Sterling Secure Proxy®

Configuration Guide

Version 3.2

Sterling Secure Proxy Configuration Guide

First Edition

Sterling Commerce, Inc.

4600 Lakehurst Court Dublin, OH 43016-2000 * 614/793-7000

Contents

Chapter 3 Manage Certificates for SSL/TLS Transactions with Trading Partners 41

Chapter 4 Store System Certificates on a Hardware Security Module (HSM) 49

Chapter 5 Manage SSH Keys for SFTP Transactions   71

Chapter 8 FTP Reverse Proxy Configuration   121

Chapter 9 HTTP Reverse Proxy Configuration   149

Contents

Chapter 11 PeSIT Proxy Configuration 199

Chapter 15 Start and Stop Remote Perimeter Servers   251

Chapter 16 Prepare for Production   253

Chapter 17 Manage Certificates Between SSP Components   255

Appendix B PeSIT Field Definitions   345

SSP Overview

Sterling Secure Proxy (SSP) acts as an application proxy between Connect:Direct nodes or between a client application and a Gentran Integration Suite (GIS) server. It provides a high level of data protection between external connections and your internal network. Define an inbound node definition for each trading partner connection from outside the company and an outbound node definition for every company server to which SSP will connect.

General Proxy Terminology

Following is an illustration of the SSP general proxy environment:
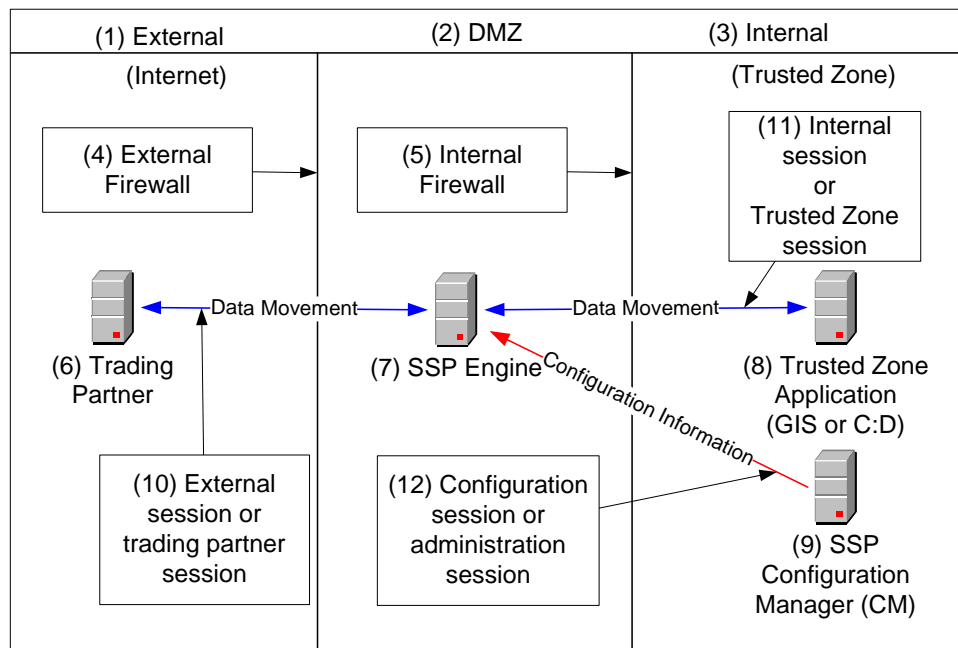
The following table describes the terminology used in the illustration:

#	Term	Description
1	External Network (Internet)	Network providing connectivity for trading partners to your network. This network is usually the Internet and is called the Internet in this documentation. The external network can also be a private network.
2	DMZ (Demilitarized Zone)	The part of the network that is neither the internal network nor the Internet. It is a network between the two networks. SSP is deployed in the DMZ and provides authentication before a trading partner can access information in the trusted zone.
3	Internal Network (Trusted Network or Trusted Zone)	The internal network behind the internal firewall and secure from outside networks.
4	External Firewall (Outer Firewall)	The firewall between the public network (Internet) and DMZ.
5	Internal Firewall (Inner Firewall)	The firewall between the DMZ and trusted zone.
6	Trading Partner	The external entity that you do business with. Trading partner may also be referred to as remote trading partner, external trading partner, or remote client.
7	SSP Engine	SSP includes two parts: an SSP engine and Configuration Manager (CM). The engine is deployed in the DMZ. It authenticates trading partners and information that is transmitted between the trading partner and trusted zone.
8	Trusted Zone Application	The application in the trusted zone with which the trading partners exchanges information. It is either Gentran Integration System (GIS) or a Connect:Direct server. It is also called the end point, destination node, or internal application.
9	SSP Configuration Manager (CM)	SSP includes two components: an engine and Configuration Manager. Configuration Manager resides in the trusted zone and configures the engine to perform its duties.
10	External Session (Trading Partner session)	The session between the remote trading partner and SSP.
11	Internal Session (Trusted Zone session)	The session between SSP and the trusted zone application.
12	Configuration Session (Administration session)	The session between CM and the engine that CM is configuring. This session is used by CM to push the configuration to the engine.

About a Reverse Proxy

A reverse proxy acts on behalf of a trusted zone application. The trading partner or remote client initiates a connection to a trusted zone application and is connected to a reverse proxy.

SSP provides reverse proxy services for GIS when the trading partners initiate FTP, HTTP, SFTP, and Connect:Direct sessions to the GIS server in the trusted zone.

SSP provides reverse proxy services for Connect:Direct servers when the trading partners initiate Connect:Direct sessions to Connect:Direct servers in the trusted zone.

Following is an illustration of SSP, labeled with reverse proxy terminology.
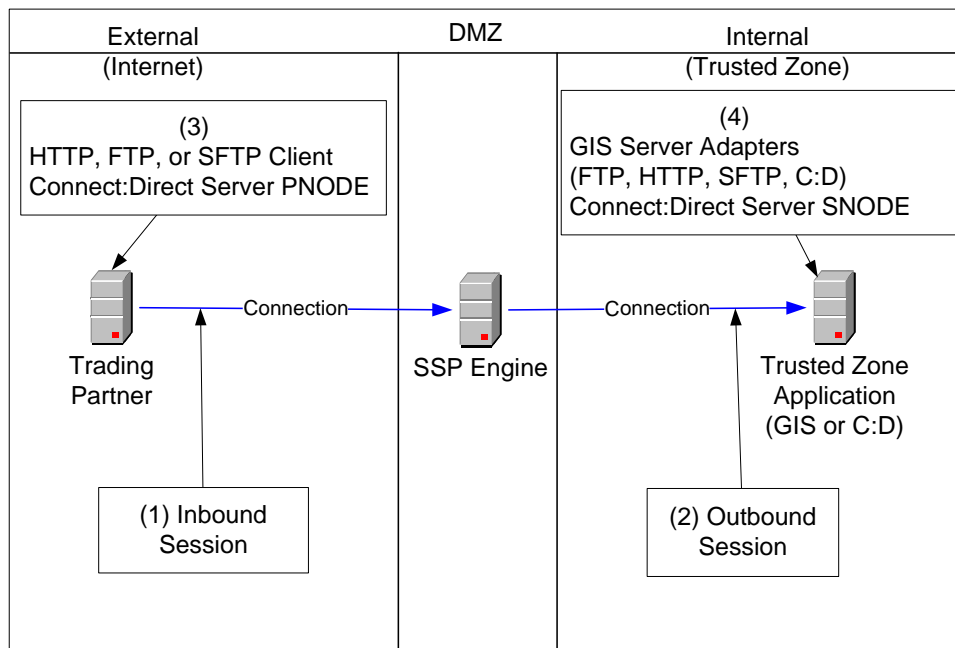
The following table explains the reverse proxy terminology used in the illustration.

#	Term	Description
1	Inbound Session	The session between the remote trading partner and SSP. It is inbound to SSP.
2	Outbound Session	The session between SSP and the GIS or Connect:Direct application in the trusted zone. It is outbound from SSP.
3	Client or Connect:Direct PNODE	The trading partner. It can be an HTTP, FTP, SFTP, or Connect:Direct client. For Connect:Direct implementations, the client is the PNODE or the initiating node.
4	Server Adapters or Connect:Direct SNODE	The GIS server in the trusted zone. The adapters at GIS are the HTTP server adapter, FTP server adapter, SFTP server adapter, and the Connect:Direct server adapter (SNODE). For Connect:Direct implementations, the trusted zone server is the SNODE.

About a Forward Proxy

A forward proxy participates in connections that originate from the trusted zone. The client in the trusted zone connects to the forward proxy in the DMZ and the forward proxy sends connection information to the destination application at the remote trading partner. SSP provides forward proxy services for Connect:Direct servers when the node in the trusted zone initiates a session to a server at a remote trading partner.

Following is an illustration of SSP, labeled with forward proxy terminology.
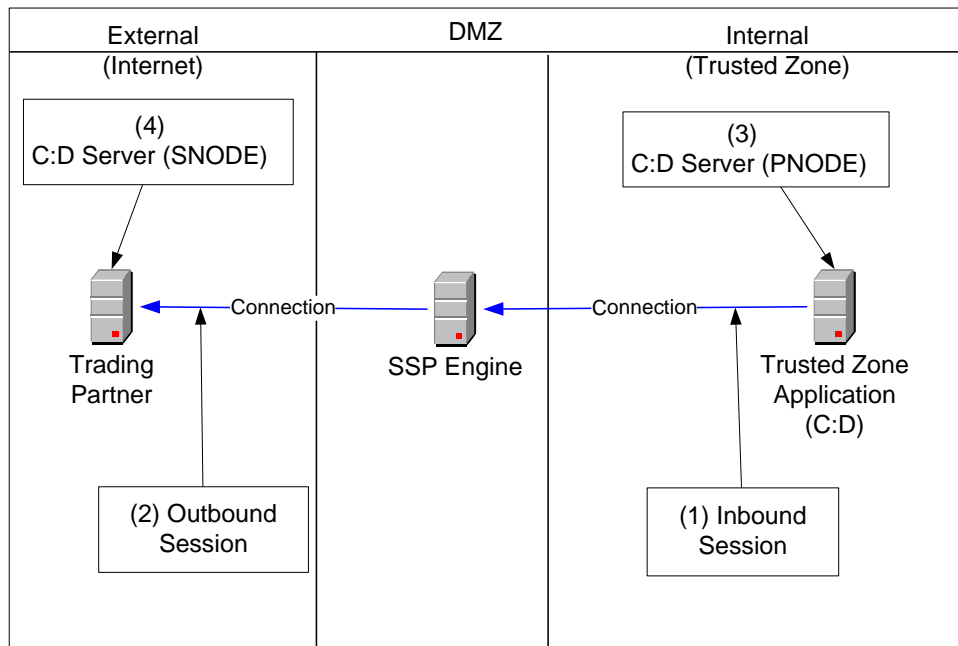
The following table describes forward proxy terminology used in the illustration.

	Term	Description
1	Inbound Session	The session between the GIS or Connect:Direct application in the trusted zone and SSP and is inbound to SSP.
2	Outbound Session	The session between SSP and the remote trading partner. It is outbound from SSP.
3	Connect:Direct PNODE	The Connect:Direct node in the trusted zone that initiates the session.
4	Connect:Direct SNODE	The Connect:Direct server at the trading partner.

About SSL Session Break

The SSL session break is a primary SSP security feature. SSP authenticates a remote trading partner in the DMZ, before creating a separate SSL session into the trusted zone. This allows you to create firewall rules to prevent trading partners from obtaining direct access to your application in the trusted zone. It also allows you to keep sensitive data out of the DMZ.

The SSL session break occurs because the trading partner connects to SSP in the DMZ and not to the application in the trusted zone. The trading partner is unaware that SSP is deployed and believes it is connecting to your backend system. SSP negotiates an SSL session with the remote trading partner and authenticates the trading partner's certificate, if SSL client authentication is configured. SSP then enforces user authentication to validate that the trading partner uses a valid user ID and password. After the SSL session is established and the user ID and password is authenticated, SSP initiates a separate SSL session to the application in the trusted zone. After the application in the trusted zone authenticates SSP via SSL client authentication and user ID and password authentication, SSP communicates messages between the trading partner and trusted zone application connection to allow the remote trading partner to move data into and out of the trusted zone application.

Following is a sample SSL session break flow:
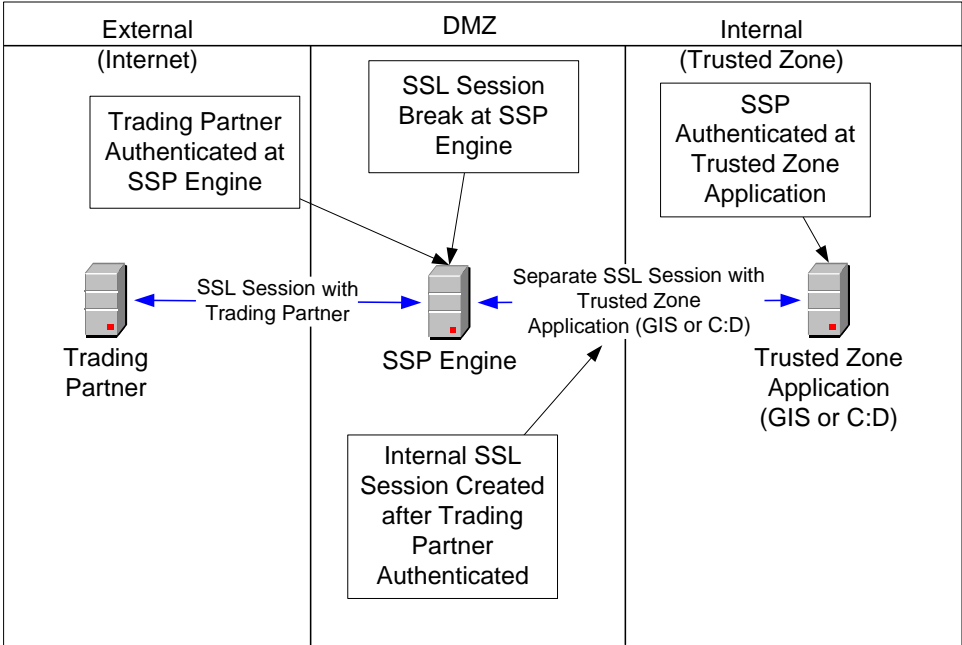
About SSH Session Break

Just as SSP creates an SSL session break for the HTTP, FTP, and Connect:Direct protocols, it creates an SSH session break when using the SFTP protocol. The SSH session break occurs because the trading partner connects to SSP in the DMZ and not to the application in the trusted zone. The trading partner is unaware that SSP is deployed and believes it is connecting to your backend system. SSP negotiates an SSH session with the remote trading partner and authenticates the trading partner's key and/or password as part of the SSH negotiation. After the SSH session is established, SSP initiates a separate SSH session to the application in the trusted zone. After the application in the trusted zone authenticates SSP using key and/or password authentication, SSP relays messages between the trading partner connection and the trusted zone application connection to allow the remote trading partner to move data into and out of the trusted zone application.

Following is a sample flow of an SSH session break:
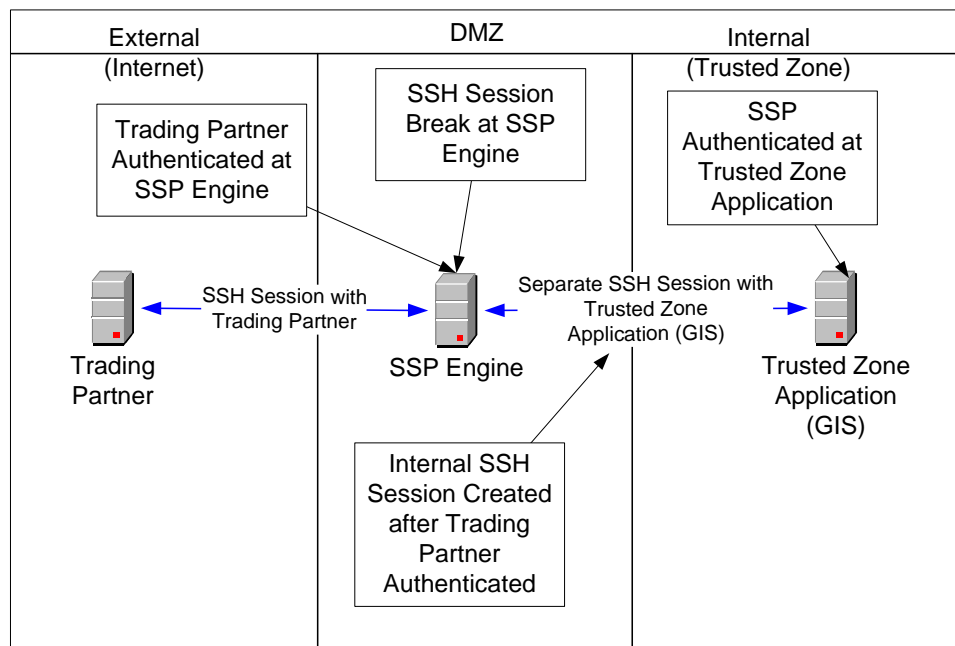
Using Digital Certificates

SSP uses X.509 digital certificates for secure data transport. Before you set up trading partner information, you must obtain and check in any digital certificates. Certificates can be stored in the SSP store or on a Hardware Security Module (HSM). An HSM is a hardware-based security device that generates, stores, and protects cryptographic keys. SSP provides support for the Eracom and nCipher HSMs.

After you store system certificates on the HSM and import information about the system certificates stored on the HSM to the SSP store, all system certificates, including those in the store and on an HSM, are displayed and available when you configure SSP.

Configuration Overview

The SSP architecture requires that only the minimum amount of configuration information be stored in the DMZ. It includes two components: the Configuration Manager (CM) and an engine. Configuration data is stored at CM, is encrypted, and does not require a database. CM is installed on the internal or trusted network.

The engine resides in the DMZ and receives configuration data from CM. The engine stores engine properties on disk in the DMZ, and the files are encrypted. The engine properties contain the minimum information required to accept and secure a connection from CM. It includes the IP address and port that the SSP engine listens on for the CM connection. It also includes the SSL key certificate, trusted certificate, and encryption cipher that will be used to secure the connection with CM.

When the engine is first started, it does not have configuration information. It listens on the configured IP address and port for a connection from CM, which tries to connect to the engine at a configurable interval. When CM connects to the engine, they negotiate an SSL session and secure the connection.

After the channel is secure, CM pushes the configuration to the engine. The engine reads the configuration and starts the appropriate proxy services. When you update a configuration in CM, CM transfers the updates to the engine.

Following is an illustration of the SSP flow of a configuration push:

External   DMZ   Internal

1. CM initiates an SSL connection to the engine in the DMZ
2. CM reads configuration from configuration store and pushes it to the engine
3. All changes are sent from CM to the engine

SSP Engine ←—Configuration Information—— SSP CM

SSP Engine Properties (Encrypted)

SSP Configuration Store (Encrypted)

1) IP and Port for Configuration Connection
2) SSL Info for Configuration Connection (KeyCert, Trusted Certificate, Cipher)

1) Engine Configuration
2) Adapter Configuration
3) Netmap
4) User Store
5) Certificate Store
6) Key Store

SSP Architecture

SSP architecture is described below:

SSP Engine—the engine resides in the DMZ and contains the minimum components necessary to manage communications sessions. The engine configuration (SSP engine properties) is created at CM and pushed to the engine. It is stored in active memory and is never stored on disk in the DMZ. No web services or UI ports are open in the DMZ.

Configuration Manager (SSP CM)—Configuration Manager is installed in the trusted zone. Use this tool to configure your environment. When you save a configuration definition (SSP configuration store) at CM, it is pushed to an engine, using an SSL session. Configuration files are encrypted and stored on the computer where CM is installed.

Note: Only one Configuration Manager should update an engine definition.

SSP configuration store—This file is encrypted on disk and contains the following information:

✦ The user store with information on user credentials

✦ The system certificate store with the certificates used for SSL/TLS sessions

✦ The key store with the SSH keys

✦ The engine configuration store with all configuration information for the engine

SSP engine properties file—These files are encrypted and contain the following information:

✦ The IP and port number to listen on for connections from Configuration Manager

✦ SSL key certificate, trusted certificate, and encryption cipher used for the connection from Configuration Manager

Web server—Configuration Manager is installed with a web server. You open a browser and access CM through a web page to configure SSP and monitor the engine activity. The web server is installed when you install Configuration Manager.

Adapter—an adapter identifies the protocol allowed for connections from trading partners. You can accept connections from clients that use different protocols; however, you must define a different adapter for each protocol. A single engine can run multiple adapters. In an adapter definition, you identify the port on which to listen for connections, the netmap to use with the adapter, the security policy, and the routing method to use. If you are using External Authentication, you identify the EA server to use in the adapter definition. If you are using a remote perimeter server, you identify the perimeter server to use in the adapter definition.

Netmap—define a netmap to identify the trading partners authorized to communicate through SSP and the company servers where connections are made.

✦ For a Connect:Direct netmap, create a node definition for all Connect:Direct nodes that will communicate through SSP. The node definition identifies the IP address and port to be used by the node and the policy to associate with the node. If SSL or TLS security is required for the connection, configure the protocol options in the node definition. You can also enable node-level logging in the node definition.

✦ For HTTP and FTP netmaps, define an inbound node definition for trading partner connections from outside the company. The inbound node definition identifies the IP address or address pattern to allow for the connection and the policy to associate with the node. If SSL or TLS security is required, configure the protocol options in the node definition. You can also enable node-level logging in the inbound node definition.

✦ For HTTP and FTP netmaps, define an outbound node for every company server to which SSP will connect. An outbound node definition identifies the address and port used to connect to the company server and enables SSL or TLS if this is required. You can also enable node-level logging and failover support in the outbound node definition.

✦ For SFTP netmaps, define an inbound node definition for trading partner connections from outside the company. The inbound node definition identifies the IP address or address pattern to allow for the connection and the policy to associate with the node.You can also enable node-level logging in the inbound node definition.

✦ For SFTP netmaps, define an outbound node for every company server to which SSP will connect. An outbound node definition identifies the address and port used to connect to the company server, the known host key that is used to authenticate the company server to SSP, and the cipher suites and MACs used to secure the connection. You can also enable node-level logging and failover support in the outbound node definition.

Policy—define a policy to identify the security features to implement for an inbound node definition or a Connect:Direct node definition.

✦ In all protocol policies, you can enable the capability to authenticate the inbound connection and identify what user ID and password to use to connect to the secure company server.

✦ For FTP, HTTP, and Connect:Direct policies, you can enable the capability to authenticate certificate information using EA.

✦ In an HTTP policy, you can enable the capability to block commonly occurring HTTP exploits.

✦ In a Connect:Direct policy, you can enable the capability to send a warning message or stop a session if a protocol error occurs, as well as prevent a Connect:Direct node from performing a runtask, runjob, copystep, or submit step function.

✦ In an SFTP policy, you identify the method required to authenticate the inbound connection. Authentication methods supported are key, password, password or key, and password and key.

Sterling External Authentication Server (EA)—a separately installed feature of SSP, EA allows you to validate digital certificates passed by the client or trading partner during SSL/TLS session requests. You can also validate certificates against one or more certificate revocation lists (CRLs), and validate certificates based on a valid date range. See the Sterling External Authentication Server documentation for more information.

EA can be configured to validate certificates and authenticate users. The functions performed by EA are defined in an EA definition. EA performs one or more of the following functions:

✦ Certificate Validation

✦ Certificate Revocation List (CRL)—certificate revocation checking using a certificate revocation list (CRL)

✦ Multi-factor Authentication

✦ Certificate Policy Enforcement

✦ LDAP Authentication

✦ User ID mapping—remote trading partners can be given IDs and passwords that do not provide access to internal systems. The ID and password presented by the trading partner is mapped to an ID and password that can then access the internal system

✦ TAM (Tivoli Access Manager) Authentication

✦ Generic Authentication

Before you can use EA with SSP, you must configure EA server definitions in SSP. Then, when configuring policies and protocol adapters, you select these server definitions. You can also select security features available in EA such as certificate authentication, user authentication, and user mapping. Refer to the Sterling External Authentication Server help for more information.

Authenticate Trading Partners in the DMZ

SSP allows you to select an authentication method to meet your security requirements. The authentication mechanisms can be used together to enforce multi-factor authentication. Authentication options include certificate authentication, user authentication, and IP address checking

Certificate Authentication Options

You can authenticate a remote trading partner using certificate authentication. Certificate authentication uses SSL client authentication and is optional. Three methods of certificate authentication are available to allow you the flexibility to choose how you want to authenticate trading partners using x.509 certificates. Certificate authentication options include no authentication, local authentication, or authentication using EA.

Option	Description
No Certificate Authentication	You can configure SSP so that the remote trading partner certificate is not authenticated. Either disable SSL security or turn on SSL security but do not enforce SSL client authentication. In both configurations, SSP will not require the client to send a certificate for authentication.
	Choose this option to enforce the following security policy requirements:
	◆ You require a single factor of authentication in the DMZ and you want to authenticate a trading partner using user authentication.
	◆ You require an SSL session break in the DMZ but you do not want to authenticate the trading partner in the DMZ. In this case, you do not enforce SSL client authentication for the connection to SSP nor do you authenticate the user.
	◆ You do not require that the session with the remote trading partner be secured and you will not use SSL to authenticate and secure the session. In this case, the remote trading partner does not present a certificate and SSL client authentication is not available.

Option	Description
Local Certificate Authentication	If SSL client authentication is configured, SSP requests the client (trading partner) to present a valid certificate. The certificate is validated against the trusted root file as part of the SSL handshake negotiation. Choose this option to enforce the following security policy requirements: ◆ You require multiple factors of authentication in the DMZ and you choose to authenticate using SSL client authentication and user authentication. ◆ You require a single factor of authentication in the DMZ and you choose to authenticate using SSL client authentication. ◆ You want to authenticate using SSL client authentication and have not implemented EA to provide additional certificate validation.
Additional Certificate Authentication Using EA (Recommended)	If SSL client authentication is configured, SSP can perform additional authentication on the certificate presented by the client, using Sterling External Authentication Server (EA). SSP performs additional certificate checks using EA, in addition to the local certificate authentication completed by SSL. If you configure SSP to use EA for certificate authentication, it will send the certificate presented by the client to EA immediately after the SSL handshake is complete. Sample validations that EA can perform include: ◆ Certificate Revocation List (CRL) checking—validates that the certificate has not been revoked. ◆ Common name check or subject name lookup— validates that the certificate is issued to a trusted trading partner by looking up the name at your LDAP server. ◆ Binary comparison—compares the certificate received to a public certificate stored in your LDAP server. ◆ Bind certificate to an IP address—validates that the certificate and IP address are associated and that the certificate can be presented by the client at the IP address the connection to SSP was initiated from. ◆ Custom Exit—transmit the certificate to your java program to interface with internal certificate validation routines. Refer to the Sterling External Authentication Server Help for additional certificate validation options. Choose this option to enforce the following security policy requirements: ◆ You require multiple factors of authentication in the DMZ and you plan to authenticate the trading partner connection using SSL client authentication and user authentication. ◆ You require a single factor of authentication in the DMZ and you plan to authenticate the trading partner connection using SSL client authentication. ◆ You want to further authenticate the client certificate using a mechanism external to SSP.

User Authentication Options

Three methods of user authentication are available to allow the flexibility to choose how to authenticate users: no user authentication, authenticate users locally or authenticate user using EA.

Option	Description
No User Authentication	Select this option if you do not want to validate trading partner credentials in the DMZ. If you select this method, we recommend that you enforce SSL client authentication to provide at least one factor of authentication in the DMZ. If you select no user authentication, you may pass the user credentials through to the destination node in the internal network and validate the user credentials at the internal network. Choose this option to enforce the following security policy requirements:
	◆ You require a single factor of authentication in the DMZ and you plan to authenticate the trading partner using SSL client authentication. If you choose this option, be sure to pass the user credentials through to the GIS or Connect:Direct trusted zone application so it will authenticate the user and differentiate between users accessing the system.
	◆ You require an SSL session break or IP break in the DMZ and are not required to authenticate the trading partner in the DMZ. In this case, you do not enforce SSL client authentication nor does SSP authenticate the user. For this option, pass the user credentials to the external network in order for the trusted zone application (GIS or Connect:Direct) to differentiate between users accessing the system.
	◆ You implement a bulletin board type system, where the user credentials are not important. This option is not a typical implementation. Carefully evaluate your environment before using this configuration.
Authenticate Users Locally	Select this option to authenticate users using information in the SSP local user store. This option requires you to maintain the users in the SSP configuration. Select this option for the following security requirements:
	◆ To store and maintain users in the SSP user store.
	◆ You do not have an external infrastructure for user authentication to interface with.
Authenticate Users With EA (Recommended)	Select this option to perform external user authentication, using EA. This option sends the user credentials presented by the client to EA for authentication. Sample user authentication validations that EA can perform include:
	◆ Through LDAP to bind to user in LDAP
	◆ Through Tivoli Access Manager (TAM)
	◆ Through a customer java exit.
	Refer to the Sterling External Authentication Server Help for additional certificate validation options.
	Choose this option to enforce the following security policy requirements:
	◆ To maintain users in an application that is external to SSP.
	◆ You have an existing infrastructure to validate users against.
	◆ To use the user mapping provided by EA. Refer to the Sterling External Authentication Server Help.
	◆ To implement multi-factor authentication and bind the factors together in the LDAP infrastructure.

IP Address Checking (Netmap Check)

IP address checking validates the IP address of the trading partner and makes sure that the IP address is an allowed address. You perform IP address checking with SSP or through EA.

✦ Inbound Node List for the FTP, HTTP, and SFTP protocols—Use SSP to validate the IP address from which a remote trading partner connects. When a trading partner connects to SSP, SSP looks up the IP address in the inbound node list of the netmap. If the IP address is not found, the session ends.

You can specify wildcard characters in the inbound node list, to provide the flexibility to be as granular in your check as you require. For example, you can specify an entry of * in the inbound node list. This value allows connections from all IP addresses. If you specify an IP address for each trading partner in the inbound node list, only connections from the client IP addresses identified are allowed. The more specific the IP address is in the inbound node list, the stricter the IP address check is.

✦ Netmap Check for Connect:Direct—For Connect:Direct connections, the netmap contains one node list that is used for both inbound and outbound nodes. Connect:Direct does not use the IP address to find the netmap entry to use. It uses the node name provided by the initiating node (PNODE). However, a parameter in the Connect:Direct adapter allows you to check the IP address of the initiating node.

✦ External Authentication (recommended)—Validate the IP address using EA to perform certificate or user validation. If SSP is configured to use EA for user or certificate authentication, it sends the IP address to EA. EA validates the IP address and determines if the IP address is valid for a user or for a certificate subject name, common name, or other specified values in the certificate.

Summary of Authentication

SSP provides flexibility in how you authenticate users and connections.

The table below summarizes the options available in SSP and the factor of authentication for each:

	SSL Client Authentication Enforced			SSL Client Authentication Not Enforced		
	No User Authentication	**Users Authenticated Locally**	**Users Authenticated via EA**	**No User Authentication**	**Users Authenticated Locally**	**Users Authenticated via EA**
Pass Through User Credentials	Single factor authentication in DMZ (SSL client auth only)	Multi-factor authentication in DMZ	Multi-factor authentication in DMZ (recommended)	No authentication in DMZ	Single factor authentication in DMZ (user auth only)	Single factor authentication in DMZ (user auth only)
Outbound User Credentials Mapped from EA	N/A	N/A	Multi-factor authentication in DMZ	N/A	N/A	Single factor authentication in DMZ (user auth only)

	SSL Client Authentication Enforced			SSL Client Authentication Not Enforced		
	No User Authentication	**Users Authenticated Locally**	**Users Authenticated via EA**	**No User Authentication**	**Users Authenticated Locally**	**Users Authenticated via EA**
Outbound User Credentials from Outbound Node in Netmap	Single factor authentication in DMZ (SSL client auth only.) All users look the same at GIS or C:D in trusted zone.	Multi-factor authentication in DMZ. All users look the same at GIS or C:D in trusted zone.	Multi-factor authentication in DMZ. All users look the same at GIS or C:D in trusted zone.	No authentication in DMZ. All users look the same at GIS or C:D in trusted zone.	No authentication in DMZ. All users look the same at GIS or C:D in trusted zone.	Single Factor authentication in DMZ (user authentication only.) All users look the same at GIS or C:D in trusted zone.

Authenticating SSP to the Trusted Zone Application

After SSP authenticates the remote trading partner, it creates another session to the application in the trusted zone. For this connection, SSP is the client and is authenticated by the trusted zone application. SSP provides SSL client authentication and user authentication.

✦ SSL client authentication (recommended)—if you want to secure the session between SSP and the application in the trusted zone, you can require that SSP present a certificate during SSL client authentication. This certificate is authenticated by the trusted zone application during the SSL handshake. Use this option if you want to enforce the following security features:

   ◆ Secure the connection from SSP to the trusted zone application (recommended).

   ◆ You require multiple factors of authentication by the trusted zone application and will authenticate SSP, using SSL client and user authentication.

   ◆ You require a single factor of authentication by the trusted zone application, and you will authenticate SSP using SSL client authentication only.

✦ User authentication—SSP is required to provide user credentials when logging on to the application in the trusted zone. The following are user authentication options:

   ◆ Pass-through (recommended)—this option sends the user credentials presented by the trading partner to the application in the trusted zone for authentication. This mechanism allows the user identity to be maintained at the trusted zone application.

   ◆ EA Mapped User Credentials—the user credentials are mapped using EA. When SSP uses EA for user authentication, it receives the user credentials from the trading partner and sends them to EA for validation. If configured, EA returns the mapped user credentials, and SSP uses them to log on to the application in the trusted zone.

   ◆ Netmap—the user credentials are defined in the outbound node of the netmap that is used by SSP to establish a session with the application, in the trusted zone. SSP logs in to the

trusted zone application as the same user for all sessions. This method is not recommended.

Authentication and Flow Diagrams

This section provides diagrams of the SSP flow for the protocols.

Connect:Direct Reverse Proxy Diagrams

The following illustration describes the Connect:Direct reverse proxy authentication steps:
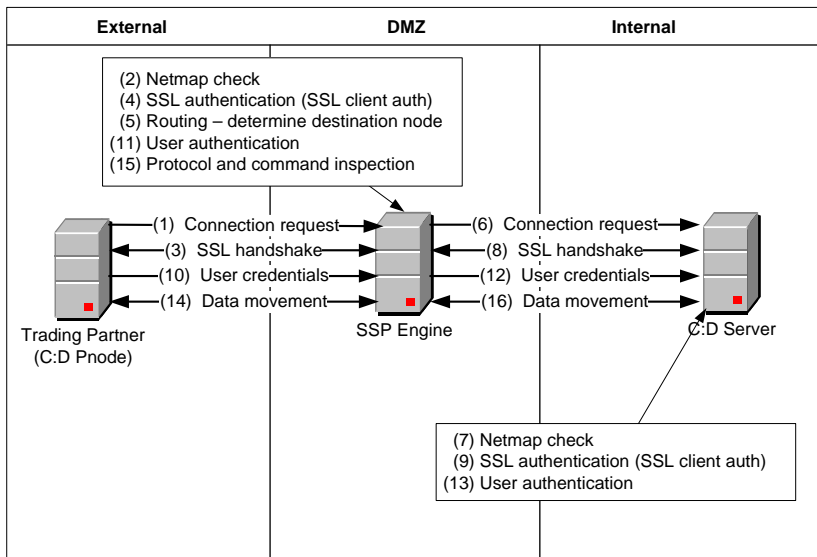
The following illustration describes a Connect:Direct reverse proxy authentication using EA:
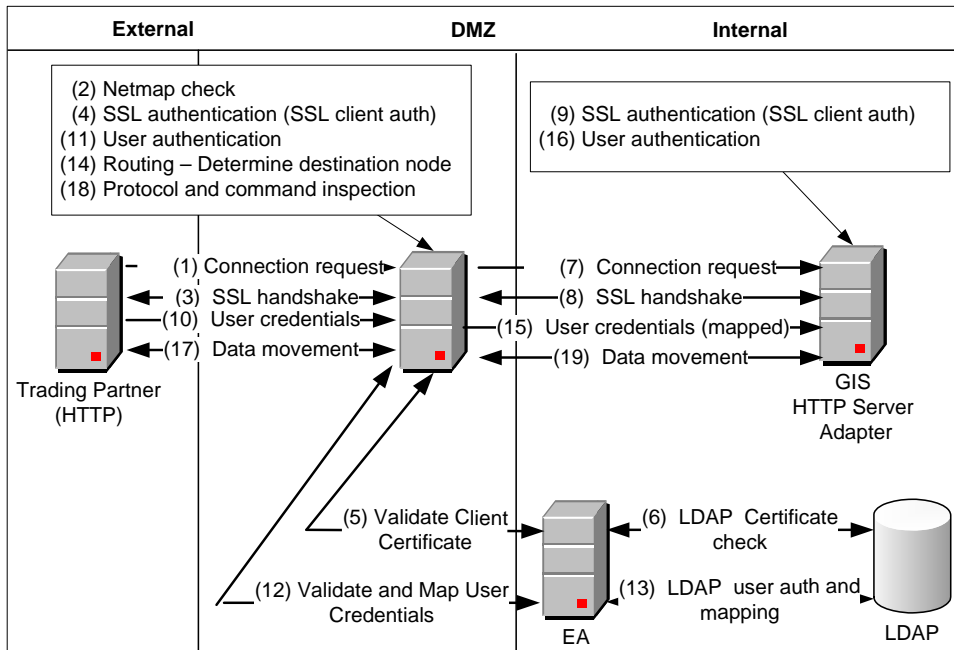
Connect:Direct Forward Proxy Diagrams

The following illustration describes the steps in a Connect:Direct forward proxy authentication:
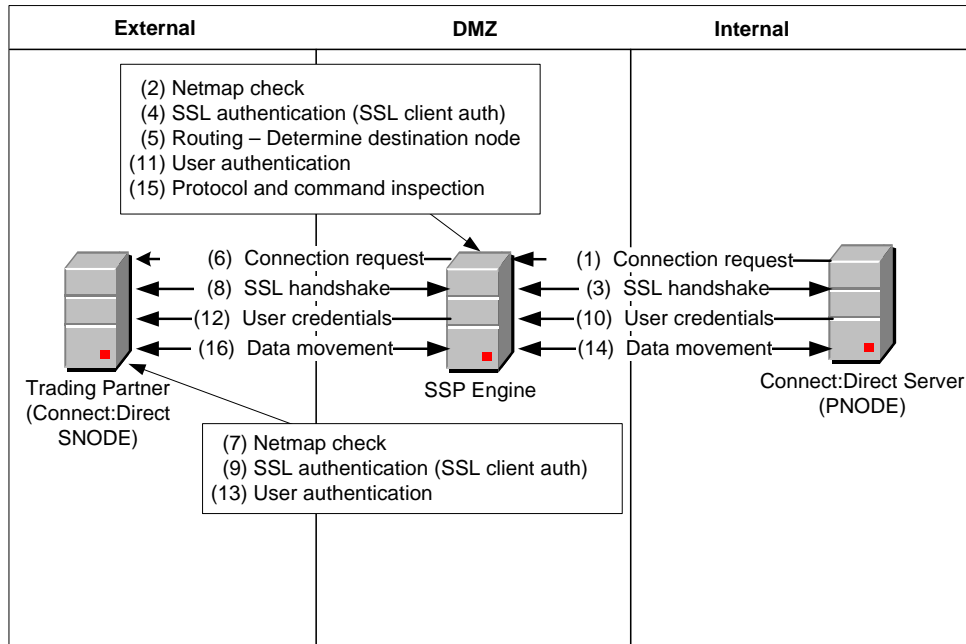
The following illustration describes the steps in a Connect:Direct forward proxy authentication using EA:
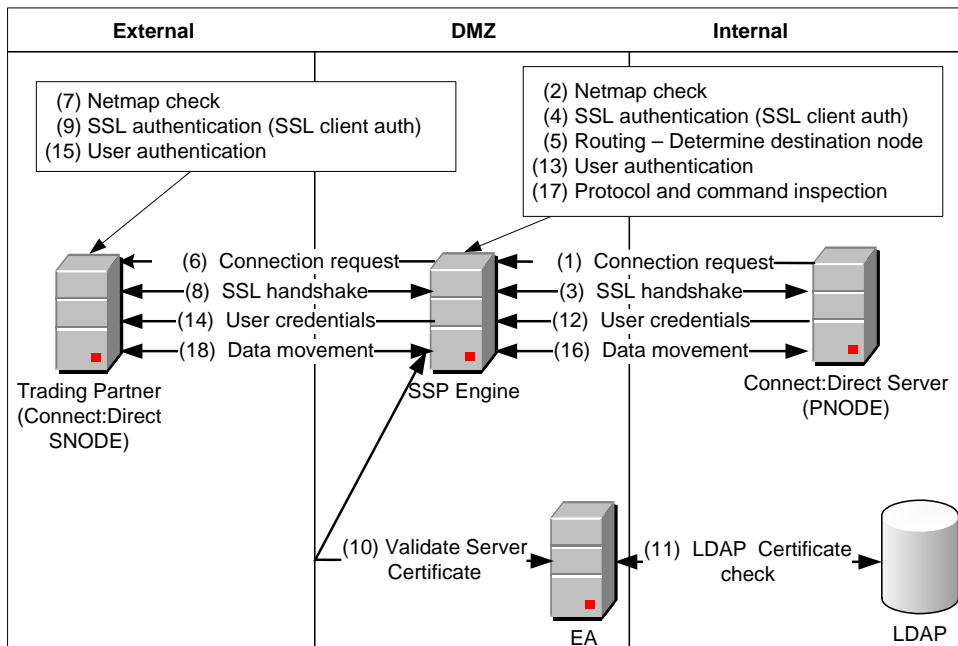
FTP Reverse Protocol

The following illustration describes the steps in an FTP reverse proxy authentication:
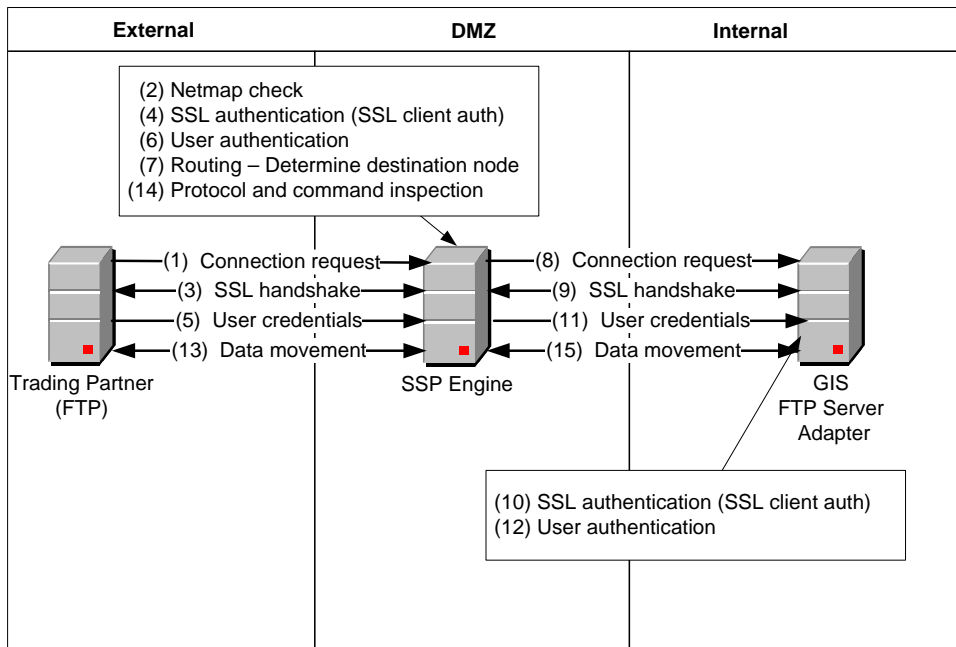
The following illustration describes the steps in an FTP reverse proxy authentication using EA:
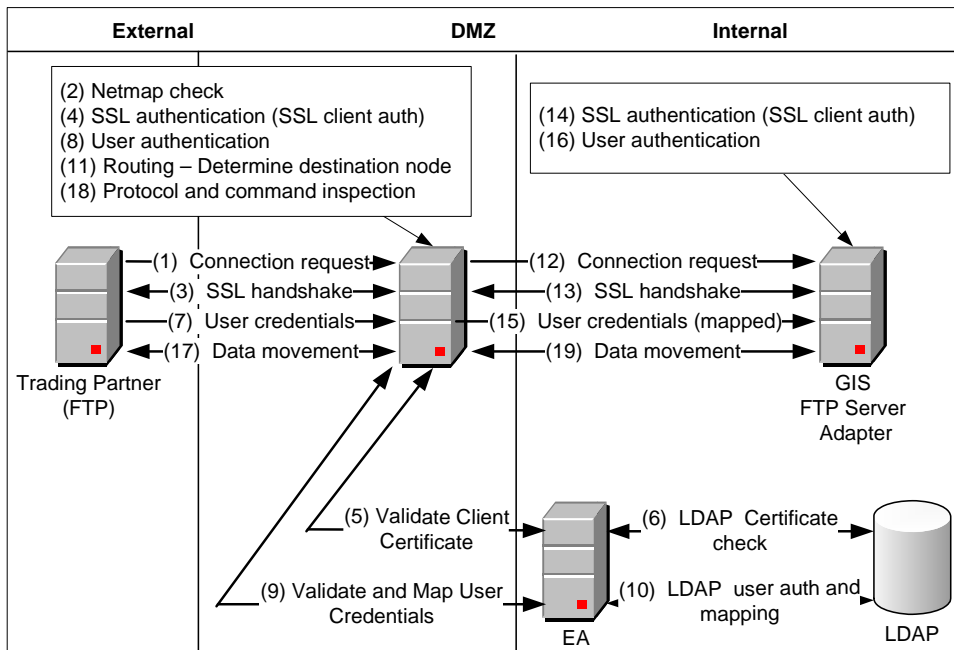
HTTP Reverse Proxy

The following illustration details the steps in an HTTP reverse proxy authentication:
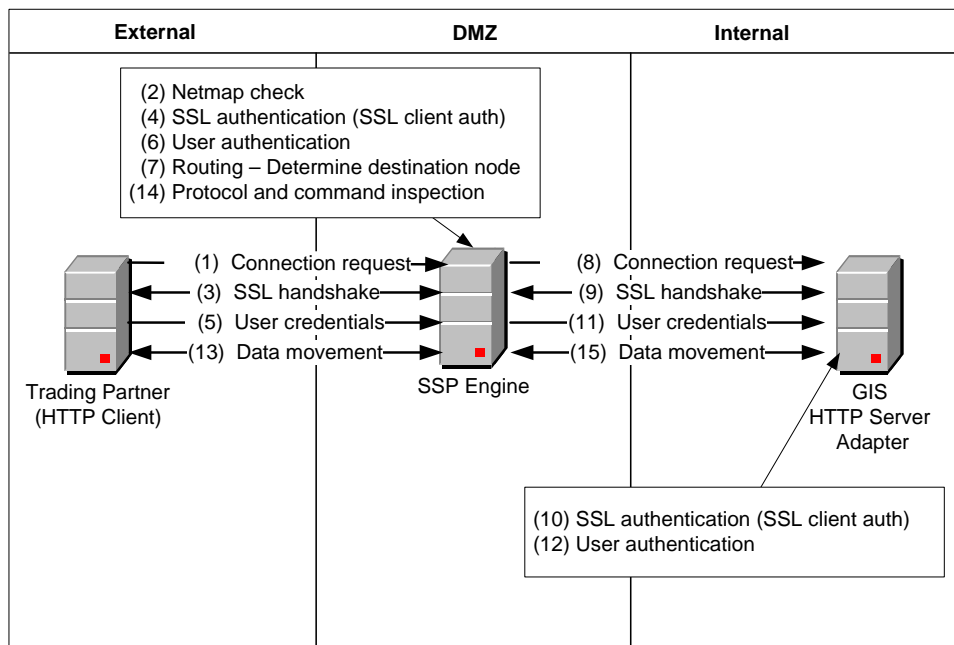
The following illustration details the steps of an HTTP reverse proxy authentication using EA:

External	DMZ	Internal

(2) Netmap check
(4) SSL authentication (SSL client auth)
(8) User authentication
(11) Routing – Determine destination node
(18) Protocol and command inspection

(14) SSL authentication (SSL client auth)
(16) User authentication

(1) Connection request
(3) SSL handshake
(7) User credentials
(17) Data movement

(12) Connection request
(13) SSL handshake
(15) User credentials (mapped)
(19) Data movement

Trading Partner
(HTTP)

GIS
HTTP Server
Adapter

(5) Validate Client Certificate
(6) LDAP Certificate check

(9) Validate and Map User Credentials
(10) LDAP user auth and mapping

EA

LDAP

SFTP Reverse Proxy

The following illustration details the steps of an SFTP reverse proxy authentication:

External	DMZ	Internal

(2) Netmap check
(4) SSH negotiation
(6) User authentication
(7) Routing – Determine destination node
(14) Protocol and command inspection

(1) Connection request
(3) SSH handshake
(5) User credentials
(13) Data movement

(8) Connection request
(9) SSH handshake
(11) User credentials
(15) Data movement

Trading Partner
(SFTP Client)

SSP Engine

GIS
SFTP Server
Adapter

(10) SSH negotiation
(12) User authentication
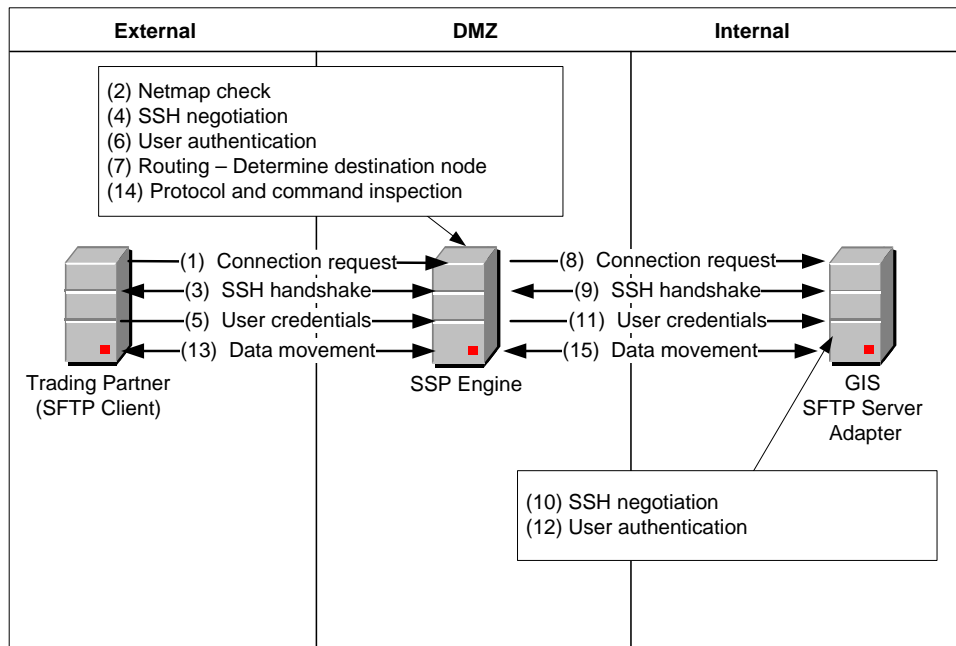
The following illustration describes the steps in an SFTP reverse proxy authentication with EA:
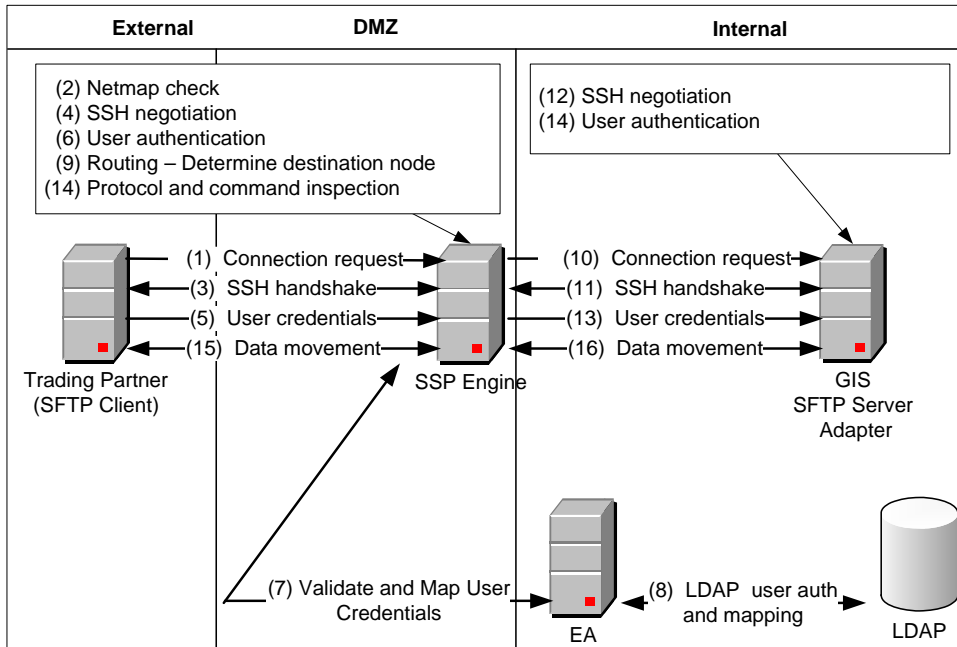
Plan Your SSP Configuration

Before you are ready to configure SSP, plan how you will implement your proxy environment and determine what level of security is required to access the server in the trusted zone.

Determine the Communications Protocol to Configure

SSP supports four protocols. Identify the protocol required for your environment, as defined below:

✦ Connect:Direct—if you are using SSP to communicate between two Connect:Direct nodes or between a Connect:Direct node and a GIS Connect:Direct server adapter, configure a Connect:Direct proxy adapter.

✦ FTP—configure an FTP reverse proxy adapter if you are using SSP to communicate between an FTP client and GIS.

✦ HTTP—configure an HTTP reverse proxy adapter if you are using SSP to communicate between an HTTP client and GIS.

✦ SFTP—configure an SFTP reverse proxy adapter if you are using SSP to communicate between an SSH client and GIS.

Follow the instructions in the scenario chapters to configure SSP for a protocol. If you plan to use more than one protocol, completely test and configure one protocol before adding a configuration for another protocol.

Identify Secure Session Requirements for a Connect:Direct, HTTP, or FTP Environment

If you are configuring a Connect:Direct, HTTP, or FTP environment, determine what secure communications sessions you will configure.

✦ Determine whether your environment requires a secure communications session between SSP and the inbound client and what security protocol is required for the session (SSL or TLS).

✦ Determine whether your environment requires a secure communications session between SSP and the outbound server and what security protocol is required (SSL or TLS).

✦   If you plan to use External Authentication for certificate or user authentication, determine whether your environment requires a secure communications session between SSP and EA and what security protocol is required for the session (SSL or TLS).

✦   If a secure connection is required, do the following:

◆   Generate a self-signed certificate or obtain a CA certificate.

◆   For a server and EA certificate, obtain the certificate or the root certificate and import the root certificate into the trust store.

◆   Import the private key and certificate into the system certificate store.

◆   For client authentication, obtain the public certificate or root certificate of the inbound trading partner node and import it into the trusted certificate store.

Refer to Chapter 3, *Manage Certificates for SSL/TLS Transactions with Trading Partners*, for instructions.

Identify Secure Session Requirements for an SSH (SFTP) Environment

If you are configuring an SFTP environment, determine what secure session options to configure. Authentication for an SFTP connection is performed with the exchange of session keys for the server and the client. To implement authentication for SFTP connections, you must create SSH key stores and import SSH keys into them. These key stores and keys can then be selected when you configure SFTP adapters.

✦   Public key server authentication is mandatory. Therefore, you must configure both a local host key and a known host key.

✦   To implement public key client authentication, you configure an authorized user key and a local user key.

Refer to Chapter 5, *Manage SSH Keys for SFTP Transactions*, for instructions.

Determine Validation Requirements for Inbound Trading Partners (Inbound Nodes) to SSP

Determine security policy requirements for the inbound connection. Security options include:

✦   Require no authentication.

✦   Configure inbound node matching to allow only specific hosts to connect to SSP.

✦   Validate the user ID by comparing it to information stored in the SSP local user store. Validate the certificate by comparing it to information in the SSP local certificate store.

✦ Validate the trading partner (client) user ID and/or certificate using EA. Some of the validation methods that can be implemented using EA include:

 ◆ Query an LDAP or HTTP server to validate dates and signature on an inbound certificate

 ◆ Authenticate user Common Name (CN) specified in the certificate of the inbound node or group name with which the CN is associated

 ◆ Validate attributes of the certificate against information stored on LDAP server

 ◆ Validate certificate against a certificate revocation list (CRL) stored on LDAP or HTTP server

 ◆ Authenticate the user ID and password submitted as logon credentials for the target server by comparing them against information stored on an LDAP or TAMS server and authorize access

Determine Connection Requirements for the Connection to the GIS Server or Connect:Direct Node (Outbound Node)

Identify requirements for connection to the secure outbound node. Possible requirements include:

✦ Not requiring that the user ID and password be authenticated in SSP. The user ID and password provided by the trading partner is passed to the GIS or Connect:Direct server for authentication.

✦ Connecting to the GIS or Connect:Direct server (outbound node) using a user ID and password stored in the SSP netmap configuration.

✦ Connecting to the GIS or Connect:Direct server (outbound node) using information accessed from EA. The EA server determines whether an alternate user ID and password mapped to the trading partner (client) user ID should be used to connect to the outbound GIS or Connect:Direct server.

Set Up a Password Policy

You have the ability to identify security requirements for a group of users and then configure a password policy to define the security requirements. After you define a password policy, you can apply it to users who configure the SSP environment or to users who connect to the SSP engine and send files to a secure server.

Refer to Chapter 6, *Manage User Accounts and Passwords*, for instructions on defining a password policy and associating it with a user.

Set Up User Accounts to Configure the SSP Environment

You must create user accounts for users who will access the SSP Configuration Manager tool to configure the SSP environment. You can create operator users who have read-only access or define administrator users who have full access to Configuration Manager.

Refer to Chapter 6, *Manage User Accounts and Passwords*, for instructions on defining Configuration Manager users.

Set Up Users for Inbound Connections

Depending upon your configuration, you may need to create a user account for a trading partner who plans to connect to the SSP engine to transfer files to a GIS or Connect:Direct server to authenticate the user ID and password in the SSP local user store.

Refer to Chapter 6, *Manage User Accounts and Passwords*, for instructions on defining inbound users.

Set Up GIS/Connect:Direct Servers (Outbound Node Servers) in the Trusted Zone

You set up servers in the trusted zone by performing the following tasks (Refer to GIS and Connect:Direct documentation for details):

✦ Define users in GIS for HTTP, FTP, and SFTP environments

✦ Define users at the Connect:Direct server for a Connect:Direct environment

✦ Make necessary configuration updates to the GIS adapters

✦ Make necessary changes to the Connect:Direct netmaps

✦ Test the connection between SSP and the GIS or Connect:Direct server

Determine Security Requirements for Communications Sessions Between CM and the Engine

When you install CM and an engine, a secure communications channel is required to communicate. By default, the SSL communication is configured using a single key for both the engine and the system where the web server and CM are installed.

To secure the communication between these components, replace the factory certificates. Refer to Chapter 17, *Manage Certificates Between SSP Components*, for instructions.

Configure a Sterling External Authentication Server

An advanced method of user and certificate authentication is provided through an optional Sterling Commerce product called Sterling External Authentication Server. If you plan to use this tool to authenticate users or certificates, you must configure an EA server.

Refer to Chapter 12, *Configure SSP for Sterling External Authentication Server (EA)*, for instructions on configuring an EA server.

Configure a Remote Perimeter Server

A local perimeter server (internal) is installed with SSP and will be used to manage communications. You can install a remote perimeter server if you want an additional perimeter server.

Refer to Chapter 14, *Configure Perimeter Servers to Manage SSP Communications*, for sample implementations of the remote perimeter server and for instructions on configuring a remote perimeter server. Refer to the *Sterling Secure Proxy Installation Guide* for instructions on installing a remote perimeter server.

Manage Certificates for SSL/TLS Transactions with Trading Partners

This section describes how to use certificates when implementing HTTP/S, FTP/S, or Connect:Direct Secure+ Option communications between SSP and your trading partner and target servers.

About Certificates

Certificates are used in secure communications to encrypt and decrypt data. You create certificates using certificate creation software such as Sterling Commerce Certificate Wizard. Each certificate is made up of two components: the public key and the private key. Always keep your private key secret.

As an added measure of security, you can obtain your certificate from a certificate authority (CA). A CA verifies all of the identity information in your certificate, then adds its signature. In an SSL or TLS transaction, your certificate is presented to your trading partner, who can recognize the signature of the CA using the CA root certificate. This assures your trading partner that you are who you say you are. There are many free and commercial certificate authorities. Some companies use an internal certificate authority.

If you use a certificate that is not validated by a CA, it is called a self-signed certificate. Self-signed certificates are used when identity verification is not required, such as internal communications or product testing.

To implement SSL or TLS over FTP or HTTP when using a CA, you need to acquire the CA root certificate from the trading partner, and you must make it available to SSP. You must also make your private key and certificate available to SSP.

To implement SSL or TLS over FTP or HTTP using self-signed certificates, provide your certificates to your trading partner. Also, acquire your trading partner certificates and make them available to SSP. You also make the private key available to SSP.

Public certificates and CA root certificates must be in base 64 or DER format. Private keys, accompanied by their matching public certificates, must be contained in a base 64 key certificate or a PKCS12 file.

Certificate Implementation Models Using SSP

The following   present several models for using certificates and shows how to implement the model in SSP.

Implement Certificates that Use a Common Certificate Authority

In this scenario, SSP, the target server, and the trading partner use the same CA. The certificate distribution looks like this:
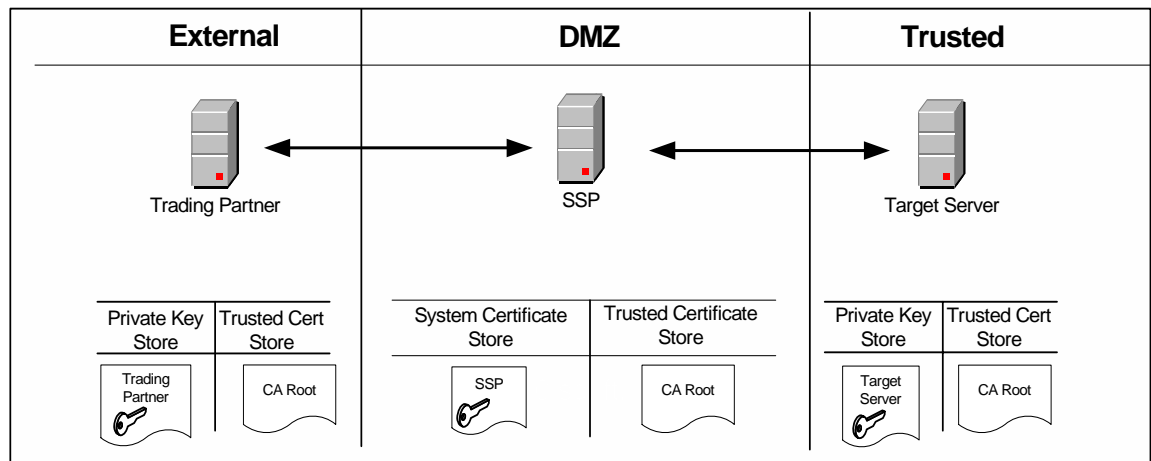
SSP has its private key and the root certificate from the CA. The trading partner has its private key and the root certificate from the CA. The target server has its private key and the root certificate from the CA.

Use the following procedure to implement this model in SSP:

1. Acquire the root certificate from the common CA.

2. Import the CA root certificate into the default store. Refer to *Import a Public Certificate into a Trusted Certificate Store* on page 47.

3. Import the SSP private key into the default system certificate store. Refer to *Import Private Keys into a System Certificate Store* on page 47.

Implement Self-Signed Certificates

In this scenario, there are no CA certificates. Self-signed certificates are used by all entities. The certificate distribution looks like this:
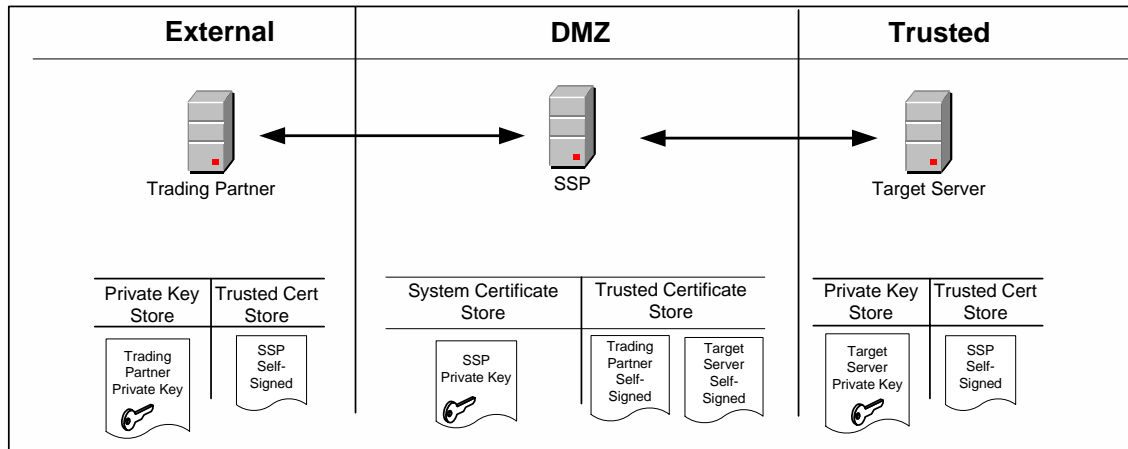
SSP has its private key and the self-signed certificates from the trading partner and the target server. The trading partner has its private key and the self-signed certificate of SSP. The target server has its private key and the self-signed certificate of SSP.

Use the following procedure to implement this model:

1. Provide the SSP self-signed certificate to your trading partner and your target server.

2. Acquire the self-signed certificates from the trading partner and target server.

3. Import the trading partner and target server self-signed certificates into the default Store. Refer to *Import a Public Certificate into a Trusted Certificate Store* on page 47.

4. Import the SSP private key into the default System Certificate Store. Refer to *Import Private Keys into a System Certificate Store* on page 47.

Implement Self-Signed Certificates with Different Certificates for Inbound and Outbound Connections

In this scenario, there are no CA certificates. Separate self-signed certificates are used for the inbound and outbound connections. The certificate distribution looks like this:
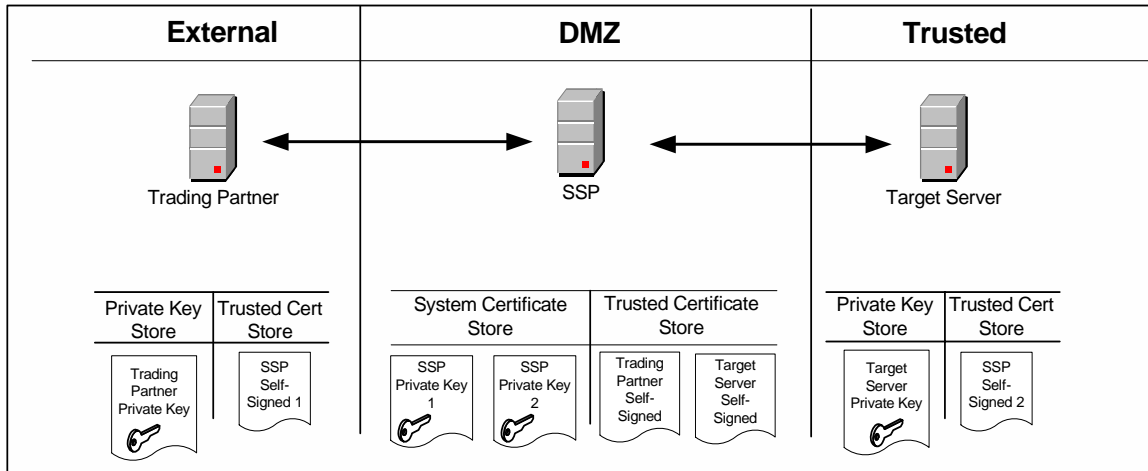
SSP has two private keys and the self-signed certificates from the trading partner and the target server. The trading partner has its private key and one self-signed certificate from SSP. The target server has its private key and the other self-signed certificate from SSP.

Use the following procedure to implement this model:

1. Provide the SSP self-signed certificate to your trading partner and your target server.

2. Acquire the self-signed certificates from the trading partner and target server.

3. Import the trading partner and target server self-signed certificates into the default Trusted Certificate Store. Refer to *Import a Public Certificate into a Trusted Certificate Store* on page 47.

4. Import the SSP private keys into the default System Certificate Store. Refer to *Import Private Keys into a System Certificate Store* on page 47.

Configure a Secure Connection to Sterling External Authentication Server (EA)

You can configure a secure connection between SSP and Sterling External Authentication Server (EA) as shown in the following diagram:
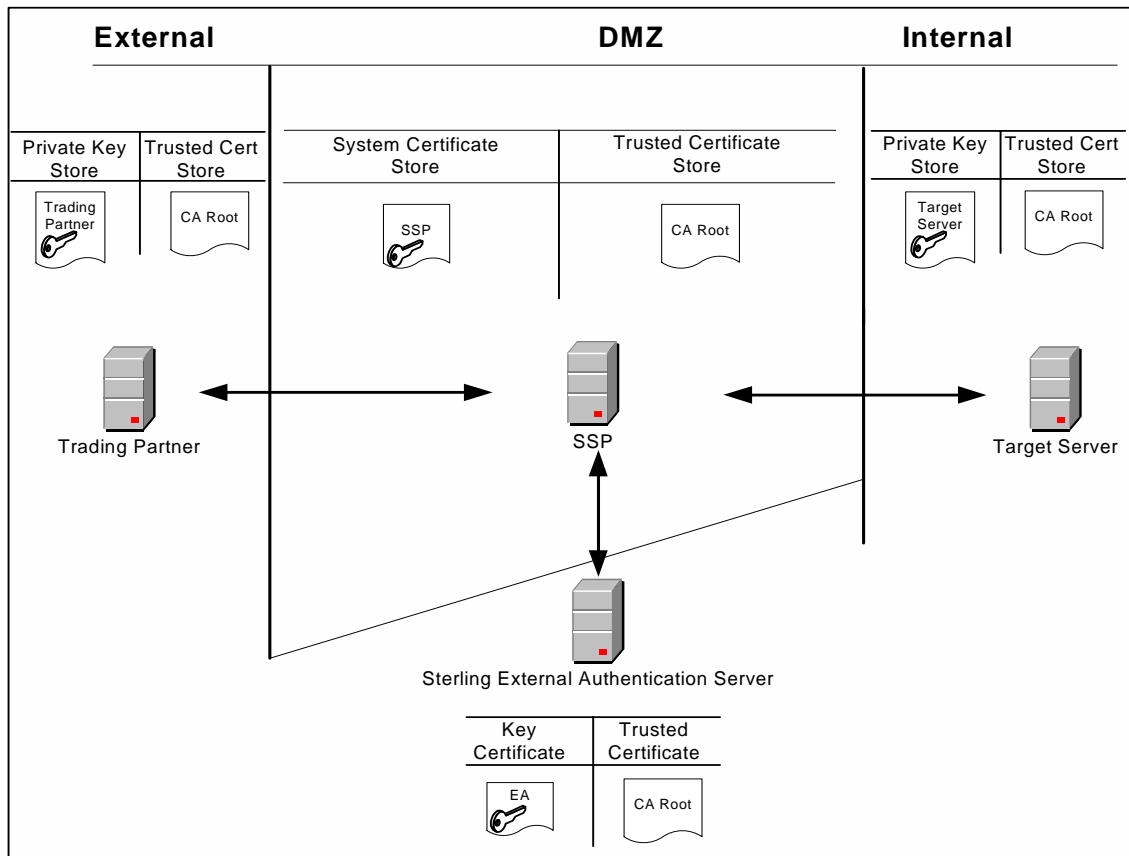
In this scenario, SSP has the private key in the system certificate store and the CA root certificate in the trusted certificate store. The trading partner has a private key and the CA root certificate. The target server has a private key and the CA root certificate. EA has a private key in its own key certificate store and the CA root certificate. Use the following procedure to implement this model.

