TIBCO® Managed File Transfer Internet Server

Security Guide

Version 8.5.2 | March 2024

Contents

Introduction

MFT Internet Server is the TIBCO® Managed File Transfer Internet Server product. MFT Internet Server is the file transfer component. MFT Internet Server supports many open protocols, and it also supports the Platform Server protocol. MFT Internet Server has an administrative component that allows you to configure all of the parameters (users, servers, transfers) to allow file transfers to execute. For additional capabilities, you can install MFT Command Center. MFT Internet Server can be installed in the DMZ or in the internal network; when executing in the DMZ, you must disable the administrative capability because it allows you to configure Internet Server transfers.

This document describes guidelines to ensure security within TIBCO Managed File Transfer (MFT) Internet Server. It provides security-related guidance and recommendations for installation, configuration, and execution of file transfers.

Security Features

TIBCO Managed File Transfer Internet Server provides many features that enhance security. Here is a summary of these features. These features are discussed in more detail later in the document.

- FTP Connections
- Platform Server Security
- OFTP2 Security
- PGP Encryption
- Miscellaneous Security Features

Note: FTP Connections, Platform Server Security, OFTP2 Security, and PGP Encryption can be configured in Command Center but are typically only used in Internet Server.

FTP Connections

You can secure FTP connections for incoming and outgoing requests.

MFT FTP Service

Parameters in the following admin page can help to lock down the FTP protocol for incoming FTP requests:

Administration > Transfer Servers > FTP Server > Configure FTP Server

Parameter	Description
PORT/EPRT Allowed in	Defines whether incoming PORT or EPRT connections are allowed. You can disable PORT/EPRT by clicking on "No".

Parameter	Description
Incoming Request	
PORT Checking	Defines whether any checking is performed on the IP Address sent by the client in the PORT request. We suggest setting this parameter to "Subnet" or "IP Address".
PASV Checking	Defines whether any checking is performed on the IP Address of the connection created by the PASV command. We suggest setting this parameter to "Subnet" or "IP Address".

Refer to the MFT Admin help pages for more information on these parameters.

MFT Server Definitions with Server Type of FTP

Parameters in the Add/Update Server page can help to lock down the FTP protocol for outgoing FTP requests.

Parameter	Description
PORT Checking	Defines whether any checking is performed on the IP Address of the connection created by the PORT command. We suggest setting this parameter to "Subnet" or "IP Address".
PASV Checking	Defines whether any checking is performed on the IP Address sent by the client in the PASV request. We suggest setting this parameter to "Subnet" or "IP Address".

Platform Server Security

MFT supports the following modes of operation for incoming and outgoing Platform Server requests. This is for both file transfer requests and administrative requests such as audit collection, server status and node and profile updates.

1. Clear text mode. The password is encrypted using a proprietary encryption algorithm but the data is not encrypted.
2. AES 256 encryption. The password and data are encrypted using AES256. The asymmetric encryption key is generated through an algorithm on both the client and server.
3. SSL (or TLS) mode. MFT establishes an SSL connection with the Partner Server. A symmetric AES 256 encryption key is exchanged through the secure TLS connection. MFT uses this AES256 encryption key to encrypt and decrypt all data. MFT also adds a message digest and sequence number to each record to prevent man in the middle attacks.
4. Tunnel Mode. All data is sent over a negotiated TLS connection. Each transfer creates a new TLS connection.

Tunnel Mode is the most secure option and is strongly suggested when communicating to partners over the internet. Tunnel Mode requires MFT Internet Server V8.2 and MFT Platform Server V8.0 or higher.

OFTP2 Security

OFTP2 allows you to transfer files in TLS and non-TLS mode. When using non-TLS mode, you can encrypt the data. Nonetheless, we suggest only supporting TLS Mode when performing OFTP Transfers.

To support only OFTP2 in TLS mode, complete the following steps.

Procedure

1. Go to **Administration > Transfer Servers > OFTP2 Server > Configure OFTP2 Server**.
2. Enter the TLS Port. All communication over this port is encrypted in a TLS session.

   ⚠ **Caution:** Do not enter the IP Port. This is the clear text port.
3. In the **OFTP2 Options > Outgoing Parameters**, set **Use TLS** to `Yes`.

PGP Encryption

TIBCO MFT Internet Server supports PGP in a streamed mode.

PGP is important in two ways:

1. It provides an additional level of encryption above what is provided in the file transfer protocol.
2. It can validate the identity of the user or server that created the file.

Whenever you are transferring any personal data, financial data or any data that must be secured, we suggest using PGP encrypting the data prior before being transferred over a network connection. This is particularly important when transferring data over an unsecured FTP connection.

