Information Builders

iWay

Omni-Gen™ Security Guide

Version 3.15

September 07, 2020

Contents

Preface

This documentation describes how to configure security for Omni-Gen™. It is intended for Omni-Gen™ solution development teams.

How This Manual Is Organized

This manual includes the following chapters:

	Chapter/Appendix	Contents
1	Introduction and Architecture Overview	Provides an introduction to Omni-Gen web services security.
2	Enabling HTTPS, Strong Encryption Support, and Password Encryption	Describes how to enable HTTPS, strong encryption support, and password encryption.
3	Updating Security Certificates	Describes how to update security certificates.
4	Use Case Scenarios and Considerations	Describes use case scenarios and considerations.
A	Implementation for PCI Security Standards	Provides recommendations, information, and configuration steps for Omni-Gen to meet the Payment Card Industry Data Security Standards.

Documentation Conventions

The following table lists and describes the documentation conventions that are used in this manual.

Convention	Description
`THIS TYPEFACE` or `this typeface`	Denotes syntax that you must type exactly as shown.
this typeface	Represents a placeholder (or variable), a cross-reference, or an important term. It may also indicate a button, menu item, or dialog box option that you can click or select.
underscore	Indicates a default setting.

Convention	Description
Key + Key	Indicates keys that you must press simultaneously.
{}	Indicates two or three choices. Type one of them, not the braces.
|	Separates mutually exclusive choices in syntax. Type one of them, not the symbol.
...	Indicates that you can enter a parameter multiple times. Type only the parameter, not the ellipsis (...).
. . .	Indicates that there are (or could be) intervening or additional commands.

Related Publications

Visit our Technical Documentation Library at *http://documentation.informationbuilders.com*. You can also contact the Publications Order Department at (800) 969-4636.

Customer Support

Do you have questions about this product?

Join the Focal Point community. Focal Point is our online developer center and more than a message board. It is an interactive network of more than 3,000 developers from almost every profession and industry, collaborating on solutions and sharing every tips and techniques. Access Focal Point at *http://forums.informationbuilders.com/eve/forums*.

You can also access support services electronically, 24 hours a day, with InfoResponse Online. InfoResponse Online is accessible through our website, *http://www.informationbuilders.com*. It connects you to the tracking system and known-problem database at the Information Builders support center. Registered users can open, update, and view the status of cases in the tracking system and read descriptions of reported software issues. New users can register immediately for this service. The technical support section of *www.informationbuilders.com* also provides usage techniques, diagnostic tips, and answers to frequently asked questions.

Call Information Builders Customer Support Services (CSS) at (800) 736-6130 or (212) 736-6130. Customer Support Consultants are available Monday through Friday between 8:00 A.M. and 8:00 P.M. EST to address all your questions. Information Builders consultants can also give you general guidance regarding product capabilities. Be prepared to provide your six-digit site code (*xxxx.xx*) when you call.

To learn about the full range of available support services, ask your Information Builders representative about InfoResponse Online, or call (800) 969-INFO.

Help Us to Serve You Better

To help our consultants answer your questions effectively, be prepared to provide specifications and sample files and to answer questions about errors and problems.

The following table lists the environment information that our consultants require.

Platform	
Operating System	
OS Version	
JVM Vendor	
JVM Version	

The following table lists additional questions to help us serve you better.

Request/Question	Error/Problem Details or Information
Did the problem arise through a service or event?	
Provide usage scenarios or summarize the application that produces the problem.	
When did the problem start?	
Can you reproduce this problem consistently?	
Describe the problem.	

Request/Question	Error/Problem Details or Information
Describe the steps to reproduce the problem.	
Specify the error messages.	
Any change in the application environment: software configuration, EIS/database configuration, application, and so forth?	
Under what circumstance does the problem *not* occur?	

The following is a list of error and problem files that might be applicable.

- ❑ Input documents (XML instance, XML schema, non-XML documents)
- ❑ Transformation files
- ❑ Error screen shots
- ❑ Error output files
- ❑ Trace files
- ❑ Custom functions and agents in use
- ❑ Diagnostic Zip
- ❑ Transaction log

User Feedback

In an effort to produce effective documentation, the Technical Content Management staff welcomes your opinions regarding this document. Please use the Reader Comments form at the end of this document to communicate your feedback to us or to suggest changes that will support improvements to our documentation. You can also contact us through our website, *http://documentation.informationbuilders.com/connections.asp*.

Thank you, in advance, for your comments.

iWay Software Training and Professional Services

Interested in training? Our Education Department offers a wide variety of training courses for iWay Software and other Information Builders products.

