Tectia® Server 6.6 for IBM z/OS

Product Description

10 August 2021

Tectia® Server 6.6 for IBM z/OS: Product Description

10 August 2021

Copyright © 1995–2020 SSH Communications Security Corporation

This software and documentation are protected by international copyright laws and treaties. All rights reserved.

ssh® and Tectia® are registered trademarks of SSH Communications Security Corporation in the United States and in certain other jurisdictions.

SSH and Tectia logos and names of products and services are trademarks of SSH Communications Security Corporation. Logos and names of products may be registered in certain jurisdictions.

All other names and marks are property of their respective owners.

No part of this publication may be reproduced, published, stored in an electronic database, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, for any purpose, without the prior written permission of SSH Communications Security Corporation.

THERE IS NO WARRANTY OF ANY KIND FOR THE ACCURACY, RELIABILITY OR USEFULNESS OF THIS INFORMATION EXCEPT AS REQUIRED BY APPLICABLE LAW OR EXPRESSLY AGREED IN WRITING.

For Open Source Software acknowledgements, see appendix *Open Source Software License Acknowledgements* in the *User Manual*.

SSH Communications Security Corporation
Kornetintie 3, FI-00380 Helsinki, Finland

Table of Contents

Chapter 1 Introduction

Tectia offers software tools to secure end-to-end communications within corporate networks. Tectia client/server solution allows secure network services over an unsecured network, such as the Internet.

Tectia products can be deployed cost-effectively to large corporate networks, because their installation and maintenance can be managed centrally.

The award-winning Secure Shell or SSH technology provides secure encrypted and authenticated communications between two non-trusted hosts. Users can establish secure connections to remote hosts, execute commands on the remote hosts securely, copy remote files securely, and forward X11 sessions (on Unix). Arbitrary TCP/IP ports can also be forwarded (tunneled) over a secure channel, enabling secure application connections, for example, to an e-mail service.

Tectia products are based on Secure Shell (SSH or SecSh) technology originally developed by the founders of SSH Communications Security. The Internet Engineering Task Force (IETF) has standardized the Secure Shell protocol, see RFC 4251 at http://www.ietf.org/rfc/rfc4251.txt.

1.1 Tectia Solution Components

The Tectia client/server solution utilizes client-server architecture. By default, the server listens to TCP port 22, which has been officially assigned for Secure Shell, and clients initiate connections to this port.

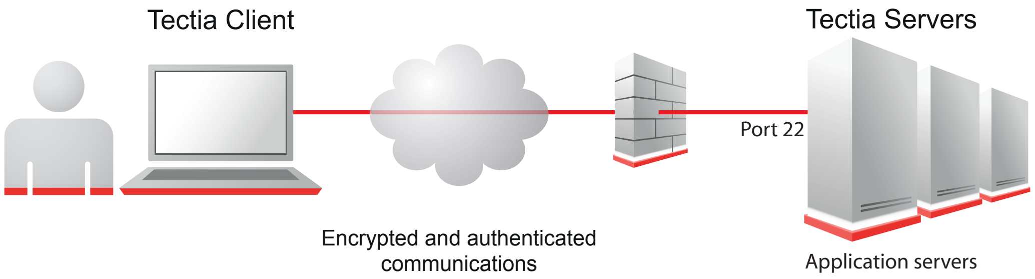


Figure 1.1. The basic idea of Tectia Client and Server

The Tectia products work ideally together, but they can also be used with other Secure Shell-based clients or servers.

1.1.1 Tectia Client

Tectia Client is a workstation product providing the basic Secure Shell client features and tools. Tectia Client takes care of securing remote connections and transfer of files. Users and system administrators need Tectia Client in order to access remote hosts running Tectia Server or another standard Secure Shell server. Tectia Client provides interactive file transfer and terminal client functionalities.

Tectia Client also includes advanced command-line tools for system administrators to set up secure automated file transfers, and tools for outgoing and incoming application tunneling, such as X11 forwarding.

Connection Broker

The Connection Broker is an integrated component of Tectia Server for IBM z/OS client tools. The Connection Broker handles all cryptographic operations and authentication-related tasks on the client side.

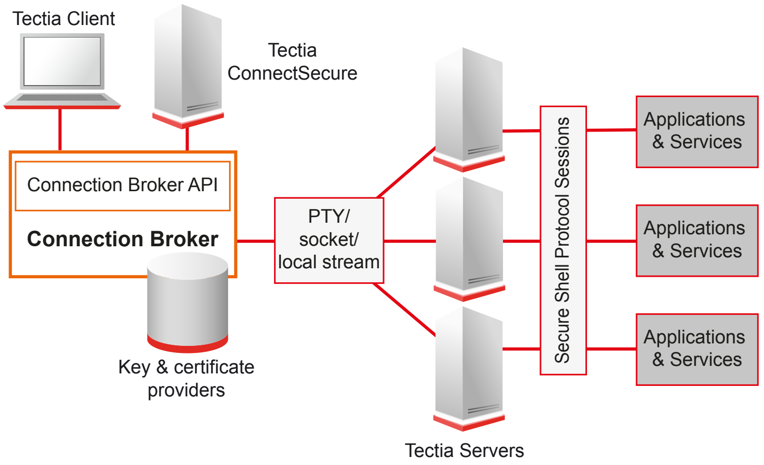


Figure 1.2. Connection Broker architecture

1.1.2 Tectia Server

Tectia Server provides the Secure Shell server features and tools. It enables secure file transfers, secure application connectivity, and secure remote administration services over unsecured networks.

Tectia Server is a robust, flexible, and field-tested server implementation of the Secure Shell protocol. Its technology has been the choice of numerous large corporations, banks, financial organizations, and governments around the world.

Tectia Server provides strong user authentication, traffic encryption/decryption, traffic integrity checking, and out-of-the-box interfaces to integrate leading third-party authentication or authorization systems (such as RSA SecurID and PAM). There is also an easy-to-use graphical user interface for configuring Tectia Server.

The Tectia Server product is available in three versions designed for different platforms:

Tectia Server

> Tectia Server is available for Unix and Windows platforms, such as Oracle Solaris, IBM AIX, HP-UX, SUSE and Red Hat Linux, and Microsoft Windows. On Unix platforms, the server license includes also Tectia Client that can be installed on the same host.
>
> Tectia Server offers all Secure Shell server functionality, including secure terminal, SFTP, and tunneling. When Tectia ConnectSecure is acting as the counterpart, Tectia Server supports also the enhanced file transfer (EFT) features, such as streaming for fast transfer of large files.

Tectia Server for Linux on IBM System z

> Tectia Server for Linux on IBM System z has been designed for Linux running on IBM System z platforms. It provides the same services as Tectia Server on Unix plus it supports hardware acceleration of encryption operations. Tectia Server for Linux on IBM System z contains also the client-side components that provide the basic Secure Shell client features and tools.

Tectia Server for IBM z/OS

> Tectia Server for IBM z/OS has been designed for z/OS platforms running on IBM mainframes. It provides the same services as Tectia Server on Unix and contains also client tools that support FTP-SFTP conversion, transparent FTP tunneling, and enhanced file transfer (EFT) features.

1.2 Multi-Platform Support

Tectia offers extensive platform support for its products. Tectia client/server solution can be deployed into a heterogeneous environment where most commonly Microsoft Windows and different flavors of Unix and Linux operating systems are used. Tectia client/server solution can also connect to Tectia Server for IBM z/OS for transferring data to and from mainframe computers.

1.3 Customer Support Services

SSH Communications Security offers Customer Support Services, the most important of which are consulting, product knowledge and training, and software upgrades. The support engineers at our global support centers are backed by the developers of the SSH information security solutions. Read more about our Customer Support Services at http://www.ssh.com/products/support/.

