FME® Server Security

SAFE SOFTWARE

Table of Contents

Safe Software's entire business is built on data, and we understand that it is among the most important assets of any organization. The security and privacy of your data is our highest priority.

FME Server

FME Server brings FME to the enterprise, and an increased focus on security. Enhanced security can often mean a compromise on ease of use and deployment, but with FME Server, advanced security features are balanced with a product that is simple to use and install.

There are 4 main components to FME Server security:

1. Authentication – Access Control
2. Authorization – Roles and Policies
3. Data – Data Security
4. Network – Transmission Security

FME Server is already trusted and used by many organizations in the oil and gas, local and federal government, utilities, higher education, and military sectors. This document describes how FME Server provides comprehensive security.

Authentication – Access Control

To prevent unauthorized access, the first level of security is to establish the user's identity. This process is referred to as authentication. FME Server supports 3 types of authentication: Default Security, Active Directory, and Trusted, as well as an option to allow un-authenticated access to the system.

Default Security

FME Server ships with an integrated security component providing user management and authentication services. This default is generally applied when you are not planning to use Active Directory or when you are deploying outside your organization's firewall (e.g. on FME Cloud). When Default Security is enabled, the FME Server is responsible for managing the entire authentication process.

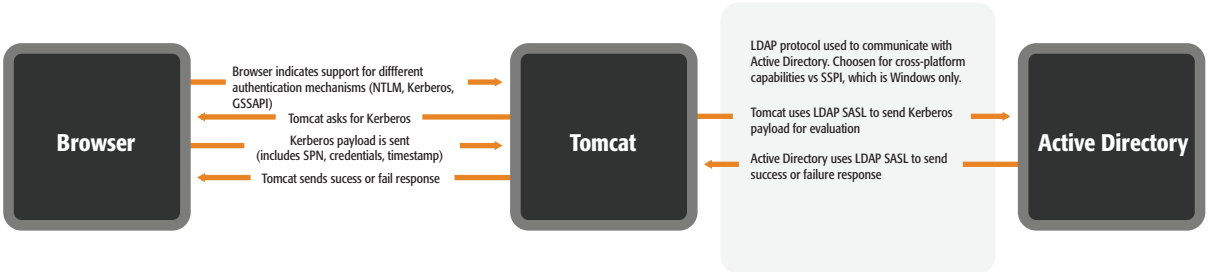
An admin can create and manage users from the web user interface. Once users are created, they can log in by manually entering their credentials. Users can also be created and destroyed programmatically using the FME Server REST API. This means that creating and managing FME Server logins can be automated and tied into your provisioning process.

Active Directory

When Active Directory is enabled for authentication, all usernames and passwords are managed by Active Directory. FME Server passes credentials to the Active Directory server, but it does not participate in the authentication process. The integration works by effectively mapping Active Directory security groups to FME Server user accounts or roles. FME Server does not store passwords when configured for Active Directory authentication.

Integrated Windows Authentication (IWA), commonly referred to as single sign-on, is also available with Active Directory. This removes the need for users to log in to the FME Server; their Windows login credentials are seamlessly passed along to FME Server and ultimately Active Directory. IWA is supported in the FME Server Web User Interface and in the publish wizard in FME Workbench.

FME Server – Integrated Windows Authentication



Once the authentication has been configured, system administrators (as with default security) have the ability to fine-tune the authorization and access to the different components of FME Server (i.e. set up roles, grant policies, and assign users to the roles).

Trusted Authentication

If you are developing a custom application, FME Server token security provides a robust and simple way for trusted applications to interact with all or parts of FME Server.

For example, you may have an internal self-service application that allows users in your company to download data out of your corporate database in a specific format. Rather than make the user log in, you can set up a trusted relationship between the application and FME Server. When FME Server receives a request from a trusted application, it takes the token, performs the authentication process and carries out the request.

