



TIBCO® Offer and Price Engine

Security Guidelines

Version 6.0.0
March 2023



Contents

Contents	2
About This Product	3
Introduction	4
Security Features	5
Security Vulnerabilities	6
Ensuring TIBCO Offer and Price Engine Security	7
	7
Configuring SSL for TIBCO Offer and Price Engine	7
Configuring on Cloud	7
Configuring On-premise	11
Encrypt Password Utility	14
TIBCO Documentation and Support Services	15
Legal and Third-Party Notices	17

About This Product

TIBCO® Offer and Price Engine is a cloud-native, in-memory omnichannel server of offers and prices for digital service providers. It answers requests from a digital service provider's customer-facing channels for offers and prices, subject to business rules such as customer eligibility and product compatibility.

TIBCO Offer and Price Engine is the next generation of, and partially replaces, TIBCO® Fulfillment Order Management. To better align the direction of TIBCO Fulfillment Order Management with market demand, the product's capabilities have been reorganized into two new products:

- TIBCO® Order Management
- TIBCO® Offer and Price Engine

Customers who are currently on maintenance for TIBCO Fulfillment Order Management are entitled to upgrade to both TIBCO Order Management and TIBCO Offer and Price Engine. TIBCO will continue to support TIBCO Fulfillment Order Management, and there is currently no plan to retire TIBCO Fulfillment Order Management. New capabilities will be developed in TIBCO Order Management and TIBCO Offer and Price Engine.

Introduction

This document describes guidelines to ensure security within the various components of TIBCO Offer and Price Engine. It also provides additional security-related guidance and recommendations for other aspects of internal and external communication. In particular, this document provides details of product connectivity and configuration of security options.

Security Features

TIBCO® Offer and Price Engine includes the following security features.

- Secure transports for communications among application peer processes.
- HTTPS for secure connections.
- Authorization Service

Security Vulnerabilities

This topic describes the key security technologies for TIBCO Offer and Price Engine. In addition to these key technologies, security also depends in parts on the correct configuration and use of its components and capabilities.

OpenSSL

Security features that protect TIBCO Offer and Price Engine connections and communications depend on the implementation of OpenSSL. If the security of OpenSSL were compromised, TIBCO Offer and Price Engine and applications that use TIBCO Offer and Price Engine could be vulnerable as well.

Ensuring TIBCO Offer and Price Engine Security

To ensure security within and among the components of TIBCO Offer and Price Engine, the following security provisions are provided.

- Authorization Service: See the 'Authorization Service' section in the *TIBCO® Offer and Price Engine User Guide*.
- [Configuring SSL for TIBCO Offer and Price Engine](#)
- [Encrypt Password Utility](#)

Configuring SSL for TIBCO Offer and Price Engine

The Configuration of SSL for TIBCO Offer and Price Engine is available for both [on cloud](#) and [on-premise](#).

Configuring on Cloud

The following section is added for testing purposes and is not recommended for the production environment. Currently, ingress is configured with SSL only for authorization service as a backend.

Procedure

1. To create a root certificate, run the following command:

```
openssl req -x509 -nodes -sha256 -days 365 -newkey rsa:2048 -subj
"/CN=test/O=TIBCO"
-keyout lab-caroot.key -out lab-caroot.crt
```

2. To create CSR for a service certificate, run the following command:

```
openssl req -out ope-auth.csr -newkey rsa:2048 -nodes -keyout ope-
auth.key -subj "/CN =
ope-auth.test / O=auth-svc organization"
```

3. To sign the certificate with the root CA, run the following command:

```
openssl x509 -req -days 365 -CA lab-caroot.crt -CAkey lab-
caroot.key -set_serial 0 -
in ope-auth.csr -out ope-auth.crt
```

4. To create K8s secret, run the following command:

```
kubectl create secret tls tls-ope-auth --key=ope-auth.key --
cert=ope-auth.crt
```

5. Add the secrets in the auth ingress yaml file:

```
```yaml
tls:
- hosts:
- ope-auth.test # This should match a DNS name in the Certificate
secretName: tls-ope-auth # This should match the Certificate
secretName
```

## Enabling SSL for TIBCO Offer and Price Engine

### Procedure

1. Go to the JAVA\_11\_HOME\bin directory and run the following commands:

```
C:\jdk11\bin>keytool -genkey -alias ope -keyalg RSA -keysize 2048 -
sigalg SHA256withRSA -validity 365 -keystore ope.pkcs12 -storepass
```

```

tibco123 -ext san=dns:configurator-
svc.default.svc.cluster.local,dns:authorization-
svc.default.svc.cluster.local,dns:ope-
svc.default.svc.cluster.local,dns:localhost,dns:orchestrator-
svc.default.svc.cluster.local,dns:aopd-
svc.default.svc.cluster.local,dns:archival-
svc.default.svc.cluster.local,dns:pc-
svc.default.svc.cluster.local,dns:jeopardy-
svc.default.svc.cluster.local
What is your first and last name?
[Unknown]: ope-auth.test
What is the name of your organizational unit?
[Unknown]: tibco
What is the name of your organization?
[Unknown]: tibco
What is the name of your City or Locality?
[Unknown]: Pune
What is the name of your State or Province?
[Unknown]: Maharashtra
What is the two-letter country code for this unit?
[Unknown]: IN
Is CN=ope-auth.test, OU=tibco, O=tibco, L=Pune, ST=Maharashtra,
C=IN correct?
[no]: Yes