For information on course descriptions, locations, and dates, or to register for classes, visit our website, *http://education.informationbuilders.com*, or call (800) 969-INFO to speak to an Education Representative.

Interested in technical assistance for your implementation? Our Professional Services department provides expert design, systems architecture, implementation, and project management services for all your business integration projects. For information, visit our website, *http://www.informationbuilders.com/consulting*.

Chapter 1

Introduction and Architecture Overview

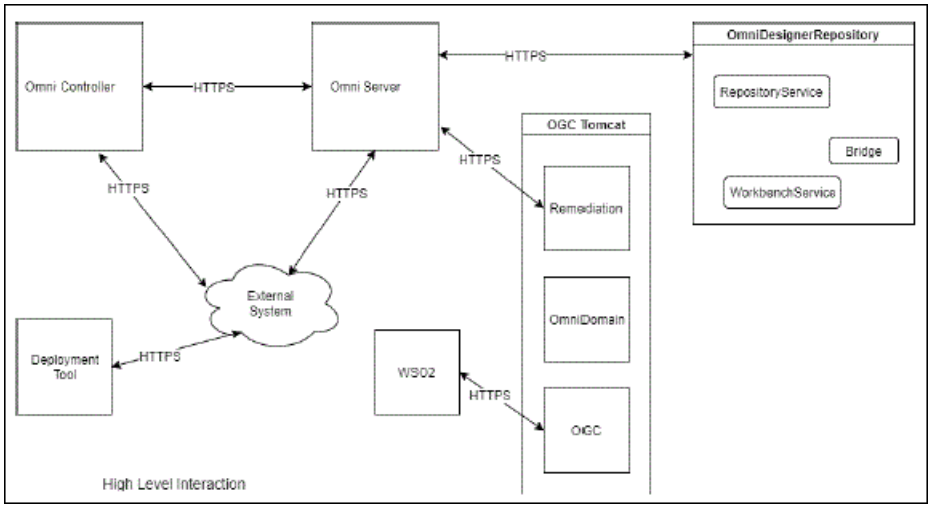
This section provides an introduction to Omni-Gen web services security.

In this chapter:

Overview

Omni-Gen consists of several applications that communicate with each other through web services. In addition, some of the web services are exposed to external systems. Therefore, the access to these web services needs to be secure and the data being transmitted must be encrypted. This document describes at a high level, the approach taken to secure web services using TLS or SSL, enforce strong encryption on the server side, and handle passwords. It also describes steps needed to import a CA-approved SSL certificate or create a self-signed certificate and import them to the keystores and truststores used by the application.

The following diagram illustrates the implementation.



Chapter 2

Enabling HTTPS, Strong Encryption Support, and Password Encryption

HTTPS is enabled on all the Omni-Gen applications with a web front end and/or those exposing RESTful web services.

To enable HTTPS, a signed SSL certificate must be used. Certificates are data files that digitally bind a cryptographic key to the details of an organization and to ensure that the content provided is from the correct (verified) sender. The following procedure describes the steps needed to enable HTTPS on the various Spring Boot applications. The TLS 1.2 protocol is enabled, by default, in the version of Spring Boot. Currently, 1.4.x is used.

Omni-Gen creates and uses a self-signed SSL certificate by default. The installation software captures the required parameters to create the certificate and configure the application software to use the certificate. This can be replaced with a CA-approved certificate.

In this chapter:

Understanding the Steps Required to Enable HTTPS

This section describes the steps that are required to enable HTTPS on the various Spring Boot applications that make up Omni-Gen.

1. Using a self-signed SSL certificate.

   A self-signed certificate is used, by default, and created at installation time. The parameters used depend upon the input provided during installation. The following syntax generates an omnigenstore keystore using the RSA algorithm with a key size of 2K with a new certificate. The application that needs to enable HTTPS references the keystore in its configuration.

```
keytool -genkey -alias boot -storetype PKCS12 -keyalg RSA -keysize 2048 -
keystore omnigenstore.p12 -storepass omnigen -noprompt -keypass omnigen -
validity 3650 -dname "cn=sr14386.ibi.com, ou=Omni, o=IBI, l=Rochester,
st=NY, c=US"
```

where:

`alias`

Specifies the certificate alias. By detault, this is set to *boot*.

`keystore`

Specifies the location or name of the keystore. This can be the file name with a fully qualified path.

`keypass`

Password used to protect the private key.

`dname`

Distinguished name associated with the alias and contains the server name.

`storepass`

Password used to protect the keystore.

The Omni-Gen installation will invoke this command (and commands in the following steps), with the appropriate arguments.