This example shows the EA implementation with a single certificate. You can also use a multiple SSP certificates model.

Use the following procedure to implement this model:

1. Acquire the root certificate from the common CA.

2. Import the CA root certificate into the default Trusted Certificate Store. Refer to *Import a Public Certificate into a Trusted Certificate Store* on page 47.

3. Import the SSP private key into the default System Certificate Store. Refer to *Import Private Keys into a System Certificate Store* on page 47.

4. Configure certificates for use by EA. Refer to the Sterling External Authentication Server documentation.

Use Multiple Key Stores in SSP

SSP gives you the option of having multiple key stores or trust stores. This is useful if you do not want all your keys in a single location. Also, if you are running multiple SSP engines, it may be better to have a separate system certificate store or trusted certificate store for each engine. The following diagram shows a very basic model using multiple key stores:
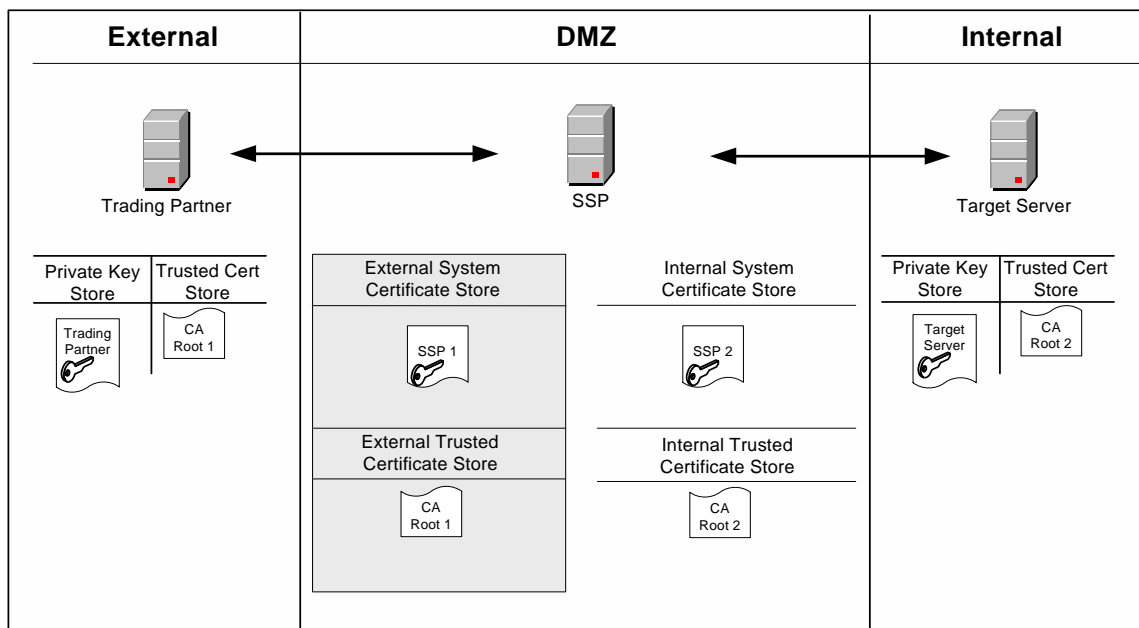
In this scenario, SSP has two key certificates: SSP1 in the external system certificate store and SSP2 in the internal system certificate store. Different CAs are used for internal and external communications. The CA root certificate for external communication (CA Root 1) is in the external trusted certificate store. The CA root certificate for internal communication (CA Root 2) is in the internal trusted certificate store. The trading partner has its own private key and the CA Root 1 certificate. The target server has its own private key and the CA Root 2 certificate. Use the following procedure to implement this model:

1. Acquire the external CA root certificate.

2. Acquire the internal CA root certificate.

3. Create a new trusted certificate store for your external communications (External Store in diagram above). Refer to *Create a New Trusted Certificate Store* on page 48.

4. Import the external CA root certificate into the External Store. Refer to *Import a Public Certificate into a Trusted Certificate Store* on page 47.

5. Create a new trusted certificate store for your internal communications (Internal Store in diagram above). Refer to *Create a New Trusted Certificate Store* on page 48.

6. Import the internal CA root certificate into the internal trusted certificate store. Refer to *Import a Public Certificate into a Trusted Certificate Store* on page 47.

7. Create a new system certificate store for external communications (External System Certificate Store in diagram above). Refer to *Create a New System Certificate Store* on page 48.

8. Import the SSP1 private key into the new external system certificate store. Refer to *Import Private Keys into a System Certificate Store* on page 47.

9. Create a new system certificate store for internal communications (Internal System Certificate Store in diagram above). Refer to *Create a New System Certificate Store* on page 48.

10. Import the SSP2 private key into the new internal system certificate store. Refer to *Import Private Keys into a System Certificate Store* on page 47.

Import a Public Certificate into a Trusted Certificate Store

Use the following procedure to import the public certificate from your trading partner, target server, or CA into a trusted certificate store:

1. Click Credentials from the menu bar.

2. Expand the Certificate Stores tree and then the Trusted Certificates Stores tree.

3. Select a trust store. The default trust store is dfltTrustStore.

4. Click New.

5. Specify a name in the Trusted Certificate Name field.

6. Click Browse to select the certificate to import.

7. Double-click the certificate to select.

8. Click OK.

Import Private Keys into a System Certificate Store

Use the following procedure to import an SSP private key into a system certificate store:

1. Click Credentials from the menu bar.

2. Expand the Certificate Stores tree and then the System Certificate Stores tree.

3. Select a key store. The default key store is dfltKeyStore.

4. Click New.

5. Specify values for the following:

   ◆ System Certificate Name

   ◆ Password (passphrase associated with the system certificate)

   ◆ Confirm Password

6. Click Browse and select the certificate to import.

7. Click OK.

Create a New Trusted Certificate Store

Use the following procedure to create a new trusted certificate store:

1. Click Credentials from the menu bar.

2. Click Actions > New Certificate Store > Trusted Certificate Store.

3. Specify a name for the certificate store in the Trusted Certificate Store Name field.

4. Click Save.

Refer to *Import a Public Certificate into a Trusted Certificate Store* on page 47 to add certificates.

Create a New System Certificate Store

To create a new system certificate store:

1. Click Credentials from the menu bar.

2. Click Actions > New Certificate Store > System Certificate Store.

3. Specify a name for the certificate store in the System Certificate Store Name field.

4. Click Save.

Refer to *Import Private Keys into a System Certificate Store* on page 47 to add certificates to the certificate store.

Store System Certificates on a Hardware Security Module (HSM)

A Hardware Security Module (HSM) is a hardware-based security device that generates, stores, and protects cryptographic keys. SSP uses keys and certificates stored in its store or on an HSM. SSP maintains information in its store about all keys and certificates.

To access keys in an HSM device, a reference to the keys and the passphrase protecting the key must be added to SSP. This reference is secure and cannot be used by an intruder to access the certificate information. You can configure keys on the HSM at CM, using command line scripts described in this chapter.

For more security, create the keys on the HSM device and store the HSM private keys on the device. To import externally-created keys into the HSM, first import the external keys into the HSM and then destroy the files containing the external private key.

HSMs implement the Java JCE API. This interface accesses the keys in the device. The JCE implementations for Eracom and nCipher have the following differences:

✦ Eracom uses slots, logical entities defined through the Eracom administration utility. Designate a slot for SSP and assign a user PIN. Configure SSP and identify the slot to use. Only one slot can be used by SSP.

✦ Eracom uses a single keystore for all keys in a slot. The user PIN protects all the keys in the slot. Each key within a slot must have a unique alias.

✦ nCipher uses a security world that contains one or more HSM modules. The modules can reside on the same or different machines. The keys in the security world are protected by an operator smart card. Create an operator smart card set for SSP, identify "1 of N" for the cards, and assign a passphrase to each card. Before SSP can start, insert the operator smart card protecting the SSP keys into the card reader.

✦ nCipher supports multiple keystores. Each keystore can contain multiple keys, but SSP only stores one key per keystore. With nCipher, multiple keys can have the same alias. For example, on GIS, all keys on an nCipher HSM have the alias Key. Each keystore has a unique instance ID defined as a 40-character hexadecimal string. The combination of the instance ID and the key alias makes each key unique.

Enable and Disable the HSM Environment

Use the setupHSM command to enable or disable the HSM environment. Run this command on the engine. If you are using a netHSM module and CM has access to the netHSM, you can also run the command on CM. Running the command on CM allows you to configure the HSM keys without requiring a running engine. However, you must stop CM.

Stop the engine or CM before you run this command. Additionally, you must have permission to write files to the SSP installation directory. If you reinstall the HSM support software, run the setupHSM -enable command again to make sure that any updated jar files and libraries are copied to the installation directory.

Enable the HSM Environment

Use the setupHSM -enable command to copy files from the HSM hardware to SSP, copy the HSM security providers in the right order, update the security.properties file with the appropriate Certicom TLS security string for the HSM you are using, and add any environment variables to the startup scripts.

To setup the HSM environment for Windows, type the following command:

```
setupHSM -enable [parameters]
```

To setup the HSM environment for UNIX or Linux, type the following command:

```
setupHSM.sh -enable [parameters]
```

Following is a description of the enable parameters:

Parameter	Description
hsm	HSM type. Required if you are enabling the HSM. Valid values = nCipher | Eracom.
slot	Slot number assigned to SSP. The optional parameter is valid for Eracom only. Default=0.
path	Path to the root directory of the HSM runtime support software. Required.
	If the path contains embedded spaces, enclose the whole parameter in double-quotes. For example, "path=C:\Program Files\Eracom".
	On UNIX, the value is normally /opt/nfast for nCipher and /opt/Eracom for Eracom.
	On Windows, the value is normally C:\nfast for nCipher and C:\Program Files\Eracom for Eracom.
netserver	Host name or IP address of the netHSM server. Optional.
	Valid for Eracom on UNIX. It is ignored on Windows.
systempass	Engine system passphrase or CM passphrase, depending upon where the command is run. Optional. If you do not configure this parameter, the user is prompted for the passphrase.

Following is a sample script to setup an Eracom HSM on UNIX:

```
setupHSM.sh -enable hsm=eracom slot=1 path=/opt/Eracom
```

Following is a sample script to setup an nCipher HSM on UNIX:

```
setupHSM.sh -enable hsm=nCipher path=/opt/nfast
```

Disable the HSM Environment

Use setupHSM -disable to delete the HSM provider files, remove the HSM security providers, restore the default Certicom TLS security provider definitions, and remove the HSM environment variables from the startup scripts.

To disable the HSM environment, type the following command:

```
setupHSM -disable systempass
```

Following is a description of the HSM setup disable parameter:

Parameter	Description
systempass	Engine system passphrase or CM passphrase, depending upon where the command is run. Optional. If you do not define this parameter, you are prompted for the passphrase.

Manage Key Certificates

Use the manageKeyCerts command to manage key certificates in the SSP system certificate store and on the HSM. Use this command to perform the following tasks:

✦ Create Self-Signed Certificates
✦ Import a Certificate
✦ Export a Certificate
✦ Obtain a Certificate from the HSM Device
✦ Store a Certificate on the HSM Device
✦ Copy a Certificate
✦ Move a Certificate from One SSP System Certificate Store To Another Store
✦ Rename a Certificate on the SSP System Certificate Store
✦ Delete a Certificate
✦ List Key Certificates on the SSP System Certificate Store

✦ List Key Certificates on the HSM Device

✦ Load References to Keys on the HSM into the SSP System Certificate Store

✦ Update the HSM Password for HSM Key Certificates Stored in the SSP System Store

Create Self-Signed Certificates

Use the manageKeyCerts -create command to create a self-signed key certificate. Stop CM before you run this command.

Consider the following before you use this command:

✦ If the engine parameter is defined, a certificate is created on the HSM configured for that engine. If a netHSM is used and multiple engines access the netHSM, any of the engines can be specified to create the certificate on the HSM.

✦ If the engine uses a PCI module and it cannot be accessed by other engines, group the key certificates for that engine in a separate system certificate store. Those key certificates cannot be shared with other engines.

✦ If the engine parameter is not defined, and HSM support is enabled on CM, the key certificate is created on the HSM configured for CM. Make sure that engines that use this key certificate can access the HSM enabled for CM.

✦ If the engine parameter is not defined and HSM support is not enabled on CM, the key certificate is created on the SSP system certificate keystore.

To create a self-signed key certificate, type the following command:

```
manageKeyCerts -create [parameters]
```

Following are the parameters used to create a key certificate:

Parameter	Description
certName	Name of the key certificate on SSP. Required.
certStore	Name of the system certificate store where the key certificate will be stored. This field is optional. If the store does not exist, it is created. Default=dfltKeyStore.
engine	Name of the engine with access to the HSM. Optional.
alias	Alias for the key certificate on the HSM. Optional. If no value is defined, the alias defaults to certificate name.
keySize	Key size of the file to create. Valid values = 1024 | 2048 | 4096. Default=1024.
CN	Certificate common name. Required. If the name contains spaces, enclose the command and string in double quotes, for example "CN=my name".
email	E-mail address. Optional.
O	Organization. Optional. If the value contains spaces, enclose the command in double quotes, for example, "O=my org".

Parameter	Description
OU	Organization unit. Optional. Repeat this parameter to specify more than one organization unit. If the value contains spaces, enclose the command in double quotes, for example, "OU=my unit".
L	Location (city). Optional. If the value contains spaces, enclose the command in double quotes, for example, "L=my location".
ST	State. Optional. If the value contains spaces, enclose the command in double quotes, for example, "ST=my state".
C	Two letter country code. Optional.
daysValid	How many days the key certificate is valid. Optional. Default=365.
serial	Serial number for the key certificate. Optional. Default=1.
certSignBit	Whether to set the certificate signing bit on in the key usage flags. Valid values = n | y| false | true. Default=n.
replace	Whether to replace a key certificate if a certificate with the same name already exists in the SSP system certificate store. Optional. Valid values = false | true. Default=false.
systempass	Passphrase for CM.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.
keystorepass	Keystore password. Optional. Prompts if not defined. For Eracom, the user PIN for the slot used by SSP. For nCipher, the passphrase for the operator smart card that will be used to protect the key. The card must be inserted in the module's card reader.
keypass	Passphrase for the key in the keystore. Optional. Prompts if not defined. For Eracom, this parameter can be anything and will be ignored. For nCipher, this must be the same value as the keystore password.

Import a Certificate

Use the manageKeyCerts -import command to import a certificate into the SSP system certificate store and the HSM. Stop CM before you run this command.

Consider the following before you use it:

✦ If you define the engine parameter, the certificate is imported to the HSM configured for that engine. If a netHSM is used and multiple engines access the netHSM, any of the engines can be specified to handle the request. Configure HSM support on the engine.

✦ If the engine uses a PCI module and that module cannot be accessed by other engines, group the key certificates for that engine in a separate system certificate store. You cannot share the key certificates on that system certificate store with other engines.

✦ If you do not define the engine parameter, and HSM support is enabled on CM, the key
certificate is imported on the HSM configured for CM. Make sure that engines that use this
key certificate can access the HSM enabled for CM.

✦ If you do not define the engine parameter and HSM support is not enabled on CM, the key
certificate is imported to the SSP system certificate store only.

To import a key certificate into the SSP system certificate store, type the following command:

```
manageKeyCerts -import [parameters]
```

Following is a description of the import parameters:

Parameter	Description
certName	Name of the key certificate on SSP. Required.
certStore	Name of the system certificate store where the key certificate will be stored. This field is optional. If the store does not exist, it is created. Default=dfltKeyStore.
engine	Name of the engine with access to the HSM. Optional.
alias	Alias for the key certificate on the HSM. Optional. If no value is defined, the alias defaults to certificate name.
file	Fully-qualified path of the key certificate file to import. Required. The file must be PEM or PKCS12. The script looks for BEGIN/END PEM markers in the file. If they are not found, the file is assumed to be PKCS12 format.
replace	Whether to replace a system certificate if a certificate with the same name already exists in the SSP system certificate store. Optional. Valid values = n | y | false | true. Default=n.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.
pemkeypass	Password for the PEM private key, if the file is PEM. Optional. Prompts if not defined.
pkcs12storepass	Password of the PKCS12 file, if the file is not PEM. Optional. Prompts if not defined.
pkcs12keypass	Passphrase for the key in the PKCS12 file, if the import file is not PEM. Optional. Prompts if not defined.
keystorepass	Keystore password. Optional. Prompts if not defined. For Eracom, the user PIN for the slot used by SSP. For nCipher, the passphrase for the operator smart card that will be used to protect the key. The card must be inserted in the module's card reader.
keypass	Passphrase for the key on the keystore. Optional. Prompts if not defined. For Eracom, this parameter is not used. For nCipher, define this parameter using the same value as the keystore password.

Export a Certificate

Use the manageKeyCerts -export command to export a certificate from the system store or the HSM. CM can be running when you run this command.

Consider the following before you use this command:

✦ If you specify the engine parameter and the certificate is stored on the HSM, the certificate is exported from the HSM configured at the engine. You must enable the HSM on the engine. If a netHSM is used and multiple engines can access it, any of the engines can be specified to export the certificate.

✦ If you do not specify the engine parameter and the key certificate is stored in an HSM, the certificate is exported from the HSM configured for CM. You must enable the HSM on CM to export a certificate from it.

✦ If the certificate is not stored on an HSM, the engine parameter is ignored and the certificate is exported from the SSP system certificate store.

✦ For key certificates stored on the HSM, only the public certificate in PEM format will be exported. The private key cannot be exported.

To export a key certificate from the SSP system certificate store, type the following command:

```
manageKeyCerts -export [parameters]
```

Following is a description of the export parameters:

Parameter	Description
certName	Name of the key certificate on SSP. Required.
certStore	Name of the system certificate store where the key certificate will be stored. This field is optional. Default=dfltKeyStore.
engine	Name of the engine with access to the HSM. Optional.
format	Format for the key certificate file. This parameter is required for non-HSM key certificates. Forced to pem for the HSM key certificates. Valid values = pem | pkcs12.
file	Fully-qualified path of the file where the key certificate file will be stored. Required.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.
pkcs12storepass	Password of the PKCS12 file, if the format is PKCS12 and the key certificate is not stored on an HSM. Optional. Prompts if not defined.
pkcs12keypass	Passphrase for the key in the PKCS12 file, if the format is PKCS12 and the key certificate is not stored on an HSM. Optional. Prompts if not defined.
pemkeypass	Passphrase to encrypt the private key if the format is PEM.

Obtain a Certificate from the HSM Device

Use the manageKeyCerts -getFromHSM command to extract a reference to a key in the HSM and add the entry into the SSP keystore. Stop CM before you run this command.

Consider the following before you use this command:

✦ If you define the engine parameter, certificate information is obtained from the HSM at the engine. You must enable the HSM on the engine.

✦ If you configure netHSM, and multiple engines access the netHSM, any of the engines can be specified in the command.

✦ If you do not specify the engine parameter, the key certificate is obtained from the HSM configured at CM. You must enable the HSM on CM to obtain information from the HSM at CM.

✦ For the nCipher HSM, the keystore blob for the key (Key Instance, as displayed by KeySafe) must be provided in the keyStoreData parameter. Obtain this 40-character hexadecimal string by running the -listHSM command.

✦ After a reference to an HSM key certificate is successfully obtained, the HSM key cannot be obtained again under a different SSP system certificate name. This action results in an error.

To obtain a key certificate from the HSM, type the following command:

```
manageKeyCerts -getFromHsm [parameters]
```

Following is a description of the getFromHSM parameters:

Parameter	Description
certStore	Name of the system certificate store where the key certificate will be stored. This field is optional. If the store does not exist, it is created. Default=dfltKeyStore.
engine	Name of the engine with access to the HSM. Optional.
certName	Name of the key certificate on SSP. Required.
alias	Alias for the key certificate on the HSM. Required.
keyStoreData	HSM keystore blob string. Required with the nCipher HSM. This is a 40-character hex string, displayed as Key Instance by the nCipher KeySafe utility. If not provided and the HSM key certificate already exists in the system certificate store, the current keystore blob is used to pull the key back into the CM store. Use the -listHsm command to get the blobs for key certificates in the HSM. Alternatively, the blob string can be written to a file. Specify that file name in the keyStoreFile parameter.
keyStoreFile	File containing the HSM keystore blob string. If defined, this parameter overrides the keyStoreData parameter.
replace	Whether to replace a key certificate if a certificate with the same name already exists in the SSP system certificate store. Optional. Valid values = n	y	false	true. Default=n.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.

Parameter	Description
adminpass	Administrator password. Optional. Prompts if not defined.
keystorepass	Keystore password. Optional. Prompts if not defined.
	For Eracom, use the user PIN for the slot used by SSP.
	For nCipher, the passphrase for the operator smart card that will be used to protect the key. The card must be inserted in the module's card reader.
keypass	Passphrase for the key on the keystore. Optional. Prompts if not defined.
	For Eracom, this parameter is not used.
	For nCipher, define this parameter using the same value as the keystore password.

Store a Certificate on the HSM Device

If you have an existing certificate in the SSP certificate store, use the manageKeyCerts -storeOnHsm command to store the key certificate in the HSM. Stop CM before you use this command.

Consider the following before you use this command:

✦ If you define the engine parameter, the certificate is stored at the HSM for the engine. You must enable HSM on the engine.

✦ If you configure a netHSM and multiple engines access the netHSM, any of the engines can be specified to run the request.

✦ If the engine uses a PCI module and the module cannot be accessed by other engines, group the key certificates for the engine into a separate system certificate store. You cannot share the key certificates on that system certificate store with other engines.

✦ If you do not specify the engine parameter, and HSM support is enabled on CM, the key certificate is stored on the HSM configured at CM.

✦ If the key certificate is already stored in an HSM, the command fails.

✦ After a key certificate is stored in an HSM, the key certificate record at CM is updated with a reference to the key in the HSM. If it has a PEM private key, the private key is deleted from the certificate store.

To store a key certificate on the HSM, type the following command:

```
manageKeyCerts -storeOnHsm [parameters]
```

Refer to the following table for a description of the storeOnHsm parameters:

Parameter	Description
certName	Name of the key certificate on SSP. Required.
certStore	Name of the system certificate store where the key certificate is stored. This field is optional. Default=dfltKeyStore.
engine	Name of the engine with access to the HSM. Optional.

Parameter	Description
alias	Alias for the key certificate on the HSM. Optional.
	If no value is defined, the alias defaults to certificate name.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.
keystorepass	Keystore password. Optional. Prompts if not defined.
	For Eracom, the user PIN for the slot used by SSP.
	For nCipher, the passphrase for the operator smart card that will be used to protect the key. The card must be inserted in the module's card reader.
keypass	Passphrase for the key on the keystore. Optional. Prompts if not defined.
	For Eracom, this parameter is not used.
	For nCipher, define this parameter using the same value as the keystore password.

Copy a Certificate

Use the manageKeyCerts -copy command to copy an existing key certificate and assign it a new name. Stop CM before you run this command.

Consider the following before you use this command:

✦ If you specify the engine parameter and the key certificate is stored in an HSM, the engine makes a copy of the key certificate on the HSM, using the new alias provided. HSM support must be enabled at the engine to run this command.

✦ If you configure netHSM and multiple engines access it, specify any of the engines to run the request.

✦ If you do not specify the engine parameter and the key certificate is stored in an HSM, the command makes a copy of the certificate on the HSM configured at CM, using the new alias provided. You must configure the HSM at CM to use this command.

✦ If the key certificate is not stored in an HSM, the engine and new alias parameters are ignored.

To copy a key certificate, type the following command:

```
manageKeyCerts -copy [parameters]
```

Following is a description of the copy parameters:

Parameter	Description
certName	Name of the key certificate on SSP. Required.
certStore	Name of the system certificate store where the key certificate is stored. This field is optional. Default=dfltKeyStore.

Parameter	Description
newName	Name for the copy of the key certificate. Required. Default=certName.
newAlias	Alias for the copy of the key certificate on the HSM. This parameter is required if the key certificate is stored on the HSM.
replace	Whether to replace a key certificate if a certificate with the new name already exists in the SSP system certificate store. Optional. Valid values = n	y	false	true. Default=n.
engine	Name of the engine with access to the HSM. Optional.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.

Move a Certificate from One SSP System Certificate Store To Another Store

Use the manageKeyCerts -move command to move the key certificate from one SSP certificate store to another store.

Stop CM before you run this command. After you run it, all references to the original system certificate are updated in the netmap definitions with references to the new certificate store and certificate name.

To move the key certificate from one SSP certificate store to another, type the following command:

```
manageKeyCerts -move [parameters]
```

Following is a description of the move parameters:

Parameter	Description
certName	Name of the key certificate on SSP. Required.
certStore	Name of the system certificate store on SSP. This field is optional. Default=dfltKeyStore.
destCertStore	Name of the system certificate store on SSP where the key certificate will be moved. Required. If the store does not exist, it is created.
newName	Name for the key certificate on the destination system certificate store. Optional. Default=certName.
replace	Whether to replace a key certificate if a certificate with the same name already exists in the destination SSP system certificate store. Optional. Valid values = n	y	false	true. Default=n.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.

Rename a Certificate on the SSP System Certificate Store

Use the manageKeyCerts -rename command to rename a key certificate in the SSP certificate store. Stop CM before you run this command.

After you run it, all references to the original system certificate are updated in the netmap definitions with references to the new certificate store and certificate name.

To rename the key certificate on the SSP certificate store, type the following command:

```
manageKeyCerts -rename [parameters]
```

Refer to following table for a description of the rename parameters:

Parameter	Description
certName	Name of the key certificate on SSP. Required.
certStore	Name of the system certificate store where the key certificate will be stored. This field is optional. Default=dfltKeyStore.
newName	New name for the key certificate. Required.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.

Delete a Certificate

Use the manageKeyCerts -delete command to delete a key certificate from the SSP keystore or from the HSM. Stop CM before you use this command.

Consider the following before you use this command:

✦ If the system certificate is in use, the command fails. The list of netmap nodes using the system certificate is displayed.

✦ If the key certificate is stored in an HSM, specify the deleteFromHsm parameter to delete the key certificate from the HSM as well.

✦ If the engine parameter is defined, the key certificate is stored in an HSM, and deleteFromHSM is set to yes, the key certificate is deleted from the HSM at the engine. You must configure HSM support at the engine to use this command.

✦ If a netHSM is configured and multiple engines access the netHSM, any of the engines can be specified to run the command.

✦ If the engine parameter is not specified, the key certificate is stored in an HSM, and the deleteFromHSM is set to yes, the command deletes the key certificate from the HSM at CM. HSM support must be enabled at CM to use this command.

✦ If the key certificate is not stored in an HSM, the deleteFromHSM and engine parameters are ignored.

To delete the key certificate from the SSP certificate store, type the following command:

```
manageKeyCerts -delete [parameters]
```

Following is a description of the delete parameters:

Parameter	Description
certName	Name of the key certificate on SSP. Required.
certStore	Name of the system certificate store where the key certificate will be stored. This field is optional. Default=dfltKeyStore.
deleteFromHsm	Determines whether to delete the key certificate from the HSM. This parameter is required if the key certificate is stored on the HSM. Valid values = y	n	true	false.
engine	Name of the engine with access to the HSM. Optional.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.

List Key Certificates on the SSP System Certificate Store

Use the manageKeyCerts -list command to list key certificates on the SSP certificate store. The command can run while CM is running.

To list the key certificate on the SSP certificate store, type the following command:

```
manageKeyCerts -list [parameters]
```

Following is a description of the list parameters:

Parameter	Description
certStore	Name of the system certificate store on SSP. This field is optional. Default=dfltKeyStore. To list certificates in all system certificates stores, define certStore=*.
systempass	CM system passphrase. Optional. Prompts if not specified.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.

List Key Certificates on the HSM Device

Use the manageKeyCerts -listHsm command to list keys on the HSM. This command can be run while CM is running.

Consider the following before you use this command:

✦ For nCipher HSMs, all HSM keys that can be loaded with the provided smart card passphrase are listed, if the keyStoreData parameter is not defined.

✦ If you define the engine parameter, the keys stored on the HSM at the engine are listed. You must configure HSM support at the engine to use this command.

✦ If a netHSM is used and multiple engines access it, any of the engines can be specified to run the request.

✦ If the engine parameter is not defined, the command lists the keys stored on the HSM at CM. HSM support must be enabled at CM.

To list the key certificate on the HSM device, type the following command:

```
manageKeyCerts -listHsm [parameters]
```

Refer to the following table for a description of the list parameters:

Parameter	Description
engine	Name of the engine with access to the HSM. Optional.
keyStoreData	HSM keystore blob string. Used with the nCipher HSM. This is a 40-character hex string, displayed as "Key Instance" by the nCipher KeySafe utility. If it is not provided, all keys that can be loaded with the provided smart card passphrase are listed. Alternatively, the blob string can be written to a file. Specify that file name in the keyStoreFile parameter.
keyStoreFile	File containing HSM keystore data. If defined, this parameter overrides the keyStoreData parameter.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.
keystorepass	Keystore password. Optional. Prompts if not defined.
	For Eracom, the user PIN for the slot used by SSP.
	For nCipher, the passphrase for the operator smart card that will be used to protect the key. The card must be inserted in the module's card reader.

Load References to Keys on the HSM into the SSP System Certificate Store

Use the manageKeyCerts -loadHsm command to load references to keys on the HSM device into the SSP system certificate store. Stop CM before you run this command.

Consider the following before you use this command:

✦ This command is the same as the -getFromHSM command invoked for a list of HSM keys in a properties file. It facilitates HSM key migration from SSP 2.0.02 to SSP 3.1.0, and provides an easy way to populate SSP with HSM keys.

✦ If you define the engine parameter, the keys stored on the HSM at the engine are listed. Enable HSM support at the engine to use this command. If a netHSM is used and multiple engines access the netHSM, any of the engines can be specified to handle the request.

✦ If you do not specify the engine parameter, keys stored on the HSM at CM are listed. HSM support must be enabled at CM to use this command.

✦ After a reference to an HSM key certificate is imported into SSP, that HSM key cannot be referenced again under a different SSP system certificate name.

✦ To override the certificate store for a certificate, use the certStore=<store name> in the input properties file.

To load references to keys on the HSM device into the system certificate store, type the following:

```
manageKeyCerts -loadHsm [parameters]
```

Refer to the following table for a description of the loadHsm parameters:

Parameter	Description
certStore	Name of the system certificate store where the key certificate will be stored. This field is optional. If the store does not exist, it is created. Default=dfltKeyStore.
engine	Name of the engine with access to the HSM. Optional.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.
keystorepass	Keystore password. Optional. Prompts if not defined. For Eracom, the user PIN for the slot used by SSP. For nCipher, the passphrase for the operator smart card that will be used to protect the key. The card must be inserted in the module's card reader.
autoGenName	Whether to auto-generate the name for the key certificate on SSP. Optional. If enabled, and the properties for the key certificate do not specify the certName property, a name is generated using the prefix "hsm_" followed by a hash of the key certificate properties (alias, keystore, type, provider, issuer, subject, serial). Default=n.
replace	Whether to replace a key certificate if a certificate with the same name already exists in the SSP system certificate store. Optional. Valid values = false	true. Default=false.

Parameter	Description
file	Fully-qualified path to the file containing information about the HSM key certificates to load. Required.
	It refers to the output of the SSP 2.0.02 RemoveSystemCert -l script, the -listHSM command, or a text file with lines in the format key=value. To load multiple key certificates, separate the properties for each with a blank line or a line starting with "[".
	All key certificates on the HSM device are listed. The command searches the property file, to find a key certificate on the HSM that matches the specified property. If a match is found, an entry for the matched HSM key certificate is added to the SSP system certificate store with the name specified in the properties, or with an auto-generated name if the parameter called autoGenName=y.
	If the file is the output of the SSP 2.0.02 RemoveSystemCert -l script, the lines on the file are mapped to properties as follows:
	◆ PrivateKeyInfo for ID—alias (for Eracom)
	◆ Name—certName
	◆ KeyStoreType—type
	◆ Issuer—issuer
	◆ Subject—subject
	◆ Serial—serial
	If the file is the output of the version 2.0.02 RemoveSystemCert -l script, remove all lines up to, but not including, the first "PrivateKeyInfo for ID" at the top of the file.
	If the file is the output of the manageKeyCerts -listHSM script, remove all lines up to, but not including, the first "[1]=====================", from the top of the file.

Update the HSM Password for HSM Key Certificates Stored in the SSP System Store

Use the manageKeyCerts -updateHsmPass command after you change the password for the HSM, using the HSM administration utilities. Stop CM before you run this command.

Consider the following before you use this command:

✦ This command does not change the HSM keystore password. It is changed through the HSM administration utilities. You must stop and restart the engine after you change a key store password through the HSM administration utilities.

✦ If you define the engine parameter, this command first tries to load the HSM keys with their current passwords. If a key cannot be loaded, it tries to load the HSM keys with the new password. If the key is successfully loaded, the password for the key is updated on SSP. HSM support must be enabled at the engine to use this command. If netHSM is used and multiple engines access the netHSM, any of the engines can be specified to handle the request.

✦ To update the keystore password on all system certificates, define the certStore=* parameter.

To update the password of the HSM on the SSP system certificate store, type the following command:

```
manageKeyCerts -updateHsmPass [parameters]
```

Following is a description of the updateHsmPass parameters:

Parameter	Description
certStore	Name of the system certificate store where the key certificates are stored. This field is optional. Default=dfltKeyStore.
engine	Name of the engine with access to the HSM. Optional.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.
newKeyStorePass	New HSM keystore password. Optional. For Eracom, the new user PIN for the slot used by SSP. For nCipher, the new passphrase for the operator smart card that will be used to protect the key. The card must be inserted in the module's card reader.

ManageCSRs

Use the manageCSRs command on CM to manage Certificate Signing Requests (CSR). CSRs created with this command cannot be viewed through the CM GUI.

First, create a CSR. The script generates a temporary self-signed key certificate in the HSM or an SSP system certificate store, if the HSM is not enabled. Then send the CSR to a Certification Authority (CA).

When the CA returns the CA-signed certificate, run the manage CSRs command again to replace the self-signed key certificate with the CA-signed certificate. The updated CA-signed certificate is added to the SSP system certificate store, and the CSR status is set to complete.

The key certificate can now be used by SSP.

Use the manageCSRs command to perform the following tasks:

✦ Create a CSR
✦ Update a CSR
✦ Delete a CSR
✦ List CSRs on the CM Store
✦ Retrieve a CSR to Send to a Certification Authority
✦ Retrieve the CA-signed Certificate

Create a CSR

Use the manageCSRs –create command to create a CSR for a key certificate at either the HSM or the SSP system certificate store. You can use this command while CM is running.

Consider the following before you use this command:

✦ If you define the engine parameter, a key certificate is created on the HSM configured for the engine. You must enable HSM support at the engine in order to run this command. If a netHSM is used and multiple engines access the netHSM, any of the engines can be specified to handle the request.

✦ If you do not define the engine parameter and HSM support is not enabled on CM, the system certificate store certificate is created on the SSP system certificate store.

To create a CSR on Windows:

```
manageCSRs –create [parameters]
```

To create a CSR on UNIX or Linux:

```
manageCSRs.sh –create [parameters]
```

Following is a description of the create CSR parameters:

Parameter	Description
csrName	Name for the CSR. Required.
engine	Name of the engine with access to the HSM. Optional.
alias	Alias for the key certificate on the HSM. Optional. If no value is defined, the alias defaults to CSR name.
keySize	Key size of the file to create. Valid values = 1024 | 2048 | 4096 Default=1024.
CN	Certificate common name. Required. If the name contains spaces, enclose the command and string in double quotes, for example, "CN=my name".
O	Organization. Optional. If the value contains spaces, enclose the command and string in double quotes, for example, "O=my org".
OU	Organization unit. Optional. Repeat this parameter to specify more than one organization unit. If the value contains spaces, enclose the command and string in double quotes, for example, "OU=my unit".
L	Location (city). Optional. If the value contains spaces, enclose the command in double quotes, for example, "L=my location".
ST	State. Optional. If the value contains spaces, enclose the command and string in double quotes, for example, "ST=my state".
C	Two letter country code. Optional.
email	E-mail address. Optional.

Parameter	Description
file	Fully-qualified path to the file where the CSR will be stored. If this parameter is not defined, the output of the CSR is displayed on the monitor. To obtain the CSR information later, use the -getpkcs10 command. Optional.
systempass	CM system passphrase. Optional.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.
keystorepass	Keystore password. Optional. Prompts if not defined. For Eracom, the user PIN for the slot used by SSP. For nCipher, the passphrase for the operator smart card, used to protect the key. Be sure that the card is in the card reader before you run this command.
keypass	Passphrase for the key on the keystore. Optional. Prompts if not defined. This value is not used by the Eracom HSM.

Update a CSR

Use the manageCSRs -update command to update a pending CSR with the CA-signed certificate. Stop CM before you run this command.

Consider the following when using this command:

✦ If the key certificate is created in an HSM and you specify the engine parameter, the command notifies the engine to update the key certificate on the HSM. Configure HSM support at the engine to use this command.

✦ If a netHSM is used and multiple engines access it, any of the engines can be specified to perform the update.

✦ If the engine uses a PCI module and that module cannot be accessed by other engines, you must group the key certificates for the engine in a separate system certificate store. You cannot share the key certificates on that system certificate store with other engines.

✦ If the key certificate was created in an HSM and you do not specify the engine parameter, the command updates the key certificate on the HSM at CM. You must enable HSM support at CM.

✦ If the key certificate was not created in an HSM, it is updated on the SSP system certificate store. The engine parameter is ignored.

To update a pending CSR, type the following command:

```
manageCSRs –update [parameters]
```

Following is a description of the update parameters:

Parameter	Description
csrName	Name for the CSR. Required.
engine	Name of the engine with access to the HSM. Optional.
file	Fully-qualified path of the CA-signed certificate file. Required.
certName	Name of the key certificate on SSP. Required.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.
newKeyStorePass	New HSM keystore password. Optional. If defined, this value overrides the keystore password used when the CSR was created. This parameter allows you to update a CSR on the HSM after the keystore password for the HSM is changed.

Delete a CSR

Use the manageCSRs -delete command to delete a CSR from the CM store. This command can be run while CM is running.

Consider the following when using this command:

✦ If the CSR is pending and its key certificate was generated on an HSM, the temporary key certificate is deleted from the HSM.

✦ If the CSR is complete, this command deletes the CSR, but does not delete the key certificate. To delete the key certificate, use the manageKeyCerts -delete command.

To delete a CSR from CM, type the following command:

```
manageCSRs -delete [parameters]
```

Following is a description of the delete parameters:

Parameter	Description
csrName	Name for the CSR. Required.
engine	Name of the engine with access to the HSM. Optional.
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.

Parameter	Description
adminpass	Administrator password. Optional. Prompts if not defined.
newKeyStorePass	New HSM keystore password. Optional.
	If defined, this value overrides the keystore password used when the CSR was created. This parameter allows you to update a CSR on the HSM after the keystore password for the HSM is changed.

List CSRs on the CM Store

Use the manageCSRs -list command to display a list of CSRs on CM. This command can be run while CM is running.

To list the CSRs in the CM store, type the following command:

```
manageCSRs -list [parameters]
```

Refer to the following table for a description of the list parameters:

Parameter	Description
systempass	CM system passphrase.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.

Retrieve a CSR to Send to a Certification Authority

Use the manageCSRs -getpkcs10 command to retrieve a CSR to send to a Certificate Authority (CA). This command can be run while CM is running.

To retrieve a CSR from the HSM that is ready to send to a CA, type the following command:

```
manageCSRs -getpkcs10 [parameters]
```

Refer to the following table for a description of the list parameters:

Parameter	Description
csrName	Name for the CSR. Required.
file	Fully-qualified path of the file where the CSR will be stored.

Parameter	Description
systempass	CM system passphrase. Optional.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.

Retrieve the CA-signed Certificate

Use the manageCSRs -getcacert command to retrieve the CA-signed certificate received from a CA, after the update command has been run. The certificate is returned in PEM format. This command can be run while CM is running.

To retrieve the CA-signed certificate from the HSM, type the following command:

```
manageCSRs -getcacert [parameter]
```

Refer to the following table for a description of the getcacert parameters:

Parameter	Description
csrName	Name for the CSR. Required.
file	Fully-qualified path where the CA-signed certificate will be stored. If not specified, the certificate text is written to the display.
systempass	CM system passphrase. Optional. Prompts if not defined.
adminid	Administrator ID. Optional. Prompts if not defined.
adminpass	Administrator password. Optional. Prompts if not defined.

Chapter 5

Manage SSH Keys for SFTP Transactions

This section describes how to use SSH keys when implementing SFTP communications between SSP and your trading partners and target servers.

About SSH/SFTP

SSH/SFTP provides a more secure means than FTP to exchange information with trading partners. During an FTP session, the user name and password are transmitted in clear text. An eavesdropper can easily log this FTP user name and password. Using SSH/SFTP instead of FTP, the entire login session, including transmission of password, is encrypted, making it much more difficult for an outsider to observe and collect passwords. By encrypting all traffic, SSH/SFTP effectively eliminates eavesdropping, connection hijacking, and other network-level attacks.

You can configure SSP to require authentication with a password and public key for SSH/SFTP connections. Authentication for SSH/SFTP connections is performed by the exchange of session keys between the server and the client. This assures that both parties know whom they are exchanging data with.

To implement authentication for SFTP connections, you must create SSH key stores and import SSH keys into them. These key stores and keys can then be selected when you are configuring SSP to support SSH/SFTP connections. Configure the following SSH keys for SFTP communications:

✦ Inbound connections

  ◆ Local Host Key—Private key used by SSP to identify itself to the client

  ◆ Authorized User Key—Public key used by SSP to authenticate the user (optional)

✦ Outbound connections

  ◆ Known Host Key—Public key used by SSP to authenticate the server

  ◆ Local User Key—Private key used by SSP to identify itself to the server during public key user authentication (optional)

Because public key server authentication is mandatory in SSH, you must configure both local host keys and known host keys. Client authentication is performed using a password or public key (or both) in SSH. As a result, authorized user keys and local user keys are required *only* if you plan to use public key authentication. You can choose different user authentication methods for the inbound and outbound connections.

In Configuration Manager, you must create at least one key within a key store to save the key store definition. You can add as many keys as needed to a key store, and they can be shared between multiple adapters. When you have configured SSH key stores, you can copy them (and the keys within them) to create new key stores.

SSH Key Implementation Models Using SSP

This section presents two models for using SSH keys and shows how to implement the model in SSP.

✦ Use Server Authentication for Inbound and Outbound Connections

✦ Implement Public Key User Authentication for Inbound and Outbound Connections

Use Server Authentication for Inbound and Outbound Connections

In a basic SSH key implementation, you use both local host keys and known host keys for SFTP communications with your trading partner and target server. The key distribution looks like the following:
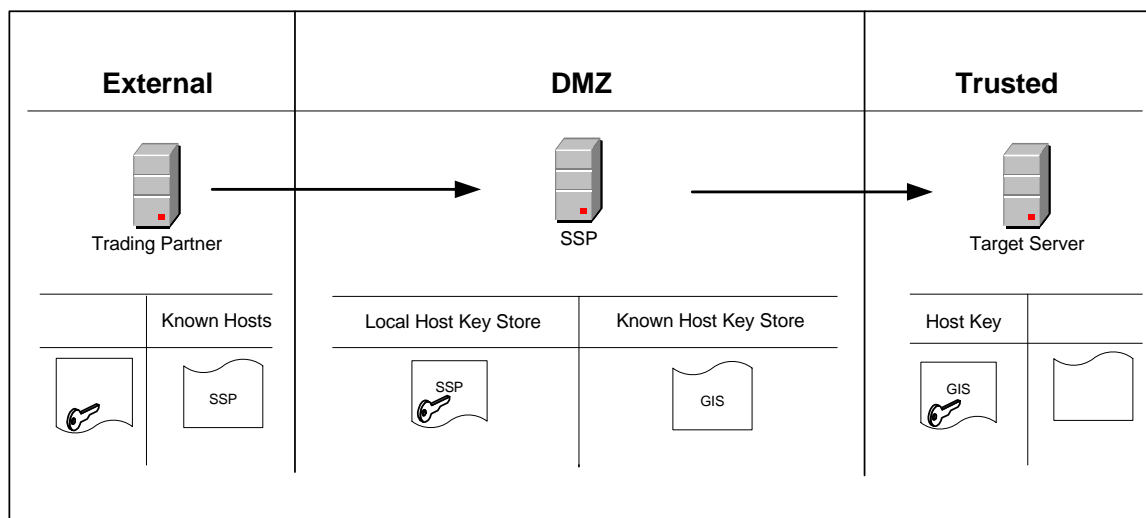
In this scenario, SSP has the private host key in the Local Host Key Store to support the inbound SFTP connection with the trading partner, and the public host key in the Known Host Key Store to support the outbound SFTP connection with the target server.

To implement this model:

1. Provide SSP's public local host key to your trading partner.

2. Acquire the target server's public host key.

3. Create a local host key store and import SSP's private key into the local host key store. Refer to *Manage Local Host Key Stores and Keys* on page 74.

4. Create a known host key store and import the target server's public host key into the known host key store. Refer to *Manage Known Host Key Stores and Keys* on page 79.

5. In the SFTP Adapter configuration on the Basic tab, select the local host key store you created and specify the location and name of the local host key you imported into the local host key store.

6. In the outbound node tab of the SFTP server connection definition in the netmap, select the known host key store you created and specify the location and name of the known host key you imported into the known host key store.

Implement Public Key User Authentication for Inbound and Outbound Connections

You can add user authentication to the basic SSH key implementation by using local user keys and authorized user keys for SFTP communications with your trading partner and target server. The key distribution looks like the following:
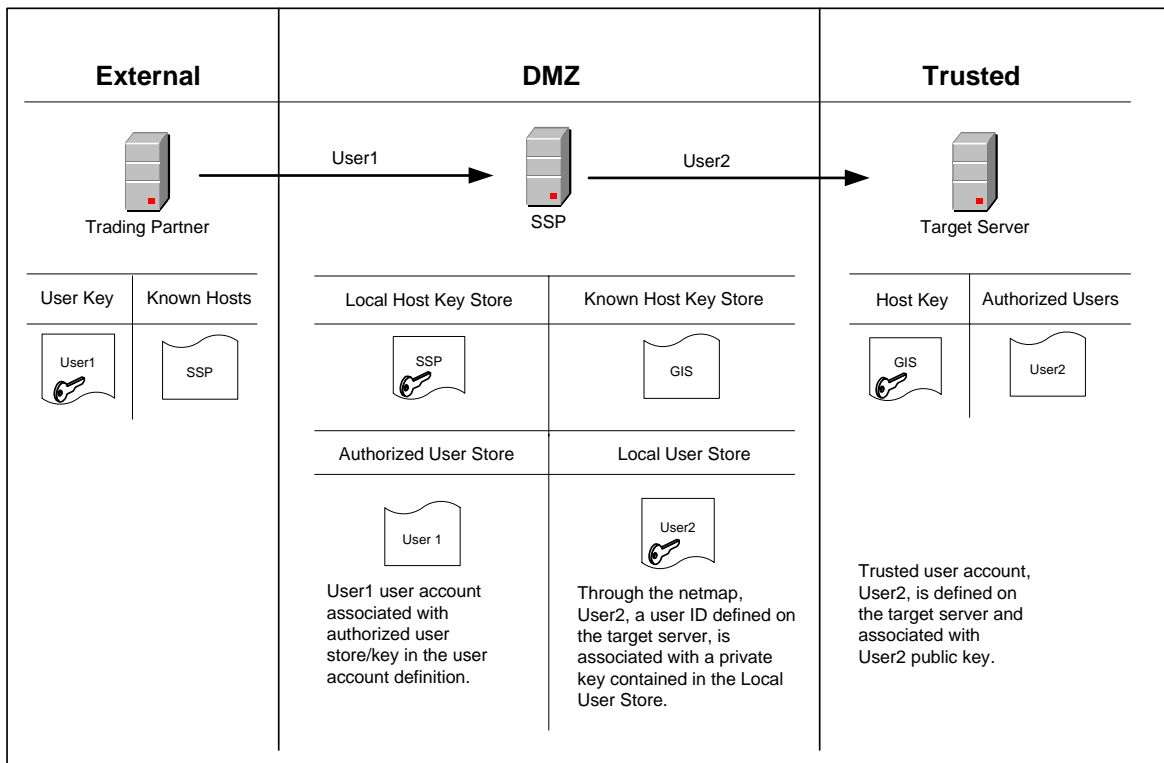
In this scenario, SSP has the mandatory local host key and known host key, and, for client authentication, it also contains an authorized user key for inbound connections to SSP and a local user key for outbound connections to the target server. In addition, the inbound user ID is replaced with a trusted user account defined on the target server.

To implement this model:

1. Complete the steps in *Use Server Authentication for Inbound and Outbound Connections* on page 72 to configure the mandatory keys required for SSH server authentication.

2. Provide the target server with the public local user key for your internal user ID.

3. Acquire the trading partner's public user key.

4. Create a local user key store and import the target server's private key into the local user key store. Refer to *Manage Local User Key Stores and Keys* on page 82.

5. Create an authorized user key store and import the trading partner's public user key into the authorized user key store. Refer to *Manage Authorized User Key Stores and Keys* on page 77.

6. In the SFTP Policy on the Configuration tab, select:

   ◆ Key as the Authentication Method on the Advanced tab. For information on how to configure the other authentication methods: Password, Password and Key, Password or Key, refer to Chapter 10, *SFTP Reverse Proxy Configuration*.

   ◆ Through Local User Store as the User Authentication Mechanism.

   ◆ Internal User ID - Netmap as the User Mapping method.

7. In the SFTP Netmap outbound node definition Advanced tab specify:

   ◆ User ID defined on the target server

   ◆ Local User Key Store and Local User Key for the outbound connection

8. Under credentials in the user account located in the User Stores, on the Advanced tab, select the Authorized User Key Store and select the Authorized User Key you imported into the key store.

Manage Local Host Key Stores and Keys

The local host key store contains the private key used by SSP to identify itself to the client during server authentication in inbound SFTP connections. To use SSH, you must configure a local host key store and import a local host key into the key store. When you are setting up the local host private key, be sure to distribute the matching public key to your trading partners.

Caution: Never distribute the private key to your trading partners.

Create a Local Host Key Store and Import a Key

To create a local host key store and key:

1. Click Credentials from the menu bar.

2. Click Actions > New SSH Key Store > Local Host Key Store.

3. In the Local Host Key Store Configuration window, type a name for the key store in the Local Host Key Store Name field (no spaces allowed).

4. Click New.

5. In the Local Host Key Configuration window, specify the following:

   ◆ Local Host Key Name

   ◆ Password

   ◆ Confirm Password

6. Click Browse and select the private key to import into the key store. The key contents display in the Key data field.

7. Click OK.

8. Click Save.

You can add as many keys as needed to this key store after it is created. You can now select this key store and key when configuring the SFTP Reverse Proxy Adapter.

Edit a Local Host Key

To edit a local host key:

1. Click Credentials from the menu bar.

1. Expand the SSH Key Stores tree and the Local Host Key Stores tree.

2. Click the key store that contains the key you want to edit.

3. Select the key you want to edit and click Edit.

4. To disable the key, click the Enable Key field.

5. Modify the key definition as necessary.

6. Click OK.

7. Click Save.

Copy a Local Host Key

To copy a local host key:

1. Click Credentials from the menu bar.

2. Expand the SSH Key Stores tree and the Local Host Key Stores tree.

3. Click the key store that contains the key you want to copy.

4. Select the key you want to copy and click Copy.

5. In the Local Host Key Configuration window, type a name for the new key.

6. Edit the properties as needed.

7. Click OK.

8. Click Save.

Delete a Local Host Key

To delete a local host key:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree and the Local Host Key Store tree.
3. Click the key store that contains the key you want to delete.
4. In the Local Host Key Store Configuration window, select the key to delete and click Delete.
5. Click Save.

Copy a Local Host Key Store

After you create a local host key store, you can copy it to create a new local host key store.

To copy a local host key store:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree and the Local Host Key Stores tree.
3. Click the key store to copy and select Actions > Copy Selected.
4. Type a name for the key store (no spaces allowed).
5. Click Save.

Edit a Local Host Key Store

To edit a local host key store:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Local Host Key Stores tree.
4. Click the key store you want to edit.
5. In the Local Host Key Configuration window, copy, edit, or delete the keys in the key store as needed.
6. Click Save.

Delete a Local Host Key Store

To delete a local host key store:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Local Host Key Stores tree.
4. Click the key store to delete and select Actions > Delete Selected.
5. Click Delete.

Manage Authorized User Key Stores and Keys

The authorized user key store contains the public key used by SSP to authenticate the user in SFTP connections. This key is required only if you plan to use public key authentication for inbound SFTP connections. Obtain the public key from your trading partner before you configure the authorized user key store so you will be ready to import the key when you configure the key store.

Create an Authorized User Key Store and Import a Key

To create an authorized user key store and key:

1. Click Credentials from the menu bar.
2. Click Actions > New SSH Key Store > Authorized User Key Store.
3. In the Authorized User Key Store Configuration window, type a name for the key store (no spaces allowed).
4. Click New.
5. In the Authorized User Key Configuration window, type a name for the user key.
6. Click Browse and select the public key to import into the key store. The key contents display in the Key Data field.
7. Click OK.
8. Click Save.

You can now select this key store and key on the User Store Advanced tab of Configuration Manager when you are configuring users in the user store.

Edit an Authorized User Key

To edit an authorized user key:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Authorized User Key Stores tree.
4. Click the key store that contains the key you want to edit.
5. In the Authorized User Key Store Configuration window, select the key you want to edit and click Edit.
6. Click Browse to select a different key. The key contents display in the Key Data field.
7. Click OK.
8. Click Save.

Copy an Authorized User Key

To copy an authorized user key:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Authorized User Key Store tree.
4. Click the user key store that contains the key you want to copy.
5. In the Authorized User Key Stores Configuration window, select the key you want to copy and click Copy.
6. In the Authorized User Key Configuration window, type a name for the new key.
7. Edit the properties as needed.
8. Click OK.
9. Click Save.

Delete an Authorized User Key

To delete an authorized user key:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Authorized User Key Stores tree.
4. Click the key store that contains the key you want to delete.
5. In the Authorized User Key Store Configuration window, select the key you want to delete and click Delete.
6. Click Save.

Copy an Authorized User Key Store

After you have created an authorized user key store, you can copy it to create new authorized user key stores. To copy an authorized user key store:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Authorized User Key Stores tree.
4. Click the key store you want to copy and select Actions > Copy Selected.

    A copy of the key store is displayed in the Authorized User Key Store Configuration window.

5. Type a name for the key store (no spaces allowed).
6. Click Save.

Edit an Authorized User Key Store

To edit an authorized user key store:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Authorized User Key Stores tree.
4. Click the key store you want to edit.
5. In the Authorized User Key Configuration window, copy, edit, or delete the keys in the key store as needed.
6. Click Save.

Delete an Authorized User Key Store

To delete an authorized user key store:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Authorized User Key Stores tree.
4. Click the key store you want to delete and select Actions > Delete Selected.
5. Click Delete.

Manage Known Host Key Stores and Keys

The known host key store contains the public key used by SSP to authenticate the server in outbound SFTP connections. To use SSH, you must configure a known host key store and import a known host key into the key store. Obtain the public key from your target server before you configure the known host key store so you will be ready to import the key when you configure the key store.

Create a Known Host Key Store and Import a Key

To create a known host key store and key:

1. Click Credentials from the menu bar.
2. Click Actions > New SSH Key Store > Known Host Key Store.
3. In the Known Host Key Store Configuration window, type a name for the key store (no spaces allowed).
4. Click New.
5. In the Known Host Key Configuration window, type a name for the known host key.
6. Click Browse and select the public key to import into the key store. The key contents display in the Key Data field.

7. Click OK.

8. Click Save.

You can now select this key store and key on the Outbound Node, Basic tab when configuring the SFTP Reverse Proxy Netmap.

Edit a Known Host Key

To edit a known host key:

1. Click Credentials from the menu bar.

2. Expand the SSH Key Stores tree.

3. Expand the Known Host Key Stores tree.

4. Click the key store that contains the key you want to edit.

5. Select the key you want to edit and click Edit.

6. In the Known Host Key Configuration window, enable or disable the key.

7. Click Browse to select a different key. The key contents display in the Key data field.

8. Click OK.

9. Click Save.

Copy a Known Host Key

To copy a known host key:

1. Click Credentials from the menu bar.

2. Expand the SSH Key Stores tree.

3. Expand the Known Host Key Stores tree.

4. Click the key store that contains the key you want to copy.

5. In the Known Host Key Store Configuration window, select the key you want to copy and click Copy.

6. In the Known Host Key Configuration window, type a name for the new key.

7. Edit the properties as needed.

8. Click OK.

9. Click Save.

Delete a Known Host Key

To delete a known host key:

1. Click Credentials from the menu bar.

2. Expand the SSH Key Stores tree.

3. Expand the Known Host Key Stores tree.

4. Click the key store that contains the key you want to delete.

5. In the Known Host Key Store Configuration window, select the key you want to delete and click Delete.

6. Click Save.

Copy a Known Host Key Store

After you have created a known host key store, you can copy it to create a new known host key store.

To copy a known host key store:

1. Click Credentials from the menu bar.

2. Expand the SSH Key Stores tree.

3. Expand the Known Host Key Stores tree.

4. Click the key store you want to copy and select Actions > Copy selected.

   A copy of the key store displays in the Known Host Key Store Configuration window.

5. Type a name for the key store (no spaces allowed).

6. Click Save.

Edit a Known Host Key Store

To edit an known host key store:

1. Click Credentials from the menu bar.

2. Expand the SSH Key Stores tree.

3. Expand the Known Host Key Stores tree.

4. Click the key store you want to edit.

5. In the Known Host Key Configuration window, copy, edit, or delete the keys in the key store as needed.

6. Click Save.

Delete a Known Host Key Store

To delete a known host key store:

1. Click Credentials from the menu bar.

2. Expand the SSH Key Stores tree.

3. Expand the Known Host Key Stores tree.

4. Click the key store you want to delete and select Actions > Delete Selected.

5. Click Delete.

Manage Local User Key Stores and Keys

The local user key store contains the private key used by SSP to identify itself to the server during public key user authentication in outbound SFTP connections. This key is required *only* if you plan to use public key authentication in outbound SFTP connections. When you are setting up the local user key, be sure to distribute the matching public key to your target server.

Caution: Never distribute the private key to your trading partners.

Create a Local User Key Store and Import a Key

To create a local user key store:

1. Click Credentials from the menu bar.

2. Click Actions > New SSH Key Store > Local User Key Store.

3. In the Local User Key Store Configuration window, type a name for the key store (no spaces allowed).

4. Click New.

5. In the Local User Key Configuration window, specify the following:

    ◆ Local User Key Name

    ◆ Password

    ◆ Confirm Password

6. Click Browse and select the private key to import into the key store. The key contents display in the Key data field.

7. Click OK.

8. Click Save.

You can now select this key store and key on the SFTP Netmap, Outbound Node, Advanced tab of Configuration Manager when you are configuring an SFTP Reverse Proxy netmap.

Edit a Local User Key

To edit a local user key:

1. Click Credentials from the menu bar.

2. Expand the SSH Key Stores tree.

3. Expand the Local User Key Store tree.

4. Click the key store that contains the key you want to edit.

5. Select the key you want to edit and click Edit.

6. In the Local User Key Configuration window, enable or disable the key.

7. Click Browse to select a different key. The key contents display in the Key data field.

8. Click OK.

9. Click Save.

Copy a Local User Key

To copy a local user key:

1. Click Credentials from the menu bar.

2. Expand the SSH Key Stores option.

3. Expand the Local User Key Store option.

4. Click the key store that contains the key you want to copy.

5. In the Local User Key Store Configuration window, select the key you want to copy and click Copy.

6. Type a name for the key store (no spaces allowed).

7. Click OK.

8. Click Save.

Delete a Local User Key

To delete a local user key:

1. Click Credentials from the menu bar.

2. Expand the SSH Key Stores tree.

3. Expand the Local User Key Store tree.

4. Select the key store that contains the key you want to delete.

5. In the Local User Key Store Configuration window, select the key you want to delete and click Delete.

6. Click Save.

Copy a Local User Key Store

After you have created a local user key store, you can copy it to create a new local user key store.

To copy a local user key store:

1. Click Credentials from the menu bar.

2. Expand the SSH Key Stores tree.

3. Expand the Local User Key Store tree.

4. Click the key store you want to copy and select Actions > Copy Selected. A copy of the key store is displayed in the Local User Key Store Configuration window.

5. Type a name for the key store (no spaces allowed).

6. Click Save.

Edit a Local User Key Store

To edit a local user key store:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Local User Key Store tree.
4. Click the key store you want to edit.
5. In the Local User Key Configuration window, copy, edit, or delete the keys in the key store as needed.
6. Click Save.

Delete a Local User Key Store

To delete a local user key store:

1. Click Credentials from the menu bar.
2. Expand the SSH Key Stores tree.
3. Expand the Local User Key Store tree.
4. Click the key store you want to delete and select Actions > Delete Selected.
5. Click Delete.

Manage User Accounts and Passwords

Two types of user accounts can be created in SSP: CM user accounts and SSP engine user accounts. CM user accounts control access to the SSP user interface. Engine user accounts control which users can send data through SSP. Password policies can be associated with both a CM user account and engine user account to help enforce your company's security policies. Some of the options in the password policy do not apply to engine users. CM user accounts also include role-based security to provide varying levels of access to users within the organization. SSP can be configured to perform user authentication based on information defined in a user account.

Manage Password Policies

Password policies are sets of security decisions you make and apply to different user accounts according to security policies in your company. These choices include items such as the number of days a password is valid and the maximum and minimum length of a password.

Use password policies to streamline security operations when adding new users. Instead of adding individual policies for each user, you create one password policy and apply it to all users who require the same access.

A password policy is applied to a new user or when the password is changed on an existing user.

You can apply a password policy only to internal user accounts. This provides you the greatest flexibility in maintaining security policies.

For example, a password policy named Test may have the following password settings:

✦ Valid for 10 days

✦ Requires a minimum of 10 characters and maximum of 20 characters

✦ Requires default password change after the initial log in

✦ Maintains three passwords in history so the user cannot reuse them

✦ Must use at least two special characters

In this example, the system administrator gives the user a user name and password. The user logs in to SSP and is prompted to change the password. If the user fails to provide a password with at least 10 and no more than 20 characters, or without at least two special characters, SSP prompts the user for corrections. After all conditions in the password policy are met, the new password is saved and the user is allowed access.

Each user account can have only one password policy associated with it, but one password policy can be applied to multiple user accounts.

Create a Password Policy

You create a password policy to assign to user accounts. You do not have to associate a password policy with a user account, but doing so helps manage your security by streamlining your security operations. A user account can have only one password policy.

To create a password policy:

1. Click Advanced from the menu bar.

2. Click Actions > New Password Policy.

3. Specify values for the following:

    *   Password Policy Name (no spaces allowed)

    *   Days Valid

    *   Minimum Length

    *   Maximum Length

    *   Keep in History

4. To enforce the policy of using at least two special characters in passwords, enable Must contain special characters.

5. Click Save.

You can now edit and delete password policies and assign them to user accounts.

Edit a Password Policy

To edit a password policy:

1. Click Advanced from the menu bar.

2. Expand the Password Policies tree.

3. Click the password policy to edit.

4. Edit the values you want to change. You cannot edit the policy name.

5. Click Save.

Copy a Password Policy

To copy a password policy:

1. Click Advanced from the menu bar.
2. Expand the Password Policies tree.
3. Click the password policy to copy.
4. Click Actions > Copy Selected.
5. Type a name for the new policy.
6. Edit any values you want to change.
7. Click Save.

Delete a Password Policy

> **Note:** If you delete a password policy, users with accounts associated with the password policy can log in, but they are not forced to change the password. If a user does change the password after the password policy is deleted, no validation is completed against the new password.

To delete a password policy:

1. If the password policy is associated with a user:
   a. Click Credentials from the menu bar.
   b. Expand the User Stores tree.
   c. Select the user store that contains the user definition.
   d. Select the user to edit and click Edit.
   e. Remove the password policy to delete from the Password Policy ID field.
   f. Click OK.
   g. Click Save.
2. Click Advanced from the menu bar.
3. Expand the Password Policies tree and click the password policy to delete.
4. Click Actions > Delete Selected.
5. Click Delete.

Manage CM User Accounts

CM accounts are assigned a user role: Admin or Operator. Admin users can create and update user accounts and have full assess to all configuration options in CM. Operator users have read-only access to accounts and cannot access system functions. Operator users can, however, change their passwords from the login screen.

In addition to role-based security, you can assign password policies to user accounts. Use the default CM user account called admin access CM to create user accounts.

Create a CM User Account

To create a CM user account:

1. Click System from the menu bar.
2. Click Actions > New CM User.
3. Specify the following values for the user account:
    *   User Name (no spaces allowed)
    *   Password
    *   Confirm Password
4. Select the user role to assign to the user account from the User role list: Admin or Operator.
5. To enforce a password policy for this account, select a password policy from the list.
6. To require that the user change the password after the first logon, enable Password Requires change.
7. Click Save.

Edit a CM User Account

To edit a CM user account:

1. Click System from the menu bar.
2. Expand the CM Users tree.
3. Select the user account to edit.
4. Edit the user properties as needed. The User Name cannot be edited.
5. Click Save.

Copy a CM User Account

You can copy a CM user account to create a new user account.

To copy an account:

1. Click System from the menu bar.
2. Expand the CM Users tree.
3. Select the user account to copy and click Actions > Copy Selected.
4. Type a name for the account.
5. Edit the user properties as needed.
6. Click OK.
7. Click Save.

Delete a CM User Account

You can delete a CM user account as needed to maintain the security of SSP.

To delete a user account:

1. Click System from the menu bar.

2. Expand the CM Users tree.

3. Select the user account to delete.

4. Click Actions > Delete Selected.

Manage Engine User Stores and User Accounts

Create engine user accounts for users who need to access SSP for file transfer. You can create SSP user accounts in the default user store, defUserStore, or you can create a new user store to manage groups of users.

For users who communicate using the SSH protocol and who use multiple keys to authorize users, identify the key store where keys are stored and the user record containing the key.

Before you begin:

✦ If you plan to use password policies for user accounts, configure the password policies prior to configuring user accounts.

✦ If you plan to perform local user authentication using SSH keys for SFTP inbound connections, import SSH keys into the SSH key stores. For more information on importing keys into the SSH key stores, see Chapter 5, *Manage SSH Keys for SFTP Transactions*.

Create a User Store

To create a user store:

1. Click Credentials from the menu bar.

2. Click Actions > New User Store.

3. Specify a user store name in the User Store Name field.

4. If desired, change the default values for the following fields:

   ◆ User Lockout Duration

   ◆ User Lockout Threshold

5. Click New to add a user account to the user store. You must create at least one user account in the user store before you can save it.

6. Specify the following values for the user account:

   ◆ User Name

   ◆ Password

   ◆ Confirm Password

7. To enforce a password policy for this account, select a password policy from the list.

8. If desired, provide the following information for the user:

    ◆   First Name

    ◆   Last Name

    ◆   Email Address

    ◆   Pager

    ◆   Manager ID

9. Click OK.

10. Click Save.

Copy a User Store

To copy a user store:

1. Click Credentials from the menu bar.

2. Expand the User Stores tree.

3. Select the user store to copy.

4. Click Actions > Copy Selected.

5. Type a name for the new user store.

6. Edit the properties as needed.

7. Click Save.

Delete a User Store

To delete a user store:

1. Click Credentials from the menu bar.

2. Expand the User Stores tree.

3. Select the user store to delete. The default user store cannot be deleted.

4. Click Actions > Delete Selected.

5. Click Delete.

Create an Engine User Account

Create a user account to provide access to the engine. To create an engine user account:

1. Click Credentials from the menu bar.

2. Expand the User Stores tree.

3. Click the user store to which you want to add a user account.

4. Click New.

5. Specify the following values for the user account:

   ◆ User Name (no spaces allowed)

   ◆ Password

   ◆ Confirm Password

6. To enforce a password policy for this account, select a password policy from the list.

7. Click OK.

8. Click Save.

Add SSH Keys to a User Account

To perform local user authentication for a user account that will be used to access SSP for SFTP connections, you can associate SSH keys with that account.

To add SSH keys to a user account:

1. Click Credentials from the menu bar.

2. Expand the User Stores tree.

3. Click the user store to where the user account is stored.

4. Select the user account to add the SSH key to and click Edit.

5. Click the Advanced tab.

6. Select an SSH Authorized User Key Store from the list or click + to create a new User Key Store. Refer to *Create a User Store* on page 89.

7. Select the SSH Authorized User Keys that can be used by this user. Use Shift + Ctrl to select mutiple keys.

8. Click OK.

9. Click Save.

Edit an Engine User Account

To edit an engine user account:

1. Click Credentials from the menu bar.

2. Expand the User Stores tree.

3. Click the user store to edit.

4. Select the user account to edit and click Edit.

5. Edit the user properties.

6. Click OK.

7. Click Save.

Copy an Engine User Account

You can copy an engine user account to create a new user account.

To copy an account:

1. Click Credentials from the menu bar.
2. Expand the User Stores tree.
3. Click the user store that contains the user account to copy.
4. Select the user account to copy and click Copy.
5. Type a name for the account.
6. Edit the user properties as needed.
7. Click OK.
8. Click Save.

Delete an Engine User Account

You can delete a user account as needed to maintain the security of SSP.

To delete a user account:

1. Click Credentials from the menu bar.
2. Expand the User Stores tree.
3. Click the user store that contains the user account to delete.
4. Select the user account to delete.
5. Click Delete.
6. Click Save.

Connect:Direct Proxy Configuration

The Connect:Direct configuration scenarios describe how to configure Connect:Direct protocol connections to and from the SSP engine using the Configuration Manager.

