TIBCO FTL®

Security

Version 6.9.1
November 2022

Contents

About this Product

TIBCO® is proud to announce the latest release of TIBCO FTL® software.

The release is the latest in a long history of TIBCO products that leverage the power of Information Bus® technology to enable truly event-driven IT environments. TIBCO FTL software is part of TIBCO Messaging®. To find out more about TIBCO Messaging software and other TIBCO products, please visit us at www.tibco.com.

Product Editions

TIBCO Messaging is available in a community edition and an enterprise edition.

TIBCO Messaging - Community Edition is ideal for getting started with TIBCO Messaging, for implementing application projects (including proof of concept efforts), for testing, and for deploying applications in a production environment. Although the community license limits the number of production processes, you can easily upgrade to the enterprise edition as your use of TIBCO Messaging expands.

The community edition is available free of charge. It is a full installation of the TIBCO Messaging software, with the following limitations and exclusions:

- Users may run up to 100 application instances or 1000 web/mobile instances in a production environment.
- Users do not have access to TIBCO Support, but you can use TIBCO Community as a resource (community.tibco.com).

TIBCO FTL in the Community Edition has the following additional limitations and exclusions:

- Excludes transport bridges
- Excludes the RDMA transport protocol
- Excludes disaster recovery features
- Excludes customizable dashboards and monitoring gateway

TIBCO Messaging - Enterprise Edition is ideal for all application development projects, and for deploying and managing applications in an enterprise production environment. It

includes all features presented in this documentation set, as well as access to TIBCO Support.

Security Introduction

At TIBCO, security is our highest priority. You can be sure that TIBCO FTL messaging is secure based on our own high standards and the best practices of our infrastructure providers.

FTL offers logging, transport security, authentication and authorization, and compatibility with other products.

This document describes procedures to ensure security within FTL components and the communication between components. It also provides security-related guidance for other aspects of internal and external communication including product connectivity and configuration of security options.

This document is useful to security officers, deployers, administrators, and purchasers. Developers may be interested in the Coordination Forms section.

This section is an overview of FTL security features. The FTL documentation also includes security recommendations for fine tuning the security of FTL as well as security tips for Developers.

TIBCO's Security Priority

At TIBCO, security is our highest priority. TIBCO maintains a company-wide information security management system and control program that includes security policies, standards, and procedures based on ISO/IEC 27001:2013.

TIBCO's incorporates the STRIDE model to help analyze and find threats to ensure system and service integrity and reliability in processes, data stores, data flows, and trust boundaries. Our approaches ensure applications and systems fulfill the CIA triad (confidentiality, integrity and availability).

Figure 1: STRIDE and TIBCO Assurance

Threat	TIBCO Assurance
Spoofing identity	Authenticity - authentication
Tampering with data	Integrity and auditing
Repudiation threats	Non-repudiability
Information disclosure	Confidentiality
Denial of service	Availability
Elevation of privilege	Authorization

TIBCO adheres to privacy and security requirements related to the protection and processing of individual personal data (collectively, the "Protected Data"). For details, see our Customer Privacy and Security Statement.

TIBCO has adopted policies and practices in alignment with industry best-practices, including quickly addressing and disclosing vulnerabilities. TIBCO has an incident response policy and plan. The policy ensures that security incidents are identified, contained, investigated, and remedied. For details or to report a potential security issue, see security@TIBCO.

TIBCO has policies that guide our Product Life Cycle (PLC). The policies include peer reviews, static code analysis, and both manual and automated QA processes. In addition, we routinely run performance testing on any new or updated software to ensure the highest quality. There is a clear division between devops and development. All changes are logged in our source code repository. We leverage standard deployment tools. Our iterative methodology ensures a functional view of the processes, milestones, activities, and artifacts, or records

TIBCO's Business Continuity Plan (BCP)

TIBCO has a Business Continuity Plan (BCP) to ensure the effects of an emergency event are minimized. If a disaster or emergency situation occurs, the Support Emergency Team (SEMT) coordinates the recovery effort and uses the Employee Communication Chains to notify staff that the BCP is activated.

TIBCO has a Information Security Management System (ISMS) to preserve the confidentiality, integrity, and availability of information. Information security is considered in the design of processes, information systems, and controls.

TIBCO’s Quality Management System (QMS) is based upon ISO 9001, which is an internationally recognized standard that sets out the criteria for a quality management system incorporating the Plan-Do-Check-Act (PDCA) cycle. TIBCO's QMS is a formalized system of business processes, procedures, and responsibilities focused on

- Customer Excellence - Meeting customer requirements and enhancing customer satisfaction by providing high-quality products and services
- Quality - Meeting TIBCO requirements for quality policies and objectives with measurable goals

TIBCO's Quality Management System is documented and structured in levels.

Security Features

TIBCO FTL software includes the following security features:

- Secure transports for communications among application peer processes by fully encrypted communications using Transport Layer Security (TLS) to secure Transmission Control Protocol (TCP) and Dynamic TCP transports
- HTTPS for secure connections to the FTL server
- Customizable authentication mechanisms, including JAAS based authentication and external LDAP authentication
- Fine-grained control over authentication and authorization permissions, including configuration of authorization groups in the flat file of the authentication service
- Full configuration control
- Monitoring and logging
- Permissions set at the cluster and store level and, for eFTL, channel level.

Security Boundaries

FTL logs key activities. Your organization is responsible for protecting and reviewing those files.

FTL ensures that administrative interfaces properly authenticate and authorize users. Your organization is responsible for configuring permissions for those users.