2. Using a CA-approved certificate.

   The CA-approved certificate can be imported into the omnigenstore keystore and the Omni-Gen applications reference the keystore. You can then import the certificate, which is described in *Importing an External Certificate* on page 16.

3. Exporting the certificate into a PEM file.

   You must create the actual certificate for the client applications using the keytool. The intermediate encoded file is created in order to create the truststore for the client applications (external or internal Omni-Gen applications). For example:

```
keytool -export -alias boot -keystore omnigenstore.p12 -storepass
omnigen -noprompt -file omnigenstore.pem
```

4. Enabling HTTPS in Spring Boot.

   To enable HTTPS, the Spring Boot applications need to be configured by setting the SSL parameters and pointing them to the keystore (created in step 3). The following properties need to be set:

   ```
   server.port = 9500
   server.ssl.enabled=true
   server.ssl.key-store = omnigetstore.p12
   server.ssl.key-store-password = omnigen
   server.ssl.keyStoreType = JKS
   server.ssl.keyAlias = boot
   ```

   Note: The Spring Boot application understands these properties, which are exposed through the installation software and its associated configuration file differently.

5. Redirecting HTTP to HTTPS.

   This is done by adding another Tomcat connector programmatically. It is configured as an HTTP connector that redirects all the traffic to the earlier configured HTTPS connector and entails adding a TomcatEmbeddedContainerFactory bean to one of the @Configuration classes. This allows supporting both HTTP and HTTPS or enabling the redirect.

   These steps ensure the web services exposed by the application can be accessed over HTTPS.

Consuming HTTPS

The applications must be able to consume the web services over HTTPS. When acting as a client, the certificate created or used earlier must be added to the Java truststore. This requires importing the certificate into the truststore using the keytool, as shown below:

```
keytool -import -alias boot -keystore ibi-cacerts -storepass boot -noprompt
-file omnigenstore.pem
```

The application is made aware of the certificate by setting the javax.net.ssl.trustStore property. This is added as a Java argument when invoked. For the applications running on Apache Tomcat (OGC, OmniDesignerRepository), this is added to CATALINA_OPTS.

UI/Configuration

The Common Name (CN) for the self-signed certificate is the fully qualified host name. The Omni-Gen installer UI captures the host name and domain, along with all the elements of the distinguished name, the keystore, and truststore locations, as part of the configuration. This is then used to build the self-signed certificate.

Importing an External Certificate

Scripts for Linux and Windows are included (in the scripts folder) to import a CA-approved certificate into the omnigenstore keystore. The following syntax shows the format.

```
importCert <certificate> <password> <key_alias>
```

Supporting Strong Encryptions

In addition to the basic privacy, integrity, and protection for the data that is transmitted between the client and the server, strong encryptions refer to a TLS implementation which provides all of the following:

- Perfect Forward Secrecy, which ensures that a compromise to the private key of a server in the present does not compromise the confidentiality of past TLS communications.
- Protection from known attacks on older SSL and TLS implementations, such as POODLE and BEAST.
- Support for the strongest ciphers available to modern web browsers and other HTTP clients.
- Rejection of clients that cannot meet these requirements.

The following configuration is in place to support this.

```
SSLProtocol=TLSv1.2
# Supported Ciphers
SSLCipherSuite=ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-
SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-
ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-
SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-
SHA256
SSLHonorCipherOrder=on
SSLCompression=off
SSLSessionTickets=off
```

Password Encryption

Passwords used by Omni-Gen are stored in property and configuration files. The files are in plain text, but the password fields are encrypted using 128-bit AES encryption.

The following syntax shows a sample password field entry.

```
server.runtime.ssl.keystore-file = ${server.runtime.dataDirectory}/
omnigenstore.p12
server.runtime.ssl.keystore-password =
yc4QxoL5oCAqqEHn11e91Q==:NqG0dkZKuVW2RSgxqsi/eQ==
```

The application is responsible for encrypting and decrypting the password fields, prior to use.

Chapter 3

Updating Security Certificates

This section describes how to update security certificates.

In this chapter:

Overview

If you need to update the default security certificate with a different certificate (for example, a certificate approved by a Certificate Authority), then you must import the certificate along with the private key into the keystore. Sample scripts for Windows and Linux are available below for reference.

If you are copying the script directly from this document, consider the fact that whitespace characters might be distorted, requiring you to reformat the script. This will be streamlined in future releases.

To update the security certificates:

1. Copy the security certificate file and paste it in the \OmniGenData folder.

   This file must be in PKCS#12 (or PFX) format. If it is in PEM format, then it must be converted.

2. Create the script and copy it to the \OmniGenData folder.

   The exact location of the script will change in future releases.