Note:	Make sure the engine is running when you configure a Connect:Direct adapter. If it is running, configuration information is transmitted to the engine when you save. Configuration information must be available at the engine before communication sessions with Connect:Direct can be established.

Organization of the Connect:Direct Configuration Scenarios

The first scenario instructs you how to do a basic setup. Each successive scenario adds an additional security feature to the basic configuration. After you go through each scenario, test the connection to ensure that it is correctly configured. You determine your security needs and configure the security features applicable to your environment.

The scenarios include the following:

✦ Create a basic Connect:Direct configuration

✦ Add SSL/TLS support

✦ Configure PNODE-based routing

✦ Add local user authentication

✦ Copy data or run a program based on the success or failure of a Connect:Direct Process step

✦ Block Connect:Direct tasks from a PNODE

The remaining configuration scenarios require Sterling External Authentication Server (EA), an optional security feature of SSP that must be configured independently of SSP. After EA is configured, you can update your basic security definitions to enable SSP to connect to EA to enforce the following advanced security features:

✦ Authenticate an inbound certificate or user using EA

✦ Configure user mapping

✦ Configure certificate-based routing

✦ Perform user mapping to the SNODE using EA

Additional procedures are provided to instruct you how to configure the following features:

✦   Define alternate nodes for failover support

✦   Enable action based on protocol errors

Complete Scenario Worksheets

Before you perform each Connect:Direct configuration, gather the information on the worksheet provided. You use this information as you configure each feature. Complete worksheets as follows:

✦   Enter a value for each listed SSP feature. Fields listed in the worksheet are required.

✦   Accept default values for fields not listed in the worksheet.

✦   The worksheet identifies the Configuration Manager field where you will specify each value.

Complete and Test Configuration Scenarios

Work through the sequence of Connect:Direct configuration scenarios in the order in which they are presented to add security features. Be sure to test each feature before you add the next one to the configuration. Before you move SSP into production, ensure that you have configured and tested all security features you need for your environment.

Note: As you complete each task, provide all required information. If information is not provided or is incorrect, the following error icon is displayed: 🔴 . To view more information about the error, hover over the icon.

Create a Basic Connect:Direct Configuration

This scenario contains all the information and tools you need to configure SSP to establish a basic connection between Connect:Direct servers. Using default values, the PNODE presents a User ID to connect to the SNODE without EA. As a result, no authentication occurs in SSP and the user ID presented by the PNODE is used to connect to the SNODE. The basic configuration uses standard

routing to route connections to the node you define in the adapter. You are instructed on how to configure PNODE routing, mixed routing, and certificate-based routing in later scenarios.
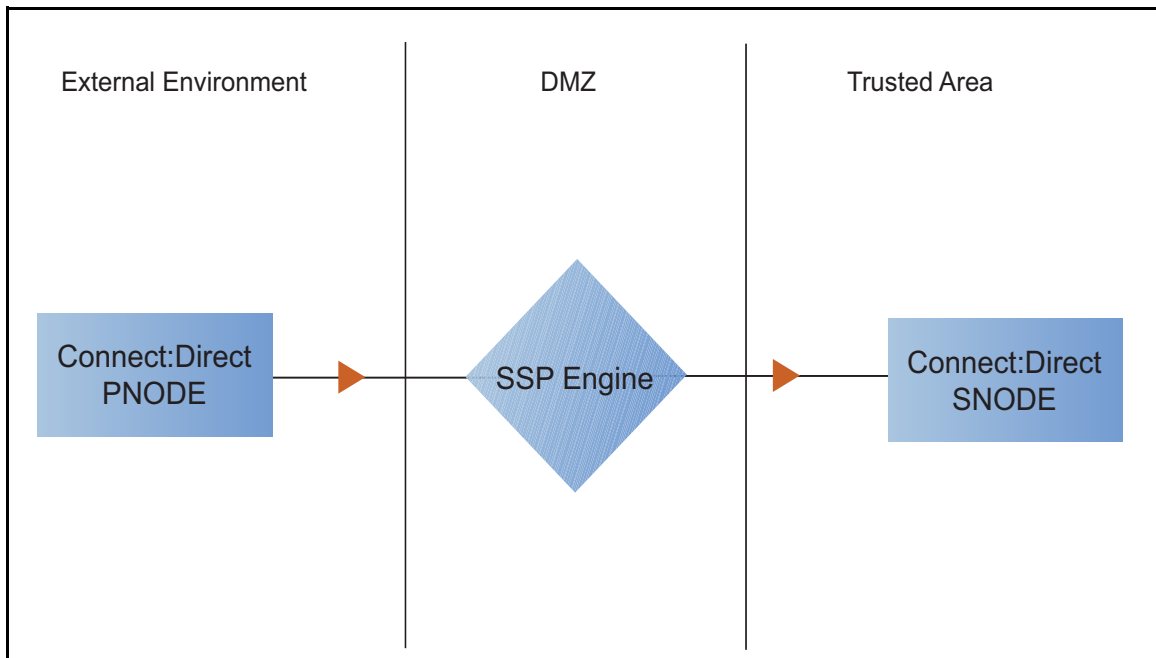
Before you configure a Connect:Direct connection, make sure that an engine has been configured. Refer to the *Sterling Secure Proxy Installation Guide* for instructions.

After you configure SSP, validate the configuration by initiating a Connect:Direct connection from the PNODE. For more information on testing the configuration, see *Test the Connect:Direct Connections* on page 117.

Complete the following tasks to define a basic Connect:Direct configuration:

✦ Create a policy
✦ Define Connect:Direct nodes in a netmap
✦ Define a Connect:Direct adapter

Basic Connect:Direct Configuration Worksheet

Before you configure SSP for Connect:Direct connections, gather the information on the basic Connect:Direct configuration Worksheet. You use this information as you configure a basic Connect:Direct connection for SSP. After you configure Connect:Direct connections, validate the configuration by initiating a Connect:Direct connection from the PNODE.

Policy

Create a basic policy. In a later Connect:Direct configuration scenario, you edit this policy to add security features to it.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy	_____

Netmap (All Connect:Direct Nodes)

Create a netmap that contains connection information for the nodes connecting to and from SSP. For each node, associate a policy with the node.

Configuration Manager Field	Feature	Value
Netmap Name	Netmap name	_____
Connect:Direct Node Definitions		
Node Name	Name to assign to the Connect:Direct node definition	_____
Connect:Direct Server Address	Host name or IP address of the Connect:Direct server	
Connect:Direct Port	Listening port number of the Connect:Direct server	_____
Policy	Name of policy you create (Select from a pull-down list.)	_____
Node Name	Name to assign to the Connect:Direct node definition	_____
Connect:Direct Server Address	Host name or IP address of the Connect:Direct server	_____
Connect:Direct Port	Listening port number of the Connect:Direct server	_____
Policy	Name of policy you create (Select from a pull-down list.)	_____

Connect:Direct Adapter

Create a Connect:Direct adapter that defines information necessary to establish Connect:Direct connections to and from SSP. When configuring the adapter, select the basic netmap and the Connect:Direct server where connections are routed and defined in the netmap definition.

Configuration Manager Field	Feature	Value
Name	Adapter name	_____
Listen Port	Listen port to use for inbound connections	_____
Netmap	Netmap to associate with the adapter	_____

Configuration Manager Field	Feature	Value
SNODE Netmap Entry	Name of Connect:Direct node where the connection is routed	_____
Engine	Engine to run the Connect:Direct adapter on	_____

Create a Basic Connect:Direct Policy

The policy defines how you impose controls to authenticate a Connect:Direct PNODE trying to communicate with a Connect:Direct SNODE over the public Internet. The basic policy does not enforce any controls over the defined node. You add security controls when you define more advanced security settings.

To define a basic policy:

1. Select Configuration from the menu bar.

2. Click Actions > New Policy > C:D Policy.

3. Type a Policy Name.

4. Click Save.

Create a Connect:Direct Netmap

You define connection information for every Connect:Direct node that communicates using SSP. These values are stored in a netmap. The netmap is associated with a policy and an adapter.

Before you begin this procedure, create a policy to associate with the netmap.

To create a netmap and define Connect:Direct nodes:

1. Select Configuration from the menu bar.

2. Click Actions > New Netmap > C:D Netmap.

3. Type a Netmap Name.

4. To define a Connect:Direct node definition, click New.

5. Specify the following values:

    ◆ Node Name

    ◆ Connect:Direct Server Address or hostname

    ◆ Connect:Direct Server Port (listening port)

    ◆ Policy

    Note: If you have not defined a policy, click the green plus sign to define one.

6. Click OK.

7. Repeat steps 3 through 5 for each node you want to define. Define at least one PNODE and at least one SNODE in order to establish a connection between two Connect:Direct nodes.

8. Click Save.

Define the Connect:Direct Adapter Used for the Connection

A Connect:Direct adapter definition specifies system-level communications information necessary for Connect:Direct connections through SSP.

Before you begin this procedure, create a netmap and an engine to associate with the adapter.

To define a Connect:Direct adapter:

1. Select Configuration from the menu bar.

2. Click Actions > New Adapter > C:D Proxy.

3. Specify values for the following:

   - Name

   - Listen Port

   - Netmap

   - SNODE Netmap Entry

   - Engine

4. Click Save.

What You Defined with the Basic Connect:Direct Configuration Scenario

Creating connections between Connect:Direct nodes when routing them through SSP requires that you organize information about the Connect:Direct nodes in a policy, a netmap, and an adapter definition. You created these items when you defined the basic Connect:Direct configuration. The next step is to test the configuration to ensure that the connections work. Before you test the configuration, be sure that:

✦ The Connect:Direct SNODE server has a definition in its netmap for the Connect:Direct PNODE. For Connect:Direct for Windows, set the netmap.check parameter to N.

✦ The PNODE server has a definition in its netmap for the SNODE, using the IP address and port of the SSP server.

✦ The user ID and password provided by the PNODE are defined at the Connect:Direct SNODE.

Refer to *Test the Connect:Direct Connections* on page 117 for information about testing the Connect:Direct proxy configuration outlined in this scenario.

As you add complexity to your security configurations using the procedures in the remaining scenarios, you modify the basic configuration to configure more complex authentication and certificate validation measures.

Add SSL/TLS Support

This scenario builds on the basic Connect:Direct configuration by enabling security for the nodes you defined in the netmap.

Adding SSL/TLS support to the netmap for the nodes involves selecting the following options for the connections:

✦ SSL or TLS Protocol

✦ Cipher suites

✦ Certificate stores and certificates

Add SSL/TLS support to the PNODE and the SNODE definitions. Set up Secure+ parameter files at both the SNODE and the PNODE servers. Obtain certificates for both sessions and check them into the certificate store. Then, test the connection.

Note: This procedure assumes you have checked in your certificates. Refer to Chapter 3, *Manage Certificates for SSL/TLS Transactions with Trading Partners*, for more information.

SSL/TLS Support Worksheet

Before you add SSL/TLS support to the connection information you created in the basic Connect:Direct configuration scenario, gather the information on the SSL/TLS Support Worksheet. You use this information as you configure the inbound and outbound nodes for SSL/TLS support.

Select the security setting and cipher suites to be used to secure the connection. To require that the certificate common name be validated in a certificate presented, enable this option and identify the common name value to check. Select the key/system certificate to use to validate the connection.

Configuration Manager	Feature	Value
Node Name	Name of the node to add security to	Select a node definition that you have already defined
Use Secure+	Enable this option to enable security checking	Enabled
Verify Common Name	Enable this option to enable common name checking. This is optional.	_____
Certificate Common Name	Value of common name in certificate presented, if Common Name Checking is enabled.	_____
Security Setting	Security protocol to use. Options include SSL, TLS, or The PNODE host controls SSL Protocol.	_____
Trust Store	Name of the store for the CA certificate or trusted root certificate.	
CA Certificates/Trusted Root	Name of CA certificate/trusted root	_____
Key Store	Name of the store for the key or system certificate is stored.	_____
Key/System Certificate	Name of the SSP system certificate presented to the Connect:Direct server.	_____
Available Cipher Suites	Select the ciphers to enable by moving them from the Available Cipher Suites to the Selected Cipher Suites field.	_____ _____ _____ _____

Secure the Connect:Direct Connection Using the SSL or TLS Protocol

The first step to strengthen security is to secure the communications channel. This procedure describes how to enable the SSL or TLS protocol for the Connect:Direct connections to and from SSP in a netmap you created in the basic configuration. To require that SSP perform common name checking, enable this option and identify the common name in the configuration.

Before you can configure this option, you must obtain the necessary certificates and place them in the SSP Cert Store. Refer to Chapter 3, *Manage Certificates for SSL/TLS Transactions with Trading Partners* for instructions.

To enable the SSL or TLS protocol:

1. Select Configuration from the menu bar.

2. Expand the Netmaps tree and select a netmap to modify.

3. Select a node to modify, and click Edit.

4. Click the Security tab, and then click Use secure+ to enable security.

5. To enable common name checking:

   a. Click Verify Common Name.

   b. Type the certificate common name in the Certificate Common Name field.

6. Select values for the following:

   ◆ Security Setting

   ◆ Trust Store

   ◆ CA Certificates/Trusted Root

   > **Note:** Be sure to highlight the certificate to select. If only one certificate is displayed in the field, it is not selected until you highlight it.

   ◆ Key Store

   ◆ Key/System Certificate

   ◆ Selected Cipher Suites

7. Click OK.

8. Click Save.

Establish a session initiated by a Connect:Direct PNODE to test the configuration.

Variation on the Add SSL/TLS Support Configuration

After you confirm that the communications session you established using the Add SSL/TLS Support scenario was successful, you may want to further modify your configuration. After testing the SSL/TLS configuration, you can configure the environment to allow the inbound and outbound sessions to use different levels of encryption.

Allow Different Levels of Encryption for the Inbound and Outbound Node

In a Connect:Direct environment where SSP is not being used, one session is established between an SNODE and a PNODE. In the SSP environment, a session break is created; therefore, two sessions are established: one between the PNODE and SSP and another between SSP and the SNODE. To use the same protocol on both sessions, use the default settings.

Complete this procedure to define one protocol for the inbound node and a different protocol for the outbound node. This function is useful when you want to secure the inbound connection but allow a nonsecure session between SSP and the outbound node.

To enable different levels of encryption for the inbound and the outbound connection:

1. Select Configuration from the menu bar.

2. Expand the Adapter tree, and select the adapter you want to modify.

3. Click the Advanced tab.

4. Enable the Inbound and outbound sessions can have different levels of encryption option.

5. Click Save.

Configure PNODE-Based Routing

The basic configuration uses standard routing to determine where a connection is routed. If you configure standard routing, all sessions through an adapter are routed to the same connection. To allow a PNODE to determine what SNODE it connects to, configure PNODE-based routing. For PNODE-based routing, you must configure a node definition in the netmap for the PNODE and for all the SNODEs you will route to.

Note: PNODE-based routing is supported for Connect:Direct for z/OS version 4.6 or higher, Connect:Direct for UNIX version 3.8 or higher, and Connect:Direct for Windows version 4.4 or higher.

PNODE-based Routing Worksheet

This scenario builds on the basic Connect:Direct configuration by enabling PNODE-based routing. Before you add PNODE-based routing to the connection information you created in the basic Connect:Direct configuration scenario, gather the information on the PNODE-based Routing Worksheet. You use this information as you configure PNODE-based routing.

In the netmap you select, make sure that you have a node definition for the PNODE and for every node where the connection is routed.

Configuration Manager Field	Feature	Value
Name	Adapter name	_____
Netmap	Netmap to associate with the adapter	_____
Listen Port	Adapter port number	_____
Routing Type	Routing type to use for this connection	PNODE-specified

Configure PNODE-based Routing

To configure a Connect:Direct adapter to use PNODE-based routing:

1. Select Configuration from the menu bar.

2. Expand the Adapter tree and select the adapter you want to modify.

3. Select PNODE-specified in the Routing field.

4. Click Save.

Configure Mixed Routing

Mixed routing allows a PNODE to determine what SNODE it connects to. If the PNODE does not identify what SNODE to connect to, mixed routing then routes to the SNODE identified in the SSP configuration. Before PNODE-based routing can be implemented, you must configure a node definition in the netmap for the PNODE and the SNODE.

Note: PNODE-based routing is supported for Connect:Direct for z/OS version 4.6 or higher, Connect:Direct for UNIX version 3.8 or higher, and Connect:Direct for Windows version 4.4 or higher.

Mixed Routing Worksheet

This scenario builds on the basic Connect:Direct configuration by enabling PNODE-based routing. Before you add PNODE-based routing to the connection information you created in the basic Connect:Direct configuration scenario, gather the information on the Mixed Routing Worksheet. You use this information as you configure PNODE-based routing.

Make sure that you have a node definition for the PNODE and for the node where the connection is routed in the netmap you select.

Configuration Manager Field	Feature	Value
Name	Adapter name	_____
Netmap	Netmap to associate with the adapter	_____
SNODE Netmap Entry	Name of Connect:Direct node where the connection is routed	_____
Routing Type	Routing type to use for this connection	PNODE-specified and then Standard (mixed)

Configure PNODE Specified and Then Standard (Mixed) Routing

To configure a Connect:Direct adapter to use PNODE specified and then standard (mixed) routing:

1. Select Configuration from the menu bar.

2. Expand the Adapter tree and select the adapter you want to modify.

3. Select PNODE-Specified, then Standard (mixed) in the Routing Type field.

4. Select the SNODE to route connections to in the SNODE Netmap Entry field.

5. Click Save.

Add Local User Authentication to a Connect:Direct Connection

This scenario builds on the basic Connect:Direct configuration by adding local user authentication to the PNODE connection using information defined in the local user store. The user ID and password presented by the PNODE are authenticated against information stored in the local user store. The values must match before a connection is established. You must add this information to the local user store before you can test this scenario.

Adding user authentication to the PNODE connection defined in the basic Connect:Direct configuration involves enabling user authentication and specifying information about the PNODE.

After you configure local user authentication, validate the configuration by establishing a session initiated by a Connect:Direct PNODE.

Connect:Direct PNODE Connection (Local User Authentication) Worksheet

Before you add local user authentication to the PNODE connection you created in the basic Connect:Direct configuration scenario, gather the information on the Connect:Direct PNODE Connection (Local User Authentication) Worksheet. Use this information as you configure user authentication for the PNODE connection.

In this scenario, you edit the policy you created in the Connect:Direct basic configuration scenario and enable user authentication. You also add a user ID and password for the Connect:Direct PNODE to the default user store.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy associated with the inbound node	_____
User Authentication	Method to use to authenticate the inbound node	Through local user store
User Store	Name of the user store you create	_____
User Name	Name of the user you define in the User Store	_____
Password Confirm Password	The password value to use to validate the inbound connection	_____

Add User Authentication to the Connect:Direct Inbound Connection

You can strengthen the security of Connect:Direct PNODE connections by enabling local user authentication. This procedure describes how to configure local user authentication.

Note: Check the netmap to ensure that the policy you select is associated with the PNODE you want to authenticate.

To add local user authentication for a PNODE connection:

1. Select Configuration from the menu bar.
2. Expand the Policies tree and select a policy.
3. Click the Advanced tab.
4. Enable the User Authentication Through Local User Store option.
5. Click Save.

Add Credentials to the Local User Store

If you enable user authentication through the local user store, you also add user information to the local user store that is validated by SSP during a Connect:Direct client connection.

Before you begin this procedure:

✦ Enable user authentication for the inbound connection.

✦ Ensure that the engine is configured to use the user store that contains the user credentials.

To add user information to the local user store:

1. Select Credentials from the menu bar.

2. Click User Stores to expand the list of user stores.

3. Select the default user store called defUserStore.

4. From the User Store Configuration panel, click New.

5. Specify values for the following:

   ◆ User Name

   ◆ Password

   ◆ Confirm Password

6. Click OK.

7. Click Save.

Copy Data or Run a Program Based on the Success or Failure of a Process Step (Step Injection)

This scenario builds on the basic Connect:Direct configuration by adding step injection functions to the PNODE connection. Step injection allows you to insert Connect:Direct Process statements into the communications session with the SNODE independent of the PNODE Process statements. These injected statements can provide real-time notification of file delivery, invoke applications, run operating system jobs and commands, and submit other Connect:Direct Processes, all without the need to provide an exit program on the SNODE or without changing the PNODE Process. Even though the PNODE has no indication that these steps have been executed on the SNODE, step injection is defined on the PNODE record in SSP. The results of these steps are logged in the statistics file of the SNODE.

To use step injection, define one or more of the following step injection functions:

✦ Copy session or certificate information to a file at the SNODE at the end of a successful step.

✦ Execute a Connect:Direct Runtask, Runjob, or Submit step on the SNODE at the end of a successful step or run operating system commands, jobs, or programs.

✦ Copy session or certificate information to a file at the SNODE at the end of a failed step.

✦ Execute a Connect:Direct Runtask, Runjob, or Submit step on the SNODE at the end of a failed step or run operating system commands, jobs, or programs.

✦ Use variables to define file names and parameters in a Runtask, Runjob, or Process step.

Step Injection actions referenced in a netmap entry are only performed when the node is communicating as the PNODE in the transaction.

Configure Step Injections Worksheet

Before you create step injection definitions and associate them with a node definition you created in the basic Connect:Direct configuration scenario, gather the information on this worksheet. You use this information as you configure a node with step injection support.

Configuration Manager Field	Feature	Value
Step Injection Name	The name to assign to the step injection you define.	_____
Copy on success	Turn on this option to copy session-specific data to the SNODE at the end of a successful step.	Enabled? Yes or No _____
Copy identifying information	The information to copy to the SNODE. This field is required if you turn on Enable Copy on success.	Select one of the following options: Copy All Information Copy Certificate Information Copy Session Information
Session information output file	Destination file on the SNODE where information is copied. Variables can be used to define the file name. Refer to *Use Variables in a Step Injection Definition* on page 109.	_____
Tcp timeout for copy	Number of seconds to wait for a request or response before ending the session.	
Execute on success	Turn on this option to execute a Runtask, Runjob, or Submit with a Process at the end of a successful step or execute an operating system command or program.	Enabled? Yes or No _____
Step selection	The step type to execute when a successful step occurs.	Select one of the following options: Runtask, Runjob, Submit
Step parameter	Type the parameters to use for the step. Variables can be used to define the parameters. Refer to *Use Variables in a Step Injection Definition* on page 109.	_____
Tcp timeout for step	Number of seconds to wait before timing out.	
Copy on failure	Turn on this option to copy session-specific data to the SNODE at the end of a failed step.	Enabled? Yes or No _____
Copy identifying information	Type of information to copy to the SNODE if a Process step is unsuccessful.	Select one of the following options: Copy All Information Copy Certificate Information Copy Session Information
Session information output file	Destination file where the copy information is written. Variables can be used to define the file name. Refer to *Use Variables in a Step Injection Definition* on page 109.	_____

Configuration Manager Field	Feature	Value
Tcp timeout for step	Number of seconds to wait before the copy instruction is timed out.	
Execute on failure	Turn on this option to execute a Runtask, Runjob, or Submit Process on the SNODE as defined in a submitted Process at the end of a unsuccessful step or execute an operating system command or program.	Enabled? Yes or No _____
Step selection	Select the step type to execute when a successful step occurs.	Select one of the following options: Runtask Runjob Submit
Step parameter	Define the parameters to use for the step. Variables can be used to define the parameters. Refer to *Use Variables in a Step Injection Definition* on page 109.	_____
Tcp timeout for copy	Identify how many seconds to wait before the copy instruction is timed out.	

Configure a Step Injection

Before you can associate a step injection with a node, you must first define the actions to take in a step injection function.

To configure a step injection:

1. Select Advanced from the menu bar.

2. Click Actions > New C:D Step Injection.

3. Type a step injection name.

4. Click the Advanced tab.

5. To copy information into a file at the SNODE:

    a. Take one of the following actions:

       • Enable Copy on success to copy information to a file after a successful Process copy statement has occurred.

       • Enable Copy on failure to copy information to a file after a Process copy statement has failed.

    b. Select the type of information to copy to the file in the Copy identifying information field. Options include Copy All Information, Copy Certificate Information, or Copy Session Information.

    c. Type the name of the file where the information is copied in the Session information output file field.

    d. Enter how many seconds to wait until the session is timed out in the Tcp timeout for copy field.

Sterling Secure Proxy Configuration Guide

6. To execute a Runtask, Runjob, or submit another Connect:Direct Process or execute an operating system command or program at the SNODE:

   a. Take one of the following actions:

      • Enable Execute on success to perform an action after a successful Process copy statement.

      • Enable Execute on failure to perform an action created after a Process copy statement fails.

   b. Select the type of step to perform in the Step selection field. Options include Runtask, Runjob, or Submit a Connect:Direct Process or execute an operating system command or program.

   c. Define the step parameters to use. Refer to the Connect:Direct documentation for more information.

   d. Enter how many seconds to wait before the session is timed out in the Tcp timeout for step field.

7. Click Save.

Use Variables in a Step Injection Definition

When you configure a step injection function, you can use variables in the Session information output field and Step parameter field. These variables allow you to name output files or execute step parameters based on information obtained during the session. Use the following variables within a step injection action definition:

Variable	Description
${%DESTFILE%}	Destination file name defined in the Process.
	Example: Runtask - cmd (copy ${%DESTFILE%} C:\outout\sspfile\dest.txt)
${%DESTUID%}	Destination user ID defined in the Process.
${%DESTWPATH%}	Destination file name defined in the Process, including the path.
${%FROMNODE%}	The node that is sending the file. Returned values are P for PNODE or S for SNODE.
	Example: CopyonSuccess = C:\Output\copysuccessallinfo_${%FROMNODE%}_${%SNODE%}_.txt
${%ORGININUID%}	User ID of the person who initiated the Process.
${%PNAME%}	Process name.
${%PNODE%}	Name of the PNODE that initiated the Process.
${%PNUM%}	Process number.
${%SNODE%}	Name of the destination SNODE name where the session is running.
${%SOURCEFILE%}	Source file name defined in the Process step.
${%SOURCEWPATH%}	Source file name defined in the Process step, including the path.

Variable	Description
${%STEPCOMPLETE%}	What time and date the step completed, in the format yyyymmdd_hhmmsshh, where yyyy is year, mm = month, dd = day, hh = hour, mm = minute, ss = seconds, and hh = hundredth of seconds.
${%STEPMSG%}	A message ID.
${%STEPNAME%}	Name of the step.
${%STEPSTART%}	The time and date the step started, in the format yyyymmdd_hhmmsshh, where yyyy is year, mm = month, dd = day, hh = hour, mm = minute, ss = seconds, and hh = hundredth of seconds.
${%TS%}	The time the session began, in milliseconds.
${%TSNOW%}	The current time, in milliseconds, in the format 1132599441883.

Associate a Step Injection With a Connect:Direct Node

After you configure step injection functions, you can then associate a step injection with a Connect:Direct node. Process steps are activated by a PNODE; therefore, step injection functions must be defined in a PNODE record.

To associate a step injection with a Connect:Direct node:

1. Select Configuration from the menu bar.

2. Expand the Netmaps tree, and select the netmap that contains the PNODE definition you want to modify.

3. Select the node to modify and click Edit.

4. Select the step injection function to associate with the node from the Step Injection drop-down

   list. If you have not defined the step injection function, click ⊕ and define a step injection. Refer to *Copy Data or Run a Program Based on the Success or Failure of a Process Step (Step Injection)* on page 106 for instructions.

5. Click OK.

6. Click Save.

Block Connect:Direct Tasks Allowed on a Node

This scenario builds on the basic Connect:Direct configuration by adding the capability to prevent Connect:Direct statements from being executed.

To prevent a Connect:Direct statement from being executed:

1. Select Configuration from the menu bar.

2. Expand the Policies tree and click the policy to modify.

3. On the Policy Configuration panel, click the Step Permissions tab.

4. Click on one or more of the following tasks to disable the task:

   ◆ Runjob step allowed

   ◆ Runtask step allowed

   ◆ Copy step allowed

   ◆ Submit step allowed

5. Click Save.

Strengthen User Authentication Using EA

This scenario builds on the basic Connect:Direct configuration by adding user authentication to the PNODE connection using information defined in EA. To provide a more advanced method of securing a Connect:Direct connection, use EA such as, authenticating certificate information or user credentials presented by the inbound node or performing user ID and password mapping.

Authenticate an Inbound Certificate or User Using EA

You can authenticate an inbound connection against information stored in an LDAP database by configuring EA to define how the connection is authenticated. The EA definition determines the options that are enabled. Refer to Sterling External Authentication Server help for a complete list of the functions that can be performed in EA.

Authenticate a Certificate or User Using EA - Worksheet

Use the following worksheet to identify the information needed to authenticate a Connect:Direct connection using information in EA. Update the policy you created in the basic Connect:Direct configuration for this scenario.

Configuration Manager Field	Information	Value
Certificate Authentication - External Authentication Certificate Validation	Will you validate the certificate presented by the PNODE?	_____ (Yes or No)
Certificate Authentication - External Authentication Profile	If yes, provide the EA certificate validation definition.	_____
User Authentication - Through External Authentication	Will you validate user information?	_____ (Yes or No)
User Authentication - External Authentication Profile	If yes, provide the EA user validation definition.	_____

Authenticate a Connect:Direct Certificate or User Using EA

To authenticate certificate information or user information about the Connect:Direct node against information stored in an LDAP database, you must configure EA. After you configure EA to enable certificate or user authentication, complete this procedure to configure SSP to use the authentication method you defined.

Before you configure SSP to use EA to authenticate a node connection, obtain the name of the EA definition.

In addition, ensure that the following procedures have been performed:

✦   The public keys for SSP have been sent to the EA server and imported into the EA keystore.

✦   The EA server connection has been configured in SSP.

To configure authentication of a Connect:Direct node using EA:

1.   Select Configuration from the menu bar.

2.   Expand the Policies tree and click the policy to modify.

3.   On the Policy Configuration panel, click the Advanced tab.

4. Configure one or more of the following options:

   ◆ To validate the certificate presented by the node against information defined in EA, enable Certificate Authentication - External Authentication Certificate Validation and enter the name of the profile you defined in EA in the Certificate Authentication - External Authentication Profile field.

   ◆ To enable user authentication through EA, enable User Authentication - Through External Authentication and type the name of the definition you defined in EA in the User Authentication - External Authentication Profile field.

5. If you do not want to authenticate the user using information in the local user store, deselect the Through Local User Store option.

6. Click Save.

You can now associate this policy with a Connect:Direct node where you want to perform user authentication using information stored in an LDAP database.

Strengthen the Connection to the SNODE With User Mapping

This scenario builds on the basic Connect:Direct configuration by adding user mapping using information defined in Sterling External Authentication Server (EA). To provide a more advanced method of securing a Connect:Direct connection, use EA to map a PNODE user ID and password or PNODE submitter ID to login credentials stored in EA. The mapped login credentials are then used to connect to the SNODE.
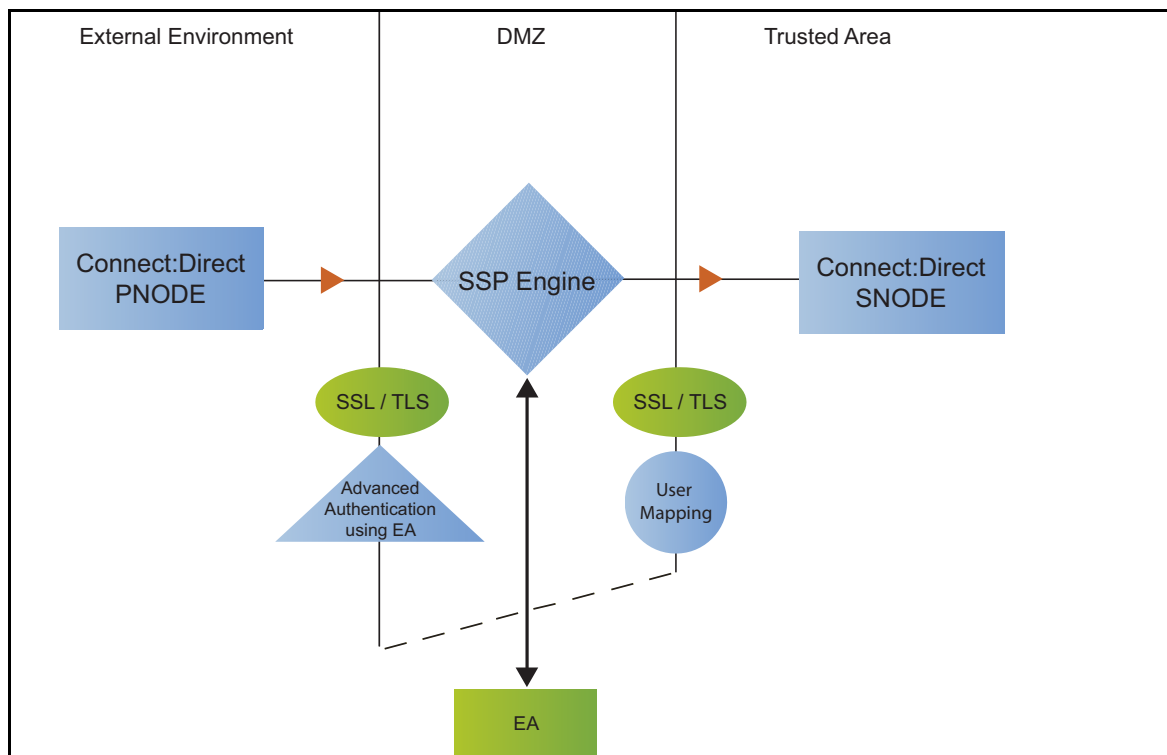
Perform User Mapping Using EA - Worksheet

Use this worksheet to identify the user mapping method to enable for the SNODE connection with information in EA:

Configuration Manager Field	Feature	Value
Replace SNODEID with Userid mapped in External Authentication	The PNODE requires a user ID for access to the SNODE and the user ID provided is replaced with a value defined in EA.	Enabled? Yes or No _____
Replace submitter id with Userid mapped in External Authentication	The PNODE requires a submitter ID to access the SNODE. The submitter ID supplied by the PNODE is replaced by a valued defined in EA.	Enabled? Yes or No _____
Destination Service Name	The name of the service. If no value is provided, the SNODE is used as the service name.	_____

Perform User Mapping Using Information Stored in EA

If you store user credentials in an LDAP database, use this procedure to map a user ID and password, or a submitter ID provided by the SNODE, to information stored in EA. Two methods are available: you can replace the SNODE ID with information stored in EA or you can replace the submitter ID.

Destination Service Name needs to be selected on the Advanced tab of the Netmap Node screen of the PNODE. If Destination Service Name is not provided, the SNODE name is used.

Before you configure this option:

✦ Configure a definition in EA.

✦ Obtain the name of the EA definition.

✦ Configure a connection between EA and the engine.

To configure user mapping:

1. Select Configuration from the menu bar.

2. Expand the Policies tree and click the policy to modify.

3. On the Policy Configuration panel, click the Advanced tab.

4. To enable user authentication through EA, enable the User Authentication Through External Authentication option and type the name of the definition you defined in EA in the External Authentication Profile field.

5. Do one of the following:

    ◆ To map the user ID presented by the PNODE to information in EA, select Replace SNODEID with UserId mapped in External Authentication.

    ◆ To map the submitter ID presented by the PNODE to information in EA, select Replace SubmitterID with UserId mapped in External Authentication.

6. Click Save.

7. In the Configuration panel, expand the Netmap option and click the netmap to modify.

8. Select the PNODE to modify and click Edit.

9. Click the Advanced tab.

10. Type the name of the service in the Destination Service Name field. If no value is provided, the SNODE name is used as the service name.

11. Click OK.

12. Click Save.

Configure Certificate-Based Routing

This scenario builds on the basic Connect:Direct configuration by configuring certificate-based routing. Certificate-based routing uses a routing name returned by EA. It is associated with the subject distinguished name found in the PNODE certificate. SSP uses this routing name to determine the SNODE where the incoming SSP connection is routed. To perform certificate-based routing, modify an adapter you defined in the basic Connect:Direct configuration.

The following diagram illustrates the certificate-based routing function:

Summary of Certificate-Based Routing

Following are the steps performed during certificate-based routing:

1. The PNODE passes a certificate chain during an SSL/TLS session. This certificate includes several attributes, such as subject and distinguished name (DN).

2. SSP passes the certificate chain to Sterling External Authentication Server (EA).

3. Using the configuration parameters in a certificate validation request, EA attempts to match PNODE certificate attributes to the LDAP server and requests the associated routing value.

4. LDAP returns the routing value to EA.

5. EA passes the routing value to the SSP engine.

6. SSP routes the PNODE request to the SNODE using the routing value.

Configure Certificate-Based Routing in SSP

Before you test certificate-based routing, you must create a certificate validation request in EA that includes an attribute query definition called Routing Names. This attribute query definition is created to retrieve a routing name value using certificate attributes as search criteria. You must also configure a connection between SSP and EA.

Refer to Chapter 12, *Configure SSP for Sterling External Authentication Server (EA)* for instructions.

To configure certificate-based routing:

1. Select Configuration from the menu bar.

2. Expand the Adapters tree and select the adapter you want to modify.

3. Select Certificate-based in the Routing Type field.

4. Click Save.

5. Click the Netmap navigation panel, expand the Netmap tree, and select the Connect:Direct adapter that contains the SNODE where the connection are routed.

6. Select the node to modify and click Edit.

7. Type the routing value to be returned from the LDAP server in the Routing Name field. The routing name must exactly match the routing value returned from the LDAP server. This routing name identifies the SNODE for routing the PNODE request.

8. Click OK.

9. Click Save.

10. Configure SSP to enable certificate authentication using EA. Refer to *Authenticate an Inbound Certificate or User Using EA* on page 112.

Test the Connect:Direct Connections

To verify that the engine can receive and initiate communications sessions, you have to establish a connection between a Connect:Direct PNODE and the engine, initiate a session from the engine to the Connect:Direct SNODE in the trusted zone, and review the SSP log for the results.

This procedure enables you to verify that the engine can:

✦ Establish a Connect:Direct session between a PNODE and SSP

✦ Initiate a session to a Connect:Direct SNODE on behalf of the Connect:Direct PNODE connection

To verify the communications sessions:

1. View the secureproxy.log.

2. Confirm that the sessions were established, as shown in the following example.

```
21 Dec 2008 16:47:16,874 INFO [PASConduit1pnode]
sys.NODE.CD_Netmap_Secure.ea_rhas40_cd3800 - protocol=cd sessid=119827723531001
CSP004I 0 Pnode session established. Pnode=ea_rhas40_cd3800
HNIPP=qarhas40.csg.stercomm.com,10.20.42.198;45892 XMLErrPolicy=NONE
FMHUpdate=granted NMCheck=NA RTPolicy=Yes RJPolicy=Yes SBPolicy=Yes CPPolicy=Yes
S+Policy=SA_OPTIONAL ExecPolicy=EX_STRONG SessLimit=20 Routing=STD
CSList=TLS_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_3DE
S_EDE_CBC_SHA, CSSelected=RSA_WITH_AES_128_CBC_SHA PNCert=Serial number: 230
Issuer:O=SCI, L=Irving, ST=Texas, C=US Subject:C=US, ST=Texas, O=SCI, OU=SV,
CN=donnieaix, EMAIL=qatest1024@stercomm.com Not Valid Before:Mon Dec 04 11:41:55
CST 2006 Not Valid After:Thu Dec 01 11:41:55 CST 2016 Signature
Algorithm:MD5withRSA

21 Dec 2009 16:47:17,490 INFO [PASConduit1pnode_3016_sessid=119827723531001]
sys.NODE.CD_Netmap_Secure.ea_rhas40_cd3800 - protocol=cd sessid=119827723531001
CSP005I 0 Snode session established with Snode=ea_sol10_cd3800
HNIPP=qasol10;23564 XMLErrPolicy=NONE FMHUpdate=granted NMCheck=NA RTPolicy=Yes
RJPolicy=Yes SBPolicy=Yes CPPolicy=Yes S+Policy=SA_OPTIONAL ExecPolicy=EX_STRONG
SessLimit=20 Routing=STD
CSList=TLS_RSA_WITH_3DES_EDE_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_WITH_AE
S_128_CBC_SHA, CSSelected=RSA_WITH_3DES_EDE_CBC_SHA SNCert=Serial number: 230
Issuer:O=SCI, L=Irving, ST=Texas, C=US Subject:C=US, ST=Texas, O=SCI, OU=SV,
CN=donnieaix, EMAIL=qatest1024@stercomm.com Not Valid Before:Mon Dec 04 11:41:55
CST 2006 Not Valid After:Thu Dec 01 11:41:55 CST 2016 Signature
Algorithm:MD5withRSA
21 Dec 2009 16:47:17,492 INFO [PASConduit1pnode_3016_sessid=119827723531001]
sys.NODE.CD_Netmap_Secure.ea_rhas40_cd3800 - protocol=cd sessid=119827723531001
CSP006I 0 Pnode/Snode proxy session established. Pnode=ea_rhas40_cd3800
.
.
.
```

If your session was unsuccessful, review the log information to determine the likely cause of the failure and the corrective action to take.

Additional Connect:Direct Configuration Options

Additional Connect:Direct configuration options support the following features:

✦ Define alternate nodes for failover support

✦ Record an error message or shutdown a connection based on protocol errors

Define Alternate Nodes for Failover Support

If you are using standard routing to connect to a Connect:Direct server in the secure zone, you identify a primary server to connect to in the adapter. The primary nodes are defined in the netmap. For each PNODE definition in the netmap, you can identify up to three alternate outbound nodes to connect to if the primary Connect:Direct server is not available.

Two methods of configuring alternate server routing are available.

✦ Select a previously defined outbound node from the drop-down list on the Netmap - Advanced tab. To configure this method, you first configure an outbound node definition in the netmap for each alternate node you want to use. Each connection uses the security and External Authentication settings defined for that outbound node in the netmap.

✦ Select IP address/port from the drop-down Node list on the Advanced tab and enter values for the IP address and port. If you use this method, you do not have to define the alternate outbound nodes in the netmap, and each alternate connection shares the security and EA settings defined in the primary node definition.

If you configure alternate server definitions in the PNODE definition, when a connection to the primary outbound node is unsuccessful SSP tries to connect to the alternate node you defined as Node 1. If the connection to the first alternate node is unsuccessful, SSP tries to connect to the second alternate node, Node 2 and then to the third alternate, Node 3. If all are unsuccessful, the inbound connection fails.

To configure alternate outbound connections:

1. Select Configuration from the menu bar.

2. Expand the Netmaps tree and click the netmap to modify.

3. Select the node to modify and click Edit.

4. Click the Advanced tab.

5. Do one of the following:

   ◆ To identify an alternate node that is defined in the netmap and use the security settings defined in the alternate node definition, select the outbound node name from the drop-down list.

   ◆ To configure an alternate node that is not in the netmap and use the security settings defined in the primary node definition:

      a. Select Address/Port from the drop-down list in the Alternate Destinations Node field.

      b. Provide the IP Address and Port number for the alternate outbound node.

6. Click OK.

7. Click Save.

Record an Error Message or Shut Down a Connection Based on Protocol Errors

To write a warning message to the log file or shut down a connection when a protocol violation occurs during a file transfer, enable this function in the Policy definition.

To enable an action based on a protocol error:

1. Select Configuration from the menu bar.

2. Expand the Policies tree and select the policy to modify.

3. Select the action to take on a protocol error in the Protocol Error Action field.

4. Click Save.

FTP Reverse Proxy Configuration

The FTP configuration scenarios describe how to configure FTP protocol connections to and from the SSP engine.

Note: Make sure the engine is running when you configure an FTP adapter. If it is running, configuration information is transmitted to the engine when you save it. Configuration information must be available on the engine before communication sessions with Gentran Integration Suite (GIS) can be established.

Organization of the FTP Configuration Scenarios

The first scenario instructs you how to configure a basic configuration. Each successive scenario adds a security feature to the basic configuration. After adding a security feature, test the connection to ensure that you have correctly configured it. You determine your security needs and configure the security features applicable for your environment.

The following scenarios help you configure and test SSP for FTP protocol connections to the GIS server:

✦ Create a basic FTP configuration

✦ Add SSL/TLS support

✦ Perform user authentication using the local user store

✦ Provide outbound credentials using the netmap

The remaining configuration scenarios require EA, an optional security feature that must be configured independently. After EA is configured, you can update your basic security definitions to enable SSP to connect to the EA to enforce the following advanced security features:

✦ Authenticate an inbound certificate or user using EA

✦ Manage connection requirements to the outbound server using EA

Other options help you do the following:

✦ Define alternate nodes for failover support

✦ Define a passive data outbound port range for an FTP Reverse Proxy adapter

✦ Define a passive NAT address for an FTP Reverse Proxy adapter

✦ Define an active data outbound port range for an FTP Reverse Proxy adapter

Complete FTP Scenario Worksheets

Before you begin configuring SSP for FTP connections, gather the information on the worksheet provided with the scenario. You use this information as you configure each feature. Complete worksheets as follows:

✦ Provide a value for each SSP feature listed. Fields listed in the worksheet are required.

✦ Accept default values for fields not listed.

✦ The worksheet identifies the Configuration Manager field where you will specify each value.

Complete and Test FTP Configuration Scenarios

Work through the sequence of FTP configuration scenarios in the order in which they are presented to add and test more security features. Be sure to test each feature before you add the next to the configuration. Before you move SSP into production, ensure that you have configured and tested the security features needed for your environment.

Note: As you complete each task, provide all required information. If information is not provided or is incorrect, the following error icon is displayed: 🔴 To view more information about the error, hover over the icon.

Create a Basic FTP Configuration

This scenario contains all the information and tools you need to configure SSP to establish a basic connection from a trading partner to the GIS server as illustrated below. You accept default values when configuring this scenario. As a result, no authentication occurs in SSP and credentials presented by the inbound node are passed through to the GIS server.
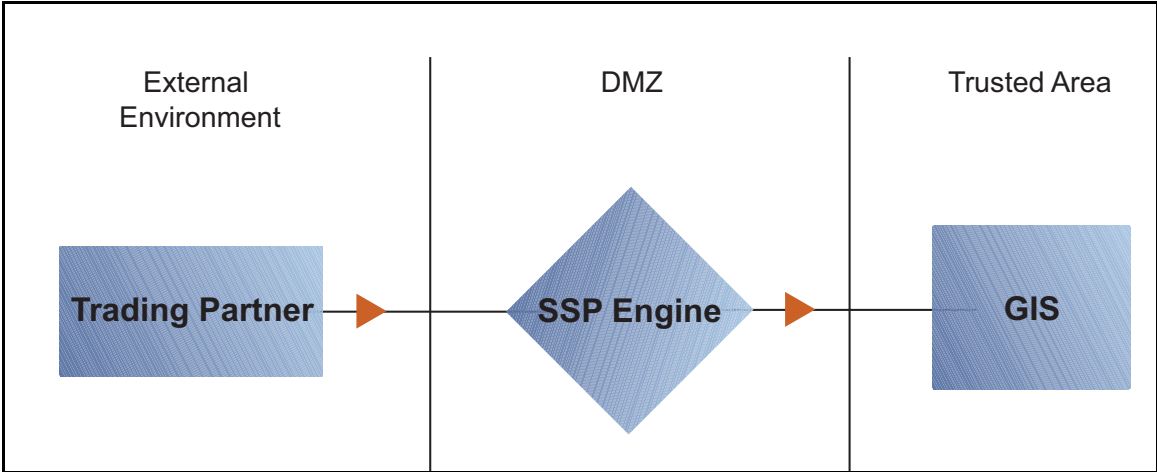
After you configure SSP, validate the configuration by initiating an FTP connection from the trading partner. For more information on testing the configuration, see *Test the Inbound and Outbound FTP Connections* on page 143.

Complete the following tasks to define a basic FTP configuration:

✦ Create a policy

✦ Define inbound and outbound connections in a netmap

✦ Define an FTP adapter

Basic FTP Configuration Worksheet

Before you configure SSP for FTP connections, gather the information on the Basic FTP Configuration Worksheet. You use this information as you configure a basic FTP connection for SSP.

FTP Policy

Create a basic policy. In a later FTP configuration scenario, you edit this policy to add security features to it.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy.	_____

FTP Netmap (Inbound and Outbound Connections)

Create a netmap that contains connection information for the nodes connecting to and from SSP: the trading partner (inbound node) and the GIS server (outbound node). You will also associate the basic security policy you create with the inbound node.

Configuration Manager Field	Feature	Value
Netmap Name	Name of the netmap.	_____
Inbound Trading Partner Information		
Inbound Node Name	Trading partner name (name to assign to inbound node definition).	_____
Peer Address Pattern	Host name or IP address pattern.	* (* allows all inbound nodes to connect to the GIS server, using this definition. To define a more specific node definition, see *Create a Basic FTP Configuration* on page 123.)
Policy	Name of policy you create.	This value is selected from a pull-down list. _____
Outbound FTPServer Connection		
Node Name	Outbound FTP server node name.	_____
Primary Destination Address	Host name or IP address to connect to the outbound FTP server.	_____
Primary Destination Port	Port number to connect to the outbound FTP server.	_____

FTP Adapter

Create an FTP adapter that defines information necessary to establish FTP connections to and from SSP. When you configure the adapter, select the basic netmap and the outbound FTP server you define in the netmap definition. If the outbound host uses virtual IP address, set the IP address in the PASV response.

Configuration Manager Field	Feature	Value
Adapter Name	Adapter name.	_____

Configuration Manager Field	Feature	Value
Listen Port	Listen port to use for inbound connections.	_____
Netmap	Netmap to associate with the adapter.	_____
Standard Routing Node	Name of the outbound node corresponding to the GIS server where inbound connections are routed.	_____
Engine	Engine to run on.	_____

Create an FTP Policy

The FTP policy defines how you impose controls to authenticate a trading partner trying to access a GIS server over the public Internet.

To define a policy:

1. If necessary, click Configuration from the menu bar.
2. Click Actions > New Policy > FTP Policy.
3. Type a Policy Name.
4. Click Save.

Create an FTP Netmap

You define inbound connection information for your trading partners and outbound connection information for the GIS server that SSP connects to. These values are stored in a netmap. The netmap is associated with a policy and an adapter.

Before you begin this procedure, create a policy to associate with the netmap.

To create a netmap and define inbound and outbound nodes:

1. Click Configuration from the menu bar.
2. Click Actions > New Netmap > FTP Netmap.
3. Type a Netmap Name.
4. To define an inbound node definition, click the Inbound Nodes tab and click New.
5. Specify the following values:
   - ◆ Inbound Node Name
   - ◆ Peer Address Pattern
   - ◆ Policy

   Note: If you have not defined a policy, click the green plus sign to define one.

6. Click OK.

7. To define an outbound node definition, click the Outbound Nodes tab and click New.

8. Specify the following values:

   ◆ Outbound Node Name

   ◆ Primary Destination Address

   ◆ Primary Destination Port

9. Click OK.

10. Click Save.

Define the FTP Adapter Used for the Connection

An FTP adapter definition specifies system-level communications information necessary for FTP connections to and from SSP. You can create multiple adapter definitions.

Before you begin this procedure, create the following definitions:

✦ A netmap to associate with the adapter

✦ An engine definition to associate with the adapter. Refer to the *Sterling Secure Proxy Installation Guide* for instructions.

To define an FTP adapter:

1. If necessary, click Configuration from the menu bar.

1. Click Actions > New Adapter > FTP Reverse Proxy.

2. Specify values for the following:

   ◆ Adapter Name

   ◆ Listen Port

   ◆ Netmap

   ◆ Standard Routing Node

   ◆ Engine

3. Click Save.

What You Defined with the Basic FTP Configuration Scenario

Creating secure connections to GIS servers on behalf of nodes external to your trusted zone requires that you organize information about the trading partners and the GIS server in a policy, a netmap, and an adapter definition. You created these items when you defined the Basic FTP Configuration. The next step is testing the configuration prior to configuring additional security features. Before you test the configuration, be sure that:

✦ The GIS server has an active FTP server adapter configured to listen for the port specified in the outbound node definition.

✦ The user ID and password provided by the inbound node is defined at the GIS server.

Refer to *Test the Inbound and Outbound FTP Connections* on page 143 for information about testing the FTP Reverse Proxy configurations outlined in this scenario.

As you add complexity to your security configurations using the procedures in the remaining scenarios, you modify the basic configuration to configure more complex authentication and certificate validation measures.

Variations on the Basic FTP Configuration

After you confirm that the communications sessions you established using the Basic FTP configuration were successful, you may want to validate sessions using other types of inbound trading partner definitions before you add complexity to the security configuration. To ensure that you can validate and troubleshoot problems, you should test one variation at a time by changing the configuration, initiating a connection, and verifying the result.

Inbound FTP Trading Partner Node Definitions

You can modify the inbound trading partner node definitions as follows:

✦ Define a specific IP address

✦ Define a wildcard peer pattern

✦ Define an IP/subnet pattern

Define Connection Requirements Between SSP and Inbound FTP Nodes

You define connection requirements between SSP and inbound nodes by defining inbound node definitions. Refer to your company security requirements to determine how tightly to define the parameters an inbound node must provide to allow a connection.

You can define inbound node definitions to allow only one individual inbound connection, or you can identify IP address patterns and create an inbound definition that allows inbound connections that match the pattern to connect to SSP. Methods of defining inbound nodes are as follows:

✦ Create an entry for an individual inbound node and define the inbound node IP address that can connect to SSP. Only connections from that IP address are allowed. A single IP address must be specified as a subnet pattern where all bits are matched, such as 11.22.33.44/32. SSP also supports individual host names. They must match the value returned by a reverse DNS lookup.

✦ Define an inbound node entry that allows all nodes that match an IP/subnet address pattern. Patterns include:

Matching the first 16 bits of an IP address pattern. For example, 10.20.0.0/16 allows all IP addresses that begin with 10.20.* to connect to SSP.

Matching the first 8 bits of an IP address pattern. For example, 10.0.0.0/8 allows all IP addresses that begin with 10.* to connect to SSP.

✦ Define an inbound node entry that allows all inbound nodes that match a wildcard host name pattern. When a connection is attempted and you have defined a wildcard host name pattern definition, a reverse DNS lookup is performed on the IP address of the inbound connection. The DNS name is compared to the wildcard pattern. Wildcard patterns include:

* matches any number of characters before or after a period. For example, *.a.com allows a connection from b.a.com but not from a.bc.com. A single * allows all inbound nodes to successfully connect to SSP.

? matches one character. For example, a.?.com allows a connection from a.b.com but not from a.bc.com.

You can define more than one inbound node definition and use a combination of the node definition methods. However, be sure to order the definitions from most specific to least specific. When an inbound node connection is attempted, SSP compares the IP address of the inbound node to the first inbound node definition. If it matches, a connection is established. If it does not match, SSP checks the next inbound node definition until a match is found. If no match is found, the connection is terminated.

Inbound FTP Connection Definition Worksheet

Use the following worksheet to identify the information needed to configure inbound node definitions for a specific inbound node or for groups of inbound nodes that match a pattern.

Configuration Manager Field	Define Inbound Trading Partner Information	Value
Note: If you define a single node and definitions for multiple nodes using pattern matching, ensure that you order the definitions from most specific to least specific, because SSP processes them in the order in which they are listed.		
Inbound Node Name	Trading partner name.	_____
Policy Name	Policy to associate with the inbound trading partner.	_____
For a Single Node		
Peer Address Pattern	IP address Create an entry for an individual inbound node and define the inbound node IP address that can connect to SSP. Only connections from that IP address will be allowed. SSP supports host name. An example definition is a.b.com. A single IP address must be specified as a subnet pattern where all bits are matched, such as 11.22.33.44/32.	_____

Configuration Manager Field	Define Inbound Trading Partner Information	Value
For Multiple IP Addresses Using IP/Subnet Pattern		
Peer Address Pattern	Peer Address IP/Subnet Pattern Options.	
		_____
For Multiple Nodes Using Wildcard Peer Address Pattern to Validate Inbound DNS		
Peer Address Pattern	Wildcard Peer Address Pattern.	
		_____

Define Inbound Node Connection Definitions for an FTP Connection

This procedure instructs you how to modify the basic FTP configuration to add inbound node definitions for 1) a group of nodes with similar information, or 2) that limit access to one specific inbound node. It assumes that you have already configured an adapter. Gather a list of all inbound trading partners, including names and IP addresses.

To define inbound connection definitions:

1. Identify patterns that can be used to define groups of inbound nodes.
2. To increase security, you need to define a trading partner connection for any individual IP address.
3. If necessary, click Configuration from the menu bar.
4. Expand the Netmaps tree and click the netmap to modify.
5. Click New to add a new inbound node definition.
6. Using the information you defined on the Inbound Connection Definition Worksheet, provide the following information, and click Save:
    * Inbound Node Name
    * Peer Address Pattern
    * Policy
7. Repeat step 6 for every group of connections and every individual IP address connection you want to define.
8. If necessary, reorder the node definitions in the netmap. Order definitions from most specific to least specific because they will be evaluated in order.
    a. Click the radio button beside the inbound node definition to move.
    b. Click Move Up or Move Down until the node definition is in the correct order.
9. Click Save.

Add SSL/TLS Support for an FTP Connection

This scenario builds on the Basic FTP Configuration by enabling security for the inbound and outbound nodes you defined in the netmap. Following is a diagram to illustrate the addition of SSL or TLS to the inbound and outbound node connections.
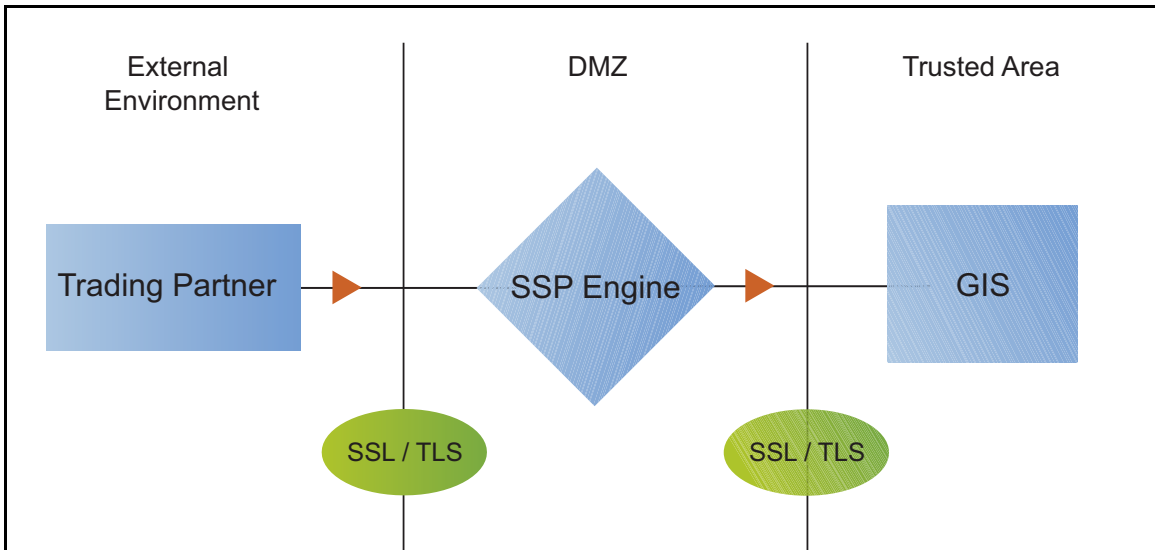
To add SSL/TLS support to the netmap for the inbound and outbound nodes, select the following options for the connections:

✦ Protocol

✦ Cipher suites

✦ Stores and certificates

To effectively configure and test this scenario:

1. Add SSL/TLS support to the inbound node definition first and establish a session initiated by an FTP client to a GIS server.

2. Then, add SSL/TLS support to the outbound node definition and establish a session initiated by an FTP client to a GIS server.

> **Note:** Before you configure SSL or TLS support, you must check in your certificates. Refer to Chapter 3, *Manage Certificates for SSL/TLS Transactions with Trading Partners*.

SSL/TLS Support Worksheet

Before you add SSL/TLS support to the connection information you created in the Basic FTP Configuration scenario, gather the information on the SSL/TLS Support Worksheet. You use this information as you configure the inbound and outbound nodes for SSL/TLS support.

Inbound Connection for FTP

Select the security setting and cipher suites to be used to secure the connection. To configure client authentication, enable this option. Select the key/system certificate to use to validate the connection.

Configuration Manager	Feature	Value
Inbound Node Name	Name of inbound node to add security to.	Select an inbound node definition from the list.
Security Setting	Security protocol to use.	(SSL v3 or TLS, SSL v2 or v3 with v3 Hello, SSL (any version) or TLS, SSL v2 or v3, TLS, or SSL v3)
Enable Client Authentication	Do you want to require that the inbound connection present its certificate for SSL or TLS client authentication?	_____ (Yes or No)
Trust Store	If client authentication is enabled, identify the trust store used to verify the client certificate.	_____
CA Certificates/Trusted Root	Name of CA certificate/trusted root (if client authentication is enabled).	_____
Key Store	The location where the keys and system certificates you want to use are stored.	_____
Key/System Certificate	Name of SSP system certificate presented to the inbound connection during the handshake.	_____
Available Cipher Suites Selected Cipher Suites	Select the ciphers to enable by moving them from the Available Ciphers to the Selected Ciphers field.	_____ _____ _____ _____

Outbound Connection for FTP

Select the security setting and cipher suites to be used to secure the outbound connection. Select the key/system certificate to use to validate the connection.

Configuration Manager Field	Feature	Value
Outbound Node Name	Name of outbound node to add security to.	Select a node definition from the list

Configuration Manager Field	Feature	Value
Security Setting	Security protocol to use.	_____ (SSL v3 or TLS, SSL v2 or v3 with v3 Hello, SSL (any version) or TLS, SSL v2 or v3, TLS, or SSL v3)
Trust Store	If client authentication is enabled, identify the trust store where the certificate is stored.	_____
CA Certificates/Trusted Root	Identify the certificate to use to secure the outbound connection.	_____
Key Store	The location where the keys and system certificates you want to use are stored.	_____
Key/System Certificate	System certificate used to validate the GIS server.	_____
Available Cipher Suites Selected Cipher Suites	Cipher suites to enable.	_____ _____ _____ _____

Secure the Inbound FTP Connection Using the TLS or SSL Protocol

The first step in strengthening security is to secure the communications channel. This procedure describes how to enable the TLS or SSL protocol for the inbound connection to authenticate SSP to the trading partner initiating the connection. To require that SSP authenticate the inbound node, enable client authentication.

Before you can configure this option, you must obtain the necessary certificates and place them in the SSP certificate store.

To enable the TLS or SSL protocol on the inbound FTP node:

1. If necessary, click Configuration from the menu bar.
2. Expand the Netmaps tree and select a netmap to modify.
3. Click the Inbound Nodes tab.
4. Select an inbound node to modify, and click Edit.
5. Click the Security tab, and then click Secure Connection to enable security.
6. Select values for the following:
   - ◆ Security Setting
   - ◆ Key Store
   - ◆ Key/System Certificate

- ◆ Available Ciphers
- ◆ Selected Ciphers

7. To enable client authentication:

   a. Click Enable Client Authentication.

   b. Select the Trust Store where the certificate you want to use is located.

   c. Select the CA Certificates/Trusted Root to use to authenticate the certificate presented by the inbound node.

   > **Note:** Be sure to highlight the certificate to select it. If only one certificate is displayed in the field, it is not selected until you highlight it.

8. Click OK.

9. Click Save.

Variations on the SSL/TLS Configuration on the Inbound FTP Node

After you confirm that the communications sessions you established using the basic FTP configuration with SSL/TLS enabled on the inbound node were successful, you may want to enable a clear control channel.

Enable a Clear Control Channel for an Inbound FTP Node Connection

If your environment requires that a firewall be able to see the flow of FTP commands and responses, enable the clear control channel option. Enabling clear control channel for the inbound node requires that the inbound FTP client send the clear control channel command and switch the control channel to an unencrypted channel after user authentication is completed.

To enable a clear control channel for an inbound node:

1. If necessary, click Configuration from the menu bar.

2. Expand the Netmaps tree and select a netmap to modify.

3. Click the Inbound Nodes tab and select the Inbound Node to modify.

4. Click Edit.

5. Click the Security tab.

6. Enable Clear Control Channel.

7. Click OK.

8. Click Save.

Secure the Outbound FTP Connection Using the TLS or SSL Protocol

If the GIS server has enabled the use of SSL or TLS to secure the connection, you must enable the TLS or SSL protocol in the SSP outbound node configuration. This procedure describes how to enable the TLS or SSL protocol to authenticate the GIS server to SSP when establishing an outbound connection.

Before you can configure this option, you must obtain the necessary certificates and place them in the SSP certificate store.

To enable the TLS or SSL protocol:

1. If necessary, click Configuration from the menu bar.

2. Expand the Netmaps tree and select a netmap to modify.

3. Click the Outbound Nodes tab.

4. Select an outbound node to modify, and click Edit.

5. Click the Security tab, and then click Secure Connection to enable security.

6. Select the following security options for the node:

   ◆ Security Setting

   ◆ Trust Store

   ◆ CA Certificates/Trusted Root

   > **Note:** Be sure to highlight the certificate to select it. If only one certificate is displayed in the field, it is not selected until you highlight it.

   ◆ Key Store

   ◆ Key/System Certificate

   ◆ Available Ciphers Suites

   ◆ Selected Ciphers Suites

7. Click OK.

8. Click Save.

Variations on the Add SSL/TLS Support on the Outbound Node

After you confirm that the communications session you established using the Add SSL/TLS Support scenario was successful, you may want to further modify your inbound and outbound nodes. To ensure that you can validate and troubleshoot problems, you should test one variation at a time by changing the configuration, initiating a connection, and verifying the result.

The following variation applies to this configuration:

> **Note:** You must obtain the necessary certificates and place them in the SSP certificate store before you can configure these options.

✦ Create your own trust store and key store

✦ Enable a clear control channel for an outbound connection

Enable a Clear Control Channel for an Outbound FTP Node Connection

If your environment requires that a firewall be able to see the flow of FTP commands and responses, enable the clear control channel option. If clear control channel is enabled on the outbound node, the FTP reverse proxy adapter sends the clear control channel command and switches the command channel to an unencrypted channel, after user authentication is completed.

To enable a clear control channel for an outbound node:

1. If necessary, click Configuration from the menu bar.
2. Expand the Netmaps tree and select a netmap to modify.
3. Click the Outbound Nodes tab and select the Outbound Node to modify.
4. Click Edit.
5. Click the Security tab.
6. Enable Clear Control Channel.
7. Click OK.
8. Click Save.

Add Local User Authentication to the Inbound FTP Connection

This scenario builds on the Basic FTP Configuration by adding local user authentication to the inbound connection using information defined in the local user store. The user ID and password presented by the inbound node are authenticated against the information stored in the local user store. The values must match before a connection is established. You must add this information to

the local user store before you can test this scenario. Following is an illustration of the secure features supported in this scenario:
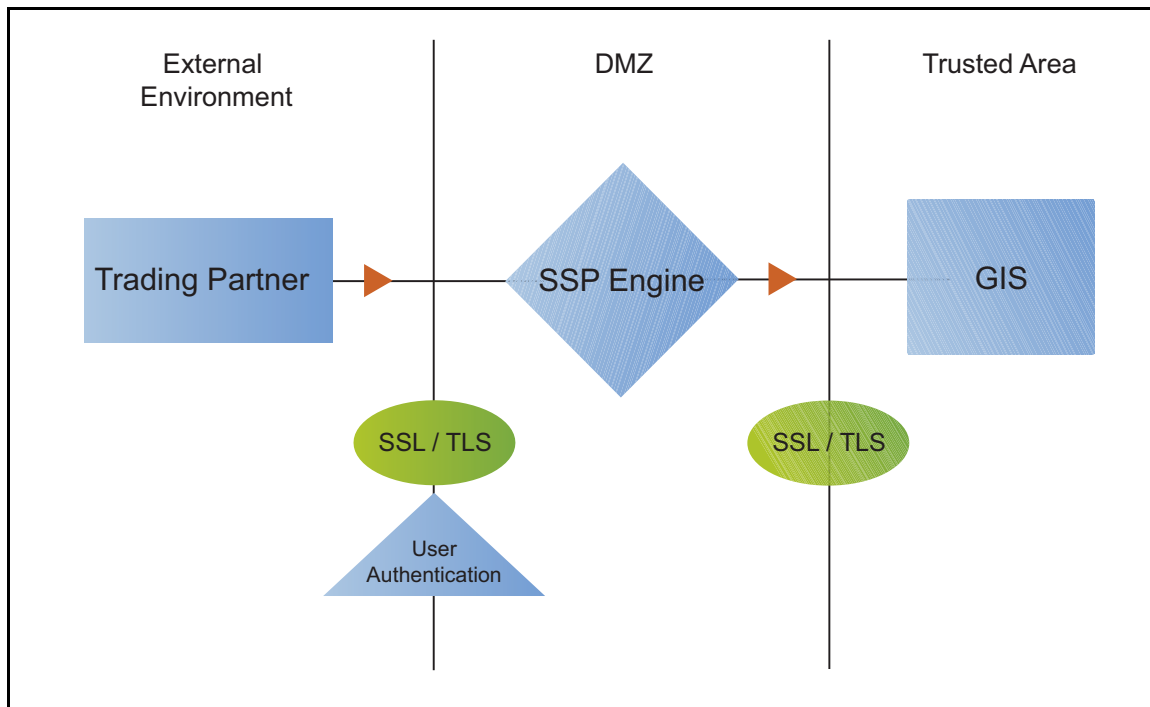
Adding user authentication to the inbound connection defined in the Basic FTP Configuration involves enabling user authentication and specifying information about the trading partner.

After you configure user authentication using the local user store information, validate the configuration by establishing a session initiated by an FTP client to a GIS server.

FTP Inbound Connection (Local User Authentication) - Worksheet

Before you add user authentication to the inbound connection you created in the Basic FTP Configuration scenario, gather the information on the FTP Inbound Connection (Local User Authentication) - Worksheet. Use this information as you configure user authentication for the inbound connection.

In this scenario, you edit the policy you created in the FTP Basic Configuration scenario and enable user authentication. You also add a user ID and password for the trading partner to the default user store.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy associated with the inbound node.	_____
User Authentication	Method to use to authenticate the inbound node.	Through Local User Store

Configuration Manager Field	Feature	Value
User Store	Name of the user store you create.	_____
User Name	Name of the user you define in the User Store.	_____
Password
Confirm Password	The password value to use to validate the inbound connection.	_____

Add Local User Authentication to the FTP Inbound Connection

You can strengthen the security of inbound connections by enabling user authentication. This procedure describes how to add user information to the local user store to be validated by the engine during an inbound FTP client connection.

> **Note:** Check the netmap to ensure that the policy you select is associated with the inbound nodes you want to authenticate.

To add user authentication for an inbound connection:

1. If necessary, click Configuration from the menu bar.
2. Expand the Policies tree and select a policy to modify.
3. Click the Advanced tab.
4. Enable the User Authentication Through Local User Store option.
5. Click OK.
6. Click Save.

Add Credentials to the Local User Store

If you enable user authentication through the local user store, you have to add user information to the local user store for validation by SSP during an inbound FTP client connection.

Before you begin this procedure:

✦ Enable user authentication for the inbound connection.

✦ Ensure that the engine is configured to use the user store containing the user credentials.