MFT has the following PGP capabilities:

- For incoming file upload requests
    - Decrypt the PGP data
    - Verify the signature of the PGP data
- For incoming file download requests
    - Encrypt the PGP data
    - Add a signature to the PGP data
- For outgoing upload requests to a target server
    - Encrypt the PGP data
    - Add a signature to the PGP data
- For outgoing file download requests
    - Decrypt the PGP data
    - Verify the signature of the PGP data

Miscellaneous Security Features

Follow these general recommendations to secure TIBCO MFT Command Center.

Java System Security

Use the newest Java JDK that is supported by the product. We suggest using Java 11 since this is a long-term support version.

Do not use GNU Java that is shipped with some Linux instances. Use Oracle Java, OpenJDK, Amazon Corretto or IBM Java that is appropriate for your MFT instance.

Setting Cookies to HTTPOnly

By default, HTTPOnly is not set for MFT server generated cookies. Cookies created by the MFT Application will be set to HTTPOnly when the cookie is not used by client JavaScript code. Cookies that do not specify HTTPOnly contain no security or private information.

Set the `usehttponly` parameter in the cfcc.xml file which is located in the

`MFTIS_Install/server/conf/catalina/localhost` directory to `true`.

Configuring the Session Timeout

The session timeout is set to 30 minutes by default. This is good for most installations. If you need to lower this, you must make the following two changes: :

- The `session-timeout` parameter in the `web.xml` file located in the `MFTIS_install/server/conf` directory
- The `SessionTimeOut` parameter in the `web.xml` file located in the `MFTIS_install/webapps/cfcc/WEB-INF` directory

Certificate/Key Authentication

MFT supports certificate authentication for the following protocols:

- Platform Server SSL and Platform Server Tunnel
- SFTP
- FTPS
- HTTPS
- OFTP2

Whenever possible, use certificate authentication. Certificate authentication is relatively simple to set up on SFTP, Platform Server, and FTPS. It is much more complicated on

HTTPS, because you need to update the certificate manager and select a certificate for the browser. Because of the difficulty in implementing HTTPS certificate authentication, it is good practice not to use this.

HTTPS can be secured using an SSO (Single SignOn) connection. See the Single SignOn Support section for more detail.

OFTP2 does not perform certificate authentication. However, you can set **Configure OFTP2 Server > Require Client Certificate** to `Yes` to request the client to send a certificate that is validated by the MFT OFTP2 server.

Two factor Authentication

MFT supports multi-factor authentication in the following ways:

1. By requiring users to log in with a password and with a key or a certificate. This is support for multiple incoming protocols, including FTPS, SFTP, Platform Server, and HTTPS.
2. When using HTTPS, MFT supports OIDC and SAML. SAML and OIDC are described in more detail in the topic titled "Single SignOn Support".
3. When Multi-Factor Authentication (MFA) is enabled the following MFA types are supported:
    - Email
    - Google Authenticator

Restrict IP Addresses

Internet Server and Command Center provides two ways to restrict usage based on IP Address:

- User Definition: You can restrict so that the users can only log in from specific IP Address or IP Address subnets.
- Transfer Definition: You can restrict so that the transfer definitions can only be used when the user logs in from specific IP Address or IP Address subnets.

Note: When a load balancer is used, the following restrictions apply:
HTTP/HTTPS: You should use the `web.xml LoadBalancerIPAddressList` parameter with the IP addresses of all load balancers. This extracts the originating IP Address from the HTTP X-Forwarded-For header.
FTP/FTPS and **Platform Server**: The `LoadBalancerIPAddressList` parameter does not work since there is no way for the load balancers to specify the originating IP address. Some load balancers can be configured to use the originating IP address when connecting to Internet Server. When the load balancer uses the originating IP address when connecting to Internet Server, these parameters can be used.
SFTP: When the load balancer supports forwarding the originating IP address, this parameter prompts the Internet server to accept the originating IP address for SSH Clients.

Single SignOn Support

MFT Supports two methods of Single SignOn for HTTPS clients: OIDC (OpenID Connect) and SAML (Secure Access Markup Language). OIDC is a newer SSO protocol and is simpler to configure than SAML. When possible, we suggest using OIDC instead of SAML.

OIDC: OIDC is built on the OAUTH2 protocol and allows https clients to verify the identity of users based on the authentication performed by an authorization server. MFT supports multiple OIDC servers in an MFT cluster. For example, you could create an OIDC server for internal users and a separate OIDC server for external users.

SAML: SAML is an open standard for exchanging authentication and authorization data between an identity provider (SAML server) and a service provider (MFT). It allows browser clients to authenticate to the SAML Identity Provider and the security assertions are sent to MFT. Only one SAML server is supported by an MFT cluster.