3. Run the script, which takes the following three arguments:

   - Source keystore (certificate)
   - Keystore password
   - Source alias

Sample Script for Windows

The following is the sample script for Windows.

```
@set KT="%JAVA_HOME%\bin\keytool"
@set OMNIGENDATA=..\OmniGenData

@if "%2" == "" goto args_count_wrong
@if "%3" == "" goto args_count_wrong
@if "%4" == "" goto args_count_ok

:args_count_wrong
@echo Invalid parameters. Usage: import.cmd srckeystore srcstorepass
srcalias
@exit /b 1

:args_count_ok

cd %OMNIGENDATA%
@del /Q omnigenstore.p* ibi-certs

%KT% -importkeystore ^
-srckeystore %1 -destkeystore omnigenstore.p12 ^
-srcstorepass %2 -deststorepass omnigen ^
-srcalias %3 -destalias boot ^
-srcstoretype pkcs12 -deststoretype JKS ^
-destkeypass omnigen ^
-noprompt

%KT% -exportcert -alias boot -keystore omnigenstore.p12 -storepass omnigen -
keypass omnigen -noprompt -rfc -file omnigenstore.pem
%KT% -importcert -alias boot -keystore ibi-certs -storepass changeit -
noprompt -file omnigenstore.pem

%KT% -delete -alias boot -keystore OmniGovConsole\data\security\client-
truststore.jks -storepass wso2carbon -noprompt
%KT% -importcert -alias boot -keystore OmniGovConsole\data\security\client-
truststore.jks -storepass wso2carbon -noprompt -file omnigenstore.pem

cd ..\scripts
```

Sample Script for Linux

The following is the sample script for Linux.

```
#!/bin/sh

KT=$JAVA_HOME/bin/keytool
OMNIGENDATA=../OmniGenData

EXPECTED_ARGS=3
E_BADARGS=65

if [ $# -ne $EXPECTED_ARGS ]
then
  echo "Invalid parameters. Usage: `basename $0` srckeystore srcstorepass
srcalias"
  exit $E_BADARGS
fi

cd $OMNIGENDATA
rm -rf omnigenstore.p* ibi-certs

$KT -importkeystore \
-srckeystore $1 -destkeystore omnigenstore.p12 \
-srcstorepass $2 -deststorepass omnigen \
-srcalias $3 -destalias boot \
-srcstoretype pkcs12 -deststoretype JKS \
-destkeypass omnigen \
-noprompt

$KT -exportcert -alias boot -keystore omnigenstore.p12 -storepass omnigen -
keypass omnigen -noprompt -rfc -file omnigenstore.pem
$KT -importcert -alias boot -keystore ibi-certs -storepass changeit -
noprompt -file omnigenstore.pem

$KT -delete -alias boot -keystore ./OmniGovConsole/data/security/client-
truststore.jks -storepass wso2carbon -noprompt
$KT -import -alias boot -keystore ./OmniGovConsole/data/security/client-
truststore.jks -storepass wso2carbon -noprompt -file omnigenstore.pem

cd ../scripts
```

Chapter 4

Use Case Scenarios and Considerations

HTTPS requires the creation and installation of signed certificates. For Omni-Gen applications, self-signed certificates are used. The steps are the same using a signed certificate from a Certificate Authority (CA). Depending on whether the individual applications run locally or on different machines, the certificate may need to be installed on one or more machines.

In this chapter:

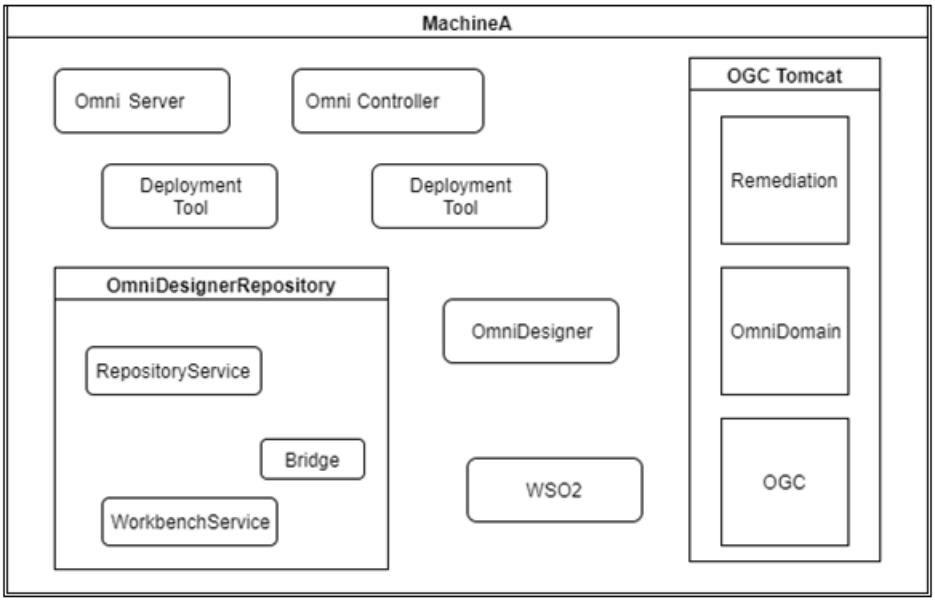
Basic Requirements

The following list describes the basic requirements for the use case scenarios and considerations.