Your organization is responsible for making sure the running engines have sufficient resources to handle the loads created by the applications.

The OWASP Top 10 document and the CWE Top 25 Most Dangerous Software Weaknesses can help your organization ensure the most critical security risks are covered.

Security Vulnerabilities

Security features that protect FTL connections and communications depend on the implementation of OpenSSL for implementation Transport Layer Security (TLS) protocols. If the security of OpenSSL were compromised, FTL and applications that use FTL could be vulnerable as well.

For information about OpenSSL configuration, components, and downloads, see OpenSSL.org.

In addition to the key security technologies for TIBCO FTL software described, security depends on your organization correctly configuring and using FTL's components and capabilities.

Persistence Service Transports

Each persistence service uses special-purpose transports of two kinds: client transports and cluster transports.

- *Client transports* communicate between a persistence cluster and its client processes.
- *Cluster transports* communicate within a persistence cluster.

You can configure the details of these transports as part of the persistence service definition (rather than as separate transport definitions).

You can define a second client transport, called the alternate client transport, and assign clients on specific hosts to use that alternate transport. All other clients use the regular client transport.

For details, in TIBCO FTL Administration, see "Persistence Service Transports". In this document, see Securing Persistence Services.

Security of Monitoring Data

TIBCO FTL components and clients can secure monitoring data and log data using TLS.

If your enterprise uses secure FTL servers, then data is secure as it travels from client to FTL server, and then to subscribers.

However, TIBCO makes no assertions about the security of third-party components, such as InfluxDB or Grafana.

For details, see "Security of Monitoring Data" in TIBCO Monitoring. In this document, see Securing Monitoring and Log Data.

Log Service Security

The log service is an FTL service component.

Security of log data requires an unbroken chain of secure connections:

- FTL clients to the FTL server
- FTL server to monitoring gateway
- Gateway to InfluxDB
- InfluxDB to log service
- Log service to its HTTPS clients

For details, see "Log Service" in in TIBCO Monitoring. In this document, see Securing Log Services.

Password Security

Keystore passwords encrypt key files, such as the private key file that FTL servers use to identify themselves to clients and to other servers.

Passwords can be masked. You can mask passwords using tibftladmin. Masked passwords have `$mask$` at the beginning of the string. Masked passwords are unmasked before being sent to the realm service.

FTL allows you to supply a password in several ways based on the level of security desired.

For details, see "Password Security", "Security: Clients" section in TIBCO Administration.

Secure Communications

The FTL server enforces TLS security in its communications with clients. If the FTL server does not use secure communications, a warning appears. For details, see "Realm Properties Details Panel" in TIBCO Administration.

Secure FTL Servers

Secure FTL servers use certificates and TLS to guarantee server identity to clients and other servers, and to protect communications among them. FTL generates a keystore file and trust file. For details, see "Secure FTL Servers" in TIBCO Administration.

The FTL servers can run in one of three security modes:

- **Insecure mode**: This mode uses neither login authentication nor data encryption.
- **Auth-only mode**: This mode uses login authentication to authorize client-server, server-server, and administrative connections.
- **Secure mode**: This mode uses both login authentication and data encryption.

FTL Server Configuration Parameters

You can set FTL server configuration parameters, including TLS security, EMS server security, and client security. For details, see "FTL Server Configuration Parameters" in TIBCO Administration.

Permissions

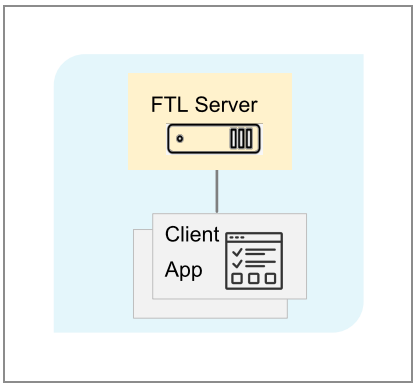
Administrators can ensure secure messaging by setting permissions at the cluster and store level. More fine grained entitlements can be set up on destinations, to preventing users from sending or receiving sensitive data. Both a secure FTL server and authentication must be turned on to use this feature.

Product Connectivity

TIBCO FTL includes several interconnecting components, and also connects with other TIBCO and third-party products. You can secure all connections within the TIBCO product family.

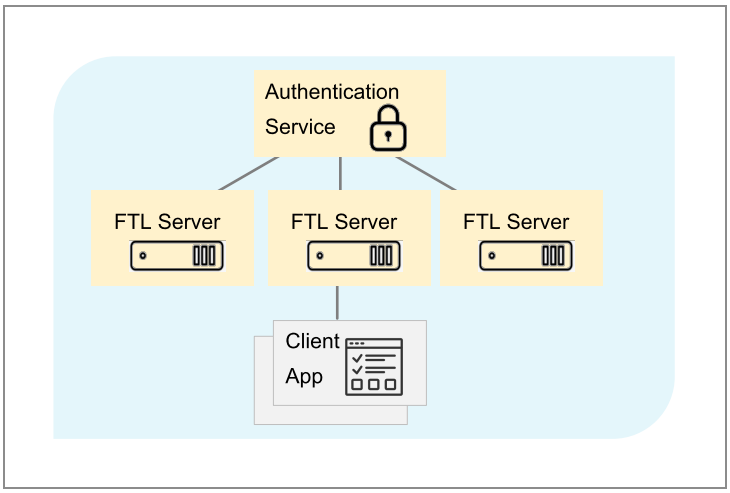
The following diagram, Development Minimum Deployment, illustrates that a development environment only requires an FTL server and a client app.