There are three `web.xml` parameters that allow you to enforce that the users use OIDC or SAML. See the TIBCO MFT Internet Server User Guide for more information on the following parameters:

- SSOLoginRequired
- SSOExcludedUsers
- SSOAllowRest

Users/Passwords

After the product is installed,

- Change the password for the administrator and for other predefined users.
- Disable any predefined users that you do not use.
- Optional: Configure time of a day and days of the week that users can access the system.
- Optional: Configure an IP address for a user that limits the user to log on to MFT only from that IP address.

Anonymous Access

You must not give anonymous users rights to upload or download sensitive data.

End User Education

- When the browser offers to save MFT password, you should select No.
- After using MFT, you have to log off and close the browser.
- You should not use MFT and browse other websites at the same time.

Security

- For SSH, TIBCO recommends that all partners use SHA-256/384/512 with a key size of 2048 bits or higher.
- For PGP, TIBCO recommends that all partners use SHA-256/384/512 with a key size of 2048 bits or higher.

ReCaptcha Support

MFT can be configured to support ReCaptcha. ReCaptcha is a Captcha service that allows web servers to distinguish between human and automated access to a web site. ReCaptcha can be configured for the following pages:

- Logon
- Forgot User

- Forgot Password
- Self Register

ReCaptcha is configured in the **Configuration > System Configuration > Recaptcha Settings** tab. By default, ReCaptcha is disabled.

SSH Algorithms

MFT allows you to define SSH Algorithm Groups and assign the algorithm groups to individual servers and to the SSH Listener service. Algorithm Groups are defined by the **Management > SSH Algorithm Group** pages.

SSH Algorithm Groups can be assigned in the following ways:

System Configuration > SSH settings: This acts as the default value for the incoming and outgoing MFT SSH requests.

Administration > Transfer Servers > SSH Server > Configure SSH Server: This overrides the system configuration and is used for incoming SSH Connections.

Partners > Servers > Add Server > SSH Properties: This overrides the system configuration and is used for outgoing SSH Connections.

Security Tasks

It is a good practice to perform security-related tasks mentioned in these sections:

- Installation
- Server Configurations

Installation

You can follow the following recommendations to secure TIBCO MFT Internet Server at installation.

Installation User on UNIX

Install as a non-root or an unprivileged user. If you want to use ports below 1025, use the UNIX iptables command to redirect these ports to ports 8443 and 8080. See *Network section* in the *Installation Guide* for more details on redirecting ports.

> **Note:** Some FTP clients fail when connecting to MFT in a non-root environment due to the way that the FTP protocol works. We recommend using the SFTP/SSH protocol in these cases.

Provide only the necessary rights to update the `MFT_Install` directory and any directories where *LOCAL files are saved.

Installation User on Windows

Install as a normal user, for example: Non Administrator. Normal users can use the ports below `1024`.

Provide only the necessary rights to update the MFT_Install directory and any directories where *LOCAL files are saved.

Securing the JDBC connection

If possible, configure the JDBC driver to use SSL/TLS. Contact your database administrator for instructions.

Using Secure Ciphers

During the installation process, you are prompted to select the TLS/SSL ciphers used. There are three options:

1. Most secure ciphers (excludes CBC ciphers)
2. All secure ciphers (includes CBC ciphers)
3. All ciphers

We suggest using the default value of "Most Secure Ciphers (excludes CBC Ciphers)" which ensures that the most secure ciphers are accepted during TLS/SSL negotiation. This applies to all the following TLS/SSL processing:

- HTTPS connections
- FTPS connections
- Platform Server TLS/SSL and Tunnel connections
- OFTP2 TLS connections

The HTTPS ciphers are then set in: `<MFT-Install>/server/conf/server.xml`

TLS Ciphers used by FTPS, Platform Server, and OFTP2 are defined in:

`<MFT-Install>/server/webapps/cfcc/WEB-INF/web.xml`

Note: By default, only TLSv1.2 is enabled.

Perfect Forward Secrecy

Perfect forward secrecy is an encryption feature whereby the keys used to encrypt data are changed on a frequent basis. If a key is compromised, a limited amount of information can be decrypted.

To implement perfect forward secrecy on the HTTPS connection, complete the following steps.

Procedure

1. Edit the server.xml:

   ```
   <MFT-Install>/server/conf/server.xml
   ```

2. Locate the ciphers parameter for the HTTPS connector.
3. Remove all ciphers starting with "TLS_RSA" and "TLS_ECDH_".
4. Restart the MFT server.

   Note: Keep the cipher starting with "TLS_ECDHE".

Admin Service

Do not install the MFT Admin service on TIBCO MFT Internet Servers on computers located in the DMZ. Do not install TIBCO MFT Command Center in the DMZ. Only install the MFT Admin service on computers in the internal network. We suggest using the TIBCO MFT Command Center to perform all admin functions and to disable the admin service on all TIBCO MFT Internet Server instances.

