling a member and dependents
- Auto-cancelling coverage for a member
- Updating member information in the system

The received files are parsed and information about the data in them is put on a JMS queue. OIPA listens for messages on the queue and updates the business data based on pre-configured rules.

USER AUTHENTICATION

The OIPA application provides an out-of-the box user authentication mechanism as well as an ability to implement alternative authentication models like a Single Sign-On (SSO) authentication through the OIPA extensions. If the system is implemented with SSO, additional measures need to be taken to properly secure the authentication infrastructure. Depending on the implementation chosen, either an authentication server should be placed within the OIPA application server and database zone, or the call to an authentication service needs to be made via a secure connection.

Out-of-the box OIPA user authentication is performed for interactive users using web browsers to access the system, and for incoming web service calls. Interactive users are prompted on the application's login page to provide a username and password to authenticate to the server. Web services are protected with WS-Security, which requires incoming web service calls (which must be transmitted on a secure SSL connection to carry a security header with a user name and password.

Both web service and interactive user authentication are implemented through the same authentication service provided by the business logic tier of the OIPA application. The authentication service retrieves a matching user record from the OIPA database that contains basic user information and a secure digest of a password. The password digest is then compared to the digest of the incoming password and an authentication decision is made based on the result of the comparison. For certain web services, apart from user authentication, additional functional security is also enforced to control whether those services can be executed.

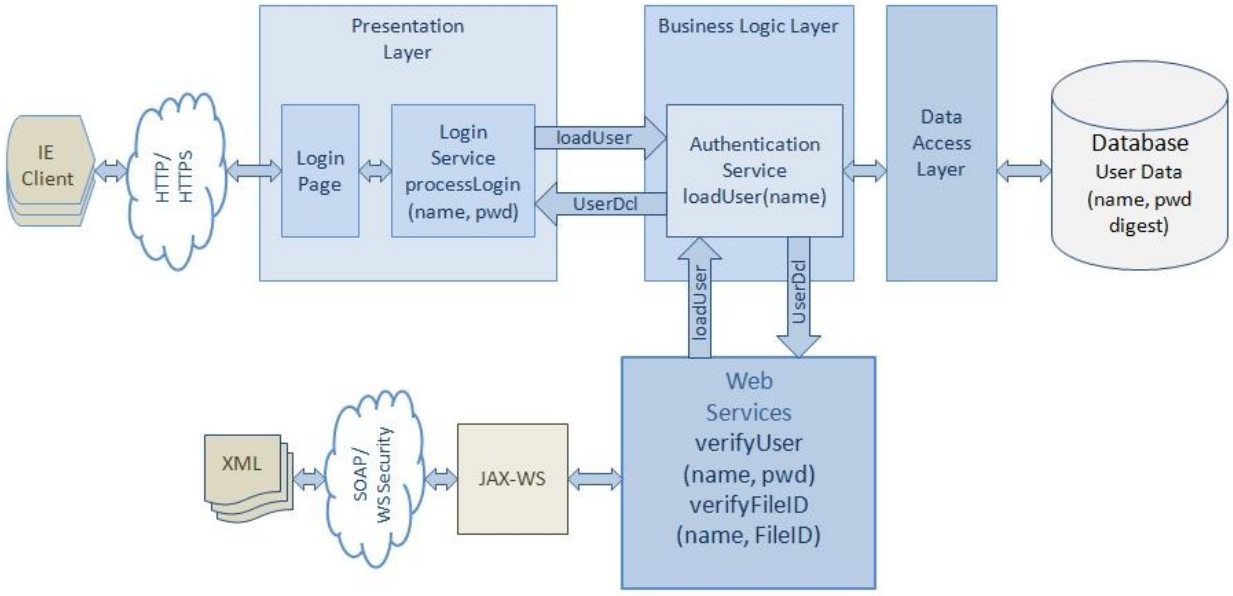

Figure 2. OIPA User Authentication

The encrypted password digest is created by the Rules Palette when a user is created. When a new OIPA environment is created using the Rules Palette's Web Application Utility, the process allows for the configuration of the encryption parameters to be used by the encryption algorithm. The settings include the particular encryption algorithm (from the list of the supported algorithms below), and the number of iterations of the algorithm.

- SHA-256
- SHA-384
- SHA-512

The number of encryption iterations is a value between 1000 and 9999. A higher number of iterations makes the password more secure, but also requires more computation to encrypt. For more information, please refer to the associated version of the Rules Palette Help System that is located on the Oracle Technology Network.