Figure 2: Development Minimum Deployment



The following diagram, Production Minimum Viable Secure Deployment, illustrates that a production environment must minimally have three FTL servers, an authentication service communicating with each FTL server, and a client application. You must secure these components.

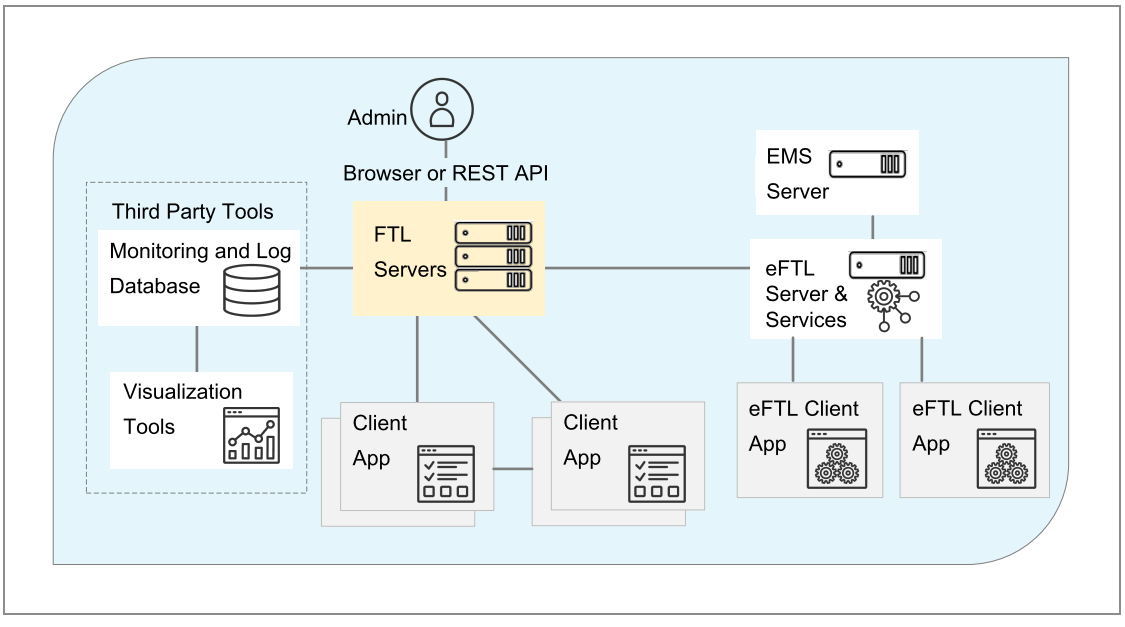
Figure 3: Production Minimum Viable Secure Deployment



The following diagram, Components Connecting in the FTL Realm, depicts other components that may connect to your FTL realm.

> **Caution:** FTL components attempt to interact with third-party components in a secure fashion. TIBCO does not warrant the security of third-party products.

Figure 4: Components Connecting in the FTL Realm



Developing Secure Applications

To implement security, application developers focus on the realm connect call and its arguments. Complete this task, or use its steps as a checklist.

Before you begin

The application developer and administrators coordinate to exchange security-related information and artifacts. See Coordination.

Procedure

1. Coordinate for secure transports.

   Coordinate with administrators to specify secure transports. Record this administrative requirement on the Endpoint Coordination Form.

2. Secure connections to FTL servers using HTTPS.

   In the realm connect call, specify HTTPS as the protocol in the `serverURL` argument.

   For example:

   ```
   https://FTLsvr1:8585|https://FTLsvr2:8585|https://FTLsvr3:8585
   ```

3. Authenticate clients to the FTL server.

   In the realm connect call, supply client credentials using the `USERNAME` and `USERPASSWORD` properties.

   The administrator must ensure that the user is in the authorization group `ftl`.

4. Arrange trust in the FTL servers.

   The application must trust the FTL servers.

   Request the FTL server trust file from the administrator.

   In the realm connect call, supply either the location of the trust file, or its contents as a string in PEM encoding. The following properties organize that information in the connect call:

   - `TRUST_TYPE`

- `TRUST_FILE`
- `TRUST_PEM_STRING`

For details, see the developer API documentation in Web Help or in the FTL source directory, public.

5. Verify authorization for requests.

   If the application responds to requests, verify that the requestor has authorization for the request.

   If a request is forwarded from an eFTL client, the `_user` field of each request message contains the requestor's user name. For details, see "User Field" in *TIBCO eFTL Concepts*.