HTTPS Certificate

Purchase an HTTPS SSL certificate from a well-known certificate authority. The default certificate is a self-signed certificate, which prompts the browser users a warning that the certificate is not trusted. When creating a keystore, use a strong password instead of the default password.

Use SFTP/SSH instead of FTP

We suggest using the SFTP protocol instead of using FTP or FTPS. While FTPS is a secure protocol, it is difficult to configure firewalls and load balancers due to the FTP requirement for Control and Data connections.

Additionally, it is difficult getting FTP and FTPS working in the cloud. If you are considering moving to the cloud, FTP/FTPS client and server transfers should be migrated to SFTP/SSH.

server.xml Parameters

There are a variety of `server.xml` parameters that affect the security mentioned in the following sections.

Parameter	Description
`allowHostHeaderMismatch`	This parameter defines whether the MFT server must reject requests that specify a host in the request line but specify a different host in the host header. This can occur when a customer is using the MFT File Transfer CLI (Command Line Interface) or has created an internal application using the file: `NonGUIApplet_0.0.0.1.jar` or `JavaApplet_0.0.0.1.jar`. The problem occurs when an older version of NonGUIApplet_0.0.0.1.jar or JavaApplet_0.0.0.1.jar is used. MFT releases before 8.2.1 do not set the header value correctly and transfers fail if the value is set to false. If the following are all true, then you can set this value to false: • You do not use the MFT FT File Transfer CLI. • You use the MFT FT File Transfer CLI, but are using the FT Command Line distributed with MFT V8.2.1 or above. • You have not created any file transfer applications using the NonGUIApplet_0.0.0.1.jar or JavaApplet_0.0.0.1.jar files. • You have created file transfer applications using file NonGUIApplet_0.0.0.1.jar or JavaApplet_0.0.0.1.jar but you are using versions of these files from MFT 8.2.1 or above. Valid values are: • false: MFT rejects requests where the header host name does not match the host in the request line. This causes problems if older versions of the file transfer jar files (NonGUIApplet_0.0.0.1.jar or JavaApplet_0.0.0.1.jar) are used. • true: MFT accepts requests where the header host name

Parameter	Description
	does not match the host in the request line. This will allow older versions of the file transfer jar files (NonGUIApplet_0.0.0.1.jar or JavaApplet_0.0.0.1.jar) to be used. This is the default value for MFT 8.2.1, but may be changed to false in a future release.
`clientAuth`	This parameter defines whether the MFT Server supports HTTPS certificate authentication. Valid values are: • false: Certificate authentication is not supported. This is the default value. • want: Certificates are requested from HTTPS client, but are not required. This is the value that we suggest signing when you want to perform HTTPS Certificate Authentication. • true: Certificates are required for HTTPS requests. But MFT can still use certificate or password authentication, based on the **System Configuration > HTTPS Client Authentication Method** parameter definition. Browser, REST, or Command-Line clients that do not have a certificate cannot log in.
`ciphers`	This parameter defines the TLS ciphers that are supported. The MFT installation fills in this field with secure ciphers. But you may want to limit the supported ciphers even more. For example, some customers remove CBC ciphers from the supported ciphers.
`sslEnabledProtocols`	This parameter defines whether TLSV1.0, TLSv1.1, or TLSv1.2 is supported. By default, the MFT Server sets this parameter to TLSv1.2.

web.xml Parameters

There are a variety of `web.xml` parameters that affect security mentioned in the following sections.

Referer HTTP request header

The Referer HTTP request header contains a complete or partial URL of the page that initiated the HTTP request. The Referer header allows MFT to identify the URL that initiated the MFT request. All MFT web pages are initiated from within the MFT application. This parameter allows you to reject HTTP requests that were initiated from another URL.

MFT has two web.xml parameters that allow you to set the referer header:

- `AllowedReferersForXferNavigation`: Used by the file transfer browser interface when navigating through a directory structure.
- `AllowedReferersAdminJSP`: Used by the Admin interface.

See the following table for more information on defining these parameters.

Parameter	Description
`admincc-service-enabled`	Enables Command Center Admin API REST calls. Default value is `True`. Only Command Center supports "`admincc`" calls.
`admin-service-enabled`	Enables Admin API REST calls. Both Command Center and Internet Server (if Admin server is enabled) support "admin" calls. Default value is `True`.
`AllowUserDefinedJavaClasses`	Defines whether admins can configure **Alert Action> Execute Java Class** and **Scheduler definition>Scheduler Job Type> Execute Java Class**.
`ft-service-enabled`	Enables File Transfer API REST calls. Only Internet Server supports "ft" calls.
`LoadBalancerIPAddressList`	When MFT is behind a load balancer and a request is received, MFT allows the X-Forwarded-For HTTP parameter. This allows MFT to extract the initiating HTTP IP address. Otherwise, MFT uses the IP address of the load balancer.