C:\jdk11\bin>keytool -export -alias ope -file ope123.crt -keystore
ope.pkcs12

C:\jdk11\bin>keytool -import -v -trustcacerts -alias ope2 -file
ope123.crt -
keystore cacerts.pkcs12 -keypass changeit

```

When prompted, provide the password as 'changeit'.

2. Copy cacerts.pkcs12 and ope.pkcs12 files from the JAVA\_HOME\bin directory to the base/1.0 directory and modify base dockerfile accordingly.  
Example: copy ope.pkcs12 and cacerts to location /home/tibuser/tibco/ope/6.0
3. Copy the cacerts.pkcs12 file inside the \$OPE\_HOME/roles/<Service\_name>/standalone/config/ directory of each service.
4. Run the copyLib.sh script from the roles directory.

5. Run the `copy-required-files.sh` script.
6. Modify the TIBCO OPE services (except authorization service )dockerfile for entrypoint as follows:

```
ENTRYPOINT ["sh","-c",
"/home/tibuser/tibco/oep/6.0/configurator/standalone/bin/
start.sh
-
Djavax.net.ssl.trustStore=/home/tibuser/tibco/oep/6.0/cacerts.pkcs1
2 -Djavax.net.ssl.trustStorePassword=changeit --run=FG"]
```

7. Create Docker images for all TIBCO OPE services.
8. Now, update the `ope_services/values.yaml` file from the `$OPE_HOME/helm` directory as follows:
  - a. Add the following properties:

```
server_ssl_key_alias: ope
server_ssl_key_store_password: tibco123
server_ssl_key_store: /home/tibuser/tibco/oep/6.0/oep.pkcs12
configuratorTrustStoreAbsolutePath:
/home/tibuser/tibco/oep/6.0/cacerts.pkcs12
configuratorTrustStorePassword: changeit
configuratorTrustStoreType: pkcs12
configuratorServiceUrl: https://configurator-
svc.default.svc.cluster.local:9090
authorizationServiceTokenEndPoint: https://authorization-
svc.default.svc.cluster.local:9091
ope_url: https://ope-svc.default.svc.cluster.local:8090
dataStoreUrl: https://redatastore-
svc.default.svc.cluster.local:8094
shoppingcart_url: https://shoppingcart-
svc.default.svc.cluster.local:8091
authorizationServiceTokenEndPoint: https://authorization-
svc.default.svc.cluster.local:9091
configuratorServiceUrl: https://configurator-
svc.default.svc.cluster.local:9090
catalogServiceBaseUrl: https://catalog-
svc.default.svc.cluster.local:9092
catalogServiceUrl: https://catalog-
svc.default.svc.cluster.local:9092
```

- b. Update the scheme for each application to HTTPS.

Example: In the configurator application-

```
readinessProbe:
 failureThreshold: 3
 httpGet:
 path: /management/health/readiness
 port: 9090
 scheme: HTTPS
 periodSeconds: 300
 successThreshold: 1
 timeoutSeconds: 3
livenessProbe:
 failureThreshold: 3
 httpGet:
 path: /management/health/liveness
 port: 9090
 scheme: HTTPS
 periodSeconds: 300
 successThreshold: 1
 timeoutSeconds: 3
```

9. Specify the backend protocol as HTTPS for the Ingress in the ope\_services/templates/ope\_ingress.yaml file.

Example of using the Nginx Ingress:

```
annotations:
 nginx.ingress.kubernetes.io/backend-protocol: https
```

10. Create the required users from the authorization service and upload required metadata, app\_properties, and config files as per components from the configurator service.

The values.yaml file contains the required properties for starting authorization service, configurator service, and configurator UI services.

## Configuring On-premise

### Procedure

1. Go to the JAVA\_11\_HOME\bin directory and run the following commands:

```
keytool -genkey -alias ope -keyalg RSA -keysize 2048 -sigalg
SHA256withRSA -validity 365 -keystore ope.pkcs12 -storepass
tibco123 -ext san=ip:10.x.x.x,dns:localhost,ip:127.0.0.1
keytool -export -alias ope -file ope123.crt -keystore ope.pkcs12
keytool -import -v -trustcacerts -alias ope2 -file ope123.crt -
keystore cacerts.pkcs12 -keypass changeit
```

When prompted, provide the password as 'changeit'.

2. Copy cacerts.pkcs12 and ope.pkcs12 files from <JAVA\_HOME>/bin directory at a location (such as /home/OPE\_600/tibco/ope/6.0/ssl), where your TIBCO OPE installation is present on VM.
3. Copy cacerts.pkcs12 inside the \$OPE\_HOME/roles/<Service\_name>/standalone/config/ directory of each service.
4. For authorization service, modify the application.properties file present inside the config directory for the following properties:

```
server.ssl.key-alias=ope
server.ssl.key-store-password=tibco123
server.ssl.key-store=/home/OPE_600/tibco/ope/6.0/ssl/ope.pkcs12
```

```
#Allowed Cross Origin Resources
```

```
allowedCorsOrigins=https://10.x.x.x:9091,https://10.x.x.x:9090,http
s://10.x.x.x:9092,
https://10.x.x.x:9094,https://10.x.x.x:9099,https://10.x.x.x:9095,h
ttps://10.x.x.x:9102,
https://10.x.x.x:9100,https://10.x.x.x:9093,https://10.x.x.x:9089,h
ttps://10.x.x.x:9104,
https://10.x.x.x:8090,https://10.x.x.x:8093,https://10.x.x.x:8090
```

5. Run the ./start.sh script to start the authorization service.
6. Create the required users. For more information, see 'Create User' topic in the *TIBCO® Offer and Price Engine User Guide*.
7. Add the following properties for the configurator service:

```
server.ssl.key-alias=ope
server.ssl.key-store-password=tibco123
server.ssl.key-store=/home/OPE_600/tibco/ope/6.0/ssl/ope.pkcs12
```

8. Start configurator service by running the following command:

```
./start.sh -Djavax.net.ssl.trustStore=/home/OPE_
600/tibco/ope/6.0/ssl/cacerts.pkcs12 -
Djavax.net.ssl.trustStorePassword=changeit
```

9. Modify the `app_properties` file from the `$OPE_HOME/seed-data/app-properties` directory for the following properties (also required minimum configurations by users):

- a. For Catalog service, there are no changes.
- b. For Common Configuration, under 'Authorization Server Configuration Properties Used for Swagger UI':

```
authorizationServiceTokenEndPoint = https://10.x.x.x:9091
```

- c. For Recommendation Engine Api service, under 'Data Store configuration':

```
dataStoreUrl: https://localhost:8094/data-store
```

- d. For Recommendation Engine Core under 'Recommendation Engine Core Initial':

```
catalogServiceUrl: https://localhost:9092
dataStoreUrl: https://localhost:8094
```

- e. For Recommendation Engine Data Mapper under 'Recommendation Engine':

```
dataStoreUrl: https://localhost:8094
```

- f. For Shopping Cart under 'Shopping Cart Initial Configuration':

```
ope.url
https://localhost:8090
```

10. Upload the metadata, app-properties, and config files from the Swagger UI. See 'Upload Configuration File for Application ID' section in the *TIBCO® Offer and Price Engine Web Services Guide*.
11. For configurator-ui and rest of the TIBCO OPE services, update the application.properties file as follows:

```
server.ssl.key-alias=opec
server.ssl.key-store-password=tibco123
server.ssl.key-store= /home/OPE_600/tibco/opec/6.0/ssl/opec.pkcs12
configuratorTrustStoreAbsolutePath= /home/OPE_
600/tibco/opec/6.0/ssl/cacerts.pkcs12
configuratorTrustStorePassword=changeit
configuratorTrustStoreType=pkcs12
```

12. Start all services by following command from the <service-name>/bin directory:

```
./start.sh -Djavax.net.ssl.trustStore=/home/OPE_
600/tibco/opec/6.0/ssl/cacerts.pkcs12 -
Djavax.net.ssl.trustStorePassword=changeit
```

## Encrypt Password Utility

1. In the \$OPE\_HOME/samples/EncryptPWDUtility/standalone/bin directory, run startEncryption.sh (for Linux) or startEncryption.ps1 (for Windows).

### Result

The password encryption and decryption can be done from EncryptPWDUtility service swagger APIs .

# TIBCO Documentation and Support Services

---

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

## How to Access TIBCO Documentation

Documentation for TIBCO products is available on the [TIBCO Product Documentation](#) website, mainly in HTML and PDF formats.

The [TIBCO Product Documentation](#) website is updated frequently and is more current than any other documentation included with the product.

## Product-Specific Documentation

The following documentation for this product is available on the [TIBCO® Offer and Price Engine](#) documentation page:

- *TIBCO® Offer and Price Engine Release Notes*
- *TIBCO® Offer and Price Engine Installation and Configuration Guide*
- *TIBCO® Offer and Price Engine Concepts Guide*
- *TIBCO® Offer and Price Engine User Guide*
- *TIBCO® Offer and Price Engine Web services Guide*
- *TIBCO® Offer and Price Engine Security Guidelines*

## How to Contact TIBCO Support

Get an overview of [TIBCO Support](#). You can contact TIBCO Support in the following ways:

- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the [TIBCO Support](#) website.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to [TIBCO Support](#) website. If you do not have a user name, you can request one by clicking **Register** on

the website.

## How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the [TIBCO Ideas Portal](#). For a free registration, go to [TIBCO Community](#).

# Legal and Third-Party Notices

---

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc.

TIBCO, the TIBCO logo, and the TIBCO O logo are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: <https://scripts.sil.org/OFL>

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the readme file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SOFTWARE GROUP, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of Cloud Software Group, Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (<https://www.tibco.com/patents>) for details.

Copyright © 2010-2023. Cloud Software Group, Inc. All Rights Reserved.