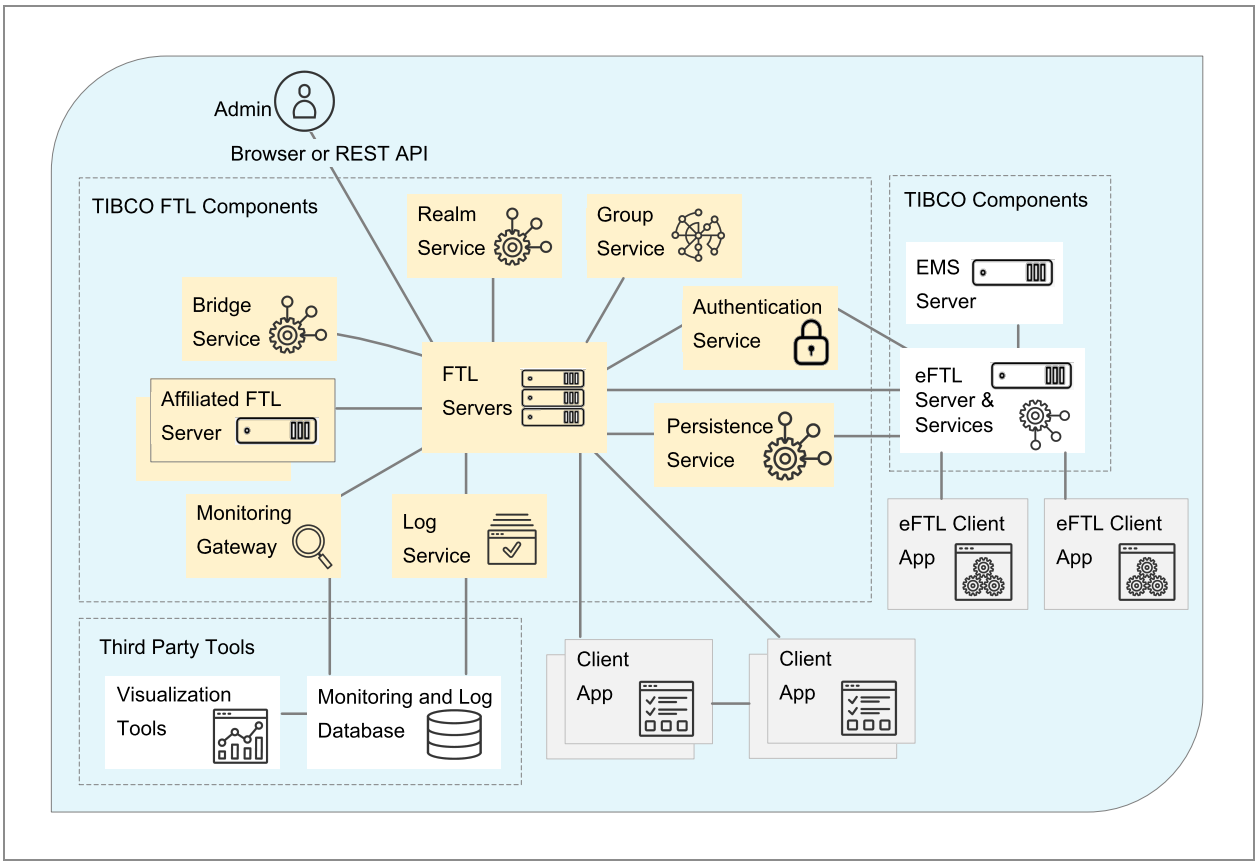
6. Set up permissions at the cluster and store level.

Ensuring FTL System Security: Tasks for Administrators

TIBCO FTL software includes components and processes that communicate within an FTL realm as shown in the following diagram, FTL Server Connections. (For simplicity, the diagram omits redundant connections to duplicate processes, such as the connection from affiliated FTL servers to the authentication service.) This section and the sections that follow address the security issues that arise for each type of connection shown in the diagram, and the actions you must take to ensure security. All processes in your deployment must be secured.

Connections among FTL Processes

Figure 5: FTL Server Connections



To ensure security within and among those components, administrators complete the following tasks.

Procedure

1. **Applications:**

    a. Coordinate with application developers to secure application programs.

    FTL application programs are clients of the FTL server. They must use HTTPS to communicate with the FTL server.

    Your role includes coordinating with application developers to ensure that application clients trust the secure FTL server, and that they supply appropriate credentials when they connect to it. For details, see Coordination.

    b. Secure all application transports.

    Application programs must use secure transports to communicate with one another. Your role includes configuring the application and transport

definitions in the realm definition so that all relevant transports use only secure transport protocols.

Use only these transport protocols:

- Secure Dynamic TCP
- Secure TCP

2. **Authentication and Authorization:**
Configure authentication and authorization.

   Your role includes configuring your enterprise authentication and authorization system (such as an LDAP service) with appropriate information to support TIBCO FTL components and application users.

   For details, see Configuring Authentication and Authorization.

3. **FTL Servers:**
Secure all FTL servers.

   A secure FTL server enforces HTTPS communication whenever it communicates with clients, affiliated FTL servers, and browsers.

   Your role is to supply FTL server command line parameters to secure those client connections.

   For details, see Securing FTL Servers.

4. **TIBCO FTL Component Services:**

   a. Secure all transport bridges.

      Verify that the transports interconnected by the bridges use only secure transport protocols.

      For details, see Securing Transport Bridges.

   b. Secure all persistence services.

      Configure the persistence clusters so that all relevant transports use only secure transport protocols.

      For details, see Securing Persistence Services.

   c. Secure all eFTL services.

      TIBCO eFTL services must use secure transports to communicate with one another, and with eFTL applications.

      Your role includes these subtasks:

- Reconfigure the automatically-generated eFTL transport definitions so that all relevant transports use only secure transport protocols.
- Configure channels with appropriate authorization groups.
- Coordinate with application developers to ensure that eFTL clients connect to the eFTL services using the secure web sockets protocol (WSS).

For details, see Securing eFTL Services.

d. Secure all FTL monitoring services.

The FTL monitoring gateway (`tibmongateway`) is a client of the FTL server. It must use HTTPS to communicate with the FTL server.

Your role includes this subtask:

- Supply appropriate command line parameters to `tibmongateway` to secure its connection to the FTL server.

For details, see Securing Monitoring and Log Data, including:

- Securing Monitoring Gateway Services
- Securing InfluxDB
- Securing Grafana
- Securing Log Services

Coordination

To secure a system that communicates using FTL software, administrators and application developers must coordinate to share security requirements and artifacts.