Parameter	Description
`AllowCustomServerDefinition`	Defines whether to allow users to transfer files to servers defined with a Server Type of "Custom". Setting this parameter to `False` does not allow servers to be configured as Custom servers and rejects transfer requests for Custom servers,
`AllowEmailServerDefinition`	Defines whether to allow users to transfer files to servers defined with a Server Type of "Email". Setting this parameter to `False` does not allow servers to be configured as Email servers and rejects transfer requests for Email servers.
`AllowLocalServerDefinition`	Defines whether to allow users to transfer files to servers defined with a Server Type of LOCAL. Setting this parameter to `False` does not allow servers to be configured as LOCAL servers and rejects transfer requests for LOCAL servers.
`AllowMailboxServerDefinition`	Defines whether you want to allow users to transfer files to servers defined with a Server Type of "Mailbox". Setting this parameter to `False` does not allow servers to be configured as Mailbox servers and rejects transfer requests for Mailbox Servers.
`MaxConnectionCntFTP`	Allows you to set a maximum number of connections for the MFT FTP server. This can help protect against Denial of Service attacks.
`MaxConnectionCntSSH`	Allows you to set a maximum number of connections for the MFT SSH server. This can help protect against Denial of Service attacks.
`MaxConnectionCntCF`	Allows you to set a maximum number of connections for the MFT Platform Server. This can help protect against Denial of Service attacks.
`MaxConnectionCntOFTP2`	Allows you to set a maximum number of

Parameter	Description
	connections for the MFT OFTP2 server. This can help protect against Denial of Service attacks.
`DenyLoginIds`	This parameter allows you to define users that are not authenticated. For example, the default values of "root,administrator" ignores authentication request for users root and administrator. You can define additional users in this list.
`SSOLoginRequired`	Allows you define whether SSO (OIDC or SAML) is required for all users.
`SSOExcludedUsers`	Allows you to define users that can log in without SSO when **SSOLoginRequired** is set to `True`.
`SSOAllowRest`	When set to `True`, this allows REST calls to be used without using OIDC.
`TLSCipherSuite`	Defines the ciphers used by MFT in any SSL/TLS connections. If you select the **Use Secure Ciphers Only** parameter during the installation process, this parameter is filled in with secure ciphers. When the FTP service is started, all secure ciphers supported are displayed. You can select any ciphers from the displayed list to add to this parameter. Multiple ciphers must be delimited with a comma. This parameter only applies to FTPS (FTP over SSL) and Platform Server SSL connections. HTTPS connections use the parameters in the `server.xml` ciphers parameter.
`TLSProtocols`	Defines TLS protocols that are supported by FTPS and Platform Server SSL. The valid values are: TLSv1, TLSv1.1, and TLSv1.2.

Parameter	Description
	By default, any TLS protocol is supported. Before changing this parameter, ensure that all FTPS and Platform Server clients and servers support the defined TLS protocol. This parameter only applies to FTPS (FTP over SSL) and Platform Server SSL connections. HTTPS connections use the parameters in the `server.xmlSSLEnabledProtocols` parameter.
`SSHCipherSuite`	Defines the ciphers supported by MFT SFTP client and servers. When the MFT SFTP service is started, all SSH ciphers supported are displayed. You can select the ciphers that you want to support. Multiple ciphers must be delimited with a comma. We suggest using the **SSH Algorithm Group** feature instead of using this parameter. This parameter is for compatibility with the older versions.
`SSHKeyExchange`	Defines SSH key exchange algorithms supported by MFT SFTP client and servers. When the MFT SFTP service is started, all SSH key exchange algorithms supported are displayed. You can select the key exchange algorithms that you want to support. Multiple key exchange algorithms must be delimited with a comma. We suggest using the **SSH Algorithm Group** feature instead of using this parameter. This parameter is for compatibility with the older versions.

Parameter	Description
	Note: By default, the diffie-hellman-group1-sha1 protocol is removed by MFT, because it is vulnerable to the logjam attack. Some old SFTP clients and servers require this parameter; therefore, occasionally you need to update this parameter to include this key exchange algorithm. You must include all key exchange algorithms that are supported.
`SSHDigestSuite`	Defines the digest (hash) suites supported by MFT SFTP client and servers. When MFT SFTP service is started, all SSH digests supported are displayed. You can select the digests that you want to support. Multiple digests must be delimited with a comma. We suggest using the **SSH Algorithm Group** feature instead of using this parameter. This parameter is for compatibility with the older versions.
`PasswordHashNew`	Defines the password digest used by MFT. You have to use the defined value of SHA-256.
`UnsecuredHTTPSupport`	Defines whether HTTP support is allowed. The default value is `No`, which indicates that HTTP support is not allowed and only HTTPS is accepted. If you require HTTP support, set this value to `Yes`. **Note:** When using HTTP, no encryption of credentials or data is performed.
`AllowedReferersForXferNavigation`	Adds HTTP referer checking to the JSP pages that are used to navigate the directory tree structure. In addition to the URL, you have to add the loopback address.

