SSH Tectia® Client/Server 6.1 (Windows)

Quick Start Guide

30 November 2009

SSH Tectia® Client/Server 6.1 (Windows): Quick Start Guide

30 November 2009

SSH Communications Security Corp.
Kumpulantie 3, FI-00520 Helsinki, Finland

Table of Contents

Chapter 1 About This Document

This guide gives quick instructions on getting started with SSH Tectia Client and Server. There are two alternative client/server products for different platform architectures: SSH Tectia Client/Server for AIX, HP-UX, Linux, Solaris, and Windows platforms, and SSH Tectia Server for Linux on IBM System z platforms.

The instructions are intended for a system where SSH Tectia Client is used to connect to SSH Tectia Server, and both are running on the Windows operating system.

The purpose of this quick guide is to help you in getting the SSH Tectia client/server solution up and running with the default settings so that you can evaluate the product.

The target audience to this guide are system administrators and other professionals who need to evaluate SSH Tectia products. To be able to use the information presented in this document, you should have system-administrator-level knowledge and know what SSH Tectia Client and Server are meant for.

SSH Tectia product family includes also SSH Tectia ConnectSecure that is capable of providing more advanced file transfer and application connectivity services in addition to the basic Secure Shell client services provided by SSH Tectia Client. Note that this guide concentrates on SSH Tectia Client as it is an entry-level product.

1.1 Reference Documents

The SSH Tectia client/server solution is described and more advanced user instuructions are given the following the product-specific manuals:

- *SSH Tectia Client/Server Product Description* contains general information about the product, its architecture, main features, and the product structure.
- *SSH Tectia Client User Manual* contains detailed instructions on installing, configuring and using SSH Tectia Client.
- *SSH Tectia Server Administrator Manual* contains detailed instructions on installing, configuring and using SSH Tectia Server.

- *SSH Tectia Client/Server Migration Guide* contains detailed instructions for upgrading the SSH Tectia client/server solution from 4.x to 6.1.

Instructions for evaluating SSH Tectia Client and Server on Unix are available in a separate quick quide *SSH Tectia Client/Server (Unix) Quick Start Guide*.

1.2 Component Terminology

The following terms are used throughout the documentation.

client computer	The computer from which the Secure Shell connection is initiated.
Connection Broker	The Connection Broker is a component included in SSH Tectia Client, SSH Tectia ConnectSecure, and SSH Tectia MFT Events as well as in the SSH Tectia Server for IBM z/OS client tools. Connection Broker handles all cryptographic operations and authentication-related tasks.
event	An event is a scheduled file transfer or command action pre-configured in the SSH Tectia MFT Events configuration. Events are run automatically according to the defined triggers and conditions.
file transfer GUI	SSH Tectia Client and ConnectSecure include a separate graphical user interface (GUI) for handling and performing file tranfers interactively.
host key pair	A public-key pair used to identify a Secure Shell server. The private key file is accessible only to the server. The public key file is distributed to users connecting to the server.
remote host	Refers to the other party of the connection, client computer or server computer, depending on the viewpoint.
Secure Shell client	A client-side application that uses the Secure Shell version 2 protocol, for example `sshg3`, `sftpg3`, or `scpg3` of SSH Tectia Client.
Secure Shell server	A server-side application that uses the Secure Shell version 2 protocol.
server computer	The computer on which the Secure Shell service is running and to which the Secure Shell client connects.
SFTP server	A server-side application that provides a secure file transfer service as a subsystem of the Secure Shell server.
SSH Tectia Client	A software component installed on a workstation. SSH Tectia Client provides secure interactive file transfer and terminal client functionality for remote users and system administrators to access and manage servers running SSH Tectia Server or other applications using the Secure Shell

	protocol. It also supports (non-transparent) static tunneling, and as an optional feature on Windows, transparent TCP tunneling.
SSH Tectia client/server solution	The SSH Tectia client/server solution consists of SSH Tectia Client, SSH Tectia ConnectSecure, SSH Tectia Server, and SSH Tectia Server for IBM z/OS.
SSH Tectia Connections Configuration interface	SSH Tectia Client, ConnectSecure, and Events have a user interface for configuring the connection settings to remote servers.
SSH Tectia ConnectSecure	A software component installed on a server host, but it acts as a Secure Shell client. SSH Tectia ConnectSecure is designed for FTP replacement and it provides FTP-SFTP conversion, transparent FTP tunneling, transparent TCP tunneling, and enhanced file transfer services. SSH Tectia ConnectSecure is capable of connecting to any standard Secure Shell server.
SSH Tectia MFT Events	A software component typically installed on a server host. SSH Tectia MFT Events is designed for automating file transfer and command events and for viewing their performance. The events can be created and maintained via the SSH Tectia MFT Events administration interface. SSH Tectia MFT Events is capable of connecting to any standard Secure Shell server.
SSH Tectia SFTP API	SSH Tectia ConnectSecure includes separate application programming interfaces (API) for C and Java. The APIs can be used by developers who develop secure file transfer applications or integrate SSH Tectia products into other systems.
SSH Tectia Server	SSH Tectia Server is a server-side component where Secure Shell clients connect to. There are three versions of the SSH Tectia Server product available: *SSH Tectia Server* for Linux, Unix and Windows platforms, *SSH Tectia Server for Linux on IBM System z*, and *SSH Tectia Server for IBM z/OS*.
SSH Tectia Server for IBM z/OS	SSH Tectia Server for IBM z/OS provides normal Secure Shell connections and supports the enhanced file transfer (EFT) features and transparent TCP tunneling on IBM mainframes.
SSH Tectia Server for Linux on IBM System z	SSH Tectia Server for Linux on IBM System z provides Secure Shell connections on Linux running on IBM System z platforms.
transparent FTP tunneling	An FTP connection transparently encrypted and secured by a Secure Shell tunnel.

transparent TCP tunneling	A TCP application connection transparently encrypted and secured by a Secure Shell tunnel.
tunneled application	A TCP application secured by a Secure Shell connection.
user key pair	A public-key pair used to identify a Secure Shell user. The private key file is accessible only to the user. The public key file is copied to the servers the user wants to connect to.

1.3 Documentation Conventions

The following typographical conventions are used in SSH Tectia documentation:

Table 1.1. Documentation conventions

Convention	Usage	Example
Bold	Menus, GUI elements, strong emphasis	Click **Apply** or **OK**.
→	Series of menu selections	Select File → Save
`Monospace`	Filenames, commands, directories, URLs etc.	Refer to `readme.txt`
Italics	Reference to other documents or products, emphasis	See *SSH Tectia Client User Manual*
#	In front of a command, # indicates that the command is run as a privileged user (root).	`# rpm --install package.rpm`
$	In front of a command, $ indicates that the command is run as a non-privileged user.	`$ sshg3 user@host`
\	At the end of a line in a command, \ indicates that the command continues on the next line, but there was not space enough to show it on one line.	`$ ssh-keygen-g3 -t rsa \ -F -c mykey`

Note

A Note indicates neutral or positive information that emphasizes or supplements important points of the main text. Supplies information that may apply only in special cases (for example, memory limitations, equipment configurations, or specific versions of a program).

Caution

A Caution advises users that failure to take or to avoid a specified action could result in loss of data.

1.3.1 Operating System Names

When the information applies to several operating systems versions, the following naming systems are used:

- **Unix** refers to the following supported operating systems:
  - HP-UX
  - IBM AIX
  - Red Hat Linux, SUSE Linux
  - Linux on IBM System z
  - Sun Solaris
  - IBM z/OS, when applicable; as SSH Tectia Server for IBM z/OS is running in USS and uses Unix-like tools.
- **z/OS** is used for IBM z/OS, when the information is directly related to IBM z/OS versions.
- **Windows** refers to all supported Windows versions.

1.4 Customer Support

All SSH Tectia product documentation is available at http://www.ssh.com/support/documentation/.

If the product documentation does not answer all your questions, you can find the SSH Tectia FAQ and Knowledge Base at https://support.ssh.com/.

If you have purchased a maintenance agreement, you are entitled to technical support from SSH Communications Security. Review your agreement for specific terms and log in at https://support.ssh.com/.

Information on submitting support requests, feature requests, or bug reports, and on accessing the online resources is available at http://www.ssh.com/support/contact/.

Chapter 2 Installation

This guide gives instructions for installing the SSH Tectia client/server solution on the Windows operating system.

The SSH Tectia products can also be run on other platforms. For a full list of supported operating systems and instructions on installing SSH Tectia on them, see *SSH Tectia Client User Manual* and *SSH Tectia Server Administrator Manual*.

The SSH Tectia installation packages for evaluation purposes are available at the http://www.ssh.com site, under **Evaluate & Buy**, or you can order an installation CD.

2.1 Preparing for Installation

Make the following preparations and check-ups before you start installing SSH Tectia Client and Server.

2.1.1 Hardware and Disk Space Requirements

SSH Tectia products do not have any special hardware requirements. They can be installed on any computer capable of running the supported operating system versions and equipped with a functional network connection.

The SSH Tectia Client installation requires about 100 MB of disk space. Note that SSH Tectia Client will save each user's settings in that particular user's personal directory.

The SSH Tectia Server installation requires 100 MB free disk space.

For general installation information, see *SSH Tectia Client User Manual* and *SSH Tectia Server Administrator Manual*.

2.1.2 Upgrading Previously Installed Secure Shell Software

If installed on the same machine, SSH Tectia Client and SSH Tectia Server should always be upgraded to be the same version, because there are dependencies between the common components.

Check if you have some Secure Shell software, for example earlier versions of SSH Tectia products or third-party Secure Shell server or client, running on the machine where you are planning to install the new SSH Tectia versions.

The following table shows you which SSH Tectia versions you need to uninstall before you can upgrade to SSH Tectia Client and Server 6.1. Versions marked "upgrade on top" will be automatically removed from the host during the installation procedure.

Table 2.1. Upgrade lines on Windows platforms

SSH Tectia version	Action
SSH Secure Shell 2.x-3.x	remove
SSH Tectia 4.0	remove
SSH Tectia 4.1-4.x	upgrade on top
SSH Tectia 5.0-6.x	upgrade on top

2.1.3 License File

SSH Tectia Client and Server requires a license to function. The license file is named `stc|sts61.dat`

Depending on the platform for which you have purchased SSH Tectia Client and Server, consider the following license-related issues:

- On Unix, you need to install the license file manually to directory: `/etc/ssh2/licenses`
- On Windows, the installation wizard automatically copies the license file to the correct directory when installing from an extracted online package or from the installation DVD.

  After installation, the license file is located in the default installation directory:

  `"C:\Program Files\SSH Communications Security\SSH Tectia\SSH Tectia AUX\licenses"`
- On the installation disk, the license files can be found in directory: `install/<platform>`
- In the online installation packages, the license files are included in the compressed files (zip/tar) together with the `releasenotes.txt` files and the PDF-format documentation.
- Evaluation packages have a temporary license for 45 days. On Unix machines, a banner message will remind users of how many days are left until the license expires.

2.1.4 Creating Operating System User Accounts

SSH Tectia Server does not have a user management program of its own - the user accounts are created with the standard operating system tools.

On Windows, user login requires the rights to log on locally and to access the remote computer from the network. Notice that on domain controllers, these rights are disabled by default. If SSH Tectia Server is installed on a domain controller, permissions to log on locally and to access the computer from the network must be enabled on the domain controller for the Domain Users group.

2.2 Installing SSH Tectia Software

This section introduces how SSH Tectia Client and Server are installed on the Windows platform.

You can download the installation packages from the web or order the installation disk.