Coordination Forms

Go to the TIBCO FTL documentation set to download coordination forms to guide the conversation between administrators and application developers as well as record important information, such as security requirements and settings. The coordination forms include:

- **Application Coordination Form:** Developers and administrators use the form to identify the application, coordinate general application information, and the detail the expected deployment.
- **Durable Coordination Form:** Developers and administrators use the form to coordinate the details of durable subscribers.
- **Endpoint Coordination Form:** Developers and administrators use the form to coordinate the details that enable effective and efficient data transmission among application programs. Administrators use the following details captured in the form to select appropriate host computers and transports.
    - Information about each endpoint ability, including a brief description of the messages that each ability carries
    - The expected volume of data in each ability
    - The priority of the data (relative to other messages).
- **Format Coordination Form:** Developers and administrators use the form to coordinate the details of message formats for an application. It captures specific details about each format including field, field name, and data type.

Trust Files, Credentials, and Authorizing Groups

Administrators and developers also coordinate credentials, trust files, and authorization groups.

FTL Application Development

- **Credentials**
  Administrators configure user credentials for authentication and authorization, and supply them to developers for testing applications and to operations staff for running applications.
- **Trust File**

  Administrators supply a the location or contents of the FTL server trust file to developers and operations staff.

  Developers code applications to specify the location or contents of the trust file in the realm connect call.
- **Authorization Groups**

Developers inform administrators about the publish and subscribe requirements of clients.

Administrators configure channels with publish and subscribe authorization groups.

eFTL Application Development

- **Credentials**

  Administrators configure user credentials for authentication and authorization. Credentials are supplied to:

  - Developers so they can test applications
  - Device users so they can run applications that connect to a secure eFTL service

- **Trust File (FTL Server)**

  An eFTL client connects to the FTL server address.

  Clients must trust a user-specified certificate (for example, using the client host's default trust store if the certificate has been signed by a well-known certificate authority).

  To configure the FTL server to present a user-specified certificate to eFTL clients, use these YAML configuration parameters that are valid in the globals section:

  `custom.cert`

  `custom.cert.private.key`

  `custom.cert.private.key.password`

  This will not affect FTL clients. A client used to access the UI or web API must trust the user-specified certificate.

- **Trust File (eFTL Service)**

  An eFTL client connects directly to an eFTL service (for example, legacy clients, or after migrating servers).

  Clients must trust a user-specified certificate.

  To configure how the eFTL service listens, use the `listen` parameter in the `eftlservice` section of the FTL server YAML. For secure connections, this must be a "wss" address.

  To configure the eFTL service to use a user-specified certificate, use the parameters in the `eftlservice` section of the FTL server YAML file:

  `server.cert`

`private.key`

`private.key.password`

- **eFTL Authorization Groups**

  Developers inform administrators about the publish and subscribe requirements of clients.

  Administrators configure channels with publish and subscribe authorization groups.

Configuring Authentication and Authorization

Complete this task to enforce enterprise authentication and authorization requirements in TIBCO FTL servers and services.

Note: Having secure mode on and authentication not enabled is not a recommended configuration.

Procedure

1. Select an authentication service.

   Choose one of the following:

   - The FTL server's internal flat-file authentication service
   - The sample external JAAS authentication service, in combination with your enterprise's LDAP service
   - Another external authentication service

   Tip: In this context, "internal" indicates that the authentication service is inside the FTL server process. "External" indicates that the authentication service is separate from the FTL server, and the FTL server connects to it.

2. Configure user names, passwords, and authorization groups.

   Configure user credentials either in a flat file, or in your enterprise LDAP, depending on your choice in step 1.

   For the file syntax of the internal authentication service, see "Using the Internal Flat-File Authentication Service" in TIBCO FTL Administration.

- Ensure that users who run FTL servers are in the authorization group `ftl-internal`.
- Ensure that administrators who configure the FTL realm definition are in the group `ftl-admin`.
- Ensure that users who run FTL application programs or FTL services are in the group `ftl`.
- Ensure that device users who run eFTL apps are in the appropriate publish and subscribe authorization groups.
- You may also configure other authorization groups to manage access within your enterprise.

3. Start the external authentication service. Perform one of the following based on whether you chose an external or internal authentication service in step 1.
   - If you chose an external authentication service in step 1, start that service *before* starting the FTL server processes.

     To start the sample external JAAS service, complete the task "Using the External JAAS Authentication Service" in TIBCO FTL Administration.
   - If you chose the internal flat-file authentication service in step 1, no further action is necessary, as that service starts automatically when you start the FTL server.

What to do next

Complete the task Securing FTL Servers.

Securing FTL Servers

Secure FTL servers are central to the security of any enterprise that communicates using TIBCO FTL messaging software. To secure the FTL servers, complete this task. An FTL server can generate all the data it requires for TLS, except for the keystore password, which you must supply.

Before you begin

- Secure the FTL server data directories and files against unwanted access by other

users.

- The enterprise authentication system (for example, and LDAP system) must define user names and associate them with appropriate FTL authorization groups.
- An authentication service (either internal or external) must be running. For background information, see Authentication Service in TIBCO FTL Administration.
- Choose a keystore file password, and determine the appropriate level of security for that password.
- Ensure that the clocks on all servers in a cluster are synchronized.

Procedure

1. Remove any obsolete TLS data files from the FTL servers' data directories.
2. Generate TLS data files.

   To generate full-security files, enter:

   ```
   tibftlserver --init-security file:<pw_file_name> -c <my_config_
   file_path> -n <svr_name>
   ```

   To prepare the server for authentication-only operation, enter:

   ```
   tibftlserver --init-auth-only
   ```

   This command instructs the FTL server to generate new TLS data files, encrypting the new keystore file with the password.