Token security works by providing an encrypted string that is passed with the request, bypassing the interactive need to log in to FME Server. Tokens can only be generated by an authenticated user with the correct policies. They are primarily used in combination with the REST API but all services (e.g. Data Download and Data Streaming) support token security. FME Server roles can be used to adjust the policies to which the token has access.

Tokens are one way SHA-1 encrypted hashes and can be set to expire. Tokens are based on three unique variables which ensures they are unpredictable. If HTTPS is enabled, all session information is stored in secure cookies.

Guest User or Anonymous (un-authenticated)

FME Server can be configured to allow anonymous access to the services that ship with FME Server. This is useful for providing access to unknown users such as the general public. For example, an FME Server workspace could be set up to allow members of the public to upload data into a centralized repository via a QA process. By default, the guest account is enabled on FME Server, however it only grants a limited number of policies.

Logging Out

There is no session timeout when logged in to FME Server. Users concerned about leaving their browsers exposed should log out of FME Server when not in use, or lock their computers manually or through a pre-configured timeout.

Authorization – Roles and Policies

In FME Server, a role is a group of one or more users. Policies define the activities that are granted for a specific role on each FME Server resource. A number of default roles ship with FME Server: fmeadmin, fmeauthor, fmeguest, fmesuperuser, and fmeuser. These roles are created around job functions and are based on how organizations traditionally use FME Server. However, you have complete control over roles; you can delete any of these and create your own.

These authorization capabilities allow an admin to implement fine-grained control over what content users can access as well as what actions (read, write, publish and remove) they can perform on the content. For example, a user assigned to the fmeauthor default installed role can publish workspaces to FME Server. They can also access the web user interface to interact with any of the workflows they have published. But they cannot access any admin tasks such as Security or Engine Management.

Once a user has correctly authenticated with FME Server, they are granted access to the system. When a user then accesses a specific component, FME Server security determines if any of the associated roles of the user have the correct policies assigned to perform the requested operation on the resource.

Permissions with respect to workspaces and data

In FME workflows, workspaces and data contain your intellectual property. FME Server comes with built-in security focusing specifically on these components.

Workspaces are managed via repositories on the FME Server, which is like a folder system on the server. It is at the repository level that permissions are assigned. Any user account with a role that has the 'manage repository permission' has control over which repositories users can access, as well as the actions (read, write, publish and remove) that can be performed on the objects within the repositories. For example, you could set permissions on a repository so users could see and run the workspaces but not download them. If a user creates a repository they are automatically granted full permissions on that repository.

Data can be uploaded with the workspace, in which case the same repository level permissions apply as above. It can be uploaded through the data upload service (via the web user interface or an API call) or uploaded into the resources folders which allows you to share data between workspaces. There are four default root resource folders: Backup, Data, Logs and Temp. Resource folders can also be created by an admin. As with repositories, roles can then be assigned to the folders to ensure only users with permission can access the data.

If a workspace contains a published parameter with a name containing password, it will be automatically encrypted (using RSA asymmetric encryption) before it is added to the database. This ensures passwords are encrypted in the database web user interface job history, job, engine, and server logs.

Network – Transmission Security

FME Server is deployed both on the internet and intranet. For internal deployments, securely transmitting data might not be of the highest priority because security is usually provided by preventing access to the network as a whole. However, it is important to securely transmit credentials across the network even with internal deployments, as more access is given to the outside world to access cloud services and send

notifications. For external deployments (including on FME Cloud), transmission security is critical to protect data and prevent malicious use of FME Server.

There are 3 main network interfaces to FME Server: Client to FME Server, FME Server to Database, and communication between the FME Server components. Each of these interfaces is described in more detail below.

Client to FME Server

HTTP Clients

The main client of FME Server is the web browser which interfaces with FME Server via the web user interface. Clients can also interact with FME Server via the REST API. This may or may not be a web browser, but the security implications discussed below are the same.