To add user information to the local user store:

1. Click Credentials from the menu bar.
2. Expand the User Stores tree and select a user store to modify.
3. From the User Store Configuration panel, click New.

4. Specify values for the following:

    ◆ User Name

    ◆ Password

    ◆ Confirm Password

5. Click Save.

Provide GIS Credentials to the Outbound FTP Node Using the Netmap

This scenario builds on the Basic FTP Configuration by enabling the use of user credentials from the netmap to connect to the outbound GIS connection. Following is an illustration of the security features supported in this scenario:
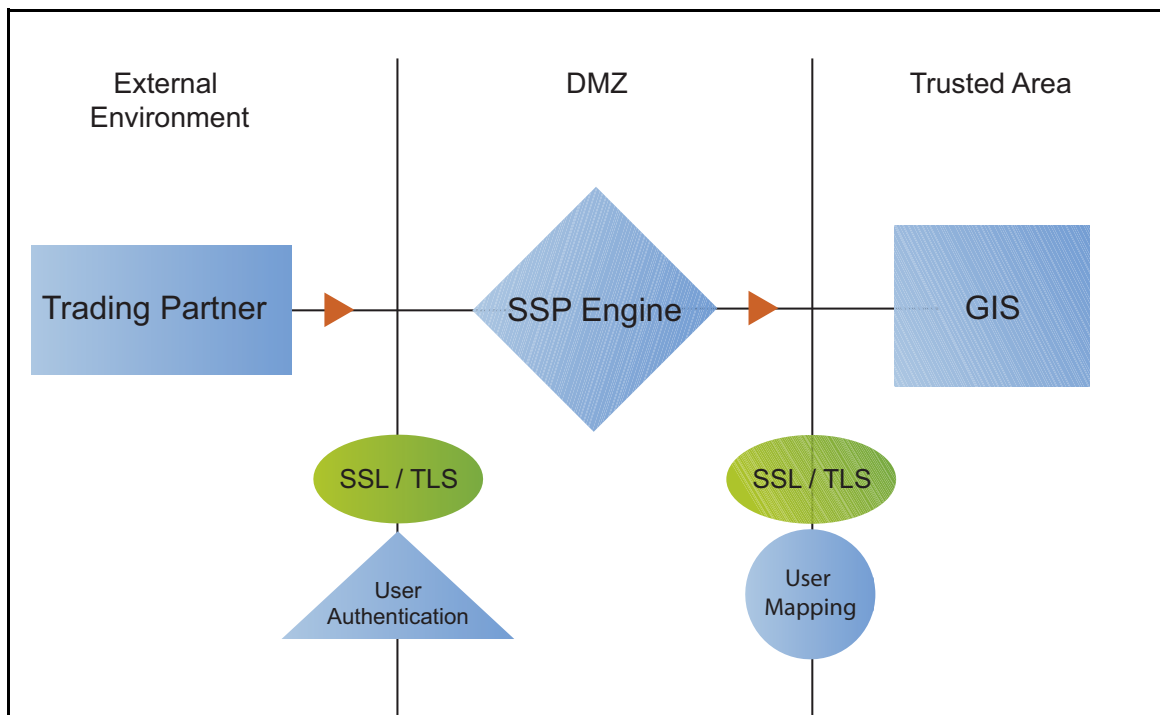
When an inbound trading partner connects to SSP, its credentials are replaced with credentials stored in the netmap. The replacement credentials are then used to connect to the outbound FTP server. This method uses SSP security features to prevent trading partners from knowing the credentials used to connect to the outbound GIS server. The outbound GIS server must have a user definition that accepts the user ID and password provided.

After you configure the environment to use credentials defined in the netmap, test the configuration by establishing a session initiated by an FTP client to a GIS server. Refer to *Test the Inbound and Outbound FTP Connections* on page 143 for more information on testing the configuration described in this scenario.

Provide Credentials for the Outbound FTP Node Using the Netmap Worksheet

In this scenario, edit the netmap and policy you created in the Basic FTP Configuration to provide user credentials stored in SSP to connect to the outbound GIS connection.

Collect the following information so you can match the SSP configuration with the GIS server configuration. Use the information on this worksheet as you edit the outbound node definition, and be sure to select the netmap and policy you created in the Basic FTP Configuration.

Configuration Manager Field	Feature	Value
User ID	User ID used to connect to the GIS server. (Must also be defined at the GIS server.)	_____
Password	Password to connect to the GIS server. (Must also be defined at the GIS server.)	_____

Connect to the Outbound FTP Server Using Credentials from the Netmap

To increase security for connections to the server in the trusted zone, you can use the netmap to store the user ID and password to connect to the outbound GIS server. If you configure this option, the inbound node uses one set of credentials to connect to SSP and SSP uses information stored in the netmap to connect to the outbound FTP server.

Before you configure this option:

✦ Ensure the user ID and password are defined on the GIS server.

✦ Obtain the user ID and password.

To configure validation for the outbound connection using credentials stored in the netmap:

1. If necessary, click Configuration from the menu bar.

2. Expand the Netmaps tree and select the FTP netmap to modify.

3. Click the Outbound Nodes tab.

4. Select the outbound node to modify and click Edit.

5. Click the Advanced tab.

6. Type values in the following fields for connecting to the GIS server:

   ◆ User ID

   ◆ Password

7. Click Save.

8. Expand the Policies tree and select the policy to modify.

9. On the FTP Policy Configuration panel, click the Advanced tab.

10. From the User Mapping: Internal User ID list, select From Netmap.

11. Click Save.

Test the configuration to ensure that this feature is working.

Strengthen Authentication of an FTP Node Using EA

Use EA to provide a more advanced method of securing the inbound or the outbound connection, such as, authenticating certificate information or user credentials presented by the inbound node, or performing user ID and password mapping for the internal credentials. The following illustrates the security features enabled in this scenario:
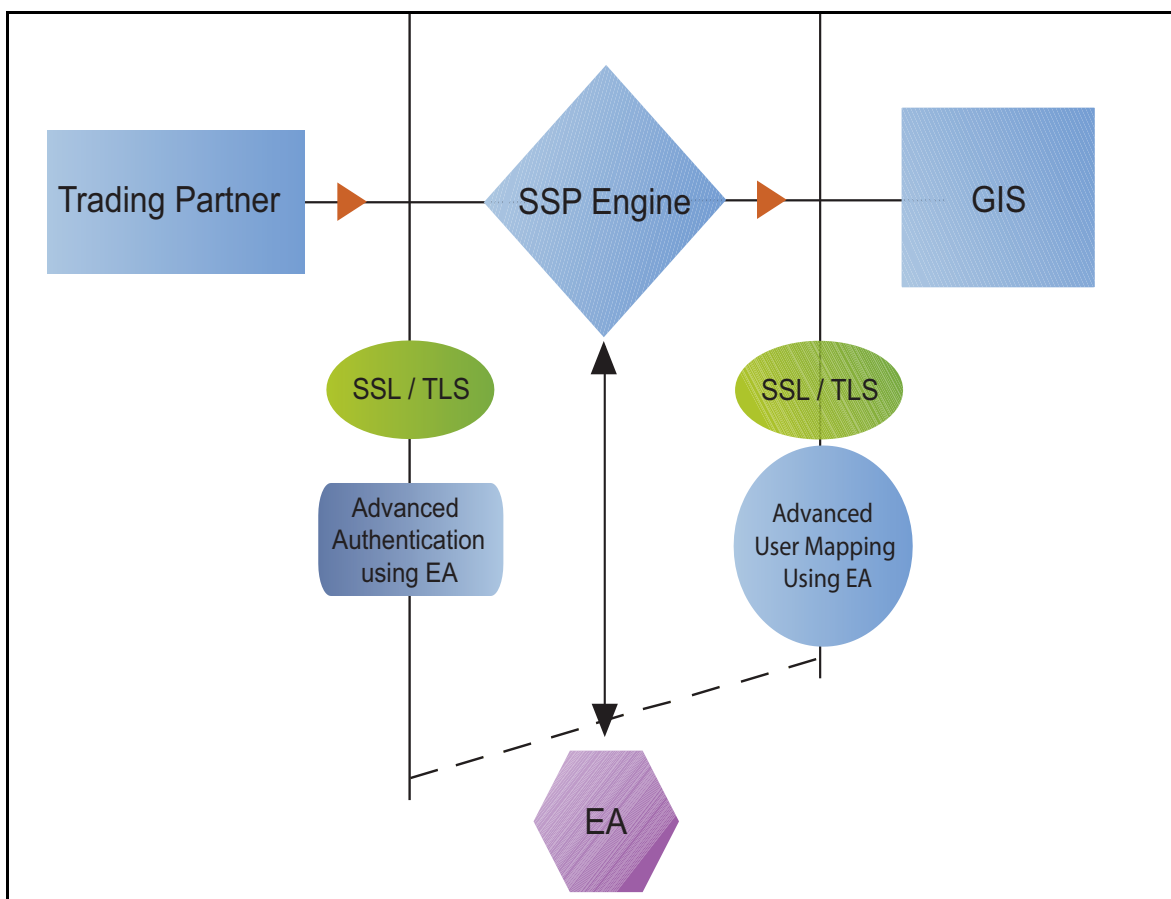
Authenticate an Inbound FTP Certificate or User Using EA

You can authenticate an inbound connection against information stored in an LDAP database by configuring EA to define how the connection is authenticated. Following are some of the functions that EA can perform:

✦ Validate certificates, including dates and signatures

✦ Verify the presence of X.509 v3 extensions

✦ Enforce minimum key length requirements

✦ Check certificates against certificate revocation lists (CRLs)

✦ Perform LDAP queries

The EA definition determines the options that are enabled.

Manage Connection Requirements to the Outbound FTP Server Using EA

For a higher level of security when connecting to the outbound server, use information stored in an LDAP database. To use information in an LDAP database, you configure EA. You can use EA to map a user ID and password provided by an inbound connection to a unique user ID and password that is not exposed to the external node.

Authenticate an Inbound FTP Certificate or User Using EA Worksheet

Use the following worksheet to identify the information needed to authenticate a trading partner using information in EA. Update the policy you created in the Basic FTP Configuration for this scenario.

Configuration Manager Field	Information	Value
Certificate Authentication - External Authentication Certificate Validation	Will you validate the inbound certificate?	_____ (Yes or No)
Certificate Authentication - External Authentication Profile	If yes, identify the EA certificate validation definition.	_____
User Authentication - Through External Authentication	Will you validate user information?	_____ (Yes or No)
User Authentication - External Authentication Profile	If yes, identify the EA user validation definition.	_____

Authenticate the Inbound FTP Node Using EA

To authenticate certificate information or user information about the inbound node against information stored in an external database, you must configure EA. After you configure EA to enable certificate validation or user authentication, use this procedure to configure SSP to use the authentication method you defined in EA.

Before you configure SSP to use EA to authenticate an inbound node authentication, obtain the name of the EA definition.

In addition, ensure that the following procedures have been performed:

✦ The policy associated with the inbound node has enabled client authentication.

✦ The public keys for SSP have been sent to the EA server and imported into the EA key store.

✦ The EA server connection has been configured in SSP.

To configure authentication of an inbound node using EA:

1. If necessary, click Configuration from the menu bar.

2. Expand the Policies tree and select a policy to modify.

3. On the Policy Configuration panel, click the Advanced tab.

4. Configure one or more of the following options:

   ◆ To validate the certificate presented by the inbound node against information defined in EA, enable Certificate Authentication - External Authentication Certificate Validation and identify the name of the profile you defined in EA in the Certificate Authentication - External Authentication Profile field.

   ◆ To validate the user, enable Through External Authentication and identify the name of the profile defined in EA in the External Authentication Profile field.

5. Click Save.

You can now associate this policy with the inbound node on which you want to perform user authentication using information stored in an LDAP server.

Connect to Outbound FTP Server Using EA Worksheet

Use this worksheet to configure a stronger outbound connection using information from an LDAP database.

Configuration Manager Field	Feature	Value
User Certification Through External Authentication	Will you validate user information?	Yes
External Authentication Profile	If yes, identify the EA user validation definition	_____

Connect to the Outbound Node Using Information Stored in EA

If you store user credentials in an external database accessed by EA, use this procedure to configure SSP to use these credentials to connect to the secure outbound server.

Before you configure this option:

✦ Configure a user validation definition in EA.

✦ Obtain the name of the EA definition.

✦ Configure the EA server to allow connections from SSP.

✦ Ensure that the policy associated with the inbound node has enabled client authentication.

✦ Ensure that the public keys for SSP have been sent to the EA server and imported into the EA key store.

To configure the use of credentials from EA:

1. If necessary, click Configuration from the menu bar.

2. Expand the Policies tree and select a policy to modify.

3. On the Policy Configuration panel, click the Advanced tab.

4. Enable the User Authentication Through External Authentication option.

5. Type the name of the definition you defined in EA in the External Authentication Profile field.

6. Deselect the Local User Store option.

7. From the Internal User ID field, select From External Authentication.

8. Click Save.

Test the Inbound and Outbound FTP Connections

To verify that the engine can receive and initiate communication sessions, you have to establish a connection between an FTP client and the engine, initiate a session from the engine to the GIS server in the trusted zone, and review the SSP audit log for the results.

Note: Make sure the engine is running when you configure an FTP adapter. If it is running, configuration files are automatically copied to the engine when you save any update. Configuration files must be available at the engine for communication sessions to be established.

This procedure enables you to verify that the engine can:

✦ Establish an FTP session initiated by a trading partner using an FTP client

✦ Initiate an outbound session to a GIS server on behalf of the FTP client connection

To verify the communications sessions:

1. Make sure the engine is running.

2. Initiate an FTP client session to the GIS server in your trusted zone.

3. View the Inbound Node Log and the Outbound Node Log.

4. Confirm that the data transfer was successful, as illustrated in the sample log below:

Sample Inbound Node Log

```
11 Sep 2009 11:38:28,914 [ProxyNearScheduler-Thread-2] INFO
sys.SESSION_NODE.HTTP_Netmap_Any - protocol=http SID=1
SNAME=sherwood.csg.stercomm.com SIP=10.20.246.121 SPORT=40134 SSP104I
Session: 1 - Session Proceeding after Node match: Any
11 Sep 2009 11:38:31,557 [ProxyFarScheduler-Thread-4] INFO
sys.SESSION_NODE.HTTP_Netmap_Any - protocol=http SID=1
SNAME=sherwood.csg.stercomm.com SIP=10.20.246.121 SPORT=40134
DNAME=lunar.csg.stercomm.com DIP=10.20.246.42 DPORT=10054 SUID=admin
DUID=admin SSP102I Session: 1 - Control:ServerAgent Connection closed
(CloseCode.EOF): Elapsed Time: 2.13 (s)
: Bytes Received: 194 [at: 7.286384976525821E-4 MBPS]
Bytes Sent: 20480595 [at: 76.92242253521127 MBPS]
```

Sample Outbound Node Log

```
11 Sep 2009 11:38:28,914 [ProxyNearScheduler-Thread-2] INFO
sys.SESSION_NODE.HTTP_Netmap_Any - protocol=http SID=1
SNAME=sherwood.csg.stercomm.com SIP=10.20.246.121 SPORT=40134 SSP104I
Session: 1 - Session Proceeding after Node match: Any

11 Sep 2009 11:38:31,557 [ProxyFarScheduler-Thread-4] INFO
sys.SESSION_NODE.HTTP_Netmap_Any - protocol=http SID=1
SNAME=sherwood.csg.stercomm.com SIP=10.20.246.121 SPORT=40134
DNAME=lunar.csg.stercomm.com DIP=10.20.246.42 DPORT=10054 SUID=admin
DUID=admin SSP102I Session: 1 - Control:ServerAgent Connection closed
(CloseCode.EOF): Elapsed Time: 2.13 (s)
: Bytes Received: 194 [at: 7.286384976525821E-4 MBPS]
Bytes Sent: 20480595 [at: 76.92242253521127 MBPS]
```

If your session was unsuccessful, review the log information to determine the likely cause of the failure and the corrective action to take.

Additional FTP Configuration Options

Additional FTP configuration options are available for the following features:

✦ Route an Outbound FTP Connection to Alternate GIS Servers

✦ Define a Passive Data Outbound Port Range for an FTP Reverse Proxy Adapter

✦ Define a Passive NAT Address for an FTP Reverse Proxy Adapter

✦ Define an Active Data Outbound Port Range for an FTP Reverse Proxy Adapter

✦ Use IP address from PASV response for outbound data connections

Route an Outbound FTP Connection to Alternate GIS Servers

When you configured the adapter, you identified the GIS server to connect to by selecting one of the outbound node connections defined in the netmap. For each outbound node definition, you can identify up to three alternate outbound nodes to connect to if the primary GIS server is not available.

Two methods of configuring alternate GIS server routing are available.

✦ Select a GIS server from the drop-down list. Using this method, you first configure an outbound node definition in the netmap for each alternate GIS server you want to use. Each connection uses the security and External Authentication settings defined in the outbound node definition.

✦ Select IP address/port from the drop-down list and enter values for the IP address and port. If you use this method, you do not have to define the alternate outbound nodes in the netmap, and each alternate connection uses the security and External Authentication settings defined in the primary node definition.

If you configure alternate GIS server definitions in the outbound node definition, when a connection to the primary outbound node is unsuccessful SSP tries to connect to the alternate node you defined as Node 1. If the connection to the first alternate node is unsuccessful, SSP tries to connect to the second alternate node, Node 2. If this connection is unsuccessful, SSP tries to connect to the third alternate, Node 3. If the connection to this node is unsuccessful, the inbound connection fails.

To configure alternate outbound connections:

1. If necessary, click Configuration from the menu bar.
2. Expand the Netmaps tree and select a netmap to modify.
3. Click the Outbound Node tab and select the node to modify.
4. Click the Advanced tab.
5. Do one of the following:
   ◆ To identify an alternate node that is defined in the netmap and use the security settings defined in the alternate node definition, select the outbound node name from the drop-down list.
   ◆ To configure an alternate node that is not in the netmap and use the security settings defined in the primary node definition:
     a. Select Address/Port from the drop-down list in the Alternate Destinations Node field.
     b. Provide the IP address and port number for the alternate outbound node.
6. Click OK.
7. Click Save.

Define a Passive Data Outbound Port Range for an FTP Reverse Proxy Adapter

Two modes can be used to open the FTP data channel: active mode and passive mode. The mode used on the inbound connection is determined by the client. SSP always uses passive mode for the outbound connection.

In active mode, the inbound FTP node sends a port command identifying the data channel listen port on which SSP needs to connect. You identify the port numbers to use for an active data connection in the Active Data Outbound Port Range field.

In passive mode, the FTP client sends a PASV command and SSP starts a listener to receive the data connections from the client. You identify a port number range to use to start the listener that receives connections.

To define a passive data outbound port range:

1. If necessary, click Configuration from the menu bar.

2. Expand the Adapters tree and select the adapter to modify.

3. Click the Advanced tab.

4. Type a value to use for the passive data outbound port range in the Passive Data Listening Port Range field.

5. Click Save.

Define a Passive NAT Address for an FTP Reverse Proxy Adapter

When a PASV command is sent to SSP from an inbound FTP client, the host and port number to which the inbound FTP client needs to connect for the data channel is returned. When SSP is behind a firewall, the host address of SSP is not visible to the inbound FTP client. To ensure that the client can obtain this information, define the passive NAT address.

Define this value if the client cannot directly connect to the proxy, such as when using a static network address translation (NAT).

If you are using a remote external perimeter server with the FTP reverse proxy adapter and the perimeter server is also behind a firewall using static network address translation, identify the name or IP address of the computer running the external perimeter server.

To define a passive NAT address:

1. If necessary, click Configuration from the menu bar.

2. Expand the Adapters tree and select the adapter to modify.

3. Click the Advanced tab.

4. Type a value to use for the passive NAT address in the Passive NAT Address field.

5. Click Save.

Define an Active Data Outbound Port Range for an FTP Reverse Proxy Adapter

Two modes are available to open the FTP data channel: active mode and passive mode. The mode used on the inbound connection is determined by the client. SSP always uses passive mode for the outbound connection.

In active mode, the inbound FTP node sends a port command identifying the data channel listen port on which the proxy needs to connect. You identify the port numbers to use for an active data connection in the Active Data Outbound Port Range field.

In passive mode, the FTP client sends a PASV command to SSP and SSP starts a listener to receive the data connections from the client. You identify a port number range that can be used to start the listener to receive connections.

To define an active data outbound port range for an FTP reverse proxy adapter:

1. If necessary, click Configuration from the menu bar.

2. Expand the Adapters tree and select the adapter to modify.

3. Click the Advanced tab.

4. Type a value to use for the active data outbound port range in the Active Data Outbound Port Range field.

5. Click Save.

Use IP Address from a PASV Response For Outbound Data Connections

As a security measure, SSP ignores PASV response and uses the same IP address as the initial control channel for all data channel connections. If VIPI (virtual IP address) is used by the outbound server, the data connections may be a different IP address than the original connection. Complete this procedure to allow data channel connections to use different IP addresses.

To allow SSP to use an IP address from a PASV response:

To define an active data outbound port range for an FTP reverse proxy adapter:

1. If necessary, click Configuration from the menu bar.

2. Expand the Adapters tree and select the adapter to modify.

3. Click the Advanced tab.

4. Enable the field Use IP from PASV Response.

5. Click Save.

HTTP Reverse Proxy Configuration

The HTTP configuration scenarios describe how to configure HTTP protocol connections to and from the engine.

Note:	Make sure the engine is running when you configure an HTTP adapter. If it is running, configuration information is transmitted to the engine when you save it. Configuration information must be available on the engine before communication sessions with Gentran Integration Suite (GIS) can be established.

Organization of the HTTP Configuration Scenarios

The first scenario instructs you on how to configure a basic configuration. Each successive scenario adds an additional security feature to the basic configuration. After configuring each scenario, test the connection to ensure that you have correctly configured it. You determine your security needs and configure the security features applicable for your environment.

The following scenarios help you configure and test SSP for HTTP protocol connections to the GIS server:

✦ Create a basic HTTP configuration

✦ Add SSL/TLS support

✦ Perform user authentication using the local user store

✦ Provide outbound credentials using the netmap

The remaining configuration scenarios require Sterling External Authentication Server (EA), an optional security feature of SSP that must be configured independently of SSP. After EA is configured, you can update your basic security definitions to enable SSP to connect to the EA to enforce the following advanced security features:

✦ Authenticate an inbound certificate or user using EA

✦ Manage connection requirements to the outbound server using EA

Additional procedures are provided to instruct you on how to configure the following features:

✦ Block common exploits

✦ Rewrite URLs in HTML content to route inbound connections through SSP

✦ Define alternate nodes for failover support

Complete Scenario Worksheets

Before you begin configuring SSP for each HTTP connection scenario, gather the information on the worksheet provided with the scenario. You use this information as you configure each feature. Complete worksheets as follows:

✦ Provide a value for each SSP feature listed. Fields listed in the worksheet are required.

✦ Accept default values for fields not listed in the worksheet.

✦ Note the Configuration Manager field(s) where you will specify the value.

Complete and Test HTTP Configuration Scenarios

Work through the sequence of HTTP configuration scenarios in the order they are presented to add additional security features. Be sure to test each feature before you add the next feature to the configuration. Before you move SSP into production, ensure that you have configured and tested all of the security features you need for your environment.

Note: As you complete each task, provide all required information. If information is not provided or is incorrect, the following error icon is displayed: 🔴 To view more information about the error, hover over the icon.

Create a Basic HTTP Configuration

This scenario contains all the information and tools you need to configure SSP to establish a basic connection from a trading partner to the GIS server as shown in the following diagram. You accept default values when configuring this scenario. As a result, no authentication occurs in SSP and credentials presented by the inbound node are passed through to the GIS server.
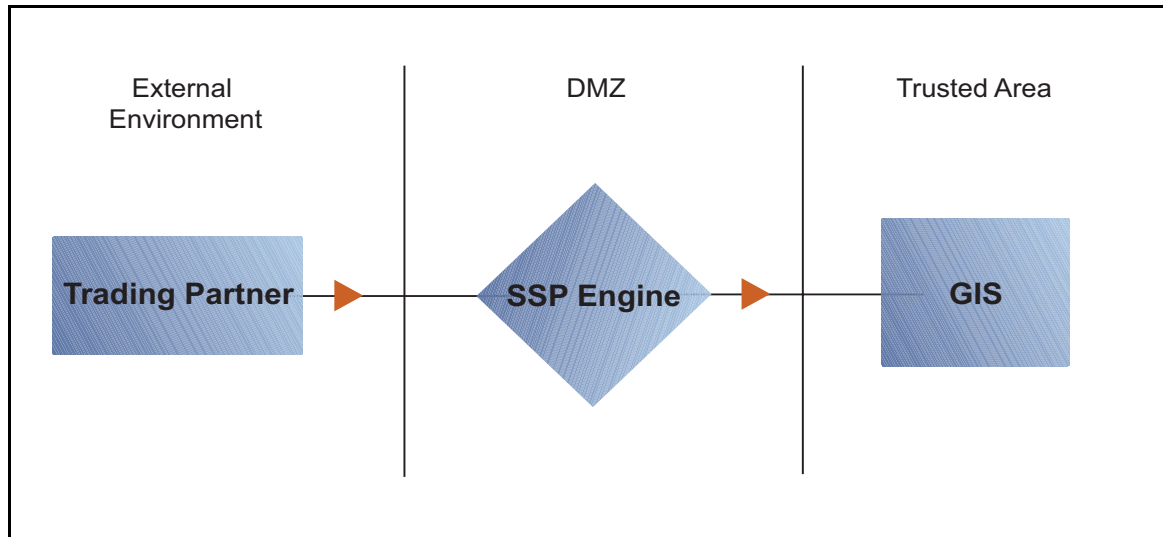
After you configure SSP, validate the configuration by initiating an HTTP connection from the trading partner. For more information on testing the configuration, see *Test the Inbound and Outbound HTTP Connections* on page 171.

Complete the following tasks to define a basic HTTP configuration:

✦ Create a policy

✦ Define inbound and outbound connections in a netmap

✦ Define an HTTP adapter

Basic HTTP Configuration Worksheet

Before you configure SSP for HTTP connections, gather the information on the Basic HTTP Configuration Worksheet. You use this information as you configure a basic HTTP connection for SSP. After you configure SSP for HTTP connections, validate the configuration by initiating an HTTP connection from the inbound node.

HTTP Policy

Create a basic policy. In a later HTTP configuration scenario, you edit this policy to add security features.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy.	_____

HTTP Netmap (Inbound and Outbound Connections)

Create a netmap that contains connection information for the nodes connecting to and from SSP: the trading partner (inbound node) and the GIS server (outbound node). You will also associate the basic security policy you create with the inbound node.

Configuration Manager Field	Feature	Value
Netmap Name	Netmap name.	_____
Inbound Trading Partner Information		
Inbound Node Name	Trading partner name (name to assign to inbound node definition).	_____
Peer Address Pattern	Host name or IP address pattern.	* (Specifying * for this value allows all inbound nodes configured on the GIS server as trading partners to connect to the GIS server. Use this value for testing purposes. To create a more specific node definition, see *Define Inbound HTTP Node Connection Definitions* on page 157.)
Policy	Name of policy you create.	This value is selected from a pull-down list.
Outbound GIS Server Connection		
Node Name	Outbound GIS server node name.	_____
Primary Destination Address	Host name or IP address to connect to the outbound GIS server.	_____
Primary Destination Port	Port number to connect to the outbound GIS server.	_____

HTTP Adapter

Create an HTTP adapter that defines information necessary to establish HTTP connections to and from SSP. When you are configuring the adapter, select the basic netmap and the outbound GIS server you define in the netmap definition.

Configuration Manager Field	Feature	Value
Adapter Name	Adapter name.	_____
Listen Port	Listen port to use for inbound connections.	_____
Netmap	Netmap to associate with the adapter.	_____
Standard Routing Node	Name of the outbound node corresponding to the GIS server where inbound connections are routed.	_____
Engine	Engine to run on.	_____

Create an HTTP Policy

The HTTP policy defines how you impose controls to authenticate a trading partner trying to access a GIS server over the public Internet.

To define a policy:

1. If necessary, click Configuration from the menu bar.
2. Click Actions > New Policy > HTTP Policy.
3. Type a Policy Name.
4. Click Save.

Create an HTTP Netmap

You define inbound connection information for your external trading partners and outbound connection information for the GIS server that SSP connects to. These values are stored in a netmap. The netmap is associated with a policy and an adapter.

Before you begin this procedure, create a policy to associate with the netmap.

To create a netmap and define inbound and outbound nodes:

1. If necessary, click Configuration from the menu bar.
2. Click Actions > New Netmap > HTTP Netmap.
3. Type a Netmap Name.

4. To define an inbound node definition, click the Inbound Nodes tab and click New.

5. Specify the following values:

    ◆   Inbound Node Name

    ◆   Peer Address Pattern

    ◆   Policy

    > **Note:**   If you have not defined a policy, click the green plus sign to define one.

6. Click OK.

7. To define an outbound node definition, click the Outbound Nodes tab and click New.

8. Specify the following values:

    ◆   Outbound Node Name

    ◆   Primary Destination Address

    ◆   Primary Destination Port

9. Click OK.

10. Click Save.

Define the HTTP Adapter Used for the Connection

An HTTP adapter definition specifies system-level communications information necessary for HTTP connections to and from SSP. You can create multiple adapter definitions.

Before you begin this procedure, create the following definitions:

✦   A netmap to associate with the adapter.

✦   An engine definition to associate with the adapter. Refer to the *Sterling Secure Proxy Installation Guide* for instructions.

To define an HTTP adapter:

1. If necessary, click Configuration from the menu bar.

2. Click Actions > New Adapter > HTTP Reverse Proxy.

3. Specify values for the following:

    ◆   Adapter Name

    ◆   Listen Port

    ◆   Netmap

    ◆   Standard Routing Node

    ◆   Engine

4. Click Save.

What You Defined with the Basic HTTP Configuration Scenario

Creating connections to GIS servers on behalf of nodes external to your trusted zone requires that you organize information about the trading partners and the GIS server in a policy, a netmap, and an adapter definition. You created these items when you defined the Basic HTTP Configuration. The next step is testing the configuration prior to configuring additional security features. Before you test the configuration, be sure that:

✦ The GIS server has an active HTTP server adapter configured to listen for the port specified in the outbound node definition

✦ The user ID and password provided by the inbound node are defined at the GIS server

Refer to *Test the Inbound and Outbound HTTP Connections* on page 171 for information about testing the HTTP Reverse Proxy Configurations outlined in this scenario.

As you add complexity to your security configurations using the procedures in the remaining scenarios, you modify the basic configuration to configure more complex authentication and certificate validation measures.

Variations on the Basic HTTP Configuration

After you confirm that the communications sessions you established using the basic HTTP configuration were successful, you may want to validate sessions using other types of inbound trading partner definitions before you add complexity to the security configuration. To ensure that you can validate and troubleshoot problems, you should test one variation at a time by changing the configuration, initiating a connection, and verifying the result.

Inbound HTTP Trading Partner Node Definitions

You can modify the inbound trading partner node definitions as follows:

✦ Define a specific IP address

✦ Define a wildcard peer pattern

✦ Define an IP/subnet pattern

Define HTTP Connection Requirements Between SSP and Inbound Nodes

You define connection requirements between SSP and inbound nodes by creating inbound node definitions. Refer to your company security requirements to determine how tightly to define what parameters an inbound node must provide to allow a connection.

You can define an inbound node definition as generic or as specific as your security environment requires. In the strictest environment, you define a specific node definition that allows only one individual inbound connection to use the definition. In an environment where you trust in the inbound node connections, you can identify a pattern of IP addresses and create an inbound definition that allows all inbound connections that match the pattern to connect to SSP.

Methods of defining inbound nodes include:

✦ Create an entry for an individual inbound node and define the inbound node IP address that can connect to SSP. Only connections from that IP address will be allowed. A single IP address must be specified as a subnet pattern where all bits are matched, such as 11.22.33.44/32. SSP also supports individual host names. They must match the value returned by a reverse DNS lookup.

✦ Define an inbound node entry that allows all nodes that match an IP/Subnet address pattern. Patterns include:

Match the first 16 bits of an IP address pattern. For example, 10.20.0.0/16 allows all IP addresses that begin with 10.20.* to connect to SSP.

Match the first 8 bits of an IP address pattern. For example, 10.0.0.0/8 allows all IP addresses that begin with 10.* to connect to SSP.

✦ Define an inbound node entry that allows all inbound nodes that match a wildcard host name pattern. When a connection is attempted and you have defined a wildcard host name pattern definition, a reverse DNS lookup is performed on the IP address of the inbound connection. The DNS name is compared to the wildcard pattern. Wildcard patterns include:

* enables a match on any number of characters. For example, *.a.com allows a connection from b.a.com but not from a.bc.com. * allows all inbound nodes to successfully connect to SSP.

? enables a match on one character. For example, a.?.com allows a connection from a.b.com but not from a.bc.com.

You can define more than one inbound node definition and use a combination of the node definition methods. However, be sure to order the definitions from most specific to least specific. When an inbound node connection is attempted, SSP compares the IP address of the inbound node to the first inbound node definition. If it matches, a connection is established. If it does not match, SSP checks the next inbound node definition until a match is found. If no match is found, the connection is terminated.

Inbound HTTP Connection Definition Worksheet

Use the following worksheet to identify the information needed to configure inbound node definitions for groups of inbound nodes that match a pattern or for specific inbound nodes.

Configuration Manager Field	Define Inbound Trading Partner Information	Value
Note: If you define a single node and definitions for multiple nodes using pattern matching, ensure that you order the definitions from most specific to least specific because SSP processes them in the order in which they are listed.		
Inbound Node Name	Trading Partner Name.	_____
Policy Name	Policy to associate with the inbound trading partner.	_____

Configuration Manager Field	Define Inbound Trading Partner Information	Value
For a Single Node		
Peer Address Pattern	IP address	
	Create an entry for an individual inbound node and define the inbound node IP address that can connect to SSP. Only connections from that IP address will be allowed. SSP supports host name. An example definition is a.b.com.	———————————
	A single IP address must be specified as a subnet pattern where all bits are matched, such as 11.22.33.44/32.	
For Multiple IP Addresses Using IP/Subnet Pattern		
Peer Address Pattern	Peer Address IP/Subnet Pattern Options:	
	◆ Match first 16 bits of IP address with pattern, for example,10.20.0.0/16 matches 10.20.*	———————————
	◆ Match first 8 bits of IP address with pattern, for example, 10.0.0.0/8 matches 10.*	
For Multiple Nodes Using Wildcard Peer Address Pattern to Validate Inbound DNS		
Peer Address Pattern	Wildcard Peer Address Pattern:	
	◆ * enables a match on any number of characters, for example, *.a.com matches b.a.com but not a.b.com	———————————
	◆ ? enables a match on any one character, for example, a.?.com matches a.b.com but not a.bc.com	

Define Inbound HTTP Node Connection Definitions

This procedure instructs you how to modify the basic HTTP configuration to add inbound node definitions for a group of nodes with similar information, and definitions that limit access to one specific inbound node. It assumes that you have already configured an adapter. Gather a list of all inbound trading partners, including names and IP addresses.

To define inbound connection definitions:

1. Identify patterns that can be used to define groups of inbound nodes.
2. Decide if you need to define a trading partner connection for any individual IP addresses.
3. If necessary, click Configuration from the menu bar.
4. Expand the Netmaps tree and select the netmap to modify.
5. Click New to add a new inbound node definition.

6. Using the information you defined on the Inbound Connection Definition Worksheet, provide the following information and click Save:

    ◆   Inbound Node Name

    ◆   Peer Address Pattern

    ◆   Policy

7. Repeat step 6 for every group of connections and for every individual IP address connection you want to define.

8. If necessary, reorder the node definitions in the netmap. Order definitions from most specific to least specific since they will be evaluated in order.

    a.   Click the radio button beside the inbound node definition to move.

    b.   Click Move Up or Move Down until the node definition is in the correct order.

9. Click Save.

Establish a session initiated by an HTTP client to a GIS server to test the configuration.

Add SSL/TLS Support for an HTTP Connection

This scenario builds on the Basic HTTP Configuration by enabling security for the inbound and outbound nodes you defined in the netmap. Following is a diagram to illustrate the addition of SSL or TLS to the inbound and the outbound node connections.
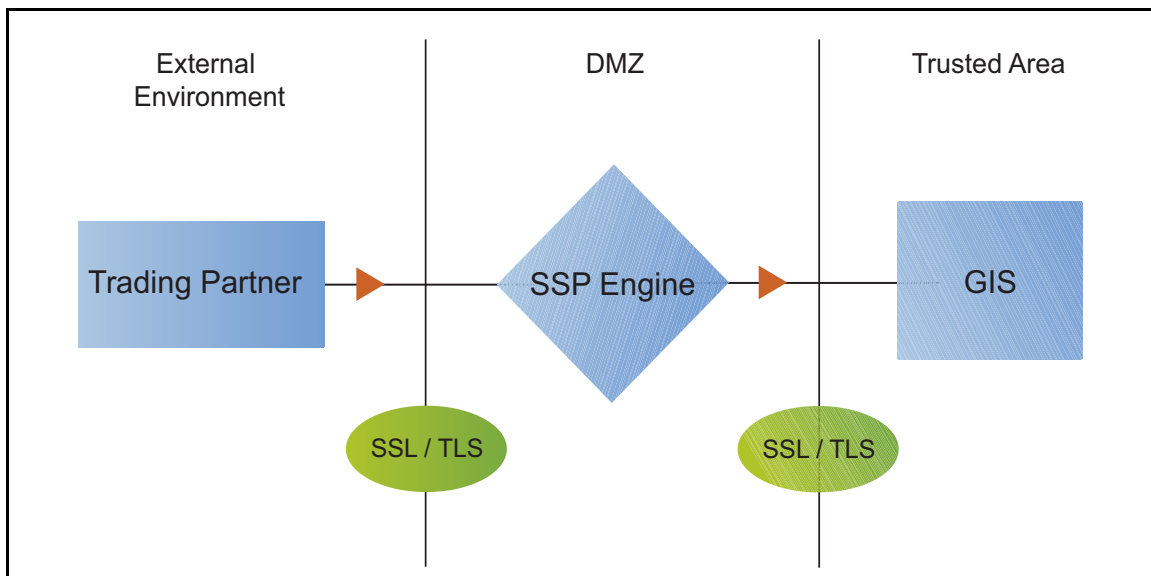
Note: Before you configure SSL or TLS support, you must check in your certificates. Refer to Chapter 3, *Manage Certificates for SSL/TLS Transactions with Trading Partners*.

To add SSL/TLS support to the netmap for the inbound and outbound nodes, define the following options for the connections:

✦ Protocol

✦ Cipher suites

✦ Stores and certificates

To effectively configure and test this scenario:

1. Add SSL/TLS support to the inbound node definition first and establish a session initiated by an HTTP client to a GIS server.

2. Then, add SSL/TLS support to the outbound node definition and establish a session initiated by an HTTP client to a GIS server.

SSL/TLS Support for HTTP Worksheet

Before you add SSL/TLS support to the connection information you created in the Basic HTTP Configuration scenario, gather the information on the SSL/TLS Support Worksheet. You use this information as you configure the inbound and outbound nodes for SSL/TLS support.

Secure Inbound HTTP Connection

Select the security setting and cipher suites to be used to secure the connection. To configure client authentication, enable this option. Select the key/system certificate to use to validate the connection.

Configuration Manager	Feature	Value
Inbound Node Name	Name of inbound node to add security to.	Select an inbound node definition from the list
Security Setting	Security protocol to use.	_____ (SSL v3 or TLS, SSL v2 or v3 with v3 Hello, SSL (any version) or TLS, SSL v2 or v3, TLS, or SSL v3)
Enable Client Authentication	Do you want to require that the inbound connection present its certificate for SSL or TLS client authentication?	_____ (Yes or No)
Trust Store	If client authentication is enabled, identify the trust store where the certificate is stored.	_____
CA Certificates/Trusted Root	Name of CA certificate/trusted root (if client authentication is enabled).	_____
Key Store	The database where the keys and system certificates you want to use are stored.	_____

Configuration Manager	Feature	Value
Key/System Certificate	Name of SSP system certificate presented to the inbound connection during the handshake.	_____
Available Cipher Suites Selected Cipher Suites	Select the ciphers to enable by moving them from the Available Ciphers to the Selected Ciphers field.	_____ _____ _____ _____

Secure Outbound HTTP Connection

Select the security setting and cipher suites to be used to secure the connection. Select the trusted certificate to use to validate the server certificate. If the server requires client authentication, you must specify a server certificate. If the server requires client authentication, you specify a key/system certificate.

Configuration Manager Field	Feature	Value
Outbound Node Name	Name of outbound node to add security to.	Select a node definition from the list.
Security Setting	Security protocol to use.	_____ (SSL v3 or TLS, SSL v2 or v3 with v3 Hello, SSL (any version) or TLS, SSL v2 or v3, TLS, or SSL v3)
Trust Store	The trust store where the certificate is stored.	_____
CA Certificates/Trusted Root	Identify the certificate to use to secure the outbound connection.	_____
Key Store	Key store where the Key/System Certificate is stored.	_____
Key/System Certificate	System certificate used to validate the GIS server.	_____
Available Ciphers Selected Ciphers	Cipher suites to enable.	_____ _____ _____ _____

Secure the Inbound HTTP Connection Using the SSL or TLS Protocol

The first step in strengthening security is to secure the communications channel. This procedure describes how to enable the TLS or SSL protocol for the inbound connection to authenticate SSP to the trading partner initiating the connection. To require that SSP authenticate the inbound node, enable client authentication.

Before you can configure this option, you must obtain the necessary certificates and place them in the SSP Cert Stores.

To enable the TLS or SSL protocol:

1. If necessary, click Configuration from the menu bar.
2. Expand the Netmaps tree and select an HTTP netmap to modify.
3. Click the Inbound Nodes tab.
4. Select an inbound node to modify, and click Edit.
5. Click the Security tab, and then click Secure Connection to enable security.
6. Select values for the following:
    - ◆ Security Setting
    - ◆ Key Store
    - ◆ Key/System Certificate
    - ◆ Available Cipher Suites
    - ◆ Selected Cipher Suites
7. To enable client authentication:
    a. Click Enable Client Authentication.
    b. Select the trust store where the CA certificate or trusted root certificate is stored.
    c. Select the CA Certificates/Trusted Root certificate to use.

Note:	Be sure to highlight the certificate to select it. If only one certificate is displayed in the field, it is not selected until you highlight it.

8. Click OK.
9. Click Save.

Establish a session initiated by an HTTP client to a GIS server to test the configuration.

Secure the Outbound HTTP Connection Using the SSL or TLS Protocol

If the GIS server has enabled the use of SSL or TLS to secure the connection, you must enable TLS or SSL protocol in the SSP outbound node configuration. This procedure describes how to enable the TLS or SSL protocol to authenticate the GIS server to SSP when establishing an outbound connection.

Before you can configure this option, you must obtain the necessary certificates and place them in the SSP cert stores.

To enable the TLS or SSL protocol:

1. If necessary, click Configuration from the menu bar.

2. Expand the Netmaps tree and select an HTTP netmap to modify.

3. Click the Outbound Nodes tab.

4. Select an outbound node to modify, and click Edit.

5. Click the Security tab, and then click Secure Connection to enable security.

6. Select the following security options for the node:

    ◆   Security Setting

    ◆   Trust Store

    ◆   CA Certificate/Trusted Root

    > **Note:**   Be sure to highlight the certificate to select it. If only one certificate is displayed in the field, it is not selected until you highlight it.

    ◆   Available Ciphers

    ◆   Selected Ciphers

7. If the GIS server requires client authentication, select the key store and key/system certificate to present to the GIS server during the SSL/TLS handshake.

8. Click OK.

9. Click Save.

Establish a session initiated by an HTTP client to a GIS server to test the configuration.

Add Local User Authentication to the HTTP Connection

This scenario builds on the Basic HTTP Configuration by adding user authentication to the inbound connection using information defined in the local user store. Following is an illustration of the security options enabled for this scenario:
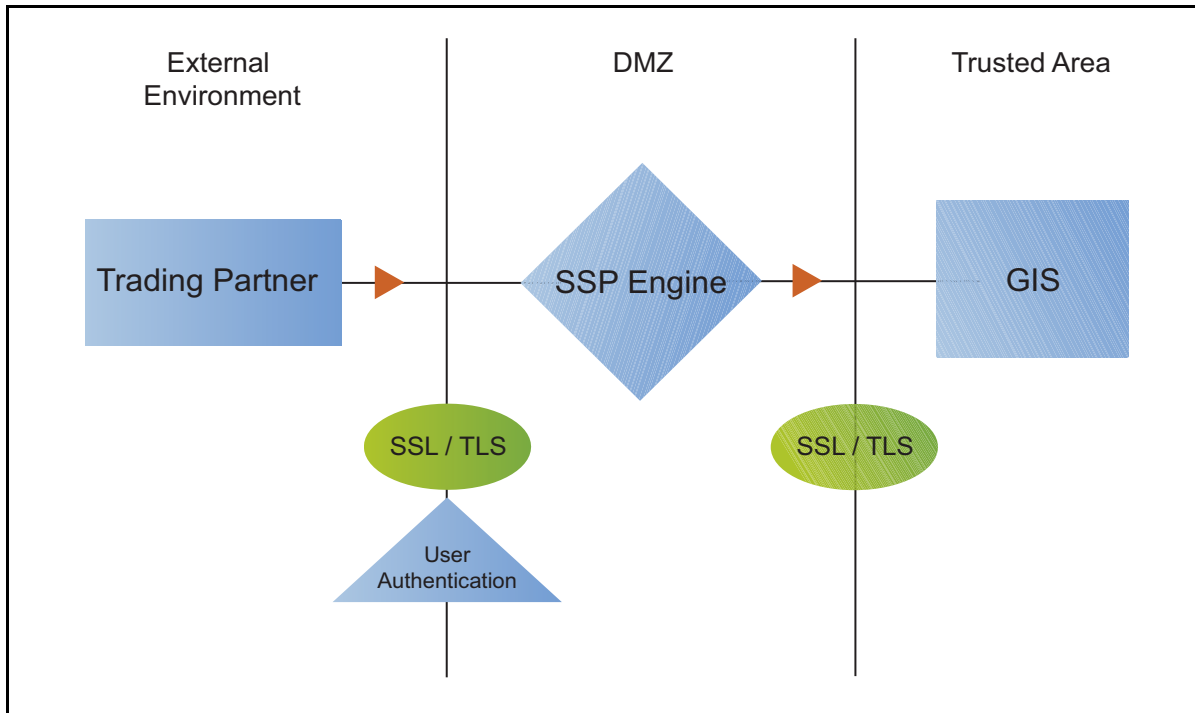
The user ID and password presented by the inbound node are authenticated against the information stored in the local user store. The values must match before a connection is established. You must add this information to the local user store before you can test this scenario.

Adding user authentication to the inbound connection defined in the Basic HTTP Configuration involves enabling user authentication and specifying information about the trading partner.

After you configure user authentication using the local user store information, validate the configuration by establishing a session initiated by an HTTP client to a GIS server.

HTTP Inbound Connection (Local User Authentication) Worksheet

Before you add user authentication to the inbound connection you created in the Basic HTTP Configuration scenario, gather the information on the HTTP Inbound Connection (Local User Authentication) Worksheet. Use this information as you configure user authentication for the inbound connection.

In this scenario, you edit the policy you created in the HTTP Basic Configuration scenario and enable user authentication. You also add a user ID and password for the trading partner to the default user store.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy associated with the inbound node.	_____
User Authentication	Method to use to authenticate the inbound node.	Through local user store
User Store	Name of the user store you create.	_____
User Name	Name of the user you define in the User Store.	_____
Password Confirm Password	The password value to use to validate the inbound connection.	_____

Enable Local User Authentication to an HTTP Inbound Connection

You can strengthen the security of inbound connections by enabling local user authentication. This procedure describes how to configure the use of the local user store to validate an inbound connection.

> **Note:**   Check the netmap to ensure that the policy you edit is associated with the inbound nodes you want to authenticate.

To add user authentication for an inbound connection:

1. If necessary, click Configuration from the menu bar.
2. Expand the Policies tree and select the policy you created in the basic configuration.
3. Click the Advanced tab.
4. Enable the User Authentication Through Local User Store option.
5. Click Save.

Add Credentials to the Local User Store for an HTTP Connection

If you enable user authentication through the local user store, you have to add user information to the local user store to be validated by SSP during an inbound HTTP client connection.

Before you begin this procedure:

✦ Enable user authentication for the inbound connection.

✦ Ensure that the engine is configured to use the user store that contains the user credentials.

To add user information to the local user store:

1. Click Credentials from the menu bar.
2. Expand the User Stores tree and select a user store to modify.
3. From the User Store Configuration panel, click New.
4. Specify values for the following:
    ◆ User Name
    ◆ Password
    ◆ Confirm Password
5. Click OK.
6. Click Save.

Establish a session initiated by an HTTP client to a GIS server to test the configuration.

Provide Credentials to the Outbound HTTP Node Using the Netmap

This scenario builds on the Basic HTTP Configuration by enabling the use of user credentials from the netmap to connect to the outbound GIS connection. Following is an illustration of the security features supported in this scenario:
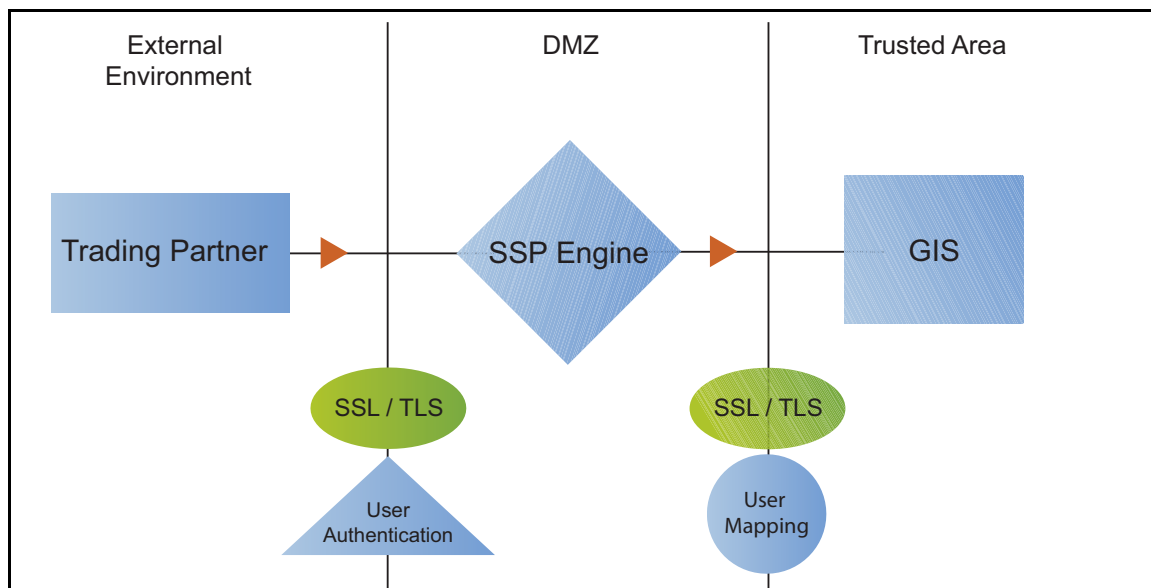
If you configure user mapping using the netmap, an inbound trading partner connects to SSP and provides one set of credentials. Its credentials are replaced with credentials stored in the netmap. The replacement credentials are then used to connect to the outbound secure server. This method uses SSP security features to prevent trading partners from knowing the credentials used to connect to the outbound GIS server. The outbound GIS server must have a user definition that accepts the user ID and password provided.

After you configure the environment to use credentials defined in the netmap, test the configuration by establishing a session initiated by an HTTP client to a GIS server. Refer to *Test the Inbound and Outbound HTTP Connections* on page 171 for more information on testing the configuration described in this scenario.

Connect to the Outbound HTTP Server Using Credentials from the Netmap Worksheet

In this scenario, edit the netmap and the policy you created in the Basic HTTP Configuration to provide user credentials stored in SSP to connect to the outbound GIS connection.

Collect the following information so you can match the SSP configuration with the GIS server configuration. Use the information on this worksheet as you edit the outbound node definition, and be sure to select the netmap and policy you created in the Basic HTTP Configuration.

Configuration Manager Field	Feature	Value
User ID	User ID used to connect to the GIS server. (Must also be defined at the GIS server)	_____
Password	Password to connect to the GIS server. (Must also be defined at the GIS server)	_____

Configure Name and Password to Connect to the Outbound HTTP Server in the Netmap

To increase security for connections to the server in the trusted zone, you can use the netmap to store the user ID and password to connect to the outbound GIS server. If you configure this option, the inbound node uses one set of credentials to connect to SSP and SSP uses information stored in the netmap to connect to the outbound HTTP server.

Before you configure this option:

✦ Ensure the user ID and password are defined on the GIS server

✦ Obtain the user ID and password

To configure validation for the outbound connection using credentials stored in the netmap:

1. If necessary, click Configuration from the menu bar.

2. Expand the Netmaps tree and select an HTTP netmap to modify.

3. Click the Outbound Nodes tab.

4. Select the outbound node to modify and click Edit.

5. Click the Advanced tab.

6. Type the following values to be used to connect to the GIS server:

   ◆ User ID

   ◆ Password

7. Click OK.

8. Click Save.

9. Expand the Policies tree and select the policy to modify.

10. On the Policy Configuration panel, click the Advanced tab.

11. From the User Mapping: Internal User ID list, select From Netmap.

12. Click Save.

Test the configuration to ensure that the updated configuration is working.

Strengthen Authentication for an HTTP Connection Using EA

To provide a more advanced method of securing the inbound or the outbound connection, use EA. Use EA to authenticate certificate information or user credentials presented by the inbound node or to perform user ID and password mapping for the internal credentials. The following illustrates the security features enabled in this scenario.
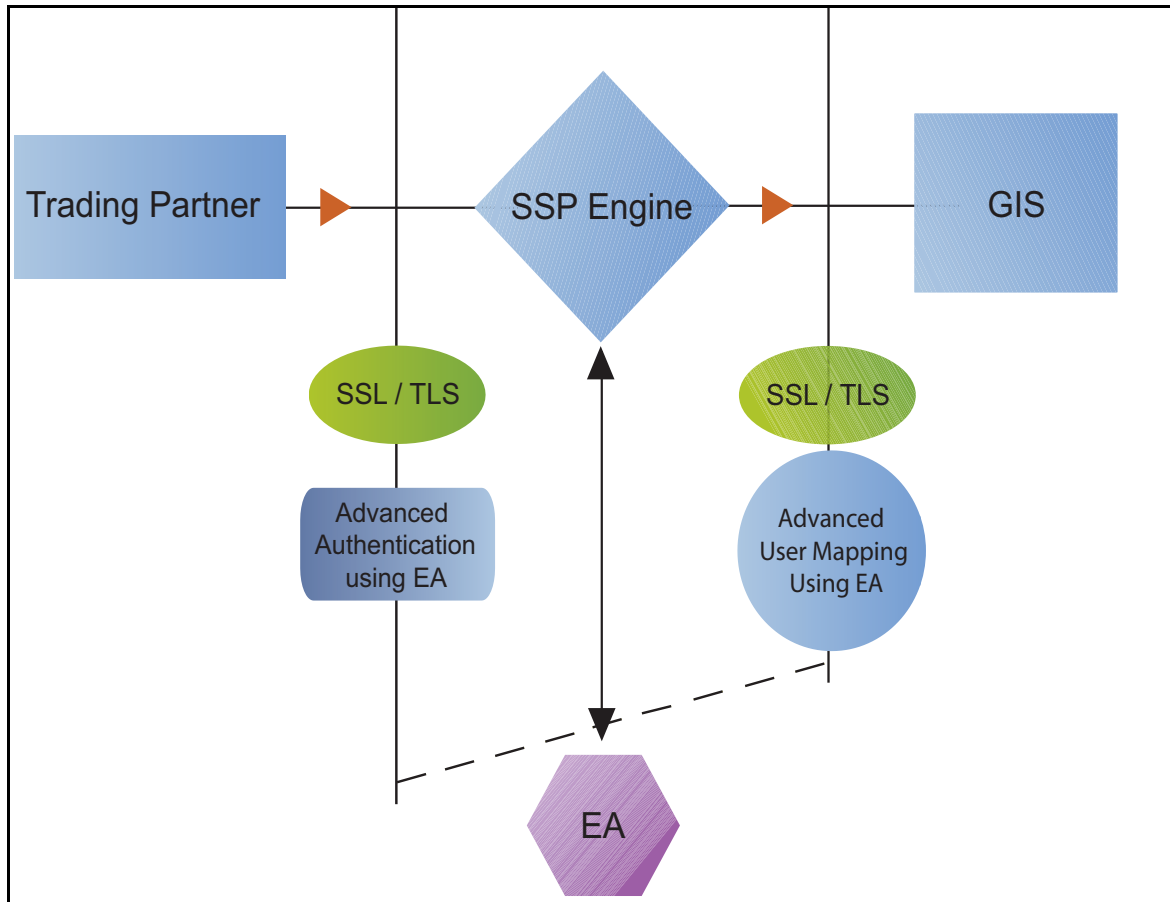
Authenticate an Inbound HTTP Certificate or User Using EA

You can authenticate an inbound connection against information stored in an LDAP database by configuring EA to define how the connection is authenticated. Following are some of the options EA can perform:

✦ Validate certificates, including dates and signatures

✦ Verify the presence of X.509 v3 extensions

✦ Enforce minimum key length requirements

✦ Check certificates against certificate revocation lists (CRLs)

✦ Perform LDAP queries

The EA definition determines which options are enabled.

Manage Connection Requirements to the Outbound HTTP Server Using EA

For a higher level of security when connecting to the outbound server, use information stored in an LDAP database to connect to the outbound server. To use information in an LDAP database, you configure EA. EA can map a user ID and password provided by an inbound connection to a user ID and password that is not exposed to the external node.

Authenticate an Inbound HTTP Certificate or User Using EA Worksheet

Use the following worksheet to specify the information needed to authenticate a trading partner with information in EA. Update the policy you created in the Basic HTTP Configuration for this scenario.

Configuration Manager Field	Information	Value
Certificate Authentication - External Authentication Certificate Validation	Will you validate the inbound certificate?	_____ (Yes or No)
Certificate Authentication - External Authentication Profile	If yes, identify the EA certificate validation definition.	_____
User Authentication - Through External Authentication	Will you validate user information?	_____ (Yes or No)
User Authentication - External Authentication Profile	If yes, identify the EA user validation definition.	_____

Authenticate the Inbound HTTP Node Using EA

To authenticate certificate information or user information about the inbound node against information stored in an LDAP database, you must configure EA. After you configure EA to enable certificate validation or user authentication, use this procedure to configure SSP to use the authentication method you defined in EA.

Before you configure SSP to use EA to authenticate an inbound node, obtain the name of the EA definition.

In addition, ensure that the following procedures have been performed:

✦ The policy associated with the inbound node has enabled client authentication.

✦ The public keys for SSP have been sent to the EA server and imported into the EA keystore.

✦ The EA server connection has been configured in SSP. Refer to Chapter 12, *Configure SSP for Sterling External Authentication Server (EA)*.

To configure authentication of an inbound node using EA:

1. If necessary, click Configuration from the menu bar.

2. Expand the Policies tree and select a policy to modify.

3. On the HTTP Policy Configuration panel, click the Advanced tab.

4. To validate the certificate presented by the inbound node against information defined in EA, enable Certificate Authentication - External Authentication Certificate Validation and identify the name of the profile you defined in EA in the Certificate Authentication - External Authentication Profile field.

5. To validate a user from EA:

    a. Enable User Authentication Through External Authentication field.

    b. Type the name of the definition you defined in EA in the User Authentication External Authentication Profile field.

    c. Deselect the Through Local User Store option.

    d. Select From External Authentication in the User Mapping:Internal User ID field.

6. Click Save.

You can now associate this policy with the inbound node on which you want to perform user authentication using EA.

Connect to the Outbound HTTP Server Using EA Worksheet

Use this worksheet to identify information required to configure a stronger outbound connection using information in an LDAP database:

Configuration Manager Field	Feature	Value
User Certification Through External Authentication	Will you validate user information against LDAP?	Yes
External Authentication Profile	If yes, identify the EA user validation definition.	_____

Connect to the Outbound HTTP Server Using Information Stored in LDAP

If you store user credentials in an LDAP database, use this procedure to configure SSP to use these credentials to connect to the secure outbound server.

Before you configure this option:

✦ Configure a definition in EA and obtain the name of the EA definition.

✦ Configure the EA server to allow connections from SSP.

✦ Ensure that the policy associated with the inbound node has enabled client authentication.

✦ Ensure that the public keys for SSP have been sent to the EA server and imported into the EA trust store.

To configure the use of credentials from an LDAP database:

1. If necessary, click Configuration from the menu bar.

2. Expand the Policies tree and click the policy to modify.

3. On the Policy Configuration panel, click the Advanced tab.

4. Enable the User Authentication Through External Authentication field.

5. Type the name of the definition you defined in EA in the User Authentication External Authentication Profile field.

6. Deselect the Local User Store option.

7. Select From External Authentication in the User Mapping:Internal User ID field.

8. Click Save.

Test the Inbound and Outbound HTTP Connections

To verify that the engine can receive and initiate communications sessions, you have to establish a connection between an HTTP client and the engine, initiate a session from the engine to the GIS server in the trusted zone, and review the SSP audit log for the results.

> **Note:** Make sure the engine is running when you configure an HTTP adapter. If it is running, configuration files are automatically copied to the engine when you save any update. Configuration files must be available on the engine for communication sessions to be established.

This procedure enables you to verify that the engine can:

✦ Establish an HTTP session initiated by a trading partner using an HTTP client

✦ Initiate an outbound session to a GIS server on behalf of the HTTP client connection

To verify the communications sessions:

1. Make sure the engine is running.

2. Initiate an HTTP client session to the GIS server in your trusted zone.

3. View the Inbound Node Log and the Outbound Node Log.

4. Confirm that the data transfer was successful, as shown in the following sample audit log output.

Sample Inbound Node Log

```
11 Sep 2009 11:38:28,914 [ProxyNearScheduler-Thread-2] INFO
sys.SESSION_NODE.HTTP_Netmap_Any - protocol=http SID=1
SNAME=sherwood.csg.stercomm.com SIP=10.20.246.121 SPORT=40134 SSP104I Session: 1 -
Session Proceeding after Node match: Any
11 Sep 2009 11:38:31,557 [ProxyFarScheduler-Thread-4] INFO
sys.SESSION_NODE.HTTP_Netmap_Any - protocol=http SID=1
SNAME=sherwood.csg.stercomm.com SIP=10.20.246.121 SPORT=40134
DNAME=lunar.csg.stercomm.com DIP=10.20.246.42 DPORT=10054 SUID=admin DUID=admin
SSP102I Session: 1 - Control:ServerAgent Connection closed (CloseCode.EOF): Elapsed
Time: 2.13 (s)
: Bytes Received: 194 [at: 7.286384976525821E-4 MBPS]
Bytes Sent: 20480595 [at: 76.92242253521127 MBPS]
```

Sample Outbound Node Log

```
11 Sep 2009 11:38:28,914 [ProxyNearScheduler-Thread-2] INFO
sys.SESSION_NODE.HTTP_Netmap_Any - protocol=http SID=1
SNAME=sherwood.csg.stercomm.com SIP=10.20.246.121 SPORT=40134 SSP104I Session: 1 -
Session Proceeding after Node match: Any

11 Sep 2009 11:38:31,557 [ProxyFarScheduler-Thread-4] INFO
sys.SESSION_NODE.HTTP_Netmap_Any - protocol=http SID=1
SNAME=sherwood.csg.stercomm.com SIP=10.20.246.121 SPORT=40134
DNAME=lunar.csg.stercomm.com DIP=10.20.246.42 DPORT=10054 SUID=admin DUID=admin
SSP102I Session: 1 - Control:ServerAgent Connection closed (CloseCode.EOF): Elapsed
Time: 2.13 (s)
: Bytes Received: 194 [at: 7.286384976525821E-4 MBPS]
Bytes Sent: 20480595 [at: 76.92242253521127 MBPS]
```

If your session was unsuccessful, review the log information to determine the likely cause of failure and the corrective action to take.

Additional HTTP Configuration Options

Additional HTTP configuration options are available for the following features:

✦ Block common exploits

✦ Change the commands that are allowed or blocked

✦ Rewrite URLs in HTML content to route inbound connections through proxy

✦ Define alternate nodes for failover support

Block Common Exploits

When a connection from an inbound HTTP node to SSP is attempted, you can enable the capability to scan the URL requested and look for commonly occurring exploits.

If block common exploits is enabled, HTTP requests cannot contain the following characters or strings, which are commonly used on attacks on HTTP servers:

```
_
|
`
"
;
<?
\u0000
```
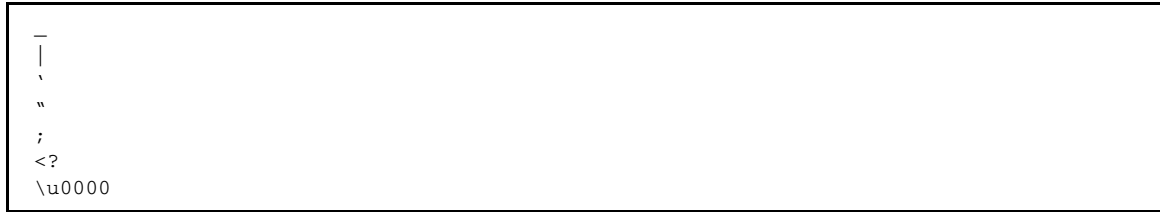
You can change the values that are blocked at the adapter level. To change the values, open an adapter and click the Properties tab.

To enable the capability to block common exploits:

1. If necessary, click Configuration from the menu bar.
2. Expand the Policies tree and click the policy to modify.
3. On the Policy Configuration panel, click the Advanced tab.
4. Enable the Block Common Exploits field.
5. Click Save.

Change the Values to Block in a URL String

When a connection from an inbound HTTP node to SSP is attempted, you can enable the ability to scan the URL requested and look for commonly occurring exploits.

If block common exploits is enabled, HTTP requests cannot contain the characters or strings, identified in the graphic above. You can change the characters that are blocked. To change the blocked strings:

1. If necessary, click Configuration from the menu bar.
1. Expand the Adapters tree and click the adapter to modify.
2. On the HTTP Adapter Configuration panel, click the Properties tab.
3. To edit an existing value, type the new value in the Value field.
4. To delete an item, click the radio button to the left of an item and click Delete.
5. To add a new item:
   a. Click New.
   b. Type block.exploit.strings.*n* as the Key value, where *n* is a unique number appended to the block.exploit.strings key. Be sure that you increment the number and do not duplicate an existing key.
6. Click OK.
7. Click Save.

Map a URL in HTML Content from the Outbound Server

HTTP Reverse Proxy HTML rewriting allows you to replace the URL links submitted by an HTTP client to the HTTP server with URL links to SSP. If the HTTP server has web pages with links to other web pages on the same host, you must map all URL connections in order for the links to work.

Before you configure this option, create a netmap definition. Create an outbound node definition for each URL containing a host and port.

Configure HTTP Rewrite to Support the GIS Dashboard

To communicate with the GIS dashboard, two connections must be established to the outbound GIS server: one connection to the GIS base port and one to the GIS base port + 33.

To configure this environment:

1. Define two outbound nodes in the netmap: Definition 1 configures a connection to the GIS host and base port. Definition 2 configures a connection to the GIS host and base port + 33.

2. Add mapping values to the netmap definition for both URL connections.

3. Configure two HTTP Reverse Proxy adapters: one to route connections to the GIS host and base port (Definition 1) and another to the GIS host and base port + 33 (Definition 2). Use the same netmap with both adapter definitions. For each adapter, select a different outbound node to route connections to in the Standard Routing Node field.

For example, assume SSP is installed and running on the host, proxy_host and HTTP Reverse Proxy adapter 1 is configured to listen on the port, adapter1_port. It uses the outbound node defined as GIS base port on a host called gis_host. HTTP Reverse Proxy adapter 2 listens on the port, adapter2_port and uses an outbound node defined as GIS baseport + 33 (the dashboard default port).

To configure this environment, define the following URL rewrite values in the netmap definition:

Server URL	Proxy URL
http://<gis_host>:<baseport>	http://<proxy_host>:<adapter1_port>
http://<gis_host>:<baseport+33>	http://<proxy_host>:<adapter2_port>

Configure HTML Rewrite

To configure HTML rewrite:

1. If necessary, click Configuration from the menu bar.

2. Expand the Netmaps tree and select an HTTP netmap to modify.

3. Make sure you have two outbound node definitions: one for the GIS server and its base port and another for GIS base port + 33. To define an outbound node definition:

   a. Click the Outbound Nodes tab and click New.

   b. Specify the following values:

      • Outbound Node Name

      • Primary Destination Address

      • Primary Destination Port

4. Click OK.

5. On the HTTP Netmap Nodes panel, click the HTML Rewrite tab.

6. Click New.

7. Enable the Support HTML Rewrite field.

8. Type the URL path for the outbound server in the Server URL field.

9. Type the URL path for the proxy in the Proxy URL field. Refer to *Configure HTTP Rewrite to Support the GIS Dashboard* on page 174 and the table of values for instructions on the URL values to define for the GIS dashboard.

10. Click Save.

11. Repeat steps 3 through 10 for all HTML Rewrite options you want to configure.

12. To reorder the HTML rewrite definitions:

    a. Click the radio button beside the URL routing definition to reorder.

    b. Click Move Up or Move Down until the item is in the correct order.

13. Click Save.

14. Expand the Adapters tree and click the adapter to modify.

15. Enable Support HTML Rewrite.

16. Click Save.

Test the configuration to ensure that the HTML rewrite is configured correctly.

Note: If the following message is written to the secureproxy.log file, correct your URL definition:

HTML Rewrite proxy URL Map entry is not a valid URI.

Define Alternate Nodes for Failover Support for an Outbound HTTP Connection

If you are using standard routing to connect to a GIS server in the secure zone, you define a primary GIS server to connect to in the adapter. For each outbound node definition, you can identify up to three alternate outbound nodes to connect to when the primary GIS server is not available.

Two methods of configuring alternate GIS server routing are available.

✦ Select a previously defined outbound node from the drop-down list on the Advanced tab. To configure this method, you first configure an outbound node definition in the netmap for each alternate GIS server you want to use. Each connection uses the security and other settings defined for that outbound node in the netmap.

✦ Select IP address/port from the drop-down list on the Advanced tab and enter values for the IP address and port. If you use this method you do not have to define the alternate outbound nodes in the netmap, and each alternate connection shares the security and other settings defined in the primary node definition.

If you configure alternate GIS server definitions in the outbound node definition, when a connection to the primary outbound node is unsuccessful SSP tries to connect to the alternate node you defined as Node 1. If the connection to the first alternate node is unsuccessful, SSP tries to connect to the

---

second alternate node, Node 2. If this connection is unsuccessful, SSP tries to connect to the third alternate, Node 3. If the connection to this node is unsuccessful, the inbound connection fails.