   (If the FTL server detects existing TLS files, it does not generate them anew. However, the FTL server does not decrypt or inspect existing files.)

   The server generates TLS files in the data directory (specified in the configuration file). If the data directory is unavailable, the server writes these files to the current directory. After writing the files, the FTL server exits.
3. Distribute the TLS files.

   The keystore file and trust file must be distributed to all FTL servers which include all core servers and auxiliary servers at all sites (including primary, satellite, and DR sites).

   Every server uses the same private key to identify itself. Every server uses the same trust file to verify the identity of FTL servers.

a. Supply copies of the keystore file and trust file to every FTL server.

   Place these files in the data directory of the servers.

   > **Note:** Specify the data directory in the configuration file for each FTL server.

b. Supply a copy of the trust file to every client including application programs and browsers that access the FTL server GUI.

   For more information, see Trust File in TIBCO FTL Administration.

> **Note:** When a server generates *new* TLS data files, you must redistribute these files.

4. Configure the FTL servers to use TLS security and supply the keystore file password as the property value:

```
globals:
    # ...
    tls.secure: <password_argument>
```

   FTL servers use the password to encrypt and decrypt the keystore file. For information on the form of the password argument, see Password Security in TIBCO FTL Administration.

5. Configure the FTL server properties related to the authentication service.

   FTL servers authenticate and authorize client credentials using the authentication service. Configure the authentication service in the FTL server configuration file.

6. Configure the username and password for communication with affiliated FTL servers. If satellite or DR FTL servers will be used, the primary FTL servers must authenticate themselves to the satellite or DR servers and vice versa. Add an appropriate username and password to the configuration file of all FTL servers (primary or satellite or DR). Ensure that this user has the `ftl-internal` role. See FTL Server Configuration Parameters, "Affiliated FTL Servers".

```
globals:
```

```
    # ...
    user: <username>
    password: <password_argument>
```

7. Start the FTL server processes.

   Start servers using a standard command line (that is, without the `--init-security` option). For example:

```
tibftlserver -c <config_file> -n <server_name>
```

> **Note:** See the `ftlstart` script in the `samples` directory. The `--secure` option illustrates a basic way to start a secure FTL server.

Securing Transport Bridges

To secure a transport bridge, complete this task.

Before you begin

All FTL servers must be secure.

Procedure

1. Verify that the transports interconnect by the bridge. Configure only secure network transport protocols.

   Use only these transport protocols:

   - Secure Dynamic TCP
   - Secure TCP

Securing Persistence Services

To secure a persistence service, complete this task.

Before you begin

All FTL servers must be secure.

Procedure

1. Verify that the persistence cluster definition specifies secure transport protocols.

   The client protocol, disaster recovery (DR) protocol, and inter-cluster protocol must be secure. For maximum performance, the cluster set protocol can be a non-secure protocol -- but only if all persistence services of the cluster run within a protected network. Otherwise use a secure protocol for cluster set communications.

   Use only these transport protocols:

   - Secure Dynamic TCP
   - Secure TCP
   - Secure Auto

   For further details, see "Clusters Grid" in TIBCO FTL Administration.

Securing eFTL Services

To secure an eFTL service, complete this task.

Before you begin

All FTL servers must be secure. See Securing FTL Servers.

If any channels use EMS servers or FTL persistence services, those services must also be secure.

Procedure

1. Verify secure transport protocols.

   The cluster-facing transport and all the channel application-facing transports must be secure. Check their protocols in the transports grid.

   Use only these transport protocols:

   - Secure Dynamic TCP

- Secure TCP
- Secure Auto

Example Configuration File

```
globals:
    core.servers:
        ftl1: host1:8585
        ftl2: host2:8585
        ftl3: host3:8585
    custom.cert: <custom_cert.pem>
    custom.cert.private.key: <custom_key.pem>
    custom.cert.private.key.password: <custom_pw>

services:
    eftl:
        name: my_eftl_cluster
        publish.user: true
        eftl.auth.url: <auth_svr_host>:<port>
        eftl.auth.user: <user_name>
        eftl.auth.password: <pw>
        eftl.auth.trust: auth-trust.pem
        ssl.params: eftl-ems-ssl.txt

servers:
    ftl1:
        - realm: {}
        - eftl: {}

  # ...
```

2. Include authenticated user names.

   Specify the parameter `publish.user` in the eFTL service section of the FTL server configuration file.

   With this option, the eFTL service appends a field to messages published by eFTL client apps when it forwards them to FTL and EMS subscribers. That field contains the authenticated user name of the eFTL publisher. FTL and EMS application code can use this user name to authorize requests.

3. If used, specify the authentication service.

   Optionally, eFTL services can use an external authentication service (JAAS, LDAP, etc.), instead of the built-in authentication service in the FTL server.

To use an external authentication service, supply the parameters `eftl.auth.url`, `eftl.auth.user`, `eftl.auth.password`, and `eftl.auth.trust` in the eftl section of the FTL server configuration file.

For further details, see the following topics in TIBCO eFTL Administration:

- Client Authentication and Authorization
- Channel Details Panel

4. Optional. For a user-specified certificate instead of the FTL server default certificate, supply the parameters `custom.cert`, `customer.cert.private.key`, and `custom.cert.private.key.password` in the globals section of the FTL server configuration file. The FTL server uses this certificate to identify itself to clients. See the "Example Configuration File" earlier and FTL Server Configuration Parameters in TIBCO FTL Administration.
5. Optional. Specify client authorization groups.

   eFTL channels can regulate client access to publish and subscribe operations. To enable this feature, complete the following steps:

   a. In the eFTL clusters grid, enable the authorization column for each relevant cluster.

   b. In the channel details panel, configure a publish group and a subscribe group for each relevant channel.

   c. Ensure that each user name is in the appropriate authorization groups.
6. Optional. Secure FTL persistence services.