USER MANAGEMENT

User Registration

A user must have an existing OIPA user account identified by username and password to log into the OIPA application. An OIPA administrator uses the Rules Palette to create a new OIPA user account. The OIPA administrator's Rules Palette credentials must be associated with a security group that allows for the management of security. With the proper security rights, the administrator may use the Rules Palette to add, edit and delete OIPA user accounts. The administrator can also add and edit Security Groups that determine what features and authorizations are available to the users that belong to each Security Group. When creating a new user account, an administrator enters or selects the following information:

- User's login name and password
- Basic information about user – first and last name, email, gender, etc.
- User's primary company
- Locale
- Security groups to which the user belongs

This information is persisted in the OIPA database, with the encrypted password digest stored as discussed in the User Authentication section of this document.

There are no pre-existing or default user accounts or security groups in the OIPA application that need to be disabled after the system is deployed. The OIPA application user interface may be accessed only after at least one user account is created through the Rules Palette.

User Privileges and Group-Based Access Control

The OIPA user privileges and access restrictions implementation is based on the role-based access control (RBAC) model. According to the model, user permissions are assigned to specific groups or roles that are created for various job functions. A user who is assigned to a particular group gains permissions through those groups to perform particular system functions. If a user is assigned to multiple groups, the user will have access to all resources authorized for all of those groups.

For example, users that are assigned to the CSR group (or role) may not be able to execute such activities as issuing a policy or paying a death benefit. By contrast, a user in an Underwriter group should be able to issue a policy. A user in an administrator group is usually allowed access to all resources.

The following figure shows what application resources are protected by OIPA security.

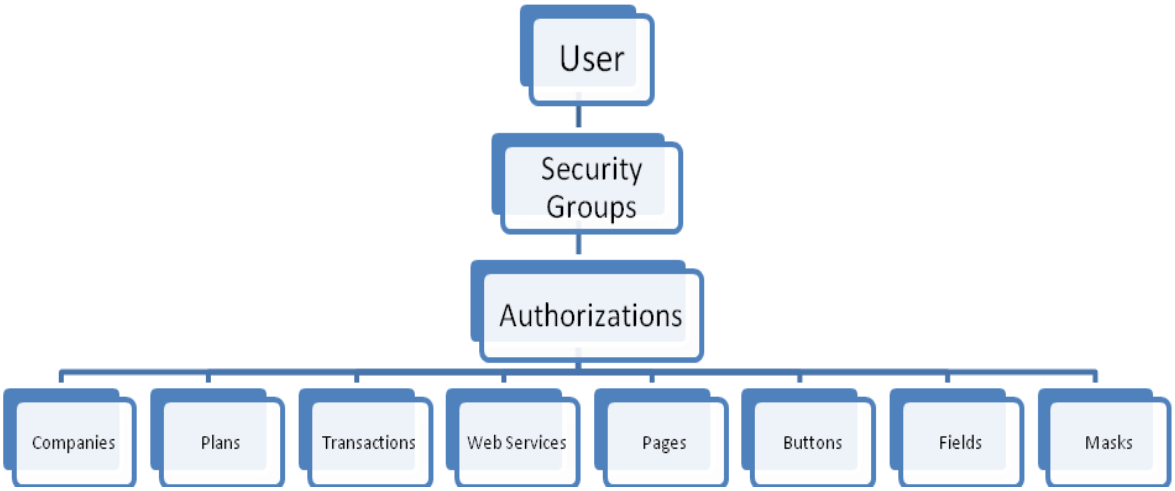

Figure 3. Hierarchy of User Authorizations

By default, a newly created user account does not have authorizations to access any of the application restricted resources. Authorizations have to be explicitly granted by an OIPA security administrator. When setting up the user groups, an administrator needs to be careful to include only the minimum set of permissions that allow users of a particular group to perform their job functions.

For more information on how to create security groups and manage user accounts please refer to the Rules Palette Help.

WEB SERVICES SECURITY

OIPA uses JAS-WS for implementing Web Services. For securing web services, WS-Security standards are used to perform authentication and authorization against OIPA user accounts. The SOAP header contains the appropriate security credentials. The password can be sent as a digest or as a text.

The SOAP header with WS-Security would look like the following when a password digest is used:

```
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Header>
      <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
         <wsse:UsernameToken wsu:Id="UsernameToken-1" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
            <wsse:Username>username</wsse:Username>
            <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-
token-profile-1.0#PasswordDigest">passwordencrypted</wsse:Password>
            <wsse:Nonce EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-
message-security-1.0#Base64Binary">kC5eI6iq8x17/qA3mzs6/g==</wsse:Nonce>
            <wsu:Created>2010-03-22T14:12:34.223Z</wsu:Created>
         </wsse:UsernameToken>
      </wsse:Security>
   </soapenv:Header>
```

For more information on the WS Security standard please refer to the website:

http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf

USING COOKIES IN THE OIPA APPLICATION

The OIPA application is accessed by users through Internet Explorer. Because OIPA uses session cookies to manage user sessions, cookies must be enabled in Internet Explorer. To allow the use of cookies in Internet Explorer, open the Privacy tab of the Internet Options dialog, then choose the Sites popup dialog and add the OIPA server address to the list of Allowed sites.