Technical Support is available according to your selected support plan - whether your business critical systems require support for 24 hours a day, 7 days a week, or less urgent response times. All offered support levels have defined services level agreements (SLAs) and include software maintenance. Read more about SSH Support Levels at http://www.ssh.com/products/support/.

Chapter 2 Key Applications

The key applications of Tectia client/server solution are secure file transfer, secure system administration, and secure application connectivity.

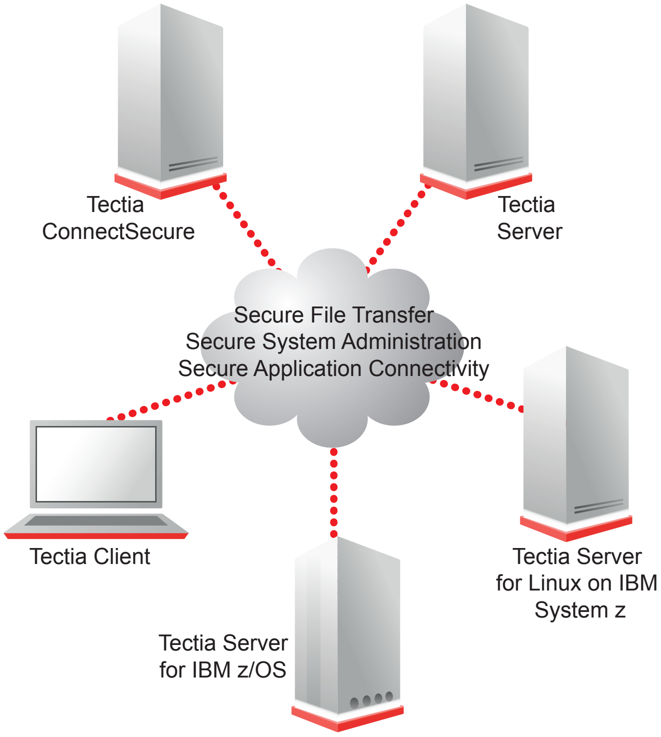

Tectia ConnectSecure
Tectia Server
Secure File Transfer
Secure System Administration
Secure Application Connectivity
Tectia Client
Tectia Server for Linux on IBM System z
Tectia Server for IBM z/OS

Figure 2.1. The key applications of Tectia products

2.1 Secure File Transfer - FTP Replacement

The Tectia client/server solution provides several methods for getting rid of the risks of plaintext File Transfer Protocol (FTP). Tectia Server, Client and ConnectSecure all include a secure alternative, Secure File Transfer Protocol (SFTP) that can be used to replace existing FTP clients and servers. If replacing all FTP clients and servers in an environment is unfeasible, Tectia ConnectSecure and Tectia Server for IBM z/OS offer also transparent FTP tunneling to encrypt the FTP connection, and FTP-SFTP conversion to convert unsecured FTP traffic to use the secure SFTP protocol, instead.

The Tectia client/server solution offers three methods for FTP replacement as illustrated in Figure 2.2:

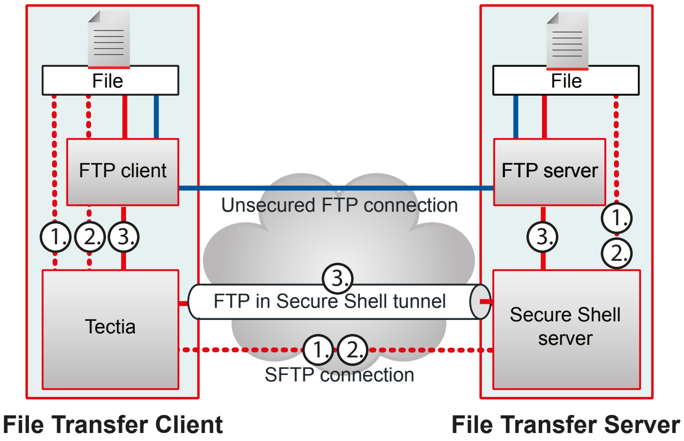


Figure 2.2. Options for replacing unsecured FTP file transfers

An **unsecured FTP connection** is shown in blue. If this is used, user IDs, passwords, and the actual transferred data are sent in plaintext, which makes them vulnerable to eavesdropping and unauthorized modifications.

Tectia products use the following methods to make file transfers secure:

1. **Native SFTP**

   The secure file transfer protocol (SFTP) transfers the files and the related control data encrypted between the client and server. SFTP can be activated by using the `sftpg3` and `scpg3` tools, or the Tectia Secure File Transfer GUI (on Windows) instead of the unsecured `ftp` tools.

   Tectia Client or ConnectSecure provides the SFTP functionality and connects to any Secure Shell SFTP server. Both the original FTP client and FTP server can be eliminated.

2. **FTP-SFTP conversion**

   Connections from the original FTP client are transparently captured by Tectia Server for IBM z/OS, converted to SFTP, and directed to a Secure Shell SFTP server. No changes to the original FTP client application are needed, and it can remain being used as before. The original FTP server, however, is eliminated.

   This feature is available with Tectia ConnectSecure and Tectia Server for IBM z/OS (client tools) on all supported platforms and requires a Secure Shell server as the counterpart.

3. **Transparent FTP tunneling**

Transparent FTP tunneling creates a secure tunnel between an FTP client and an FTP server. All material is sent in encrypted format and so secured from eavesdropping. This feature is available with Tectia ConnectSecure and Tectia Server for IBM z/OS (client tools).

The Tectia client/server solution supports also non-transparent FTP tunneling on both Tectia Client and ConnectSecure. Non-transparent FTP tunneling can be implemented as SOCKS tunnels defined in the Tectia connection profiles, or as automatic tunnels defined in the Connection Broker configuration.

The following table lists the benefits offered by each of the FTP replacement methods.

Table 2.1. Differences between FTP and secure file transfer methods

Feature	FTP	SFTP	FTP-SFTP Conversion	Transparent FTP Tunneling
Automated scripts can be used	x	x	x	x
FTP application used unmodified	x		x	x
Original FTP client running	x		x	x
Original FTP server running	x			x
Configured on Tectia		x	x	x
Fallback to FTP is possible	N/A			x
SFTP GUI available on Windows	N/A	x	N/A	N/A

2.1.1 Secure File Transfer Protocol (SFTP)

The Secure File Transfer Protocol (SFTP) is a de facto industry standard for secure file transfers, and it is natively supported by the Tectia client/server solution.

SFTP allows secure copying, moving, editing, and removing of files over TCP/IP networks. Scripted file transfers between enterprise servers can be secured by using the Tectia command-line SFTP and SCP (Secure Copy) tools with automated and ad hoc file transfers. For secure interactive file transfers, Windows users have the Tectia Secure File Transfer GUI.

Migration to secure file transfer can be made gradually. The Tectia products can be configured to allow fallback to plaintext FTP, in case an SFTP connection cannot be established. This makes it possible to use SFTP where it is applicable, but still permits connections to those FTP servers that must remain operational, for example, on a third-party network.

SFTP features include:

File transfer resume

The file transfer resume feature allows resuming interrupted file transfers instead of restarting the whole operation. The file transfer resume uses file hashing to determine the point of resume. For increased performance, you can apply the checkpoint/restart mechanism for optimum handling of interruptions in large file transfers.

Anonymous secure file transfers

Tectia Server can be configured to allow anonymous file transfers in environments, where user authentication is not required. When anonymous authentication is in use, users do not need to type in a password.

Easy SFTP subsystem chrooting

Tectia Server can be easily configured to confine users to a specific directory tree (e.g., home directory) for added security and ease of use.

2.1.2 FTP-SFTP Conversion

Tectia Server for IBM z/OS provides a FTP-SFTP conversion feature which captures plaintext FTP connections initiated by an FTP client and converts them to SFTP before the file transfer is started. All user names, passwords, and data are then transferred in encrypted format.