Parameter	Description
	This parameter is defined in the `web.xml` file. It only needs to be set in Internet Server instances. It is ignored in TIBCO MFT Command Center.
`AllowedReferersAdminJSP`	Adds HTTP referer checking to the Administrator JSP pages. In addition to the URL, you have to add the loopback address. This parameter needs to be set both in TIBCO MFT Command Center instances and Internet Server instances, where the Admin service is installed.
`DisplayFTPBanner`	Defines whether MFT displays FTP and SFTP banners. If this parameter is set to `Yes`, you can define the banners or welcome message displayed in the Admin Configure SSH Server and Configure FTP Server pages.
`Anonymous`	Defines whether an anonymous user can be used without authenticating the password. If you enter the value anonymous in this parameter, you must also create a user called anonymous. Because the password is not validated, you must not give the anonymous user access to any secure file or folders.
`user-data-constraint`	Allows you to redirect HTTP requests to HTTPS port. Uncomment the following parameter within the web.xml file in two separate locations. This automatically redirects HTTP requests to the HTTPS port. `<!-user-data-constraint> <transport-guarantee>CONFIDENTIAL</transport-guarantee> </user-data-constraint->`

Parameter	Description
SecurityFilter	Defines whether a browser can be allowed to render a page in a frame, an iFrame, or an object. This parameter prevents you from framing and clickjacking attacks. By setting this parameter to SAMEORIGIN, the browser can use the page in a frame if the server including it in a frame is the same as the one serving the page. By setting this parameter to DENY, all attempts to load the page in a frame fails. The default value is SAMEORIGIN.
ChangedPasswordEmailEnabled	Defines whether an email is sent to a user when the user changes the password. We suggest setting this parameter to Yes to notify the user that the password has been changed.

Changing the jSession Cookie name

The name used by the session ID should not be descriptive or offer unnecessary details about the purpose and meaning of the ID. TIBCO recommends changing the default session ID of the web development framework to a generic name, such as “id”.

Follow these instructions to change the name of the jSession cookie.

Procedure

1. Edit the following file:

   ```
   Edit file: <MFT-Install>/server/conf/Catalina/localhost/cfcc.xml
   ```

2. Add the following parameter to the context after the cookies="true" parameter.

   ```
   sessionCookieName="id"
   ```

3. Restart the TIBCO MFT Internet Server.

Server Configurations

Follow these recommendations to secure TIBCO MFT Internet Server through configurations.

Configuration in Admin Client

Remove unnecessary default users or unnecessary rights from these users.

- Assign only necessary rights to users.
- Use LDAP for authentication.
- Enable global password rules.
- Enable global lockout.
- Allow users to reset their passwords.
- Use the MFT delegated administration feature if possible.
- AdministratorRight must be limited to a selected few people.
- Assign the minimum right that a user needs to access the system.
- Be cautious running commands or Java class on an alert or scheduled job. Commands and java programs run under the rights of the MFT server process.
- Configure the time of a day and days of the week that transfers can be run.

Server Options: Server File Name Prefix

When defining a server, you can expand the **Server Options** section on the **Add Server** page and use the **Server File Name Prefix** parameter.

This parameter defines the directory that is prefixed to the server file name defined on the transfer definition. With this parameter, you can perform the following actions:

- Restrict user access to a particular directory.
- Ensure when a transfer definition is created, the transfer definition cannot access data outside of the defined directory.

This parameter can be used for all server types, but it is particularly important when defining a server of *Local type.

SFTP and FTP banners

Banner pages are displayed by MFT when you log in to the MFT SFTP and FTP servers. It is good practice to create a generic banner page that does not include the name of the software running or the release.

SMTP TLS communication

MFT supports TLS communication to SMTP servers when sending emails. MFT supports Implicit SSL and StartTLS. We suggest using Implicit SSL since it is more secure than StartTLS. However, this depends on the TLS support of the SMTP server.

Configuring Multi-factor Authentication (MFA)

To configure Multi-Factor Authentication (MFA) for browser HTTP configuration, go to **Configuration > Multi-Factor Authentication**.

You can configure two methods of Multi-factor Authentication for browser HTTP configuration:

- **Email**: An email is sent to a user that has an authentication code.
- **Google Authenticator**: The Google Authenticator app computes an authentication code.

Note: As a best practice, exclude one or more admin users from MFA, in case MFA fails. You can use the **Common MFA Configuration** and **MFA Excluded Users and IP Addresses** parameters to define users and IP address subnets that are excluded from MFA.