- All Omni-Gen applications use the omnigenstore keystore, which is created by the installation software.
- Omni-Gen applications that need to communicate with HTTPS-enabled applications use the ibi-certs truststore.
- The keystore, pem, and truststore files are in the OmniGenData directory.

Installing the Application on a Single Host

All applications are running on a single host. The following image shows the workflow example behind the application running on Machine A.

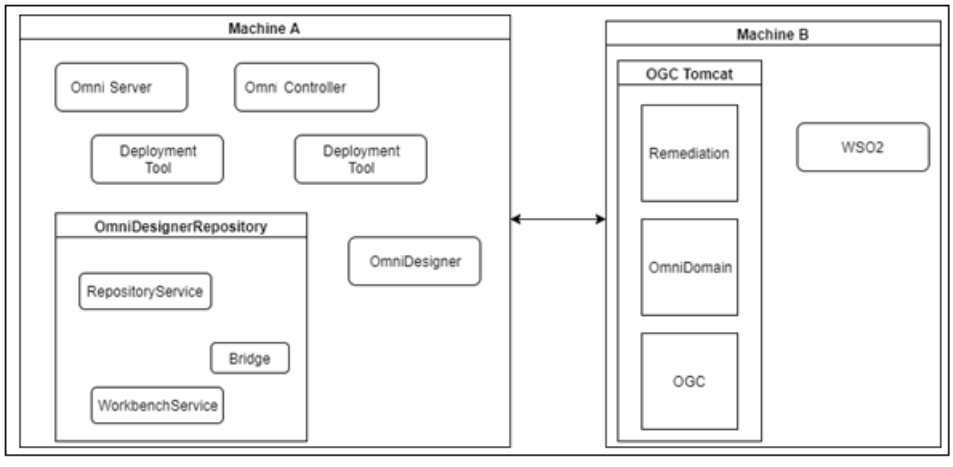


The following list describes how the application is installed on a single host.

- The installation process captures the installation type (single host and multiple hosts), along with the fully qualified host names.
- It collects the locations of the keystore and truststore.
- By default, the cert directory is the OmniGenData directory.
- Installation software creates the omnigenstore keystore, omnigenstore cert file, and the ibi-certs truststore in the cert directory.
- If multiple certificates need to be included (for example, in OGC), separate truststores are used. This is handled by the installation software.
- The fully qualified host name is used when accessing each web service.

Deploying on Multiple Hosts

Applications must be run on different hosts. In the following example, OGC Tomcat and WSO2 are running on Machine B while the Omni Server and the other applications are running on Machine A.



The following list describes how the application is deployed on multiple hosts.

- The installation process captures the installation type (single host and multiple hosts), along with the fully qualified host names.
- It collects the locations of the keystore and truststore.
- By default, the cert directory is the OmniGenData directory.
- Installation software creates the omnigenstore keystore, omnigenstore cert file, and the ibi-cert truststore in the cert directory on Machine A.
- Installation software remote copies the certificate files to the cert directory on Machine B and creates the required truststore locally (on Machine B).
- The invoker software that starts up the application references the local truststore.
- The fully qualified host name is used when accessing web services.

The same model is followed for any other variations.

Appendix A

Implementation for PCI Security Standards

This topic provides recommendations, information, and configuration steps for Omni-Gen to meet the Payment Card Industry Data Security Standards that are outlined in the PCI DSS Version 3.0 document. This document is located at:

https://www.pcisecuritystandards.org

Customers can use this information to implement the required steps for PCI compliance.

Information Builders is committed to work in partnership with our customers, to further develop the standards in anticipation of future Omni-Gen versions and changes in the PCI Security Standards.