FTP-SFTP conversion works transparently

Existing FTP connections, including automated file transfers, can be transparently converted to SFTP without the need to modify the existing scripts or applications. Users can keep working with their familiar applications and use the existing IDs and authentication methods.

Easy and cost-effective conversion

The FTP-SFTP conversion module allows easy and cost-effective replacement of plaintext file transfers in large enterprise environments. Existing FTP scripts and client applications need no modifications. Only the FTP server will be replaced with an SFTP server.

Existing FTP clients can be used

Any existing client with FTP functionality can be used as before:

- Application hard-coded FTP
- Script-based automated FTP
- Interactive passive FTP, for example Windows Explorer FTP, web-browser-based FTP, command-line `ftp`, or FTP GUI applications.

Any Secure Shell server as counterpart

With Tectia Server for IBM z/OS (client tools), the FTP-SFTP conversion feature can connect to Tectia Server or any other Secure Shell server. When Tectia is used as the server-side counterpart, it can run on any supported platform: on Linux, HP-UX, AIX, Solaris, Windows, or IBM mainframe.

Easy filter rule configuration

Tectia Server for IBM z/OS can be configured to extract the user name, password, and destination host name from the secured FTP application, and to use them for authentication and connection setup on the Secure Shell SFTP server. The configuration is made as a filter rule in the Connection Broker configuration file, and the same rule can be defined to cover all FTP traffic. In large FTP environments, this simple rule setting can save the effort of defining hundreds of connection profiles which would otherwise be needed separately for each destination.

2.1.3 Transparent FTP Tunneling

Tectia Server for IBM z/OS provides transparent FTP tunneling which is the quickest way to secure file transfers. Both the original FTP client and server are retained and the file transfers are secured by encrypted tunnels.

Transparent tunneling of existing FTP connections

Transparent FTP tunneling provides a quick and easy way to secure FTP file transfers without the need to change the existing FTP scripts. Users can keep using the existing applications with their familiar IDs and authentication methods.

Full compatibility with FTP

Transparent FTP tunneling uses the Secure Shell v2 protocol to tunnel the existing FTP client and server connections providing full compatibility with existing unsecured FTP file transfer environment. Transparent FTP tunneling can be used to secure both interactive and unattended FTP sessions. Tectia Server for IBM z/OS supports passive FTP sessions that are initiated by an FTP client.

The existing FTP clients and servers are kept running, and they can continue performing their tasks, for example post-processing the transferred files.

Easy migration

Transparent FTP tunneling is an ideal solution for environments with thousands of complex FTP jobs with possible file transfer pre- and post-processing.

Transparent FTP tunneling also allows falling back to plaintext FTP, in case a Secure Shell tunnel cannot be established. This makes it possible to start migrating to secure file transfer usage immediately, and still be able to connect to the remaining FTP applications.

2.2 Secure System Administration

The Tectia client/server solution is used by system administrators as a replacement for unsecured login protocols, such as Rlogin, Telnet, and FTP. The Tectia Client software is installed in the system administrator's workstation and the Tectia Server software in the managed server. Typically, the number of servers is much higher than the number of client installations. In numerous active Tectia implementations there are thousands of Tectia Server instances installed within a corporate network.

With the Tectia client/server solution, administrators can log in to remote hosts securely, as their user ID and authentication information are transmitted over the network in encrypted format.

Users can connect to remote servers also with a command-line tool. In this example, a user named Susan connects to a server host for the first time, and receives the host key for validation:

```
$ sshg3 susanstrict@examplehost
Host key not found from database.
Key fingerprint:
```

```
xecic-fifub-kivyh-kohag-zedyn-logum-pycuz-besug-galoh-gupah-xaxby
You can get a public key's fingerprint by running
% ssh-keygen-g3 -F publickey.pub
on the keyfile.
Are you sure you want to continue connecting (yes/no)?
```

With the Tectia client/server solution, login can be easily done also in heterogeneous network environments including Windows, Unix, Linux, and IBM mainframe systems. This eliminates the need to deploy and maintain Secure Shell implementations from multiple vendors.

2.3 Secure Application Connectivity

The Tectia client/server solution can be used to replace unsecured TCP-based terminal connections (for example, Telnet) to business-critical enterprise applications. Through strong encryption and data integrity, the Tectia client/server solution protects sensitive data and passwords against unauthorized access, facilitating compliance with regulations and best practices. Strong authentication of users is supported through broad integration with third-party authentication systems, including RSA SecurID and public-key infrastructure (PKI).

The Tectia client/server solution offers the following ways of securing data communications between standard TCP-based applications:

- **Tunneling**, or port forwarding, is a way to forward otherwise unsecured application traffic through Secure Shell. Tunneling can provide secure application connectivity, for example, to POP3, SMTP, and HTTP-based applications.

  Tunneling provides encryption and strong two-factor authentication to third-party network client applications. Tectia allows different forms of tunnels depending on the environment and type of usage of the workstations or user terminals.

  The Secure Shell v2 connection protocol provides channels that can be used for a wide range of purposes. All of these channels are multiplexed into a single encrypted tunnel and can be used for tunneling (forwarding) arbitrary TCP/IP ports and X11 connections.

- The `sshg3` command-line tools can be used interactively or in scripts.

Secure connectivity over Internet

Tunneling makes it possible to access e-mail from any type of Internet service irrespective of the access method (modem, GPRS, 3G, a DSL line, a cable connection, or a hotel Internet service). As long as the users have a TCP/IP connection to the Internet, they can get their e-mail and access other resources from anywhere in the world securely.

Non-transparent tunnels

Tectia Client supports non-transparent application tunneling, which means that the tunneled applications need to be defined on the basis of the TCP ports they use. Applications with dynamic ports are not supported.

Transparent tunnels

With the transparent TCP tunneling feature activated on the client-side (on Tectia ConnectSecure), TCP-based applications can be tunneled transparently without changing the end-user experience or requiring any modifications on the applications, thus reducing the total cost of ownership.

Static tunnels

Tectia Client can also be used for application protection using the static tunneling feature. As opposed to transparent TCP tunneling, static tunnels are configured so that an application connects to a local port running Tectia Client, and the Client tunnels the application to a specified remote host.

Chapter 3 Features and Benefits

The Tectia client/server solution is an ideal lightweight solution for secure file transfer, secure remote logins for system administration purposes, and secure use of e-mail and other business applications. The Tectia client/server solution is available for most Unix and Linux platforms (including Linux on IBM System z), for Microsoft Windows, and for IBM mainframes.

The Tectia client/server solution works with any type of Internet (TCP/IP) connection - ADSL, ISDN, modem, Ethernet, WLAN, PPPoE - thus making it widely applicable, totally independent of the network topology, and independent of network address translations.

Here we describe first the features common to all Tectia client/server solution products and then those specific to Tectia Server for IBM z/OS.

There are some differences in the availability of features between versions of the Tectia products. For information on the features supported on each product version, see Table 6.3.

3.1 Tectia Client/Server Solution Features

The following general features are available with all products of the Tectia client/server solution.

Compliance with the IETF Secure Shell standards

The Tectia client/server solution implements the Secure Shell (version 2) protocol as defined by the IETF Proposed Standard RFC specifications. SSH Communications Security is the original developer of Secure Shell and has been an active driver of the Secure Shell standardization in the IETF.

Comprehensive cryptographic support

The Tectia client/server solution offers state-of-the-art encryption with broad support for symmetric ciphers including 3DES, AES, Arcfour, Blowfish, SEED, and Twofish. Supported message authentication and public-key algorithms include MD5, SHA-1, SHA-2, Diffie-Hellman, DSA, RSA, and ECDSA.

Versatile command-line tools

The Tectia products include versatile command-line tools that can be used for remote login, remote command execution, and file transfer operations. These tools allow easy scripting of automated jobs such as secure file transfers or starting and stopping of services in remote locations.