For evaluation purposes, we recommend downloading an online installation package from the web:

1. Go to http://www.ssh.com/buy/downloads
2. Under **Evaluation Versions**, select **SSH Tectia Client and Server**.
3. Register and select the relevant product version.
4. Study the license agreement, and if you approve, click **Accept and continue**.
5. The SSH download service sends a verification code to your e-mail. Copy the code from e-mail to the Email-Address Verification window on the download site, and click **Complete registration and download**.
6. Download the selected SSH Tectia products. Save the compressed files to your local disk.
7. Proceed to the installation. See the instructions for SSH Tectia Client and Server per platform in the following sections.

2.2.1 Installing SSH Tectia Client on Windows

The Windows installation packages are provided in the MSI (Microsoft Installer) format. The same package is compatible with the supported 32-bit (x86) and the 64-bit (x64) versions of Microsoft Windows.

SSH Tectia requires that the operating system includes the following service packs:

- Microsoft Windows XP: Service Pack 2
- Microsoft Windows Server 2003: Service Pack 2
- Microsoft Windows Server 2008: Service Pack 1.

The online installation package is a zip file containing the license file and the executable Microsoft Installer (MSI) package.

On the installation DVD, the installation package for Windows is located in the `/install/windows/` directory.

You must have administrator rights to install SSH Tectia Client on Windows.

SSH Tectia Client includes support for Entrust certificates on Windows XP. The necessary libraries are included in the installation automatically.

The installation is carried out by a standard installation wizard. The wizard prompts you for information, copies the program files and sets up the client.

To install SSH Tectia Client on Windows, follow the instructions below:

1. *Online package:* Extract the installation zip file contents to any temporary location.

   

   Note

   The license file will be automatically imported to the correct location, if you extract the contents of the online `.zip` package before running the `.msi` installer, or if you are installing from an installation disk.

   If you run the `.msi` installer directly from the online `.zip` package, you need to manually import the (`stc61.dat`) license file. The installation wizard will show an error message about missing license file, and when you attempt to start the SSH Tectia Client, you are prompted to import the license manually to the correct directory:

   `"C:\Program Files\SSH Communications Security\SSH Tectia\SSH Tectia AUX\licenses"`

2. Locate the installation file `ssh-tectia-client-<version>-windows.msi` (where `<version>` corresponds to the version and build number, for example `6.1.4.123`).

3. Double-click the installation file, and the installation wizard will start.

4. Follow the wizard through the installation steps and fill in information as requested.

5. Select **Typical** installation. For SSH Tectia Client, it includes the `sshg3.exe`, `scpg3.exe`, and `sftpg3.exe` command-line tools, and the graphical user interface for terminal and file transfer.

   To install all components, select **Complete** when the wizard prompts for the setup type.

6. When the installation has finished, click **Finish** to exit the wizard.

The default installation directory is "`C:\Program Files\SSH Communications Security\SSH Tectia`".

The installation creates a new program group in the **Start** → **Programs** menu. The default name for this program group is **SSH Tectia Client**.

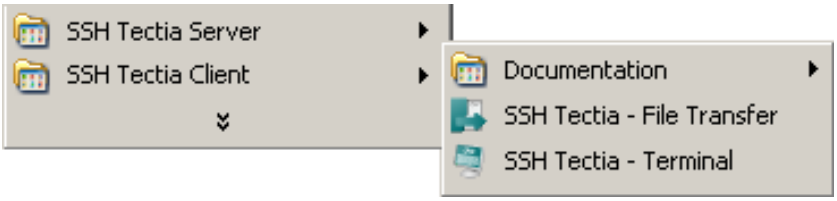

Figure 2.1. The SSH Tectia Client program group

2.2.1.1 Desktop Icons

During installation, SSH Tectia icons are added to your desktop. There are separate program icons for SSH Tectia Terminal and File Transfer windows. They both start the same application, `ssh-client-g3.exe`, but the former icon starts with the terminal window and the latter with the file transfer window.

Figure 2.2. The SSH Tectia Terminal icon

Figure 2.3. The SSH Tectia File Transfer icon

2.2.2 Installing SSH Tectia Server on Windows

The Windows installation packages are provided in the MSI (Microsoft Installer) format. The same package is compatible with the 32-bit (x86) and the 64-bit (x64) versions of Microsoft Windows Server 2003 and Server 2008.

SSH Tectia requires that the operating system includes the following service packs:

- Microsoft Windows XP: Service Pack 2
- Microsoft Windows Server 2003: Service Pack 2
- Microsoft Windows Server 2008: Service Pack 1.

The online installation package is a zip file containing the license file and the executable Microsoft Installer (MSI) package.

On the installation disk, the installation package for Windows is located in the `/install/windows/` directory.

You must have administrator rights to install SSH Tectia Server on Windows.

Note

SSH Tectia Server cannot be installed on file systems that do not support permissions (for example, FAT16 or FAT32). The hard disk partition where SSH Tectia Server is installed must use the NTFS file system.

SSH Tectia Server includes support for Entrust certificates on Windows XP. The necessary libraries are automatically included in the installation.

The installation is carried out by a standard installation wizard. The wizard will prompt you for information and will copy the program files, install the services, and generate the host key pair for the server.

To install SSH Tectia Server on Windows, follow the instructions below:

1. *(Online package)* Extract the installation zip file contents to any temporary location.

   

   ### Note

   The license file will be imported automatically, if you extract the contents of the online `.zip` package before running the `.msi` installer, or if you are installing from the DVD.

   If you run the `.msi` installer directly from the online `.zip` package, you need to manually import the license file (`sts61.dat`). The installation wizard will show an error message about missing license file, and when you attempt to start the SSH Tectia Server, you are prompted to import the license manually to the correct directory:

   ```
   "C:\Program Files\SSH Communications Security\SSH Tectia\SSH Tectia AUX\licenses"
   ```

2. Locate the installation file `ssh-tectia-server-<version>-windows.msi` (where `<version>` shows the SSH Tectia Server version and build number, for example `6.1.4.123`).

3. Double-click the installation file, and the installation wizard will start.

4. Follow the wizard through the installation steps and fill in information as requested.

   The installation wizard will display options **Typical**, **Custom** and **Complete**. However, there is only one version of SSH Tectia Server in the package, so all options install the same software.

   The server host key is generated during the installation. The key generation may take several minutes on slow machines.

5. When the installation has finished, click **Finish** to exit the wizard.

6. Reboot your computer always after installing or upgrading SSH Tectia Server.

SSH Tectia Server will start automatically every time the computer is started, and it stays running in the background. SSH Tectia Server displays no icons on the desktop, but you can see it listed in the Windows **Start** → **Programs** menu.

7. Usually there is no need to manually restart SSH Tectia Server.

   If you need to restart SSH Tectia Server (for example because of a missing license or because some other secure shell software is running on port 22), use the SSH Tectia Server Configuration GUI as follows:

   a. In the Windows **Start** menu, open **Programs** → **SSH Tectia Server** → **SSH Tectia Server Configuration**.

   b. Under the Server Status, click the **Start Server** button.

      The Server will start, and the status changes first to `Starting...` and then to `Running`.

   c. To exit the SSH Tectia Server configuration GUI, click **OK**.

2.2.3 Installation Complete

After a successful installation, SSH Tectia Client and SSH Tectia Server are automatically started at reboot and they keep running in the background until you stop them manually, or shut the host down.

You can use SSH Tectia Client and SSH Tectia Server with the default settings to test their functions. For instructions of opening a secure connection for the first time, see Chapter 3.

It is also possible to customize the behaviour of the SSH Tectia client/server solution according to current needs. To learn more about modifying the SSH Tectia configuration for different purposes, refer to the later chapters in this manual:

- Chapter 4 explains configuring of authentication methods
- Chapter 5 explains secure file transfer
- Chapter 6 explains securing application connections.

2.3 Removing SSH Tectia Software

If you need to remove the SSH Tectia Client and Server software, follow the instructions below.

Note

The uninstallation procedure removes only the files that were created when installing the software. Any configuration files have to be removed manually from directory `"C:\Program Files\SSH`

`Communications Security\SSH Tectia\SSH Tectia Server"` and from each user's `%APPDATA%\SSH` directory.

2.3.1 Removing SSH Tectia Client and Server from Windows

To remove SSH Tectia Client and Server from a Windows environment, follow the instructions below:

1. From the **Start** menu, open the Windows **Control Panel**.
2. Double-click **Add or Remove Programs**.
3. From the program list, select first **SSH Tectia Client** and click **Remove**, and then select **SSH Tectia Server** and click **Remove**.
4. Click **Yes** to confirm in both cases.
5. After you have uninstalled SSH Tectia Server, the system will prompt you to restart Windows.

Chapter 3 Connecting to Remote Server

This section explains how you can log in from SSH Tectia Client to SSH Tectia Server using password authentication with the default settings. The default settings on SSH Tectia Client and Server allow login with passwords, public keys, and GSSAPI. By default, passwords are used for user authentication, and public keys for server authentication.

You are expected to have a user account on the remote server, where you will connect, and the server must have a SecureShell server running. In the example below, you can connect within the local machine, to make sure that you know the server's address and that it has SSH Tectia Server running.

3.1 First Connection with Password

On Windows, you can connect to a remote host by using the SSH Tectia terminal GUI as follows:

1. Open the SSH Tectia Terminal by clicking its icon on your desktop:

Figure 3.1. The SSH Tectia Terminal icon

2. To open a Secure Shell connection, click the **Connect** icon on the toolbar, or click **File** → **Connect** in the menu. Alternatively, you can hit `Enter` or `Space` on the keyboard when the (still disconnected) terminal window is active.

3. This opens the **Connect to Server** dialog where you can define the host you want to connect to:

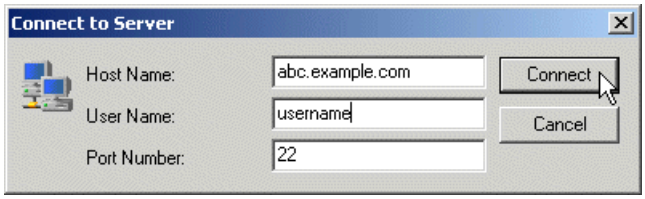

Figure 3.2. The Connect to Server dialog

Define the following information and click **Connect**:

- Host Name – the FQDN, short hostname, or the IP address of the remote host
- User Name – your user name on the remote host
- Port Number – 22 is the default Secure Shell listener port.

With later sessions, the values used in the previous connection will be pre-filled.

4. The server authentication phase starts. The remote server host will provide your local computer with its host public key. The host key identifies the server host.

   SSH Tectia Client checks if information on this key is already stored in your own host key directory. If not, the host key directory common to all users on your computer is checked next. If information on this host key is not found, you are asked to verify the new key.

   When public-key authentication is used to authenticate the server, *the first connection is very important*. When SSH Tectia Client receives a new server host key, it will display the host identification message. For example Figure 3.3:

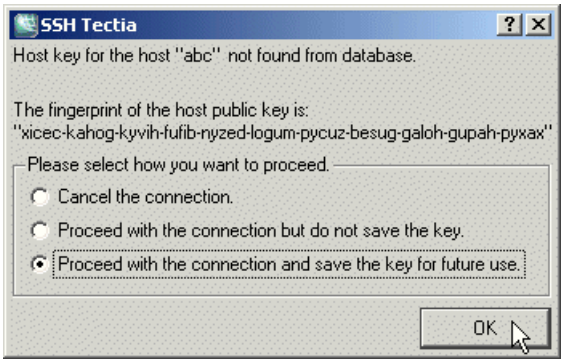

Figure 3.3. The host identification dialog – the first connection to a remote host

The message displays the fingerprint of the host's public key in the SSH Babble format that is a series of pronounceable five-letter words in lower case and separated by dashes.