In this appendix:

About the PCI Security Standards

The Payment Card Industry (PCI) Data Security Standard (DSS) was developed to encourage and enhance cardholder data security, and to facilitate the broad adoption of consistent data security measures globally. PCI DSS provides a baseline of technical and operational requirements that are designed to protect cardholder data. PCI DSS applies to all entities involved in payment card processing. This includes merchants, processors, acquirers, issuers, and service providers, as well as all other entities that store, process, or transmit cardholder data. PCI DSS comprises a minimum set of requirements for protecting cardholder data, and may be enhanced by additional controls and practices to further mitigate risks. The twelve requirements and subrequirements for PCI DSS compliance apply to all system components around technology and security, particularly that of the protection of cardholder data.

Build and Maintain a Secure Network and Systems

The following are recommendations and information for the Build and Maintain a Secure Network and Systems requirements.

Requirement 1: Install and maintain a firewall configuration to protect cardholder data

- The Omni-Gen products should be installed on an internal (trusted) network segment.
- Data acquisition channels for bringing data into the Omni-Gen on-ramps should be configured through the Demilitarized Zone (DMZ). If the client is using the Information Builders iWay 8 product for integration services on the data acquisition, then the PCI compliance chapter in that document will provide more information on configuring Integration channels within and outside of the DMZ.
- TCP/IP listener ports are required for the Omni server and applications to communicate internally and externally. The ports are configurable by the user and can be changed during the product installation.

Omni-Gen Ports

The following table lists the default Omni-Gen ports and their use.

Component	Type	Port	Security
Omni Controller/Console	external	9500	https
Omni Server	internal	9514	https

Component	Type	Port	Security
Omni Server Data Quality High-Speed TCP	internal	9532	TLS 1.0
Data Quality Cleanse	external	9504	https
Data Quality Cleanse	internal	9505	TLS 1.0
Data Quality Match	external	9506	https
Data Quality Match	internal	9507	TLS 1.0
Data Quality Merge	external	9508	https
Data Quality Merge	internal	9509	TLS 1.0
Data Quality Remediation	external	9510	https
Data Quality Remediation	internal	9511	TLS 1.0
OGC Tomcat Shutdown	internal	9524	TLS 1.0
OGC WSO2	external	9503	https
OGC Tomcat Console	external	9526	https
WSO2 RMI Registry	internal	9534	WSO2 Config
WSO2 RMI Server	internal	9535	WSO2 Config
WSO2 LDAP Server	internal	9536	WSO2 Config
WSO2 KDC Server	internal	9537	WSO2 Config
Omni Designer TCP Shutdown	internal	9515	TLS 1.0
Omni Designer Console	external	9516	https
Omni Designer Redirect	internal	9518	https
Omni Designer TCP Jmx	internal	9519	Tomcat Config
Omni Designer EMF	internal	9520	https
Deployment Console	external	9521	http

Component	Type	Port	Security
Deployment Console	external	9502	https
GIT/SVN	external	80/(8800,8443)	Repository Config

Requirement 2: Do not use vendor-supplied defaults for system passwords and other security parameters

Omni-Gen consists of several components. It is strongly advised to change all default credentials to client controlled and maintained credentials.

The user is advised not to install any unrelated components, scripts, jars, or any other files on the production systems, other than the ones required for the product to run. The client is also advised to disable any Omni components not in use to prevent accidental and unintended access.

- **Omni Server Console.** This is utilized for operations and monitoring. It is meant for the internal operations user and not for external communication. The console can be disabled, if needed, and other operation monitoring components can be used. The user is advised to change the default log in for the Omni Server Console, regardless of their plan on using this component.
- **Omni Governance Console.** This is a business user-facing interface. The security for the end user is managed by the available Tomcat and WSO2 configurations. The user is advised to change the default settings for accessibility and create different roles for different types of users to prevent unintended data access.
- **Omni Designer.** This is a developer tool for creating a Master Data Management model and is required only during the development time. The Omni Designer should not be running in a production environment. It utilizes integration with the source management system (SVN/GIT), which provides for user accessibility.

Protect Cardholder Data

The following are recommendations and information for the Protect Cardholder Data requirements.

Requirement 3: Protect stored cardholder data

Recommendations and Information for Requirement 3.1

The source system data should not be exposed directly to Omni-Gen for processing. The source data, which the client has residing in the existing infrastructure, should be protected based on the existing client needs. As the data is presented (on-ramped) into Omni-Gen, the client should select which data is required for processing, and any sensitive data should be properly masked. The client is advised to limit the intermediate storage of data and protect direct access to the data store.

Recommendations and Information for Requirement 3.2, 3.3, and 3.4

Developers and Operations users responsible for creating Omni Governance views into the data, across the Omni 360 Viewer and Omni Remediation, are responsible for ensuring that sensitive data is not presented to an unauthorized user. In cases where the data should be presented to the user, the data or part of the sensitive data should be masked according to guidelines.