Tunneling (port forwarding)

One of the key features of Secure Shell, in addition to secure terminal access and secure file transfers, is its ability to tunnel TCP-based application connections. The Tectia products allow static application tunneling where application client connections are routed through the local TCP port, and then securely tunneled to a remote Secure Shell server.

Automatic tunneling

Before an application can be tunneled, a Secure Shell connection needs to be established. When using the automatic tunneling feature, the Tectia client-side component listens to a specific port and establishes the encrypted connection automatically when the specific application is connecting to the local host port.

Firewall traversal

The Tectia products support SOCKS (4 and 5) and HTTP proxy for accessing Secure Shell servers located behind firewalls.

Multi-channel support

Multi-channel support allows users to have multiple terminal sessions, file transfers, and application tunnels that are multiplexed to a single Secure Shell connection without the need to authenticate every session separately.

Configurable re-keying policies

Administrators can configure the renewal period for session encryption keys according to the security requirements.

3.2 High Performance

Tectia applies third-generation Secure Shell protocol implementation, SSH G3, which has been optimized for higher performance in demanding file transfer and application tunneling environments. The SSH G3 architecture provides unparalleled Secure Shell encryption throughput and scalability for large organizations.

Client-side Connection Broker

The Connection Broker is a key component in the SSH G3 architecture, handling all protocol and cryptographic operations. Client-side memory consumption is reduced since there needs to be only a single Connection Broker instance running per user. Security is also further improved by isolating all security-critical operations including authentication data handling in a single component.

Higher throughput

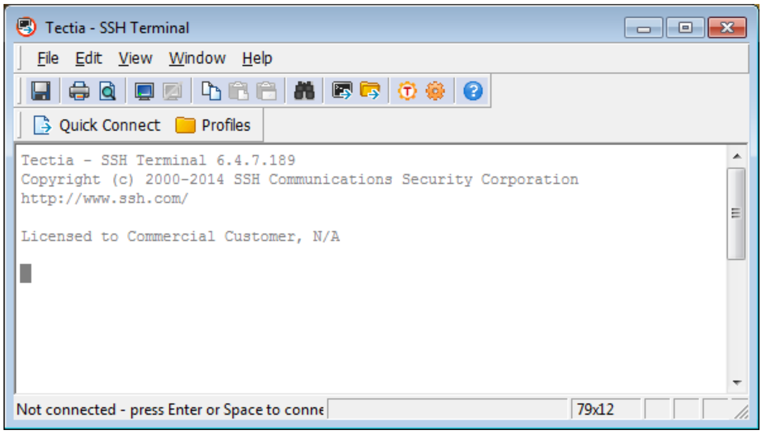
The SSH G3 architecture has been designed to minimize internal data handling such as data copy operations to minimize the throughput time in large file transfers.

3.3 Ease of Use

Tectia client/server solution is easy to install, and it can be configured using a GUI or by editing an XML configuration file.

Drop-in replacement for Telnet and FTP

The Tectia client/server solution provides easy and cost-effective means of securely replacing plaintext Telnet connections and file transfers in heterogeneous enterprise networks. Instead of plaintext connections to remote hosts, end users can use Tectia SSH Terminal that secures connections by encrypting all data. Alternatively, administrators can define that connections are automatically converted to secure SFTP or tunneled (encrypted) transparently to the users and their existing applications.



```
Tectia - SSH Terminal 6.4.7.189
Copyright (c) 2000-2014 SSH Communications Security Corporation
http://www.ssh.com/

Licensed to Commercial Customer, N/A
```

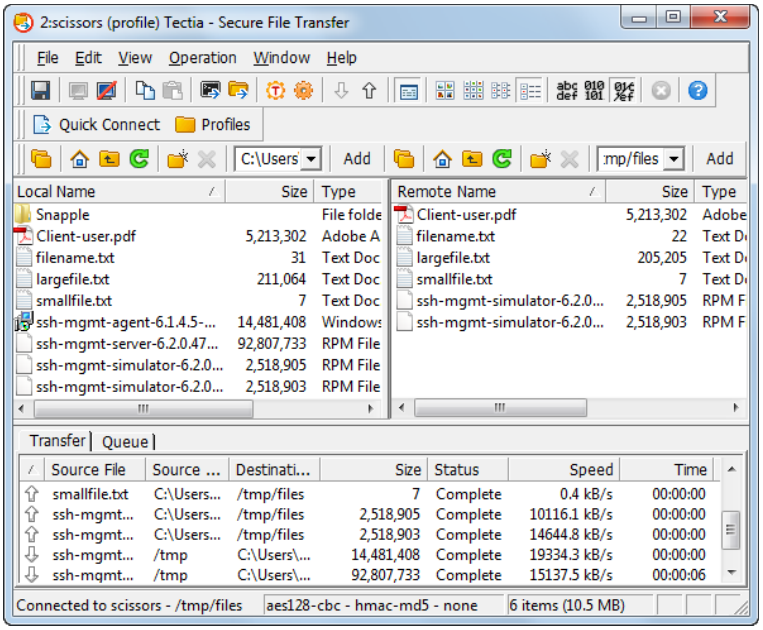
Not connected - press Enter or Space to conne

79x12

Figure 3.1. Tectia SSH Terminal for Secure Shell operations

Drag-and-drop file transfers

Windows users are provided with the Tectia Secure File Transfer user interface that allows users to securely drag-and-drop files between local Windows and remote Unix, Linux, Windows, and mainframe systems.



Local Name	Size	Type
Snapple		File folde
Client-user.pdf	5,213,302	Adobe A
filename.txt	31	Text Doc
largefile.txt	211,064	Text Doc
smallfile.txt	7	Text Doc
ssh-mgmt-agent-6.1.4.5-...	14,481,408	Windows
ssh-mgmt-server-6.2.0.47...	92,807,733	RPM File
ssh-mgmt-simulator-6.2.0...	2,518,905	RPM File
ssh-mgmt-simulator-6.2.0...	2,518,903	RPM File

Remote Name	Size	Type
Client-user.pdf	5,213,302	Adobe
filename.txt	22	Text D
largefile.txt	205,205	Text D
smallfile.txt	7	Text D
ssh-mgmt-simulator-6.2.0...	2,518,905	RPM F
ssh-mgmt-simulator-6.2.0...	2,518,903	RPM F

/	Source File	Source ...	Destinati...	Size	Status	Speed	Time
⇧	smallfile.txt	C:\Users...	/tmp/files	7	Complete	0.4 kB/s	00:00:00
⇧	ssh-mgmt...	C:\Users...	/tmp/files	2,518,905	Complete	10116.1 kB/s	00:00:00
⇧	ssh-mgmt...	C:\Users...	/tmp/files	2,518,903	Complete	14644.8 kB/s	00:00:00
⇩	ssh-mgmt...	/tmp	C:\Users\...	14,481,408	Complete	19334.3 kB/s	00:00:00
⇩	ssh-mgmt...	/tmp	C:\Users\...	92,807,733	Complete	15137.5 kB/s	00:00:06

Connected to scissors - /tmp/files | aes128-cbc - hmac-md5 - none | 6 items (10.5 MB)

Figure 3.2. Tectia Secure File Transfer GUI

3.4 Features Specific to Tectia Server for IBM z/OS

Tectia Server for IBM z/OS includes the following features in addition to the generic Tectia Server features described above:

File system support

Tectia Server for IBM z/OS supports MVS (including PS, PDS, PDSE, and VSAM) and USS file systems, and Generation Data Groups (GDG).

Coded character set translation

Full and configurable ASCII/EBCDIC conversion is supported as well as configurable CONVXLAT conversion tables for seamless cross-platform compatibility between IBM z/OS and Unix/Linux/Windows hosts.