   If any channels use FTL persistence stores, then complete the task Securing Persistence Services.
7. Optional. Secure connections to EMS servers.

   If any channels use EMS messaging, specify the `ssl.params` parameter in the eFTL service section of the FTL server configuration file. Supply the location of a configuration file as its value.

   For details about the content of that file, see SSL Parameters for EMS Connections in TIBCO eFTL Administration.

Securing Monitoring and Log Data

To ensure end-to-end security for monitoring and log data, complete all the subtasks of this task.

Before you begin

Ensure that the FTL server is secure, and its clients can connect using HTTPS.

Procedure

1. Secure the monitoring gateway.

   See Securing Monitoring Gateway Services.

2. Secure the InfluxDB server.

   See Securing InfluxDB.

3. Secure the Grafana server.

   See Securing Grafana.

4. Secure the log service.

   See Securing Log Services.

Securing Monitoring Gateway Services

To secure an FTL monitoring gateway service (`tibmongateway` process), complete this task.

Before you begin

All FTL servers must be secure.

The enterprise authentication system must define user names and associate them with appropriate FTL authorization groups.

Secure realm servers automatically use secure transports for the stream of monitoring data.

Procedure

Example Command Line

```
tibmongateway
               --ftlserver
https://ftl1:8585|https://ftl2:8585|https://ftl3:8585
               --password-file mon-gw-creds.txt
               --ftlserver-trust-file ftl-trust.pem
               --influx-server https://influx-host:8086
               --influx-trust-file inflx.pem
```

1. Connect only to secure FTL servers using HTTPS.

   When you supply the `--ftlserver` parameter on the gateway command line, specify a URL with HTTPS protocol.

2. Arrange authentication credentials to the FTL server.

   Supply the location of the gateway's credentials as the value of the `--password-file` parameter on the gateway command line. Ensure that this file is protected from unauthorized access.

   The user name in the file must be in the authorization group `ftl`.

   For further details, see "Monitoring Gateway Command Line Reference (`tibmongateway`)" in *TIBCO FTL Monitoring*.

   For file syntax, see "Password File" in TIBCO FTL Administration.

3. Arrange trust in the FTL servers.

   Arrange access to a copy of the FTL server trust file.

   Supply the file location as the value of the `--ftlserver-trust-file` parameter on the gateway command line.

   For further details, see "Trust File" in TIBCO FTL Administration.

4. Connect to the InfluxDB server.

   Supply a URL with HTTPS as the protocol as the value of the `--influx-server` parameter on the gateway command line.

5. Arrange trust in the InfluxDB server.

   Arrange access to a copy of the InfluxDB server public certificate file.

   Supply the file location as the value of the `--influx-trust-file` parameter on the gateway command line.

Securing InfluxDB

To secure the InfluxDB server and its client connections, complete this task.

Procedure

1. Obtain and install a certificate and private key for the InfluxDB server.

   Ensure that the private key file is protected from unauthorized access.

   InfluxDB uses this certificate to identify itself to its clients, including Grafana, and the FTL server.

2. Configure InfluxDB to for secure connections from its clients.

   a. In a text editor, open the configuration file `influxdb.conf`.

      The script that starts the FTL monitoring components uses the file in the location `<FTL_HOME>/monitoring/influxdb/etc/influxdb/influxdb.conf`.

      If you start the InfluxDB server independently, modify the corresponding configuration file in the appropriate location.

   b. Locate the `http` section.

   c. Set `https-enabled=true`.

   d. Set `https-certificate` to the location of the InfluxDB server's certificate.

   e. Set `https-private-key` to the location of the InfluxDB server's key file.

   For details see InfluxDB documentation.

3. Restart InfluxDB server.

Securing Grafana

To secure Grafana, complete this task.

Before you begin

InfluxDB must be configured for HTTPS connections.

Procedure

1. Obtain and install a certificate and private key for the Grafana server.

Ensure that the private key file is protected from unauthorized access.

Grafana uses this certificate to identify itself to its web clients.

2. Configure the Grafana server for HTTPS connections from its clients.
   a. In a text editor, open the configuration file `<FTL_HOME>/monitoring/grafana/conf/default.ini`.
   b. Locate the `server` section.
   c. Set `protocol=https`.
   d. Set `cert_file` to the location of the Grafana server's certificate.
   e. Set `cert_key` to the location of the Grafana server's key file.

   For more detail, see Grafana documentation.
3. Restart the Grafana server.
4. In a browser, log in to the Grafana server.

   Supply a URL with HTTPS as the protocol.
5. In the Grafana server web GUI, modify the data source definition for FTL to use HTTPS.

   Supply the location of the InfluxDB server's certificate, so the Grafana server trusts the InfluxDB server.

   For details, see Grafana documentation.

Securing Log Services

To secure an FTL log service (`tiblogsvc` process), complete this task.

Before you begin

All FTL servers must be secure.

The enterprise authentication system must define user names and associate them with appropriate FTL authorization groups. The monitoring data base (InfluxDB) must be secure.