For critical and sensitive data, which rests in the data store, the client is advised to follow the data store (database or such) specific PCI compliance instructions for protecting the data store from unintended access.

Requirement 4: Encrypt transmission of cardholder data across open, public networks

The client is advised to use the TLS and proper WSO2 role-based configuration for consumer facing applications, such as Omni Governance Console. The sensitive data should not be transmitted and presented to the Omni Governance Console, unless the user is within the trusted network and the protection of the sensitive data can be guaranteed by the network configuration.

Maintain a Vulnerability Management Program

The following are recommendations and information for the Maintain a Vulnerability Management Program requirements.

Requirement 5: Protect all systems against malware and regularly update anti-virus software or programs

Requirements are not applicable to the Omni-Gen product line.

Requirement 6: Develop and maintain secure systems and applications

Recommendations and Information for Requirement 6.1

- ❑ Ensure that the latest Omni-Gen service packs and patches are applied. For the latest service packs and patches, see *http://techsupport.ibi.com*.
- ❑ Third-party software provided by Omni-Gen, such as Tomcat, should be updated as recommended by those vendors.
- ❑ Third-party software which is not provided, but is required by Omni-Gen, such as Java, should be updated as recommended by those vendors.

Recommendations and Information for Requirement 6.3

- ❑ Adhere to the Internal Software Development Life Cycle (SDLC) recommendations for application development to ensure that any customizations do not introduce new vulnerabilities.
- ❑ Remove any test accounts created during development prior to a production rollout.
- ❑ Remove any test jars or scripts used during the development life cycle.
- ❑ If any custom code is used, the client is responsible for reviewing the code for vulnerabilities.

Recommendations and Information for Requirement 6.4

- ❑ Create separate Omni-Gen environments for Development, Performance, Production, and any other use, to ensure separation and accessibility. Ensure to use different repositories and authentication/authorization domains.
- ❑ Do not develop any components directly on the Performance or Production systems. The Production system should be a code-frozen environment with the only exception where a debug component may need to be installed for issues which are encountered in production, but are not able to be reproduced in any other non-production environment. Such debugging components would be provided by Information Builders as part of the support for the Production Issues.
- ❑ It is recognized that for Master Data Management applications, the production data may need to be used in the Test and Performance environments, in addition to the Production environment. In such cases, the client is advised to limit the access to the data and such environments. The developers should not have access to the production data and should work only with non-production/simulated data.
- ❑ Remove all test accounts and test data from the Production environment.

- ❑ Establish a process of installing service packs and patches across environments, as well as the roll-back procedures, based on the instructions provided in the Omni-Gen Installation manual and Omni-Gen Release Notes for the corresponding patch or service pack.

Recommendations and Information for Requirement 6.5 and 6.6

- ❑ Follow the best practices and guidelines provided by Information Builders for the development and maintenance of the applications.
- ❑ Any public-facing application, which exposes parts of the Omni-Gen data, should not be connecting to the live master data repository, but rather should be presenting the data off the generated consumption view layer, thus minimizing the data access and cross contamination.
- ❑ Any customized applications, which are written utilizing the available RESTful APIs, are the responsibility of the client. The client must perform regular web application vulnerability assessments and/or install external firewalls.
- ❑ The client is responsible for ensuring that any data access to the Mastered Data repository is under their full control and no external application can access this data without proper authentication/authorization.

Implement Strong Access Control Measures

The following are recommendations and information for the Implement Strong Access Control Measures requirements.

Requirement 7: Restrict access to cardholder data by business need to know

- ❑ Omni-Gen user-facing applications, such as Omni Governance Console, utilize WSO2 for secure access. The user is advised to follow the WSO2 guidelines for role and user management or enable the integration with existing security system such as LDAP/AD using WSO2. For detailed information, see the WSO2 user manual and the *Omni Governance Console User's Guide*.
- ❑ Omni-Gen operations interfaces, such as Omni-Gen Server Console and Deployment Console, utilize internal user authentication and should be made available only to a given operations user able to access the system-level information only.
- ❑ Omni-Gen developer tools, such as Data Quality Server and Omni Designer, should be granted access to developers only. The client is advised to utilize a source management system for user management.

- The Administrative users who are authorized to assign roles and manage user access should be given proper training on what components are required to be accessed by which role-based user. This information should be documented and referred to.

Requirement 8: Identify and authenticate access to system components

Recommendations and Information for Requirement 8.1

Omni-Gen does not provide an internal user management facility, but instead uses externalized systems, such as AD/LDAP, WSO2, and Source Management, for user access. The client is advised to refer to the available documentation for the user management aspect based on the utilized component. Integration with the corporate level systems, such as Active Directory (AD), should ensure that user access is automatically synchronized across corporate access and Omni-Gen Governance access, eliminating the need for double maintenance.