Tectia Server for IBM z/OS allows the coded character sets to be configured for file transfer or terminal connections, and users can invoke the change code page command, **chcp**, to change the character set encodings dynamically.

Direct MVS data set access

Tectia Server for IBM z/OS incorporates direct streaming for all MVS file system operations, which improves file transfer performance by eliminating any additional memory and disk staging operations required previously for transferring files in MVS.

Direct MVS data set access is supported on both the client and server modules of Tectia Server for IBM z/OS. To work together with Windows, Unix, and Linux client hosts, direct streaming requires Tectia ConnectSecure on the client side.

MVS data set listing

Users of Tectia Client and ConnectSecure can list MVS data sets as files and folders, facilitating easy cross-platform file transfer between mainframe and non-mainframe systems. Users can drag-and-drop files with IBM z/OS by using SFTP GUI of Tectia Client.

Integrated mainframe authentication

Tectia Server for IBM z/OS supports RACF, ACF2, and TSS through standard SAF for seamless integration with the IBM mainframe authentication methods. Existing authentication and access control management tools can be used, and there is no need to create new profiles or passwords. Public-key authentication is also supported for both interactive and unattended connections.

Hardware-based key generation and storage

Both client and server-side private keys can be generated and stored on hardware by using Integrated Cryptographic Service Facility (ICSF) for maximum security.

SAF keyring support for certificate storage

Tectia Server for IBM z/OS supports storing client, server, and Certification Authority (CA) certificates on System Authorization Facility (SAF) keyrings. Optionally, the Tectia certificate validation can be omitted so that only the checks done by SAF will be used.

Hardware acceleration

Tectia Server for IBM z/OS supports 3DES, SHA-1, and AES hardware acceleration facilities for optimized encryption performance and lower CPU usage. All IBM-provided cryptographic hardware including CCF, PCICA, PCICC, PCIXCC, CPACF, and CryptoExpress2 are supported for acceleration.

Versatile command-line tools for scripting

Tectia Server for IBM z/OS includes versatile command-line tools that can be used for secure remote login, remote command execution, and secure file transfer operations. These tools allow easy scripting of automated jobs using JCL batch and USS scripts.

Secure TN3270 connectivity

Tectia Server for IBM z/OS allows transparent encryption of TN3270 application connections between Windows workstations and mainframes. There is no need to reconfigure existing terminal emulators. Mainframe RACF passwords can be used for authenticating Secure Shell connections. For more information, see *Tectia Server for IBM z/OS Administration Manual*.

File transfer profiles

File transfer profiles improve the usability of file transfers that involve automatic code set translation. File transfer profiles allow users to specify file transfer parameters (e.g., ASCII/EBCDIC translation and data set allocation parameters) that are used for specific file transfers. Both global and user-specific file transfer profiles are supported.

Support for System Management Facility (SMF)

Login and file transfer information can be collected and stored as SMF type 119 records.

Support for interfacing with Job Entry Subsystem (JES)

Tectia Server for IBM z/OS provides support for interfacing with the z/OS JES over SFTP: it is possible to submit a job and receive the spool output of the job, to display the status of all the user's jobs, and to delete jobs from the JES spool.

Chapter 4 Authentication

The Tectia client/server solution provides mutual authentication for the server and the client user. The Tectia client components authenticate the server and the Tectia Server authenticates the user's identity.

Secure Shell provides confidentiality also for user authentication. All data for identification and authentication purposes is exchanged in encrypted format between the client and server. User identity is not revealed to eavesdropping parties since the user is authenticated only after the connection has been secured.

Secure Shell supports multiple standardized methods for user authentication.

4.1 Server Authentication

Tectia uses cryptographic authentication for Secure Shell server hosts. Each server has a cryptographic key pair (a public key and a private key) that identifies the server. Whenever a Secure Shell client connects to a Secure Shell server, the server authenticates itself to the client cryptographically. This ensures that encryption and integrity protection are provided end-to-end between the client and the intended server. Server authentication also eliminates the danger of certain cryptographic attacks, especially man-in-the-middle attacks.

For the cryptographic authentication to work, the client must know the server's public key so that it can securely authenticate the server. The public key of the server must be distributed to each client. The private key of the server is never sent anywhere outside the server computer, but it is used by the server to create a digital signature that can then be verified by the client using the public key.

Secure Shell authenticates the Secure Shell server service of the server host. A host may run several Secure Shell server listeners on different ports. On Unix, each server can have a unique identity, if desired. However, typically there is only one Secure Shell server listener. If separate policies are needed or different services need to be offered for multiple use cases on a particular host, they can be defined dynamically in the Tectia Server configuration.

The server is authenticated with a digital signature based on a DSA or RSA public-key algorithm. Each server must have a public-private key pair. In implementations without support for certificates, clients refer to a local database of trusted server public keys.

Secure Shell also supports certificate authentication. Servers can authenticate themselves to the client with X.509 v3 certificates. When certificates are used, the client does not need to have a local database of trusted server public keys or server certificates. Instead, just the few trusted CA (certification authority) certificates are stored on the client, and the client trusts the servers whose certificates are signed by a trusted CA and certificate contents match the server hostname. Certificates provide scalability for authentication.

The Secure Shell server may have multiple identities (one DSA key, one RSA key, one DSA certificate, and one RSA certificate). During the key exchange in the Secure Shell connection, the Secure Shell client and server agree on which identity will be used in server authentication. Tectia Client prefers certificates over keys if trusted CA certificates have been configured, and otherwise DSA keys over RSA keys.

Tectia Server for IBM z/OS can also use X.509 certificates and RSA keys managed by the z/OS System Authorization Facility (SAF) as a host key.

4.2 User Authentication

Different methods can be used to authenticate users in Tectia. These authentication methods can be used separately or combined, depending on the level of functionality and security you want.

The Secure Shell server defines what methods are allowed in user authentication, and the Secure Shell client defines the order in which they will be tried.

By default, the Tectia client/server solution uses these user authentication methods:

- public-key
- password authentication
- keyboard-interactive

Public-key and certificate authentication are combined into the public-key authentication method.

The most commonly used user authentication methods are password, public-key, and host-based authentication. In public-key authentication, the users upload their public key files to the server and edit a configuration file. The server thus has a database of user public keys, similar to the client having its database of server public keys.

When certificates are used in user authentication, the user does not need to upload any public key files to the server prior to the connection, and the server does not need to have a database of user public keys or certificates. The server validates the user certificate by using the CA certificates that the server has been configured to trust and authorizes the login based on the user certificate contents.

Tectia Server for IBM z/OS can also use X.509 certificates and RSA keys managed by the z/OS System Authorization Facility (SAF) in user public-key authentication. Tectia Server for IBM z/OS includes two implementations of certificate authentication:

- keys and X.509 certificates in files and software cryptography; this is the same implementation that is available in the Tectia products on all platforms.
- keys and certificates managed by the z/OS System Authorization Facility (SAF) and cryptographic operations handled by the z/OS Integrated Cryptographic Service Facility (ICSF).

  The SAF validation may be complemented with the Tectia certificate validator and the Tectia implementation may use trusted keys stored in SAF.

If only SAF validation is used, the certificate validity period and revocation status are not checked. Securitywise, this equals normal public-key authentication, with keys stored securely in SAF. Note also that if SAF is used purely as a key store, the certificates have to be distributed to each host separately and the scalability advantage of PKI is lost.

Keyboard-interactive is not an authentication method in itself, but more like a common interface to various other authentication methods that are based on keyboard input. Password authentication, RSA SecurID, PAM (Pluggable Authentication Module), and RADIUS are examples of authentication methods that can be used over keyboard-interactive.

The highest security is achieved by using token-based certificate authentication where the certificate and the private key are stored on a cryptographic token, such as a smart card. Secure Shell supports also several other strong authentication methods, including the proprietary RSA SecurID.