5. Verify the validity of the fingerprint, preferably by contacting the administrator of the remote host computer by telephone. After verifying the fingerprint, it is safe to save information on the host key for future use. You can also choose to cancel the connection, or to proceed with this connection without saving the host public key information.

   

   Caution

   Never save a host public key without verifying its authenticity!

6. Click **OK** to close the host identification dialog.

   Information on the server public key will be stored on the client-side machine so that the client can later validate the key. On SSH Tectia Client, the public key information is stored in the "`%APPDATA%\SSH\HostKeys`" directory.

   After the first connection, only the locally stored information about the server public key will be used in server authentication.

7. The user authentication phase starts. You will be prompted to authenticate yourself to the server with your password.

   After the server has successfully authenticated you, the Secure Shell connection to the server is opened.

3.2 Creating Connection Profiles

On SSH Tectia Client on Windows, you can configure separate connection settings for each Secure Shell server you connect to. You can also create several profiles for the same server, for example, with different user accounts.

You can add connection profiles via the following views:

- Start SSH Tectia Terminal and click the **Profiles** button. Select **Add profile** from the drop-down menu, as shown in the following figure.

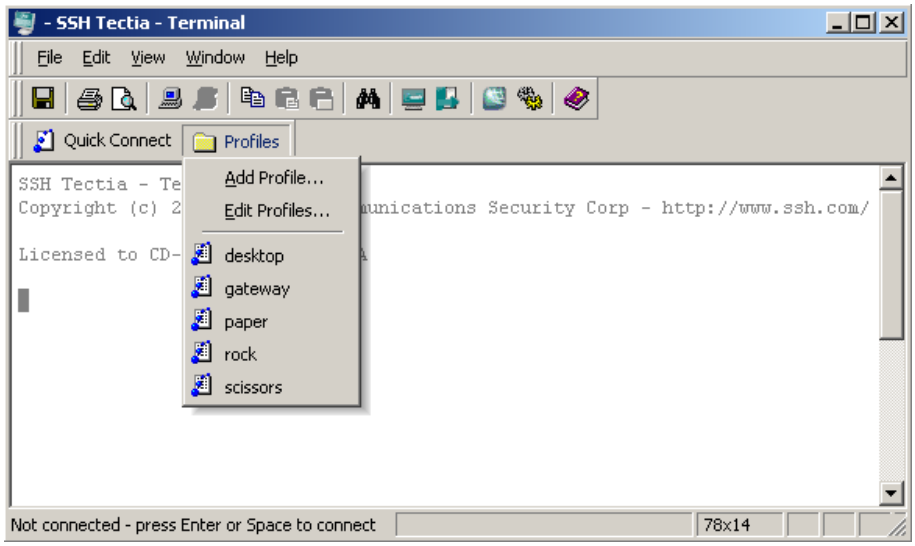

Figure 3.4. Adding connection profiles in SSH Tectia configuration GUI

- Start SSH Tectia Terminal and open the SSH Tectia Configuration tool by clicking the SSH Tectia icon on the toolbar.

  Or you can open the **SSH Tectia Configuration** tool by right-clicking the SSH Tectia tray icon in the system task bar and selecting **Configuration** from the shortcut menu.

  In the SSH Tectia Configuration view, go to the **Connection Profiles** page (as shown below) and click **Add profile**. Type a name for the profile and click **OK**.

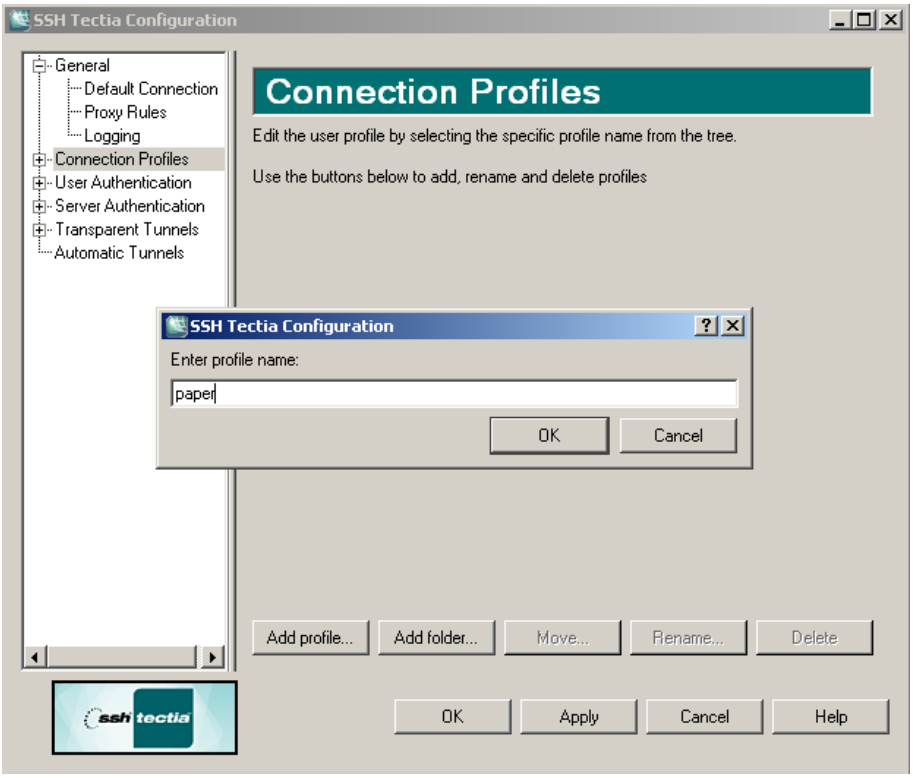

Figure 3.5. Adding connection profiles

Newly created connection profiles will inherit the default values for authentication, ciphers, MACs, tunneling, and advanced server settings defined under the **General** → **Default Connection** page. The values can be customized on the profile-specific tabbed pages, see Figure 3.6.

To rename a connection profile, select a profile and click **Rename**. Type a new name and click **OK**.

To remove a connection profile, select a profile and click **Delete**. You will be asked for confirmation. Click **OK** to proceed with the deletion.

3.2.1 Defining Connection Profile Settings

Under the **Connection Profile** page, on the **Connection** tab, you can define the protocol settings used in the connection. Any changed connection settings will take effect the next time you log in.

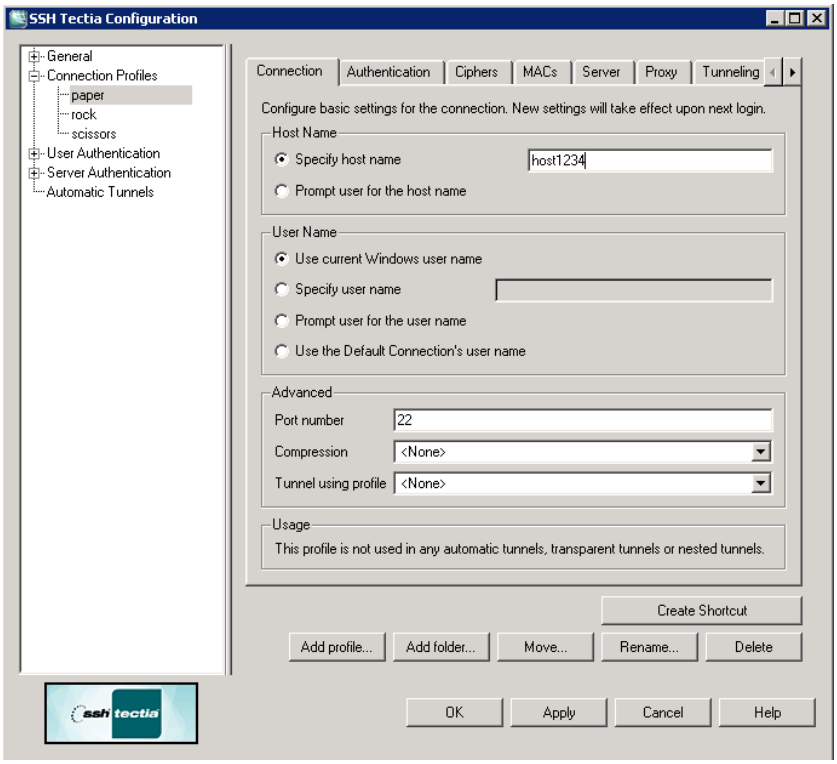

Figure 3.6. Configuring connection profiles

Hostname

Select **Specify host name** if you want to define a connection profile. Enter the name of the remote host computer to which you want to connect with the profile.

On Windows, when transparent TCP tunneling is used, '`%DESTINATION_HOSTNAME%`' is supported as the hostname definition. This option exists for backward compatibility reasons. From 6.0 onwards, you can define that SSH Tectia Client and Server uses the destination IP address received from the tunneled application with setting **Transparent Tunnels** → **Filter Rules** → **Use host name from the application** (in XML configuration: `hostname-from-app="yes"`).

Select **Prompt user for the host name** if you want that the user should enter the host name manually when connecting.

User Name

Select **Use current Windows user name** if the connection should always be made using the currently logged in Windows user name. This is similar to defining `%USERNAME%` (note the percent signs) as the user name.

Select **Specify user name** and enter the user name, if you want to define the user name to be used when connecting to the remote host computer. If you specify `%USERNAME%` (note the percent signs) as the username, it will be replaced with the name of the current Windows user account upon connecting.

Select **Prompt user for the user name** if the user should enter the user name manually every time when connecting.

Advanced

In **Port number**, enter the port number you want to use for the Secure Shell connection. The default port is 22.

Note

A Secure Shell server program must be listening to the specified port on the remote host computer or the connection attempt will not succeed. If you are unsure which port the remote host computer is listening to, contact the system administrator of the remote host.

Not needed now: In **Compression**, select the desired compression setting from the drop-down menu. Valid choices are **zlib** and **none**. Compression is disabled by default.

Not needed now: In **Tunnel using profile**, select the desired connection profile from the drop-down menu. Any nested tunnels will be created through the profile. For information on the tunneling features, refer to the *SSH Tectia Client User Manual*.

Chapter 4 Configuring Authentication Methods

The SSH Tectia client/server solution has separate authentication procedures for authenticating the servers and the users. The authentication is mutual, the client authenticates the server and the server authenticates the user.

The server configuration defines which authentication methods are allowed, and the client configuration defines the order in which the methods will be tried.

In this guide we introduce how public-key authentication is used in authenticating the remote SSH Tectia Server host. For user authentication, we introduce both the password authentication method, as it is set up by default, and public keys, which provide stronger security and make it possible to use non-interactive login securely.

For general information on user authentication methods, see technical note *SSH Tectia Client/Server User Authentication Methods* at http://www.ssh.com/resources/.

4.1 Server Authentication Methods

The server is authenticated with a digital signature based on a DSA or RSA public-key algorithm.

During the server installation process, one RSA key pair (with the file names `hostkey` and `hostkey.pub`) is generated and stored on the server host in directory "`C:\Program Files\SSH Communications Security\SSH Tectia\SSH Tectia Server`". By default, this key pair is used for server authentication.

For information on connecting to a remote server for the first time, see Chapter 3.

4.2 User Authentication with Passwords

The password and public-key authentication methods are set up by default on both SSH Tectia Client and Server. Passwords are the easiest method for authenticating users as then no configuring is needed on the server side. The passwords are protected from eavesdroppers, since all communication is encrypted.

On Windows, password authentication uses the Windows password to authenticate the user at login time.

For information on differences in user name handling on local and domain accounts, see *User Authentication with Passwords* in *SSH Tectia Server Administrator Manual*.