Configuring ReCaptcha

To configure ReCaptcha settings, go to **Configuration > System Configuration > ReCaptcha settings**.

On this page, you can define whether ReCaptcha is required for HTTP browser requests.

You can set the following options to enable ReCaptcha for the login tasks:

- **Enable ReCaptcha for Login**
- **Enable ReCaptcha for Forgot User**
- **Enable ReCaptcha for Forgot Password**
- **Enable ReCaptcha for Self Register**

Note: As a best practice, exclude one or more admin users from Re-Captcha, in case Re-Captcha fails. With the **ReCaptchaExcludedUsersList** web.xml parameter, you can define users that are excluded from Re-Captcha. If you change any web.xml parameter, you must restart the TIBCO MFT Internet Server or TIBCO MFT Command Center before this parameter takes effect.

Define Database Password Using Environment Variable

The `web.xml` **DBPass** parameter includes an obfuscated database password. The most secure way to define **DBPass** is to programmatically extract the database password from a password vault and set an environment variable to define the password.

Note: The password vault program is not supplied by TIBCO.

MFT supports the `COM_TIBCO_MFT_CE_DB_PWD` environment variable that overrides the **DBPass** parameter.

The following formats are valid for the **DBPass** environment variable:

- `COM_TIBCO_MFT_CE_DB_PWD=ObfuscatedPassword`
- `COM_TIBCO_MFT_CE_DB_PWD=PWD:Obfuscated Password`
- `COM_TIBCO_MFT_CE_DB_PWD=B64:Base64EncodedPassword`
- `COM_TIBCO_MFT_CE_DB_PWD=CLR:ClearTextPassword`

To obfuscate the database password and set the environment variable, perform the following steps:

Procedure

1. Update the script to extract the password from a password vault program.
2. Run the following command from the MFT-Install directory. For example, if *abc123* is

the extracted password, run the following command:

```
pushd cloud/dbconfig; export COM_TIBCO_MFT_CE_DB_
PWD=$(./clouddbconfig.sh encrypt abc123) ; popd
```

Note:

- If you choose to define the database password through an environment variable, change the value of the **DBPass** parameter in the web.xml file to an empty string.
- The web.xml file is located in the following directory: `<MFT-Install>/server/webapps/cfcc/WEB-INF`

Define server.xml Keystore Password Using Environment Variable

The server.xml **keystorePass** parameter includes an obfuscated keystore password. The most secure way to define **keystorePass** is to programmatically extract the keystore password from a password vault and set an environment variable to define the password.

MFT supports the `COM_TIBCO_MFT_CE_KEYSTORE_PWD` environment variable that overrides the **keystorePass** parameter.

The following formats are valid for this environment variable:

- `COM_TIBCO_MFT_CE_KEYSTORE_PWD=ObfuscatedPassword`
- `COM_TIBCO_MFT_CE_KEYSTORE_PWD=PWD:Obfuscated Password`
- `COM_TIBCO_MFT_CE_KEYSTORE_PWD=B64:Base64EncodedPassword`
- `COM_TIBCO_MFT_CE_KEYSTORE_PWD=CLR:ClearTextPassword`

To obfuscate the keystore password and set the environment variable, perform the following steps:

Procedure

1. Update the script to extract the password from a password vault program.
2. Run the following command from the MFT-Install directory. For example, if *abc123* is

the extracted password, run the following command:

```
pushd cloud/dbconfig; export COM_TIBCO_MFT_CE_KEYSTORE_
PWD=$(./clouddbconfig.sh encrypt abc123) ; popd
```

Note:

- If you choose to define the keystore password through an environment variable, you should change the value of the keystorePass parameter in the server.xml file to an empty string.
- The server.xml file is located in the following directory: `<MFT-Install>/server/conf`

Define Encryption Key Environment Variable

By default, MFT uses an internal AES256 symmetric encryption key when encrypting passwords stored in the database. A unique key is generated for each new database installation. Optionally, you can use an environment variable to define the Encryption key.

The most secure way to define an encryption key is to programmatically extract the **Encryption Key** from a password vault and set an environment variable to define the key.

MFT supports the `COM_TIBCO_MFT_ENCRYPT_KEY` environment variable that overrides the internally generated AES256 symmetric encryption key.

Note: You can enter a passphrase in this environment variable. This passphrase is used to generate the encryption key. For example, you can enter a 1024 byte or a longer passphrase, provided that the passphrase is identical on all TIBCO MFT Internet Server and TIBCO MFT Command Center instances.