4.3 Strong Authentication

The Tectia client/server solution offers several methods for user and server authentication, and true strong authentication using either public keys or public-key certificates. On Tectia Server for IBM z/OS, the public keys and certificates can be generated and stored on the mainframe's cryptographic hardware.

Server authentication with public keys or certificates

The Tectia client-side components authenticate the Secure Shell server in order to verify that they are connecting to the correct server. Likewise, the Secure Shell server authenticates the client user. The server can be authenticated by either (plain) public-key authentication or certificate authentication.

In (plain) public-key authentication, the server sends its public key to the client at the beginning of the first connection, and after the user has once verified and accepted the key, it is used in all future connections to that server.

In certificate authentication, the Tectia client-side components rely on a trusted third party, a certification authority (CA), to verify the server's identity. The signature of the certification authority in the server certificate guarantees the authenticity of the server certificate. When certificate authentication is used, the public key is included in the certificate that the server sends to the client.

User authentication with certificates and public keys

Client-side users can use certificates as proof of their identity. Certificates work like passports; the user proves his or her identity to a certificate authority once using public keys, receives a certificate, and from then on can authenticate using the certificate.

Public keys

Public-key authentication (without certificates) provides an easy-to-deploy and secure means of authenticating the users without the need to deploy and maintain a public-key infrastructure (PKI). Users will create key pairs for themselves and upload the public keys to the server for verification.

Authentication agent

Tectia Client and ConnectSecure incorporate authentication agent functionality that allows the caching of passphrases, eliminating the need to retype the passphrase each time when a connection is made. Passphrases are used in public-key authentication, which is more secure than password authentication. In addition, authentication can be "forwarded" to another host, allowing administrators to hop from one server to another without the need to store private keys in multiple servers.

Passwords

Tectia supports secure password-based authentication. Unlike in plaintext protocols such as Telnet and FTP, passwords are never sent in plaintext format over the network, thus eliminating the risk of exposing the password to unauthorized parties.

X.509 v3 certificates

Tectia supports X.509 v3 certificates for further security and scalability in large and dynamic network environments. Comprehensive support for IETF PKIX and PKCS standards ensures seamless interoperability with third-party PKI products.

Flexible certificate revocation

Tectia supports both Certificate Revocation Lists (CRLs) and Online Certificate Status Protocol (OCSP) for centralized revocation of user credentials. CRLs are automatically fetched from a local file or by using HTTP or LDAP, depending on the local settings and the CRL Distribution Point extension in the certificate. CRLs can also be imported offline in legacy environments.

Certificate lifecycle management

Tectia supports IETF PKIX standards (CMPv2) and Cisco Systems' Simple Certificate Enrollment Protocol (SCEP) for online certificate enrollment. Certificates can also be imported by using the PKCS #12 envelope format supported by most Certification Authorities (CAs).

Smart cards and PKI tokens

Tectia Client and ConnectSecure support smart cards, USB tokens, and other PKI authentication devices by supporting PKCS #11 and MSCAPI for interfacing with authentication keys. Strong, two-factor authentication overcomes the inherent security issues of password authentication.

Host-based authentication on Unix

Host-based authentication is a form of delegated trust authentication, where the Secure Shell server trusts the Secure Shell client host to authenticate the user. The user is verified by a `suid` binary (ssh-signer) on the client host which then confirms the user identity to the server in a communication signed with a root-owned host key. The client host is authenticated strongly with public key cryptography, thus the authentication does not rely solely on a host IP address or domain name. The Secure Shell host-based authentication utilizes strong cryptography for host identity verification.

Keyboard-interactive

Keyboard-interactive is a standards-based method of integrating Secure Shell with third-party authentication mechanisms that are based on keyboard input, without the need to modify the client-side application (Tectia Client). Keyboard-interactive is commonly used in conjunction with PAM and RADIUS on the server side.

OpenSSH and IBM Ported Tools key support

Tectia Server for IBM z/OS supports the legacy OpenSSH public-key format used by IBM Ported Tools, eliminating the need for manual key conversions in multi-vendor Secure Shell environments. The key-compatibility feature also allows easy migration of OpenSSH and IBM Ported Tools environments to Tectia.

Chapter 5 Use Cases

This chapter introduces typical use cases of the Tectia client/server solution. The use cases show examples of security services that you can quickly and effectively implement in typical enterprise environment.

For information on replacing unsecured FTP with secure file transfer, see Section 5.2 and Section 5.3.

For information on securing application connections, see Section 5.4 and Section 5.5.

For information on securing remote administrator connections, see Section 5.1.

5.1 Secure System Administration

One of the most widely used applications of Secure Shell-based products is to replace the unsecured login protocols, Rlogin, Telnet, and FTP, with secure alternatives. The Tectia client/server solution is based on SSH Secure Shell, used worldwide for secure system administration.

Figure 5.1 shows a typical Tectia environment for secure system administration. In this example, the managed servers reside in the perimeter network (DMZ), and the system administrator connects to them over the Internet using Tectia Client. In this scenario, Tectia Client software is installed in the system administrator's workstation and the Tectia Server software in the managed server. Tectia Server for IBM z/OS could be running at either end of the connection.

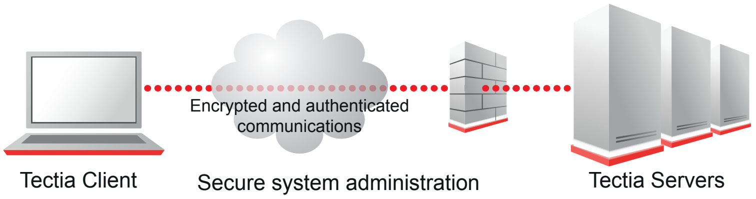

Encrypted and authenticated communications
Tectia Client
Secure system administration
Tectia Servers

Figure 5.1. Secure system administration with Tectia client/server solution

5.2 Replacing Plaintext FTP with FTP-SFTP Conversion

Tectia Server for IBM z/OS offers an easy way to secure plaintext FTP connections with a feature called FTP-SFTP conversion.

When FTP-SFTP conversion is enabled on Tectia Server for IBM z/OS client tools, it automatically captures all FTP connections initiated on the client side and converts the data to use the Secure File Transfer Protocol (SFTP), instead. The transferred files are sent to a Secure Shell SFTP server in encrypted format.

Tectia Server for IBM z/OS should be installed on the same host with the FTP client, and a Secure Shell server must be installed on the same host with the original FTP server.

FTP-SFTP conversion can be configured to pick the user name, password, and destination host directly from the secured FTP client, and use them to open the secured communication channel. This removes the need for any additional configuration modifications or changes to the original FTP scripts or applications. In the Connection Broker configuration, this is done simply with one rule that can fit all FTP connections.

When the FTP-SFTP conversion is used, there is no need for a plaintext FTP server, as the connection is made to an SFTP server instead. This requires that any post-processing done by the FTP server must be redirected to be performed elsewhere.