4.3 User Authentication with Public Keys

Public-key authentication is based on the use of digital signatures and provides very good authentication security.

To be able to use public keys in user authentication, you must first create a key pair on the client. One of the created key files is your public key, and the other is your secret private key.

The security level of the key pair depends on the complexity (or bit length) of the key. Larger keys are more secure, but generating and using them takes a longer time.

Note

The default size (2048 bits) of public-key pairs generated using SSH Tectia Client is very secure. Never generate keys smaller than 768 bits.

The server must know the user's public key, so you need to upload the public key to the server, but the private key is only in your possession.

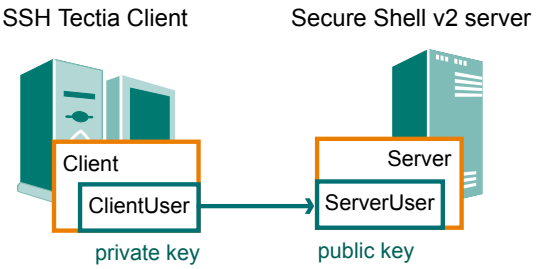

Figure 4.1. User public-key authentication

When you start logging in to a remote server, the client sends a signature to the server, and the server checks for matching public keys. If the key is protected with a passphrase, the server requests you to enter the passphrase.

Remember that your private-key file is used to authenticate you. Keep your private-key file in a secure place and make sure that no one else has access to it. If anyone else can access your private-key file, they can attempt to log in to the remote host computer pretending to be you. Define a passphrase to protect your private key, whenever possible.

Caution

Generate keys only on your personal computer that no one else can access! Do not store your private key on a computer that is shared with other users.

When you start using public-key authentication, do the following actions:

1. Generate a key pair. You can generate your own key files with the help of a built-in Public-Key Authentication Wizard (see Section 4.3.1).

   You can also import existing keys on the **Keys and Certificates** page of the SSH Tectia Configuration tool.

2. Upload your public key to the remote host computer (running the SSH Tectia Server) automatically (see Section 4.3.2).

Note

SSH Tectia Server supports also user public keys generated with OpenSSH. SSH Tectia Server can be configured to check the OpenSSH `authorized_keys` file in addition to the SSH Tectia `authorization` file and/or the `authorized_keys` directory. Public keys defined in the SSH Tectia locations have precedence over the keys in the OpenSSH file if the same key is defined in both.

These instructions assume that the client user is allowed to log in to the remote host, where SSH Tectia Server is running, using password authentication.

4.3.1 Creating Keys with Public-Key Authentication Wizard

On Windows, you can use the SSH Tectia **Public-Key Authentication Wizard** to generate a key pair. The wizard will generate two key files, your private key and your public key, and stores them in directory `%APPDATA%\SSH\UserKeys` on your local computer. The public key has `.pub` as the file extension, and the private key file has the same base file name as the public key but no file extension.

Public key pairs can also be generated with a command line tool `ssh-keygen-g3`. For instructions, see *Client User Manual*.

1. Open the SSH Tectia Configuration tool by clicking the SSH Tectia icon on the system task bar or on the SSH Tectia Client toolbar.

2. Go to **User Authentication** and select the **Keys and Certificates** page. Click **New key**.

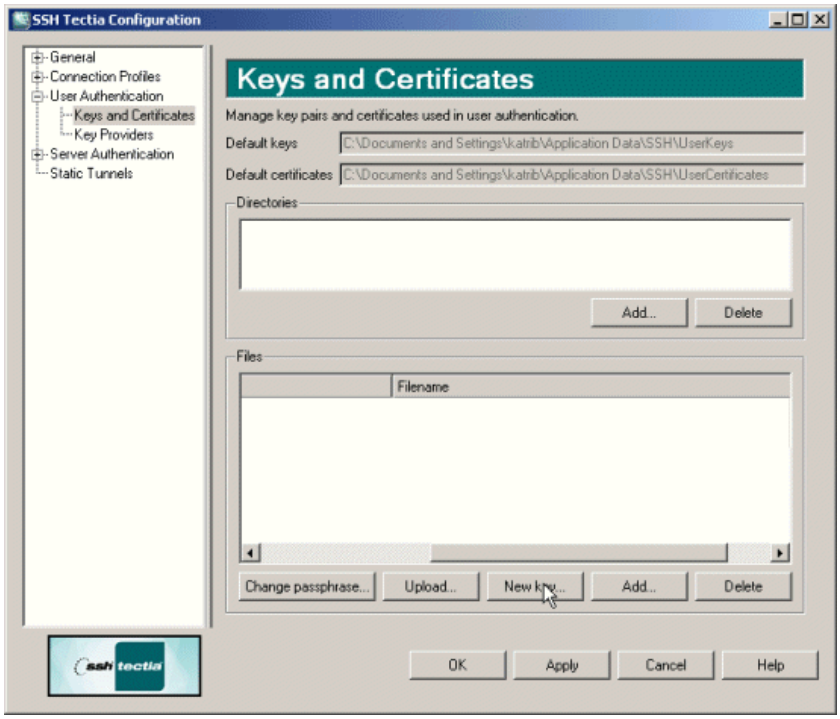

Figure 4.2. Configuration tool, Keys and Certificates view

3. The Public-Key Authentication Wizard starts.

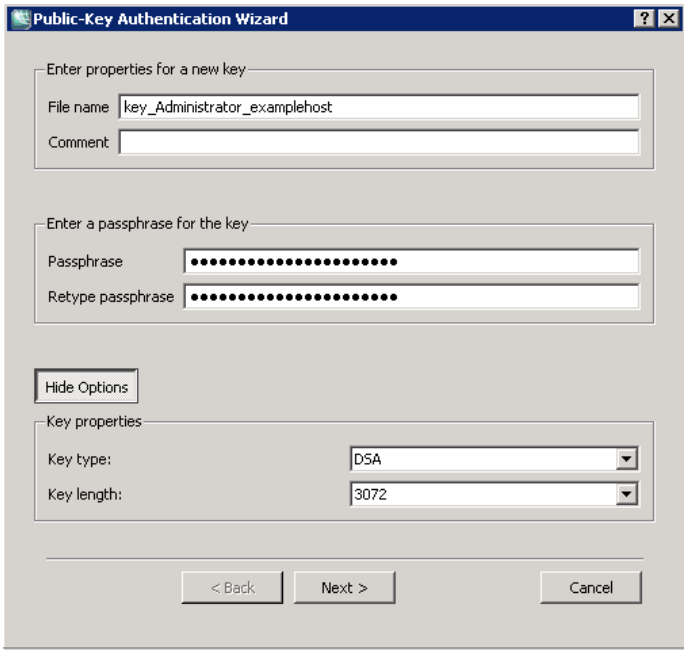

Figure 4.3. The Public-Key Authentication Wizard

4. Define the key properties and the required passphrase to protect your key pair.

   File Name
   : Type a unique name for the key file. SSH Tectia Client and Server suggest a name consisting of the user name and the host name.

   Comment
   : Write a short comment that describes the key pair. For example, describe the connection the key is used for. This field is not obligatory, but it helps to identify the key later.

   Passphrase
   : Type a phrase that is difficult to guess. Use at least 8 characters, both letters and numbers. Any punctuation characters can be used as well.

     If the key pair will used for automated jobs, you can leave the passphrase field empty to generate the key without a passphrase.

     You will be requested to enter the passphrase always when using the keys to authenticate yourself. The passphrase works in a way similar to a password and gives some protection for your private key.

     Memorize the passphrase carefully, and do not write it down.

   Retype passphrase
   : Type the passphrase again. This ensures that you have not made a typing error.

5. Click the **Advanced Options** if you what define the type of the key to be generated and the key length to be different from the defaults. By default, SSH Tectia Client and Server generates a pair of 2048-bit DSA keys.

   In the **Key Properties** fields, you can make the following selections:

   Key Type
   : Select the type of the key to be generated. Available options are DSA or RSA.

   Key Length
   : Select the length (complexity) of the key to be generated. Available options are 1024, 2048 or 3072 bits. Larger keys are more secure, but also slower to generate.

6. Click **Next** to proceed to uploading the key. The wizard continues with Step 3 in Section 4.3.2.

Uploading existing public keys to new remote servers is instructed in Section 4.3.2.

4.3.2 Uploading Public Key Automatically

Public keys can be automatically uploaded to servers that have the SFTP subsystem enabled, and by default, SFTP is enabled on SSH Tectia Servers. The **Public-Key Authentication Wizard** automatically uploads

each new public key to a remote host of your choise. The wizard lists all existing keys, and you can select a key to upload it to a remote server at any time.

The public key will be uploaded to the default user home directory (`%USERPROFILE%\.ssh2` on Windows, `$HOME/.ssh2` on Unix) on the remote server.

Note

The key user is required to have the `write` permissions to the to the key directory on the server, otherwise the automatic upload will fail. The administrator of the remote host computer may have restricted user access so that users are not able to configure public-key authentication for themselves even if public-key authentication is allowed in the server configuration.

1. To access the **Public-Key Authentication Wizard**, click **User Authentication** → **Keys and Certificates** on the tree view.
2. Select a key from the list and click **Upload**.
3. The **Upload Public Key** view of the wizard appears.

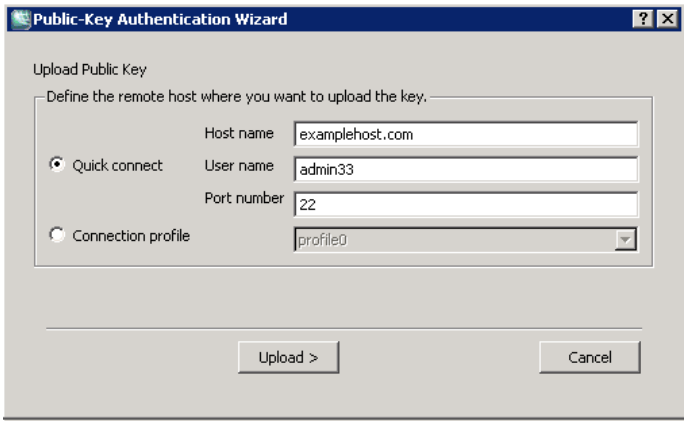

Figure 4.4. Uploading a key

Define the remote host where to upload the key:

Quick connect
: Select this option to define the remote **Host name** and your **User name** there. The default Secure Shell port is 22.

Connection profile
: Select a **Connection profile** from the drop-down list that specifies the desired remote host and user name.

4. Click **Upload** to transfer the key to the selected server. If you are already connected to the remote server host, the key upload starts immediately. If you are not connected, you will be prompted to authenticate on the server (by default with password).

4.4 Setting up Non-interactive Authentication for Automatic Scripts

When SSH Tectia Server is used for automated file transfer, you can create separate user accounts for file transfer purposes. When such user accounts are used only for non-interactive file transfers, it is advisable to disable terminal access on the server side. See instructions in Section 5.2.5.

Non-interactive authentication with public keys and scripted commands can be set for the SFTP accounts. For non-interactive batch jobs, you can use public-key authentication without a passphrase.

Running the client non-interactively requires that you have already saved the server's public host key on the client, and set up a non-interactive method for user authentication. Batch mode should be used non-interactively with command-line tools.

1. Generate an RSA key pair and leave the passphrase field empty.

   See instructions in Section 4.3.1

2. For uploading the keys, see instructions in Section 4.3.2 .

Caution

Make sure your private key is not accessible to others. This is especially important when the key is stored without a passphrase.

For more information on other non-interactive authentication methods, see *Authentication* in *SSH Tectia Server Administrator Manual*.