Procedure

Example Command Line

```
tiblogsvc
          --ftlserver
https://ftl1:8585|https://ftl2:8585|https://ftl3:8585
          --ftlserver-password-file logsvc-creds.txt
          --ftlserver-trust-file ftl-trust.pem
          --influx-server https://influx-host:8086
          --influx-password-file logsvc-influx-creds.txt
          --influx-trust-file influx-trust.pem
          --http-certificate logsvc-cert.pem
          --http-key logsvc-key.pem
          --http-password-file my_pw_file
```

1. Connect only to secure FTL servers using HTTPS.

   When you supply the `--ftlserver` parameter on the log service command line, specify URLs with HTTPS protocol.

2. Arrange authentication credentials to the FTL server.

   Supply the location of the log service's credentials as the value of the `--ftlserver-password-file` parameter on the log service command line. Ensure that this file is protected from unauthorized access.

   The user name in the file must be in the authorization group `ftl`.

   For further details, see Log Service Command Line Reference (tiblogsvc) in TIBCO FTL Monitoring.

   For file syntax, see Password Security in TIBCO FTL Administration.

3. Arrange trust in the FTL servers.

   Arrange access to a copy of the FTL server trust file.

   Supply the file location as the value of the `--ftlserver-trust-file` parameter on the log service command line.

   For further details, see Trust File in TIBCO FTL Administration.

4. Connect only to a secure InfluxDB server using HTTPS.

   When you supply the `--influx-server` parameter on the log service command line, specify a URL with HTTPS protocol.

5. Arrange authentication credentials to the InfluxDB server.

   Supply the location of the log service's credentials as the value of the `--influx-`

`password-file` parameter on the log service command line. Ensure that this file is protected from unauthorized access.

For further details, see Log Service Command Line Reference (tiblogsvc) in TIBCO FTL Monitoring.

For file syntax, see "Password File" in TIBCO FTL Administration.

6. Arrange trust in the InfluxDB servers.

   Arrange access to a copy of the InfluxDB server trust file.

   Supply the file location as the value of the `--influx-trust-file` parameter on the log service command line.

   For further details, see Trust File in TIBCO FTL Administration.

7. Arrange TLS artifacts so the log service can authenticate itself to clients.

   a. Obtain a certificate identity for the log service.

   b. Supply the location of the certificate file as the value of the `--http-certificate` parameter on the log service command line.

   c. Supply the location of the key file as the value of the `--http-key` parameter on the log service command line.

      Ensure that this file is protected from unauthorized access.

   d. Supply the key file password using the `--http-password-file` parameter.

      (The `--http-password` parameter is not sufficiently secure.)

   e. Ensure that HTTPS clients trust the log service's certificate.

      - **Browser Client**: Install the certificate (or the CA certificate) in the requesting browser.
      - **Utility Client**: Supply the certificate (or the CA certificate) to the request utility. For example, `curl --cacert <certificate>`.

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

Documentation for TIBCO FTL® is available on the TIBCO FTL® Product Documentation page.

TIBCO FTL® Documentation Set

The following documents can be found on TIBCO FTL Product Documentation as Web Help or PDFs.

- *Installation*: Read this guide before installing or uninstalling the product.
- *Concepts*: Review this guide for an introduction to FTL software fundamentals.
- *Quick Start*: Use this guide to quickly start FTL and send and receive a message.
- *Getting Started*: Use the guide to set up, start, and run various TIBCO FTL sample programs that demonstrate typical messaging functionality.
- *FTL Tutorials*: Use this guide for a step-by-step approach to build TIBCO FTL applications. You will use options and properties, exception handling, programming models, and the user interface for configuration and monitoring.
- *Administration:* Administrators read this manual to learn how to use the FTL server, its interfaces, and its services, and how to define a realm. Developers can also benefit from understanding FTL software from an administrator's perspective.

- *Development*: Application developers and architects read this manual to understand concepts relevant in any supported programming language.
- *Monitoring*: Administrators read this manual to learn about monitoring and metrics. Developers read this manual to learn how an application can subscribe to the stream of monitoring data.
  Important: Use TIBCO Messaging Monitor for FTL in place of the FTL monitoring component. As of FTL 6.9.0, the FTL Monitoring component (monitoring directory) including Grafana and tibmongateway are deprecated. Also, as of FTL 6.9.0, monitoring metric types are deprecated. See the Release Notes, Deprecated and Removed Features for a complete list.
- *Shifting to FTL:* This manual contrasts TIBCO FTL with TIBCO Enterprise Message Service™, and offers suggestions to smooth your transition to TIBCO FTL. Application developers, architects, and administrators familiar with TIBCO Enterprise Message Service read this manual.
- *Security*: This manual contains security-related tasks for administrators and security tips for application developers.
- *API Documentation*: Application developers use this documentation to learn the details of the FTL API in specific programming languages. This is available as Web Help only.
- *TIBCO FTL Glossary:* The glossary contains brief definitions of key terms used in all other parts of the documentation set.
- *Release Notes:* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

Additional information resources can be found, after file extraction, in the samples directory. These include a Getting Started guide, Tutorials, readme.txt files, and sample applications.

Updated Resources on TIBCO Community

Supplemental resources are now distributed at the TIBCO FTL Community Wiki in the Reference Info tab. You can always find the latest versions of these resources in that location.

Those resources include *TIBCO FTL Getting Started Guide* and *TIBCO FTL Tutorials*. They also include sample FTL server configuration files and sample realm definition files.

TIBCO eFTL™ Documentation Set

TIBCO eFTL software is documented separately. Administrators use the FTL server GUI to configure and monitor the eFTL service. For information about these GUI pages, see the documentation set for TIBCO eFTL software.

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