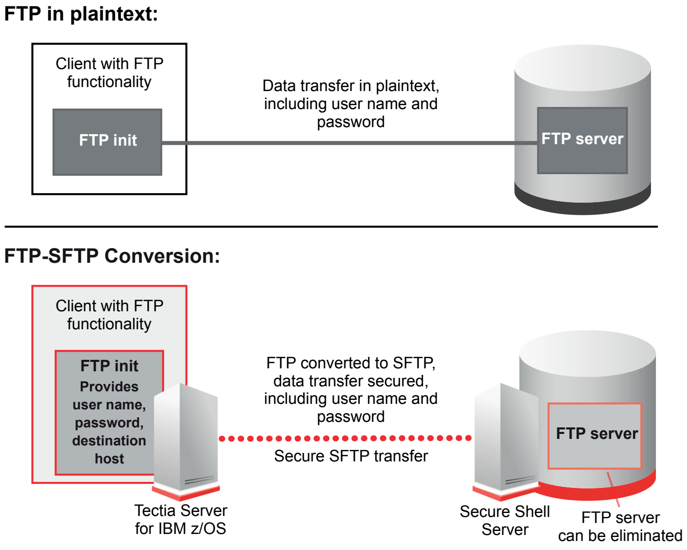

FTP in plaintext:
Client with FTP functionality
FTP init
Data transfer in plaintext, including user name and password
FTP server
FTP-SFTP Conversion:
Client with FTP functionality
FTP init
Provides user name, password, destination host
FTP converted to SFTP, data transfer secured, including user name and password
Secure SFTP transfer
FTP server
Tectia Server for IBM z/OS
Secure Shell Server
FTP server can be eliminated

Figure 5.2. Using FTP-SFTP conversion

5.3 Securing Plaintext FTP with Transparent FTP Tunneling

Plaintext FTP is an inherently unsecured, but a widely used method of transferring files. Tectia Server for IBM z/OS with its client tools offers an easy way to secure file transfer connections with transparent FTP tunneling. This feature is most useful when there is need to secure lots of FTP scripts.

Transparent FTP tunneling allows the FTP service to use the existing scripts and applications as they are, so to the users and applications the Tectia FTP tunneling happens transparently. As the existing FTP applications are left running, for example, the FTP servers can keep performing all their designated post-processing jobs as earlier.

Transparent FTP tunneling captures the connections that use the FTP protocol and tunnels them in encrypted format via a Secure Shell server to the FTP server. Transparent FTP tunneling can be configured to pick the user name, password and destination host directly from the FTP client, and use them to open the secured communication channel. In the Connection Broker configuration, this is done simply with one rule that can fit all FTP connections.

The users can define connection profiles to perform transparent FTP tunneling of certain connections, or they can request the tunneling per FTP connection on command line.

For end-to-end security, Tectia Server for IBM z/OS should be installed on the same host with the FTP client, and a Secure Shell server should be installed on the same host with the FTP server. If end-to-end security is not required, the FTP server can also reside on a third host.

The FTP server side can be on any platform, Unix, Windows or mainframe. Tectia Server for IBM z/OS works ideally with Tectia products, but supports any SSH2-capable Secure Shell servers.

Transparent FTP tunneling can be used to secure both interactive and unattended FTP sessions. It also provides an option to fall back to plaintext FTP for easier migration.

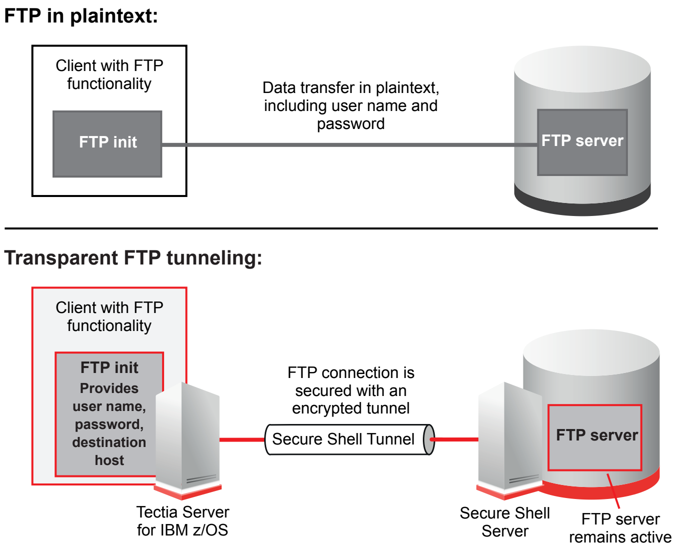

FTP in plaintext:
Client with FTP functionality
FTP init
Data transfer in plaintext, including user name and password
FTP server
Transparent FTP tunneling:
Client with FTP functionality
FTP init
Provides user name, password, destination host
FTP connection is secured with an encrypted tunnel
Secure Shell Tunnel
FTP server
Tectia Server for IBM z/OS
Secure Shell Server
FTP server remains active

Figure 5.3. Transparent FTP tunneling

5.4 Secure TN3270 Application Connectivity to IBM Mainframes

TN3270 terminal emulation is widely used on Windows workstations to provide enterprise end users with a direct access to IBM mainframe applications. While many organizations have not implemented encryption controls for TN3270 application connections, sensitive data and user passwords are constantly exposed in the enterprise networks.

With the Tectia solution, organizations can easily and cost-effectively secure their TN3270 connections completely transparently to end users and continue to use their existing TN3270 applications as before.

Transparent TN3270 tunneling requires that Tectia Client or ConnectSecure is installed on the Windows workstations. Next, the administrator specifies tunneling rules for the TN3270 application connection(s) that need to be secured. Alternatively, it is possible to require that all terminal connections initiated by a certain terminal emulator will be tunneled.

When the terminal client accesses a remote mainframe, Tectia captures the connection transparently and establishes a secure tunnel between the workstation and IBM z/OS system. All TN3270 application connection traffic is then transmitted over an encrypted Secure Shell tunnel, ensuring confidentiality of passwords and application data.

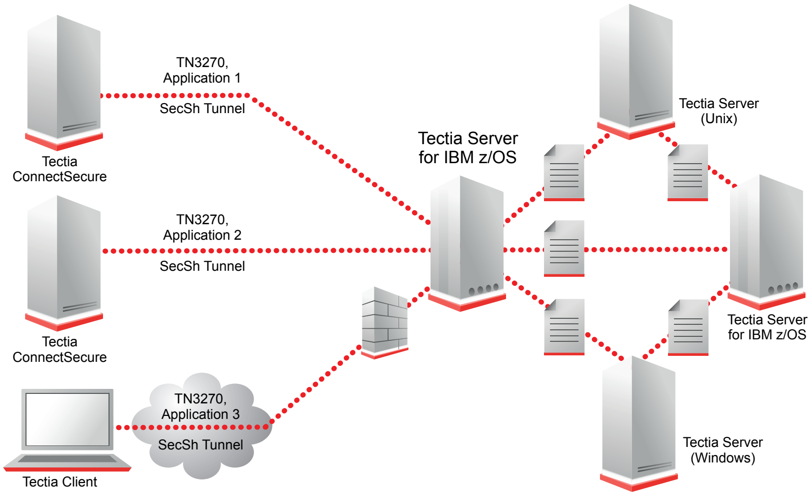


Figure 5.4. Secure TN3270 application connectivity to IBM mainframe and secure file transfer to and from IBM mainframes

End users can continue to use their existing terminal emulator clients and there is no need to introduce a new authentication layer, as RACF passwords or certificates can be used for authentication. End-user and application transparency makes Tectia a highly cost-effective solution for securing both interactive end-user connections and automated file transfers to and from IBM mainframes.

5.5 Securing Database Replication

Database replication is a frequently used operation, and often sensitive information is transmitted between the database server and its clients. The connections between database servers and their clients can be secured with the tunneling feature of Tectia. Tunneling means that data is transmitted in encrypted format and so secured from eavesdroppers.

In this example environment, we have Tectia Server for IBM z/OS running on the database client host, and Tectia Server has been installed on the database server. In addition, non-interactive authentication has been set up and enabled to perform the replication by scripts without user attendance.

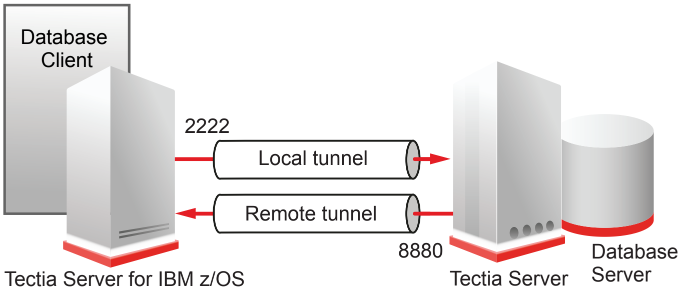


Figure 5.5. Tunneling database replication connections

The whole procedure of database replication through secure tunnels can be activated on the command line or with JCL scripts. For the purposes of database replication, you will need a script that establishes the tunnels, performs the replication and then closes the tunnels and the Secure Shell connection.