By default, FME Server uses standard HTTP requests and responses which are suitable for most intranet deployments. For internet or sensitive deployments, HTTPS can be configured using customer supplied security certificates. When SSL is enabled on FME Server, all content and communications between clients are encrypted and use the HTTPS protocol. FME Cloud uses HTTPS as default as the instances reside on the public web, and we provide the certificate.

If Integrated Windows Authentication (IWA) is enabled on FME Server, the two components can be configured to communicate data unencrypted or encrypted. If encryption is not required, the SSL connection can be disabled and LDAP (Lightweight Directory Access Protocol) used. If encryption is required, LDAPS (Lightweight Directory Access Protocol with SSL) can be enabled which encrypts communication via SSL over port 636. LDAP and LDAPS are equivalent to HTTP and HTTPS.

Web Application Server

FME Server's native Web Services are served by the Apache Tomcat application server and FME Server uses their SSL library. Tomcat is patched with the latest security updates with every major release and service pack. Oracle WebLogic can also be configured to work with FME Server.

Cross-origin resource sharing (CORS) is supported on FME Server. This allows web applications to be created that access FME Server Web Services or REST API functionality on a different domain. For example, if FME Server is on http://domain1/fmeserver and your application is on http:\\webapp.com, you will be able to connect using JavaScript. This access would normally be forbidden because of the same origin policy which permits scripts running on pages originating from the same site, but prevents access to DOM from different sites. CORS defines a way for the browser and the server to interact securely, and determine whether to allow the cross-origin request.

Other

The notification server that ships as part of FME Server allows you to connect over several different protocols to both receive and send data.

- **Email:** To ensure e-mail messages sent to FME Server are encrypted and secure:
  - The FME Server SMTP e-mail publisher supports SSL and TLS secured.
  - The FME Server E-mail IMAP Publisher supports SSL, TLS, and StartTLS.
- **JMS:** The Java Messaging Service protocol is secured with username and password. JMS servers with encryption enabled are also fully supported.
- **WebSocket Server:** The built in WebSocket Server can support either unsecure or secure connections. WebSockets over SSL/TLS (WSS), like HTTPS, are encrypted and protect against Man-in-the-Middle attacks. WSS can be configured using customer supplied security certificates.
- **FTP:** The FTP protocol is supported allowing files to be placed on an FTP after translation. Both FTPS (SSL/TLS encrypted) and FTPES (explicit FTP over SSL/TLS) are supported.
- **UDP:** The UDP protocol is open to any client that knows the configured UDP port of the publication, but the transmission of the data is guarded by "topic" security. FME Server will discard data unless it's configured to publish to a topic.
- **Push:** The push protocol is an HTTP subscriber that allows data to be sent to an HTTP endpoint. HTTPS is supported so data can be sent encrypted. If the endpoint requires authentication, a username and password can be supplied.

Communication Between FME Server and the Database

FME Server uses the database to store metadata related to the published workspaces and data, security, transformation, configuration, and jobs. FME Server uses JDBC to connect to the database. Server can also be configured with your own database (PostgreSQL, Oracle, SQL Server). However, since FME Server to database communication is usually behind a firewall, most customers will not encrypt this.

Specific sensitive data within the database is encrypted. Passwords and tokens used to authenticate with FME Server are saved, hashed, and salted in the database. Passwords defined in workspace published parameters are encrypted using RSA asymmetric encryption.

Other

Risk Assessments

Application design is a combination of secure design practices and regular audits. To ensure the security of FME Server, a third-party Certified Information Systems Security Professional (CISSP) was hired to complete an application and network security audit. This included network vulnerability scanning, penetration testing, and an architecture review.

Summary

FME Server provides a robust, secure way to transform and automate your data connections at an enterprise level. Customizable security settings enable you to configure FME Server to adapt exactly to your organization's needs. If you wish to discuss any aspect of FME Server security, please contact us and we will be more than happy to help.