Chapter 5 Using Secure File Transfer

Secure File Transfer Protocol (SFTP) is a secure replacement for the plain-text FTP service. The SFTP service encrypts all files during the transfer.

This chapter shows how secure file transfer is used and describes a use case plus the required configuration changes.

5.1 Using SFTP on SSH Tectia Client

On SSH Tectia Client, the default settings for SFTP are applicable in most cases, so you can start experimenting with file transfers immediately. The SFTP service can be used on command line or via GUI. The GUI includes tooltips to guide you.

5.1.1 Using SFTP on Command Line

Command `sftpg3` is used on command line to connect to any host that is running a Secure Shell version 2 server with the SFTP server subsystem enabled.

The basic syntax of `sftpg3` command is:

```
sftpg3 username@remotehost
```

This logs you in to the remote host. After a successful login, you can, for example, fetch a file from the remote host to your local host with a command like this:

```
sftpg3> get file
```

To view the commands and options available with `sftpg3`, type `help` at the SFTP prompt:

```
sftp> help
```

For more instructions on the command, see the *SSH Tectia Client User Manual*.

5.1.2 Using SSH Tectia File Transfer GUI

SSH Tectia Client on Windows provides a graphical user interface for secure file transfers. To open the file transfer GUI, click the SSH Tectia File Transfer icon on your desktop.

Figure 5.1. The SSH Tectia File Transfer icon

In the SSH Tectia file transfer window, you can open a connection to a remote host through a connection profile defined in the Connection Broker configuration (click **Profiles**), or by using the **Quick Connect** option.

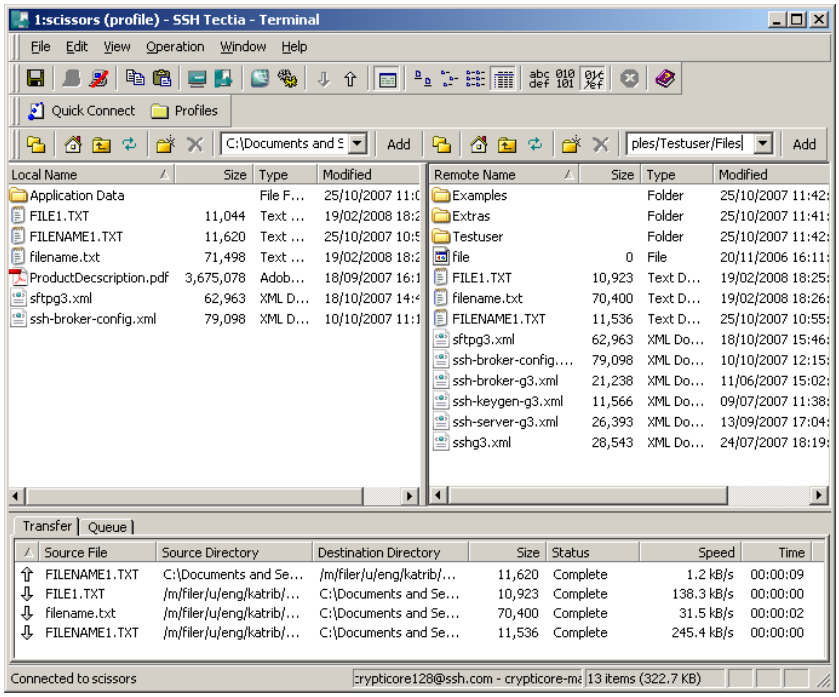

Figure 5.2. SSH Tectia File Transfer GUI

The SSH Tectia file transfer GUI makes it easy to download files from a remote host computer into your local computer and to upload files to a remote host. The SSH Tectia file transfer window operates much like Windows Explorer.

5.2 Configuring SSH Tectia Server for a Secure File Transfer Use Case

In this section we introduce a use case where SSH Tectia Server is used for automated secure file transfer, and show how to configure the SSH Tectia Server for it. SSH Tectia Client does not require any configuration changes.

The target of the SSH Tectia Server configuration changes is to improve the security of the system for automated file transfers. This calls for some user restrictions on the SFTP usage. In this secure file transfer use case, we define the following restrictions on the SSH Tectia Server:

1. Public keys are the only allowed authentication method. See instructions in Section 5.2.2.
2. SFTP service is allowed only for specially created user groups `SFTP-users` and `admin`. SFTP service is denied from all other users. See instructions in Section 5.2.3, Section 5.2.4 and Section 5.2.5.
3. Members of `SFTP-users` have access to their user-specific home folders only. This can be defined with virtual folders. See instructions in Section 5.2.4 and Figure 5.15.
4. Terminal access is allowed only for administrators, from everyone else, it is denied. See instructions in Section 5.2.3 and Section 5.2.5.

5.2.1 Opening SSH Tectia Server Configuration GUI

On Windows, SSH Tectia Server is configured through a graphical user interface.

Open the SSH Tectia Server Configuration GUI by clicking **Start** → **Programs** → **SSH Tectia Server** → **SSH Tectia Server Configuration**.

In order to gain access to the necessary sts; configuration settings, we first need to enable the advanced settings by clicking **Advanced** under **GUI Mode** on the **SSH Tectia Server** view:

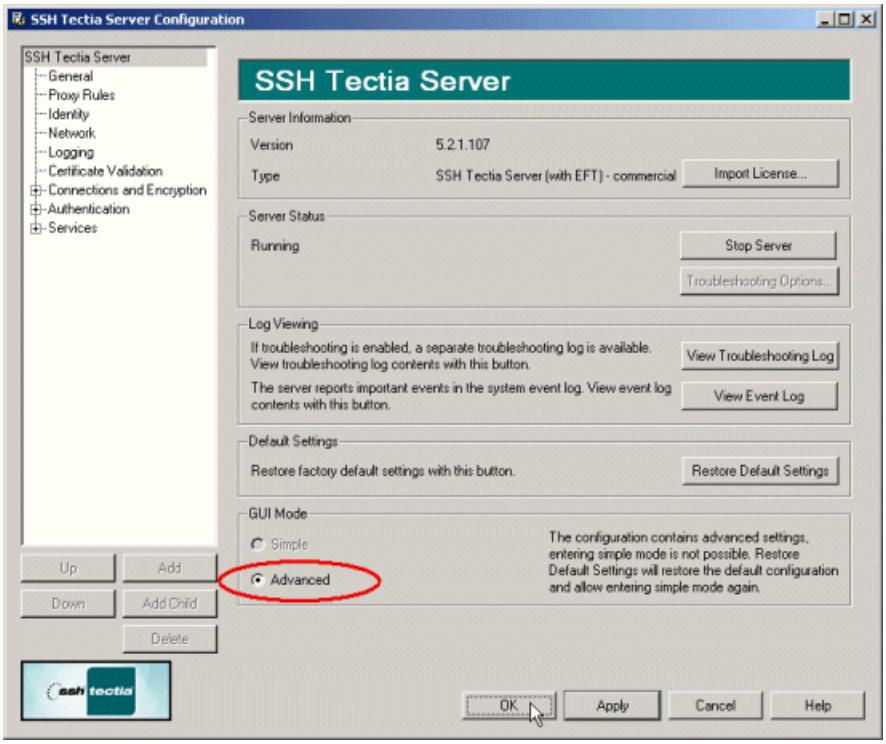

Figure 5.3. Enable Advanced GUI Mode

Now we can proceed to the actual configuration settings. See the example views below.

5.2.2 Enabling Public-Key Authentication

Define the public-key authentication as the only allowed authentications method under the **Authentication - Default-Authentication** page, on the **Parameters** tab.

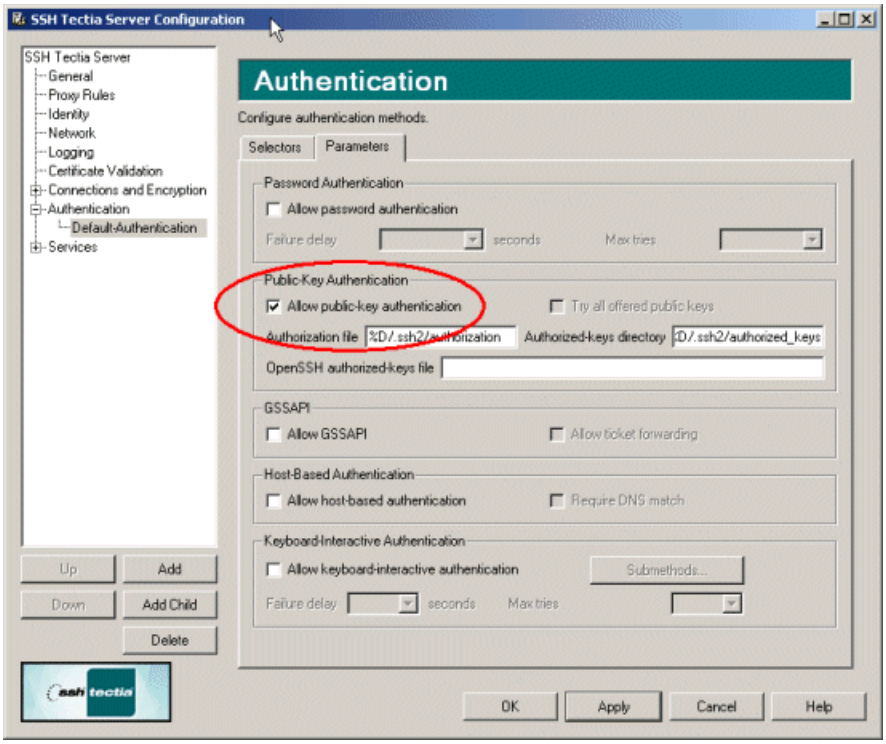

Figure 5.4. Enable only public-key authentication

5.2.3 Settings for the Admin Group

Here we create a user group with administrator rights and allow all actions and services for the members of the group.

1. Under the **Services** page, click **Add** to create a group for administrators.

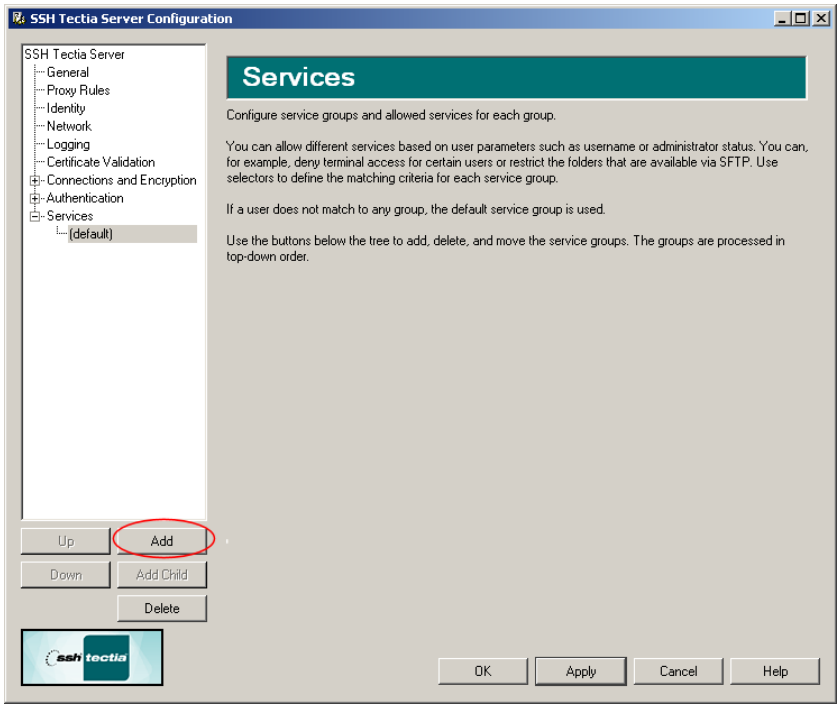

Figure 5.5. Start creating a user group

SSH Tectia Server will use a placeholder name `group1` for a newly created group.