The following formats are valid for this environment variable:

- `COM_TIBCO_MFT_ENCRYPT_KEY=ObfuscatedPassword`
- `COM_TIBCO_MFT_ENCRYPT_KEY=PWD:ObfuscatedPassword`
- `COM_TIBCO_MFT_ENCRYPT_KEY=B64:Base64EncodedPassword`
- `COM_TIBCO_MFT_ENCRYPT_KEY=CLR:ClearTextPassword`

Perform the following steps to obfuscate the Encryption key and set the environment variable:

Procedure

1. Update the script to extract the encryption key from a password vault program.
2. Run the following command from the MFT-Install directory. For example, if abc123 is the extracted encryption key:

```
pushd cloud/dbconfig; export COM_TIBCO_MFT_ENCRYPT_KEY=$(./clouddbconfig.sh
encrypt abc123) ; popd
```

Note:
- The password vault program is not supplied by TIBCO.
- Encryption of passwords using the encryption key defined by this environment variable only occurs when the following conditions are met:
  - All Internet Server and Command Center instances have been upgraded to version 8.5.0 or higher
  - All Internet Server and Command Center instances have set this environment variable to the same value

When the `COM_TIBCO_MFT_ENCRYPT_KEY` environment variable is defined, passwords are encrypted using the passphrase defined by this environment variable. MFT can identify that a password is encrypted using this environment variable and decrypts the password using the same passphrase.

The following errors occur if an MFT Internet Server or Command Center instance defines a different environment variable, or the environment variable is not set:

- The MFT instance fails to decrypt the password.
  - All the transfers fail for an Internet Server instance, regardless of whether the required passwords were encrypted using the new encryption key.
  - The **Manage** function in the Internet Server and Command Center admin page receives an exception and no entries are displayed in the results table.
- Password encryption does not use the new encryption key environment variable until each server defines and uses the same environment variable.

The new `ThrowEnvKeyPwdException` web.xml parameter defines whether an exception is prompted when the Internet Server or Command Center cannot decrypt a password. The following values are valid for the `ThrowEnvKeyPwdException` web.xml parameter:

- **true** - This is the default value. When the parameter is set to **true** MFT generates an exception when encryption fails.
- **false** - When the parameter is set to **false** MFT does not generate an exception when encryption fails, it continues with a null password. Admin manage functions work, but transfers dependent on the passwords that are not decrypted fail.

MFT detects and reports on problems with the `COM_TIBCO_MFT_ENCRYPT_KEY` environment variable in the following ways:

- On the landing page, MFT displays an information message under the following conditions:
  - All servers have defined the same value for the `COM_TIBCO_MFT_ENCRYPT_KEY` environment variable.
  - Some servers have defined the `COM_TIBCO_MFT_ENCRYPT_KEY` environment variable, but some have not defined it.
  - Some servers have defined different values for the `COM_TIBCO_MFT_ENCRYPT_KEY` environment variable.
- In the diagnostics page, a new section displays if any server has defined the `COM_TIBCO_MFT_ENCRYPT_KEY` environment variable. The **Environment Variable Key Values** section displays the SHA512 message digest for the encryption key on each Internet Server or Command Center instance.
- A message is written to the `catalina.out` file when you cannot decrypt a password because the server does not have the correct encryption key.

Note: You must remember the `COM_TIBCO_MFT_ENCRYPT_KEY` passphrase. If you lose or forget the `COM_TIBCO_MFT_ENCRYPT_KEY` passphrase, MFT cannot decrypt the passwords using the encryption key. To use that MFT instance you need to set the web.xml `ThrowEnvKeyPwdException` parameter to **false** and change all passwords using the admin pages. There is no alternate way to decrypt the password if the passphrase is lost. You must change all passwords encrypted with the lost passphrase.

Disable User Defined Java Classes

An admin can configure user defined Java classes in the following ways:

- **Custom Server definitions**
- **Alert Action>Execute Java Class**
- **Scheduler Job Type>Execute Java Class**

When a user-defined java class runs in the MFT sandbox, the java class has access to many objects in the sandbox as well as System Environment variables and Java Properties. We suggest setting the `AllowCustomServerDefinition` and `AllowUserDefinedJavaClasses` web.xml parameters to **false**. This disallows the admin from configuring TIBCO MFT Internet Server or TIBCO MFT Command Center to run user-defined java classes.

```
AllowCustomServerDefinition     Set to False
AllowUserDefinedJavaClasses     Set to False
```

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 Product Documentation website, mainly in HTML and PDF formats.

The Product Documentation website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The documentation for this product is available on the TIBCO® Managed File Transfer Internet Server Documentation page.

How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our product Support website.
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the product Support website. If you do not have a username, 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 CLOUD SOFTWARE GROUP, INC. (“CLOUD SG”) SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, “INCLUDED SOFTWARE”). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

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

TIBCO, the TIBCO logo, the TIBCO O logo, and Slingshot 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.

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. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG’s Third Party Trademark Notices (https://www.cloud.com/legal) for more information.

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.

Cloud SG 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 a specific version of Cloud SG software on a specific operating system platform.