The tunnels can be local or remote. For local tunnels, the client application is configured to connect to a localhost port (2222 in this example) instead of the application server port. The script orders the client tools of Tectia Server for IBM z/OS to listen to local port 2222 and to tunnel its connections to the database server.

For the remote tunnels, you need to allocate a listener port (8880 in this example) on the remote server. Whenever a connection is made to this listener, the Tectia Server for IBM z/OS tunnels the connection over Secure Shell to the local client host and another connection is made from the client to a specified destination host and port.

The actual replication is then performed by the command script, and the data is transmitted securely in encrypted format.

Chapter 6 Product Specification

In this section you can find information on the supported operating systems, hardware requirements and supported authentication methods, cryptographic algorithms and protocols, and third-party products.

6.1 Supported Operating Systems

Tectia client/server solution products can be installed on Linux, Unix, and Windows platforms and they can run on any standard hardware capable of running the supported operating systems.

On Linux, Unix, and IBM System z platforms, Tectia Server includes also the client-side tools.

Note

Keep the operating system always fully patched, according to recommendations by the operating system vendor. The minimum patch levels required by Tectia products are mentioned in the Tectia product-specific installation instructions in the User Manuals and Administrator Manuals.

Tectia Client/Server support the mainframe operating systems listed in Table 6.1. For a complete list of supported operating systems per Tectia product, see *Tectia Client/Server Product Description*.

Table 6.1. Mainframe operating systems supported by Tectia Server and its client components

Operating system	Versions
IBM z/OS (zSeries)	2.2, 2.3, 2.4

6.2 Hardware and Space Requirements

The Tectia products can be run on any standard hardware capable of running the supported operating system versions. The machine should have a TCP/IP connection, and a DVD drive if the software is installed from the installation disk.

Table 6.2 summarizes the memory and disk space requirements.

Table 6.2. Memory and disk space requirements

	Client	ConnectSecure	Server	Server for z/OS	
RAM	any	any	1 GB[a]	1 GB[a]	
Disk space	100 MB	100 MB	100 MB	200 MB, 250 cylinders	

[a] For hundreds of simultaneous tunnels

6.3 Tectia Features per Product

The following table lists the features provided by Tectia Client, ConnectSecure, and Server.

Table 6.3. Tectia products and their features

Feature	Client	Connect-Secure	Server	Server for Linux z	Server for z/OS
FTP-SFTP conversion		x			x
Transparent FTP tunneling		x			x
Transparent TCP tunneling		x			
SFTP API for C (on all platforms)		x			
SFTP API for Java 1.4.2 (on 32-bit Linux, Solaris SPARC, and 32-bit and 64-bit Windows)		x			
Checkpoint/restart mechanism	x	x	x	x	x[a]
Streaming for high-speed transfers	x[b]	x[b]	x	x	x
Prefix for files in transfer	x	x	x	x	x
Versatile command-line tools (sshg3, scpg3, sftpg3)	x	x	x[c]	x	x
Terminal GUI on Windows	x	x			
File transfer GUI on Windows	x	x			
Configuration GUI on Windows	x	x	x		
CryptiCore encryption (on Linux and Windows)	x	x	x		
Support for connections to standard Secure Shell v2 servers	x	x	x[c]	x	x
Secure terminal server			x	x	x
FTP-SFTP conversion (server side)			x	x	x
SFTP server			x	x	x
Transparent tunneling server			x	x	x
Support for SFTP API (from other platforms)			x	x	x

[a] With HFS files.

[b] Requires Tectia Server as the counterpart.

[c] On Unix platforms (AIX, HP-UX, Solaris, Linux), only. On Windows, Tectia Client needs to be purchased separately.

6.4 Supported Authentication Methods

The following authentication methods and features are supported in the Tectia client/server solution.

6.4.1 Supported User Authentication Methods

The following user authentication methods are supported in the Tectia client/server solution.

Table 6.4. User authentication methods supported by the Tectia client/server solution

Authentication method	Tectia Server			Tectia Client, ConnectSecure, and the client tools on Server for IBM z/OS		
	Unix[a]	Windows	z/OS	Unix[a]	Windows	z/OS
Password	x	x	x	x	x	x
Public-key	x	x	x	x	x	x
Certificate	x	x	x[b]	x	x	x[b]
Host-based	x	x	x	x		x
Keyboard-interactive	x	x	x	x	x	x
PAM[c]	x			x	x	x
RSA SecurID[c]	x	x		x	x	x
RADIUS[c]	x	x		x	x	x
GSSAPI/Kerberos	x	x		x	x	

[a] Including Tectia Server for Linux on IBM System z

[b] Including certificates in files and SAF certificates

[c] Through keyboard-interactive

6.5 Supported Cryptographic Algorithms, Protocols, and Standards

Tectia client/server solution supports the following cryptographic algorithms and standards.

Table 6.5. Tectia client/server solution supports the following algorithms

Used for	Algorithm	
Key exchange	SHA-1:	diffie-hellman-group1-sha1
		diffie-hellman-group14-sha1
		diffie-hellman-group-exchange-sha1
	SHA-2:	diffie-hellman-group14-sha256
		diffie-hellman-group16-sha512
		diffie-hellman-group18-sha512
		diffie-hellman-group14-sha224@ssh.com
		diffie-hellman-group14-sha256@ssh.com
		diffie-hellman-group15-sha256@ssh.com
		diffie-hellman-group15-sha384@ssh.com
		diffie-hellman-group16-sha384@ssh.com
		diffie-hellman-group16-sha512@ssh.com
		diffie-hellman-group18-sha512@ssh.com
		diffie-hellman-group-exchange-sha256
	Elliptic curve:	ecdh-sha2-nistp256
		ecdh-sha2-nistp384
		ecdh-sha2-nistp521
Public key	DSA (768-, 1024-, 2048-, or 3072-bit key)	
	RSA (768-, 1024-, 2048-, or 3072-bit key)	
	ECDSA (256-, 384-, or 521-bit key)	
Data integrity	hmac md5 (16-byte key)	
	hmac md5-96 (16-byte key)	
	hmac sha-1 (20-byte key, FIPS PUB 198)	
	hmac sha-1-96 (20-byte key, FIPS PUB 198)	
	hmac-sha2-256 (32-byte key, FIPS PUB 180-3)	
	hmac-sha2-512 (64-byte key, FIPS PUB 180-3)	
	hmac sha224@ssh.com (28-byte key, FIPS PUB 198)	
	hmac sha256@ssh.com (16-byte key, FIPS PUB 198)	
	hmac sha256-2@ssh.com (32-byte key, FIPS PUB 198)	
	hmac sha384@ssh.com (48-byte key, FIPS PUB 198)	
	hmac sha512@ssh.com (64-byte key, FIPS PUB 198)	
Session encryption	3DES (168-bit key)	
	AES (128-, 192-, or 256-bit key, CBC or CTR mode)	
	Arcfour (128-bit key)	
	Blowfish (128-bit key)	
	SEED (128-bit key)	
	Twofish (128-, 192-, or 256-bit key)	

6.5.1 Hardware Acceleration of Cryptographic Operations

Tectia Server for IBM z/OS supports hardware acceleration on cryptographic operations with the following algorithms:

- 3DES
- AES (CBC and CTR)
- ECC (ECDH and ECDSA)
- SHA-1
- SHA-2
- RNG (random number generation)