2. On the **Basic** tab, name the group `admin` and choose **Allow** or **Allow all** for all services, **Terminal, Commands, Local Tunnels,** and **Remote Tunnels**.

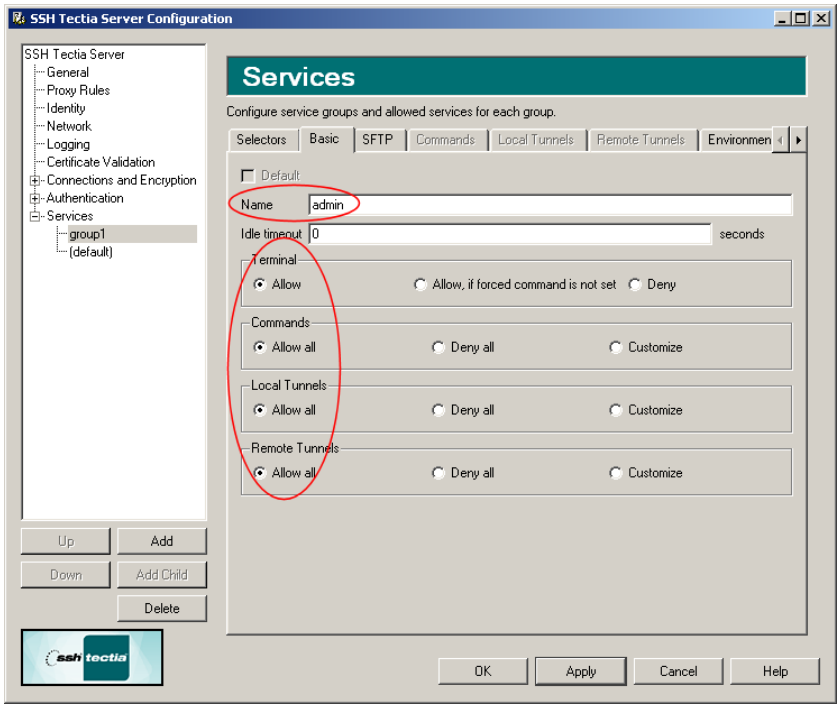

Figure 5.6. Name the group 'admin' and allow all services

3. Go to the **Selectors** tab, and click **Add Selector**. On the **Add Selector** tab, choose selector type **Administrator**, and click **OK**.

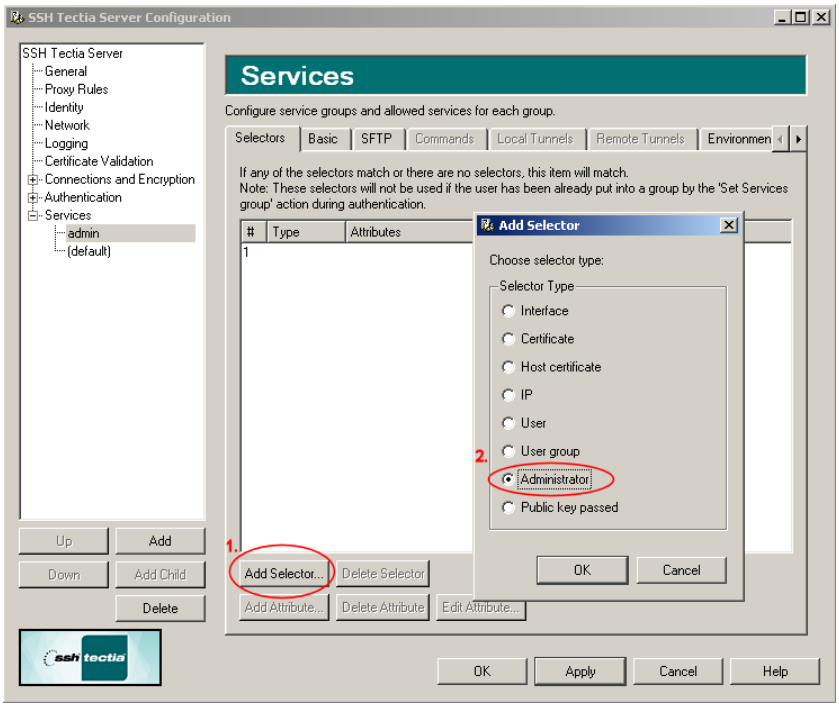

Figure 5.7. Define the group selector as administrator

4. When the **Administrator Selector** view opens, select **Is Administrator**, and click **OK**.

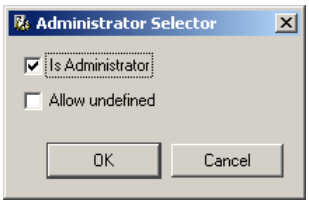

Figure 5.8. Define user group as administrator group

5. On the **SFTP** tab, allow the SFTP service for the `admin` group, and keep the default settings.

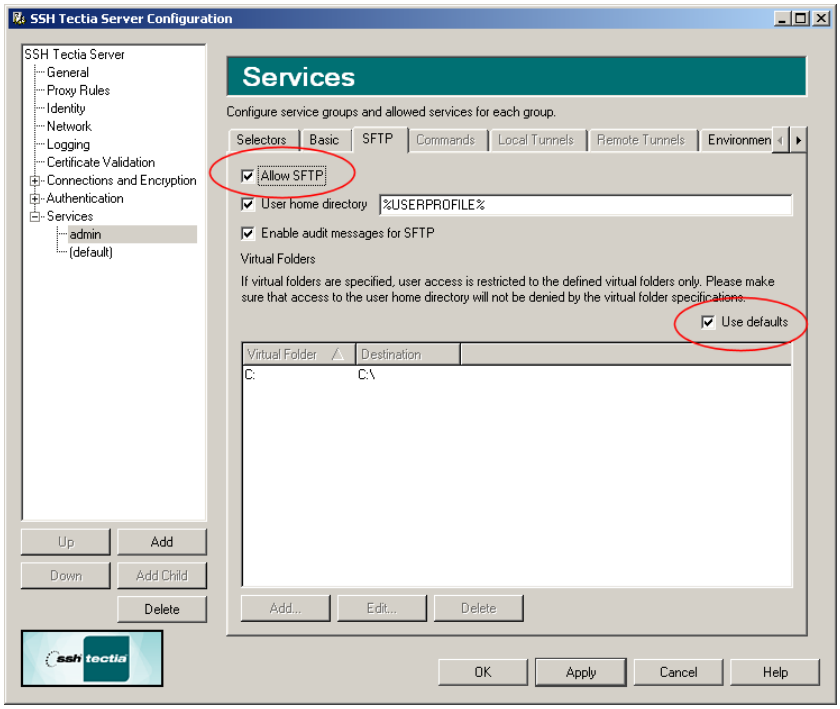

Figure 5.9. Allow SFTP for the group 'admin'

5.2.4 Settings for the SFTP-users Group

Here we create a dedicated user group for secure file transfer users. We attach an existing operating-system-related user group to the SSH Tectia SFTP group, and allow them access only to their user-specific home folders.

1. Under the **Services** page, click **Add** to create a group for SFTP users.

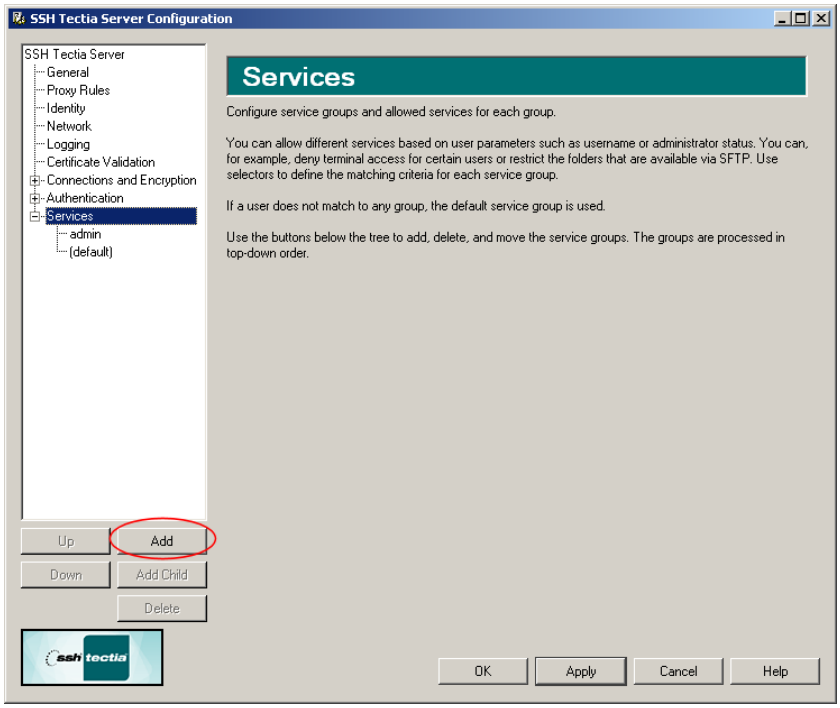

Figure 5.10. Start creating the SFTP user group

2. On the **Basic** tab, name the group `SFTP-users` and choose **Deny** or **Deny all** for all the listed services, **Terminal, Commands, Local Tunnels,** and **Remote Tunnels**. For more information on restricting terminal access, see Section 5.2.5.