Recommendations and Information for Requirement 8.2

Access to the user management systems themselves should be made available only to vetted administrators who are trusted to have access to such systems. The monitoring of any user administrative tasks, such as the addition of a user or the altering of user roles should be done based on client requirements.

Recommendations and Information for Requirement 8.7

Any access to the data sources, which may contain sensitive information, shall be managed and restricted by the client network and security policies in place outside of the Omni-Gen product.

Any direct access to the Omni-Gen database repositories shall be protected by the client's existing security model, ensuring that only approved users can get direct access. The physical systems where the data may rest in place, shall be protected by the network security model following the client requirements.

Any externalization of data to the outside non-Omni-Gen consumer, such as customized application and reports, should be done by creating a layer of abstraction-like Consumption Views to limit or filter authorized data to be exposed. The Omni-Gen Consumption View builder enables the client to generate a slice of data for a specific type of end user application, such ensuring that no sensitive data is included, unless the end user application is authorized to access it.

Requirement 9: Restrict physical access to cardholder data

Requirements are not applicable to the Omni-Gen product line.

Regularly Monitor and Test Networks

The following are recommendations and information for the Regularly Monitor and Test Networks requirements.

Requirement 10: Track and monitor all access to network resources and cardholder data

Omni-Gen Server Operations Console provides information on current and prior configuration values, enabling the user to revert back to a prior configuration. The Operations section of the Console can be used to monitor system health and audit information. It provides detailed information on:

- Work order execution showing what data subjects have been processed with the statistics measures.
- Number of records, transactions, and sources, being processed with statistics measures.
- Database accessibility and latency, as well as specific query execution and measures.
- Network information and access.
- Diagnostic log files with detailed information for data processing.
- Performance indicators across the system components.

Omni Governance Console, which is an end user facing tool, provides auditing of the user access, and utilizes the WSO2 security model.

The following log files are available for diagnostics and audit purposes. For convenience, commonly-used log files generated by most Omni-Gen processes can be found in the *OmniGenData/logs* directory, inside the Omni-Gen installation directory. The log files are further organized into the following subdirectories based on the process that generated them:

- **bundler.** Deployment bundle service logs.
- **command.** Output from any omni shell command.
- **controller.** Omni Controller service logs.
- **dq.** Logs from the Data Quality services.
- **OGC.** OGC Tomcat standard output.
- **OmniDesignerRepository.** All repository service Tomcat logs (including web applications).
- **server.** Omni server logs.

In some cases, more detailed logs or output data can be found in the following locations:

- **deploymentbundle.** Saved copies of deployed bundles.
- **deploymentbundle/logs.** Zipped archives of deployment bundle service logs.
- **install/Omnigen_install_logs.** Installer logging and debug output.
- **OmniDesignerRepository/webapps/Bridge/WEB-INF/lib/configuration.** EMF bridge web application detail messages.
- **OmniGenData/deployment.** Detailed deployment event timings.
- **OmniServer/dbms/changelogs.** Most recent LiquiBase migration changesets.
- **OmniGovConsole/log.** OGC Tomcat and web application logs.
- **wso2_is/repository/logs.** WSO2 server logs.

Requirement 11: Regularly test security systems and processes

Recommendations and Information for Requirements Section 11.5

The client is advised on preventing any unauthorized direct access to the file system where the Omni-Gen product is installed and to ensure that the system is protected from any unintended access. The client is also advised not to modify or edit any of the files directly, unless instructed by Information Builders personnel in written form.

Maintain an Information Security Policy

The following are recommendations and information for the Maintain an Information Security Policy requirements.

Requirement 12: Maintain a policy that addresses information security for all personnel

Requirements are not applicable to the Omni-Gen product line.

Feedback

Customer success is our top priority. Connect with us today!

Information Builders Technical Content Management team is comprised of many talented individuals who work together to design and deliver quality technical documentation products. Your feedback supports our ongoing efforts!

You can also preview new innovations to get an early look at new content products and services. Your participation helps us create great experiences for every customer.

To send us feedback or make a connection, contact Sarah Buccellato, Technical Editor, Technical Content Management at *Sarah_Buccellato@ibi.com.*

To request permission to repurpose copyrighted material, please contact Frances Gambino, Vice President, Technical Content Management at *Frances_Gambino@ibi.com.*

Information Builders

iWay

Omni-Gen™ Security Guide

Version 3.15

DN3502331.0920

Information Builders, Inc.
Two Penn Plaza
New York, NY 10121-2898