To configure alternate outbound connections:

1. If necessary, click Configuration from the menu bar.
2. Expand the Netmaps tree and select an HTTP netmap to modify.
3. Click the Outbound Nodes tab.
4. Select the outbound node to modify and click Edit.
5. Click the Advanced tab.
6. Do one of the following:

   ◆ To identify an alternate node defined in the netmap and use the security settings defined in the alternate node definition, select the outbound node name from the drop-down list.

   ◆ To configure an alternate node that is not in the netmap and use the security settings defined in the primary node definition:

      a. Select Address/Port from the drop-down list in the Alternate Destinations Node field.

      b. Provide the IP Address and Port number for the alternate outbound node

7. Click OK.
8. Click Save.

SFTP Reverse Proxy Configuration

The SFTP configuration scenarios describe how to configure SFTP protocol connections to and from the engine.

Note: Make sure the engine is running when you configure an SFTP adapter. If it is running, configuration information is transmitted to the engine when you save any configuration. Configuration information must be available on the engine before communication sessions with Gentran Integration Suite (GIS) can be established.

Organization of the SFTP Configuration Scenarios

The first scenario instructs you on how to configure a basic configuration. Each successive scenario adds another security feature to the basic configuration. After adding a security feature, test the connection to ensure that you have correctly configured it. You determine your security needs and configure the security features applicable for your environment.

The following scenarios help you configure and test SSP for SFTP protocol connections to the SFTP server:

✦ Create a basic configuration

✦ Perform user authentication using the local user store

✦ Provide user mapping using the netmap

The remaining configuration scenarios require EA, an optional security feature of SSP that must be configured independently of SSP. After EA is configured, you can update your basic security definitions to enable SSP to connect to the EA to enforce the following advanced security features:

✦ Authenticate an inbound user using EA

✦ Manage connection requirements to the outbound server using EA

Additional procedures instruct you how to define alternate nodes for failover support.

Complete SFTP Scenario Worksheets

Before you configure SSP for SFTP connections, gather the information on the worksheet provided with the scenario. You use this information as you configure each feature. Complete worksheets as follows:

✦ Provide a value for each SSP feature listed. Fields listed in the worksheet are required.

✦ Accept default values for fields not listed in the worksheet.

✦ The worksheet identifies the Configuration Manager field where you specify each value.

Complete and Test SFTP Configuration Scenarios

Work through the sequence of SFTP configuration scenarios in the order in which they are presented to add and test security features. Be sure to test each feature before you add the next feature to the configuration. Before you move SSP into production, ensure that you have configured and tested all of the security features you need for your environment.

> **Note:**   As you complete each task, provide all required information. If information is not provided or is incorrect, the following error icon is displayed: 🔴 To view more information about the error, hover over the icon.

Create a Basic SFTP Configuration

This scenario contains all the information and tools to configure SSP to establish a basic connection from a trading partner to the SFTP server as shown in the following diagram. You are configuring the minimum requirements to allow you to test the connections and ensure that communications sessions can be established between the inbound node and SSP, and to the outbound SFTP node. The basic configuration requires that SSP present its key to the inbound node for authentication and that the SFTP server present its key to SSP for authentication. It does not configure user authentication. After you create and test the basic SFTP configuration and all connections are working, you then add user authentication.

You accept default values when configuring this scenario. As a result, user credentials presented by the inbound node are used to connect to the outbound SFTP server.
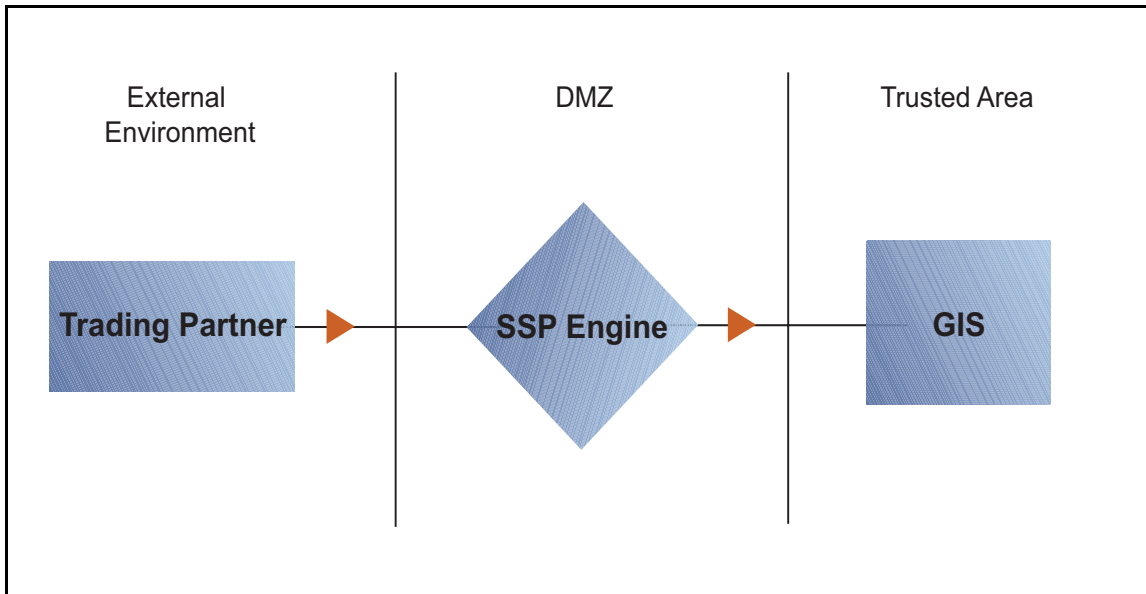
After you configure the basic SFTP configuration, validate it by initiating an SFTP connection from the trading partner. For more information on testing the configuration, see *Test the Inbound and Outbound Connections* on page 196.

Complete the following tasks to define a basic SFTP configuration:

✦ Create a policy
✦ Define inbound and outbound connections in a netmap
✦ Define an SFTP adapter

Basic SFTP Configuration Worksheet

Before you configure SSP for SFTP connections, gather the information on the Basic SFTP Configuration Worksheet. You use this information as you configure a basic SFTP connection for SSP. After you configure SSP for SFTP connections, validate the configuration by initiating an SFTP connection from the inbound node.

SFTP Policy

Create a basic policy. The default authentication method is password authentication. However, the password is not authenticated in the basic configuration because you do not select an authentication mechanism. Instead, it is passed through to the outbound node for authentication. In a later SFTP configuration scenario, you add the configuration information needed to authenticate an inbound node.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy.	_____

SFTP Netmap (Inbound and Outbound Connections)

Create a netmap that contains connection information for the nodes connecting to and from SSP: the trading partner (inbound node) and the GIS SFTP server (outbound node). For the outbound node, you must identify the host name and IP address to connect to the node as well as the known host key to use for server authentication and the ciphers or message authentication codes (MACs) to use to encrypt the data. You also associate the basic policy you create with the inbound node.

Note: You must have SSH keys to authenticate SSP to the inbound node (local host keys) and to authenticate the outbound SFTP server to SSP (known host keys). Create a key store for the keys and check the keys into the key store. Refer to *Manage Local Host Key Stores and Keys* on page 74 for instructions on creating a local host key store and add a key to the key store. Refer to *Manage Known Host Key Stores and Keys* on page 79 for instructions on creating the known host key store and importing the key.

You must have SSP is required by the SFTP server to present its user key for authentication, you must have SSH keys for the local user for this authentication exchange. Refer to *Manage Local User Key Stores and Keys* on page 82 for instructions on creating the local host key store and importing the key.

Configuration Manager Field	Feature	Value
Netmap Name	Netmap name.	_____
Inbound Trading Partner Information		
Inbound Node Name	Trading partner name (name to assign to inbound node definition).	_____ (No spaces allowed.)
Peer Address Pattern	Host name/IP address pattern.	* (Specifying * for this value allows all inbound nodes configured on the SFTP server as trading partners to connect to the SFTP server. To define a more specific node definition, see *Define SFTP Connection Requirements Between SSP and Inbound Nodes* on page 185.)
Policy	Name of policy you create. (Select it from the pull-down list.)	_____

Configuration Manager Field	Feature	Value
Outbound SFTP Server Connection		
Outbound Node Name	Outbound SFTP server node name.	_____
Primary Destination Address	Host name/IP address of SFTP server.	_____
Primary Destination Port	Port number to connect to SFTP server.	_____
Known Host Key Store	Name of the key store where the known host key is stored.	_____
Known Host Key	Location and name of the public key presented to SSP by the outbound SFTP server during authentication.	_____

SFTP Adapter

Create an SFTP adapter that defines information necessary to establish SFTP connections to and from SSP. When you configure the adapter, select the basic netmap and outbound SFTP server in the netmap definition and the local host key that SSP presents to its clients.

Configuration Manager Field	Feature	Value
Adapter Name	Adapter name.	_____
Listen Port	Listen port to use for inbound connections.	_____
Netmap	Netmap to associate with the adapter.	_____
Standard Routing Node	Name of the outbound node corresponding to the GIS server where inbound connections are routed.	_____
Engine	Engine to run on.	_____
Startup Mode	How the adapter is started. **auto** starts the adapter as soon as it is pushed to the engine. **manual** requires that the adapter be manually started.	_____

Configuration Manager Field	Feature	Value
Local Host Key Store	Name of the key store where the local host key is stored.	_____
Local Host Key	Location and name of the private part of the key presented by SSP to the inbound connection during authentication.	_____
Available Cipher Suites Selected Cipher Suites	Cipher suites to enable. (Be sure to match the configuration of the SFTP client.)	_____ _____ _____ _____
Available Cipher Suites Selected Cipher Suites	Cipher suites to enable. (Be sure to match the configuration of the SFTP client.)	_____ _____ _____ _____
Available Key Exchange Selected Key Exchange	Key exchange to enable. (Be sure to match the configuration of the SFTP client.)	_____ _____ _____ _____

Create an SFTP Policy

The SFTP policy defines how you impose controls to authenticate a trading partner trying to access an SFTP server over the public Internet. The basic policy does not enable any security features. You add user authentication to the policy definition in later scenarios.

To define a policy:

1. If necessary, click Configuration from the menu bar.
2. Click Actions > New Policy > SFTP Policy.
3. Specify a name for the policy in the Policy Name field.
4. Click Save.

Create an SFTP Netmap

You define inbound connection information for your external trading partners and outbound connection information for the SFTP server SSP connects to. These values are stored in a netmap. The netmap is associated with a policy and an adapter.

The SFTP protocol requires that the server authenticate itself to the client.

✦ Inbound connection—Server authentication for the inbound connection requires that SSP use its private key to verify its identity to the inbound connection. Before you can configure authentication of SSP, you must configure a local host key store and add the private key to the local host key store. You must also send the public key to the inbound trading partner. Refer to *Manage Local Host Key Stores and Keys* on page 74 for instructions. The keys used to authenticate SSP to the inbound node connection are configured in the adapter definition.

✦ Outbound connection—Server authentication for the outbound connection requires that the SFTP server present its public key to SSP. SSP must use the public key to validate the server connection. Before you can configure authentication of the SFTP server, you must configure a known host key store and add the public key received from the SFTP server to this key store. Refer to *Manage Known Host Key Stores and Keys* on page 79 for instructions.

For authentication of the SFTP server connection, you must determine what ciphers are allowed for encryption and what MACs are allowed for message integrity protection. These MACs and ciphers must also include the required settings from the inbound nodes, the outbound node, and all keys checked into the key stores.You also determine the order of preference for both the ciphers and the MACs. Communicate with the SFTP server administrator to ensure that your configuration matches the SFTP server configuration.

Before you begin this procedure, create a policy to associate with the netmap.

To create a netmap and define inbound and outbound nodes:

1. Click Configuration from the menu bar.

2. Click Actions > New Netmap > SFTP Netmap.

3. Type a name for the netmap in the Netmap Name field.

4. To define an inbound node definition:

    a. Click New.

    b. Specify the following values:

        • Inbound Node Name

        • Peer Address Pattern

        • Policy

    c. Click OK.

5. To define an outbound node definition:

    a. Click the Outbound Nodes tab and click New.

    b. Specify the following values:

        • Outbound Node Name

        • Primary Destination Address

        • Primary Destination Port

        • Known Host Key Store

        • Known Host Key

    c. Click the Security tab.

      d.   Specify the following values:

- Available Cipher Suites
- Available MAC Suites
- Available Key Exchange

      e.   If necessary, reorder the selected cipher suites, MAC suites, and key exchanges.

      f.   Click Ok.

6. Click Save.

Define the Adapter for the SFTP Connection

An SFTP adapter definition specifies both the system-level communications information necessary to establish SFTP connections to and from SSP and the local host key used to validate SSP to an inbound connection. Because the SFTP protocol requires that SSP present its key to the inbound node for authentication, you must configure the adapter with the local host key store and the local host key to present to the inbound connection. Before you can configure the adapter, create a local host key store and a local host key. Refer to *Manage Local Host Key Stores and Keys* on page 74 for instructions.

You must also determine what ciphers are allowed for encryption and what MACs are allowed for message integrity protection, as well as the order of preference for both the ciphers and the MACs. Communicate with the administrator of the inbound node to ensure that your configurations match.

You can create multiple adapter definitions.

Before you begin this procedure, create the following definitions:

✦ A netmap to associate with the adapter

✦ An engine definition to associate with the adapter. Refer to the *Sterling Secure Proxy Installation Guide* for instructions.

To define an SFTP adapter:

1. If necessary, click Configuration from the menu bar.

2. Click Actions > New Adapter > SFTP Reverse Proxy.

3. Specify values for the following:

- Adapter Name
- Listen Port
- Netmap
- Standard Routing Node
- Engine
- Local Host Key Store
- Local Host Key

4. Click the Security tab.

5. Specify values for the following fields:

   - Available Cipher Suites
   - Available MAC Suites
   - Available Key Exchange

6. If necessary, reorder the selected cipher suites, MAC suites, or key exchange algorithms.

Note: If you change one of the following values, you must restart the adapter before the change takes effect: listen port, local host key, selected cipher suites, selected MAC suites, key exchange, compression, maximum sessions, session timeout, inbound perimeter server, outbound perimeter server, or external authentication perimeter server.

---

7. Click Save.

What You Defined with the Basic SFTP Configuration Scenario

Creating secure connections to SFTP servers on behalf of nodes external to your trusted zone requires that you organize information about the trading partners and the SFTP server in a policy, a netmap, and an adapter definition. You created these items when you defined the Basic SFTP Configuration. Be sure to test the Basic SFTP Configuration before you configure additional security features. Refer to *Test the Inbound and Outbound Connections* on page 196 for information about testing the SFTP reverse proxy configurations outlined in this scenario.

As you add complexity to your security configurations using the procedures in the remaining scenarios, you modify these items to configure more complex authentication measures.

Variations on the Basic SFTP Configuration

After you confirm that the communications sessions you established using the basic SFTP configuration were successful, you may want to validate sessions using other types of inbound trading partner definitions before adding complexity to the security configuration. To ensure that you can validate and troubleshoot problems, you should test one variation at a time by changing the configuration, initiating a connection, and verifying the result.

Inbound Trading Partner Node Definitions

You can modify the inbound trading partner node definitions as follows:

- Define a specific IP address
- Define a wildcard peer pattern
- Define an IP/subnet pattern

Define SFTP Connection Requirements Between SSP and Inbound Nodes

You define connection requirements between SSP and inbound nodes by defining inbound node definitions. Refer to your company security requirements to determine how tightly to define the parameters that an inbound node must provide to allow a connection.

You can create inbound node definitions to allow only one individual inbound connection, or you can identify a pattern of IP addresses and create an inbound definition to allow inbound connections matching the pattern to connect to SSP. Methods of defining inbound nodes are as follows:

✦ Create an entry for an individual inbound node and define the inbound node IP address to connect to SSP. Only connections from that IP address are allowed. A single IP Address must be specified as a subnet pattern where all bits are matched, such as 11.22.33.44/32. SSP also supports individual host names. They must match the value returned by a reverse DNS lookup.

✦ Create an inbound node entry that allows all nodes that match an IP/Subnet address pattern. Patterns include:

Match the first 16 bits of an IP address pattern. For example, 10.20.0.0/16 allows all IP addresses that begin with 10.20.* to connect to SSP.

Match the first 8 bits of an IP address pattern. For example, 10.0.0.0/8 allows all IP addresses that begin with 10.* to connect to SSP.

✦ Define an inbound node entry that allows all inbound nodes that match a wildcard host name pattern. When a connection is attempted and you have defined a wildcard host name pattern definition, a reverse DNS lookup is performed on the IP address of the inbound connection. The DNS name is compared to the wildcard pattern. Wildcard patterns include:

Asterisk (*) enables a match on any number of characters. For example, *.a.com allows a connection from b.a.com but not from a.bc.com. Using only the * allows all inbound nodes to successfully connect to SSP.

Question mark (?) enables a match on one character. For example, a.?.com allows a connection from a.b.com but not from a.bc.com.

You can define more than one inbound node definition and use a combination of the node definition methods. However, be sure to order the definitions from most specific to least specific. When an inbound node connection is attempted, SSP compares the IP address of the inbound node to the first inbound node definition. If it matches, a connection is established. If it does not match, SSP checks the next inbound node definition until a match is found. If no match is found, the connection is terminated.

Inbound SFTP Connection Definition - Worksheet

Use the following worksheet to identify the information needed to configure inbound node definitions specific inbound nodes or groups of inbound nodes that match a pattern.

Configuration Manager Field	Define Inbound Trading Partner Information	Value
Note: If you define a single node and definitions for multiple nodes using pattern matching, order the definitions from most specific to least specific. SSP processes them in the order in which they are listed.		
Inbound Node Name	Trading Partner Name.	_____
Policy	Policy to associate with the inbound trading partner.	_____

Configuration Manager Field	Define Inbound Trading Partner Information	Value
For a Single Node		
Peer Address Pattern	IP address/32 or hostname.	_____
	Create an entry for an individual inbound node and define the inbound node IP address that can connect to SSP. Only connections from that IP address will be allowed. SSP supports host name. An example definition is a.b.com.	
	A single IP address must be specified as a subnet pattern where all bits are matched, such as 11.22.33.44/32.	
For Multiple IP Addresses Using IP/Subnet Pattern		
Peer Address Pattern	Peer Address IP/Subnet Pattern Options.	_____
For Multiple Nodes Using Wildcard Peer Address Pattern to Validate Inbound DNS		
Peer Address Pattern	Wildcard Peer Address Pattern.	_____

Define Inbound Node Connection Definitions

This procedure instructs you how to modify the basic SFTP configuration to add inbound node definitions for a group of nodes with similar information, and definitions that limit access to one specific inbound node. It assumes that you have already configured an adapter. Gather a list of all inbound trading partners, including names and IP addresses.

To define inbound connection definitions:

1. Identify patterns that can be used to define groups of inbound nodes.

2. Decide if you need to define a trading partner connection for any individual IP addresses, to increase security.

3. If necessary, click Configuration from the menu bar.

4. Expand the Netmaps tree and click the netmap to modify.

5. Click New to add a new inbound node definition.

6. Using the information you defined on the Inbound Connection Definition Worksheet, provide the following information and click Save:

   ◆ Inbound Node Name

   ◆ Peer Address Pattern

   ◆ Policy

7. Repeat step 6 for every group of connections and for every individual IP address connection you want to define.

8. If necessary, reorder the node definitions in the netmap. Order definitions from most specific to least specific since they will be evaluated in order.

    a. Click the radio button beside the inbound node definition to move.

    b. Click Move Up or Move Down until the node definition is in the correct order.

9. Click OK.

10. Click Save.

Authenticate an Inbound SFTP Node Against Information Stored in the Local User Store

The Create a Basic SFTP Configuration scenario does not authenticate the inbound node. For additional security, you may configure the authentication method to use for the inbound node. You can choose from the following user authentication methods:

✦ Authenticate the password presented by the inbound node against information stored in the local user store.

✦ Authenticate the key presented by the inbound node against information stored in the authorized user key store.

✦ Authenticate either the password or the key presented by the inbound node using information stored in the local user store or the authorized user key store.

✦ Authenticate both the password and the key presented by the inbound node using information stored in the local user store and the authorized user key store.

✦ Authenticate a password using information stored in the EA.

The following scenarios build on the Create a Basic SFTP Configuration scenario by adding user authentication of the inbound node using information from the local user store. Determine which authentication method you want to enable and then complete the procedure to implement it. Refer to *Strengthen the SFTP User Authentication Using EA* on page 193 for instructions on configuring user authentication using EA.

Add Local Authentication to an Inbound Node Worksheet

Before you add user authentication to the inbound connection you created in the Basic Configuration scenario, gather the information on the Add Local Authentication to an Inbound Node Worksheet. Use this information as you configure user authentication for the inbound connection.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy associated with the inbound node.	_____

Configuration Manager Field	Feature	Value
Required Authentication Method	Method to use for the inbound node. Options include: ◆ Password ◆ Key ◆ Password and Key ◆ Password or Key	_____
Internal User ID	The source to use to for the internal user ID.	Pass-Through or Netmap
Name	Name to assign to the user you create.	_____
Password Confirm Password	If you are authenticating the user-supplied password, identify the password value to use to validate the inbound password.	_____

Add Local Authentication to the Inbound Node Using Password Information

This scenario builds on the Basic SFTP Configuration by adding user authentication to the inbound connection. It compares a password presented by the inbound node to information defined in the local user store. You must add the password information to the local user store before you can test this scenario. Refer to *Manage CM User Accounts* on page 87 for instructions.

To add support for password authentication:

1. If necessary, click Configuration from the menu bar.
2. Expand the Policies tree and select the policy to modify.
3. Click the Advanced tab.
4. Select Password as the Required Authentication Method.
5. Enable the User Authentication Mechanism: Through Local User Store option.
6. Click Save.

Authenticate an Inbound Node Using Key Information

This scenario builds on the Basic SFTP Configuration by adding inbound user authentication using a key. This authentication method requires that credentials for the GIS server be defined in the netmap since only the password can be passed through to the GIS server. You must add the key information to the user definition before you can test this scenario. Refer to *Add SSH Keys to a User Account* on page 91 for instructions.

To add support for key authentication:

1. If necessary, click Configuration from the menu bar.
2. Expand the Policies tree and select a policy.

---

3. Click the Advanced tab.

4. Select Key as the Required Authentication Method.

5. Enable the User Authentication Method: Through Local User Store option.

6. In the Internal User ID field, select Netmap.

7. Click Save.

8. Expand the netmap tree and open the netmap to edit.

9. Click the Outbound Nodes tab.

10. Select the outbound node to edit and click Edit.

11. Click the Advanced tab.

12. Type the user ID and password or key to use to connect to the outbound GIS server.

13. Click OK.

14. Click Save.

Authenticate an Inbound Node by Comparing Either a Password or a Key to the Local User Store

This scenario builds on the Basic SFTP Configuration by adding support for either key or password authentication of the inbound connection. The inbound node may present a key or a password. Only one must be authenticated for a communications session to be established. This authentication method requires that credentials for the GIS server be defined in the netmap, since only a password can be passed through to the GIS server.

You must add the password and key information to the user definition in the local user store before you can test this scenario. Refer to *Create an Engine User Account* on page 90 for instructions on creating a user account and assigning a password. Refer to *Add SSH Keys to a User Account* on page 91 for instructions on adding a key to a user account definition.

To add support for either password or key authentication:

1. If necessary, click Configuration from the menu bar.

2. Expand the Policies tree and select a policy.

3. Click the Advanced tab.

4. Select Password or Key as the Required Authentication Method.

5. Enable the User Authentication Mechanism: Through Local User Store option.

6. In the Internal User ID field, select Netmap.

7. Click Save.

8. Expand the netmap tree and open the netmap to edit.

9. Click the Outbound Nodes tab.

10. Select the outbound node to edit and click Edit.

11. Click the Advanced tab.

12. Type the user ID and password to use to connect to the outbound GIS server.

13. Click OK.

14. Click Save.

Authenticate an Inbound Node by Comparing Both a Password and a Key to the Local User Store

This scenario builds on the Basic SFTP Configuration by adding support for both key and password authentication of the inbound connection. The inbound node must present both a key and a password and both must be authenticated for a communications session to be established.

You must add the password and key information to the user definition in the local user store before you can test this scenario. Refer to *Create an Engine User Account* on page 90 for instructions on creating a user account and assigning a password. Refer to *Add SSH Keys to a User Account* on page 91 for instructions on adding a key to a user account definition.

To add support for password and key authentication:

1. If necessary, click Configuration from the menu bar.

2. Expand the Policies tree and select a policy.

3. Click the Advanced tab.

4. Select Password and Key as the Required Authentication Method.

5. Enable the User Authentication Method: Through Local User Store option.

6. In the Internal User ID field, select Pass-through.

7. Click Save.

After you configure user authentication using both key and password information, validate the configuration by establishing a session initiated by an SFTP client to an SFTP server.

Provide User Mapping Using the Netmap

This scenario builds on the Basic SFTP Configuration by enabling the use of different user credentials for the outbound connection to the SFTP server.

If you configure this option, the credentials presented by the inbound trading partner are not used to connect to the SFTP server. Credentials stored in the netmap are used to connect to the SFTP server. This method prevents trading partners from accessing the actual credentials used to connect to the internal SFTP server.

After you configure the use of alternate credentials to connect to the SFTP server using information from the netmap, test the configuration by establishing a session initiated by an SFTP client to a SFTP server. Refer to *Test the Inbound and Outbound Connections* on page 196 for more information on testing the configuration described in this scenario.

Provide User Mapping Using the Netmap - Worksheet

In this scenario, edit the netmap and the policy you created in the basic configuration to strengthen the outbound connection by providing user credentials and a mapping method to use to secure the outbound connection to the SFTP server.

Collect the following information so you can match the SSP configuration with the SFTP server configuration. Use the information on this worksheet as you edit the outbound node definition, and be sure to select the netmap and policy you created in the Basic Configuration.

Configuration Manager Field	Feature	Value
Netmap	Name of netmap to modify.	_____
Policy	Name of policy to modify.	
User ID	User ID to connect to the SFTP server (Defined at the SFTP server).	_____
Password	Password to connect to the SFTP server (Defined at the SFTP sever).	_____
Local User Key Stores	The name of the key store where the key to authenticate SSP to the outbound connection is stored.	_____
Local User Key	The local user key to use to authenticate SSP to the outbound connection.	_____

Connect to the Outbound Server Using Credentials from the Netmap

To increase security for connections to the server in the trusted zone, you can use the netmap to prevent the user ID and password, or the key provided by the trading partner, from being used to connect to the server. If you configure this option, the inbound node uses one set of credentials to connect to SSP and uses information stored in the netmap to connect to the outbound server.

Before you configure this option:

✦ Ensure that a user ID and password or key are defined for the outbound connection on the SFTP server

✦ Obtain the user ID and password.

To configure validation for the outbound connection using credentials stored in the netmap:

1. If necessary, click Configuration from the menu bar.

2. Expand the Netmaps tree and click the netmap to modify.

3. Click the Outbound Nodes tab.

4. Select the outbound node to modify and click Edit.

5. Click the Advanced tab.

6. Type the values to use to connect to the SFTP server:

   ◆ User ID

   ◆ Password

   ◆ Local User Key Stores

   ◆ Local User Key

7. Click Save.

8. Expand the Policies tree and click the policy to modify.

9. On the Policy Configuration panel, click the Advanced tab.

10. From the User Mapping: Internal User ID list, select Netmap.

11. Click Save.

Strengthen the SFTP User Authentication Using EA

This scenario builds on the basic SFTP configuration by adding user and password authentication or user and key authentication using information defined in EA. To provide a more advanced method of securing the SFTP connection, use EA.

Authenticate an Inbound SFTP User or Key Using EA

You can authenticate an inbound connection against information stored in an LDAP database by configuring EA to define how the connection is authenticated. The EA definition determines the options that are enabled. EA will return a user ID, password, and routing name for a local user key stored on SSP. Refer to Sterling External authentication Server help for a complete list of the functions that can be performed in EA.

Authenticate an Inbound SFTP User or Key Using EA Worksheet

Use the following worksheet to identify the information needed to authenticate a trading partner user ID, password, or key using EA. Update the policy you created in the Basic Configuration for this scenario.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy associated with the inbound node.	_____

Configuration Manager Field	Feature	Value
Required Authentication Method	Method to use to authenticate the inbound node. Options include: ◆ Password ◆ Key ◆ Password and Key ◆ Password or Key	_____
User Authentication Mechanism - Through External Authentication	Enable this option because you will validate user information using EA.	_____
User Authentication Profile	If you are authenticating a user ID and password, type the name of the profile defined in EA used to authenticate the user.	_____
Key Authentication Profile	If you are authenticating the user ID and key, type the name of the profile defined in EA to authenticate the key.	_____

Authenticate the Inbound User ID and Password Using EA

To authenticate the user ID and password provided by the inbound node against information stored in an LDAP database, you must configure EA. After you configure EA to enable user authentication, use this procedure to configure SSP to use the authentication method you defined.

Before you configure SSP, obtain the name of the EA definition and ensure that the EA server connection has been configured.

To configure authentication of an inbound node password using EA:

1. If necessary, click Configuration from the menu bar.

2. Expand the Policies tree and click the policy to modify.

3. On the Policy Configuration panel, click the Advanced tab.

4. Select Password or Password or Key in the Required Authentication Method field.

5. Enable User Authentication Mechanism - Through External Authentication and type the name of the user authentication definition you defined in EA in the User Authentication Profile field.

6. Deselect the Through Local User Store option.

7. Click Save.

You can now associate this policy with a inbound node for which you want to perform user authentication using EA.

Authenticate the Inbound User ID and Key Using EA

To authenticate key information about the inbound node against information stored in an LDAP database, you must configure EA. After configuring EA, use this procedure to configure SSP to use the authentication method you defined.

Before you configure SSP, obtain the name of the EA definition and ensure that the EA server connection has been configured.

To configure authentication of an inbound node password using EA:

1. If necessary, click Configuration from the menu bar.

2. Expand the Policies tree and click the policy to modify.

3. On the Policy Configuration panel, click the Advanced tab.

4. Select Key in the Required Authentication Method field.

5. Enable Key Authentication Mechanism - Through External Authentication and type the name of the key authentication definition you defined in EA in the Key Authentication Profile field.

6. Deselect the Through Local User Store option.

7. Click Save.

You can now associate this policy with a inbound node for which you want to perform key authentication using EA.

Strengthen the Outbound SFTP Connection With EA User Mapping

This scenario builds on the basic SFTP configuration by adding user or key mapping using information defined in EA. To provide a more advanced method of securing an SFTP connection, use EA to map a user ID and password or user key presented by the inbound node to login credentials stored in EA. The mapped login credentials are used to connect to the outbound server in the secure zone.

Manage SFTP User Mapping Using EA

For a higher level of security when connecting to the outbound server, use information stored in an LDAP database to connect to the outbound server. To use information in an LDAP database, you configure EA. You can use EA to map a user ID, password, or key provided by an inbound connection to a user ID, password, or key that is not exposed to the external node.

Perform User Mapping Using EA in an SFTP Environment Worksheet

Use this worksheet to identify the information needed to authenticate a trading partner user ID, password, or key using EA. Update the policy you created in the Basic Configuration for this scenario.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy associated with the inbound node.	_____
Internal User ID	The source to use to for the internal user ID.	External Authentication

Connect to the Outbound SFTP Node Using Information Stored in LDAP

If you store user credentials in an LDAP database, use this procedure to configure SSP to use these credentials to connect to the secure outbound server.

Before you configure this option:

✦ Configure a SSH key authentication definition in EA and obtain the name of the EA definition.

✦ Configure the EA server to allow connections from SSP.

✦ Ensure that the public keys for SSP have been sent to the EA server and imported into the EA trust store.

✦ Configure SSP for user authentication through EA. Refer to *Strengthen the SFTP User Authentication Using EA* on page 193.

To configure the use of a password or a key from the LDAP database:

1. If necessary, click Configuration from the menu bar.

2. Expand the Policies tree and click the policy to modify.

3. On the Policy Configuration panel, click the Advanced tab.

4. Select From External Authentication in the User Mapping:Internal User ID field.

5. Click Save.

Test the Inbound and Outbound Connections

To verify that the engine can receive and initiate communications sessions, you have to establish a connection between an SFTP client and the engine, initiate a session from the engine to the SFTP server in the trusted zone, and review the SSP audit log for the results.

> **Note:** Make sure the engine is running when you configure an SFTP adapter. If it is running, configuration files are automatically copied to the engine after you save any update. Configuration files must be available at the engine for communication sessions to be established.

This procedure enables you to verify that the engine can:

✦ Establish a session initiated by a trading partner using an SFTP client

✦ Initiate an outbound session to an SFTP server on behalf of the client connection

To verify the communications sessions:

1. Make sure the engine is running.

2. Initiate a client session to the SFTP server in your trusted zone from a trading partner.

3. View the log file at the client to ensure that the connection from the inbound node to SSP was successful.

4. View the log file of the engine to ensure that the connection to SSP was successful.

Route an Outbound Connection to Alternate SFTP Servers

When you configure an SFTP adapter, you define a primary SFTP server to connect to. For each outbound node definition, you can identify up to three alternate outbound nodes to connect to if the primary SFTP server is not available.

Two methods of configuring alternate SFTP server routing are available.

✦ Select an SFTP server from the drop-down list. To configure this method, you first configure an outbound node definition in the netmap for each alternate SFTP server. Each alternate connection uses the security and advanced settings defined for the outbound node in the netmap.

✦ Select IP address/port from the drop-down list and enter values for the IP address and port. If you use this method, you do not have to define the alternate outbound nodes in the netmap. Each alternate connection uses the security and advanced settings defined in the primary node definition.

If you configure alternate SFTP server definitions in the outbound node definition, when a connection to the primary outbound node is unsuccessful SSP tries to connect to the alternate node you defined as Node 1. If the connection to the first alternate node is unsuccessful, SSP tries to connect to the second alternate node, Node 2. If this connection is unsuccessful, SSP tries to connect to the third alternate, Node 3. If the connection to this node is unsuccessful, the inbound connection is aborted.

To configure alternate outbound connections:

1. If necessary, click Configuration from the menu bar.

2. Expand the Netmap tree and click the netmap to modify.

3. Click the Outbound Nodes tab.

4. Select the outbound node to modify and click Edit.

5. Click the Advanced tab.

6. Do one of the following:

    ◆   To identify an alternate node that is defined in the netmap and use the security settings defined in the alternate node definition, select the outbound node name from the drop-down list.

    ◆   To configure an alternate node that is not in the netmap and use the security setting defined in the primary node definition:

        a.   Select Address/Port from the drop-down list in the Alternate Destinations Node field.

        b.   Provide the IP Address and Port number for the alternate outbound node

7. Click OK.

8. Click Save to save the netmap updates.

PeSIT Proxy Configuration

SSP and PeSIT Overview

Sterling Secure Proxy (SSP) acts as an application proxy between Connect:Express and PeSIT nodes. It provides a high level of data protection between external PeSIT connections and your internal network. Define an inbound node definition for each trading partner connection from outside the company and outbound node definition for every company server to which SSP will connect.

SSP provides reverse proxy services for Connect:Express servers when the trading partners initiate sessions to Connect:Express servers in the trusted zone. SSP provides forward proxy services for Connect:Express servers when the node in the trusted zone initiates a session to a server at a remote trading partner.

SSP provides these services for Connect:Express and PeSIT nodes in a manner similar to the way it provides these services for other protocols.

The PeSIT configuration scenarios describe how to configure PeSIT protocol connections to and from the SSP engine using Configuration Manager.

Supported PeSIT Software

The following software is supported for use with the SSP PeSIT Proxy Adapter:

✦   Connect:Express for z/OS (formerly OS.390) version 4.2.2 or later

✦   Connect:Express for UNIX version 1.4.4 or later

✦   Connect:Express for Windows version 3.0.5 or later

Organization of the PeSIT Configuration Scenarios

The first scenario instructs you how to do a basic setup. Each successive scenario adds an additional security feature to the basic configuration. After you go through each scenario, test the connection to ensure that it is correctly configured. You determine your security needs and configure the security features applicable to your environment.

The scenarios include the following:

✦ Create a basic PeSIT configuration

✦ Add SSL/TLS support

✦ Configure PNODE-based routing

✦ Add local Logon ID authentication

✦ Provide outbound credentials using the netmap

The remaining configuration scenarios require Sterling External Authentication Server (EA), an optional security feature of SSP that must be configured independently of SSP. After EA is configured, you can update your basic security definitions to enable SSP to connect to EA to enforce the following advanced security features:

✦ Authenticate an inbound certificate or user using EA

✦ Configure logon ID mapping to the SNODE using EA

✦ Configure certificate-based routing

Additional procedures are provided to instruct you how to configure the following features:

✦ Define alternate nodes for failover support

✦ Enable action based on protocol errors

✦ Block a PeSIT command from a PNODE

Complete Scenario Worksheets

Before you perform each PeSIT configuration, gather the information on the provided worksheet. You use this information as you configure each feature. Complete worksheets as follows:

✦ Enter a value for each listed SSP feature. Fields listed in the worksheet are required.

✦ Accept default values for fields not listed in the worksheet.

✦ The worksheet identifies the Configuration Manager field where you will specify each value.

Complete and Test Configuration Scenarios

Work through the sequence of PeSIT configuration scenarios in the order in which they are presented to add security features. Be sure to test each feature before you add the next one to the configuration. Before you move SSP into production, ensure that you have configured and tested all security features you need for your environment.

Note: As you complete each task, provide all required information. If information is not provided or is incorrect, the following error icon is displayed: 🔴 . To view more information about the error, hover over the icon.

Create a Basic PeSIT Configuration

This scenario contains all the information and tools you need to configure SSP to establish a basic connection between PeSIT servers. Using default values, the PNODE presents a Logon ID to connect to the SNODE without EA. As a result, no authentication occurs in SSP and the logon ID presented by the PNODE is used to connect to the SNODE. This scenario assumes that all nodes are Connect:Express nodes.

The basic configuration uses standard routing to route connections to the node you define in the adapter. You are instructed on how to configure PNODE routing, mixed routing, and certificate-based routing in later scenarios.
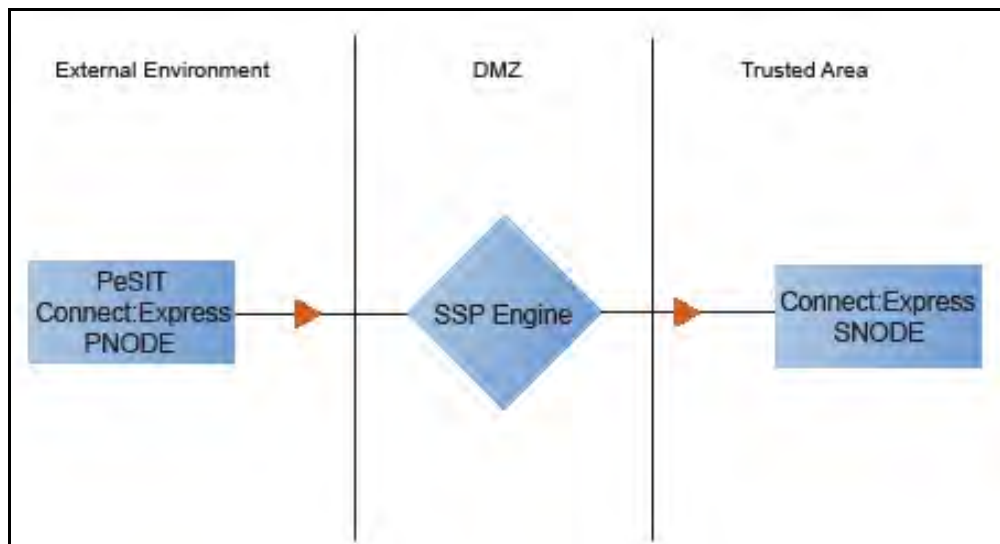
Before you configure a PeSIT connection, make sure that an engine has been configured. Refer to the *Sterling Secure Proxy Installation Guide* for instructions.

After you configure SSP, validate the configuration by initiating a PeSIT connection from the PNODE. For more information on testing the configuration, see *Test the PeSIT Connections* on page 221.

Complete the following tasks to define a basic PeSIT configuration:

✦ Create a policy

✦ Define PeSIT nodes in a netmap

✦ Define a PeSIT adapter

Basic PeSIT Configuration Worksheet

Before you configure SSP for PeSIT connections, gather the information on the basic PeSIT configuration Worksheet. You use this information as you configure a basic PeSIT connection for SSP.

Policy

Create a basic policy. In a later PeSIT configuration scenario, you edit this policy to add security features to it.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy	_____

Netmap (All PeSIT Nodes)

Create a netmap that contains connection information for the nodes connecting to and from SSP. For each node, associate a policy with the node

Configuration Manager Field	Feature	Value
Netmap Name	Netmap name	_____
PeSIT Node Definition		
Node Name	Name to assign to the PeSIT node definition	_____
PeSIT Server Address	Host name or IP address of the PeSIT server	_____
PeSIT Port	Listening port number of the PeSIT server	_____
Policy	Name of policy you create (Select from a pull-down list.)	_____

Configuration Manager Field	Feature	Value
Node Name	Name to assign to the PeSIT node definition	_____
PeSIT Server Address	Host name or IP address of the PeSIT server	_____
PeSIT Port	Listening port number of the PeSIT server	_____
Policy	Name of policy you create (Select from a pull-down list.)	_____
Node Name	Name to assign to the PeSIT node definition	_____
PeSIT Server Address	Host name or IP address of the PeSIT server	_____
PeSIT Port	Listening port number of the PeSIT server	_____
Policy	Name of policy you create (Select from a pull-down list.)	_____

PeSIT Adapter

Create a PeSIT adapter that defines information necessary to establish PeSIT connections to and from SSP. When configuring the adapter, select the basic netmap and the PeSIT server where connections are routed and defined in the netmap definition.

Configuration Manager Field	Feature	Value
Name	Adapter name	_____
Listen Port	Listen port to use for inbound connections	_____
Netmap	Netmap to associate with the adapter	_____
SNODE Netmap Entry	Name of PeSIT node where the connection is routed	_____
Engine	Engine to run the PeSIT adapter on	_____

Create a Basic PeSIT Policy

The policy defines how you impose controls to authenticate a PeSIT PNODE trying to communicate with a PeSIT SNODE over the public Internet. The basic policy does not enforce any controls over the defined node. You add security controls when you define more advanced security settings.

To define a basic policy:

1. Select Configuration from the menu bar.

2. Click Actions > New Policy > PeSIT Policy.

3. Type a Policy Name.

4. Click Save.

Create a PeSIT Netmap

You define connection information for every PeSIT node that communicates using SSP. These values are stored in a netmap. The netmap is associated with a policy and an adapter.

Before you begin this procedure, create a policy to associate with the netmap.

To create a netmap and define PeSIT nodes:

1. Select Configuration from the menu bar.

2. Click Actions > New Netmap > PeSIT Netmap.

3. Type a Netmap Name.

4. To define a PeSIT node definition, click New.

5. Specify the following values:

   ◆ Node Name

   ◆ PeSIT Server Address or hostname

   ◆ PeSIT Server Port (listening port)

   ◆ Policy

6. Click OK.

7. Repeat steps 4 through 6 for each node you want to define. Define at least one PNODE and at least one SNODE in order to establish a connection between two PeSIT nodes.

8. Click Save.

Define the PeSIT Adapter Used for the Connection

A PeSIT adapter definition specifies system-level communications information necessary for PeSIT connections through SSP.

Before you begin this procedure, create a netmap and an engine to associate with the adapter.

To define a PeSIT adapter:

1. Select Configuration from the menu bar.

2. Click Actions > New Adapter > PeSIT Proxy.

3. Specify values for the following:

   - Name

   - Listen Port

   - Netmap

   - SNODE Netmap Entry

   - Engine

4. Click Save.

What You Defined with the Basic PeSIT Configuration Scenario

Creating connections between PeSIT nodes when routing them through SSP requires that you organize information about the PeSIT nodes in a policy, a netmap, and an adapter definition. You created these items when you defined the basic PeSIT configuration. The next step is to test the configuration to ensure that the connections work. Before you test the configuration, be sure that:

✦ The Connect:Express SNODE server has a definition in its partner's directory for the Connect:Express or PeSIT PNODE. The definition must use the IP address of the SSP server. The local name must be the SNODE name.

✦ The Connect:Express PNODE server has a definition in its partner's directory for the Connect:Express or PeSIT SNODE, using the IP address and port of the SSP server. The local name must be the PNODE name.

Refer to *Test the PeSIT Connections* on page 221 for information about testing the PeSIT proxy configuration outlined in this scenario.

As you add complexity to your security configurations using the procedures in the remaining scenarios, you modify the basic configuration to configure more complex authentication and certificate validation measures.

Add SSL/TLS Support

This scenario builds on the basic PeSIT configuration by enabling security for the nodes you defined in the netmap.

Adding SSL/TLS support to the netmap for the nodes involves selecting the following options for the connections:

✦ SSL or TLS Protocol
✦ Cipher suites
✦ Certificate stores and certificates

Add SSL/TLS support to the PNODE and the SNODE definitions. Set up SSL/TLS parameter files at both the SNODE and the PNODE servers. Obtain certificates for both sessions and check them into the certificate store. Then, test the connection.

> **Note:** This procedure assumes you have checked in your certificates. Refer to Chapter 3, *Manage Certificates for SSL/TLS Transactions with Trading Partners*, for more information.

SSL/TLS Support Worksheet

Before you add SSL/TLS support to the connection information you created in the basic PeSIT configuration scenario, gather the information on the SSL/TLS Support Worksheet. You use this information as you configure the inbound and outbound nodes for SSL/TLS support.

Select the security setting and cipher suites to be used to secure the connection. To require that the certificate common name be validated in a certificate presented, enable this option and identify the common name value to check. Select the key/system certificate to use to validate the connection.

Configuration Manager	Feature	Value
Node Name	Name of the node to add security to, from the nodes you've already defined.	_____
Use SSL	Enable this option to enable security checking	Enabled
Verify Common Name	Enable this option to enable common name checking. This is optional.	Enabled/Disabled
Certificate Common Name	Value of common name in certificate presented, if Common Name Checking is enabled.	_____
Security Setting	Security protocol to use. Options include SSL or TLS.	_____
Enable Client Authentication	Do you want to require the inbound connection to present its certificate for SSL or TLS client authentication?	_____
Trust Store	Name of the store for the CA certificate or trusted root certificate	_____
CA Certificates/Trusted Root	Name of CA certificate/trusted root	_____
Key Store	Name of the store for the key or system certificate is stored	_____
Key/System Certificate	Name of the SSP system certificate presented to the PeSIT server	_____
Available Cipher Suites	Select the ciphers to enable by moving them from the Available Cipher Suites to the Selected Cipher Suites field	_____ _____ _____ _____

Secure the PeSIT Connection Using the SSL or TLS Protocol

The first step to strengthen security is to secure the communications channel. This procedure describes how to enable the SSL or TLS protocol for the PeSIT connections to and from SSP in a netmap you created in the basic configuration. To require that SSP perform common name checking, enable this option and identify the common name in the configuration.

Before you can configure this option, you must obtain the necessary certificates and place them in the SSP Cert Store. Refer to Chapter 3, *Manage Certificates for SSL/TLS Transactions with Trading Partners* for instructions.

To enable the SSL or TLS protocol:

1. Select Configuration from the menu bar.

2. Expand the Netmaps tree and select a netmap to modify.

3. Select a node to modify, and click Edit.

4. Click the Security tab, and then click Use SSL to enable security.

5. To enable common name checking:

    a. Click Verify Common Name.

    b. Type the certificate common name in the Certificate Common Name field.

6. Select values for the following:

    ◆ Security Setting

    ◆ Key Store

    ◆ Key/System Certificate

    ◆ Available Ciphers

    ◆ Selected Ciphers

7. To enable client authentication:

    a. Click Enable Client Authentication.

    b. Select the Trust Store where the certificate you want to use is located.

    c. Select the CA Certificates/Trusted Root to use to authenticate the certificate presented by the inbound node.

    > **Note:**   Be sure to highlight the certificate to select. If only one certificate is displayed in the field, it is not selected until you highlight it.

8. Click OK.

9. Click Save.

Establish a session initiated by a Connect:Express PNODE to test the configuration.

Variation on the Add SSL/TLS Support Configuration

After you confirm that the communications session you established using the Add SSL/TLS Support scenario was successful, you may want to further modify your configuration. After testing the SSL/TLS configuration, you can configure the environment to allow the inbound and outbound sessions to use different levels of encryption.

Allow Different Levels of Encryption for the Inbound and Outbound Node

In a PeSIT environment where SSP is not being used, one session is established between an SNODE and a PNODE. In the SSP environment, a session break is created; therefore, two sessions are established: one between the PNODE and SSP and another between SSP and the SNODE. To use the same protocol on both sessions, use the default settings.

Complete this procedure to define one protocol for the inbound node and a different protocol for the outbound node. This function is useful when you want to secure the inbound connection but allow a nonsecure session between SSP and the outbound node.

To enable different levels of encryption for the inbound and the outbound connection:

1. Select Configuration from the menu bar.
2. Expand the Adapter tree, and select the adapter you want to modify.
3. Click the Advanced tab.
4. Enable the Inbound and outbound sessions can have different levels of encryption option.
5. Click Save.

Configure PNODE-Based Routing

The basic configuration uses standard routing to determine where a connection is routed. If you configure standard routing, all sessions through an adapter are routed to the same connection. To allow a PNODE to determine what SNODE it connects to, configure PNODE-based routing. For PNODE-based routing, you must configure a node definition in the netmap for the PNODE and for all the SNODEs you will route to.

PNODE-based Routing Worksheet

This scenario builds on the basic PeSIT configuration by enabling PNODE-based routing. Before you add PNODE-based routing to the connection information you created in the basic PeSIT

configuration scenario, gather the information on the PNODE-based Routing Worksheet. You use this information as you configure PNODE-based routing.

Make sure that you have a node definition for the PNODE and for every node where the connection is routed in the netmap you select.

Configuration Manager Field	Feature	Value
Name	Adapter name	_____
Routing Type	Routing type to use for this connection	PNODE-specified

Configure PNODE-based Routing

To configure a PeSIT adapter to use PNODE-based routing:

1. Select Configuration from the menu bar.
2. Expand the Adapter tree and select the adapter you want to modify.
3. Select PNODE-specified in the Routing Type field.
4. Click Save.

Configure Mixed Routing

Mixed routing allows a PNODE to determine what SNODE it connects to. If the PNODE does not identify what SNODE to connect to, mixed routing then routes to the SNODE identified in the SSP configuration. Before PNODE-based routing can be implemented, you must configure a node definition in the netmap for the PNODE and the SNODE.

Mixed Routing Worksheet

This scenario builds on the basic PeSIT configuration by enabling PNODE-based routing. Before you add PNODE-based routing to the connection information you created in the basic PeSIT configuration scenario, gather the information on the Mixed Routing Worksheet. You use this information as you configure PNODE-based routing.

Make sure that you have a node definition for the PNODE and for the node where the connection is routed in the netmap you select.

Configuration Manager Field	Feature	Value
Name	Adapter name	_____
Routing Type	Routing type to use for this connection	PNODE-specified and then Standard (mixed)
SNODE Netmap Entry	SNODE to route connections to	_____

Configure PNODE Specified and Then Standard (Mixed) Routing

To configure a PeSIT adapter to use PNODE specified and then standard (mixed) routing:

1. Select Configuration from the menu bar.
2. Expand the Adapter tree and select the adapter you want to modify.
3. Select PNODE-Specified, then Standard (mixed) in the Routing Type field.
4. Select the SNODE to route connections to in the SNODE Netmap Entry field.
5. Click Save.

Add Local User Authentication to a PeSIT Connection

This scenario builds on the basic PeSIT configuration by adding local user authentication to the PNODE connection using information defined in the local user store. The logon ID and password presented by the PNODE are authenticated against information stored in the local user store. The

values must match before a connection is established. You must add this information to the local user store before you can test this scenario.
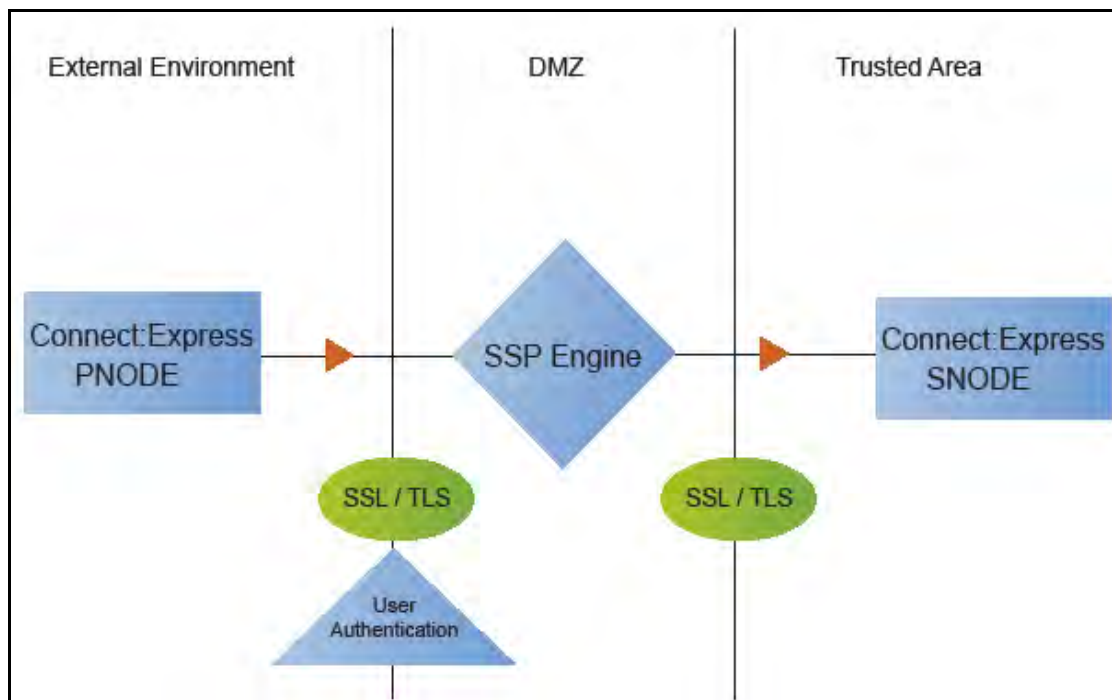
Adding logon ID authentication to the PNODE connection defined in the basic PeSIT configuration involves enabling logon ID authentication and specifying information about the PNODE.

After you configure local logon ID authentication, validate the configuration by establishing a session initiated by a Connect:Express PNODE.

PeSIT PNODE Connection (Local LogonID Authentication) Worksheet

Before you add local logonID authentication to the PNODE connection you created in the basic PeSIT configuration scenario, gather the information on the PeSIT PNODE Connection (Local LogonID Authentication) Worksheet. Use this information as you configure logonID authentication for the PNODE connection.

In this scenario, you edit the policy you created in the PeSIT basic configuration scenario and enable logonID authentication. You also add a logonID and password for the PeSIT PNODE to the default user store.

Configuration Manager Field	Feature	Value
Policy Name	Name of policy associated with the inbound node	_____
LogonID Authentication	Method to use to authenticate the inbound node	Through local user store

Configuration Manager Field	Feature	Value
User Store	Name of the user store you create	_____
User Name	Name of the user you define in the User Store	_____
Password Confirm Password	The password value to use to validate the inbound connection	_____

Add User Authentication to the PeSIT Inbound Connection

You can strengthen the security of PeSIT PNODE connections by enabling local logonID authentication. This procedure describes how to configure local logonID authentication.

To add local logonID authentication for a PNODE connection:

1. Select Configuration from the menu bar.

2. Expand the Policies tree and select a policy.

3. Click the Advanced tab.

4. Enable the LogonID Authentication Through Local User Store option.

5. Click Save.

Add Credentials to the Local User Store

If you enable logon ID authentication through the local user store, you also add logonID information to the local user store that is validated by SSP during a PeSIT client connection.

Before you begin this procedure:

✦ Enable logon ID authentication for the inbound connection.

✦ Ensure that the engine is configured to use the user store that contains the user credentials.

To add user information to the local user store:

1. Select Credentials from the menu bar.

2. Click User Stores to expand the list of user stores.

3. Select the default user store called defUserStore.

4. From the User Store Configuration panel, click New.

5. Specify values for the following:

   ◆ User Name

   ◆ Password

   ◆ Confirm Password

6. Click OK.

7. Click Save.

Provide Credentials to the Outbound PeSIT Node Using the Netmap

This scenario builds on the Basic PeSIT Configuration by enabling the use of Logon credentials from the netmap to connect to the outbound PeSIT connection. Following is an illustration of the security features supported in this scenario:

When an inbound trading partner connects to SSP, its credentials are replaced with credentials stored in the netmap. The replacement credentials are then used to connect to the outbound PeSIT server. This method uses SSP security features to prevent trading partners from knowing the credentials used to connect to the outbound Connect:Express. The outbound Connect:Express must have a partner definition that accepts the LogonID and password provided.

After you configure the environment to use credentials defined in the netmap, test the configuration by establishing a session initiated by a Connect:Express client to a Connect:Express server. Refer to *Test the PeSIT Connections* on page 221 for more information on testing the configuration defined in this scenario.

Provide Credentials for the Outbound PeSIT Node Using the Netmap - Worksheet

In this scenario, edit the netmap and policy you created in the Basic PeSIT Configuration to provide user credentials stored in SSP to connect to the outbound PeSIT connection.

Collect the following information so that you can match the SSP configuration with the Connect:Express server configuration. Use the information on this worksheet as you edit the outbound node definition. Select the netmap and policy you created in the Basic PeSIT Configuration.

Configuration Manager Field	Feature	Value
Logon ID	Partner ID to use to connect to the Connect:Express server. (Must also be defined at the Connect:Express server.)	_____
Password	Password to use to connect to the Connect:Express server. (Must also be defined at the Connect:Express server.)	_____

Connect to the Outbound PeSIT Server Using Credentials from the Netmap

To increase security for connections to the server in the trusted zone, you can use the netmap to store the Logon ID and password to connect to the outbound Connect:Express server. If you configure this option, the inbound node uses one set of credentials to connect to SSP and SSP uses information stored in the netmap to connect to the outbound Connect:Express server.

Before you configure this option:

✦ Ensure that the Logon ID and password are defined on the Connect:Express server.

✦ Obtain the Logon ID and password.

To configure validation for the outbound connection using credentials stored in the netmap:

1. Click Configuration from the menu bar.

2. Expand the Netmaps tree and select the PeSIT netmap to modify.

3. Select the Connect:Express server node to modify and click Edit.

4. Click the Advanced tab.

5. Type values in the following fields for connecting to the GIS server:

   ◆ Logon ID

   ◆ Password

6. Click Save.

7. Expand the Policies tree and select the policy to modify.

8. On the PeSIT Policy Configuration panel, click the Advanced tab.

9. From the LogonID Mapping: Internal LogonID list, select Replace LogonID with Netmap LogonID.

10. Click Save.

Test the configuration to ensure that this feature is working.

Strengthen LogonID Authentication Using EA

This scenario builds on the basic PeSIT configuration by adding logonID authentication to the PNODE connection using information defined in EA. To provide a more advanced method of securing a PeSIT connection, use EA to authenticate certificate information or logonID credentials presented by the inbound node or to perform logonID and password mapping.
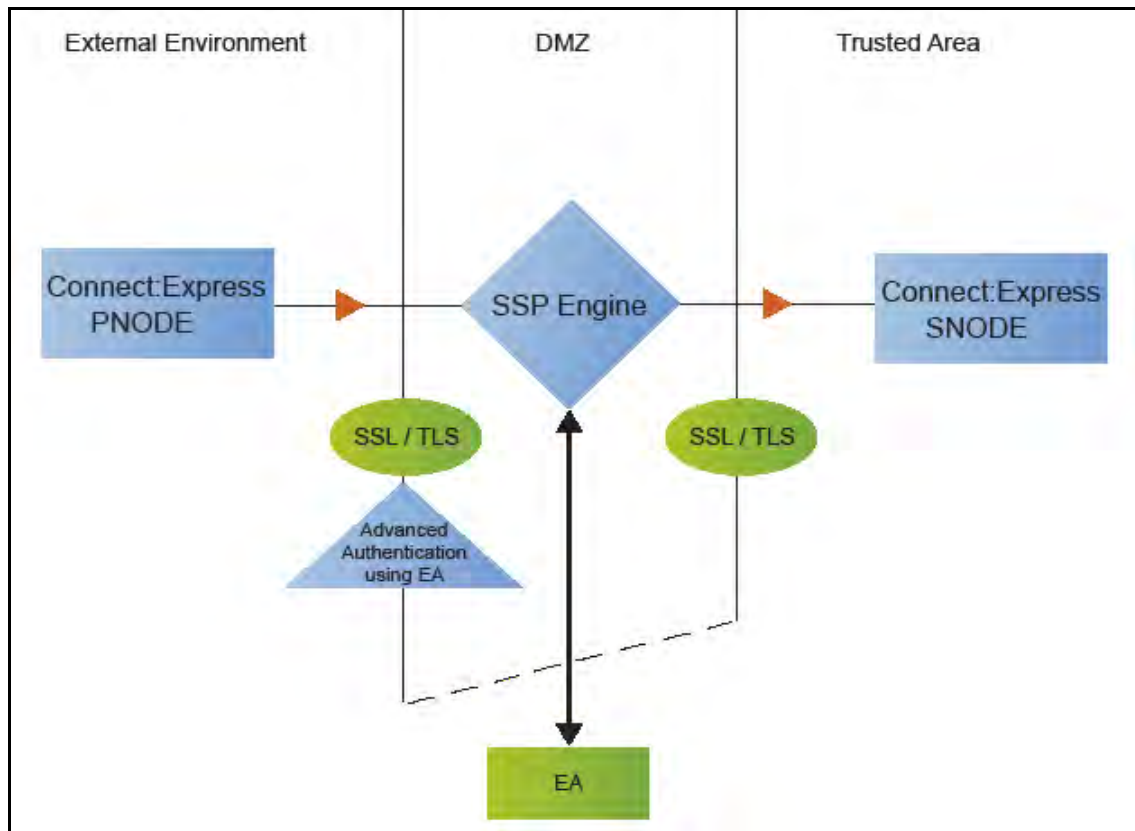
Authenticate an Inbound Certificate or LogonID Using EA

You can authenticate an inbound connection against information stored in an LDAP database by configuring EA to define how the connection is authenticated. The EA definition determines the options that are enabled. Refer to Sterling External Authentication Server help for a complete list of the functions that can be performed in EA.

Authenticate a Certificate or LogonID Using EA - Worksheet

Use the following worksheet to identify the information needed to authenticate a PeSIT connection using information in EA. Update the policy you created in the basic PeSIT configuration for this scenario.

Configuration Manager Field	Information	Value
Certificate Authentication - External Authentication Certificate Validation	Will you validate the certificate presented by the PNODE?	_____ (Yes or No)
Certificate Authentication - External Authentication Profile	If yes, provide the EA certificate validation definition.	_____
User Authentication - Through External Authentication	Will you validate user information?	_____ (Yes or No)
User Authentication - External Authentication Profile	If yes, provide the EA user validation definition.	_____

Authenticate a PeSIT Certificate or LogonID Using EA

To authenticate certificate information or logonID information about the PeSIT node against information stored in an LDAP database, you must configure EA. After you configure EA to enable certificate or logonID authentication, complete this procedure to configure SSP to use the authentication method you defined.

Before you configure SSP to use EA to authenticate a node connection, obtain the name of the EA definition.

In addition, ensure that the following procedures have been performed:

✦ The public keys for SSP have been sent to the EA server and imported into the EA keystore.

✦ The EA server connection has been configured in SSP.

To configure authentication of a PeSIT node using EA:

1. Select Configuration from the menu bar.

2. Expand the Policies tree and click the policy to modify.

3. On the Policy Configuration panel, click the Advanced tab.

4. Configure one or more of the following options:

    ◆ To validate the certificate presented by the node against information defined in EA, enable Certificate Authentication - External Authentication Certificate Validation and enter the name of the profile you defined in EA in the Certificate Authentication - External Authentication Profile field.

    ◆ To enable logonID authentication through EA, enable LogonID Authentication - Through External Authentication and type the name of the definition you defined in EA in the LogonID Authentication - External Authentication Profile field.

5. Deselect the Through Local User Store option.

6. Click Save.

You can now associate this policy with a PeSIT node where you want to perform logonID authentication using information stored in an LDAP database.

Strengthen the Connection to the SNODE With LogonID Mapping

This scenario builds on the basic PeSIT configuration by adding logonID mapping using information defined in Sterling External Authentication Server (EA). To provide a more advanced method of securing a PeSIT connection, use EA to map a PNODE logonID and password to login credentials stored in EA. The mapped login credentials are then used to connect to the SNODE.
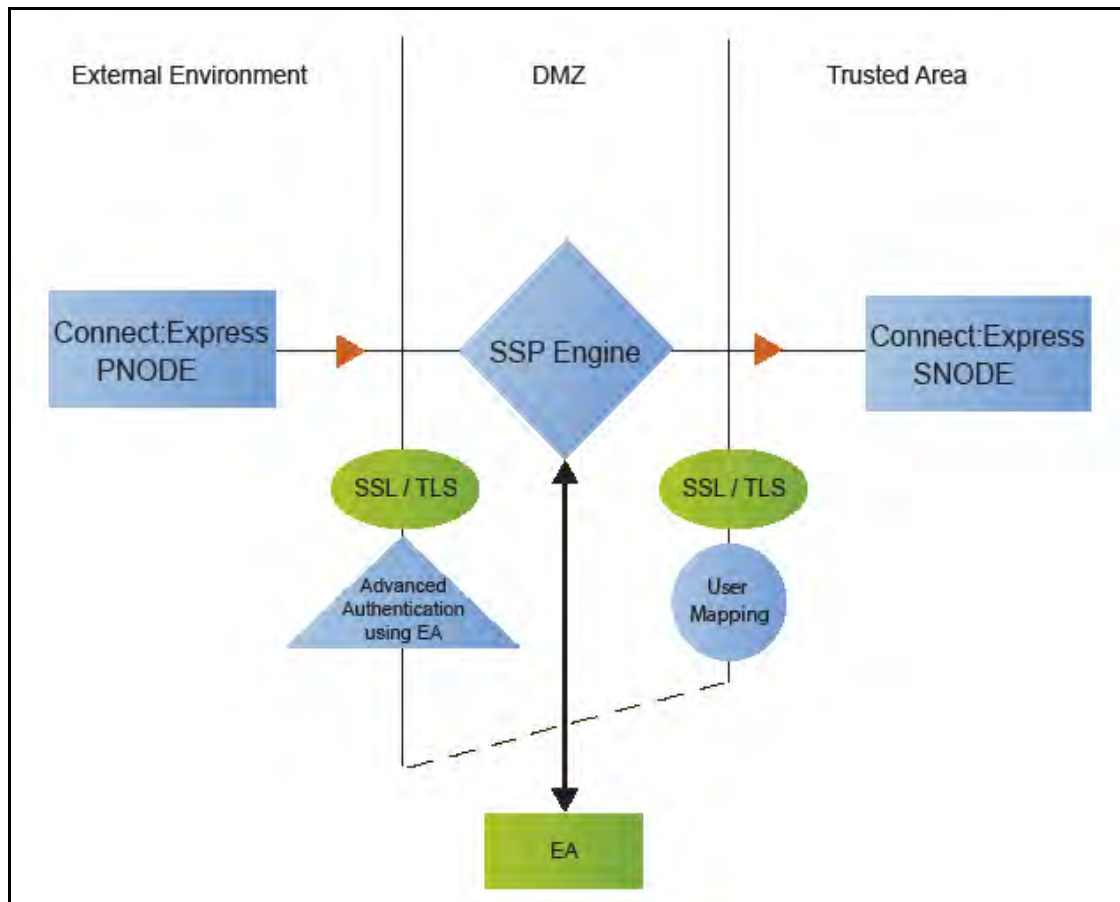
Perform LogonID Mapping Using EA - Worksheet

Use this worksheet to identify the logonID mapping method to enable for the SNODE connection with information in EA:

Configuration Manager Field	Feature	Value
Replace LogonID with Userid mapped in External Authentication	The PNODE requires a LogonID to access the SNODE. The LogonID provided is replaced with a value defined in EA.	Enabled
Destination Service Name	Name of the service. If no value is provided, the SNODE is used as the service name.	_____

Perform LogonID Mapping Using Information Stored in EA

If you store user credentials in an LDAP database, use this procedure to map a logonID and password to information stored in EA.

Destination Service Name needs to be selected on the Advanced tab of the Netmap Node screen of the PNODE. If Destination Service Name is not provided, the SNODE name is used.

Before you configure this option:

✦ Configure a definition in EA.

✦ Obtain the name of the EA definition.

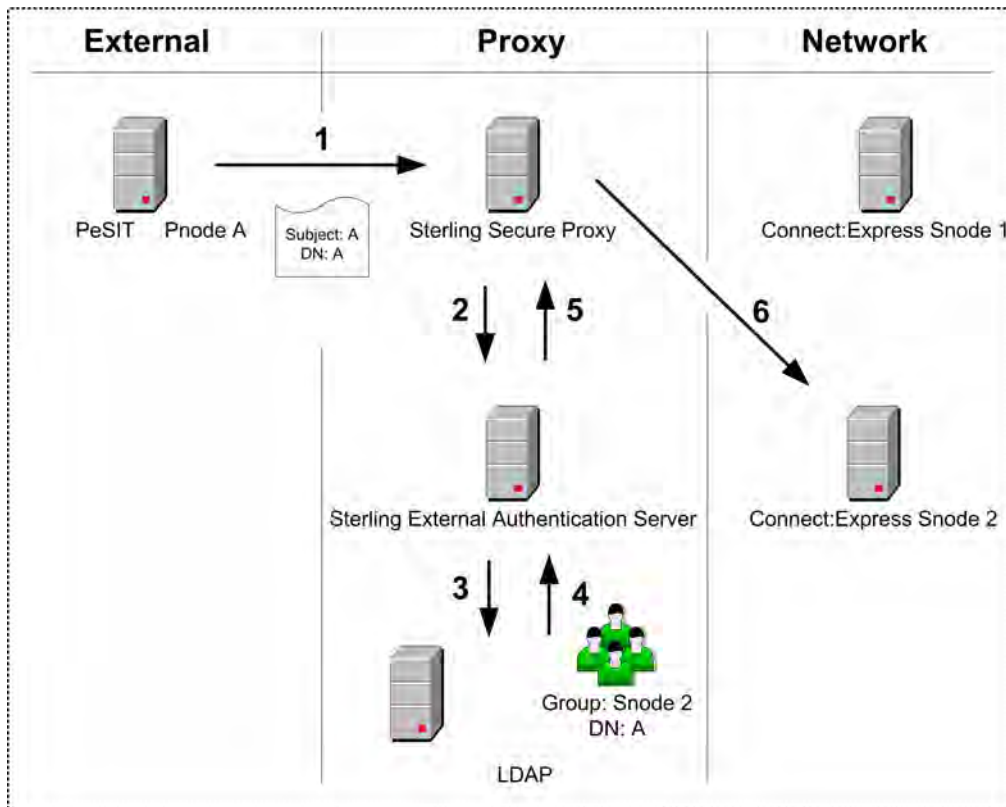✦ Configure a connection between EA and the engine.