The *JSESSIONID* session cookie contains session ID generated for a user to manage data associated with the user's session. A unique session ID is generated when a user successfully logs into the OIPA application. The session ID is generated by the J2EE web server and passed to a browser as a non-persistent cookie. The browser retains it for the duration of the session, and deletes it when the user logs out or the session times out. During a session, when a browser issues a request back to the application server, it sends the session cookie in the HTTP header of the request. Requests that do not contain valid session IDs are not processed by the server.

The *ice.sessions* cookie is generated by the IceFaces library used by OIPA to implement the user interface. The cookie is a session-scope cookie used by IceFaces to maintain an IceFaces user session.

HIPAA CERTIFICATION WITH OIPA SOLUTIONS

The following are consideration points when seeking to achieve HIPAA certification using OIPA.

- **Unique User Identification** - OIPA supports being configured to run under a client defined SSO via extensions or with its own built in user ID and password support (see the OIPA Extensibility guide)
- **Controlling concurrent sessions** – Support for controlling concurrent sessions can be configured via an extension to OIPA that is similar to the extension returned to support SSO via a technique like this:
    - When a user logs into OIPA, the login extension will record the following information:
        - Username
        - Key that uniquely identifies the machine the user is logged in from. This could be an IP.
        - The number of times the user has logged in from the same machine. (Login Counter)
    - If the user has entered a valid username and password, the extension can check if the user is already logged into the system. If the user is logging in for the first time, the user can be given access to the system and the information listed above will be recorded. If the user has a prior active log in, and this log in attempt is from a new machine, access to the application can be denied. If the user has a prior active log in, and this log in attempt is from the same machine, the extension can check the maximum allowable concurrent sessions defined to the extension. If allowing access to the application will exceed that limit, access to the application can be denied. If it does not, the user can be given access to the system and the information listed above will be recorded with the session count incremented.
    - A logout extension will also be required to be triggered when a user session ends. When the session ends, the Login Counter for the appropriate user can be decremented.
- **Data visibility** - Masking support is available in OIPA's UI on text fields and field level security is available for all fields in the UI to control value visibility that can be configured to be dependent on user. Reports from OIPA via Oracle Business Intelligence Publisher are also configurable to provide compliant data masking. Masking does not extend to all search grids or activity result windows but the contents of these UI controls is configurable.
- **Data encryption while on the wire** - OIPA supports being configured to run on a secure socket layer (SSL) configuration. There is no other built in FTP or other non-HIPPA compliant network protocol support that cannot be disabled in an OIPA deployment.
- **Auditing** - OIPA has strong logging enabling auditing for activities and most UI actions.
- **Data encryption while at rest** - Full Disk Encryption for logs from OIPA's Web Application server and Transparent Data Encryption (TDE) that is part of the Oracle database provide the encryption required for data at rest.

ADDITIONAL SOURCES OF SECURITY INFORMATION

In addition to securing the OIPA application, all infrastructure resources –Linux/Windows servers, J2EE application and database servers – that compose an OIPA environment must be secured. The following list of links should be helpful while planning how to secure an OIPA environment.

Coherence 3.7.1 User Guide

http://docs.oracle.com/cd/E24290_01/index.htm

Oracle 11g Database

http://download.oracle.com/docs/cd/E11882_01/network.112/e16543/toc.htm

http://download.oracle.com/docs/cd/E11882_01/network.112/e10746/toc.htm

http://download.oracle.com/docs/cd/E11882_01/network.112/e10744/toc.htm

Microsoft SQL Server 2008 Database

http://www.microsoft.com/sqlserver/2008/en/us/Security.aspx

IBM DB2 10.1 Database

http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/index.jsp

Note: Search for DB2 Security model or Security.

Microsoft Windows 2008 Server

http://www.microsoft.com/download/en/details.aspx?id=17606

JBoss 5.1 J2EE Application Server

http://docs.jboss.org/jbossas/docs/Server_Configuration_Guide/5/html/index.html

Oracle WebLogic 10.3 J2EE Application Server

http://download.oracle.com/docs/cd/E12840_01/wls/docs103/security.html