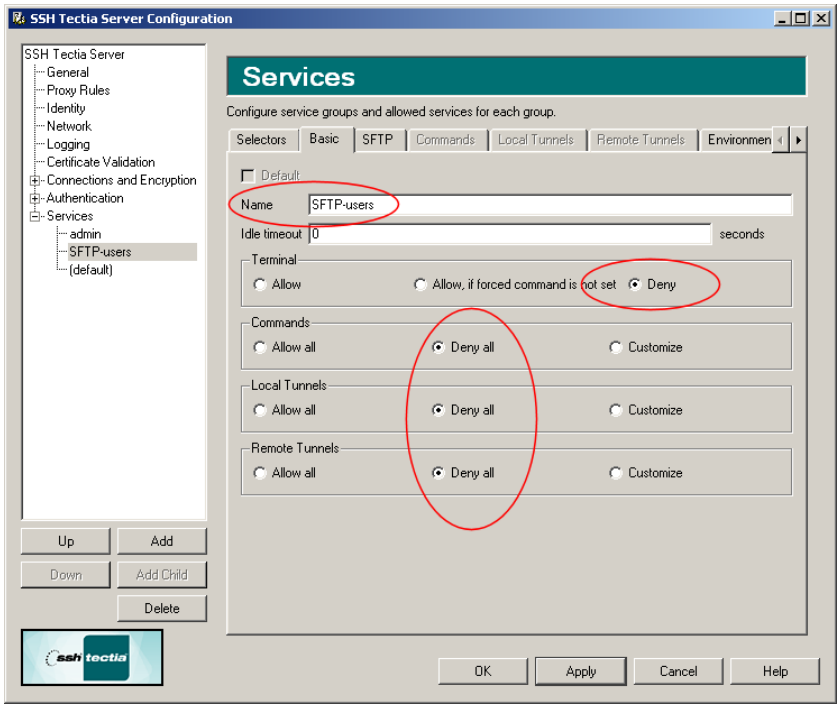

Figure 5.11. Name the group 'SFTP-users' and deny all services

3. On the **Selectors** tab, click **Add Selector** and choose the selector type **User Group**, and click **OK**.

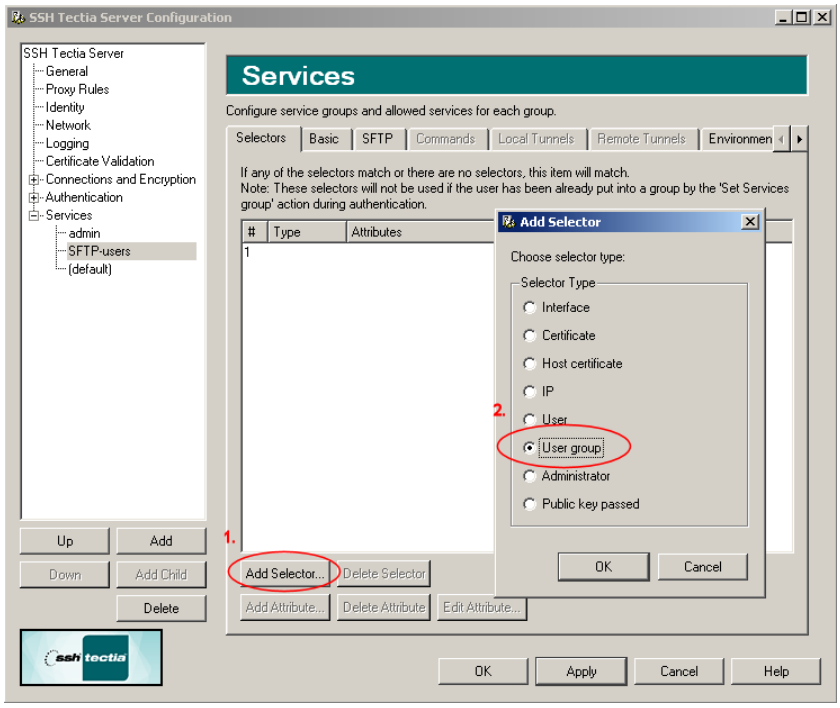

Figure 5.12. Define the group selector as user group

4. When the **User Group Selector** view opens, attach the relevant existing operating-system-related user group (named `staff` in this example) to the group.

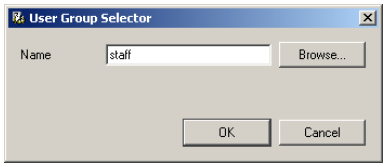

Figure 5.13. Attach user group 'staff'

Data on the newly created group selectors appears on the **Selectors** tab.

5. On the **SFTP** tab, allow the SFTP service for the `SFTP-users` and define the **User Home Directory** for the user group. This is the SFTP starting directory. Use the default `%USERPROFILES%`, as shown in the following figure.

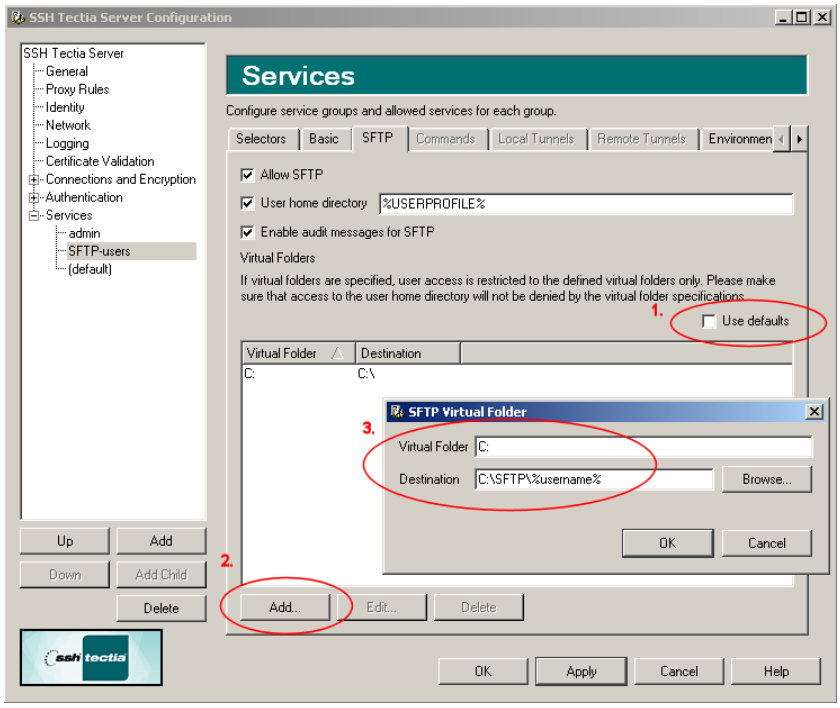

Figure 5.14. Allow SFTP service for group SFTP-users

6. To be able to define **Virtual Folders** for the user group, first un-select the **Use defaults** option on the **SFTP** tab. Then click the **Add** button. When the **SFTP Virtual Folder** dialog opens, define the virtual folder as `C:`, and its destination as the user-specific subdirectory under the `SFTP` directory on the `C:` drive (when users change directory to `C:`, they are actually directed to their user-specific SFTP directory). The session starts in the user's home directory. No other directory can be accessed via SFTP.

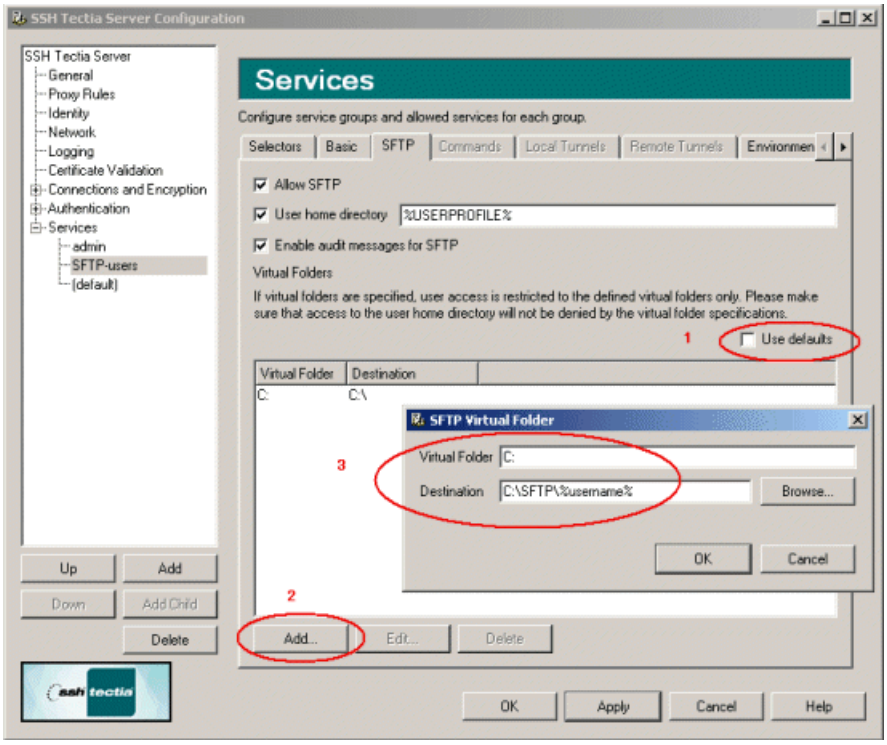

Figure 5.15. Define virtual folders for group SFTP-users

By default, file access by the user using the SFTP subsystem is restricted by the file system access controls. You can define more restrictions by defining virtual folders on Windows.

By default, if no virtual folders are explicitly defined in the configuration, the user can access all drives via SFTP and SCP2 operations, the user's SFTP session starts in the `C:\SFTP\%username%` directory, and that is the target directory for SCP2 operations.

When any virtual folders are defined, the user access is limited to the specified folders only. Note that the user's home directory must be under one of the defined virtual folders.

Note

The virtual SFTP root directory is not an actual directory on disk and no files can be written there.

The value of virtual folder can contain the same special strings as the value of home (`%username%`, `%username-without-domain%`, `%homedir%`, and `%hostname%`).

5.2.5 Settings for the Rest of Users

The `default` service settings are applied to all users who do not belong to the admin group or the SFTP group. Deny all services from them on the **Basic** and **SFTP** tabs.

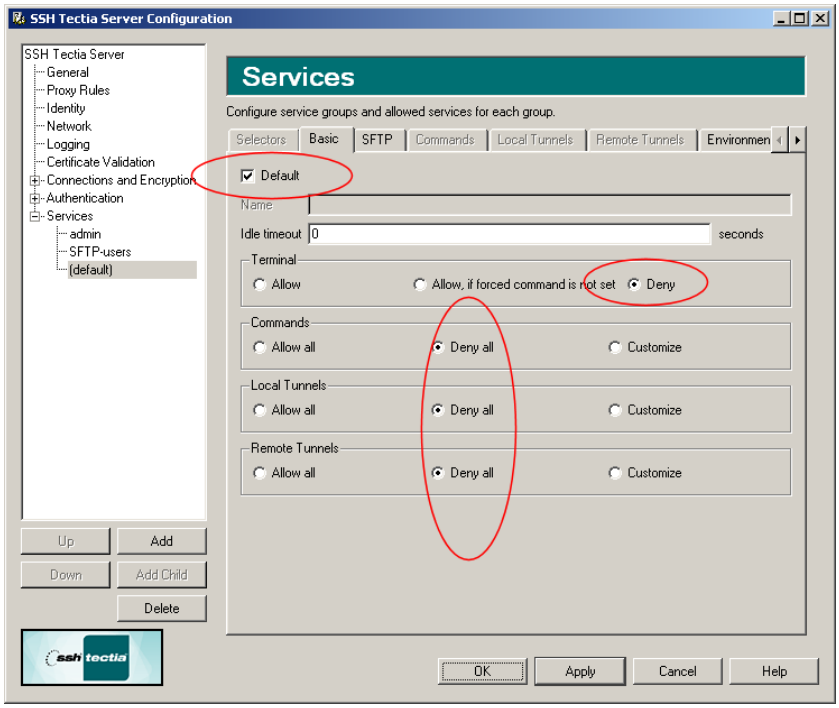

Figure 5.16. All services denied from default groups

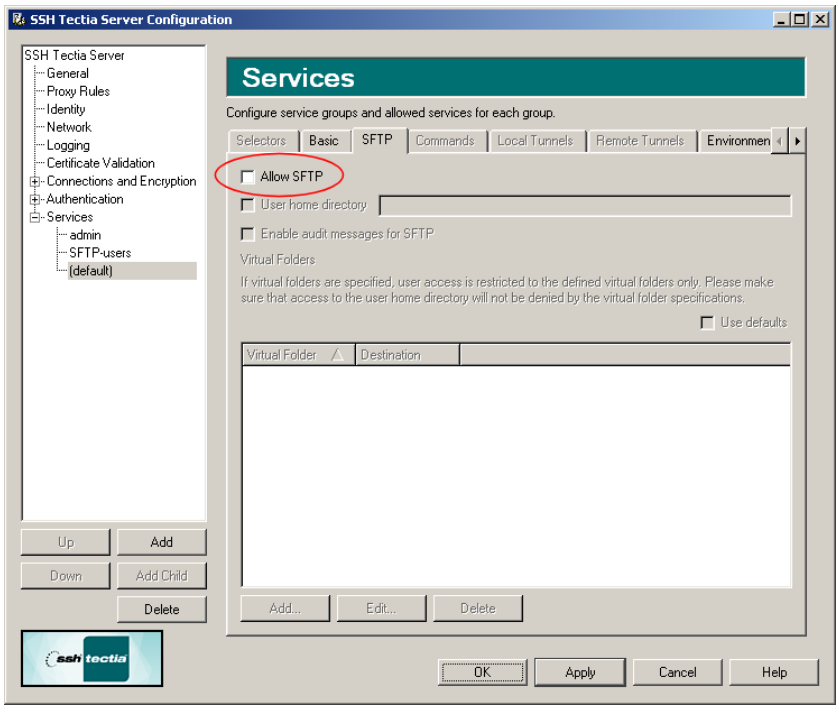

Figure 5.17. SFTP service denied from default groups

Notice that denying the terminal service, denies also X11 and agent forwarding and shell commands for the specified group (unless some commands are explicitly allowed).