To configure logonID mapping:

1. Select Configuration from the menu bar.

2. Expand the Policies tree and click the policy to modify.

3. On the Policy Configuration panel, click the Advanced tab.

4. To enable logonID authentication through EA, enable the LogonID Authentication Through External Authentication option and type the name of the definition you defined in EA in the External Authentication Profile field.

5. Select Replace LogonID with LogonID mapped in External Authentication.

6. Click Save.

7. In the Configuration panel, expand the Netmap option and click the netmap to modify.

8. Select the PNODE to modify and click Edit.

9. Click the Advanced tab.

10. Type the name of the service in the Destination Service Name field. If no value is provided, the SNODE name is used as the service name.

11. Click OK.

12. Click Save.

Configure Certificate-Based Routing

This scenario builds on the basic PeSIT configuration by configuring certificate-based routing. Certificate-based routing uses a routing name returned by EA. It is associated with the subject distinguished name found in the PNODE certificate. SSP uses this routing name to determine the SNODE where the incoming SSP connection is routed. To perform certificate-based routing, modify an adapter you defined in the basic PeSIT configuration.

The following diagram illustrates the certificate-based routing function:

Summary of Certificate-Based Routing

Following are the steps performed during certificate-based routing:

1. The PNODE passes a certificate chain during an SSL/TLS session. This certificate includes several attributes, such as subject and distinguished name (DN).

2. SSP passes the certificate chain to Sterling External Authentication Server (EA).

3. Using the configuration parameters in a certificate validation request, EA attempts to match PNODE certificate attributes to the LDAP server and requests the associated routing value.

4. LDAP returns the routing value to EA.

5. EA passes the routing value to the SSP engine.

6. SSP routes the PNODE request to the SNODE using the routing value.

Configure Certificate-Based Routing in SSP

Before you test certificate-based routing, you must create a certificate validation request in EA that includes an attribute query definition called Routing Names. This attribute query definition is created to retrieve a routing name value using certificate attributes as search criteria. You must also configure a connection between SSP and EA.

Refer to Chapter 12, *Configure SSP for Sterling External Authentication Server (EA)* for instructions.

To configure certificate-based routing:

1. Select Configuration from the menu bar.

2. Expand the Adapters tree and select the adapter you want to modify.

3. Select Certificate-based in the Routing Type field.

4. Click Save.

5. Click the Netmap navigation panel, expand the Netmap tree, and select the PeSIT adapter that contains the SNODE where the connection are routed.

6. Select the node to modify and click Edit.

7. Type the routing value to be returned from the LDAP server in the Routing Name field. The routing name must exactly match the routing value returned from the LDAP server. This routing name identifies the SNODE for routing the PNODE request.

8. Click OK.

9. Click Save.

10. Configure SSP to enable certificate authentication using EA. Refer to *Authenticate an Inbound Certificate or LogonID Using EA* on page 216.

Test the PeSIT Connections

To verify that the engine can receive and initiate communications sessions, you have to establish a connection between a Connect:Express PNODE and the engine, initiate a session from the engine to the Connect:Express SNODE in the trusted zone, and review the SSP log for the results.

This procedure enables you to verify that the engine can:

✦ Establish a Connect:Express session between a PNODE and SSP

✦ Initiate a session to a Connect:Express SNODE on behalf of the Connect:Express PNODE connection

To verify the communications sessions:

1. View the secureproxy.log.

2. Confirm that the sessions were established, as shown in the following example.

```
23 avr. 2009 13:39:37,459 INFO  [ProxyNearScheduler-Thread-9]
sys.ADAPTER.PeSITClearAdapter - protocol=pesit sessid=124048677744407 SSP103I
Session started from Peer Address: PNODE1.eur.sci.local/10.87.15.82
23 avr. 2009 13:39:37,459 INFO  [ProxyNearScheduler-Thread-9]
sys.ADAPTER.PeSITClearAdapter - protocol=pesit sessid=124048677744407 SSP104I
Session Proceeding after Node match: INSERVER
23 avr. 2009 13:39:37,459 INFO  [ProxyNearScheduler-Thread-9]
sys.ADAPTER.PeSITClearAdapter - protocol=pesit sessid=124048677744407 SSE1831I
Authentication mechanism: no authentication.
23 avr. 2009 13:39:37,475 INFO  [ProxyNearScheduler-Thread-3]
sys.ADAPTER.PeSITClearAdapter - protocol=pesit sessid=124048677744407 SSE0103I
Connecting to server.
23 avr. 2009 13:39:37,475 INFO  [ProxyNearScheduler-Thread-3]
sys.ADAPTER.PeSITClearAdapter - protocol=pesit sessid=124048677744407 SSP0237I
Attempting outbound connection with
10.20.129.3/InetSocketAddress-host:/10.20.129.3-port:4004 ...
23 avr. 2009 13:39:37,631 INFO  [ProxyFarScheduler-Thread-12]
sys.ADAPTER.PeSITClearAdapter - protocol=pesit sessid=124048677744407 SSP0238I
Primary outbound plain socket connection with
10.20.129.3/InetSocketAddress-host:/10.20.129.3-port:4004 established.
23 avr. 2009 13:39:44,444 INFO  [ProxyNearScheduler-Thread-13]
sys.ADAPTER.PeSITClearAdapter - protocol=pesit sessid=124048677744407 SSP102I
Control:ClientAgent Connection closed (CloseCode.EOF):  Elapsed Time: 6.985 (s)
> :  Bytes Received: 646 [at:  7.398711524695777E-4 MBPS]
> Bytes Sent: 402 [at:  4.604151753758053E-4 MBPS]
>
23 avr. 2009 13:39:44,444 INFO  [ProxyNearScheduler-Thread-13]
sys.ADAPTER.PeSITClearAdapter - protocol=pesit sessid=124048677744407 SSP0242I
Inbound connection from
GCAZENAVE.eur.sci.local/GCAZENAVE.eur.sci.local/10.87.15.82 closed. End of File.
23 avr. 2009 13:39:44,459 INFO  [ProxyFarScheduler-Thread-14]
sys.ADAPTER.PeSITClearAdapter - protocol=pesit sessid=124048677744407 SSP102I
Control:ServerAgent Connection closed (CloseCode.FULL_CLOSE):  Elapsed Time:
6.828 (s)
> :  Bytes Received: 402 [at:  4.710017574692443E-4 MBPS]
> Bytes Sent: 646 [at:  7.568834212067955E-4 MBPS]
>
23 avr. 2009 13:39:44,459 INFO  [ProxyFarScheduler-Thread-14]
sys.ADAPTER.PeSITClearAdapter - protocol=pesit sessid=124048677744407 SSP0244I
Outbound connection from
10.20.129.3/InetSocketAddress-host:/10.20.129.3-port:4004 closed. Connection
fully closed.
23 avr. 2009 13:39:44,459 INFO  [ProxyFarScheduler-Thread-14]
sys.ADAPTER.PeSITClearAdapter - protocol=pesit SSP119I Session Manager: Cleaning
out session with id: 124048677744407
23 avr. 2009 13:39:44,459 INFO  [ProxyFarScheduler-Thread-14]
sys.ADAPTER.PeSITClearAdapter - protocol=pesit sessid=124048677744407 SSE0112I
Session ended.
```

If your session was unsuccessful, review the log information to determine the likely cause of the failure and the corrective action to take.

Additional PeSIT Configuration Options

Additional PeSIT configuration options support the following features:

✦ Define alternate nodes for failover support

✦ Record an error message or shutdown a connection based on protocol errors

✦ Block PeSIT command from a PNODE

Define Alternate Nodes for Failover Support

If you are using standard routing to connect to a Connect:Express server in the secure zone, you identify a primary server to connect to in the adapter. The primary nodes are defined in the netmap. For each PNODE definition in the netmap, you can identify up to three alternate outbound nodes to connect to if the primary Connect:Express server is not available.

Two methods of configuring alternate server routing are available.

✦ Select a previously defined outbound node from the drop-down list on the Netmap - Advanced tab. To configure this method, you first configure an outbound node definition in the netmap for each alternate node you want to use. Each connection uses the security and External Authentication settings defined for that outbound node in the netmap.

✦ Select IP address/port from the drop-down Node list on the Advanced tab and enter values for the IP address and port. If you use this method, you do not have to define the alternate outbound nodes in the netmap, and each alternate connection shares the security and EA settings defined in the primary node definition.

If you configure alternate server definitions in the PNODE definition, when a connection to the primary outbound node is unsuccessful SSP tries to connect to the alternate node you defined as Node 1. If the connection to the first alternate node is unsuccessful, SSP tries to connect to the second alternate node, Node 2 and then to the third alternate, Node 3. If all are unsuccessful, the inbound connection fails.

To configure alternate outbound connections:

1. Select Configuration from the menu bar.

2. Expand the Netmaps tree and click the netmap to modify.

3. Select the node to modify and click Edit.

4. Click the Advanced tab.

5. Do one of the following:

   ◆ To identify an alternate node that is defined in the netmap and use its security settings, select the outbound node name from the drop-down list.

   ◆ To configure an alternate node that is not in the netmap and use the security settings defined in the primary node definition:

   a. Select Address/Port from the drop-down list in the Alternate Destinations Node field.

   b. Provide the IP Address and Port number for the alternate outbound node.

6. Click OK.

7. Click Save.

Record an Error Message or Shut Down a Connection Based on Protocol Errors

To write a warning message to the log file or shut down a connection when a protocol violation occurs during a file transfer, enable this function in the Policy definition.

To enable an action based on a protocol error:

1. Select Configuration from the menu bar.

2. Expand the Policies tree and select the policy to modify.

3. Select the action to take on a protocol error in the Protocol Error Action field.

4. Click Save.

Block PeSIT Command from a PNODE

This scenario builds on the basic PeSIT configuration by adding the capability to prevent a PeSIT command from being executed.

To prevent a PeSIT command from being executed:

1. Select Configuration from the menu bar.

2. Expand the Policies tree and click the policy to modify.

3. On the Policy Configuration panel, click the Transfer Directions tab.

4. Click on one of the following commands to disable the command:

   ◆ Receive a File Allowed (SELECT)

   ◆ Send a File Allowed (CREATE)

5. Click Save.

Change the Logging Levels

The following configuration events are written to the engine audit log:

✦ All fields from an initial engine configuration received from the Configuration Manager

✦ Changed fields from an engine configuration update from the Configuration Manager

✦ Inbound connections received for all protocols

✦ Inbound handshakes completed for the FTP, HTTP, PeSIT, and Connect:Direct protocols

✦ Inbound login successes and failures for the FTP, HTTP, and SFTP protocols

✦ Outbound connections established for all protocols

✦ Outbound handshakes completed for the FTP, HTTP, PeSIT, and Connect:Direct protocols

✦ Outbound login successes and failures for the FTP, HTTP, and SFTP protocols

When you configure a PeSIT node, the logging level for the node is set to None and no log is created. You can change the logging level to one of the following options: ERROR to write error messages, WARN to write error and warning messages, INFO to write error, warning, and informational messages, and DEBUG to write all messages to the log including debugging messages.

To change the logging level for a PeSIT node so that a log is created:

1. Click Configuration from the menu bar.

2. Expand the Netmaps tree and select the netmap where the PeSIT node to modify is defined.

3. Select a node to modify and click Edit.

4. Click the Advanced tab.

5. Select the logging level in the Logging level field.

6. Click Save.

Use Perimeter Servers to Manage PeSIT Communications

You can use a perimeter server with SSP to manage inbound and outbound PeSIT and Connect:Express communications. Configure perimeter servers for PeSIT and Connect:Express nodes the same way you configure perimeter servers for Connect:Direct nodes. Refer to Chapter 14, *Configure Perimeter Servers to Manage SSP Communications.*

Manage Your SSP Configuration

After you set up your SSP configuration, refer to Chapter 18, *Manage Your SSP Configuration*. The following sections describe PeSIT-specific considerations:

Modify Properties in an Adapter Definition

Adapters are configured with default settings. Use this procedure to modify a property. For FTP and HTTP adapters, the properties and default values are displayed. To change a property, type a new value for the property key. For SFTP, Connect:Direct, and PeSIT adapters, the properties are not displayed.

Refer to the field level help for a description of the properties. To change a property, type the property name and its key value.

To modify an adapter property:

1. Click Configuration from the menu bar.

2. Expand the Adapters tree and click the adapter to modify.

3. Click the Properties tab.

4. Click New to add a new property definition.

5. For each property, specify values for the following:

   ◆ Key

   ◆ Value

6. Click Save.

Copy a PeSIT Node

To quickly create a PeSIT node definition, you can copy an existing definition and make the changes necessary to create a new item.

To copy a PeSIT node:

1. Click Configuration from the menu bar.

2. Expand the Netmap tree and click the PeSIT netmap where the node is defined.

3. Click the radio button beside the node to copy and click Copy.

   A new node is created and renamed to Copyof*ItemName* where *ItemName* is the name of the original node you created.

4. Modify the node definition as necessary.

5. Click OK.

6. Click Save.

Sterling Secure Proxy Configuration Guide

Configure SSP for Sterling External Authentication Server (EA)

To provide a more advanced method of securing an inbound or outbound connection to SSP, use Sterling External Authentication Server (EA). EA allows you to authenticate certificate information or user credentials presented by the inbound node or to perform user ID and password mapping for the credentials used to attach to the outbound node.

EA Server Configuration - Worksheet

Before you begin configuring SSP for authentication options using EA, gather the information on this worksheet from the EA administrator. Collect this information for each EA server you will configure.

Configuration Manager Field	Value
EA Server Name	_____
EA Server Address	_____
EA Server Port	_____
Outbound Port Range	_____
Security Setting	_____ (SSL or TLS)
Trust Store	_____
CA/Trusted Certificates	_____

Configuration Manager Field	Value
Key Store	_____
Key/System Certificate	_____
Cipher Suites	_____
	_____
	_____
	_____

Configure an EA Server Connection

You can use EA to increase the security of your SSP environment. EA can be used to validate certificates from an inbound node, authenticate inbound users, and provide more secure credentials to the outbound node.

Before you can configure SSP to use EA, you must configure an EA server definition.

To configure an EA server definition:

1. Click Advanced from the menu bar.

2. Select Actions > New External Authentication Server.

3. Specify values for the following fields:

    ◆ EA Server Name

    ◆ EA Server Address

    ◆ EA Server Port

    ◆ Outbound Port Range

4. To enable SSL or TLS for the EA server connection, click the Security tab and enable Use Secure Connection.

5. Set the following values:

    ◆ Security Setting

    ◆ Trust Store

    ◆ CA/Trusted Certificates

    ◆ Key Store

    ◆ Key/System Certificate

    ◆ Cipher Suites

6. Click Save.

Specify Alternate EA Servers for Failover Support

You can specify alternate EA servers that SSP connects to if a connection to the primary EA server cannot be made. Up to three alternate EA servers can be defined for each EA server.

You must first configure an EA server connection for each EA server you want to identify for failover support. Then you can identify alternate EA servers to use if an EA server is not available by selecting an EA server definition from the list.

To specify an alternate EA server for failover support:

1. Click Advanced from the menu bar.

1. Expand the External Authentication Servers tree and select the EA server you want to edit.

2. Click the Advanced tab.

3. Select an alternate server from the Alternate EA Server #1 list.

4. Select additional servers as needed from the remaining lists. Connection attempts will be made to the alternate servers in the order in which they are specified.

5. Click Save.

Use a Perimeter Server to Connect to EA

You can configure EA to use a remote perimeter server in the trusted zone to manage connections to and from EA. This configuration enables you to have one outbound opening in your more trusted firewall. For more information on configuring and mapping perimeter servers, refer to Chapter 14, *Configure Perimeter Servers to Manage SSP Communications*.

Change Logging Levels

SSP provides the following log files:

✦ Audit Log
✦ Secure Proxy Log
✦ Node Logs
✦ Certicom Log
✦ Perimeter Server Log
✦ SFTP Logs

Audit Log

The audit log contains messages about system operations and events. You view the log for information about suspected misuse, and identify the user, application, or remote trading partner responsible for the misuse.

The audit log is created for both CM and the engine in the *install_dir*/logs/audit directory and is named auditlog.xml.

It is always enabled and provides log backups when the log reaches a preconfigured size. Audit log records can also be sent to a syslog daemon to be routed elsewhere for other processing.

Audit log records are formatted in XML and are written to a file with an .inc suffix. Another file with suffix .xml contains an XML prolog and epilog information. The two files together make up one version of the audit log.

When an audit log file reaches a predefined size, it is archived and saved as auditlog1.xml. If archive files have already been created, each archive file is renamed. For example, a log called auditlog3.xml is renamed to auditlog4.xml, the log auditlog2.xml is renamed auditlog3.xml, and so on. You configure the maximum number of archive files to maintain.

Audit log settings are configured in the log.properties file located in the *install_dir*/bin directory.

Audit Log Parameters

Following are the parameters that can be defined for an audit log in the log.properties file:

Parameter	Description
audit.log.filename	The location and file name to assign to an audit log. Default=../logs/audit/auditlog.xml.
audit.log.maxfilesize	The maximum size allowed in an audit log. When the maxfilesize is reached, the audit log is closed and a new log is opened. Default =500KB.
audit.log.maxbackupindex	The number of archive files to maintain. When the maximum file size is reached, it is closed and a new log is opened. Default=100.
audit.log.file.routing	Determines if the audit log is written to a file. y = write the log to a file. Default=y. n = do not create an audit log file. **Note:** If you configure the audit log to write to the syslog daemon, this parameter can be set to n. Otherwise, an audit log is written to a file, regardless of the value of this parameter.
audit.log.syslog.routing	Determines if the audit log is written to syslog. y = write the log to syslog. n = do not write the audit log to syslog. Default=n. You must configure a valid syslog.port and syslogd.host in order to write to syslog.
audit.log.syslog.facility	The facility number to associate with audit log messages. Default=18.

Audit Log Parameters to Enable SysLog Support

To route audit log content to a syslog in a UNIX or Linux environment, configure the following parameters in the log.properties file:

Parameter	Description
syslogd.enable	Enables or disables syslog daemon support. Available options include: y = enabled n = disabled n is the default setting.
syslogd.host	The name or IP address of the syslog host.
syslogd.port	Port of the syslog host. The default is 514.

Configuration Manager Audit Log Events

Following are the configuration events written to the Configuration Manager audit log:

✦ A list of all fields when you create a new configuration object.

✦ Changed fields of an updated configuration object.

✦ A list of all fields when you delete an object.

✦ All fields of a configuration pushed to an engine.

Engine Audit Log Events

Following are the configuration events written to the engine audit log:

✦ All fields from an initial engine configuration received from the Configuration Manager

✦ Changed fields from an engine configuration update from the Configuration Manager

✦ Inbound connections received for all protocols

✦ Inbound handshakes completed for the FTP, HTTP, and Connect:Direct protocols

✦ Inbound login successes and failures for the FTP, HTTP, and SFTP protocols

✦ Outbound connections established for all protocols

✦ Outbound handshakes completed for the FTP, HTTP, and Connect:Direct protocols

✦ Outbound login successes and failures for the FTP, HTTP, and SFTP protocols

Secure Proxy Log

Use the secure proxy log to troubleshoot SSP issues. Secure proxy logs are created for CM and the engine. The file is called secureproxy.log on the engine and cms.log on CM.

When a secure proxy log file reaches a predefined size, the current log is archived and the file name changed to secureproxy.log.1. If archive files already exist, each archive file is renamed by increasing the number appended to the file. For example, a log called secureproxy3.log is renamed to secureproxy4.log and the log secureproxy.log.2 is renamed secureproxy.log.3. The maximum number of archive files to maintain is configurable. Secure proxy log settings are configured in the log.properties file located in the *install_dir*/bin directory.

Secure Proxy Log Parameters

Following are the parameters that can be defined for a secure proxy log in the log.properties file:

Parameter	Description
proxy.log.file.routing	Determines if the secure proxy log is written to a file. y = write the log to a file. default=y. n = do not create a log file. **Note:** If you configure the secure proxy log to be written to the syslog daemon, this parameter can be set to n. Otherwise, a debug log is written to a file, regardless of the value of this parameter.
proxy.log.filename	The location and file name to assign to a debug log. The default value is ../logs/secureproxy.log
proxy.log.maxfilesize	The maximum size allowed in a debug log. When the maxfilesize is reached, the debug log is closed and a new log is opened. The default log file size is 50MB.
proxy.log.maxbackupindex	The number of archive files to maintain. If the number of archive files identified in this parameter is exceeded, the oldest archive file is deleted. The default value is 10.
proxy.log.level	The logging level for the secure proxy log. Default=INFO. This value can be set using CM.
proxy.log.syslog.routing	Determines if the secure proxy log is written to syslog. y = write the log to syslog. Default=n. n = do not write the debug log to syslog. You must configure a valid syslogd.port and syslogd.host in order to write to syslog.
proxy.log.syslog.facility	The facility number to associate with secure proxy log messages. The default value is 17.

Node Logs

You can turn on node-level logging to log sessions for a specific node. The node-level logs are named secureproxy-<*netmapName*>.<*nodeName*>.log where *netmapName* is the name of the netmap and *nodeName* is the name of the node for which activity is being logged.

When the session for a node ends, the node-level log file for the session is closed. A new session appends to the end of the node log file. Both inbound and outbound nodes log both sides of the connection. Enabling logging on one of the nodes captures end-to-end session events.

Certicom Log

Use the Certicom log to troubleshoot communications issues when using SSL or TLS. The file is called certicom.log.

Following are the parameters that can be modified for a Certicom log in the log.properties file:

Parameter	Description
certicom.log.file.routing	Determines if the certicom log is written to a file. y = write the log to a file. Default=y. n = do not create a log file. **Note:** If you configure the log to be written to the syslog daemon, this parameter can be set to n. Otherwise, a log is written to a file, regardless of the value of this parameter.
certicom.log.filename	The location and file name to assign to a log. Default=../logs/certicom.log.
certicom.log.maxfilesize	The maximum file size allowed for a certicom log. When the maximum file size is reached, the log is closed and a new log is opened. The default log file size is 100MB.
certicom.log.maxbackupindex	The number of archive files to maintain. If the number of archive files identified in this parameter is exceeded, the oldest archive file is deleted. The default value is 1.
certicom.log.level	The logging level for the certicom log. The default value is ERROR.
certocm.log.syslog.routing	Determines if the certicom log is written to syslog. y = write the log to syslog. n is the default setting. n = do not write the debug log to syslog. You must configure a valid syslogd.port and syslogd.host in order to write to syslog.
certicom.log.syslog.facility	The facility number to associate with Certicom proxy log messages. The default value is 17.

Perimeter Server Log

Perimeter server log information is written to a log file called perimeter.log. The default maximum size for the perimeter log is 100 MB.

When a log file reaches a predefined size, the log is renamed, and a new file is created. For example, an older log file called perimeter.log1 is renamed to perimeter.log2 and perimeter.log2 is renamed perimeter.log3. The maximum number of archive files to keep and the maximum file size can be configured in the log.properties file.

Perimeter Server Log Parameters That Can Be Configured

Perimeter server log parameters are defined in the log.properties file. You can change one or more of the following parameters:

Parameter	Description
perimeter.log.file.routing	Determines if the perimeter log is written to a file.
	y = write the log to a file. Default=y.
	n = do not create a perimeter log file.
	Note: If you configure the perimeter log to be written to the syslog daemon, this parameter can be set to n. Otherwise, a perimeter log is written to a file, regardless of the value of this parameter.
perimeter.log.filename	The location and file name to assign to a perimeter server log. Default=../logs/perimeter.log.
perimeter.log.maxfilesize	The maximum size allowed in a perimeter server log. When the maxfilesize is reached, the log is closed and a new log is opened. Default=100MB.
perimeter.log.maxbackupindex	The number of archive files to maintain. If the number of archive files identified in this parameter is exceeded, the oldest archive file is deleted. Default=1.
perimeter.log.level	The logging level for the perimeter log. The default value is ERROR. This value can be set using CM.
perimeter.log.syslog.routing	Determines if the perimeter log is written to syslog.
	y = write the log to syslog.
	n = do not write the log to syslog. Default=n.
	You must configure a valid syslogd.port and syslogd.host in order to write to syslog.
perimeter.log.syslog.facility	The facility number to associate with the perimeter log messages. Default=17.

SFTP Logs

If you configure SSP for an SFTP environment, two additional logs are maintained: a maverick log and an SFTP adapter log.

Maverick Log

The Maverick toolkit is used to manage communications in an SFTP environment. All protocol messages generated by the Maverick toolkit are written to a separate log called maverick.log. If you have problems communicating in an SFTP environment, view this log to help troubleshoot the issue.

The default size of the maverick.log file is 100MB. The maverick log is set up to maintain one archive file so that when the maverick.log file reaches 100MB, a new file is created and the archive file is renamed to maverick.log.1.

Following are the properties for the maverick log that you can change in the log.properties file:

Field	Description
maverick.log.filename	The location and file name to assign to a perimeter server log. Default= ../logs/maverick.log.
maverick.log.maxlogsize	The maximum size of a maverick log file before being archived. Default=100MB.
maverick.log.backupindex	The number of backup files to maintain. Default=1.
maverick.log.level	The logging level for writing to the maverick log file. Available options include: NONE, ERROR, WARN, INFO, and DEBUG. Default=INFO.

SFTP Adapter Log

A log is maintained for SFTP adapter activity. The file is called sftp.adapter<*adapterName*>.log where *adapterName* is the name of the adapter as configured in SSP.

The SFTP adapter log is set up to maintain 10 archive files so that when the log file reaches 50MB, a new file is created and the archive file is renamed to sftp.adapterAdapterA.log.1. If older versions exist, they are renamed first. For example, sftp.adapter<*adapterName*>.log is renamed to sftp.adapter<*adapterName*>.log.1 and sftp.adapter<*adapterName*>.log.2 is renamed to sftp.adapter<*adapterName*>.log.3. The maximum number of versions to keep is configurable in the log.properties file.

Following are the properties for the SFTP log that you can change in the log.properties file:

Field	Description
sftp.log.enable	Specifies whether SFTP adapter messages are written to a separate log. Valid values are true | false. Default=true. If this parameter is set to false, the adapter log information is written to the secure proxy log file.
sftp.log.filename	The location and file name to assign to an SFTP adapter log. Default= ../logs/sftp.adapter-*adaptername*.log where *adaptername* is the name assigned to the adapter in SSP.
sftp.maxfilesize	The maximum size of an SFTP log file before being archived. Default= 50MB.
sftp.log.maxbackupindex	The number of backup files to maintain. Default=10.

Change the Logging Level for an Engine

When you configure an engine, the logging level for the engine is set to Error by default. Error logging level writes all error messages for the engine to the log.

To change the logging level for an engine:

1. If necessary, click Configuration from the menu bar.

2. Expand the Engines tree to view the list of available engines and select the engine to modify.

3. Click the Advanced tab.

4. Select the logging level in the Engine Logging Level field.

   The logging levels are ERROR to write only error messages, WARN to write error and warning messages, INFO to write error, warning, and informational messages, and DEBUG to write all messages to the log including debugging messages.

5. Click Save.

Change the Logging Level for CM

When you configure a CM, the logging level is set to Info by default.

To change the logging level for a CM:

1. Select System from the menu bar.

2. Expand the System Settings tree and click CMSystemSettings.

3. Click the Globals tab.

4. Select the logging level in the Logging level field.

   The logging levels are ERROR to write only error messages, WARN to write error and warning messages, INFO to write error, warning, and informational messages, and DEBUG to write all messages to the log including debugging messages.

5. Click Save.

Change the Logging Level for a Connect:Direct Node

When you configure a Connect:Direct node, the logging level for the node is set to None and no log is created. You can change the logging level to one of the following options: ERROR to write error messages, WARN to write error and warning messages, INFO to write error, warning, and informational messages, and DEBUG to write all messages to the log including debugging messages.

To change the logging level for a Connect:Direct node so that a log is created:

1. If necessary, click Configuration from the menu bar.

2. Expand the Netmaps tree and select the netmap where the Connect:Direct node to modify is defined.

3. Select a node to modify and click Edit.

4. Click the Advanced tab.

5. Select the logging level in the Logging level field.

6. Click Save.

Change the Logging Level for an Inbound Node

When you configure an inbound node for the HTTP, FTP, or SFTP protocol, the logging level for the node is set to None and no log is created for the node. You can change the logging level to one of the following: ERROR to write error messages, WARN to write error and warning messages, INFO to write error, warning, and informational messages, and DEBUG to write all messages to the log including debugging messages.

To change the logging level for an inbound HTTP, FTP, or SFTP node:

1. If necessary, click Configuration from the menu bar.

2. Expand the Netmaps tree and click the netmap where the inbound node to modify is defined.

3. Select the inbound node to modify and click Edit.

4. Click the Advanced tab.

5. Select the logging level in the Logging level field.

6. Click Save.

Change the Logging Level for an Outbound Node

When you configure an outbound node for the HTTP, FTP, or SFTP protocol, the logging level is set to None and no log is created for the node. You can change the logging level to one of the following: ERROR to write error messages, WARN to write error and warning messages, INFO to write error, warning, and informational messages, and DEBUG to write all messages to the log including debugging messages.

To change the logging level for an outbound HTTP, FTP, or SFTP node:

1. From the Configuration navigation panel, click Netmap to expand the list of available netmaps.

2. Click the netmap where the outbound node to modify is defined.

3. Click the Outbound Node tab.

4. Select an outbound node to modify and click Edit.

5. Click the Advanced tab.

6. Select the logging level in the Logging level field.

7. Click Save.

Change the Logging Level for a Local Perimeter Server

When you configure an engine, the logging level for the local perimeter server is set to Error by default. Error logging level writes all error messages for the local perimeter server to the log. You can change the logging level to one of the following options: ERROR to write error messages, WARN to write error and warning messages, INFO to write error and informational messages, and DEBUG to write all messages to the log including debugging messages.

To change the logging level for a local perimeter server:

1. If necessary, click Configuration from the menu bar.

2. Expand the Engines tree and click the engine to modify.

3. Click the Advanced tab.

4. Select the logging level in the Local Perimeter Server Logging Level field.

5. Click Save.

Chapter 14

Configure Perimeter Servers to Manage SSP Communications

A perimeter server is used by SSP to manage inbound and outbound TCP communication. This software tool enables you to manage the communications flow between outer layers of your network and the TCP-based transport adapters. Perimeter servers can be used to restrict areas where TCP connections are initiated: from more secure areas to less secure areas.

During the SSP installation, a perimeter server is installed. This perimeter server is referred to as the local perimeter server. You can use this default local perimeter server to restrict connections or you can install other perimeter server instances as needed. You can install additional perimeter servers on different computers or you can install different instances on the same computer, if you want to use different network cards for inbound and outbound traffic. A perimeter server requires a perimeter server definition in SSP.

After you install and configure a remote perimeter server, you need to map how the perimeter server is used: inbound, outbound, or External Authentication. Refer to *Map Perimeter Servers* on page 249.

Before you configure remote perimeter servers in SSP, complete the installation procedures outlined in Chapter 5, *Install a Remote Perimeter Server* in the *Sterling Secure Proxy Installation Guide*.

Typical Installation

The following figure illustrates a typical SSP installation with perimeter servers:
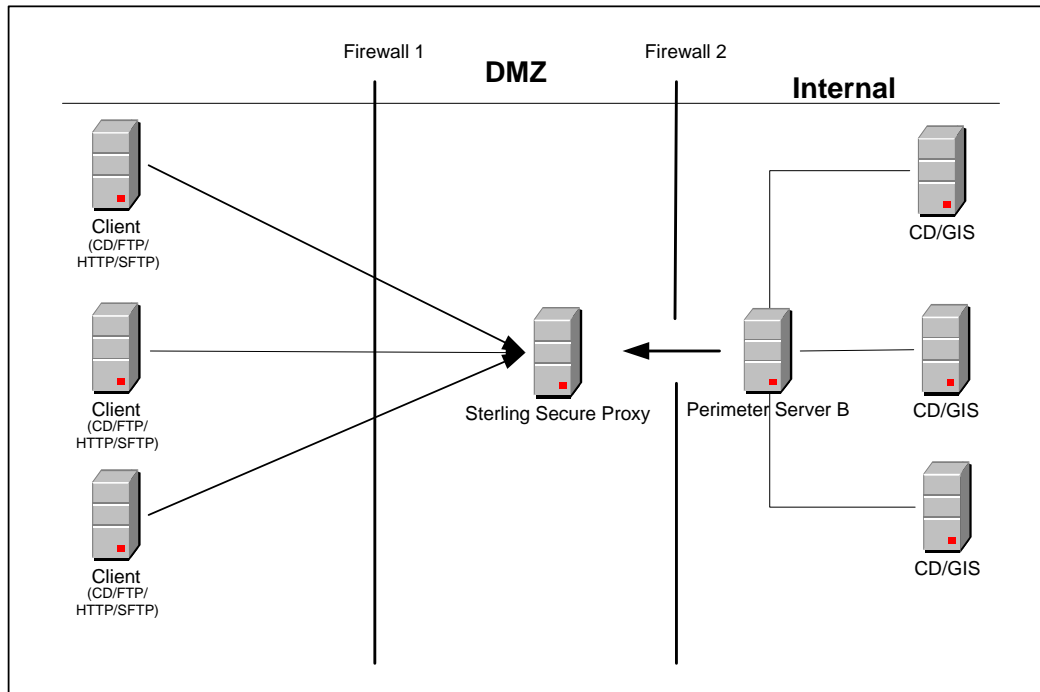
The preceding figure shows the following:

1. The persistent connection is established from the perimeter server in the internal trusted network to SSP in the DMZ. This allows for only an outbound hole to be configured in the Firewall 2 (no inbound hole is needed with this configuration)

2. SSP has an HTTP server adapter configured for two scenarios, one secure HTTP (HTTPS) and the other non-secure HTTP.

3. Two trading partners with separate host and port numbers are configured to communicate with SSP.

A perimeter server and all adapters that communicate with the local perimeter server must be configured on the same SSP engine. An engine can have more then one perimeter server but a perimeter server can be used by only one engine.You can configure a perimeter server for one trading partner with large files and low transaction volume, and another perimeter server on the same engine for a different trading partner with smaller files and high transaction volume. By configuring each perimeter server according to the trading partner, you increase SSP performance.

Sample Remote Perimeter Server Configurations

Use remote perimeter servers with SSP if you want to:

✦ Eliminate an inbound hole in your firewall to allow connections from less secure to more secure areas.

✦ Send data to your customers from the perimeter server as the originating IP address.

✦ Use different network cards for inbound and outbound traffic.

✦ Implement multiple DMZ scenarios. You can use perimeter servers in your outer DMZ with SSP in the internal DMZ.

You have flexible deployment options for using perimeter servers with SSP: from a simple IP break to no inbound holes in the firewall. Following are sample deployment options.

Deployment Option Example—Two Remote Perimeter Servers on a Computer with Two NIC Cards

The following sample illustrates a configuration where two remote perimeter servers are installed. One remote perimeter server manages inbound traffic and the other manages outbound traffic.
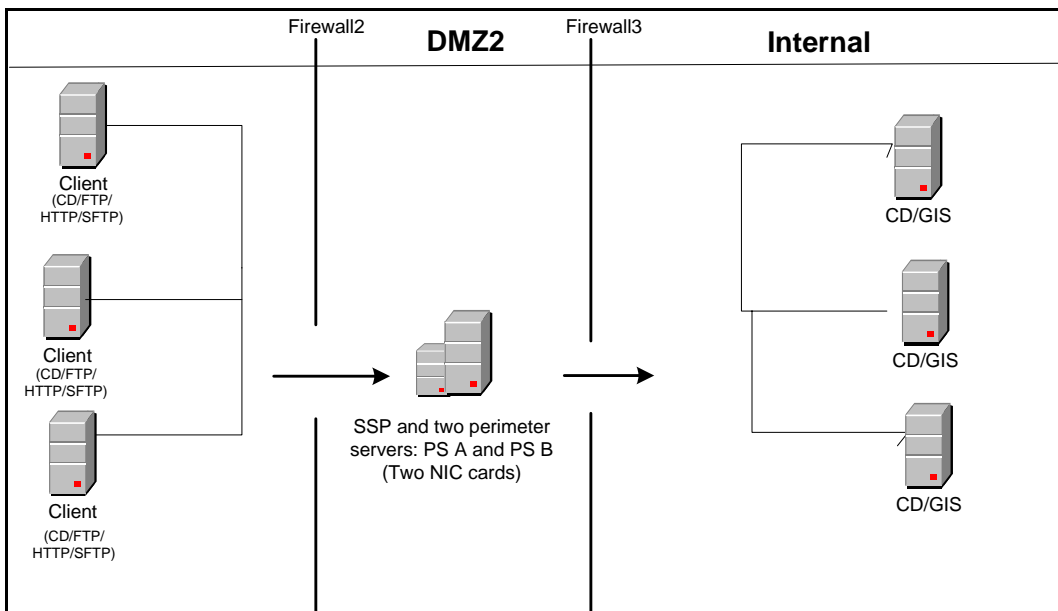
In this configuration, the firewall is configured to allow connections from trading partners to the remote perimeter server A. Remote PS A then routes traffic to SSP. Outbound traffic is routed from SSP through remote PS B to the GIS or Connect:Direct server.

Deployment Option Example—From More Secure to Less Secure

For additional firewall security, you can install a perimeter server in a more secure area than SSP and set up your firewall to allow only connections initiated from a more secure area to a less secure area. The following diagram demonstrates a configuration with two perimeter servers:
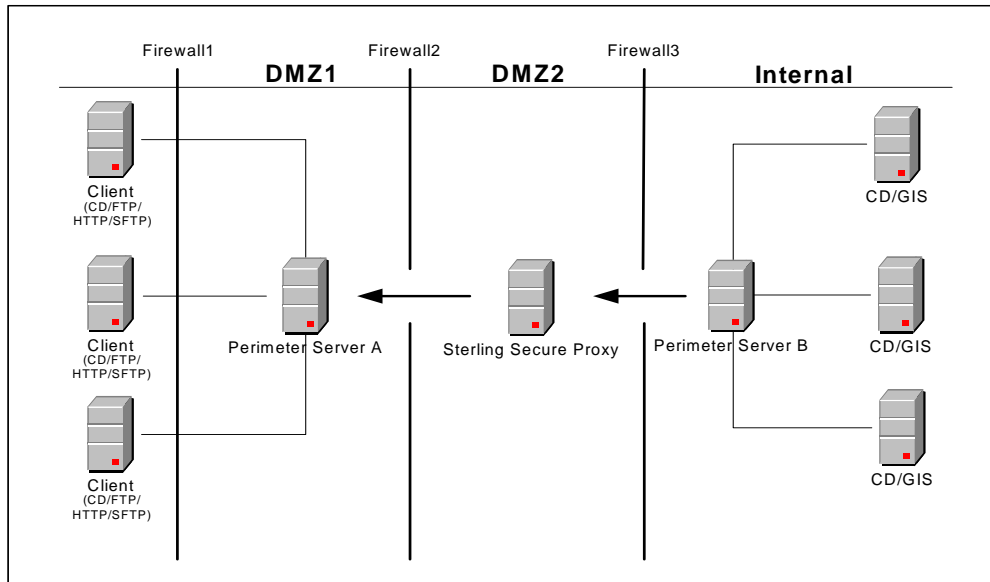
In this example, SSP is configured to use two perimeter servers that reside on remote computers: one on an external network (Perimeter Server A), and one in the internal network (Perimeter Server B). The firewalls are configured to allow only connections initiated from inside a more secure area (only an outbound hole in the firewall). When SSP is started, a connection is established from Perimeter Server B to SSP and from SSP to Perimeter Server A. Through these communication lines, SSL/TLS sessions can be established between clients and SSP, and between SSP and Connect:Direct or GIS.

Deployment Option Example—From Less Secure to More Secure

The following sample illustrates a configuration where the remote perimeter server is installed in a more secure location than SSP:
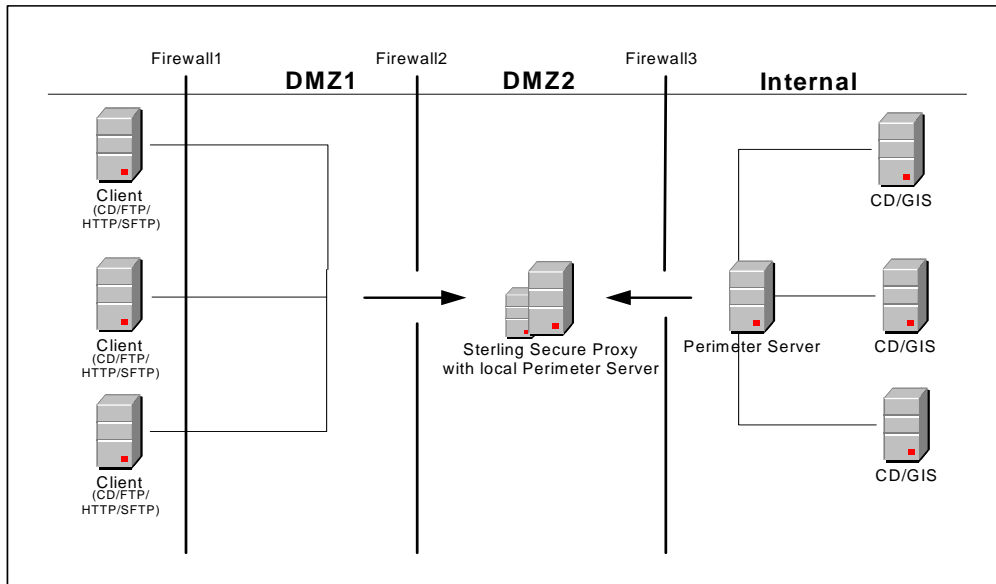
In this configuration, the firewall is configured to allow connections from external trading partners to SSP and from the internal perimeter server to SSP. In this configuration, traffic is moving from a less secure to a more secure location. To restrict unauthorized access, you can limit the perimeter server to perform only those activities required for SSP operations. Refer to *Install a Remote Perimeter Server* in the *Sterling Secure Proxy Installation Guide* for more information.

Deployment Option Example —External Authentication Perimeter Server

The following sample illustrates a configuration where the remote perimeter server in the trusted zone is used to connect to EA:
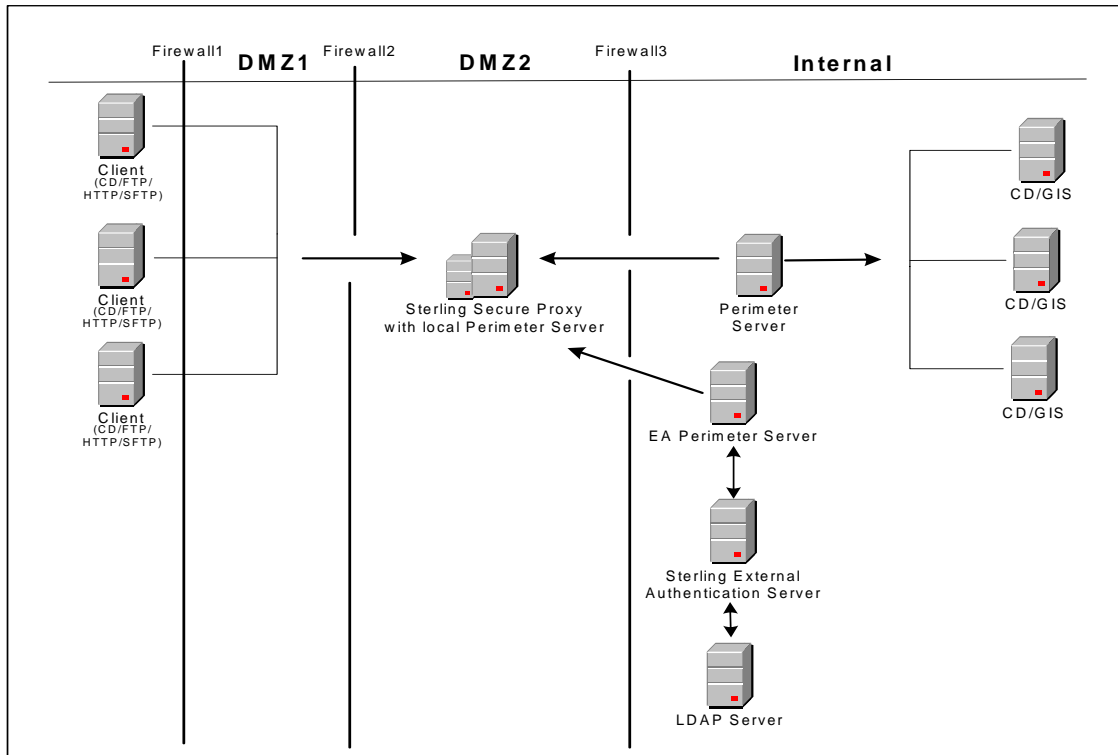
In this example, SSP is configured to use two remote perimeter servers and the local perimeter server. When SSP is started, a connection is established from SSP to the remote perimeter server. Through this communication line, SSL/TLS sessions can be established between clients and SSP. Another remote perimeter server is used to communicate between EA and SSP.

Define a Remote Perimeter Server for a Less Secure Environment

A common network configuration pattern is for SSP to reside in the innermost, secure network zone and the perimeter server to reside in the DMZ. In this case the connection should be established from SSP to the perimeter server—that is, from the more secure towards the less secure network zone.

Configure a Remote Perimeter Server in a Less Secure Zone

To configure a perimeter server in a less secure zone:

1. Select Advanced from the menu bar.

2. Select Actions > New Perimeter Server > Less Secure Zone.

3. Specify the following values:

   ◆ Perimeter Server Name

   ◆ Perimeter Server Host

   ◆ Perimeter Server Port

4. Click Save.

Edit a Remote Perimeter Server in a Less Secure Zone Definition

To edit the definition of a perimeter server in a less secure zone:

1. Select Advanced from the menu bar.

2. Expand the Perimeter Servers tree and expand the Less Secure Zone tree.

3. Click the perimeter server definition you want to edit.

4. Edit the values as needed.

5. Click Save.

Modify the Water Mark Values and Local Host Information of a Remote Perimeter Server in a Less Secure Zone

To modify the water mark values and local host information of a perimeter server in a less secure zone:

1. Select Advanced from the menu bar.

2. Expand the Perimeter Server tree and expand the Less Secure Zone tree.

3. Select the perimeter server to edit.

4. Click the Advanced tab.

5. Change the following values as needed:

   ◆ Perimeter Server Outbound Low Water Mark

   ◆ Perimeter Server Outbound High Water Mark

   ◆ Perimeter Server Inbound Low Water Mark

   ◆ Perimeter Server Inbound High Water Mark

   ◆ Proxy Local Interface

   ◆ Proxy Local Port

6. From the Perform DNS Resolution list, select the place where DNS resolution occurs.

7. Click Save.

Configure and Edit a Remote Perimeter Server Definition When Installed in a More Secure Network

In some cases, it is desirable for SSP to communicate with a perimeter server installed in a more secure network zone. In this case establish the network connection from the perimeter server to SSP.

Configure a Remote Perimeter Server in a More Secure Zone

To configure a perimeter server in a more secure zone:

1. Select Advanced from the menu bar.
2. Select Actions > New Perimeter Server > More Secure Zone.
3. Specify the following values:
    - Perimeter Server Name
    - Proxy Local Listen Port
4. Click Save.

Edit A More Secure Zone Remote Perimeter Server Definition

To edit a more secure zone perimeter server definition:

1. Select Advanced from the menu bar.
2. Expand the Perimeter Servers tree and expand the More Secure Zone tree.
3. Click the perimeter server definition to edit.
4. Edit the values as needed.
5. Click Save.

Modify Water Mark Values and Local Host Information of a Remote Perimeter Server Installed in a More Secure Zone

To modify water mark values and local host information of a perimeter server in a more secure zone:

1. Select Advanced from the menu bar.
2. Expand the Perimeter Server tree and expand the More Secure Zone tree.
3. Select the perimeter server to edit.
4. Click the Advanced Tab.
5. Change the following values as needed:
    - Perimeter Server Outbound Low Water Mark
    - Perimeter Server Outbound High Water Mark
    - Perimeter Server Inbound Low Water Mark
    - Perimeter Server Inbound High Water Mark
    - Proxy Local Interface

6. From the Perform DNS Resolution list, select the place where DNS resolution occurs.

7. Click Save.

Map Perimeter Servers

After you configure perimeter servers, map how they are used by each adapter: inbound perimeter server, outbound perimeter server, or EA perimeter server.

To map perimeter servers:

1. If necessary, click Configuration from the menu bar.

2. Expand the Adapters tree and select the adapter you want to edit.

3. Click the Advanced tab.

4. Select a perimeter server for each of the following as needed. The default is local.

   ◆ Inbound Perimeter Server

   ◆ Outbound Perimeter Server

   ◆ External Authentication Perimeter Server

5. Click Save.

Repeat this process for each adapter that uses a remote perimeter server.

Note: If you change the perimeter server mapped to an adapter, you must restart the adapter and the perimeter server before the change is enabled.

Modify Perimeter Server Properties

Two property values are defined in the perimeter.properties file located in the *install_dir*/bin folder. These properties determine SSL Session caching. Modify the following properties as necessary:

Parameter	Description
SslSessionDatabaseTimeoutSeconds	How long a cached SSL session is valid.
	The valid range is 30 seconds to 24 hours (60*60*24 = 86400 seconds). Default=1 hour or 3600 seconds.
SslSessionDatabaseSize	Maximum number of sessions to cache. This parameter is used by FTP and HTTP reverse proxy adapters. SSL sessions are not cached for Connect:Direct proxy adapters.
	Valid range is 1024 to 16384. Default=4096.

Start and Stop Remote Perimeter Servers

Use the procedures in this chapter to start and stop a remote perimeter server.

Start a Perimeter Server on UNIX or Linux

To start a perimeter server on UNIX or Linux:

1. Change the directory to /*install_dir*/bin where *install_dir* is the directory where the perimeter server is installed.

2. Type startupPs.sh and press **Enter**.

Stop a Perimeter Server on UNIX or Linux

To stop a perimeter server:

1. Change the directory to /*install_dir*/bin where *install_dir* is the directory where the perimeter server is installed.

2. Type stopPs.sh and press **Enter**.

Start Perimeter Servers in a Windows Environment

To start a perimeter server:

1. Change to the installation directory where the perimeter server is installed.

2. Type startPSService.cmd to start the perimeter server.

Stop a Perimeter Server on Windows

The remote perimeter server is installed as a Windows service. You can stop the remote perimeter server using the Windows service option or you can stop the perimeter server from the command line.

To stop a perimeter server on Windows from the command line:

1. Change the directory to *install_dir*\bin where *install_dir* is the directory where the perimeter server is installed.

2. Type stopPSService.cmd.

Prepare for Production

After you configure SSP and test to ensure that connections are working, you are ready to move to a production environment. This section describes production considerations.

Configure SSP to Interface with a Load Balancer

If you configure a Connect:Direct or HTTP environment, you can define an HTTP ping response to perform a health check, such as when using a load balancer tool. If you define these options, you can create a configuration for a BigIP connection and perform different levels of security checks.

Following are some possible scenarios for configuring tools like BigIP to monitor the status of the HTTP or Connect:Direct proxy adapter. The scenarios are presented in increasing order of security.

- ◆ **Simple health check**—In this scenario the monitoring agent makes a TCP connection to the listening port of the adapter and immediately disconnects. A successful connection indicates that the adapter is running. This health check has the least effect on performance.

- ◆ **Medium health check**—The monitoring agent sends an HTTP GET request with a specific URI. If the information matches the ping URI specified in the HTTP Reverse Proxy adapter, the adapter responds with the configured ping response. This allows the monitoring agent to determine that the adapter is alive and responsive.

- ◆ **Comprehensive health check**—The request from the monitoring agent is sent all the way to the GIS HTTP server via the HTTP proxy adapter. To allow this connection, either the URI of the GET request sent by the monitoring agent should not match the ping URI specified in the adapter configuration, or the ping URI in the adapter configuration should be empty. In either case, the adapter passes the request to the GIS server or to another SSP in the chain, depending upon the configuration. It is the responsibility of the monitoring agent and the backend server to ensure that the ping URI and response match.

Modify the Node-Level TCP Timeout Value in a Connect:Direct Node

TCP timeout identifies the maximum number of seconds SSP waits for a TCP buffer when communicating with a Connect:Direct node. For inbound sessions, this field is used after the first buffer is received from the remote node and the connecting node is identified. For outbound sessions from the proxy, this field is used from the start of the session. The default value is 90 seconds. Use this procedure to modify the TCP timeout value.

To modify the TCP timeout value in a Connect:Direct node:

1. From the Configuration navigation panel, click Netmap to expand the list of available netmaps.
2. Click the netmap where the node is defined.
3. Click the radio button beside the node you want to modify and click Edit.
4. Click the Advanced tab.
5. Change the value in the TCP timeout field.
6. Click OK.
7. Click Save.

Manage Certificates Between SSP Components

To maintain security in SSP, the engine and Configuration Manager (CM) communicate using SSL. SSP uses TCP/IP communications links between the web browser and the Jetty web server, the web server and CM, and CM and the engine. The only link that can be unsecure is between the web browser and the Jetty web server.

When you install SSP, a default certificate is installed to allow you to communicate. All components of the SSP system including CM, engine, and the Jetty web server share the same certificate. This self-signed certificate is called the factory certificate and has a ten year expiration.

Before you can begin production, you must import a secure certificate. The default configuration uses a single key to secure the connection between the engine and CM. The certificate distribution looks like this:
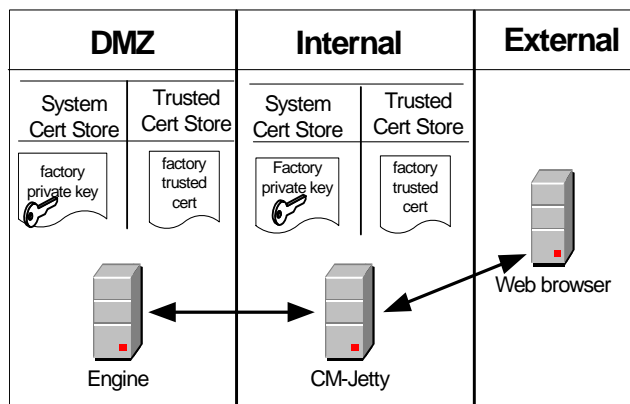
To secure the communication between these components, replace the factory certificates using one of the models in this chapter.

Use a Common Certificate for the Engine and CM

The simplest way to update the certificate distribution is to replace the factory certificate with a new certificate and use that certificate for both the engine and CM. The certificate distribution looks like this:
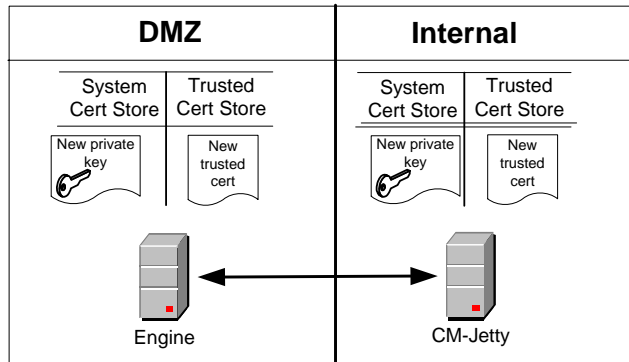
Replace the Factory Certificate with a Common Certificate on UNIX or Linux

To replace the factory certificate used between the engine and CM on UNIX or Linux:

1. Stop CM.

    a. At CM, navigate to the *install_dir*/bin directory, where *install_dir* is the installation directory, and type the following command:

    ```
    ./stopCM.sh
    ```

    b. At the passphrase prompt, type the passphrase defined for the engine and press **Enter**.

    c. Type the user name and password for the administrator.

2. Type the following command to replace the factory certificate:

    ```
    ./configureCmSsl.sh -u commonCert=<cert file> commonCertAlias=<alias>
    ```

    where:

    - ◆ *<cert file>* is the path and file name to the certificate that replaces the factory certificate.

    - ◆ *<alias>* is an alias name for the new certificate. This can be any value other than **factory**. If you do not specify an alias, **common** is assigned as the default.

3. Type the following command to create an export file of the certificate store:

    ```
    ./configureCmSsl.sh -e file=<export file>
    ```

    where *<export file>* is the path and file for the export file.

4. Copy the file you created in step 3 to the engine.

5. Stop the engine.

   a. At the engine, navigate to the *install_dir*/bin directory and type the following command:

   ```
   ./stopEngine.sh
   ```

   b. Type the passphrase defined for the engine and press **Enter**.

6. At the engine, navigate to the *install_dir*/bin directory and type the following command to import the certificate store created in step 3:

   ```
   ./configureEngineSsl.sh -i file=<export file> engCertAlias=<alias>
   ```

   where:

   - *<export file>* is the path and file for the export file.
   - *<alias>* is the alias name for the new certificate assigned in step 2

7. Start the engine.

   a. From *install_dir*/bin, type the following command:

   ```
   ./startEngine.sh
   ```

   b. At the passphrase prompt, type the passphrase for the engine and press **Enter**.

8. Start CM.

   a. Navigate to the *install_dir*/bin directory and type the following command:

   ```
   ./startCM.sh
   ```

   b. At the passphrase prompt, type the passphrase defined for CM and press **Enter**.

Replace the Factory Certificate with a Common Certificate on Windows

To replace the factory certificate used between the engine and CM on Windows:

1. Stop CM on Windows from Windows services.

2. Using a command line interface on CM, navigate to the *install_dir*\bin directory.

3. Type the following command to replace the factory certificate:

   ```
   configureCmSsl –u commonCert=<cert file> commonCertAlias=<alias>
   ```

   where:

   - *<cert file>* is the path and file name to the certificate that replaces the factory certificate.
   - *<alias>* is an alias name for the new certificate. This can be any value other than **factory**. If you do not specify an alias, **common** is assigned as the default.

4. Type the following command to create an export file of the certificate store:

```
configureCmSsl -e file=<export file>
```

where *<export file>* is the path and file for the export file.

5. Copy the file you created in step 3 to the engine.

6. Stop the engine on Windows from Windows services.

7. Using a command line at the engine, navigate to the *install_dir*\bin directory and type the following command to import the certificate store created in step 4.

```
configureEngineSsl -i file=<export file> engCertAlias=<alias>
```

where:

◆   *<export file>* is the path and file for the export file.

◆   *<alias>* is the alias name for the new certificate assigned in step 4.

8. Start the engine on Windows from Windows services.

9. Start CM on Windows from Windows services.

Use Different Certificates for the Engine and CM

You can use different certificates to secure the engine-to-CM connection and to secure the Jetty web server-to-CM connection. This certificate distribution is illustrated below:
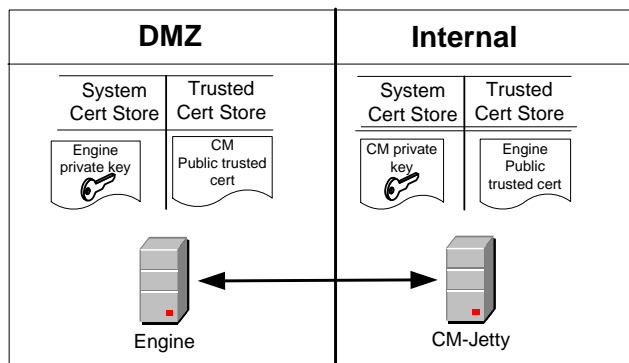
Replace the Factory Certificate with an Engine Certificate and CM Certificate on UNIX or Linux

To replace the factory certificates, with one certificate at the engine and a different certificate at CM on UNIX or Linux:

1. Stop CM.

   a. Navigate to the CM *install_dir*/bin directory, where *install_dir* is the installation directory, and type the following command:

   ```
   ./stopCM.sh
   ```

   b. At the passphrase prompt, type the passphrase defined for the engine and press **Enter**.

   c. Type the user name and password for the administrator.

2. From *install_dir*/bin, type the following command to replace the factory certificate with a CM certificate:

   ```
   ./configureCmSsl.sh -u cmCert=<cert file> cmCertAlias=<alias>
   ```

   where:

   - *<cert file>* is the path and file name to the certificate that replaces the factory certificate for CM.

   - *<alias>* is the alias name for the new CM certificate. It can be any value other than **factory**. If you do not specify an alias, "cm" is assigned as the default.

3. On the CM computer, type the following command to replace the factory certificate with an engine certificate:

   ```
   ./configureCmSsl.sh -u engCert=<cert file> engCertAlias=<alias>
   ```

   where:

   - *<cert file>* is the path and file name to the certificate you want to use to replace the factory certificate for the engine.

   - *<alias>* is the alias name for the new engine certificate. This can be any value other than **factory**. If you do not specify an alias, "engine" is assigned as the default.

4. Type the following command to create an export file of the certificate store:

   ```
   configureCmSsl -e file=<export file>
   ```

   where *<export file>* is the path and file for the export file.

5. Copy the file you created in step 4 to the engine.

6. Stop the engine.

   a. On the engine, navigate to the *install_dir*/bin directory and type the following command:

   ```
   ./stopEngine.sh
   ```

   b. At the passphrase prompt, type the passphrase defined for the engine and press **Enter**.

7. From the *install_dir*/bin directory, type the following command to import the certificates created in step 2 and step 3.

   ```
   configureEngineSsl -i file=<export file> engCertAlias=<alias>
   ```

   where:

   ◆ *<export file>* is the path and file for the export file.

   ◆ *<alias>* is an alias name for the engine certificate assigned in step step 3. If an engCertAlias was omitted in step 3, specify *engine* as the alias.

8. Start the engine.

   a. Type the following command:

   ```
   ./startEngine.sh
   ```

   b. At the passphrase prompt, type the passphrase for the engine and press **Enter**.

9. Start CM.

   a. Navigate to the *install_dir*/bin directory and type the following command:

   ```
   ./startCM.sh
   ```

   b. At the passphrase prompt, type the passphrase defined for CM and press **Enter**.

Replace the Factory Certificate with an Engine and CM Certificate on Windows

To replace the factory certificates, with one certificate at the engine and a different certificate at CM on UNIX or Linux:

1. Stop CM on Windows from Windows services.

2. Using a command line interface at CM, navigate to the *install_dir*\bin directory.

3. Type the following command to replace the factory certificate with a CM certificate:

```
configureCmSsl -u cmCert=<cert file> cmCertAlias=<alias>
```

where:

* *<cert file>* is the path and file name to the certificate that replaces the factory certificate for CM.

* *<alias>* is the alias name for the new CM certificate. It can be any value other than **factory**. If you do not specify an alias, *cm* is assigned as the default.

4. On the CM computer, type the following command to replace the factory certificate with an engine certificate:

```
configureCmSsl -u engCert=<cert file> engCertAlias=<alias>
```

where:

* *<cert file>* is the path and file name to the certificate you want to use to replace the factory certificate for the engine.

* *<alias>* is the alias name for the new engine certificate. This can be any value other than **factory**. If you do not specify an alias, *engine* is assigned as the default.

5. Type the following command to create an export file of the certificate store:

```
configureCmSsl -e file=<export file>
```

where *<export file>* is the path and file for the export file.

6. Copy the file you created in step 5 to the engine.

7. Stop the engine on Windows from Windows services.

8. Type the following command to import the certificates created in step 3 and step 4.

```
configureEngineSsl -i file=<export file> engCertAlias=<alias>
```

where:

* *<export file>* is the path and file for the export file.

* *<alias>* is an alias name for the engine certificate assigned in step 3. If an engCertAlias was omitted in step 3, specify *engine* as the alias.

9. Start the engine on Windows from Windows services.

10. Start CM on Windows from Windows services.

Restore Factory Certificates

Use these procedures to restore the internal certificates used to the factory certificates.

Restore the Factory Certificate on UNIX or Linux

To restore the certificate distribution to the factory settings on UNIX or Linux:

1. Stop CM.

   a. Navigate to the *install_dir*/bin directory, where *install_dir* is the installation directory, and type the following command:

   ```
   ./stopCM.sh
   ```

   b. At the passphrase prompt, type the passphrase defined for the engine and press **Enter**.

   c. Type the user name and password for the administrator.

2. From the *install_dir*/bin directory, type the following command to restore the factory certificate:

   ```
   ./configureCmSsl.sh -r
   ```

3. Type the following command to export the factory-restored certificate store:

   ```
   ./configureCmSsl.sh -e file=<export file>
   ```

   where *<export file>* is the path and file for the export file.

4. Copy the export file to the engine.

5. Stop the engine.

   a. Navigate to the *install_dir*/bin directory and type the following command:

   ```
   ./stopEngine.sh
   ```

   b. At the passphrase prompt, type the passphrase defined for the engine and press **Enter.**

6. From the *install_dir*/bin directory, type the following command to import the factory-restored certificate store:

   ```
   configureEngineSsl -i file=<export file> engCertAlias=factory
   ```

   where *<export file>* is the path and file for the certificate store.

7. Start the engine.

   a. Type the following command:

   ```
   ./startEngine.sh
   ```

   b. At the passphrase prompt, type the passphrase for the engine and press **Enter**.

8. Start CM.

   a. On CM, navigate to the *install_dir*/bin directory and type the following command:

   ```
   ./startCM.sh
   ```

   b. At the passphrase prompt, type the passphrase defined for CM and press **Enter**.

> **Note:** Restoring the configuration to use the factory certificate does not delete the certificates that were previously in use.

Restore the Factory Certificate on Windows

To restore the certificate distribution to the factory settings on UNIX or Linux:

1. Stop CM on Windows from Windows services.

2. From the *install_dir*\bin directory, type the following command to restore the factory certificate:

   ```
   configureCmSsl -r
   ```

3. Type the following command to export the factory-restored certificate store:

   ```
   configureCmSsl -e file=<export file>
   ```

   where *<export file>* is the path and file for the export file.

4. Copy the export file to the engine.

5. Stop the engine on Windows from Windows services.

6. From the *install_dir*/bin directory, type the following command to import the factory-restored certificate store:

   ```
   configureEngineSsl -i file=<export file> engCertAlias=factory
   ```

   where *<export file>* is the path and file for the certificate store.

7. Start the engine on Windows from Windows services.

8. Start CM on Windows from Windows services.

> **Note:**   Restoring the configuration to use the factory certificate does not delete the certificates that were previously in use.

Change the Password of the CM Key Store and Trust

The password for the key store and the trust store is set to "password" at installation. Use the following procedures to change the password.

Change the Password of the CM Key Store and Trust Store on UNIX or Linux

To change the password:

1. Stop CM.

    a. Navigate to the *install_dir*/bin directory, where *install_dir* is the installation directory, and type the following command:

    ```
    ./stopCM.sh
    ```

    b. At the passphrase prompt, type the passphrase defined for the engine and press **Enter**.

    c. Type the user name and password for the administrator.

2. From the *install_dir*/bin directory, type the following command:

    ```
    ./configureCmSsl.sh -x
    ```

3. When prompted, type the existing password and press **Enter**.

4. Type the new password and press **Enter**.

5. Start CM.

    a. From the *install_dir*/bin directory, type the following command:

    ```
    ./startCM.sh
    ```

    b. At the passphrase prompt, type the passphrase defined for CM and press **Enter**.

Change the Password of the CM Key Store and Trust Store on Windows

To change the password:

1. Stop CM on Windows from Windows services.

2. From the *install_dir*/bin directory, type the following command:

```
configureCmSsl -x
```

3. When prompted, type the existing password and press **Enter**.
4. Type the new password and press **Enter**.
5. Start CM on Windows from Windows services.

Change the Password of the Engine Key Store and Trust Store

The password for the key store and the trust store is set to password at installation.

Change the Password of the Engine Key Store and Trust Store on UNIX or Linux

To change the password:

1. Stop the engine.

    a. Navigate to the *install_dir*/bin directory, where *install_dir* is the installation directory, and type the following command:

    ```
    ./stopEngine.sh
    ```

    b. At the passphrase prompt, type the passphrase defined for the engine and press **Enter**.

    c. Type the user ID and password of the administrator.

2. Using a command line interface on CM, navigate to the *install_dir*/bin directory and type the following command:

```
./configureEngineSsl.sh -x
```

3. When prompted, type the existing password and press **Enter**.
4. Type the new password and press **Enter**.
5. Retype the password and press **Enter**.
6. Start the engine.

    a. Navigate to the *install_dir*/bin directory on the engine and type the following command:

    ```
    ./startEngine.sh
    ```

    b. At the passphrase prompt, type the passphrase for the engine and press **Enter**.

Change the Password of the Engine Key Store and Trust Store on Windows

To change the password:

1. Stop the engine on Windows from Windows services.

2. Using a command line interface on CM, navigate to the *install_dir*/bin directory and type the following command:

```
configureEngineSsl -x
```

3. When prompted, type the existing password and press **Enter**.

4. Type the new password and press **Enter**.

5. Retype the password and press **Enter**.

6. Start the engine on Windows from Windows services.

Configuration Utilities

Two utilities are used in the previous procedures to configure SSL: configureCmSsl and configureEngineSsl. Refer to the tables below to identify the functions that can be performed on the engine and CM. You are prompted for a password when one is required.

Use the following functions to configure CM, using the configureCmSsl utility:

Parameter	Description
-s	Show current configuration.
-u	Update configuration. Available options include:
	◆ commonCert—fully-qualified location of the common certificate to be shared by the SSP components engine, CM, and web server.
	◆ commonCertAlias—alias for the common certificate and shared by all SSP components. If the certificate file name is omitted, a certificate with this alias must exist in the key store. If no alias is provided, the value defaults to *common*.
	◆ cmCert—the fully-qualified location of CM and jetty web server certificate.
	◆ cmCertAlias—alias for the CM/jetty web server certificate. If no file name is provided, a certificate with this alias must exist in the key store. If no alias is provided, the value defaults to *cm*.
	◆ engCert—the fully-qualified location of the engine certificate.
	◆ engCertAlias—alias for the engine certificate.
	◆ webCert—the fully-qualified location of the jetty web server certificate.
	◆ webCertAlias—alias for the jetty web server certificate. If no file name is provided, a certificate with this alias must exist in the key store. If no alias is provided, the value defaults to *webserver*.

Parameter	Description
	◆ cmClientCert—the fully-qualified location of the CM client certificate.
	◆ cmClientCertAlias—alias for the CM client certificate. If no file name is provided, a certificate with this alias must exist in the key store. This certificate is used by CM to communicate with the engine. If no alias is provided, the value defaults to *cmServer*.
	◆ cmServerCert—the fully-qualified location of the CM server certificate.
	◆ cmServerCertAlias—alias for the CM server certificate. If no file name is provided, a certificate with this alias must exist in the key store. This certificate is used by CM to communicate with the jetty web server. If no alias is provided, the value defaults to *cmClient*.
	◆ cmSslProt—the SSL or TLS protocol used for the session between CM and the engine. Valid values are: SSLv2, SSLv3, TLSv1, or SSLv2Hello.
	◆ cmCiphers—ordered list of cipher suites for communication between CM and the engine. Separate ciphers with a comma, colon, or semicolon.
	◆ https—identifies if security is enabled between a web browser and the jetty web server. n = disable security, Y = security enabled. https is enabled by default.
	◆ webHost—the IP bind address for the jetty web server. The default value is localhost. If CM has multiple NIC cards, use the field to specify the IP address of the NIC card to use for the jetty web server.
	◆ webPort—the listen port for the jetty server. The default value is 8443.
	◆ webSslProt—the SSL or TLS protocol for the link between the web browser and the jetty web server. Valid values include SSLV2, SSLv3, TLSv1, or SSLv2Hello.
	◆ webCiphers—an ordered list of cipher suites to use on the connection between the web browser and jetty web server. Separate ciphers with a comma, colon, or semicolon.
	◆ clientAuth—enables client authentication for web browser clients. n= disabled. y = enabled. This option is set to n by default. If you enable clientAuth, you must add trusted certificates for the web server clients.
	◆ trustedCert—fully-qualified location of the trusted certificate for the web client.
-e	Export configuration. The export option is:
	file—to identify the fully-qualified location of the export file. It can be imported into CM or the engine.
-i	Import configuration. The import option is:
	file—to identify the fully-qualified location of the export file. It can be imported into CM or the engine.
-d	Delete a certificate. The delete option is:
	alias—the alias of the certificate to delete. This can be specified multiple times.
-x	Change key store password.
-r	Restore factory settings.
-h	List the usage and parameters of the command.

Use the following functions to configure SSL on the engine, using the configureEngineSsl utility:

Parameter	Description
-s	Show current configuration.
-i	Import configuration. Options include ◆ file—the fully-qualified location of the import file. ◆ engCertAlias—the alias for the engine certificate.
-d	Delete a certificate. Options include alias—the alias of the certificate to delete. This can be specified multiple times.
-x	Change key store password.
-h	List the usage and parameters of the command.

Manage Your SSP Configuration

After you set up your SSP configuration, use CM to edit and manage the properties of the definitions you created.

Modify Properties in an Adapter Definition

Adapters are configured with default settings. Use this procedure to modify a property. For FTP and HTTP adapters, the properties and default values are displayed. To change a property, type a new value for the property key. For SFTP and Connect:Direct adapters, the properties are not displayed. Refer to the field level help for a description of the properties. To change a property, type the property name and its key value.

To modify an adapter property:

1. If necessary, click Configuration from the menu bar.

2. Expand the Adapters tree and click the adapter to modify.

3. Click the Properties tab.

4. Click New to add a new property definition.

5. For each property, specify values for the following:

    ◆   Key

    ◆   Value

6. Click Save.

Copy and Delete Engines, Adapters, Netmaps, Nodes, and Policies

After you create an engine, adapter, netmap, node, or policy, you can copy or delete it as necessary. For nodes, you can filter the list to view only those nodes that meet your requirements.

Copy an Engine, Adapter, Netmap, or Policy

To quickly create an adapter, netmap, or policy, you can copy an existing definition and make the changes necessary to create a new item.

To copy a configured engine, adapter, netmap, or policy:

1. If necessary, click Configuration from the menu bar.

2. Expand an Engine, Adapter, Netmap, or Policy tree and click the item to copy.

3. Select Actions > Copy Selected.

   A new item is created and renamed to Copyof*ItemName* where *ItemName* is the name of the original item you created.

4. Modify the item as necessary.

5. Click Save.

Copy a Node

To quickly create an inbound or outbound node definition, you can copy an existing definition and make the changes necessary to create a new one.

To copy a node:

1. If necessary, click Configuration from the menu bar.

2. Expand the Netmaps tree and click the netmap where the node is defined.

3. Click the radio button beside the node to copy and click Copy.

   A new node is created and renamed to Copyof*ItemName* where *ItemName* is the name of the original node you created.

4. Modify the node definition as necessary.

5. Click Save.

Copy a Connect:Direct Node

To quickly create a Connect:Direct node definition, you can copy an existing definition and make the changes necessary to create a new item.

To copy a Connect:Direct node:

1. If necessary, click Configuration from the menu bar.

2. Expand the Netmap tree and click the Connect:Direct netmap where the node is defined.

3. Click the radio button beside the node to copy and click Copy.

   A new node is created and renamed to Copyof*ItemName* where *ItemName* is the name of the original node you created.

4. Modify the node definition as necessary.

5. Click OK.

6. Click Save.

Delete an Engine, Adapter, Netmap, or Policy

If you determine that an engine, adapter, netmap, or policy is no longer needed, you can delete it. Before you can delete the item, you must remove any references to it in other items. For example, if a netmap is associated with an adapter definition, it cannot be deleted.

To delete a configured engine, adapter, netmap, or policy:

1. If necessary, click Configuration from the menu bar.
2. Expand an Engine, Adapter, Netmap, or Policy tree and click the item to delete.
3. Select Actions > Delete Selected.
4. Click Delete.

Delete an Inbound Node or Outbound Node

If you determine that a node definition is no longer needed, you can delete it.

To delete a node:

1. If necessary, click Configuration from the menu bar.
2. Expand the Netmaps tree and click the netmap where the node is defined.
3. Select a node to delete and click Delete.
4. Click Save.

Delete a Connect:Direct Node

If you determine that an node definition is no longer needed, you can delete it.

To delete a Connect:Direct node:

1. If necessary, click Configuration from the menu bar.
2. Expand the Netmap tree and click the Connect:Direct netmap to where the node is defined.
3. Select the node to delete and click Delete.
4. Click Save.

Change the User Store Associated With an Engine

When you configure an engine, the user store associated with the engine is automatically configured to use the default user store called defUserStore. If you have created a user store and defined users in it, you must modify the engine definition to identify the user store.

To change the user store associated with an engine definition:

1. From the Configuration navigation panel, click Engine to expand the list of available engines.
2. Click the engine to modify.
3. Click the Advanced tab.

4. Select the user store in the User store field.

5. Click Save.

Filter a Node List

If you define a large set of inbound or outbound nodes, all of the nodes cannot be displayed on the main page. To view a subset of all available inbound nodes or outbound nodes, use the filter function. You can filter the list to display nodes that match the criteria you specify.

To filter a list:

1. If necessary, click Configuration from the menu bar.

2. Expand the Netmap tree and click the netmap to modify.

3. To filter an outbound node list:

   a. Click the Outbound Node tab.

   b. Type filter criteria to limit the list. For example, type HTTP* to view all node definitions that begin with HTTP.

   Note: Filters are case sensitive.

4. To filter an inbound node list:

   a. Click the Inbound Node tab.

   b. Type filter criteria to limit the list.

   Note: Filters are case sensitive.

SSP Field Definitions

This section provides field definitions for all dialogs in SSP. Following are definitions of the fields for each SSP screen.

Engines Field Definitions

Following are the field definitions for the engine screens.

SSP Engine Configuration - Basic

The SSP engine resides in the DMZ and runs the proxy adapters that handle client communication requests to servers in your trusted zone. Use this screen to specify basic information to configure an engine. You must obtain the engine host name and port from installation.

Refer to the field definitions in the following table. For additional information, refer to the proxy configuration chapter for the adapter type you are configuring in the *Sterling Secure Proxy Installation Guide*.

Field Name	Description
Engine Name	Engine Name identifies the name to assign the engine. It can be up to 150 characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help identify the engine. Description can be up to 255 characters.
Engine Host	Engine Host is the host or IP address where the engine is running. Valid values are 1 to 255 alphanumeric characters. Special characters allowed are underscore (_), dash (-), colon (:), and period (.).
Engine Listen Port	Engine listen port is the port on which the engine listens for configuration information from CM. Valid values are 1–65535.

SSP Engine Configuration - Advanced

Use this screen to change advanced information about an engine, including level of logging for the engine, level of logging for the local perimeter server, and whether to use a user store other than the default.

Refer to the field definitions in the following table. For additional information, refer to the *Sterling Secure Proxy Installation Guide*.

Field Name	Description
Engine Logging	Engine Logging identifies the level of logging to write to the engine log file. Logging options include: ◆ ERROR writes only error messages to the log. ERROR is the default value. ◆ WARN writes error and warning messages to the log. ◆ INFO writes error and informational messages to the log. ◆ DEBUG writes all messages to the log including debugging messages. Use this logging level when instructed to by Sterling Commerce Support.
Local Perimeter Server Logging Level	Local Perimeter Server Logging Level identifies the level of logging to write to the local perimeter server log. Logging options include: ◆ ERROR writes only error messages to the log. ERROR is the default value. ◆ WARN writes error and warning messages to the log. ◆ INFO writes error and informational messages to the log. ◆ DEBUG writes all messages to the log including debugging messages. Use this logging level when instructed to by Sterling Commerce Support.
Certicom Logging Level	Certicom Logging Level identifies the level of logging to write to the certicom log. Logging options include: ◆ ERROR writes only error messages to the log. ERROR is the default value. ◆ WARN writes error and warning messages to the log. ◆ INFO writes error and informational messages to the log. ◆ DEBUG writes all messages to the log including debugging messages. Use this logging level when instructed to by Sterling Commerce Support.
User Store	User Store identifies the user store to use with this engine. It is the location where users who access the engine are defined. defUserStore is the default user store.

Connect:Direct Protocol Field Definitions

Following are the field definitions for the Connect:Direct protocol screens.

Connect:Direct Adapter Configuration - Basic

Use this screen to specify system-level communications information for Connect:Direct connections to and from SSP. Before you can click the Advanced or Properties tabs, you must specify Adapter Name and Listen Port and select a netmap to associate with the adapter.

Refer to the field definitions in the following table. For additional information, refer to Chapter 6, *Connect:Direct Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Name	Name identifies the name to assign to the adapter you create. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the adapter you create. Description can be up to 255 characters.
Type	Type identifies the protocol being used: Connect:Direct.
Listen Port	Listen Port identifies the port number to use to listen for inbound connections. Valid values include 1-65535.
Netmap	Netmap identifies the name of the netmap to associate with the adapter you are defining. If the netmap has not been created, click ⊕ to add the netmap.
Routing Type	Select the Routing Type to identify how inbound connections are routed to the server in the trusted zone. Routing options include: ◆ Standard— select Standard to direct connections to the outbound node specified in the SNODE Netmap Entry field. ◆ Certificate-based—select this option to use the certificate presented by the inbound PNODE to determine which outbound SNODE to connect to. Certificate-based routing uses External Authentication and requires that you configure External Authentication server. ◆ PNODE-specified—select this option to route outbound connections based on information provided by the inbound PNODE. ◆ PNODE-specified, then Standard—select this option to route outbound connections based first on information provided by the inbound PNODE. If no routing information is presented by the PNODE, the connection is routed to the outbound node specified in the SNODE Netmap Entry field.
SNODE Netmap Entry	SNODE Netmap Entry identifies the name of the Connect:Direct server where the node connections are routed, after connecting to SSP. Select this value from a pull-down list.

Field Name	Description
Engine	Engine identifies the SSP server in the DMZ where traffic is first routed before being sent to the outbound secure Connect:Direct server. Select an engine from the list. You must define an engine before you can create an adapter.
Startup Mode	Startup Mode identifies how the adapter is started. **auto** starts the adapter as soon as it is pushed to the engine. **manual** requires that the adapter be manually started.

Connect:Direct Adapter Configuration - Advanced

Use this screen to specify additional communications information, and to specify the perimeter servers to use for this adapter.

Refer to the field definitions in the following table. For additional information, refer to Chapter 6, *Connect:Direct Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Logging Level	Logging Level identifies the level of logging to write to the log file for the adapter. Logging options include: ◆ ERROR writes only error messages to the log. ERROR is the default value. ◆ WARN writes error and warning messages to the log. ◆ INFO writes error and informational messages to the log. ◆ DEBUG writes all messages to the log including debugging messages. Use this logging level when instructed to turn it on by Sterling Commerce Support.
Maximum Sessions	Maximum Sessions identifies the maximum number of concurrent sessions that the adapter allows. The default is 20.
Session Timeout	Session Timeout identifies the amount of time allowed, in seconds, between transmissions of TCP packets before a session is terminated. The default is 90 seconds.
Http Ping Response	Http Ping Response identifies the response sent when an HTTP GET is received on the listen port. Provide this value to send a health check response to a third-party IP load balancer, such as Big IP. To test the response, ping the URL and port of the engine. For example if you configure an adapter on port 13640 and you want to get an HTTP 1.0 response, send a ping to http://ProxyServerURL:13640/. The value you supplied in the Http Ping Response field is returned. If you provide a value in this field, the value is displayed in a browser window. You can provide HTML syntax and text values.
Outbound Port Range	Outbound Port Range identifies the range of ports to use for the adapter. Valid values include a list of ports that are allowed with each value separated by a comma such as 1234, 2340, 16570 or a range of ports allowed, such as 16570 -17950.

Field Name	Description
External Authentication Server	External Authentication Server identifies the Sterling External Authentication Server server to use. Select the Sterling External Authentication Server server from the pull-down list. You must define an Sterling External Authentication Server server before you can select the server from the list.
Perimeter Server Mapping - Inbound Perimeter Server	Select the perimeter server to use for the inbound connection in the Perimeter Server Mapping - Inbound Perimeter Server field. To use a remote perimeter server, you must define the remote perimeter server before you can associate it with an inbound connection.
Perimeter Server Mapping - Outbound Perimeter Server	Select the perimeter server to use for the outbound connection in the Perimeter Server Mapping - Outbound Perimeter Server field. To use a remote perimeter server, you must define it before you can associate it with an outbound connection.
Perimeter Server Mapping - External Authentication Perimeter Server	Select the perimeter server to use for the EA connection in the Perimeter Server Mapping - External Authentication Perimeter Server field. To use a remote perimeter server, you must define it before you can associate it with an Sterling External Authentication Server connection.
Inbound and outbound sessions can have different levels of encryption	This field can have different levels of encryption to allow the connections from the PNODE to SSP and from SSP to the SNODE to use different encryption methods. Set this option to enable a secure connection on the inbound session but use a nonsecure connection on the outbound session, or to use different protocols on the inbound and the outbound connection.

Connect:Direct Adapter Definition - Properties

Use this screen to edit properties associated with how the Connect:Direct protocol is implemented. The keys are not displayed. To change a default key value, type the key value and assign a value to the key. Change or add a key value when instructed to do so by Sterling Support.

Refer to the field definitions in the following table. For additional information, refer to Chapter 6, *Connect:Direct Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Connect:Direct Netmap Definition

Use this screen to define the Connect:Direct netmap and all nodes allowed to connect through SSP.

Refer to the field definitions in the following table. For additional information, refer to Chapter 6, *Connect:Direct Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Netmap Name	Netmap Name identifies the name to assign to the netmap you create. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).

Field Name	Description
Description	Description assigns a description to help you identify the netmap you create. Description can be up to 255 characters.
Type	Type identifies the protocol being used: HTTP, FTP, SFTP, or Connect:Direct.
Filter	Filter allows you to view a subset of available nodes. Use the wildcard characters, * and ?, to identify the nodes to display. Filters are case-sensitive. For example, the filter n* will display node1 but will not display Node1.

Connect:Direct Netmap Node Definition - Basic

Use this screen to define the minimum Connect:Direct connection parameters for a Connect:Direct node. You must define a node name, address, and port before you can save the node definition.

Refer to the field definitions in the following table. For additional information, refer to Chapter 6, *Connect:Direct Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Node Name	Node Name identifies the name of the node server you are configuring in Connect:Direct proxy. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Routing Name	Routing Name identifies a value used to select this SNODE as the outbound node during certificate-based routing. It must match the routing name returned by EA. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_). Set this field only if you are configuring certificate-based routing in EA.
Description	Description assigns a description to help you identify the node you create. Description can be up to 255 characters.
Connect:Direct Server Address	Connect:Direct Server Address identifies the IP address or host name of the Connect:Direct server. Valid values are 1-200 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), colon (:), and underscore (_).
Connect:Direct Server Port	Connect:Direct Server Port identifies the port number of the Connect:Direct server. Valid values are 1-65535.
Policy	Policy identifies the policy you want to associate with the node you are creating. If a policy with the security attributes required has not been created, click ⊕ .
Step Injection	Step Injection identifies the function you want to associate with the node you are defining. If a step injection policy with the attributes required has not been created, click ⊕ .

Connect:Direct Netmap Node Definition - Security

Use this screen to define secure connection requirements for the node.

Refer to the field definitions in the following table. For additional information, refer to Chapter 6, *Connect:Direct Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Use secure+	Enable Use secure+ to turn on the use of SSL/TLS to provide secure communications with transport protocols and to ensure that data is secured as it is transmitted across a single socket.
Verify Common Name	Enable Verify Common Name if your security environment requires that the common name in the certificate presented be verified. If you enable Verify Common Name, you must provide Certificate Common Name.
Certificate Common Name	Certificate Common Name identifies the common name value to validate. If the common name in the certificate does not match the value defined in this field, the session fails.
Security Setting	Security Setting identifies the security protocol allowed for connections to this node. Options include: ◆ SSL - select this option to require SSL for the connection. ◆ TLS - select this option to require TLS for the connection. ◆ The PNODE host controls SSL Protocol - select this option to use the protocol specified at the PNODE.
Trust Store	Trust Store identifies the location where the system and CA certificates are stored. System and CA certificates are used during a secure connection to verify that a certificate received from a server is signed by a trusted source.
CA Certificates/Trusted Root	CA Certificates/Trusted Root identifies the trusted certificate to use to authenticate the certificate presented by the client. You select a CA certificate or trusted root from the list of certificates stored in the trust store you selected in the Trust Store field. When a client presents a certificate to establish a secure connection, the trusted root certificate, located at the server, must match or be the entity who signed the certificate presented by the client during the SSL handshake.
Key Store	Key Store identifies the location where the keys and system certificates you want to use are stored.
Key/System Certificate	Key/System Certificate identifies the certificate presented by SSP to the node to authenticate itself during the SSL handshake, or the certificate presented by the Connect:Direct server to authenticate itself to the SSP server. Select the Key/System Certificate to use for the node from the list. The list contains the key or system certificates stored in the key store you selected in the Key Store field.
Available Cipher Suites	Available Cipher Suites is the list of ciphers that can be enabled to encrypt data transmitted during a secure SSL or TLS connection between SSP and a Connect:Direct node. Enable at least one cipher. To enable a cipher, highlight it and click Add. To enable multiple ciphers, highlight the ciphers to enable and click Add.

Field Name	Description
Selected Cipher Suites	Selected Cipher Suites identifies the ciphers you have enabled to encrypt data during a secure SSL or TLS connection. Ciphers are negotiated based on their location in the Selected Ciphers list. To reorder a cipher in the list, highlight it and click Up or Down.

Connect:Direct Netmap Node Definition - Advanced

Use this screen to change the logging level for a Connect:Direct node definition and the TCP timeout value to wait for a response as well as to identify a destination service name to use in an EA transaction and to configure nodes to use for failover support.

Refer to the field definitions in the following table. For additional information, refer to Chapter 6, *Connect:Direct Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Logging level	Logging level identifies the level of logging at which to write to the node log file. Logging options include: ◆ NONE turns logging off. NONE is the default value. ◆ ERROR writes only error messages to the log. ◆ WARN writes error and warning messages to the log. ◆ INFO writes error and informational messages to the log. ◆ DEBUG writes all messages to the log including debugging messages. Use this logging level when instructed to turn it on by Sterling Commerce Support.
TCP Timeout	TCP Timeout identifies the number of seconds to wait for a TCP/IP request or response before ending the session. The default is 90.
Destination Service Name	Destination Service Name identifies the name of the service that is passed to Sterling External Authentication Server (EA) for use in authenticating services. If no value is provided, the SNODE name is used as the service name.
Alternate Destinations - Node	Alternate Destinations Node 1 identifies the node name or IP address and port to use to connect to an alternate Connect:Direct outbound server, if a connection to the primary node cannot be made. Up to three alternate destination nodes can be defined for each outbound node. To use different security and Sterling External Authentication Server definitions for the alternate destination node, first configure an outbound node definition for the alternate node in the netmap. Then, open the primary outbound node definition and select the alternate node name from the drop-down list in the Alternate Destinations - Node 1 field. To use the security and Sterling External Authentication Server definition defined in the primary outbound node for an alternate destination node, you do not have to define the alternate node in the netmap. Select IP Address/Port from the drop down list and then provide a value in the IP Address and Port fields. Define up to three alternate node names.

Field Name	Description
Alternate Destinations - IP Address	Alternate Destinations Address identifies the IP address to use to connect to an alternate destination node, if a connection to the primary node cannot be made. Up to three alternate destination nodes can be selected. Valid values are 1-200 alphanumeric characters and special characters: underscore (_), dash (-), colon (:), and period (.). If you provide an IP address and port as an alternate destination and a connection to the alternate node is attempted, the security and EA definition from the primary node is used for the connection.
Alternate Destinations - Port	Alternate Destinations Port identifies the port to use to connect to an alternate destination node if a connection to the primary node cannot be made. Up to three alternate destination nodes can be selected. Valid values are 1-65535. If you provide an IP address and port as an alternate destination and a connection to the alternate node is attempted, the security and EA definition from the primary outbound node is used for the connection.

Connect:Direct Policy Definition

Use the screen and the tabs to define the Connect:Direct policy and define how you impose controls to authenticate a trading partner.

Connect:Direct Policy Configuration - Basic

Use the fields on this tab to define the policy name, the protocol being used, and the action to take if a protocol violation occurs.

Refer to the field definitions in the following table. For additional information, refer to Chapter 6, *Connect:Direct Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Policy Name	Policy Name identifies the name to assign to the policy you create. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the policy you create. Description can be up to 255 characters.
Type	Type identifies the protocol being used: Connect:Direct.
Protocol Error Action	Protocol Error Action identifies the action to perform if SSP detects protocol violations during a communications session. Valid values are: ◆ NONE - select this option to disable checking of protocol errors. ◆ IGNORE - select this option to ignore protocol errors. ◆ WARN - select this option if you want SSP to write an error message to the log but continue the session when protocol errors are detected. ◆ ABORT - select this option to terminate a communications session when protocol errors are detected.

Connect:Direct Policy Configuration - Advanced

Use this tab to specify the type of user authentication for inbound access requests. For Certificate Authentication and User Authentication through External Authentication, you must install and configure EA.

Refer to the field definitions in the following table. For additional information, refer to Chapter 6, *Connect:Direct Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Certificate Authentication - External Authentication Certificate Validation	Turn on External Authentication Certificate Validation to validate information presented in certificates received from trading partners using EA.
Certificate Authentication - External Authentication Profile	External Authentication Profile identifies the name of the Certificate Validation Definition you defined in the EA. You must enable certificate validation before you can provide a profile. Valid values are 1-255 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
User Authentication - Through External Authentication	Turn on User Authentication through External Authentication (EA) to send an incoming user ID and password to Sterling External Authentication Server for validation. Sterling External Authentication Server validates the user ID and password using one of the following methods: ◆ An LDAP bind to authenticate a user and password ◆ An LDAP search, based on the user ID, destination service name, or certificate subject and issues attributes ◆ Fully-specified DN (distinguished name) or a DN returned from a certificate validation search
User Authentication - External Authentication Profile	If you enabled user authentication through EA, identify the certificate authentication profile you defined in EA in the External Authentication Profile field. Valid values are 1-255 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
User Authentication - Through Local User Store	User Authentication Through Local User Store validates the user ID and password of the inbound node using information defined in the user store. You must add the user to the user store in order to successfully use this method.

Field Name	Description
User Mapping - Internal User ID	User Mapping - Internal User ID is enabled in the policy and determines what user ID and password is used to connect to the Connect:Direct server in the secure environment. For the user ID and password to successfully access the Connect:Direct server, a user definition must be defined at the server. User mapping options include: ◆ Pass-through for PNODE—Uses the user ID and password supplied by the PNODE to connect to the Connect:Direct server in the secure zone. To successfully connect to the Connect:Direct server, the user ID and password must be defined at the server. ◆ Replace SNODEID with UserId mapped in External Authentication—Maps the user ID provided by the inbound SNODE to a value defined in External Authentication and uses the EA-supplied value to connect to the SNODE. If you select this option, both the SNODE ID and the submitter ID are replaced with a new value. Do not select this option if you have enabled secure point of entry checking in Connect:Direct. ◆ Replace SubmitterID with UserId mapped in External Authentication—Maps the submitter ID provided by the inbound PNODE to a value defined in External Authentication and uses the EA-supplied value to connect to the SNODE.

Connect:Direct Policy Definition - Step Permissions

Use this tab to block Connect:Direct tasks from being performed on a node.

Refer to the field definitions in the following table. For additional information, refer to Chapter 6, *Connect:Direct Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Runjob step allowed	Allows runjob steps to be performed on the PNODE.
Runtask step allowed	Allows runtask steps to be performed on the PNODE.
Copy step allowed	Allows copy steps to be performed on the PNODE.
Submit step allowed	Allows submit steps to be performed on the PNODE

Connect:Direct Step Injection Configuration

Use this screen to create a step injection function. Use the Step Injection Advanced tab to define the functions implemented with the step injection you define.

Step injection allows you to insert Connect:Direct Process statements into the communications session with the SNODE independent of the PNODE Process statements. These injected statements can provide real-time notification of file delivery, invoke applications, submit operating system jobs, and submit other Connect:Direct Processes, all without the need to provide an exit program on the SNODE or without changing the PNODE Process. The PNODE receives no indication that these steps have executed. However, execution results of these steps are logged in the statistics file of the SNODE.

Connect:Direct Step Injection Configuration - Basic

Use this tab to create a step injection function.

Refer to the field definitions in the following table. For additional information, refer to the Chapter 6, *Connect:Direct Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Step Injection Name	Step Injection Name identifies the name to assign to the step injection policy you create. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the step injection function you create. Description can be up to 255 characters.

Connect:Direct Step Injection Advanced

Use this tab to define the functions implemented the step injection.

Refer to the field definitions in the following table. For additional information, refer to Chapter 6, *Connect:Direct Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Copy on success	Enable Copy on success to copy information to the SNODE at the end of a successful step. Information that can be copied includes certificate information, metadata returned by Sterling External Authentication Server associated with the entity represented by the certificate, and Process information such as a file name or step name.
Copy identifying information	If you enable Copy on success, identify what information to copy to the SNODE. Options include: ◆ Copy All Information to copy all information about the session to the SNODE ◆ Copy Certificate Information to copy only certificate information to the SNODE ◆ Copy Session Information to copy only session information to the SNODE
Session information output file	Session information output file identifies the name of the file where information about the successful session is written.
Tcp timeout for copy	Tcp timeout for copy identifies the number of seconds to wait for a TCP/IP request or response before ending the session.
Execute on success	Enable this option to execute an operating system command, program, or Submit Connect:Direct Process on the SNODE at the end of a successful step.
Step selection	If you enable Execute on success, identify the type of step to execute: Runtask, Runjob, or Submit.
Step parameter	Step parameter provides a place to type the step parameters. Refer to the Connect:Direct Process Information on the Sterling Commerce Customer Center for information on step parameters.

Field Name	Description
Tcp timeout for step	Tcp timeout for step identifies the number of seconds to wait for a TCP/IP request or response before ending the session.
Copy on failure	Enable this option to copy session-specific data to the SNODE at the end of a failed step.
Copy identifying information	If you enable Copy on failure, identify the information to copy to the SNODE. Options include: ◆ Copy All Information—to copy all information about the session to the SNODE ◆ Copy Certificate Information—to copy only certificate information to the SNODE ◆ Copy Session Information—to copy only session information to the SNODE
Session information output file	Session information output file identifies the name of the file where information about the failed session is to be written.
Tcp timeout for copy	Tcp timeout for copy identifies the number of seconds to wait for a TCP/IP request or response before ending the session.
Execute on failure	Enable Execute on failure to execute an operating system command, program, or Submit Connect:Direct Process on the SNODE at the end of a failed step.
Step selection	If you enable Execute on failure, identify the type of step to execute: Runtask, Runjob, or Submit.
Step parameter	Step parameter provides a place to type the step parameters. Refer to the Connect:Direct process Guide for information on step parameters.
Tcp timeout for step	Tcp timeout for step identifies the number of seconds to wait for a TCP/IP request or response before ending the session.

FTP Protocol Field Definitions

Following are the field definitions for the FTP protocol screens.

FTP Adapter Definition

Use this screen and its tabs to specify system-level communications information for FTP connections.

FTP Adapter Definition - Basic

Use this tab to specify basic communications information for FTP connections to and from SSP. Before you can click the Advanced or Properties tabs, you must specify Adapter Name and Listen Port. Refer to the field definitions in the following table.

For additional information, refer to Chapter 7, *FTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Adapter Name	Adapter Name identifies the name to assign to the adapter you create. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the adapter you create. Description can be up to 255 characters.
Type	Type identifies the protocol being used as FTP.
Listen Port	Listen Port identifies the port number to use to listen for inbound connections. Default = 13640. Valid values are between 1-65535.
Netmap	Netmap identifies the name of the netmap to associate with the adapter you are defining. If the netmap has not been created, click ⊕ to add the netmap.
Standard Routing Node	Standard Routing Node identifies the name of the Connect:Direct secure server where the inbound node connections are routed after connecting to SSP.
Engine	Engine identifies the SSP server in the DMZ where traffic is first routed before being sent to the outbound secure Connect:Direct server. Select an engine from the list. You must define an engine before you can create an adapter.
Startup Mode	Startup Mode identifies how the adapter is started. **auto** starts the adapter as soon as it is pushed to the engine. **manual** requires that the adapter be manually started.

FTP Adapter Definition - Advanced

Use this screen to specify additional communications information and to specify the perimeter servers to use for this adapter. Refer to the field definitions in the following table. For additional information, refer to Chapter 7, *FTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

For more information on perimeter servers, refer to Chapter 4, *Configure Perimeter Servers to Manage SSP Communications*.

Field Name	Description
Logging Level	Logging Level identifies the level of logging at which to write to the adapter log file. Logging options include: ◆ ERROR writes only error message to the log. ◆ WARN writes error messages and warning messages to the log. ◆ INFO writes error messages and informational messages to the log. ◆ DEBUG writes all messages to the log including debugging messages. Use this logging level when instructed to turn it on by Sterling Commerce Support.
Maximum Sessions	Maximum Sessions identifies the maximum number of sessions that the adapter allows. The default is 20.
Session Timeout	Session Timeout identifies the amount of time allowed, in minutes, between TCP packets before a session is terminated. The default is 3 minutes.
Outbound Port Range	Outbound Port Range identifies the range of ports to use for the adapter. Valid values include a list of ports with each value separated by a comma such as 1234, 2340, 16570, or a range of ports, such as 16570 -17950.
Active Data Outbound Port Range	Active Data Outbound Port Range identifies the port range to use to listen for connections on the client data channel, when the client submits a PORT command.
Passive Data Listening Port Range	Passive Data Listening Port Range identifies the port range used to listen for connections on the data channel when the client issues a PASV command. If no value is specified, any available port is used.
Passive NAT Address	Passive NAT Address identifies the IP address sent to the FTP client in response to a PASV command. Define this value if the client cannot directly connect to the proxy, such as when using a remote perimeter server or static network address translation (NAT). The default value is the remote perimeter server address. If you are using a remote external perimeter server with the FTP reverse proxy adapter, identify the name or IP address of the computer running the external perimeter server.
External Authentication Server	External Authentication Server identifies the server where EA is installed. Select the EA server from list. You must define the EA server before you select it from the list.
Use IP from PASV Response	Enable this option to use the IP address from the PASV response for outbound data connections.
Perimeter Server Mapping - Inbound Perimeter Server	Select the perimeter server to use for the inbound connection in the Perimeter Server Mapping - Inbound Perimeter Server field. You must define the remote perimeter server before you can associate it with an inbound connection.

Field Name	Description
Perimeter Server Mapping - Outbound Perimeter Server	Select the perimeter server to use for the outbound connection in the Perimeter Server Mapping - Outbound Perimeter Server field. You must define the remote perimeter server before you can associate it with an outbound connection.
Perimeter Server Mapping - External Authentication Perimeter Server	Select the perimeter server to use for the EA connection in the Perimeter Server Mapping - External Authentication Perimeter Server field. You must define the remote perimeter server before you can associate it with an EA connection.

FTP Adapter Definition - Custom

Use this screen to define a custom banner to display for a server greeting banner and a login banner. Refer to the field definitions in the following table:

Field Name	Description
Server Greeting Banner Text	Identify the text to display when a user successfully connects to the server. For example, type Welcome to my ftp site.
	If no banner text is provided in this field, the default banner, 220 FTP Server ready, is displayed.
	Valid values are 0-4000 alphanumeric characters.
Login Banner Text	Identify the text to display as a login message for an FTP proxy adapter. If no banner text is provided, the default banner, 230 User %user logged in, is displayed.
	Valid values are 0-4000 alphanumeric characters.

FTP Adapter Definition - Properties

Use this screen to edit properties associated with how the FTP protocol is implemented. The keys are not displayed. To change a default key value, type the key value as defined in the following table and assign a value to the key.

Refer to the field definitions in the following table. For additional information, refer to Chapter 7, *FTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description	
Key	Key identifies properties that you can change for an adapter. Available keys include:	
	◆ max.ps.server.threads	Maximum number of threads in the pool used during a connection with a server. Default value is 10.
	◆ ftp.ssl.pbsz.required	Identifies whether the SSL command, PBSZ, is required. Valid values include Y|Yes|y|No|N|n. The default is Y.
		Set this property to N for certain clients, like Tumbleweed, that do not send a PBSZ command during an SSL session.
	◆ ftp.commands.prohibited	Identifies the FTP commands that cannot be used when an FTP client initiates a connection.
	◆ ftp.ssl.prot.required	Identifies whether the SSL command, PROT, is required. Valid values include Y|Yes|y|No|N|n. The default is Y.
		Set this property to N for certain clients, like Tumbleweed, that do not send a PROT command during an SSL session.
	◆ max.ps.client.threads	Maximum number of threads in the pool used during a connection with a client. Default value is 10.
	◆ ftp.commands.allowed	Identifies the FTP commands that can be used when an FTP client initiates a connection.
	◆ ftp.max.command.length	Maximum length allowed for a client command. The default is 1024. The command length is unlimited if this parameter is set to 0. If this length is exceeded, an error is logged and the connection is closed.
	◆ ftp.max.response.length	Maximum length allowed for a server ftp response. The default is 4096. The server ftp length is unlimited if the parameter is set to 0. If this length is exceeded, an error is logged and the connection is closed.
		Note: Set this parameter to 0 when communicating with a z/OS FTP server.
Value	Value to assign to a property key. See Key for a list of properties you can modify.	

FTP Netmap Definition

Use this screen to define the FTP netmap. Click Inbound Nodes to add connection information for external trading partners. Click Outbound Nodes to add connection information for internal FTP server. Refer to the field definitions in the following table. For additional information, refer to Chapter 7, *FTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Note:	You must define a netmap, at least one inbound node, and at least one outbound node before you can save the netmap. If you exit the application before all three elements are defined, you lose the netmap definition.

Field Name	Description
Netmap Name	Netmap Name identifies the name to assign to the netmap you create. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the netmap you create. Description can be up to 255 characters.
Type	Type identifies the protocol being used as FTP.
Filter	Filter allows you to view a subset of available inbound or outbound nodes. Use the wildcard characters, * and ?, to identify the nodes to display. Filters are case sensitive. For example, the filter i* will display inboundnode1 but will not display InboundNode1.

FTP Netmap Inbound Node Definition - Basic

Use this screen to define the minimum FTP connection requirements for an external trading partner.

Refer to the field definitions in the following table. For additional information, refer to Chapter 7, *FTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Inbound Node Name	Inbound Node Name is the name associated with the inbound node connection. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the inbound node you create. Description can be up to 255 characters.

Field Name	Description
Peer Address Pattern	Peer Address Pattern identifies the pattern to allow for the inbound connections to SSP. Valid values are alphanumeric characters and the following special characters: dash (-), underscore (_), colon (:), period (.), dollar sign ($), forward slash (/), exclamation mark (!), tilde (~), asterisk (*), open parenthesis ("), close parenthesis (") semicolon (;), question mark (?), at (@), and comma (,). You can define one of the following types of patterns: ◆ Wildcard validates incoming DNS names. If a wildcard pattern is provided, SSP performs a reverse lookup on the incoming IP address and the DNS name is compared to the wildcard patterns. Wildcard characters allowed are ? and *. For example, *.a.com allows a connection from b.a.com but not from b.b.com. ◆ IP/Subnet validates incoming IP addresses. Use the format IP-address/num-bits where IP-address identifies an IP address template and num-bits identifies the number of leading (highest-order) bits in the template that are significant. An IP match is performed by comparing the leading (highest-order) num-bits of the incoming IP address against num-bits of the template. For example, 10.20.0.0/16 searches for a match to the first 16 bits. All IP addresses beginning with 10.20.* are allowed. 10.0.0.0/8 searches for a match to the first 8 bits. All addresses beginning with 10.* are allowed. 0.0.0.0/0 searches for a match the first zero bits. All IP addresses are allowed.
Policy	Policy is a list of policies you have created. Select the policy you want to associate with the inbound node you are creating. If a policy with the security attributes required has not been created, click .

FTP Netmap Inbound Node Definition - Security

Use this screen to define secure connection requirements for an external trading partner.

Refer to the field definitions in the following table. For additional information, refer to Chapter 7, *FTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Secure Connection	Enable Secure Connection to turn on the use of SSL/TLS to provide secure communications with transport protocols and to ensure that data is secured as it is transmitted across a single socket.

Field Name	Description
Security Setting	Security Setting identifies the security protocol allowed for connections to this node. Options include: ◆ SSL v3 or TLS—sends a TLS Hello and accept SSLv3 or TLS ◆ SSL v2 or v3 with v3 Hello—sends an SSLv3 Hello and accept SSLv3 or SSLv2 ◆ SSL (any version) or TLS—sends an SSL v2 Hello and accept SSLv3, SSLv2, or TLS ◆ SSL v2 or v3—sends SSLv2 Hello and accept SSLv3 or SSLv2 ◆ TLS—sends TLS Hello and accept TLS only ◆ SSL v3—sends SSLv3 Hello and accept SSLv3 only
Enable Client Authentication	Enable Client Authentication on the inbound node connection to require that the SSP server authenticate the certificate presented by the inbound node connection.
Trust Store	Trust Store identifies the location where the system and CA certificates are stored. System and CA certificates are used during secure connection to verify that a certificate received from a server is signed by a trusted source.
CA Certificates/Trusted Root	CA Certificates/Trusted Root identifies the trusted certificate to use to authenticate the certificate presented by the client. You select a CA certificate or trusted root from the list of certificates stored in the trust store you selected in the Trust Store field. When a client presents a certificate to establish a secure connection, the trusted root certificate located at the server must match or be the entity who signed the certificate presented by the client during the SSL handshake.
Key Store	Key Store identifies the location where the key certificates you want to use are stored.
Key/System Certificate	Key/System Certificate identifies the certificate presented by SSP to the inbound node to authenticate itself during the SSL handshake or the certificate presented by the Connect:Direct server to authenticate itself to the SSP server. Select the Key/System Certificate to enable for the node from the list. The list identifies the key or system certificates stored in the key store you selected in the Key Store field.
Available Cipher Suites	Available Cipher Suites provides a list of ciphers that can be enabled to encrypt data transmitted during a secure SSL or TLS connection. Enable at least one cipher. To enable a cipher, highlight it and click Add. To enable multiple ciphers, highlight them and click Add.
Selected Cipher Suites	Selected Cipher Suites identifies the ciphers you have enabled to encrypt data during a secure SSL or TLS connection. A cipher suite is negotiated during a secure channel connection between a client and a server. Ciphers are negotiated based on their location in the Selected Ciphers list. To reorder a cipher in the list, highlight it and click Up or Down.
Clear Control Channel	Enable Clear Control Channel to allow an inbound or outbound node to use an unencrypted control channel for file transmission after the SSL or TLS handshake is complete.

FTP Netmap Inbound Node Definition- Advanced

Use this screen to specify the level at which to log information on this inbound connection.

Refer to the field definitions in the following table. For additional information, refer to Chapter 7, *FTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Logging Level	Logging Level identifies the level of logging to write to the log file for the inbound node. Logging options include the following: ◆ NONE turns logging off. NONE is the default value. ◆ ERROR writes only error messages to the log. ◆ WARN writes error and warning messages to the log. ◆ INFO writes error and informational messages to the log. ◆ DEBUG writes all messages to the log including debugging messages. Use this logging level when instructed to turn it on by Sterling Commerce Support.

FTP Netmap Outbound Node Definition - Basic

Use this screen to define the minimum connection requirements for your internal FTP server.

Refer to the field definitions in the following table. For additional information, refer to Chapter 7, *FTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Outbound Node Name	Outbound Node Name identifies the name of the outbound server in the secure zone which is the destination of the communications session. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the outbound node you create. Description can be up to 255 characters.
Primary Destination Address	Primary Destination Address identifies the IP address or host name to use to connect to the Connect:Direct outbound server. Valid values are 1-200 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), colon (:), and underscore (_).
Primary Destination Port	Primary Destination Port identifies the port to use to connect to the secure Connect:Direct server. Valid values are 1-65535.

FTP Outbound Node Definition - Security

Use this screen to define the secure connection requirements for your internal FTP server.

Refer to the field definitions in the following table. For additional information, refer to Chapter 7, *FTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

For more information on setting up certificates, refer to Chapter 3, *Manage Certificates for SSL/TLS Transactions with Trading Partners*.

Field Name	Description
Secure Connection	Enable Secure Connection to turn on the use of SSL/TLS to provide secure communications with transport protocols and to ensure that data is secured as it is transmitted across a single socket.
Security Setting	Security Setting identifies the security protocol allowed for connections to this node. Options include: ◆ SSL v3 or TLS - select this option to send an TLS Hello and accept SSLv3 or TLS ◆ SSL v2 or v3 with v3 Hello - select this option to send an SSLv3 Hello and accept SSLv3 or SSLv2 ◆ SSL (any version) or TLS - select this option to send an SSL v2 Hello and accept SSLv3, SSLv2, or TLS ◆ SSL v2 or v3 - select this option to send SSLv2 Hello and accept SSLv3 or SSLv2 ◆ TLS - select this option to send TLS Hello and accept TLS only ◆ SSL v3 - select this option to send SSLv3 Hello and accept SSLv3 only
Trust Store	Trust Store identifies the location where the system and CA certificates are stored. System and CA certificates are used during secure connection to verify that a certificate received from a server is signed by a trusted source.
CA Certificate /Trusted Root	CA Certificate/Trusted Root identifies the trusted certificate to use to authenticate the certificate presented by the client. You select a CA certificate or trusted root from the list of certificates stored in the trust store you selected in the Trust Store field. When a client presents a certificate to establish a secure connection, the trusted root certificate, located at the server, must match or be the entity that signed the certificate presented by the client during the SSL handshake.
Key Store	Key Store identifies the location where the key certificates you want to use are stored.
Key/System Certificate	Key/System Certificate identifies the certificate presented by SSP to the inbound node to authenticate itself during the SSL handshake or the certificate presented by the Connect:Direct server to authenticate itself to the SSP server. Select the Key/System Certificate to enable for the node from the list. The list identifies the key or system certificates stored in the key store you selected in the Key Store field.
Available Cipher Suites	Available Cipher Suites provides a list of ciphers that can be enabled to encrypt data transmitted during a secure SSL or TLS connection. Enable at least one cipher. To enable a cipher, highlight the cipher in the Available Cipher Suites dialog and click Add. To enable multiple ciphers, highlight the ciphers to enable and click Add.
Selected Cipher Suites	Selected Cipher Suites identifies the ciphers you have enabled to encrypt data during a secure SSL or TLS connection. A cipher suite is negotiated during a secure channel connection between a client and a server. Ciphers are negotiated based on their location in the Selected Ciphers list. To reorder a cipher in the list, highlight the cipher to reorder and click the Up or Down button.
Clear Control Channel	Enable Clear Control Channel to use an unencrypted control channel after login is complete.

FTP Netmap Outbound Node Definition - Advanced

Use this screen to define advanced parameters for your internal FTP server.

Refer to the field definitions in the following table. For additional information, refer to Chapter 7, *FTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Logging Level	Logging Level identifies the level of logging to write to the log file for the outbound node. Logging options include: ◆ NONE turns logging off. ◆ ERROR writes only error message to the log. ◆ WARN writes error messages and warning messages to the log. ◆ INFO writes error messages and informational messages to the log. ◆ DEBUG writes all messages to the log including debugging messages. Use this logging level when instructed to turn it on by Sterling Commerce Support.
Destination Service Name	Destination Service Name identifies a destination server that can be accessed by the outbound node, when using EA to map a user ID and password. Valid values are 1-255 alphanumeric characters and certain special characters. The following special characters are not allowed: ! @ # % ^ * ( ) + ? , < > { } [ ] | ; " '
User ID	User ID identifies the user ID to use to connect to the secure outbound server, if the policy is defined to required that the user ID and password from the netmap be used. Valid values are 1-255 alphanumeric characters and certain special characters. The following special characters are not allowed: ! @ # % ^ * ( ) + ? , < > { } [ ] | ; " '
Password	Password identifies the password to use to connect to the secure outbound server, if the policy is defined to require the user ID and password from the netmap. Valid values are 1-255 alphanumeric characters and certain special characters. The following special characters are not allowed: , " '
Alternate Destinations - Node	Alternate Destinations Node 1 identifies the node name or IP address and port to use to connect to an alternate Connect:Direct outbound server if a connection to the primary node cannot be made. Up to three alternate destination nodes can be defined for each outbound node. To use different security and EA definitions for the alternate destination node, first configure an outbound node definition for the alternate node in the netmap. Then, open the primary outbound node definition and select the alternate node name from the drop-down list in the Alternate Destinations - Node 1 field. To use the security and EA definition defined in the primary outbound node for an alternate destination node, you do not have to define the alternate node in the netmap. Select IP Address/Port from the drop down list and then provide a value in the IP Address and Port fields. Define up to three alternate node names.

Field Name	Description
Alternate Destinations - IP Address	Alternate Destinations Address identifies the IP address to use to connect to an alternate destination node, if a connection to the primary node cannot be made. Up to three alternate destination nodes can be selected. Valid values are 1-200 alphanumeric characters and special characters: underscore (_), dash (-), colon (:), and period (.).
	If you provide an IP address and port as an alternate destination and a connection to the alternate node is attempted, the security and EA definition from the primary node is used for the connection.
Alternate Destinations - Port	Alternate Destinations Port identifies the port to use to connect to an alternate destination node, if a connection to the primary node cannot be made. Up to three alternate destination nodes can be selected. Valid values are 1-65535.
	If you provide an IP address and port as an alternate destination and a connection to the alternate node is attempted, the security and EA definition from the primary outbound node is used for the connection.

FTP Policy Configuration - Basic

Use this screen to define how you impose controls to authenticate a trading partner trying to access your FTP server.

Refer to the field definitions in the following table. For additional information, refer to Chapter 7, *FTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Policy Name	Policy Name identifies the name to assign to the policy you create. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the policy you create. Description can be up to 255 characters.
Type	Type identifies the protocol being used as FTP.

FTP Policy Configuration- Advanced

Use this tab to specify the type of user authentication to use for inbound access requests. For Certificate Authentication and User Authentication through External Authentication, you must have installed and configured EA. You can also use this screen to map an incoming user ID and password to a different user ID and password to present to the internal server.

Refer to the field definitions in the following table. For additional information, refer to Chapter 7, *FTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Certificate Authentication - External Authentication Certificate Validation	Turn on External Authentication Certificate Validation to validate information presented in certificates received from trading partners using EA.
Certificate Authentication - External Authentication Profile	External Authentication Profile identifies the name of the Certificate Validation Definition you defined in EA. You must enable certificate validation before you can provide a profile.
User Authentication - Through External Authentication	Turn on User Authentication through External Authentication to send an incoming user ID and password to EA for validation. EA validates the user ID and password using one of the following methods: ◆ An LDAP bind to authenticate a user and password ◆ An LDAP search, based on the user ID, destination service name, or certificate subject and issues attributes ◆ Fully specified DN or a DN returned from a certificate validation search
User Authentication - External Authentication Profile	If you enabled user authentication through EA, identify the certificate authentication profile you defined in EA in the External Authentication Profile field.
User Authentication - Through Local User Store	User Authentication Through Local User Store validates the user ID and password of the inbound node using information defined in the user store. You must add the user to the user store in order to successfully use this method.
User Mapping - Internal User ID	User Mapping - Internal User ID is enabled in the policy and determines what user ID and password is used to attach to the Connect:Direct server in the secure environment. In order for the user ID and password presented to the GIS server to successfully access the server, a user definition must be defined at the GIS server. User mapping options include: ◆ Pass-through uses the user ID and password supplied by the inbound node to connect to the Connect:Direct server in the secure zone. To successfully connect to the Connect:Direct server, the user ID and password must be defined in the user store at the server. ◆ From External Authentication uses a user ID and password from External Authentication to connect to the Connect:Direct server. To successfully connect using this option, the user ID and password must be defined in the LDAP database. ◆ From Netmap uses the user ID and password defined in the netmap to connect to the Connect:Direct server. To successfully connect using this option, define the user ID and password to use to connect to the Connect:Direct server in the outbound node definition.

HTTP Protocol Field Definitions

Following are the field definitions for the HTTP protocol screens.

HTTP Adapter Configuration

Use this screen and its tabs to specify system-level communications information for HTTP connections to and from SSP.

HTTP Adapter Configuration - Basic

Use this tab to specify basic communications information for HTTP connections. Before you can click the Advanced or Properties tabs, you must specify Adapter Name and Listen Port. Refer to the field definitions in the following table. For additional information, refer to Chapter 8, *HTTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Adapter Name	Adapter Name identifies the name to assign to the adapter you create. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the adapter you create. Description can be up to 255 characters.
Type	Type identifies the protocol being used as HTTP.
Listen Port	Listen Port identifies the port number to use to listen for inbound connections. Default = 13640. Valid values include 1-65535.
Netmap	Netmap identifies the name of the netmap to associate with the adapter you are defining. If the netmap has not been created, click ⊕ to add the netmap.
Routing Type	Select the Routing Type to identify how inbound connections are routed to the HTTP server in the trusted zone. For HTTP, select standardRouting.
Standard Routing Node	Standard Routing Node identifies the name of the Connect:Direct secure server where the inbound node connections are routed, after connecting to SSP. Select this value from a pull-down list.
Support HTML Rewrite	Support HTML Rewrite identifies if HTML rewrite has been enabled on the netmap. This is a read-only field.
Engine	Engine identifies the SSP server in the DMZ where traffic is first routed before being sent to the outbound secure Connect:Direct server. Select an engine from the list. You must define an engine before you can create an adapter.
Startup Mode	Startup Mode identifies how the adapter is started. **auto** starts the adapter as soon as it is pushed to the engine. **manual** requires that the adapter be manually started.

HTTP Adapter Definition - Advanced

Use this screen to specify additional communications information, and to specify the perimeter servers to use for this adapter.

Refer to the field definitions in the following table. For additional information, refer to Chapter 8, *HTTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Logging Level	Logging Level identifies the level of logging to write to the log file for the adapter. Logging options include: ◆ ERROR writes only error message to the log. ◆ WARN writes error messages and warning messages to the log. ◆ INFO writes error messages and informational messages to the log. ◆ DEBUG writes all messages to the log including debugging messages. Use this logging level when instructed to turn it on by Sterling Commerce Support.
Maximum Sessions	Maximum Sessions identifies the maximum number of sessions that the adapter allows. The default is 20.
Session Timeout	Session Timeout identifies the amount of time allowed, in minutes, between TCP packets before a session is terminated. The default is 3 minutes.
HTTP Ping Response	HTTP Ping Response identifies the response to send when an HTTP GET is received on the listen port. Provide this value To perform a health check response to a third-party IP load balancer, such as Big IP. If you provide a value in this field, the value is displayed in a browser window. You can provide HTML syntax and text values in this field. To test the response, ping the URL that you define in the HTTP Ping URI field and port of the engine. For example if you configure an adapter on port 13640 and you want to get an HTTP 1.0 response, send a ping to http://ProxyServerURL:13640/<*HTTP Ping URI*>. The value you supplied in the HTTP Ping Response field is returned. If you provide a value in this field, the value is displayed in a browser window. You can provide HTML syntax and text values.
HTTP Ping URI	HTTP Ping URI identifies the URI to monitor for incoming requests from an inbound node. If SSP receives a request for this URI, it returns the ping response, provided in the HTTP Ping Response field.
Outbound Port Range	Outbound Port Range identifies the range of ports to use for the adapter. Valid values include a list of ports that are allowed with each value separated by a comma such as 1234, 2340, 16570 or a range of ports allowed, such as 16570 -17950.
External Authentication Server	External Authentication Server (EA) identifies the server where EA is installed. Select the EA server from the pull-down list. You must define an EA server before can select the server from the list.

Field Name	Description
Perimeter Server Mapping - Inbound Perimeter Server	Select the perimeter server to use for the inbound connection in the Perimeter Server Mapping - Inbound Perimeter Server field. To use a remote perimeter server, you must define the remote perimeter server before you can associate it with an inbound connection.
Perimeter Server Mapping - Outbound Perimeter Server	Select the perimeter server to use for the outbound connection in the Perimeter Server Mapping - Outbound Perimeter Server field. To use a remote perimeter server, you must define the remote perimeter server before you can associate it with an outbound connection.
Perimeter Server Mapping - External Authentication Perimeter Server	Select the perimeter server to use for the EA connection in the Perimeter Server Mapping - External Authentication Perimeter Server field. To use a remote perimeter server, you must define the remote perimeter server before you can associate it with an EA connection.

HTTP Adapter Definition - Properties

Use this screen to edit properties associated with how the HTTP protocol is implemented.

Refer to the field definitions in the following table. For additional information, refer to Chapter 8, *HTTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description	
Key	Properties that you can change for an adapter. Available keys include:	
	◆ block.exploit.strings.1=-- ◆ block.exploit.strings.2=| ◆ block.exploit.strings.3=' ◆ block.exploit.strings.4=" ◆ block.exploit.strings.5=; ◆ block.exploit.strings.6=<? ◆ block.exploit.strings.7=\u0000	If you enable block common exploits in an HTTP adapter, requests with URLs containing the strings defined in block.exploit.strings.# are blocked, by default.
	◆ http.commands.allowed	HTTP method allowed. Commands that are allowed by default include GET, HEAD, PUT, POST, TRACE, OPTIONS, and DELETE. WebDAV methods allowed are PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK, and UNLOCK. To identify more than one allowed method, separate each value with a comma.
	◆ http.commands.prohibited	HTTP methods, such as DELETE, MOVE, that are prohibited. By default, CONNECT is prohibited. To identify more than one prohibited method, separate each value with a comma.

Field Name	Description	
Key	◆ httpMaxHeaderFieldLength	Identifies the maximum length allowed for any HTTP header in the incoming HTTP request. The default value is 8192.
	◆ httpMaxNumHeaderFields	Identifies the maximum number of HTTP headers allowed in the incoming HTTP request. The default value is 1024.
	◆ max.html.rewrite.threads	The maximum number of threads in the pool used to service rewriting URLs in the HTML pages coming from the backend HTTP Server. The default is 10.
	◆ max.ps.client.threads	Maximum number of threads in the pool used during a connection with a client. Default value is 10.
	◆ max.ps.server.threads	Maximum number of threads in the pool used during a connection with a server. Default value is 10.
	◆ html.rewrite.threads	Number of threads used in the thread pool for HTML rewrite processing. Default is 10.
	◆ html.rewrite.threads.queue.size	The buffer size used to queue the request for HTML rewrite processing.
		Note: This parameter is not displayed. If you want to change its value, you must type the field in the Key field and the new value in the Value field.
Value	Value to assign to a property key. See Key for a list of properties you can modify.	

HTTP Netmap Definition

Use this screen to define the HTTP netmap. The netmap includes inbound and outbound node definitions, and HTML rewrite definitions.

✦ HTTP Netmap Inbound Node Definition- Basic—Click Inbound Nodes to add connection information for your external trading partner.

✦ HTTP Netmap Outbound Node Definition - Basic—Click Outbound Nodes to add connection information for your internal Connect:Direct server.

✦ HTML Rewrite Definition—Click HTML Rewrite to reroute an incoming URL request.

Refer to the field definitions in the following table. For additional information, refer to Chapter 8, *HTTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Note: You must define a netmap, at least one inbound node, and at least one outbound node before you can save the netmap. If you exit the application before all three elements are defined, you lose the netmap definition.

Field Name	Description
Netmap Name	Netmap Name identifies the name to assign to the netmap you create. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the netmap you create. Description can be up to 255 characters.
Type	Type identifies the protocol being used as HTTP.
Filter	Filter allows you to view a subset of available inbound or outbound nodes. Use the wildcard characters, * and ?, to identify the nodes to display. Filters are case-sensitive. For example, the filter i* will display inboundnode1 but will not display InboundNode1.

HTTP Netmap Inbound Node Definition- Basic

Use this screen to define the minimum HTTP connection requirements for an external trading partner.

Refer to the field definitions in the following table. For additional information, refer to Chapter 8, *HTTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Inbound Node Name	Inbound Node Name is the name associated with the inbound node connection. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the inbound node you create. Description can be up to 255 characters.

Field Name	Description
Peer Address Pattern	Peer Address Pattern identifies the pattern to allow for the inbound connections to SSP. Valid values are alphanumeric characters and the following special characters: dash(-), underscore(_), colon(:), period(.), dollar sign($), forward slash(/), exclamation mark(!), tilde(~), asterisk(*), open parenthesis '(', close parenthesis ')' semicolon(;), question mark(?), at(@), and comma(,). You can define one of the following types of patterns: ◆ Wildcard validates incoming DNS names. If a wildcard pattern is provided, SSP performs a reverse lookup on the incoming IP address and the DNS name is compared to the wildcard patterns. Wildcard characters allowed are ? and *. For example, *.a.com allows a connection from b.a.com but not from b.b.com ◆ IP/Subnet validates incoming IP addresses. Use the format IP-address/num-bits where IP-address identifies an IP address template and num-bits identifies the number of leading (highest-order) bits in the template that are significant. An IP match is performed by comparing the leading (highest-order) num-bits of the incoming IP address against num-bits of the template. For example, 10.20.0.0/16 searches for a match to the first 16 bits. All IP addresses beginning with 10.20.* are allowed. 10.0.0.0/8 searches for a match to the first 8 bits. All addresses beginning with 10.* are allowed. 0.0.0.0/0 searches for a match the first zero bits. All IP addresses are allowed.
Policy	Policy is a pull-down list of policies you have created. Select the policy you want to associate with the inbound node you are creating. If a policy with the security attributes required has not been created, click .

HTTP Netmap Inbound Node Definition- Security

Use this screen to define secure connection requirements for an external trading partner.

Refer to the field definitions in the following table. For additional information, refer to Chapter 8, *HTTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Secure Connection	Enable Secure Connection to turn on the use of SSL/TLS to provide secure communications with transport protocols and to ensure that data is secured as it is transmitted across a single socket.

Field Name	Description
Security Setting	Security Setting identifies the security protocol allowed for connections to this node. Options include: ◆ SSL v3 or TLS - select this option to send an TLS Hello and accept SSLv3 or TLS ◆ SSL v2 or v3 with v3 Hello - select this option to send an SSLv3 Hello and accept SSLv3 or SSLv2 ◆ SSL (any version) or TLS - select this option to send an SSL v2 Hello and accept SSLv3, SSLv2, or TLS ◆ SSL v2 or v3 - select this option to send SSLv2 Hello and accept SSLv3 or SSLv2 ◆ TLS - select this option to send TLS Hello and accept TLS only ◆ SSL v3 - select this option to send SSLv3 Hello and accept SSLv3 only
Enable Client Authentication	Enable Client Authentication on the inbound node connection to require that the SSP server authenticate the certificate presented by the inbound node connection.
Trust Store	Trust Store identifies the location where the system and CA certificates are stored. System and CA certificates are used during secure connection to verify that a certificate received from a server is signed by a trusted source.
CA Certificates/Trusted Root	CA Certificates/Trusted Root identifies the trusted certificate to use to authenticate the certificate presented by the client. You select a CA certificate or trusted root from the list of certificates stored in the trust store you selected in the Trust Store field. When a client presents a certificate to establish a secure connection, the trusted root certificate, located at the server, must match or be the entity who signed the certificate presented by the client during the SSL handshake.
Key Store	Key Store identifies the location where the key certificates you want to use are stored.
Key/System Certificate	Key/System Certificate identifies the certificate presented by SSP to the inbound node to authenticate itself during the SSL handshake or the certificate presented by the Connect:Direct server to authenticate itself to the SSP server. Select the Key/System Certificate to enable for the node from the list. The list identifies the key or system certificates stored in the key store you selected in the Key Store field.
Available Cipher Suites	Available Cipher Suites provides a list of ciphers that can be enabled to encrypt data transmitted during a secure SSL or TLS connection. Enable at least one cipher. To enable a cipher, highlight the cipher in the Available Cipher Suites dialog and click Add. To enable multiple ciphers, highlight the ciphers to enable and click Add.
Selected Cipher Suites	Selected Cipher Suites identifies the ciphers you have enabled to encrypt data during a secure SSL or TLS connection. A cipher suite is negotiated during a secure channel connection between a client and a server. Ciphers are negotiated based on their location in the Selected Ciphers list. To reorder a cipher in the list, highlight the cipher to reorder and click the Up or Down button.

HTTP Netmap Inbound Node Definition- Advanced

Use this screen to specify what level to log information on this inbound connection.

Refer to the field definitions in the following table. For additional information, refer to
Chapter 8, *HTTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Node Logging Level	Node Logging Level identifies the level of logging to write to the log file for the inbound node. Logging options include: ◆ NONE turns logging off. ◆ ERROR writes only error messages to the log. ◆ WARN writes error and warning messages to the log. ◆ INFO writes error and informational messages to the log. ◆ DEBUG writes all messages to the log including debugging messages. Use this logging level when instructed to turn it on by Sterling Commerce Support.

HTTP Netmap Outbound Node Definition - Basic

Use this screen to define the minimum connection requirements for your internal HTTP server.

Refer to the field definitions in the following table. For additional information, refer to
Chapter 8, *HTTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Outbound Node Name	Outbound Node Name identifies the name of the outbound server in the secure zone which is the destination of the communications session. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the outbound node you create. Description can be up to 255 characters.
Primary Destination Address	Primary Destination Address identifies the IP address or host name to use to connect to the Connect:Direct outbound server. Valid values are 1-200 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), colon (:), and underscore (_).
Primary Destination Port	Primary Destination Port identifies the port to use to connect to the secure Connect:Direct server. Valid values are 1-65535.

HTTP Netmap Outbound Node Definition - Security

Use this screen to define the secure connection requirements for your internal HTTP server. Refer
to the field definitions in the following table. For additional information, refer to Chapter 3, *Manage
Certificates for SSL/TLS Transactions with Trading Partners* of the *Sterling Secure Proxy
Configuration Guide*.

For more information on configuring an HTTP Netmap Node Definition, refer to Chapter 8, *HTTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Secure Connection	Enable Secure Connection to turn on the use of SSL/TLS to provide secure communications with transport protocols and to ensure that data is secured as it is transmitted across a single socket.
Security Setting	Security Setting identifies the security protocol allowed for connections to this node. Options include: ◆ SSL v3 or TLS - select this option to send an TLS Hello and accept SSLv3 or TLS ◆ SSL v2 or v3 with v3 Hello - select this option to send an SSLv3 Hello and accept SSLv3 or SSLv2 ◆ SSL (any version) or TLS - select this option to send an SSL v2 Hello and accept SSLv3, SSLv2, or TLS ◆ SSL v2 or v3 - select this option to send SSLv2 Hello and accept SSLv3 or SSLv2 ◆ TLS - select this option to send TLS Hello and accept TLS only ◆ SSL v3 - select this option to send SSLv3 Hello and accept SSLv3 only
Trust Store	Trust Store identifies the location where the system and CA certificates are stored. System and CA certificates are used during secure connection to verify that a certificate received from a server is signed by a trusted source.
CA Certificate/Trusted Root	CA Certificate/Trusted Root identifies the trusted certificate to use to authenticate the certificate presented by the client. You select a CA certificate or trusted root from the list of certificates stored in the trust store you selected in the Trust Store field. When a client presents a certificate to establish a secure connection, the trusted root certificate, located at the server, must match or be the entity who signed the certificate presented by the client during the SSL handshake.
Key Store	Key Store identifies the location where the key certificates you want to use are stored.
Key/System Certificate	Key/System Certificate identifies the certificate presented by SSP to the inbound node to authenticate itself during the SSL handshake or the certificate presented by the Connect:Direct server to authenticate itself to the SSP server. Select the Key/System Certificate to enable for the node from the list. The list identifies the key or system certificates stored in the key store you selected in the Key Store field.
Available Ciphers	Available Ciphers provides a list of ciphers that can be enabled to encrypt data transmitted during a secure SSL or TLS connection. Enable at least one cipher. To enable a cipher, highlight the cipher in the Available Cipher Suites dialog and click Add. To enable multiple ciphers, highlight the ciphers to enable and click Add.
Selected Ciphers	Selected Ciphers identifies the ciphers you have enabled to encrypt data during a secure SSL or TLS connection. A cipher suite is negotiated during a secure channel connection between a client and a server. Ciphers are negotiated based on their location in the Selected Ciphers list. To reorder a cipher in the list, highlight the cipher to reorder and click the Up or Down button.

HTTP Netmap Outbound Node Definition - Advanced

Use this screen to define advanced parameters for your internal HTTP server.

For more information on configuring an HTTP Netmap Node Definition, refer to Chapter 8, *HTTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Node Logging Level	Node Logging Level identifies the level of logging to write to the log file for the outbound node. Logging options include: ◆ NONE turns logging off. ◆ ERROR writes only error messages to the log. ◆ WARN writes error and warning messages to the log. ◆ INFO writes error and informational messages to the log. ◆ DEBUG writes all messages to the log including debugging messages. Use this logging level when instructed to turn it on by Sterling Commerce Support.
Destination Service Name	Destination Service Name identifies a destination server that can be accessed by the outbound node, when using EA to map a user ID and password. Valid values are 1-255 alphanumeric characters and certain special characters. The following special characters are not allowed: ! @ # % ^ * ( ) + ? , < > { } [ ]	; " '
User ID	User ID identifies the user ID to use to connect to the secure outbound server, if the policy is defined to required that the user ID and password from the netmap be used. Valid values are 1-255 alphanumeric characters and certain special characters. The following special characters are not allowed: ! @ # % ^ * ( ) + ? , < > { } [ ]	; " '
Password	Password identifies the password to use to connect to the secure outbound server, if the policy is defined to required that the user ID and password from the netmap be used. Valid values are 1-255 alphanumeric characters and certain special characters. The following special characters are not allowed: , " '
Alternate Destinations - Node	Alternate Destinations Node 1 identifies the node name or IP address and port to use to connect to an alternate Connect:Direct outbound server, if a connection to the primary node cannot be made. Up to three alternate destination nodes can be defined for each outbound node. To use different security and EA definitions for the alternate destination node, first configure an outbound node definition for the alternate node in the netmap. Then, open the primary outbound node definition and select the alternate node name from the drop-down list in the Alternate Destinations - Node 1 field. To use the security and EA definition defined in the primary outbound node for an alternate destination node, you do not have to define the alternate node in the netmap. Select IP Address/Port from the drop down list and then provide a value in the IP Address and Port fields. Define up to 3 alternate node names.

Field Name	Description
Alternate Destinations - IP Address	Alternate Destinations Address identifies the IP address to use to connect to an alternate destination node, if a connection to the primary node cannot be made. Up to three alternate destination nodes can be selected. Valid values are 1-200 alphanumeric characters and special characters: underscore (_), dash (-), colon (:), and period (.). If you provide an IP address and port as an alternate destination and a connection to the alternate node is attempted, the security and EA definition from the primary node is used for the connection.
Alternate Destinations - Port	Alternate Destinations Port identifies the port to use to connect to an alternate destination node, if a connection to the primary node cannot be made. Up to three alternate destination nodes can be selected. Valid values are 1-65535. If you provide an IP address and port as an alternate destination and a connection to the alternate node is attempted, the security and EA definition from the primary outbound node is used for the connection.

HTML Rewrite Definition

Use this screen to route incoming URL requests to a the URL of your internal HTTP server. Click New to define a new Server-Proxy URL pair.

Refer to the field definitions in the following table. For additional information, refer to Chapter 8, *HTTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Support HTML Rewrite	Enable Support HTML Rewrite to allow HTML client connections to replace an HTML value that routes a connection to a GIS server with a value at the SSP.
Server URL	Server URL identifies the URL of the HTTP server where the inbound connection is attempting to connect. Because the inbound node does not have access to the HTTP server, you must identify the proxy URL to use when a request to the Server URL is presented.
Proxy URL	Proxy URL identifies the URL of the SSP server where the inbound connection is redirected when it requests a connection to the Server URL.

HTTP Policy Configuration

Use this screen and its tabs to define how you authenticate a trading partner to access an HTTP server.

HTTP Policy Configuration- Basic

Use this screen to define basic authentication information.

Refer to the field definitions in the following table. For additional information, refer to Chapter 8, *HTTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Policy Name	Policy Name identifies the name to assign to the policy you create. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the policy you create. Description can be up to 255 characters.
Type	Type identifies the protocol being used as HTTP.

HTTP Policy Configuration- Advanced

Use this tab to specify the type of user authentication to use for inbound access requests. For Certificate Authentication and User Authentication through External Authentication, you must have installed and configured EA.

Refer to the field definitions in the following table. For additional information, refer to Chapter 8, *HTTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
External Authentication Certificate Validation	Turn on External Authentication Certificate Validation To validate information presented in certificates received from trading partners using EA.
External Authentication Profile	External Authentication Profile identifies the name of the certificate validation definition you defined in the EA. You must enable certificate validation before you can provide a profile.
Through External Authentication	Turn on User Authentication through External Authentication to send an incoming user ID and password to EA for validation. EA validates the user ID and password using one of the following methods: ◆ An LDAP bind to authenticate a user and password ◆ An LDAP search based on the user ID, destination service name, or certificate subject and issues attributes ◆ Fully specified Distinguished Name (DN) or a DN returned from a certificate validation search
External Authentication Profile	If you enabled user authentication through EA, identify the certificate authentication profile you defined in EA.
Through Local User Store	Validates the user ID and password of the inbound node using information defined in the user store. You must add the user to the user store to successfully use this method.

Field Name	Description
Internal User ID	This option determines what user ID and password is used to attach to the server in the secure environment. For the user ID and password presented to the GIS server to successfully access the server, a user definition must be defined at the GIS server. User mapping options include: ◆ User ID/Password passed through from client—Uses the user ID and password supplied by the inbound node to connect to the server in the secure zone. To successfully connect to the server, the user ID and password must be defined in the user store at the server. ◆ User ID/Password From External Authentication—Uses a user ID and password from EA to connect to the server. To successfully connect using this option, the user ID and password must be defined in the LDAP database. ◆ User ID/Password from netmap—Uses the user ID and password defined in the netmap to connect to the outbound server. To successfully connect using this option, define the user ID and password to use in the outbound node definition.
Block Common Exploits	Enable Block Common Exploits to scan inbound URI queries for any of the most commonly occurring strings. If a match is found, the request is rejected and the connection is closed. Default blocked strings include: (_), (|),('),("),(;),(<), (?). To modify the common exploits that are block, modify the properties in an adapter.

SFTP Protocol Field Definitions

Following are the field definitions for the SFTP protocol screens.