5.3 Automated Secure File Transfer Script

You can set up automated file transfer between SSH Tectia Client and Server hosts using scripts.

When SSH Tectia Server is used for automated file transfer, separate user accounts can be created for the file transfer users. Non-interactive authentication with public keys and scripted commands are then set for these accounts, and the file transfers are carried out as the current user.

The following example script first transfers `testfile` from SSH Tectia Client to SSH Tectia Server and then transfers the file back. The script logs the command and the return values to a file.

```
@echo off
REM Transfer file from localhost to sftpserver.example.com and back

set SRV=sftpserver.example.com
set logfile=C:\SCP-logs\scpg3_%SRV%

echo Script started %date% %time% >> %logfile%

REM This 'scpg3 put' command transfers the file from client to server.
echo scpg3.exe -B -q testfile.dat %SRV%:test >> %logfile%
scpg3.exe -B -q testfile.dat %SRV%:test
echo Result: %ERRORLEVEL% >> %logfile%

REM This 'scpg3 get' command fetches the file from server to client.
echo scpg3.exe -B -q %SRV%:test test >> %logfile%
scpg3.exe -B -q %SRV%:test test
echo Result: %ERRORLEVEL% >> %logfile%

echo Script ended %date% %time% >> %logfile%
echo *** >> %logfile%
```


Chapter 6 Using Secure Application Connectivity

This chapter shows how to set up easy application tunneling with pre- configured automatic tunnels for secure e-mail server access. The client machine where the e-mail application is running requires SSH Tectia Client.

The tunneling capability of SSH Tectia is a feature that allows, for example, company employees to access their e-mail, company intranet pages and shared files securely even when working outside the office.

Tunneling, or port forwarding, is a way of forwarding otherwise unsecured TCP application traffic through SSH Tectia in secure encrypted format. You can secure for example POP3, SMTP, and HTTP connections that would otherwise be unsecured.

Tunneling makes it possible to access e-mail from any type of Internet service, whether accessed via modem, GPRS, 3G, a DSL line or 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.

The SSH Tectia Connection Broker takes care of the tunneling in the background. When the Connection Broker starts up, it opens the listeners for the defined automatic tunnels and asks the user to enter the password or passphrase. If the connections are authenticated with public keys that have empty passphrases, the user does not need to take any actions. The actual tunnel is formed the first time a connection is made to the listener port.

Note

The user applications using the tunnel will carry out their own authentication procedures (if any) the same way they would without the encrypted tunnel.

The automatic tunnels are local (outgoing) tunnels, which means that they protect TCP connections that your local computer forwards from a specified local port to a specified port on the remote host computer where you are connecting to. It is also possible to forward the connection beyond the remote host computer, but the connection is encrypted only between SSH Tectia Client and the Secure Shell server.

The following figure Figure 6.1 shows an example where the Secure Shell server resides in the DMZ network. The connection is encrypted from SSH Tectia Client to the Secure Shell server and continues unencrypted within the corporate network to the IMAP server.

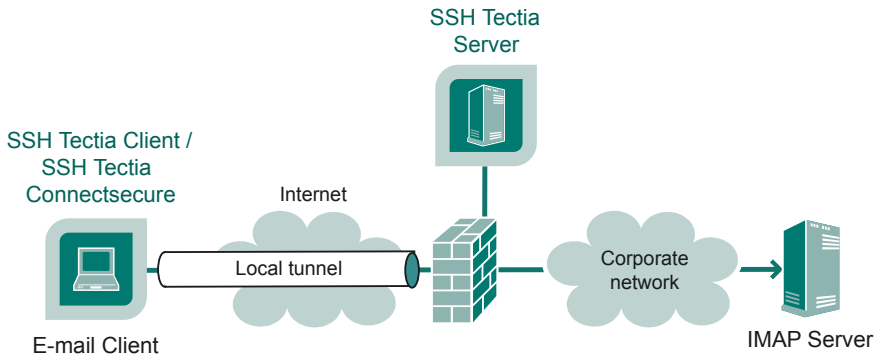

Figure 6.1. Local tunnel to an IMAP server

6.1 Defining Automatic Tunnels

Automatic tunnels are pre-configured secure connections to servers and the connections are opened automatically when SSH Tectia Client starts up (usually when the session is started). The actual tunnel is formed the first time an application connects the listener port. If the connection to the server is not open at that time, it will be opened automatically as well.

Automatic tunneling requires settings on SSH Tectia Client and on the application. For instructions on defining the automatic tunnels on SSH Tectia Client, see Section 6.1.1.

For instructions on defining the automatic tunnels on the application to be tunneled, see Section 6.1.2.

6.1.1 Settings in SSH Tectia Client

Automatic tunnels are configured through the SSH Tectia Configuration tool. Open the tool from the Windows taskbar by right-clicking the SSH Tectia tray icon and selecting **Configuration**.

Select **Automatic Tunnels** in the tree menu and click **Add** to open the **Automatic Tunnel** dialog box.

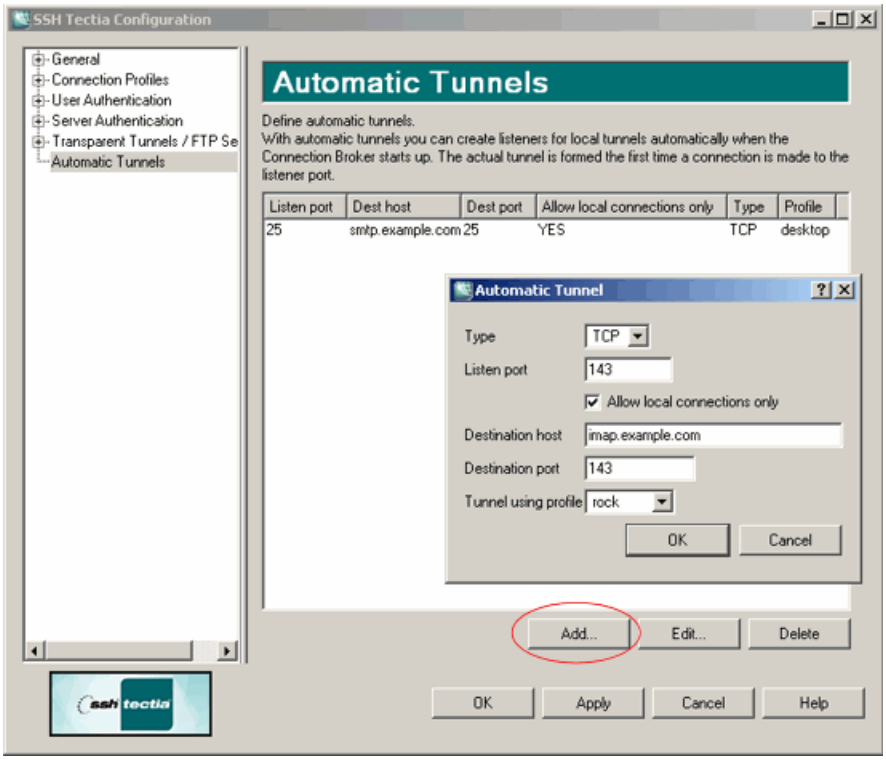

Figure 6.2. Defining a automatic tunnel

Fill in the fields as follows:

- **Type:** Select the type of the tunnel from the drop-down list. Valid choices are TCP and FTP.
- **Listen port:** Define the number of a local port that SSH Tectia Client listens to and that the applications connect to. Do not use a reserved port number.

Note

The protocol or application for which you wish to create the tunnel may have a fixed port number (for example 143 for IMAP and 25 for SMTP) that it needs to use to connect successfully. Other protocols or applications may require an offset (for example 5900 for VNC) that you will have to take into an account.

- **Allow local connections only**: Leave a check mark in this box if you want to allow only local connections to be made. This means that other computers will not be able to use the tunnel created by you. By default, only local connections are allowed. This is the right choice for most situations. You should carefully consider the security implications if you decide to also allow outside connections.
- **Destination host**: This field defines the destination host for the port forwarding. The default value is `localhost`.

Note

The destination host address is resolved after the Secure Shell connection has been established, so here `localhost` refers to the SSH Tectia Server host you have connected to.

- **Destination port**: The destination port defines the port to which the forwarded connection is made on the destination host.
- **Tunnel using profile**: Select a connection profile through which the tunnel will be created. See Section 3.2 for instructions on creating connection profiles.

To edit an automatic tunnel, select the tunnel from the list and click **Edit**.

To delete an automatic tunnel, select the tunnel from the list and click **Delete**.

6.1.2 Settings in the Tunneled Application

The application (in this example, the IMAP and SMTP e-mail) must be configured to connect to the `localhost` port instead of the application server port.

Figure 6.3 shows an example of e-mail account settings in Microsoft Outlook 2003.

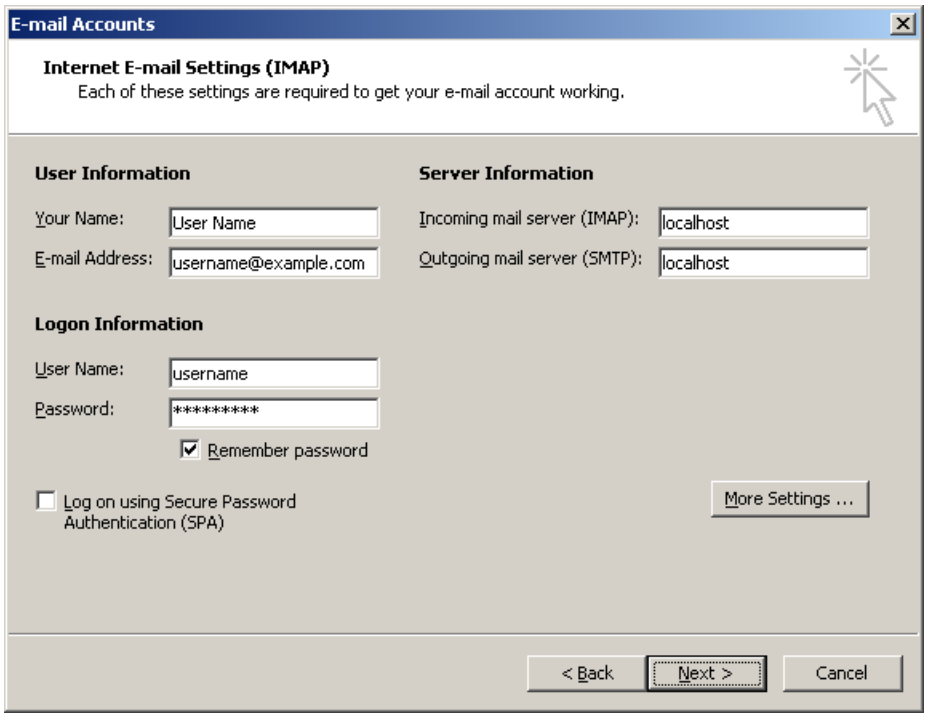

Figure 6.3. Defining e-mail settings

Now when the tunneled application connects to the `localhost` port, the connection is forwarded in encrypted format to the SSH Tectia Server, and from there unencrypted to the application server.

Index

R

S

T

U

V

W