SFTP Adapter Configuration - Basic

Use this screen to specify system-level communications information for SFTP connections to and from SSP. Refer to the field definitions in the following table.

For more information on configuring an SFTP Adapter, refer to Chapter 9, *SFTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

For more information on configuring your SSH keys, refer to Chapter 4, *Manage SSH Keys for SFTP Transactions* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Adapter Name	Adapter Name identifies the name to assign to the adapter you create. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).

Field Name	Description
Description	Description assigns a description to help you identify the adapter you create. Description can be up to 255 characters.
Type	Type identifies the protocol being used as SFTP.
Listen Port	Listen Port identifies the port number to use to listen for inbound connections. Valid values include 1-65535. If you make changes to the listen port, you must restart the adapter before the change is recognized.
Netmap	Netmap identifies the name of the netmap to associate with the adapter you are defining. If the netmap has not been created, click to add the netmap.
Standard Routing Node	Standard Routing Node identifies the name of the SFTP outbound node where the inbound node connections are routed, after connecting to SSP. Select this value from a pull-down list.
Engine	Engine identifies the SSP server in the DMZ where traffic is first routed before being sent to the outbound secure SFTP server. Select an engine from the list.
Startup Mode	Startup Mode identifies how the adapter is started. **auto** starts the adapter as soon as it is pushed to the engine. **manual** requires that the adapter be manually started.
Local Host Key Store	Local Host Key Store identifies the local host key store you created to store local host keys. Select the local host key store that contains the local host key to use to authenticate SSP to the inbound node connections.
Local Host Key	Local Host Key identifies the local host key you have added to the key store. Select the local host key from the drop-down list that will used to authenticate SSP to the inbound node connections. If you make changes to the local host key, you must restart the adapter before the change is recognized.

SFTP Adapter Configuration - Security

Use this screen to specify security information for SFTP connections to and from SSP. Refer to the field definitions in the following table.

For more information on configuring an SFTP Adapter, refer to Chapter 9, *SFTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Available Cipher Suites	Available Cipher Suites provides a list of ciphers that can be enabled to encrypt data transmitted during a secure SSH connection. Enable at least one cipher.
	To enable a cipher, highlight the cipher in the Available Cipher Suites dialog and click Add. To enable multiple ciphers, highlight the ciphers to enable and click Add.

Field Name	Description
Selected Cipher Suites	Selected Cipher Suites identifies the ciphers you have enabled to encrypt data during a secure connection. A cipher suite is negotiated during a secure channel connection between a client and a server. Ciphers are negotiated based on their location in the Selected Ciphers list. To reorder a cipher in the list, highlight the cipher to reorder and click the Up or Down button. If you make changes to the selected cipher suites, you must restart the adapter before the change is recognized. **Note:** If you define multiple SFTP adapters on an engine, all adapters must define a common set of ciphers. Unintended results may occur if this parameter is defined differently.
Available MAC Suites	Available MAC Suites provides a list of MACs that can be enabled to provide message integrity protection. Enable at least one MAC. To enable a MAC, highlight the MAC in the Available MAC Suites dialog and click Add. To enable multiple MACs, highlight the MACs to enable and click Add.
Selected MAC Suites	Selected MAC Suites identifies the MACs you have enabled to provide message integrity protection. MACs are negotiated based on their location in the Selected MAC Suites list. To reorder a MAC in the list, highlight the MAC to reorder and click the Up or Down button. If you make changes to the selected MAC suites, you must restart the adapter before the change is recognized. **Note:** If you define multiple SFTP adapters on an engine, all adapters must define a common set of MACs. Unintended results may occur if this parameter is defined differently.
Available Key Exchange	Identifies the available key exchanges you can configure for an SFTP adapter.
Selected Key Exchange	Identifies the key exchanges that have been configured. Key exchanges are used in the order in which they are selected in this field. To reorder a key exchange in the list, highlight the exchange to reorder and click the Up or Down button.

SFTP Adapter Configuration - Advanced

Use this screen to specify additional communications information, and to specify the perimeter servers to use for this adapter. Refer to the field definitions in the following table.

For more information on configuring an SFTP Adapter, refer to Chapter 9, *SFTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.* For more information on perimeter servers, refer to Chapter 4, *Configure Perimeter Servers to Manage SSP Communications.* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Logging Level	Level of logging to write to the log file for the adapter. Logging options include: ◆ ERROR writes only error messages to the log. ◆ WARN writes error and warning messages to the log. ◆ INFO writes error and informational messages to the log. ◆ DEBUG writes all messages to the log including debugging messages. Use this logging level when instructed to turn it on by Sterling Commerce Support.
Maximum Sessions	Maximum number of sessions that the adapter allows. The default is 20. If you make changes to the maximum sessions, you must restart the adapter before the change is recognized.
Session Timeout	Amount of time allowed, in minutes, between TCP packets before a session is terminated. The default is 3 minutes. If you make changes to the session timeout, you must restart the adapter before the change is recognized.
Pre-Authentication Banner Text	The remote software version to send when an inbound connection is made. The text is sent to the client before server authentication is performed. Valid values are 0-255 alphanumeric and special characters. The default value is SSHD_Maverick.
Post-Authentication Banner Text	Text displayed when an inbound connection is made. It is sent to the client after server authentication is performed. Valid values are 0-4000 alphanumeric and special characters. No default value is defined.
External Authentication Server	Server where EA is installed. Select the EA server from the pull-down list. You must define an EA server before can select the server from the list.
Compression	Compression method to use to compact files before they are transmitted and is negotiated with the client. Compression methods include none or Zlib. The default is none. If you make changes to the compression value, you must restart the adapter before the change is recognized. If you select Zlib, both Zlib and None are supported. **Note:** If you define multiple SFTP adapters on an engine, you must set compression to the same value for each adapter. Unintended results may occur if this parameter is defined differently.
Outbound Port Range	Range of ports to use for the adapter. Valid values include a list of ports separated by commas, such as 1234, 2340, 16570, or a range of ports, such as 16570 -17950.
Perimeter Server Mapping - Inbound Perimeter Server	The perimeter server to use for the inbound connection in the Perimeter Server Mapping - Inbound Perimeter Server field. To use a remote perimeter server, you must define the remote perimeter server before you can associate it with an inbound connection. If you make changes to the inbound perimeter server, you must restart the adapter before the change is recognized.

Field Name	Description
Perimeter Server Mapping - Outbound Perimeter Server	The perimeter server to use for the outbound connection in the Perimeter Server Mapping - Outbound Perimeter Server field. To use a remote perimeter server, you must define the remote perimeter server before you can associate it with an outbound connection. If you make changes to the outbound perimeter server, you must restart the adapter before the change is recognized.
Perimeter Server Mapping - External Authentication Perimeter Server	Select the perimeter server to use for the EA connection in the Perimeter Server Mapping - External Authentication Perimeter Server field. To use a remote perimeter server, you must define the remote perimeter server before you can associate it with an EA connection. If you make changes to the EA perimeter server, you must restart the adapter before the change is recognized.

SFTP Adapter Definition - Properties

Use this screen to edit the default values assigned to properties used to determine how the SFTP protocol is implemented. The keys are not displayed. To change a default key value, type the key value as defined in the following table and assign a value to the key.

For more information on configuring an SFTP Adapter, Chapter 9, *SFTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Key	Key identifies properties that you can change for an adapter. Available keys include:
	◆ max.ps.client.threads — Maximum number of threads in the pool used during a connection with a client. Default is 10.
	◆ max.ps.server.threads — Maximum number of threads in the pool used during a connection with a server. Default is 10.
	◆ sftp_invalidadapter — Identifies how many users can perform an unsuccessful log in attempt before all log in attempts to the adapter fail.
	◆ sftp_rekeycount — Identifies how many packets are transmitted before a key renegotiation is performed. The default is 20.000.
	◆ sftp_threadpoolsize — A tuning parameter that defines the minimum thread pool size.
	◆ sftp_selectorthreads — A tuning parameter to determine how backend threads are managed.
	◆ sftp_maxchannels — Identifies the maximum channels allowed for an SFTP server thread. The default is 3.
	◆ sftp_acceptthreads — Identifies how many threads are available to accept inbound client connections. The default value is 50.

Field Name	Description
◆ sftp_connectthreads	Identifies how many threads are available for permanent connect threads. When existing SSH connections make socket connections through port forwarding, these threads manage the asynchronous connection process. The default value is 50.
◆ sftp_xferthreadpools	Identifies how many threads are available for permanent transfers. This thread asynchronously performs the IO for the socket. The default value is 50.
◆ sftp_maxauthentications	Identifies how many authentication attempts can be established before the SFTP server closes the connection. This parameter does not lock out a user. The default is 10.
◆ sftp_maxPacketLength	Identifies the maximum SFTP packet size supported. The default is 65535.
Value	Value identifies the value to assign to a property key. See Key for a list of properties you can modify.

SFTP Netmap Definition

Use this screen to define the SFTP netmap. Click Inbound Nodes to add connection information for your external trading partner. Click Outbound Nodes to add connection information for your internal SFTP server. Refer to the field definitions in the following table.

For more information on configuring an SFTP Netmap Definition, refer to Chapter 9, *SFTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Note: You must define a netmap, at least one inbound node, and at least one outbound node before you can save the netmap. If you exit the application before all three elements are defined, you lose the netmap definition.

Field Name	Description
Netmap Name	Netmap Name identifies the name to assign to the netmap you create. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the netmap you create. Description can be up to 255 characters.
Type	Type identifies the protocol being used as SFTP.
Filter	Filter allows you to view a subset of available inbound or outbound nodes. Use the wildcard characters, * and ?, to identify the nodes to display. Filters are case-sensitive. For example, the filter i* will display inboundnode1 but will not display InboundNode1.

SFTP Netmap Inbound Node Definition - Basic

Use this screen to define the minimum SFTP connection requirements for an external trading partner. Refer to the field definitions in the following table.

For more information on configuring an SFTP Netmap Node, refer to Chapter 9, *SFTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Inbound Node Name	Inbound Node Name assigns a name to the inbound node connection. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the inbound node you create. Description can be up to 255 characters.
Peer Address Pattern	Peer Address Pattern identifies the pattern to allow for the inbound connections to SSP. Valid values are alphanumeric characters and the following special characters: dash(-), underscore(_), colon(:), period(.), dollar sign($), forward slash(/), exclamation mark(!), tilde(~), asterisk(*), open parenthesis '(', close parenthesis ')' semicolon(;), question mark(?), at(@), and comma(,). You can define one of the following patterns: ◆ Wildcard validates incoming DNS names. If a wildcard pattern is provided, SSP performs a reverse lookup on the incoming IP address and the DNS name is compared to the wildcard patterns. Wildcard characters allowed are ? and *. For example, *.a.com allows a connection from b.a.com but not from b.b.com ◆ IP/Subnet validates incoming IP addresses. Use the format *IP-address*/*num-bits* where IP-address identifies an IP address template and num-bits identifies the number of leading bits in the template that are significant. An IP match is performed by comparing the leading num-bits of the incoming IP address against num-bits of the template. For example, 10.20.0.0/16 searches for a match to the first 16 bits. All IP addresses beginning with 10.20.* are allowed. 10.0.0.0/8 searches for a match to the first 8 bits. All addresses beginning with 10.* are allowed. 0.0.0.0/0 allows connections from all IP addresses.
Policy	Policy is a pull-down list of policies you have created. Select the policy you want to associate with the inbound node you are creating. If a policy with the security attributes required has not been created, click .

SFTP Netmap Inbound Node Definition- Advanced

Use this screen to specify what level to log information on this inbound connection. Refer to the field definitions in the following table.

For more information on configuring an SFTP Netmap Node, refer to Chapter 9, *SFTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Node Logging Level	Node Logging Level identifies the level of logging to write to the log file for the inbound node. Logging options include: ◆ NONE turns logging off. NONE is the default value. ◆ ERROR writes only error messages to the log. ERROR is the default value. ◆ WARN writes error and warning messages to the log. ◆ INFO writes error and informational messages to the log. ◆ DEBUG writes all messages to the log including debugging messages. Use this logging level when instructed to turn it on by Sterling Commerce Support.

SFTP Netmap Outbound Node Definition - Basic

Use this screen to define the minimum connection requirements for your internal SFTP server. Refer to the field definitions in the following table.

For more information on configuring an SFTP netmap node, refer to Chapter 9, *SFTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Outbound Node Name	Name of an outbound SFTP server in the secure zone. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description to help you identify the outbound node you create. Description can be up to 255 characters.
Primary Destination Address	IP address or host name to use to connect to the SFTP outbound server. Valid values are 1-200 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), colon (:), and underscore (_).
Primary Destination Port	Port to use to connect to the secure SFTP server. Valid values are 1-65535.
Known Host Key Store	Key store you created to store the public keys of GIS servers. Select the known host key store that contains the known host key to use to authenticate the outbound GIS server that you are defining in the outbound node definition.
Known Host Key	Known host keys you added to the key store. Select the known host key from the drop-down list used to authenticate the GIS server to SSP.

SFTP Netmap Outbound Node Definition - Security

Use this screen to define the secure connection requirements for your internal SFTP server. Refer to the field definitions in the following table.

For more information on configuring an SFTP Netmap Node, refer to Chapter 9, *SFTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Available Cipher Suites	Available Cipher Suites provides a list of ciphers that can be enabled to encrypt data transmitted during a secure SSH connection. Enable at least one cipher. To enable a cipher, highlight the cipher in the Available Cipher Suites dialog and click Add. To enable multiple ciphers, highlight the ciphers to enable and click Add.
Selected Cipher Suites	Selected Cipher Suites identifies the ciphers you have enabled to encrypt data during a secure connection. A cipher suite is negotiated during a secure channel connection between a client and a server. Ciphers are negotiated based on their location in the Selected Ciphers list. To reorder a cipher in the list, highlight the cipher to reorder and click the Up or Down button. To remove a cipher from the selected list, highlight the cipher and click Remove.
Available MAC Suites	Available MAC Suites provides a list of MACs that can be enabled to provide message integrity protection. Enable at least one MAC. To enable a MAC, highlight the MAC in the Available MAC Suites dialog and click Add. To enable multiple MACs, highlight the MACs to enable and click Add.
Selected MAC Suites	Selected MAC Suites identifies the MACs you have enabled to provide message integrity protection. MACs are negotiated based on their location in the Selected MAC Suites list. To reorder a MAC in the list, highlight the MAC to reorder and click the Up or Down button. To remove a MAC from the selected list, highlight the cipher and click Remove.
Available Key Exchange	The available key exchanges you can configure for an SFTP outbound node.
Selected Key Exchange	Identifies the key exchanges that have been configured. Key exchanges are used in the order in which they are selected in this field. To reorder a key exchange in the list, highlight the exchange to reorder and click the Up or Down button.

SFTP Netmap Outbound Node Definition - Advanced

Use this screen to define advanced parameters for your internal SFTP server. Refer to the field definitions in the following table.

For more information on configuring an SFTP Netmap Node, refer to Chapter 9, *SFTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Node Logging Level	Node Logging Level identifies the level of logging to write to the log file for the outbound node. Logging options include: ◆ NONE turns logging off. NONE is the default value. ◆ ERROR writes only error message to the log. ◆ WARN writes error messages and warning messages to the log. ◆ INFO writes error messages and informational messages to the log. ◆ DEBUG writes all messages to the log including debugging messages. Use this logging level when instructed to turn it on by Sterling Commerce Support.
Destination Service Name	Destination Service Name identifies a destination service accessed by the outbound node. This value can be sent to the EA to use when authenticating a user for access to specific services on the GIS server. The following special characters are not allowed: ! @ # % ^ * ( ) + ? , < > { } [ ] | ; " '
User ID	User ID to use to connect to the outbound server, if the policy is defined to require that the user ID and password from the netmap be used. Valid values are 1-255 alphanumeric characters and certain special characters. The following special characters are not allowed: ! @ # % ^ * ( ) + ? , < > { } [ ] | ; " '
Password	Password identifies the password to use to connect to the outbound server, if the policy is defined to require that the user ID and password from the netmap be used. Valid values are 1-255 alphanumeric characters and certain special characters. The following special characters are not allowed: , " '
Local User Key Stores	Local User Key Stores identifies the name of the key store where the key to authenticate SSP to the outbound connection is stored. Select the local user key store from a drop-down list.
Local User Key	Local User Key identifies the local user key to use to authenticate SSP to the outbound connection. Select the local user key from the drop-down list.
Compression	Compression identifies the compression method to use to compact files before they are transmitted to the outbound node. Compression methods include none or Zlib.

Field Name	Description
Alternate Destinations - Node	Alternate Destinations Node identifies the node name or IP address and port to use to connect to an alternate SFTP outbound server if a connection to the primary node cannot be made. Up to three alternate destination nodes can be defined for each outbound node.
	To use different security and advanced definitions for the alternate destination node connection, first configure an outbound node definition for the alternate node in the netmap. Then, open the primary outbound node definition and select the alternate node name in the Alternate Destinations - Node field.
	To use the security and advanced definition defined in the primary outbound node for an alternate destination node connection, you do not have to define the alternate node in the netmap. Select IP Address/Port and then provide a value in the IP Address and Port fields.
Alternate Destinations - IP Address	Alternate Destinations - IP Address identifies the IP address to use to connect to an alternate destination node, if a connection to the primary node cannot be made. Up to three alternate destination nodes can be defined. Valid values are 1-200 alphanumeric characters and special characters: underscore (_), dash (-), colon (:), and period (.).
	If you provide an IP address and port as an alternate destination and a connection to the alternate node is attempted, the security and advanced definition from the primary node is used for the connection.
Alternate Destinations - Port	Alternate Destinations Port identifies the port to use to connect to an alternate destination node if a connection to the primary node cannot be made. Up to three alternate destination nodes can be defined. Valid values are 1-65535.
	If you provide an IP address and port as an alternate destination and a connection to the alternate node is attempted, the security and advanced definition from the primary outbound node is used for the connection.

SFTP Policy Configuration

Use this screen and its tabs to define how to authenticate a trading partner trying to access your SFTP server.

SFTP Policy Configuration - Basic

Use this screen to define how you impose controls to authenticate a trading partner trying to access your SFTP server. Refer to the field definitions in the following table.

For more information on configuring an SFTP Policy Configuration, refer to Chapter 9, *SFTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Policy Name	Policy Name identifies the name to assign to the policy you create. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the policy you create. Description can be up to 255 characters.
Type	Type identifies the protocol being used as SFTP.

SFTP Policy Configuration - Advanced

Use this tab to specify the type of user authentication to use for inbound access requests. For Certificate Authentication and User Authentication through External Authentication, you must have installed and configured EA. You can also use this screen to map an incoming user ID and password to a different user ID and password to present to the internal server.

For more information on configuring an SFTP Policy Configuration, refer to Chapter 9, *SFTP Reverse Proxy Configuration* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Required Authentication Method	Required Authentication Methods identifies the method to use to authenticate the inbound node connection. Valid values include: ◆ Password - select Password to require that the inbound node password be authenticated against information stored in the local user store or in EA. ◆ Key - select Key to require that the inbound node present a key authenticated against information stored in the local user store or in EA. ◆ Password and Key - select Password and Key to require that the password and the key presented by the inbound node be authenticated against information stored in the local user store or in EA. ◆ Password or Key - select Password or Key to require that the inbound node connection present either a password or a key authenticated against information stored in the local user store.
User Authentication Mechanism - Through External Authentication	Turn on User Authentication Mechanism - Through External Authentication (EA) to send an incoming user ID and password to EA for validation. EA validates the user ID and password using one or more of the following methods: ◆ An LDAP bind to authenticate a user and password ◆ An LDAP search, based on the user ID or destination service name
User Authentication Mechanism - User Authentication Profile	If you enabled user authentication through EA, identify the user authentication profile you defined in EA.
User Authentication Mechanism - Key Authentication Profile	If you enabled key authentication through EA, identify the key authentication profile you defined in EA.
User Authentication Mechanism - Through Local User Store	User Authentication Mechanism - Through Local User Store validates the user ID and password and/or the public key of the inbound node using information defined in the user store. You must add the user to the user store in order to successfully use this method.

Field Name	Description
User Mapping - Internal User ID	User Mapping - Internal User ID is enabled in the policy and determines what user ID and password is used to attach to the SFTP server in the secure environment. In order for the user ID and password presented to the GIS server to successfully access the server, a user definition must be defined at the GIS server. User mapping options include: ◆ Pass-Through uses the user ID and password supplied by the inbound node to connect to the SFTP server in the secure zone. To successfully connect to the SFTP server, the user ID and password must be defined in the user store at the server. ◆ External Authentication uses a user ID and password and/or key from EA to connect to the SFTP server. To successfully connect using this option, the user ID and password must be defined in the LDAP database. ◆ Netmap uses the user ID and password and private key defined in the netmap to connect to the SFTP server. To successfully connect using this option, define the user ID and password and key to use to connect to the SFTP server in the outbound node definition.

Monitoring Field Definitions

Use the following screens to view information on configured engines, manually push a configuration to an engine, view adapters configured at an engine, or stop an engine.

Engine Status (All)

Adapters are configured at CM and then pushed to the engine. The Engine Status Page for all Engines provides information on the engines that are configured including when configuration files were pushed to the engine, the version of the configuration file at CM and at the engine. You can also use the engine status page to manually push a configuration to an engine and to stop an engine. Refer to the *Sterling Secure Proxy Operations Guide* for instructions.

Field Name	Description
Refresh Interval (secs)	How often CM polls the engine to obtain updates and how often the display is refreshed. The Engine Status window is not a real time display. The default polling interval is 30 seconds.
Engine Name	Name of the engine for which information is displayed.
Last Pushed	Date and time when the last configuration file was sent to the engine.
Message	Message returned from the engine.
CM Ver	Version of the configuration file stored at CM.
Engine Ver	Version of the configuration file stored at the engine. If the CM Ver and the Engine Ver do not match, manually push the configuration.

Field Name	Description
Refresh	Click Refresh to manually update the display.
Stop Engine	Select an engine from the list and click Stop Engine to stop the engine.
Push Config	Select an engine from the list and click Push Config to manually push the adapter configuration from CM to the engine.

Engine Detail

The Engine Detail Page provides information on an engine that has been configured including adapters that are configured at the engine, the type of adapter and the port where it is configured. You can also use this tab to stop an adapter.

Field Name	Description
Poll Interval (secs)	Poll Interval (secs) identifies how often to refresh the display.
Refresh	Click Refresh to manually update the display.
Adapter Name	Adapter lists the adapters that are configured at the engine.
Type	Type identifies the type of adapter that is configured at the engine.
Port	Port identifies the port where the adapter is configured.
Message	Message displays the last message that was returned from the engine.
Action	Action allows you to start or stop an adapter. Click Stop to stop the adapter. If the adapter is not running, click Start to start the adapter.

Credentials Field Definitions

Use the following screens to view information about a trusted certificate store, trusted certificate configuration, or system certificate store.

Trusted Certificate Store Configuration

The Trusted Certificate Stores dialog shows you the name and description of the selected Trusted Certificate Store. The certificates that are in the store are displayed in a table, with a radio button indicating the active certificate. From this screen, you can change the active certificate. You can also add, edit, copy or delete a certificate. Refer to the field definitions in the following table.

For additional information on configuring certificates, refer to Chapter 3, *Manage Certificates for SSL/TLS Transactions with Trading Partners* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Trusted Certificate Store Name	Trusted Certificate Store Name identifies the name to assign to the certificate store you create. The name can be up to 150 characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the certificate store you create. Description can be up to 255 characters.

Trusted Certificate Configuration

Use this screen to update a certificate currently in the trust store or to create a new certificate in the trust store. Refer to the field definitions in the following table.

For additional information on configuring certificates, refer to Chapter 3, *Manage Certificates for SSL/TLS Transactions with Trading Partners*.

Field Name	Description
Enable Certificate	Check Enable Certificate to allow the certificate to be used to authorize a secure communications session.
Trusted Certificate Name	Trusted Certificate Name identifies the name to associate with the trusted certificate you are adding to the certificate store. Trusted Certificate Name can be up to 150 characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the certificate you create. Description can be up to 255 characters.
Import from file	Import from file identifies the location and certificate to import to the certificate definition. Use the Browse button to locate the file.
Certificate Data	Certificate Data displays the contents of the certificate that you imported.

System Certificate Store Configuration

The System Certificate Stores dialog shows you the name and description of the selected System Certificate Store. The private keys that are in the store are displayed in a table, with a radio button indicating the active key. From this screen, you can change the active key. You can also add, edit, copy or delete a key. Refer to the field definitions in the following table.

For additional information on configuring certificates, refer to Chapter 3, *Manage Certificates for SSL/TLS Transactions with Trading Partners* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
System Certificate Store Name	System Certificate Store Name identifies the name to assign to the certificate store you create. The name can be up to 150 characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the certificate store you create. Description can be up to 255 characters.

System Certificate Configuration

Use this screen to update a software key or HSM key. Click the Software Keys tab to update a software key currently in the trust store or create a new key in the key store. Click the HSM Keys tab to view an HSM key currently in the trust store or change the description of an HSM key.

Software Keys

From the Software Keys tab, you can add or update a software key. For additional information on configuring keys, refer to Chapter 3, *Manage Certificates for SSL/TLS Transactions with Trading Partners* of the *Sterling Secure Proxy Configuration Guide*.

Refer to the field definitions in the following table:

Field Name	Description
Enable Certificate	Check Enable Certificate to allow the certificate to be used to authorize a secure communications session.
System Certificate Name	System Certificate Name identifies the name to associate with the system certificate you are adding to the certificate store. The name can be up to 150 characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the certificate you create. Description can be up to 255 characters.
Import from File	Import from file identifies the location and certificate to import to the certificate definition. Use the Browse button to locate the file.
Certificate Data	Certificate Data displays the contents of the certificate that you imported.

HSM Keys

From the HSM Keys tab, you can view an HSM key currently in the trust store or modify the description of an HSM key.

For additional information on configuring HSM keys, refer to *Chapter 4, Store System Certificates on a Hardware Security Module (HSM)*.

Refer to the field definitions in the following table:

Field Name	Description
Enable Certificate	Check Enable Certificate to allow the certificate to be used to authorize a secure communications session.
System Certificate Name	Name of the HSM certificate you are viewing. This information cannot be edited.
Description	Description to help you identify the certificate you view. Description can be up to 255 characters. You can edit the HSM key description.
Certificate Data	Contents of the certificate that you imported. This information cannot be edited.

Authorized User Key Store Configuration

The Authorized User Key Store dialog shows you the name and description of the selected Authorized User Key Store. The keys that are in the store are displayed in a table with a radio button indicating the active key. From this screen, you can change the active key. You can also add, edit, copy or delete a key. Refer to the field definitions in the following table.

For additional information on configuring keys, refer to Chapter 4, *Manage SSH Keys for SFTP Transactions* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Authorized User Key Store Name	Authorized User Key Store Name identifies the name to assign to the key store you create. Authorized User Key Store Name can be up to 150 characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the user key store you create. Description can be up to 255 characters.

Authorized User Key Configuration

Use this screen to update a key currently in the Authorizes User Key Store or to create a new key in the store. Refer to the field definitions in the following table.

For additional information on configuring keys, refer to *Manage SSH Keys for SFTP Transactions* on page 47 of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Enable Key	Check Enable Key to allow the key to authorize a user.

Field Name	Description
Authorized User Key Name	Authorized User Key Name identifies the name to associate with the key you are adding to the Key Store. Authorized User Key Name can be up to 150 characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the user key you create. Description can be up to 255 characters.
Import from file	Import from file identifies the location and key to import to the key definition. Use the Browse button to locate the file.
Key Data	Key Data displays the contents of the key that you imported.

Known Host Key Store Configuration

The Know Host Key Store dialog shows you the name and description of the selected Known Host Key Store. The keys that are in the store are displayed in a table with a radio button indicating the active key. From this screen, you can change the active key. You can also add, edit, copy or delete a key. Refer to the field definitions in the following table.

For additional information on configuring keys, refer to Chapter 4, *Manage SSH Keys for SFTP Transactions* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Known Host Key Store Name	Known Host Key Store Name identifies the name to assign to the key store you create. Authorized User Key Store Name can be up to 150 characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the user key store you create. Description can be up to 255 characters.

Known Host Key Configuration

Use this screen to update a key currently in the Known Host User Key Store or to create a new key in the store. Refer to the field definitions in the following table.

For additional information on configuring keys, refer to Chapter 4, *Manage SSH Keys for SFTP Transactions* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Enable Key	Check Enable Key to allow the key to be used to authorize a user.
Known Host Key Name	Known Host Key Name identifies the name to associate with the key you are adding to the Key Store. Known Host Key Name can be up to 150 characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the user key you create. Description can be up to 255 characters.

Field Name	Description
Import from file	Import from file identifies the location and key to import to the key definition. Use the Browse button to locate the file.
Key Data	Key data displays the contents of the key that you imported.

Local User Key Store Configuration

The Local User Key Store dialog shows you the name and description of the selected Local User Key Store. The keys that are in the store are displayed in a table with a radio button indicating the active key. From this screen, you can change the active key. You can also add, edit, copy or delete a key. Refer to the field definitions in the following table.

For additional information on configuring keys, refer to Chapter 4, *Manage SSH Keys for SFTP Transactions* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Local User Key Store Name	Local User Key Store Name identifies the name to assign to the key store you create to store keys used to authenticate SSP to the inbound connection. Local User Key Store Name can be up to 150 characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the user key store you create. Description can be up to 255 characters.

Local User Key Configuration

Use this screen to update a key currently in the Local User Key Store or to create a new key in the store. Refer to the field definitions in the following table.

For additional information on configuring keys, refer to Chapter 4, *Manage SSH Keys for SFTP Transactions* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Enable Key	Check Enable Key to allow the key to be used to authorize a user.
Local User Key Name	Local User Key Name identifies the name to assign to the key you create. This field can be up to 150 characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the user key you create. Description can be up to 255 characters.
Password	Password identifies the password to use to access the key. Password can be up to 255 alphanumeric characters and does not allow comma (,), double quotes ("), or single quotes (').

Field Name	Description
Confirm Password	Confirm Password requires that you retype the password value.
Import From File	Import from File identifies the location and key to import to the key definition. Use the Browse button to locate the file.
Key Data	Key Data displays the contents of the key that you imported.
Routing Name	The value that EA will return to map to this key.

Local Host Key Store Configuration

The Local Host Key Store dialog shows you the name and description of the selected Local Host Key Store. The keys that are in the store are displayed in a table with a radio button indicating the active key. From this screen, you can change the active key. You can also add, edit, copy or delete a key. Refer to the field definitions in the following table.

For additional information on configuring keys, refer to Chapter 4, *Manage SSH Keys for SFTP Transactions* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Local Host Key Store Name	Local Host Key Store Name identifies the name to assign to the key store you create. Local Host Key Store Name can be up to 150 characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the key store you create. Description can be up to 255 characters.

Local Host Key Configuration

Use this screen to update a key currently in the Local Host Key Store or to create a new key in the store. Refer to the field definitions in the following table.

For additional information on configuring keys, refer to Chapter 4, *Manage SSH Keys for SFTP Transactions* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Enable Key	Check Enable Key to allow the key to be used to authorize a user.
Local Host Key Name	Local Host Key Name identifies the name to associate with the key you are adding to the Key Store. Local Host Key Name can be up to 150 characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description assigns a description to help you identify the user key you create. Description can be up to 255 characters.

Field Name	Description
Password	Password identifies the password to use to access the key. Password can be up to 255 alphanumeric characters and does not allow comma (,), double quotes ("), or single quotes (').
Confirm Password	Confirm Password requires that you retype the password value.
Import From File	Import From File identifies the location and key to import to the key definition. Use the Browse button to locate the file.
Key Data	Key data displays the contents of the key that you imported.

User Store Configuration

The User Store is a file that contains user accounts. A default user store called defUserStore is available with the product. If desired, you can create additional user stores. Use this screen to add a user store or modify the default user store. Refer to the field definitions in the following table.

For additional information on configuring engine user stores and accounts, refer to *Managing Engine User Stores and User Accounts* on page 65 in the *Sterling Secure Proxy Configuration Guide*

Field Name	Description
User Store Name	User Store Name identifies the name to assign to the user store you create. User Store Name can be up to 150 characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_). A default user store is provided called defUserStore.
Description	Description assigns a description to help you identify the user store you create. Description can be up to 255 characters.
User Lockout Duration	User Lockout Duration identifies how long a user is unable to access SSP, after too many incorrect logon attempts. The default value is 300 seconds.
User Lockout Threshold	User Lockout Threshold identifies how many unsuccessful logon attempts are allowed before a user is locked out.

User Configuration - Basic

The User Configuration allows you to define users who are allowed to access SSP. Use this screen to define the basic requirements for user access to SSP. Refer to the field definitions in the following table.

For additional information on configuring engine user stores and accounts, refer to *Manage Engine User Stores and User Accounts* on page 89 in the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
User Name	Name to assign to the user you configure. User Name can be up to 150 characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).

Field Name	Description
Description	Description assigns a description to help you identify the user you create. Description can be up to 255 characters.
Password	Password that the user must provide to access SSP. It can be up to 28 alphanumeric characters and cannot include commas (,), double quotes ("), or single quotes ('). The length of the password required is defined in the password policy configuration.
Confirm Password	Retype the password value required by the user.
Password Policy ID	Password policy to associate with the user you are configuring. You must configure a password policy before you can associate it with a user. Select a Password Policy ID from the pull-down list.
User Active	Identifies that the user can communicate with SSP. Disable this option to prevent a user from accessing SSP.
First Name	First name of the user. Optional.
Last Name.	Last Name of the user. Optional.
Email Address	Email address of the user. Optional.
Pager	Pager number for the user. Optional.
Manager ID	Manager information for the user.

User Configuration - Advanced

The User Configuration Advanced tab allows you to associate SSH keys with a user definition. Refer to the field definitions in the following table.

For additional information on configuring engine user stores and accounts, refer to *Manage Engine User Stores and User Accounts* on page 89 in the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
SSH Authorized User Key Store	Select an SSH Authorized User Key Store to associate with the user from the drop-down list.
SSH Authorized User Key	Select the SSH Authorized User Keys to associate with the user from the drop-down list.

Advanced Menu Field Definitions

Perimeter Servers Field Definitions

From the Advanced menu, you can configure remote perimeter servers that you want to use with a SSP engine. Identify if the perimeter server is installed in a more secure zone or a less secure zone and then configure the perimeter server.

Less Secure Zone PS Configuration - Basic

Use this screen to configure a remote perimeter server in a less secure zone. Refer to the field definitions in the following table.

For more information on configuring an Perimeter Servers, refer to *Chapter 12, Configure Perimeter Servers to Manage SSP Communications* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Perimeter Server Name	Name to assign to the perimeter server you create. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description to help you identify the perimeter server you create. Description can be up to 255 characters.
Perimeter Server Host	DNS name or TCP/IP address where the DMZ perimeter server is installed.
Perimeter Server Port	Port number that the DMZ perimeter server monitors for connections. This is the port number you specified when installing your perimeter server in the DMZ.

Less Secure Zone PS Configuration - Advanced

Use this screen to edit advanced properties associated with a remote perimeter server installed in a less secure zone. Refer to the field definitions in the following table.

For more information on configuring an Perimeter Servers, refer to *Chapter 12, Configure Perimeter Servers to Manage SSP Communications* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Perimeter Server Outbound Low Water Mark	Lowest outbound connection buffer size. This is the low water mark. The default is 150 KB.
	When SSP sends data to a trading partner faster than the trading partner can receive it, the excess data accumulates inside perimeter services in the outbound connection buffer. When the buffer size reaches the High Outbound Connection value, perimeter services stops sending data through that connection until enough of the excess data has been sent that the outbound connection buffer size drops to the Low Outbound Connection value.
	For example, if you set the High Outbound Connection value to 500 KB and the Low Outbound Connection value to 250 KB, perimeter services will stop sending data when the outbound connection buffer size reaches 500 KB and will resume sending data when the outbound connection buffer size drops to 250 KB.

Field Name	Description
Perimeter Server Outbound High Watermark	Highest outbound connection buffer size. This is the high water mark. The default is 250 KB.
	When SSP sends data to a trading partner faster than the trading partner can receive it, the excess data accumulates inside perimeter services in the outbound connection buffer. When the buffer size reaches the Perimeter Server High Outbound Connection value, perimeter services stops sending data through that connection until enough of the excess data has been sent that the outbound connection buffer size drops to the Perimeter Server Low Outbound Connection value.
	For example, if you set the Perimeter Server High Outbound Connection value to 500 KB and the Perimeter Server Low Outbound Connection value to 250 KB, perimeter services will stop sending data when the outbound connection buffer size reaches 500 KB and will resume sending data when the outbound connection buffer size drops to 250 KB.
Perimeter Server Inbound Low Water Mark	Lowest inbound connection buffer size. This is the low watermark. The default is 150 KB.
	When a trading partner sends data faster than SSP can process it, the excess data accumulates inside perimeter services in the inbound connection buffer. When the buffer size reaches the High Inbound Connection value, perimeter services stops receiving data for that connection until enough of the excess data has been processed that the inbound connection buffer size drops to the Low Inbound Connection value.
	For example, if you set the High Inbound Connection value to 500 KB and the Low Inbound Connection value to 250 KB, perimeter services will stop receiving data when the inbound connection buffer size reaches 500 KB and will resume receiving data when the inbound connection buffer size drops to 250 KB.
Perimeter Server Inbound High Water Mark	Highest inbound connection buffer size. This is the high watermark. The default is 250 KB.
	When a trading partner sends data faster than SSP can process it, the excess data accumulates inside perimeter services in the inbound connection buffer. When the buffer size reaches the Perimeter Server High Inbound Connection value, perimeter services stops receiving data for that connection until enough of the excess data has been processed that the inbound connection buffer size drops to the Perimeter Server Low Inbound Connection value.
	For example, if you set the Perimeter Server High Inbound Connection value to 500 KB and the Perimeter Server Low Inbound Connection value to 250 KB, perimeter services will stop receiving data when the inbound connection buffer size reaches 500 KB and will resume receiving data when the inbound connection buffer size drops to 250 KB.
Proxy Local Interface	Network interface SSP uses to connect to the perimeter server. The default is *, which allows the operating system to make the selection. You can specify any IP address or DNS name of an interface which exists on this machine.

Field Name	Description	
Proxy Local Port	Port number to use for the local end of the socket to the perimeter server. The default is 0, which allows the operating system to select any free port. Valid values are 1–65,535.	
Perform DNS Resolution	Place where DNS resolution occurs. The default is At Local Host.	
	◆ At Local Host	The DNS name is resolved at the local host where SSP is installed.
	◆ At Perimeter Server Host	The DNS name is resolved at the perimeter server host.

More Secure Zone PS Configuration - Basic

Use this screen to configure a remote perimeter server in a more secure zone. Refer to the field definitions in the following table.

For more information on configuring an Perimeter Servers, refer to *Chapter 12, Configure Perimeter Servers to Manage SSP Communications* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Perimeter Server Name	Perimeter Server Name identifies the name to assign to the perimeter server you create. Valid values are 1–150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description for the perimeter server to help identify the perimeter server you create. Description can be up to 255 characters.
Proxy Local Listen Port	Port number that the perimeter server monitors for connections. This is the port number you specified when installing your perimeter server. Valid values are 1–65,535.

More Secure Zone PS Configuration - Advanced

Use this screen to edit the default properties associated with a perimeter server installed in a more secure zone. Refer to the field definitions in the following table.

For more information on configuring an Perimeter Servers, refer to *Chapter 12, Configure Perimeter Servers to Manage SSP Communications* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Perimeter Server Outbound Low Water Mark	Lowest outbound connection buffer size. This is the low water mark. The default is 150 KB.
	When SSP sends data to a trading partner faster than the trading partner can receive it, the excess data accumulates inside perimeter services in the outbound connection buffer. When the buffer size reaches the High Outbound Connection value, perimeter services stops sending data through that connection until enough of the excess data has been sent that the outbound connection buffer size drops to the Low Outbound Connection value.
	For example, if you set the High Outbound Connection value to 500 KB and the Low Outbound Connection value to 250 KB, perimeter services will stop sending data when the outbound connection buffer size reaches 500 KB and will resume sending data when the outbound connection buffer size drops to 250 KB.
Perimeter Server Outbound High Watermark	Highest outbound connection buffer size. This is the high water mark. The default is 250 KB.
	When SSP sends data to a trading partner faster than the trading partner can receive it, the excess data accumulates inside perimeter services in the outbound connection buffer. When the buffer size reaches the Perimeter Server High Outbound Connection value, perimeter services stops sending data through that connection until enough of the excess data has been sent that the outbound connection buffer size drops to the Perimeter Server Low Outbound Connection value.
	For example, if you set the Perimeter Server High Outbound Connection value to 500 KB and the Perimeter Server Low Outbound Connection value to 250 KB, perimeter services will stop sending data when the outbound connection buffer size reaches 500 KB and will resume sending data when the outbound connection buffer size drops to 250 KB.

Field Name	Description
Perimeter Server Inbound Low Water Mark	The lowest inbound connection buffer size. This is the low watermark. The default is 150 KB.
	When a trading partner sends data faster than SSP can process it, the excess data accumulates inside perimeter services in the inbound connection buffer. When the buffer size reaches the High Inbound Connection value, perimeter services stops receiving data for that connection until enough of the excess data has been processed that the inbound connection buffer size drops to the Low Inbound Connection value.
	For example, if you set the High Inbound Connection value to 500 KB and the Low Inbound Connection value to 250 KB, perimeter services will stop receiving data when the inbound connection buffer size reaches 500 KB and will resume receiving data when the inbound connection buffer size drops to 250 KB.
Perimeter Server Inbound High Watermark	The highest inbound connection buffer size. This is the high watermark. The default is 250 KB.
	When a trading partner sends data faster than SSP can process it, the excess data accumulates inside perimeter services in the inbound connection buffer. When the buffer size reaches the Perimeter Server High Inbound Connection value, perimeter services stops receiving data for that connection until enough of the excess data has been processed that the inbound connection buffer size drops to the Perimeter Server Low Inbound Connection value.
	For example, if you set the Perimeter Server High Inbound Connection value to 500 KB and the Perimeter Server Low Inbound Connection value to 250 KB, perimeter services will stop receiving data when the inbound connection buffer size reaches 500 KB and will resume receiving data when the inbound connection buffer size drops to 250 KB.
Proxy Local Interface	The network interface is used by SSP to listen for connections form the perimeter server. The default is *, which means SSP will listen on all available interfaces. You can specify any IP address or DNS name of an interface which exists on this machine.
Perform DNS Resolution	Perform DNS Resolution identifies where the DNS resolution occurs. The default is At Local Host.
	◆ At Local Host—DNS name is resolved at the local host where SSP is installed
	◆ At Perimeter Server Host—DNS name is resolved at the perimeter server host.

Sterling Secure Proxy Configuration Guide

EA Server Configuration Field Definitions

If you plan to use EA to authenticate users or certificates, you must install an EA server and configure the EA server in SSP. Refer to the field definitions in the following table.

For more information on configuring EA, refer to *Chapter 12, Configure Perimeter Servers to Manage SSP Communications* of the *Sterling Secure Proxy Configuration Guide.*

SSP EA Server Configuration - Basic

Use this screen to configure an EA server. Refer to the field definitions in the following table.

For more information on configuring EA, refer to *Chapter 10, Configure SSP for Sterling External Authentication Server (EA)* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
EA Server Name	Name to assign to the EA server definition you create. Valid values are 1–150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description to help you identify the EA server definition you create. Description can be up to 255 characters.
EA Server Address	IP address or host name to use to connect to the EA server. Valid values are 1-200 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), colon (:), and underscore (_).
EA Server Port	Port number to use to connect to the EA server. Valid values include 1-65535.
Outbound Port Range	Range of ports to use to connect to the EA server. Valid values include a list of ports that are allowed with each value separated by a comma such as 1234, 2340, 16570 or a range of ports allowed, such as 16570 -17950.

SSP EA Server Configuration - Security

Use this screen to define secure connection requirements for an EA server definition. Refer to the field definitions in the following table.

For more information on configuring EA, refer to *Chapter 10, Configure SSP for Sterling External Authentication Server (EA)* of the *Sterling Secure Proxy Configuration Guide.*

Field Name	Description
Use Secure Connection	Enable Use Secure Connection to turn on the use of SSL/TLS to provide secure communications with transport protocols and to ensure that data is secured as it is transmitted across a single socket.
Security Setting	Security protocol allowed for connections to the EA server. Options include: ◆ SSL—select this option to require SSL for the connection ◆ TLS—select this option to require TLS for the connection

Field Name	Description
Trust Store	Location where the system and CA certificates are stored. System and CA certificates are used during a secure connection to verify that a certificate received from a server is signed by a trusted source.
CA /Trusted Certificates	The trusted certificate to use to authenticate the certificate presented by EA. You select a CA certificate or trusted root from the list of certificates stored in the trust store you selected in the Trust Store field. When EA presents a certificate to establish a secure connection, the trusted root certificate, located at the SSP server, must match or be the entity who signed the certificate presented by EA during the SSL handshake.
Key Store	Location where the keys and system certificates you want to use are stored.
Key/System Certificate	Certificate presented by the EA server to SSP to authenticate itself during the SSL handshake. Select the Key/System Certificate to use for the node from the list. The list identifies the key or system certificates stored in the key store you selected in the Key Store field.
Cipher Suites	List of ciphers that can be enabled to encrypt data transmitted during a secure SSL or TLS connection between SSP and an EA server. Enable at least one cipher.

SSP EA Server Configuration - Advanced

Use the advanced tab for an EA server definition to allow failover support for an EA server. If failover support is configured and a connection to the primary EA server cannot be configured, SSP connects to the first alternate server. If a connection to the first alternate EA server cannot be made, SSP connects to the second alternate server. Refer to the field definitions in the following table.

For more information on configuring EA, refer to *Chapter 10, Configure SSP for Sterling External Authentication Server (EA)* of the *Sterling Secure Proxy Configuration Guide.*

> **Note:** If a connection to the primary EA server is established but the connection is closed because a secure handshake could not be performed, SSP does not attempt a failover connection. This failure is not a connection failure.

Field Name	Description
Alternate EA Server	EA server name to use to connect to an alternate EA server, if a connection to the primary EA server cannot be made. Up to three alternate EA servers can be defined for each EA server. The servers are used in sequence 1, 2, 3. You must first configure each EA server. Then you can identify alternate EA servers to use if an EA server is not available by selecting a server definition from the list.

Password Policy Field Definitions

The Password Policy tab is used to define password policies, a set of security decisions that you make and apply to different user accounts according to security policies in your company. After you

create a password policy, you can associate it with a user definition. Refer to the field definitions in the following table.

For more information on configuring an Password Policies, refer to *Chapter 5, Manage User Accounts and Passwords* of the *Sterling Secure Proxy Configuration Guide.*

Field	Description
Password Policy Name	The name that displays in the user interface when a reference is made to the password policy.
Description	Description to help you identify the password policy you create. Description can be up to 255 characters.
Days Valid	Number of days that a user password is valid. The user is prompted to change the password when this time period expires. The default is 0, which means the password never expires. You can change this number to any number you want. There is no maximum value. The expiration count down starts the first time a user logs in to SSP after a password is assigned to the user account.
Minimum Length	Minimum length that the password must be. This field is required. Valid values are any numerals. The default value is 6. If no policy is applied, SSP enforces a minimum length of 6.
Maximum Length	How long the password can be.This field is required. Valid values are any numerals. This number must be set to at least the same number as the minimum length. The default value is 28
Kept in History	How many passwords to keep in the PWD_HISTORY table in the file for a user. Values store in history cannot be used when defining a new password value. After this number of passwords is exceeded, the oldest password is removed from the table and can be re-used by the user. The default value is 5.
Must contain special characters	The password must contain at least one special character, such as numeral, capital letter, !, @, #, $, %, ^, &, or *.

System Menu Field Definitions

Use the System menu to configure certificates for CM connection to the web server, configure CM Users, modify system settings, or unlock objects.

CM Trusted Certificate Store Configuration

The CM Trusted Certificate Stores dialog shows you the name and description of the CM Trusted Certificate Store. The certificates that are in the store are displayed in a table. These certificates are used to authenticate an SSL or TLS secure communications session with the web server and the engine. From this screen, you can view the active certificate. Refer to the field definitions in the following table.

For additional information on configuring CM certificates, refer to Chapter 15, *Manage Certificates Between SSP Components* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
CM Trusted Certificate Store Name	Name of the certificate store you are viewing.
Description	Description to help you identify the certificate store you create. Description can be up to 255 characters.

CM Trusted Certificate Configuration

Use this screen to view a certificate currently in the trust store. Refer to the field definitions in the following table.

For additional information on configuring CM certificates, refer to Chapter 15, *Manage Certificates Between SSP Components* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Enable Certificate	This field is disabled.
CM Trusted Certificate Name	Name of the trusted certificate you are viewing.
Description	Description to help you identify the certificate you are viewing.
Import from file	This field is disabled.
Certificate Data	Contents of the certificate.

CM System Certificate Store Configuration

The CM System Certificate Stores dialog shows you the name and description of the selected System Certificate Store. The private keys that are in the store are displayed in a table. From this screen, you can change the active key. You can also add, edit, copy or delete a key. Refer to the field definitions in the following table.

For additional information on configuring CM certificates, refer to Chapter 15, *Manage Certificates Between SSP Components* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
CM System Certificate Store Name	Name of the certificate store you are viewing.
Description	Description to help you identify the certificate store you are viewing.

CM System Certificate Configuration

Use this screen to view a key currently in the trust store. Refer to the field definitions in the following table.

For additional information on configuring CM certificates, refer to Chapter 15, *Manage Certificates Between SSP Components* of the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Enable Certificate	This field is disabled.
CM System Certificate Name	Name associated with the system certificate you are viewing in the certificate store.
Description	Description to identify the certificate you are viewing.
Password	This field is disabled.
Confirm Password	This field is disabled.
Import From File	This field is disabled.
Certificate Data	Contents of the certificate that you are viewing.

CM User Configuration

The CM User Configuration tab is used to create accounts for users who will access CM. You can assign roles to users based on how they will use CM. The operator role has read-only access to CM. The administrator role has full access to all of the configuration options available in CM.

Field Name	Description
User Name	User you define to allow access to CM. User Name can be up to 150 characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description to help you identify the CM user you create. Up to 255 characters.
Password	Password required by the user to access CM. Up to 255 alphanumeric characters and does not allow comma (,), double quotes ("), or single quotes (').
Confirm Password	Retype the password value.
User role	Role allowed by the user you create. The Operator role has read-only access to CM; whereas, the Admin role has full access to create and edit all of the configuration options available in CM. Admin is the default user role value
Policy ID	Password policy to associate with the user you configure. Configure a password policy before you can associate it with a user. Select a Password Policy ID from the pull-down list.
Requires change	Requires that the user change the default password after the initial log in. A prompts is displayed to the user.

System Settings - Listeners

System Settings - Listeners identifies the IP address and ports that CM uses to listen for secure connections. For additional information to configure listen port and address on CM, refer to the Sterling *Secure Proxy Operations Guide*.

Field Name	Description
IPAddress	IP address at CM to use to listen for secure connections.
Port	Port at CM to use to listen for secure connections.

System Settings - Security

System Settings - Security identifies the security information used during a secure connection from CM to the engine. Setting up the internal certificate, using this screen, does not completely configure internal certificates. We recommend that you use the scripts provided to set up the internal certificates. Refer to Chapter 15, *Manage Certificates Between SSP Components* in the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Protocol	Identifies that TLS is used to secure the connection. TLS is the only protocol that can be used for the connection between CM and the web server. This is a read-only field.
Key store file	Location where the keys and system certificates you want to use are stored.
Key/System Certificate	Certificate presented by the web server to CM to authenticate itself during the TLS handshake. Select the Key/System Certificate to use from the list. The list identifies the key or system certificates stored in the key store you selected in the Key Store field.
Cipher Suites	List of ciphers that can be enabled to encrypt data transmitted during a secure connection between CM and the web server. Enable at least one cipher.

System Settings - Globals

System Settings - Global identifies the system settings for sessions between CM and the web server. Use this panel to modify the default settings. For more information on configuring global settings, refer to the *Sterling Secure Proxy Operations Guide*.

Field Name	Description
Logging level	Level of logging to write to the log file for CM. Logging options include: ◆ NONE turns logging off. ◆ ERROR writes only error messages to the log. ◆ WARN writes error and warning messages to the log. ◆ INFO writes error and informational messages to the log. INFO is the default value. ◆ DEBUG writes all messages to the log including debugging messages. Use this logging level when instructed to turn it on by Sterling Commerce Support.
Listen backlog	Number of client connections allowed in a queue before connections are refused. Valid values range from 0 to 2147483.
Accept timeout	Number of seconds that the accepter listens before a timeout occurs. The default is 30. Valid values range from 0 to 2147483.
SSL handshake timeout	How many seconds are allowed for an SSL handshake. If the SSL handshake does not occur during this time, the session is terminated. This parameter ensures that a connecting client authenticates within a fixed amount of time. The default is 30 seconds. Valid values range from 0 to 2147483.
Connect timeout	How many seconds are allowed for an outbound connection from the server before a timeout occurs, if the connection is not accepted. Valid values range from 0 to 2147483.
Read timeout	How many seconds elapse before a read operation times out, if unsuccessful. Valid values range from 0 to 2147483.

System Settings - Lock Manager

Lock Manager allows you to unlock CM components. Refer to the Sterling *Secure Proxy Operations Guide* for instructions.

Field Name	Description
show	A drop-down list of all components that you can select and unlock. Available options include:
	◆ All Objects—select All Objects to view all objects that are locked.
	◆ Engines—select Engines to view all engines that are locked.
	◆ Adapters—select Adapters to view locked adapters. You can filter this list and select Connect:Direct, HTTP, FTP, or SFTP to view only locked adapters of a specific protocol
	◆ Netmaps—select Netmaps to view locked netmap. You can filter this list and select Connect:Direct, HTTP, FTP, or SFTP to view only locked netmaps of a specific protocol
	◆ Policies—select Policies to view locked policies. You can filter this list and select Connect:Direct, HTTP, FTP, or SFTP to view only locked policies of a specific protocol
	◆ EA Servers—select EA Servers to view all EA servers that are locked.
	◆ Perimeter Servers—select Perimeter Servers to view all perimeter servers that are locked.
	◆ Password Policies—select Password Policies to view all password policies that are locked.
	◆ Step Injections—select Step Injections to view all step injection objects that are locked.
	◆ Key Stores—select Key Stores to view all key stores that are locked.
	◆ User Stores—select User Stores to view all user stores that are locked.
	◆ CM Users—select CM Users to view all CM users whose account is locked.
Name	Name of the object that is locked.
Object	Type of object that is locked.
Protocol	Protocol of the locked object.
Locked By	The user ID that locked the object.
Lock Time	When the object was locked.
Expiration	When the lock expires.

PeSIT Field Definitions

This section provides field definitions for PeSIT protocol dialogs in SSP. Following are definitions of the fields for each SSP screen.

PeSIT Adapter Configuration - Basic

Use this screen to specify system-level communications information for PeSIT/Connect:Express connections to and from SSP. Before you can click the Advanced or Properties tabs, you must specify Adapter Name and Listen Port and select a netmap to associate with the adapter.

Refer to the field definitions in the following table. For additional information, refer to Chapter 11, *PeSIT Proxy Configuration* in the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Name	Name to assign to the adapter you create. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description to help you identify the adapter you create. Description can be up to 255 characters.
Type	Protocol being used: PeSIT.
Listen Port	Port number to use to listen for inbound connections. Valid values include 1-65535.
Netmap	Netmap to associate with adapter.

Field Name	Description
Routing Type	Select the Routing Type to identify how inbound connections are routed to the server in the trusted zone. Routing options include: ◆ Standard (Default)—select Standard to direct connections to the outbound node specified in the SNODE Netmap Entry field. ◆ Certificate-based—select this option to use the certificate presented by the inbound PNODE to determine which outbound SNODE to connect to. Certificate-based routing uses External Authentication and requires that you configure External Authentication server. ◆ PNODE-specified—select this option to route outbound connections based on information provided by the inbound PNODE. ◆ PNODE-specified, then Standard—select this option to route outbound connections based first on information provided by the inbound PNODE. If the routing information presented by the PNODE is not configured in the netmap, the connection is routed to the outbound node specified in the SNODE Netmap Entry field.
SNODE Netmap Entry	Name of the Connect:Express server where the node connections are routed, after connecting to SSP. Select this value from a pull-down list. **Note:** Only nodes defined with an IP address are outbound nodes, and are shown in the list.
Engine	SSP server in the DMZ where traffic is first routed before being sent to the outbound secure Connect:Express server. Select an engine from the list. You must define an engine before you can create an adapter.
Startup Mode	Identifies how the adapter is started. **auto** (Default) starts the adapter as soon as the configuration is pushed to the engine. **manual** requires that the adapter be manually started.

PeSIT Adapter Configuration - Advanced

Use this screen to specify additional communications information, and to specify the perimeter servers to use for this adapter.

Refer to the field definitions in the following table. For additional information, refer to Chapter 11, *PeSIT Proxy Configuration* in the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Logging Level	Level of logging to write to the log file for the adapter. Logging options include: ◆ ERROR writes only error messages to the log. ERROR is the default value. ◆ WARN writes error and warning messages to the log. ◆ INFO writes error and informational messages to the log. ◆ DEBUG writes all messages to the log including debugging messages. Use this logging level when instructed to turn it on by Sterling Commerce Support.
Maximum Sessions	Maximum number of concurrent sessions that the adapter allows. The default is 20.
Session Timeout	Amount of time allowed, in minutes, between transmissions of TCP packets before a session is terminated. The default is 3 minutes.
Http Ping Response	Response sent when an HTTP GET is received on the listen port. Provide this value to send a health check response to a third-party IP load balancer, such as Big IP. To test the response, ping the URL and port of the engine. For example, if you configure an adapter on port 13640 and you want to get an HTTP 1.0 response, send a ping to http://ProxyServerURL:13640/. The value you supplied in the Http Ping Response field is returned. If you provide a value in this field, the value is displayed in a browser window. You can provide HTML syntax and text values.
Outbound Port Range	Range of ports to use for the adapter. Valid values include a list of ports that are allowed with each value separated by a comma such as 1234, 2340, 16570, or a range of ports allowed, such as 16570-17950.
External Authentication Server	Sterling External Authentication Server to use. Select the server from the pull-down list. You must define an Sterling External Authentication Server before you can select the server from the list.
Perimeter Server Mapping - Inbound Perimeter Server	Select the perimeter server to use for the inbound connection. To use a remote perimeter server, you must define the remote perimeter server before you can associate it with an inbound connection.
Perimeter Server Mapping - Outbound Perimeter Server	Select the perimeter server to use for the outbound connection. To use a remote perimeter server, you must define the it before you can associate it with an outbound connection.
Perimeter Server Mapping - External Authentication Perimeter Server	Select the perimeter server to use for the EA connection. To use a remote perimeter server, you must define it before you can associate it with an Sterling External Authentication Server connection.
Inbound and outbound sessions can have different levels of encryption	Enable this field to allow the connections from the PNODE to SSP and from SSP to the SNODE to use different encryption methods. Set this option to enable a secure connection on the inbound session but use a non-secure connection on the outbound session or to use different protocols on the inbound and the outbound connection.

PeSIT Adapter Definition - Property

Use this screen to edit properties associated with how the PeSIT protocol is implemented. The key is not displayed. To change a default key value, type the key and assign a value.

The *SsslHeaderBytesUsed* key is used for specific SSL links. Change or add the key when instructed to do so by Sterling support. Valid values are false or true.

PeSIT Netmap Definition

Use this screen to define secure connection requirements for the node.

Refer to the field definitions in the following table. For additional information, refer to Chapter 11, *PeSIT Proxy Configuration* in the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Netmap Name	Name to assign to the netmap you create. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description to help you identify the netmap you create. Description can be up to 255 characters.
Type	Protocol being used: HTTP, FTP, SFTP, PeSIT, or Connect:Direct.
Filter	Filter allows you to view a subset of available nodes. Use the wildcard characters, * and ?, to identify the nodes to display. Filters are case-sensitive. For example, the filter n* will display node1 but will not display Node1.
Move Up Move Down	Use Move Up and move Down to arrange the node list from more specific to less Specific IP address if you are mixing IP addresses and Patterns

PeSIT Netmap Node Definition - Basic

Use this screen to define the minimum PeSIT connection parameters for a PeSIT/Connect:Express node. You must define a node name, address, and port before you can save the node definition.

Refer to the field definitions in the following table. For additional information, refer to Chapter 11, *PeSIT Proxy Configuration* in the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Node Name	Name of the Connect:Express/PeSIT node you are configuring in PeSIT proxy. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).

Field Name	Description
Routing Name	A value used to select this SNODE as the outbound node during certificate-based routing. It must match the routing name returned by EA. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_). Set this field only if you are configuring certificate-based routing in EA.
Description	Description to help you identify the node you create. It can be up to 255 characters.
PeSIT Server Address	IP address or host name of the Connect:Express/PeSIT node. Valid values are 1-200 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), colon (:), and underscore (_). You can also define one of the following types of patterns: ◆ Wildcard validates incoming DNS names. If a wildcard pattern is provided, SSP performs a reverse lookup on the incoming IP address and the DNS name is compared to the wildcard patterns. Wildcard characters allowed are ? and *. For example, *.a.com allows a connection from b.a.com but not from b.b.com. ◆ IP/Subnet validates incoming IP addresses. Use the format IP-address/num-bits where IP-address identifies an IP address template and num-bits identifies the number of leading (highest-order) bits in the template that are significant. An IP match is performed by comparing the leading (highest-order) num-bits of the incoming IP address against num-bits of the template. For example, 10.20.0.0/16 searches for a match to the first 16 bits. All IP addresses beginning with 10.20.* are allowed. 10.0.0.0/8 searches for a match to the first 8 bits. All addresses beginning with 10.* are allowed. 0.0.0.0/0 searches for a match to the first zero bits. All IP addresses are allowed. **Note:** When you use a pattern, you are configuring an inbound only node. The Routing Name and PeSIT Server Port are used for outbound only, and these fields are disabled.
PeSIT Server Port	Port number of the Connect:Express/PeSIT node. Valid values are 1 - 65535. If the server address is a pattern, this field can be left blank.
Policy	Policy you want to associate with the node you are creating. If a policy with the security attributes required has not been created, click (+).

PeSIT Netmap Node Definition - Security

Use this screen to define secure connection requirements for the node.

Refer to the field definitions in the following table. For additional information, refer to Chapter 11, *PeSIT Proxy Configuration* in the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Use SSL	Enable SSL to turn on the use of SSL/TLS to provide secure communications with transport protocols and to ensure that data is secured as it is transmitted across a single socket.

Field Name	Description
Verify Common Name	Enable Verify Common Name if your security environment requires that the common name in the certificate presented be verified. If you enable Verify Common Name, you must provide Certificate Common Name.
Certificate Common Name	Common name value to validate. If the common name in the certificate does not match the value defined in this field, the session fails.
Enable Client Authentication	Enable Client Authentication on the inbound node connection to require that the inbound node authenticate the certificate presented by the SSP server.
Security Setting	Security protocol allowed for connections to this node. Options include: ◆ SSL V3 or TLS (Default) - select this option to require SSL V3 or TLS for the connection. ◆ TLS - select this option to require TLS for the connection. ◆ SSL V3 - select this option to require SSL V3 for the connection.
Trust Store	Database where the system and CA certificates are stored. System and CA certificates are used during a secure connection to verify that a certificate received from a server is signed by a trusted source.
CA Certificates/Trusted Root	Trusted certificate to use to authenticate the certificate presented by the client. You select a CA certificate or trusted root from the list of certificates stored in the trust store you selected in the Trust Store field. When a client presents a certificate to establish a secure connection, the trusted root certificate, located at the server, must match or be the entity who signed the certificate presented by the client during the SSL handshake.
Key Store	Database where the keys and system certificates you want to use are stored.
Key/System Certificate	Certificate presented by SSP to the node to authenticate itself during the SSL handshake, or the certificate presented by the Connect:Express server to authenticate itself to the SSP server. Select the Key/System Certificate to use for the node from the list. The list contains the key or system certificates stored in the key store you selected in the Key Store field.
Available Cipher Suites	List of ciphers that can be enabled to encrypt data transmitted during a secure SSL or TLS connection between SSP and a Connect:Express node. Enable at least one cipher. To enable a cipher, highlight it and click Add. To enable multiple ciphers, highlight the ciphers to enable and click Add.
Selected Cipher Suites	Ciphers you enabled to encrypt data during a secure SSL or TLS connection. Ciphers are negotiated based on their location in the Selected Ciphers list. To reorder a cipher in the list, highlight it and click Up or Down.

PeSIT Netmap Node Definition - Advanced

Use this screen to change the logging level for a PeSIT node definition as well as to identify a destination service name to use in an EA transaction and to configure nodes to use for failover support.

Refer to the field definitions in the following table. For additional information, refer to Chapter 11, *PeSIT Proxy Configuration* in the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Logging level	The level of logging to write to the node log file. Logging options include: ◆ NONE turns logging off. NONE is the default value. ◆ ERROR writes only error messages to the log. ◆ WARN writes error and warning messages to the log. ◆ INFO writes error and informational messages to the log. ◆ DEBUG writes all messages to the log including debugging messages. Use this logging level when instructed to turn it on by Sterling Commerce Support.
Destination Service Name	Name of the service that is passed to Sterling External Authentication Server (EA) for use in authenticating services. If no value is provided, the SNODE name is used as the service name.
Logon ID	ID to use to connect to the outbound server if the policy is requires that the credentials from the netmap be used. Valid values are 1-8 alphanumeric characters and certain special characters. The following special characters are not allowed: ! @ # % ^ * ( ) + ? , < > { } [ ] |; " '
Password	Password to use to connect to the outbound server if the policy requires that credentials from the netmap be used. Valid values are 1-8 alphanumeric characters and certain special characters. The following special characters are not allowed: , " '
Alternate Destinations - Node	Node name or IP address and port to use to connect to an alternate Connect:Express/PeSIT outbound node if a connection to the primary node cannot be made. Up to three alternate destination nodes can be defined for each outbound node. To use different security and Sterling External Authentication Server definitions for the alternate destination node, first configure an outbound node definition for the alternate node in the netmap. Then, open the primary outbound node definition and select the alternate node name from the drop-down list in the Alternate Destinations - Node 1 field. To use the security and Sterling External Authentication Server definition defined in the primary outbound node for an alternate destination node, you do not have to define the alternate node in the netmap. Select IP Address/Port from the drop-down list and then provide a value in the IP Address and Port fields. Define up to three alternate node names. **Note:** Only nodes defined with an IP address are outbound nodes and shown in the list.
Alternate Destinations - IP Address	IP address to use to connect to an alternate destination node if a connection to the primary node cannot be made. Up to three alternate destination nodes can be selected. Valid values are 1-200 alphanumeric characters and special characters: underscore (_), dash (-), colon (:), and period (.). If you provide an IP address and port as an alternate destination and a connection to the alternate node is attempted, the security and EA definition from the primary node is used for the connection.

Field Name	Description
Alternate Destinations - Port	Port to use to connect to an alternate destination node if a connection to the primary node cannot be made. Up to three alternate destination nodes can be selected. Valid values are 1-65535. If you provide an IP address and port as an alternate destination and a connection to the alternate node is attempted, the security and EA definition from the primary outbound node is used for the connection.

PeSIT Policy Configuration - Basic

Use this screen to define how you impose controls to authenticate an inbound node. A trading partner trying to access your Connect:Express server will need a stronger policy than a Connect:Express server trying to access a PeSIT node.

Refer to the field definitions in the following table. For additional information, refer to Chapter 11, *PeSIT Proxy Configuration* in the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Policy Name	Name to assign to the policy you create. Valid values are 1-150 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
Description	Description to help you identify the policy you create. Description can be up to 255 characters.
Type	Protocol being used: PeSIT.
Protocol Error Action	Action to perform if SSP detects protocol violations during a communications session. Valid values are: ◆ NONE (Default)—select this option to disable checking of protocol errors. ◆ IGNORE—select this option to ignore protocol errors. ◆ WARN—select this option if you want SSP to write an error message to the log but continue the session when protocol errors are detected. ◆ ABORT—select this option to terminate a communications session when protocol errors are detected.

PeSIT Policy Configuration - Advanced

Use this tab to specify the type of PNODE authentication for inbound access requests. For Certificate Authentication and LogonID Authentication through External Authentication, you must install and configure EA.

Refer to the field definitions in the following table. For additional information, refer to Chapter 11, *PeSIT Proxy Configuration* in the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Certificate Authentication - External Authentication Certificate Validation	Turn on External Authentication Certificate Validation to validate information presented in certificates received from trading partners using EA.
Certificate Authentication - External Authentication Profile	Name of the Certificate Validation Definition you defined in the EA. You must enable certificate validation before you can provide a profile. Valid values are 1-255 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
LogonID Authentication - Through External Authentication	Turn on this option to send an incoming PNODE LogonID and password to Sterling External Authentication Server for validation. Sterling External Authentication Server validates the PNODE LogonID and password using one of the following methods: ◆ An LDAP bind to authenticate a user and password ◆ An LDAP search, based on the PNODE LogonID, destination service name, or certificate subject and issues attributes ◆ Fully-specified DN (distinguished name) or a DN returned from a certificate validation search
LogonID Authentication - External Authentication Profile	If you enabled LogonID authentication through EA, identify the certificate authentication profile you defined in EA in this field. Valid values are 1-255 alphanumeric characters with no spaces. Special characters allowed are period (.), dash (-), and underscore (_).
LogonID Authentication - Through Local User Store	Validates the PNODE LogonID and password of the inbound node using information defined in the user store. You must add the PNODE LogonID to the user store in order to successfully use this method.

Field Name	Description
LogonID Mapping - Internal logon ID	Enable this option in the policy to determine what PNODE LogonID and password is used to connect to the Connect:Express server in the secure environment. For the PNODE LogonID and password to successfully access the Connect:Express server, a PNODE definition must be defined at the server. PNODE mapping options include: ◆ **Pass-through for PNODE** (Default)—uses the LogonID and password supplied by the PNODE to connect to the Connect:Express server in the secure zone. To successfully connect to the Connect:Express server, the LogonID and password must be defined at the server. ◆ **Replace LogonID with LogonID mapped in External Authentication**—maps the LogonID provided by the inbound PNODE to a value defined in External Authentication and uses the EA-supplied value to connect to the SNODE. ◆ **Replace LogonID with Netmap LogonID**—maps the LogonID provided by the inbound PNODE to a value defined in the netmap and uses the netmap value to connect to the SNODE.

PeSIT Policy Configuration - Transfer Direction

Use this tab to block a transfer direction for this node. Refer to the field definitions in the following table. For additional information, refer to Chapter 11, *PeSIT Proxy Configuration* in the *Sterling Secure Proxy Configuration Guide*.

Field Name	Description
Receive a file allowed (SELECT)	Allows the PNODE to request reception of a file. Disable this option to block PeSIT SELECT commands from the PNODE.
Send a file allowed (CREATE)	Allows the PNODE to request transmission of a file. Disable this option to block PeSIT CREATE commands from the PNODE.

Index

C

D

I

S

T

U

Sterling Secure Proxy Configuration Guide